A PRACTICAL AND SCALABLE HYBRID QUANTUM-BASED/QUANTUM-SAFE GROUP KEY ESTABLISHMENT

by

Mohammed Y. Aldarwbi

Bachelor of Computer Science, Thamar University, Yemen, 2008
Master of Computer Networks,
King Fahd University of Petroleum and Minerals, Saudi Arabia, 2017

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Ali A. Ghorbani, Ph.D., Faculty of Computer Science Arash H. Lashkari, Ph.D., Faculty of Computer Science
Examining Board:	Rongxing Lu, Ph.D., Faculty of Computer Science, Gerhard Dueck, Ph.D., Faculty of Computer Science Nicholas Touikan, Ph.D., Department of Mathematics and Statistics
External Examiner:	Qiang Ye, Ph.D. Faculty of Computer Science, Dalhousie University

This dissertation is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

July, 2023

© Mohammed Y. Aldarwbi, 2023

Abstract

This Ph.D. thesis investigates key establishment protocols, focusing on two-member and group key establishment protocols, with subcategories including classical, quantum-safe, and quantum-based solutions. I have identified research gaps such as the lack of quantum resistance in classical solutions, the inefficiency of quantum-safe solutions, and the impracticality of group-based quantum key distribution.

To address these gaps, I have proposed several novel protocols and analyses. Our first contribution is a novel quantum-safe key management scheme called KeyShield. KeyShield is a scalable and quantum-safe system that offers a high level of security. KeyShield achieves re-keying using a single broadcast message in an open channel, rather than establishing pairwise secure channels. I have also proposed another version of KeyShield, called KeyShield2, which is obtained by applying a set of countermeasures to the original KeyShield protocol after conducting cryptanalysis and identifying its vulnerabilities.

Furthermore, I have introduced two receiver-device-independent quantum key distribution protocols, QKeyShield and DGL22, based on entanglement-swapping and quantum teleportation, respectively. Both protocols minimize the attack surface, increase the key rate, and provide additional security enhancements. Security proofs and analyses demonstrate their effectiveness in establishing a secret key between Alice and Bob.

I found that group-based quantum key distribution protocols are not effective or practical due to several limitations. I have conducted a thorough literature review and proved the impracticality of these protocols. I have proposed a model that can determine the maxi-

mum number of members for which group-based protocols are useful.

Finally, I have proposed a scalable and practical hybrid group key establishment scheme. This protocol combines the power of the two-member quantum-based protocol, DGL22, to establish symmetric data encryption keys. The symmetric keys are then used to distribute the quantum-safe (KeyShield2) keying materials. The quantum-safe protocol, KeyShield2, is used to distribute a secure lock that can be opened by legitimate members using the keying materials to extract the traffic encryption key, TEK. This hybrid protocol enables a smooth transition between quantum-safe and quantum-based solutions, addressing distance limitations and pairwise channel requirements. It also allows for forward and backward compatibility.

Dedication

Dedicated to my beloved **father**, May Allah grant him Jannatul Firdaus.

His guidance and support will always be cherished.

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to Allah, for His endless blessings and guidance throughout my Ph.D. journey.

I would like to thank my family for their unwavering love and support throughout my Ph.D. journey. My mother, for her constant encouragement and for instilling in me a love for learning. My wife, for her love, patience, and understanding during the time I spent on this research. My children, for being my inspiration and motivation.

I would also like to extend my sincere appreciation to my supervisors, Prof Ali Ghorbani and Dr. Arash Lashkari, for their guidance, support, and invaluable feedback throughout the entire research process. Their knowledge and experience have been instrumental in shaping my research. I am grateful for the opportunity to have worked under their supervision, and I am deeply indebted to them for their guidance, support, and mentorship.

I am also thankful to all my colleagues and friends who have helped me in one way or the other to make this research a success. I am honored to have had the opportunity to work with such talented and dedicated individuals.

Table of Contents

Abstract ii

Dedication iv

Acknowledgments v

Table of Contents xi

List of Tables xiii

List of Figures xvii

Abbreviations xviii

1 Introduction 1

1.1 Overview 1

1.2 Problem statement 3

1.3 Contributions 6

1.4 Thesis Outline 8

2 Quantum Computing Fundamentals 10

2.1 Quantum unique features 10

2.1.1 Superposition 11

2.1.2 Nonlocality 11

2.1.3 Probabilistic Laws . . . 11
2.1.4 Uncertainty . . . 12
2.2 Hilbert space . . . 12
2.3 Qubits . . . 14
2.4 Density Operator . . . 15
2.5 Entanglement (Bell states) . . . 16
2.6 Quantum operation . . . 17
2.7 Quantum Correlation Property . . . 19
2.8 Concluding Remarks . . . 21

3 Related works **22**
3.1 Two-member key establishment . . . 23
3.1.1 Classical solution . . . 24
3.1.2 Quantum-safe . . . 25
3.1.3 Quantum-based solutions . . . 26
3.1.3.1 Main QKD Protocols types . . . 27
3.1.3.2 Entanglement-swapping-based . . . 28
3.1.3.3 Devices Independence . . . 31
3.2 Group Key distribution . . . 33
3.2.1 Classical solutions . . . 33
3.2.1.1 Pairwise key management: . . . 33
3.2.1.2 Logical Key Hierarchy: . . . 34
3.2.1.3 Broadcast Secret: . . . 35
3.2.2 Quantum-safe solutions . . . 38
3.2.3 Quantum-based solutions . . . 40
3.2.3.1 Group-based QKD challenging issue . . . 41
3.2.3.2 Available solutions . . . 47
3.3 Concluding Remarks . . . 49

4 Research Gaps and Methodology **51**

4.1 Research gaps . . . 51

4.2 Research methodology . . . 66

4.2.1 Objective . . . 67

4.2.2 Research Hypotheses . . . 68

4.2.3 Quantum-safe evaluation . . . 69

4.2.3.1 Security properties . . . 69

4.2.3.2 Performance Criteria . . . 71

4.2.4 Quantum-based evaluation . . . 73

4.2.4.1 Security definitions . . . 73

4.2.4.2 Eavesdropping strategies evaluation . . . 74

4.2.4.3 Performance Criteria . . . 76

4.3 Concluding Remarks . . . 77

5 Quantum Safe Group key Establishment **79**

5.1 KeyShield Protocol . . . 79

5.1.1 KeyShield System conditions . . . 82

5.1.2 Design objectives . . . 82

5.1.2.1 Banded linear System . . . 83

5.1.2.2 Members Grouping . . . 84

5.1.2.3 Re-keying . . . 89

5.1.2.4 KeyShiled security parameter . . . 91

5.1.3 KeyShield's Indistinguishability . . . 93

5.1.4 Security analysis . . . 94

5.1.4.1 Forward secrecy . . . 94

5.1.4.2 Backward secrecy . . . 95

5.1.4.3 Guessing/Brute-force resistance . . . 95

5.1.4.4 Grouping security analysis . . . 97

5.1.4.5 Collusion freedom . . . 98
5.1.4.6 Birthday attack . . . 99
5.1.4.7 Membership inference . . . 99
5.1.4.8 Quantum-safe . . . 99
5.1.5 Comparison with Related Works . . . 103
5.1.6 Discussion . . . 105
5.2 KeyShield2 Protocol . . . 106
5.2.1 KeyShield's Cryptanalysis . . . 108
5.2.1.1 Scalar product vulnerability . . . 108
5.2.1.2 Un-obfuscation . . . 109
5.2.1.3 Fake secret vector attack . . . 111
5.2.1.4 Updating secret vector Vulnerability . . . 113
5.2.1.5 Window Vulnerability . . . 113
5.2.2 Countermeasures . . . 114
5.2.3 Security . . . 117
5.2.4 Indistinguishability . . . 118
5.2.5 Performance . . . 121
5.2.6 Discussion . . . 121
5.3 Concluding remarks . . . 122

6 Quantum-Based Key Establishment 123

6.1 QKeyShield Protocol . . . 123
6.1.1 Error correction . . . 129
6.1.2 Time-reversed QKeyShield . . . 131
6.1.3 Key rate . . . 132
6.1.4 Finite-key security proof . . . 136
6.1.5 Eavesdropping strategies . . . 138
6.1.5.1 Entangle-measure attack . . . 138

6.1.5.2 Collective attack . . . 139
6.1.5.3 Coherent attack . . . 142
6.1.5.4 Detector blinding attack . . . 143
6.1.6 Protocol Efficiency . . . 144
6.1.6.1 Communication efficiency (σ) . . . 144
6.1.6.2 Resources efficiency (Υ) . . . 145
6.1.6.3 Key rate efficiency (δ_{fin}) . . . 149
6.1.6.4 Key establishment delay (T_{sif}) . . . 151
6.1.6.5 Detectors efficiencies . . . 152
6.1.7 Discussion . . . 153
6.2 DGL22 Protocol . . . 155
6.2.1 Security analysis . . . 160
6.2.1.1 Finite-key security . . . 161
6.2.2 Sophisticated attacks security analysis . . . 164
6.2.2.1 Detector blinding attack . . . 164
6.2.3 Protocol performance . . . 165
6.2.4 Discussion . . . 170
6.3 QKeyShield vs DGL22 . . . 171
6.4 Concluding Remarks . . . 173

7 Hybrid Quantum-Safe/Quantum-based Key Establishment 175

7.0.1 System model . . . 177
7.0.1.1 Quantum Channels . . . 178
7.0.1.2 Classical Channels . . . 180
7.0.2 Hybridizing benefits . . . 183
7.0.3 Hybrid Protocol Security . . . 185
7.0.4 Re-keying . . . 189
7.0.5 Grouping . . . 189

7.1 Performance 190
7.2 Concluding Remarks 191

8 Conclusion **193**
8.1 Future Work 196

9 Publications **199**

Bibliography **219**

Curriculum Vitae

List of Tables

3.1 Three members random actions: Randomly selecting to measure qubits on one of the basis $\boldsymbol{X}$, Z, or Y. 42

4.1 Summary of the main attacks on group key management schemes. 53

5.1 A comparison between KeyShiled and literature works 104

5.2 Summary of the main key management attacks 104

5.3 The main differences between KeyShield and Tang 105

5.4 A comparative analysis of efficiency parameters between recent quantum-safe protocols and KeyShield2. 121

6.1 Desired state preparation illustration. 127

6.2 Possible scenarios when an eavesdropper applies random operators, given that the eavesdropper's initial state is $|P^{\pm}\rangle_{DEF}$. 140

6.3 Communication efficiency σ of different entanglement-swapping-based QKD protocols along with the most recent related protocols. 145

6.4 Example of pairwise comparison: two scenarios were considered: 1) all the criteria are "equal significant" to each other, and 2) some criteria are "more significant" than others ($C4 > C2 > C5 > C1, C3, C6$). 147

6.5 Resources efficiency of different entanglement-swapping-based QKD protocols per each obtained classical bit. 147

6.6 Communication efficiency, σ, of DGL22 in comparison with the most recent related protocols. This table considers the messages sent before and during the sifting process. The messages required for error correction, information reconciliation, and privacy amplification are not included. . . 167
6.7 Comparison of key differences between QKeyShield and DGL22 RDI-QKD protocols 172

List of Figures

3.1 Taxonomy of key establishment solutions. 23

3.2 Probability of agreement, P_{agr}, as the number of members increases with respect to the random action selection probability p_1. 43

3.3 Comparison of the number of required qubits for group-based QKD and multiple instances of two-member QKD for various values of n, assuming a probability of successful measurement $p_1 = 0.5$. 45

3.4 A log-scaled illustration of the number of required qubits for the group-based QKD approach in red and the n-copy of the two-party QKD approach in blue, plotted against the number of members. 46

3.5 A visualization that shows the largest number of members for which group-QKD is more efficient than running multiple instances of two-member QKD, obtained by solving Equation 3.12 numerically using the bisection method. 46

3.6 Solutions for addressing the inefficiency of the random-based group-based QKD approach. 47

4.1 A summary of key establishment limitation. 52

4.2 The probabilistic model of the frequent re-keying problem. 64

4.3 Research methodology. 66

5.1 KeyShiled system model. 81

5.2 Banded matrix of bandwidth=1 ($b_L = b_U = 1$). 84

5.3 Finding the optimal number of subsets, s, theoretically where $n = 1000000$,$b_{LU} = 8$, w varies 88
5.4 Finding the optimal number of subsets, s, theoretically where $w = 1$, $b_{LU} = 8$, and n varies 88
5.5 Maximum size of the affected blocks resulting from updating the secret vector A_i. 91
5.6 Security parameter (A_L) visualisation. 92

6.1 QKeyShield Protocol. The diagram illustrates the steps of the QKeyShield protocol for establishing secret keys with customers. 124
6.2 QKeyShield protocol. (a) Initial state where Alice and Bob prepare a cat state consisting of N_q qubits. Alice also prepares another entangled state ($|\phi\rangle\ AB$) and sends qubit B to Bob. (b) Alice and Bob perform a BSM on $|SWA\rangle$ and $|SW_B\rangle$ which results in the creation of a larger cat state (qubits linked in red). 130
6.3 (a) QKeyShield (b) Time-reversed QKeyShield. 132
6.4 Illustration of Eve's attempt to carry out a collective attack strategy. . . . 140
6.5 Comparison of discarded results to useful ones when $q_{total} = 10000$ and the selection probability varies from $0 \leq p \leq \frac{1}{2}$. Fair random-based protocols like RDI-QKD [76] discard a minimum of 50% of the results. . 142
6.6 Comparison of expected sifting efficiencies between random-based and memory-assisted. 149
6.7 Finite Secret-key rate $\log_2(\delta_{fin})$ of the three categories shown in Figure 6.6. Parameters: $\omega = 1$, $e_{max} = 0.11$, $b_{leak} = 1.05h(e_{max})$, $\varepsilon_{EC} = 10^{(-10)}$, and $\varepsilon_{abort} = 10^{-1}$. The value of p of QKeyShield is set in such a way that $q_{total}(p^2 - \lambda) = n_1 = \Omega(logq_{total})$, where λ is a small positive number (i.e., the error due to statistical fluctuations) chosen by Alice and Bob. 151

6.8 Sifted key establishment delay. 152
6.9 An illustration of the channel disturbance (D_r) effects on the key rate. The key rates are depicted in blue and the channel disturbance rates are depicted in red. The parameters are as inFigure 6.7 except Bob's detector efficiency is $D_q \in \{0.6, 0.7, 0.8, 0.9\}$, $q_{total} = 10^5$, and the fibre channel loss is 0.2/km. 153
6.10 Visual representation of the proposed DGL22 protocol. 156
6.11 An illustration of the time-reversed fashion of DGL22. a) Alice and Bob's initial states. b) Alice and Bob start measuring qubits A and 2 then Bob performs BSM on quibits B and 1. 159
6.12 A numerical comparison between DGL22 and several recent protocols in terms of key establishment delay. $t_{qm} = 10^{-2}$ms and $t_{cm} = 10^{-1}$ ms are assumed. 167
6.13 Finite Secret-key rate of DGL22 compared with several recent protocols: Fair random-based protocols [104, 105, 158, 162], and [76]; and memory-assisted protocols [7, 74, 118], and [117]. 169
6.14 An illustration of the Bob's detector efficiency, D_q, effects on the key rate. The parameters are as in Figure 6.13 except Bob's detector efficiency is $D_q \in \{0.65, 0.75, 0.85, 0.95\}$, $N = 10^5$, and the fibre channel loss , α, is 0.2/km. 170

7.1 The proposed protocols in this thesis. 177
7.2 A random network topology of the members, with quantum channels represented by links. These channels are used to establish Symmetric Data Encryption Keys ($SDEKs$). 179
7.3 Diagram of chained nodes for delivering the group representative key to the root. 180

7.4 A network topology showing the use of classical secure channels (represented by links) to transmit encrypted secret vectors from the key server to group representatives, who then forward the vectors to the members. No intermediate nodes are required in this process. 181
7.5 A network topology showing the use of classical channels (represented by links) to broadcast the secure lock X to all members. No intermediate nodes or representative nodes are required in this process. 181
7.6 Proposed hybrid protocol. 182
7.7 Components of the group communication system. The process sequence (1-5) begins with the proposed hybrid group key establishment followed by communication. Key servers are depicted as black circles while blue circles represent members. 185
7.8 Combining the old distinguisher Γ with G^{real} yields a new distinguisher $\bar{\Gamma}$. 188

List of Symbols, Nomenclature or Abbreviations

ASC	Accredited Standards Committee
CRT	Chinese Remainder Theorem
ECDH	Elliptic Curve Diffie-Hellman
GC	Group Controller
KEM	Key Encapsulation Mechanism
SSH	Secure Shell
1SDI	One-Sided-Sevice-Independent
BSM	Bell State Measurement
DDoS	Distributed Denial of Service
DH	Diffie-Hellman
DI	Device-Independent
DoS	Denial of Service
ES	Entanglement Swapping
$\mathbb{F}_q OP$	Arithmetic Operations Over A Finite Field
GHZ	Greenberger–Horne–Zeilinger
GKE	Group Key Exchange
GKMP	Group Key Management Protocol
GQK	group representative key
IoT	Internet of Things
KS	Key Server
LKH	Logical Key Hierarchy
LWE	Learning With Errors
MDI	Measurement-Device-Independent
MQK	member representative key
NIST	National Institute of Standards and Technology
OFT	One-way Function Trees

P2M	Point-to-Multipoint
P2P	Point-to-Point
QBER	Quantum Bit Error Rate
QKD	Quantum Key Distribution
RDI	Receiver-Device-Independent
SDEK	Symmetric Data Encryption Key
SIDH	Supersingular Isogeny Key Exchange
SQM	Single Qubit Measurement
TEK	Traffic Encryption Key

$\hbar$	Planck Constant
σ_x	Standard Deviation of Position
σ_p	Standard Deviation of Momentum
$\mathcal{H}$	Hilbert Space
$\mathbb{C}$	Complex Numbers
$\|\psi\rangle$	Quantum State (Ket)
$\langle\psi\|$	Quantum State (Bra)
$\|\phi^{\pm}\rangle$, $\|\psi^{\pm}\rangle$	Bell States
Nq	Number of quits
$\|GHZ\rangle$	GHZ State (A state of three entangled qubits)
$\|Cat\rangle$	Cat State (A state of Nq maximally entangled qubits)
$\mathbb{I}$, $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{Z}$	Quantum Pauli operators
Z	Measurement Computational basis
$\boldsymbol{X}$	Measurement Hadamard basis
X	The solution of a linear system, also known as a secure lock
H	Hadamard operator
TEK	Traffic Encryption Key
$TEK^{'}$	New Traffic Encryption Key
KEK	Key Encryption Key
$KGen$	Key generation
Enc	Encapsulation or Encryption
Dec	Decapsulation or Decryption
sk	Private key
pk	Public key
n	Number of Participants/Members
k	Old Encryption Key
$k^{'}$	New Encryption Key
GK	Group Key
P_L	Polynomial
p	Probability
p_1 & p_2	The probabilities of performing the random actions
P_{agr}	Probability of agreement
R_N^{group}	Number of qubits required for a 256-classical-bit key for a group of n members using group-based QKD

$R_N^{2member}$	The number of qubits required for a 256-bit classical key for a group of n members using n-Two-party QKD
Δ	The delay in re-keying messages
T_{rek}	The maximum delay in re-keying
t_{nfy}	The time of issuance for the notification
T_{in}	The delay in joining a group
d_p	The probability of successful decryption
S_A & S_B	The bit strings of Alice and Bob
ρ_{AB}	The joint quantum state of the protocol
τ_K	The mixed state of the key
$\varepsilon_{correct}$	The threshold for the protocol's correctness failure
ε_{secret}	The threshold for the protocol's secrecy failure
ε_{secure}	The threshold for the protocol's security failure
pr	A prime number
$\mathbb{F}pr$	A finite field
A	A secret matrix (a linearly independent matrix)
A_i	A secret vector used in the protocol
B	A vector of the shared Traffic Encryption Key
b_L	The lower bandwidth of the banded system
b_U	The upper bandwidth of the banded system
q	The maximum number of non-zero entries in the banded system
A_L	The bit-length of the secret vector
a_L	The bit-length of matrix entry Aij
$\mathcal{V}$	The ground set of members where V_i is a subset
s	The number of logical (disjoint) subsets
w	The weight of a matrix or vector
$\mathcal{C}$	A linearly independent matrix
TEK_L	The bit-length of the Traffic Encryption Key
$\overline{X_j}$	The $[b_L, n - b_U]$-truncation of the X vector
R_i	A random vector used in the protocol
r_i	A random number used in the protocol
W_m	A member subspace used in the protocol
L_X	A linear mapper used in the protocol
K_m	The intersection of the nullspace of L_X and the member subspace W_m
P	A preconditioner matrix used in the protocol
q_{total}	The total number of transmitted qubits
n_1 & n_2	The number of test samples chosen for error estimation
$e^{bit-flip}$	The bit-flip error rate
e_1 & e_2	The Quantum Bit Error Rate (QBER)
q_k	The number of qubits used in key generation.
η	The percentage of utilized qubits in key generation
η_{dis}	The number of measurement results discarded

N	The total number of rounds in the protocol
η_{ES}	The number of measurements used for creating the sifted key
K_A^M & K_B^M	The raw keys of Alice and Bob
L	The length of the secret key
ε_{abort}	The probability of the protocol aborting
γ	The probability of the protocol succeed $(1 - \varepsilon_{abort})$
δ_{fin}	The finite secret key rate
δ	The asymptotic secret key rate
$H(.)$	The binary Shannon entropy
H_{min}^{ε}	The conditional smooth min-entropy
H_{max}^{ε}	The conditional smooth max-entropy
U	The unitary operator used
σ	The communication efficiency
b_s	The number of secret bits generated
q_t	The number of qubits exchanged
$\acute{b_t}$	The number of classical bits sent
T_{sif}	The delay in key establishment
Vis	The visibility of the quantum channel
Fid	The fidelity of the quantum channel
D_r	The disturbance of the quantum channel
t_{qm}	The propagation time of quantum messages
t_{cm}	The propagation time of classical messages
P_e	The error rate per clock cycle
$CNOT$	The controlled NOT gate
$\|.\|_1$	The trace norm of the system
ρ_{ideal}	The ideal system performance
ρ_{real}	The actual system performance
b_t^{Bm}	The number of classical messages required before measurements
b_t^{Am}	The number of classical messages required after measurements
α	The attenuation of the fibre
μ	The average number of photons sent
Q^{real} & Q^{ideal}	The real and ideal QKD protocol comparison
G^{real} & G^{ideal}	The real and ideal group-based protocol comparison
π	The probability of failure
Γ	A distinguisher

Chapter 1

Introduction

1.1 Overview

A cryptographic system's effectiveness critically relies upon the mystery/private cryptographic key's security. Any strong encryption technique would be jeopardized by a poor key management scheme [102]. In this sense, the encryption technique's effectiveness is dependent on the key's security, which is difficult to ensure using traditional key distribution. Over an insecure network controlled by enemies, key establishment protocols are meant to establish a shared secret key for a group of n individuals. This key can be used as a session key to allow the parties to communicate securely later. In this way, the issue of securing information is reduced to the issue of securing such keys from unauthorized access. Key establishment is essential for the effective use of cryptographic techniques. Like keys in the real world, it doesn't matter how strong your global security is if the key falls into the wrong hands. As the assets that are protected by encryption increase, the number of keys to be managed also increases. A compromised cryptographic key could prompt a huge information breach leading to reputation damage, loss of investors, and clients' confidence.

The need for a key establishment protocol over an insecure communication channel is

raised to prevent unauthorized access or accidental disclosure of the information during the transmission process between entities over a network. A critical component of the protocol architecture is the number of members that are intended to participate in a protocol's session. The majority of protocols have concentrated on the case where two members wish to establish a session key for point-to-point communications. The number of members is sometimes two. There is a large number (group) of people, devices, processes, or systems that want to communicate securely. Let us further add that the group membership is ever-changing, meaning members may join or leave at any time. Designing group key establishment schemes can be particularly challenging because of the complexity of membership changes, the computation overhead, storage cost, the message overhead, and the quantum threat. Secure group communication over open and insecure networks is required for group-oriented applications like multimedia conferencing, multi-user interactive computation, and others. For group communication to be secure, various group members must negotiate group keys. There are two types of group key establishment: group key distribution and group key agreement. In group key distribution, a trusted third party, a centralized node, is responsible for choosing a session key and distributing it to all group members. In group key agreement, all members collaboratively generate the session key.

Quantum computing is on the horizon, and i am aware that it will enable computers to perform a wide range of calculations at speeds that are almost beyond comprehension. This will enable fantastic advancements in areas like the ability to find novel materials and create brand-new life-saving medications. Unfortunately, fast quantum computers will also make it possible to quickly crack today's "unbreakable" encryption. As things stand, within the next eight to fifteen years, quantum computers pose a serious threat to the encryption that supports the security of society's critical infrastructure. Malicious actors are now intercepting encrypted data and storing it until a quantum computer is available to decrypt it. This attack is called "harvest now, decrypt later". There is a growing recognition of the need for society to prepare for increasingly sophisticated cyber

threats to the critical infrastructure and democratic institutions that rely on them. This will necessitate significant investments in cybersecurity tools, services, and skills, including those required to address the quantum threat.

Aside from several proposed quantum-based or quantum-safe two-party key establishment schemes, quantum-safe/quantum-based **group** key establishment has not received adequate attention from the researchers.

1.2 Problem statement

Designing group key establishment schemes can be particularly challenging when the group membership is ever-changing, as members may join, leave, or be evicted at any time. This dynamic nature of membership introduces complexity to the process of establishing secure group keys and can lead to various challenges such as high computation overhead, significant storage cost, large message overhead, and the potential for quantum threats. To address these challenges, researchers have proposed various solutions that can be broadly classified into three categories: classical solutions, quantum-safe solutions, and quantum-based solutions. These solutions have been extensively studied in the literature, and it is important to carefully consider their trade-offs and limitations when choosing a group key establishment scheme for a given scenario. Several challenges have been identified in the literature for classical solutions, quantum-safe solutions, and quantum-based solutions, which are as follows.

- **Classical Key establishment:**

    1. Quantum computers pose a significant threat to classical group key establishment schemes that rely on integer factorization and discrete logarithms, such as ECC, RSA, and other variations of these ciphers [10]. These cryptosystems can be quickly broken by a quantum computer, making it untrustworthy to use classical group key establishment schemes that rely on them.

- **Quantum-safe key establishment:**

    1. Quantum-safe key establishment schemes are cryptographic methods that are designed to be secure against both quantum attacks and classical attacks. These schemes are not inherently secure, but rather their security relies on the difficulty of solving a particular mathematical problem. One common approach to quantum-safe cryptography is to use mathematical problems that are believed to be hard to solve with classical computers, but easy to verify once a solution is found. This means that while it is difficult for an attacker to break the encryption, it is relatively easy for the intended recipient to verify the solution and decrypt the message.

       However, the security of quantum-safe schemes is not absolute. It is always possible that a new mathematical breakthrough or the development of a more powerful quantum computer could potentially make it easier to solve the underlying mathematical problem. As a result, it is important to continue to research and develop new quantum-safe schemes as the field of cryptography and quantum computing evolves.

    2. Quantum-safe schemes have not been fully tested and several side-channel attacks have been discovered in the 4^{th} round of the post-quantum algorithms competition held by the National Institute of Standards and Technology (NIST). For example, Kyper, which NIST considers to be the new standard, has been found to be vulnerable to several side-channel attaches, including Fault Attacks [111, 120, 121], Simple Power analysis [81, 156], Electromagnetic Attacks [120, 121, 156], Template Attacks [103], Single-Trace Attack [110], and Cold-Boot Attacks [9]. One algorithm, SIKE, was recently broken. It is important to thoroughly test and evaluate quantum-safe schemes to ensure their security against attacks.

       This highlights the need for further testing and evaluation of these schemes to

ensure their security and robustness against potential attacks.

3. One challenge in the development of quantum-safe key establishment schemes is the lack of options that are both secure and feasible for implementation in resource-constrained devices, such as those found in the Internet of Things (IoT). To meet these requirements, a quantum-safe scheme that relies on simple arithmetic operations is necessary. This can help to ensure the security of communication in resource-constrained environments while minimizing computational overhead.
4. If a group key establishment scheme is not designed to withstand attacks by a quantum computer, it could be compromised when such an attack occurs. This would leave the group communication vulnerable to eavesdropping and tampering.
5. Some proposed protocols in the literature do not adequately address the issue of key metadata management, which is important for the proper use and management of keys and can help prevent security vulnerabilities.
6. Some proposed schemes in the literature do not offer a solution to minimize the impact of the *1-affect-n* phenomenon.

- **Quantum-based key establishment:** Quantum-based key establishment methods are usually referred to as Quantum Key Distribution (QKD). Some challenges of using quantum-based methods for group key establishment include:

  1. Scalable quantum-based group key establishment schemes may not be technically or economically feasible in the near future.
  2. QKD protocols may not be able to be easily integrated with existing network protocols due to their specific requirements and the organization of networks.
  3. The current QKD network only provides point-to-point (P2P) key distribution and does not have point-to-multipoint (P2M) mechanisms.

4. QKD has limited range and is not practical for use over long distances due to high channel loss and decoherence.

5. QKD technology has some limitations that can affect its scalability. One issue is that some QKD protocols, such as BB84, have a relatively low key rate, which can make it difficult to obtain key bits when the number of users increases. Additionally, some group-based QKD protocols rely on the use of quantum memory, which is a relatively new field and may not be able to support the storage of large numbers of qubits. Finally, the use of a pre-shared key for QKD protocols, which is shared classically, has been questioned by some researchers due to potential security and other disadvantages.

6. Some proposed schemes in the literature do not adequately address the issue of key metadata management, which is important for proper key management and can help prevent security vulnerabilities.

7. Some proposed schemes in the literature do not offer a solution to minimize the impact of the 1-affect-n phenomenon (the idea that the compromise of one key can cause a disturbance to all the members' keying materials).

1.3 Contributions

The main contributions of this thesis are as follows:

1. A comprehensive literature review has been conducted. I have investigated two main categories of key establishment protocols: two-member key establishment protocols and group key establishment protocols. For each category, I have investigated three main subcategories, namely, classical solutions, quantum-safe solutions, and quantum-based solutions. I have organized and summarized their limitations and identified the research gaps.

2. A novel quantum-safe key establishment scheme is proposed, called KeyShield. KeyShield is a scalable and quantum-safe key establishment system, which offers a high-security level KeyShield achieves the rekeying using a single broadcast message in an open channel instead of establishing pairwise secure channels. KeyShield could be used to protect the three types of communication models, namely, unicast, multicast, and broadcast.

3. A new version of the KeyShield protocol, called KeyShield2, is proposed. The process of developing KeyShield2 involve conducting a cryptanalysis of the original KeyShield protocol to identify its vulnerabilities. Based on this analysis, a set of countermeasures was suggested and incorporated into the new protocol.

4. In a scenario where an organization wants to establish secret keys with its customers, the organization may invest in a reliable measuring device, but the customers may have low-cost, unsafe detectors. In this situation, two Receiver-Device-Independent (RDI) QKD protocols are proposed. In these protocols, the measuring device and entanglement sources used by Alice, the organization, are trusted, but the measuring device used by Bob, the customer, is not trusted. The first proposed protocol is called QKeyShield and is based on entanglement-swapping. The second protocol is called DGl22 which is based on quantum teleportation. Both protocols require Bell State Measurement (BSM) and Single Qubit Measurement (SQM) devices for the user. DGL22 relies on the fact that Bob has two measurement devices (BSM and SQM), while Alice has only one (SQM). Bob's SQM device is isolated to protect it from detector blinding attacks. QKeyShield and DGL22 eliminate the need for Alice and Bob to choose between alternative measurements, increasing the rate of generated bits. They also minimize the attack surface by allowing Alice to send a single qubit of an entangled state, called the shared state, to Bob, and integrating several measures to protect the qubit. Security proofs and analyses have been provided to show that the proposed protocols can securely and effectively allow Alice

and Bob to agree on a secret key.

5. During the search for a group-based QKD protocol, I discovered that these protocols are not effective or practical due to various limitations. A thorough literature research provided proof of their impracticality. As an alternative, I proposed a model suggesting that running multiple copies of two-member QKD protocols may be more effective than using a group-based QKD protocol. This model also indicates the maximum number of members for which group-based protocols may be useful.

6. A scalable and practical hybrid group key establishment scheme has been proposed. The scheme leverages the power of two-member quantum-based protocols (QKeyShield or DGL22) to establish a Symmetric Data Encryption Key (SDEK). The SDEK is used to distribute quantum-safe (KeyShield2) keying materials, also known as secret vectors. The server then uses the secret vectors and the group Traffic Encryption Key (TEK) to create a secure lock. The lock can be opened using the keying materials to extract the group TEK. This protocol combines the benefits of quantum-safe and quantum-based solutions, providing ultimate flexibility and allowing for a smooth transition between the two. It also solves the distance limitations of QKD and removes the need for KeyShield2 to use an untrusted key encapsulation mechanism for distributing secret vectors. It allows for forward and backward compatibility.

1.4 Thesis Outline

The organization of this proposal is as follows. Chapter 2 presents a review of the fundamental concepts of quantum computing. Chapter 3 provides a literature review of classical, quantum-safe, and quantum-based key establishment protocols. It includes a proposed model that indicates the maximum number of members for which group-based protocols

may be useful. Chapter 4 focuses on the research methodology, including the objective and hypothesis of the thesis, proposed system conditions, design objectives, and requirements, grouping criteria, security parameters, implementation tests, and evaluation criteria. Finally, Chapter 5, titled Proposed Schemes, presents the details of the proposed schemes that are being studied in this work. It includes a description of the schemes and their functions, as well as an analysis of their security and performance. The security analysis includes an evaluation of the schemes' vulnerabilities and the measures that have been taken to protect against potential attacks. The performance evaluation assesses the efficiency and effectiveness of the schemes in meeting the design objectives and requirements. This chapter also includes any relevant results and discussions related to the proposed schemes.

Chapter 2

Quantum Computing Fundamentals

Quantum mechanics, a subfield of physics that includes the ideas of quantization of energy, wave-particle duality, and the uncertainty principle deals with the mathematical description of the motion and interaction of subatomic particles. Quantum mechanics is a system that is governed by a wave function. In classical mechanics, objects exist at particular locations and times. Instead, according to quantum mechanics, objects have a certain probability of being at point A, a different probability of being at point B, and so on. The nice thing about quantum mechanics is that it's comparatively simple to apply it to a physical system in order to understand how it behaves, despite the fact that many aspects of it are extremely difficult to understand. Fortunately, it is not necessary to fully comprehend quantum mechanics in order to make use of it. If scientists had waited until quantum mechanics was fully understood before using it, we would still be living in the 1920s.
In this chapter, we will provide a brief introduction to quantum information that is required for understanding our contributions.

2.1 Quantum unique features

It is first important to become familiar with some fundamental details of the quantum environment in order to study quantum computing. In the following, some unique features

of quantum mechanics are introduced.

2.1.1 Superposition

A microscopic object may exist in multiple locations simultaneously. According to [62], I refer to an object as being in a superposition rather than in one position or another, meaning that it is simultaneously present in more than one location. This phenomenon affects physical characteristics such as energy, momentum, and spin, and is a consequence of the wave-particle duality of matter and energy. A superposition of states collapses into a single clearly defined state each time we look at it, or more accurately, measure it. However, it is simultaneously in a number of states before we measure it.

2.1.2 Nonlocality

To comprehend why a phenomenon occurs in a particular location, it's necessary to investigate all nearby phenomena and factors. This principle is known as locality, which means that the rules of physics function at a local level. However, quantum physics contains equations that forecast specific nonlocal phenomena, which is one of its most astonishing features. One such phenomenon is that when quantum systems are separated in space and unable to communicate, their measurements may exhibit correlations that cannot be explained by classical means, even when they share information beforehand [33]. These correlations are known as nonlocal, and their detection requires the testing of Bell inequality violation.

2.1.3 Probabilistic Laws

The laws of quantum mechanics indicate that we can only know the probability of the outcome, unlike other fields of physics where the laws are deterministic, meaning that every experiment has a unique result. The leading researchers of the time doubted it. Even

Einstein expressed his skepticism; however, the probabilistic nature of quantum mechanics is no longer in question due to numerous experimental findings.

2.1.4 Uncertainty

In the quantum world, there are intrinsic limits to the amount of information one can learn about a physical system, as per [126]. The renowned Heisenberg's uncertainty principle is the main illustration of such a restriction. The principle of uncertainty was developed in 1927 by German physicist Werner Heisenberg, thus its name. It argues that specific pairs of physical properties of a system cannot be measured with high precision at the same time because measuring one would cause the other to be disturbed. To put it another way, it limits the precision with which these pairs of attributes may be measured. This principle does not influence macroscopic things, but it is essential in the case of quantum particles. For example, a photon from a measuring instrument must collide with an electron in an atom and return to the device to measure its position. The photon, however, passes its limited momentum to the electron, modifying the electron's momentum and the uncertainty in its value [124],

$$\sigma_x \sigma_p \geq \frac{\hbar}{2}, \tag{2.1}$$

where σ_x is standard deviation of position, σ_p is the standard deviation of momentum, and $\hbar$ is the reduced Planck constant. As a result, it is impossible to accurately detect both the position and momentum of one electron.

2.2 Hilbert space

When describing a quantum mechanical system, the starting point is usually the Hilbert space H, which, according to the postulates of quantum mechanics, determines the system's state space. We begin by recalling the basic definition and some important properties.

Definition 2.2.1. A Hilbert space is a vector space over the complex numbers $\mathbb{C}$[1] equipped with an inner product

$$(\cdot,\cdot) : \mathcal{H} \times \mathcal{H} \to \mathbb{C} \tag{2.2}$$

such that the following conditions are fulfilled:

1. $(\varphi, \psi) = \overline{(\psi, \varphi)}$ for all $\varphi, \psi \in \mathcal{H}$, where the bar denotes the complex conjugate.

2. $(\varphi, \lambda\psi + \eta) = \lambda(\varphi, \psi) + (\varphi, \eta)$ for all $\varphi, \psi, \eta \in \mathcal{H}$ and $\lambda \in \mathbb{C}$[2]

3. $(\varphi, \varphi) \leq 0$ for all $\varphi \in \mathcal{H}$, and $(\varphi, \varphi) = 0$ if and only if $\varphi = 0$.

We will use the Dirac notation, bra-ket, to represent vectors in a $\mathcal{H}$. The vector symbol $|\psi\rangle$ is referred to as ket. To each vector $|\psi\rangle$ in $\mathcal{H}$, a unique dual vector $\langle\psi|$ in the dual Hilbert space $\mathcal{H}^*$, i.e. the space of linear maps from $\mathcal{H}$ to $\mathbb{C}$, corresponds. A bra is the symbol $\langle\psi|$ of a dual vector. It is worth noting that the dual of a linear combination of vectors $\alpha\,|a\rangle + \beta\,|b\rangle$ is defined as $\alpha^*\,\langle a| + \beta^*\,\langle b|$, where α^* is the complex conjugate of $\alpha \in \mathbb{C}$. The action of a linear map $\langle\phi| \in \mathcal{H}^*$ on a vector $|\psi\rangle \in \mathcal{H}$ is written as a bra-ket $|\psi\rangle \mapsto \langle\phi \mid \psi\rangle \in \mathbb{C}$ and defines the inner product of vectors $|\psi\rangle$ and $|\phi\rangle$ in $\mathcal{H}$, $\langle\phi \mid \psi\rangle \equiv (\psi, \phi)$. Two vectors are said to be orthogonal if their inner product is zero. The norm induced by the inner product is given by: $\||\psi\rangle\| = \sqrt{\langle\psi \mid \psi\rangle}$. A vector $|\psi\rangle$ is said to be normalized or called a unit vector if $\|\psi\rangle\| = 1$.

Definition 2.2.2. An orthonormal basis of a Hilbert space $\mathcal{H}$ is a set of labelled vectors $\{|e_i\rangle\}_i$ such that

$$\langle e_i \mid e_j\rangle = \delta_{ij} \tag{2.3}$$

for all i, j and every vector $|\varphi\rangle \in \mathcal{H}$ can be written as a linear combination of the basis vectors in the following way:

$$|\varphi\rangle = \sum_i \langle e_i \mid \varphi\rangle\,|e_i\rangle\,. \tag{2.4}$$

Apart from the inner product of two vectors, we can also calculate the outer product

$|\varphi\rangle\langle\psi|$, which yields a linear operator on the Hilbert space $\mathcal{H}$ and acts on a vector $|\eta\rangle \in \mathcal{H}$ in the following way:

$$(|\varphi\rangle\langle\psi|)|\eta\rangle = |\varphi\rangle\langle\psi \mid \eta\rangle = \langle\psi \mid \eta\rangle|\varphi\rangle. \tag{2.5}$$

We can make use of the outer product to derive the completeness relation for orthonormal vectors. Suppose we have an orthonormal basis $\{|e_i\rangle\}_i$ of a Hilbert space $\mathcal{H}$. We can then write a vector $|\varphi\rangle \in \mathcal{H}$ as $|\varphi\rangle = \sum_i \varphi_i |e_i\rangle$ for some set of complex numbers φ_i. Note that $\varphi_i = \langle e_i \mid \varphi\rangle$, and hence

$$\left(\sum_i |e_i\rangle \langle e_i|\right) |\varphi\rangle = \sum_i |e_i\rangle \langle e_i \mid \varphi\rangle = \sum_i \varphi_i |e_i\rangle = |\varphi\rangle \tag{2.6}$$

Since this is true for all vectors $|\varphi\rangle \in \mathcal{H}$, is follows that

$$\sum_i |e_i\rangle \langle e_i| = \mathbb{I}, \tag{2.7}$$

which is known as the completeness relation.

2.3 Qubits

The fundamental quantum system is a two-level system called quantum bit or qubit in many quantum information applications. Physical realizations of qubits are, for example, a photon that can be found in one of two distinct paths, two orthogonal polarizations of a photon, the spin state of spin- $\frac{1}{2}$ particles, or the two lowest energy levels of an electron orbiting a nucleus.

The state of a qubit is described by a density operator acting on a two-dimensional Hilbert space, $\mathcal{H}_2$. The commonly used basis for $\mathcal{H}_2$ is the computational basis $\{|0\rangle, |1\rangle\}$. Thus,

any pure qubit state is represented by a superposition of the form

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.8}$$

where $|\alpha|^2 + |\beta|^2 = 1$.

2.4 Density Operator

We have so far identified the state of a quantum system by its wave function $|\psi\rangle$, implicitly assuming that it can be completely determined. However, this is not always feasible from a practical point of view. Consider, for instance, an electron-target scattering experiment where the electron beam is prepared without polarizers. The electron spin will probably be oriented in a random direction for each beam electron. Thus, the spin of the beam cannot be described by a pure state of the form:

$$|\psi\rangle = \alpha|\uparrow\rangle_z + \beta|\downarrow\rangle_z$$

since the latter describes a spin oriented in a specific direction, fixed by the polar angles $\theta = 2\arccos|\alpha|$ and $\varphi = \arg\beta - \arg\alpha$. Rather, the state of the beam spin is described by an ensemble of spins oriented in all directions, weighted by their probability of occurrence: a mixed state.

In cases like this, where the lack of information on an ensemble of single states prevents us from describing them one by one entirely, we can still study such a collection of states statistically using the density operator, introduced by von Neumann in 1927.

Definition 2.4.1 (Density Operator)**.** Consider a quantum system that is found in one of the pure states $\{|\psi_i\rangle\}$ with probabilities $p_i < 1$, where $\sum_i p_i = 1$. Then, the state of the

system is called a mixed state and is described by the density operator

$$\rho = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|$$

The matrix representation of ρ is called density matrix, which is often used to indicate the operator itself.

2.5 Entanglement (Bell states)

A state of two maximally entangled qubits, the simplest form of entanglement, is called a Bell state, and a state of three or more maximally entangled states is called Greenberger–Horne–Zeilinger (GHZ) state. The following are known as the Bell basis in the computational basis $\{|0\rangle, |1\rangle\}$ and Hadamard basis $\{|+\rangle, |-\rangle\}$ [59]:

$$\begin{aligned}
|\Phi_Z^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi_{\boldsymbol{X}^+}\rangle = \frac{1}{\sqrt{2}}(|{+}{+}\rangle + |{-}{-}\rangle),\\
|\Phi_Z^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |\Phi_{\boldsymbol{X}^-}\rangle = \frac{1}{\sqrt{2}}(|{+}{-}\rangle + |{-}{+}\rangle),\\
|\psi_Z^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = |\psi_{\boldsymbol{X}^+}\rangle = \frac{1}{\sqrt{2}}(|{+}{+}\rangle - |{-}{-}\rangle),\\
|\psi_Z^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = |\psi_{\boldsymbol{X}^-}\rangle = \frac{1}{\sqrt{2}}(|{+}{-}\rangle - |{-}{+}\rangle),
\end{aligned} \tag{2.9}$$

where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, and the subscript $(Z, \boldsymbol{X})$ of Bell state represents the state base such as $|\Phi_Z^+\rangle$ in the computational basis and $|\Phi_{\boldsymbol{X}^+}\rangle$ on Hadamard basis. In this work, all the measurements are done in the computational basis (Z); therefore, the subscripts $(Z, \boldsymbol{X})$ are dropped to represent a Bell state indifferently of the base ($|\Phi^\pm\rangle$ and $|\psi^\pm\rangle$).

Similar to Bell state, let us define the eight GHZ basis, $|P\rangle^\pm$, $|Q\rangle^\pm$, $|R\rangle^\pm$, $|S\rangle^\pm$,

$$
\begin{aligned}
|P^{\pm}\rangle &= \frac{1}{\sqrt{2}}(|000\rangle \pm |111\rangle) \\
|Q^{\pm}\rangle &= \frac{1}{\sqrt{2}}(|001\rangle \pm |110\rangle) \\
|R^{\pm}\rangle &= \frac{1}{\sqrt{2}}(|010\rangle \pm |101\rangle) \\
|S^{\pm}\rangle &= \frac{1}{\sqrt{2}}(|011\rangle \pm |100\rangle)
\end{aligned} \tag{2.10}
$$

For generalization purposes, a state of Nq maximally entangled qubits can be called cat state $|Cat\rangle$,

$$
|Cat\rangle = \frac{1}{\sqrt{2}}(\prod_{j=1}^{Nq} |b_j\rangle \pm \prod_{j=1}^{Nq} |\bar{b_j}\rangle), \tag{2.11}
$$

where $b \subset \{0, 1\}$ and the bar above b indicates its logical negation.

2.6 Quantum operation

It is possible to transform a known quantum state into other states, and the mathematical representation is known as a quantum operation. The following are some popular quantum operations:

$$
\mathbb{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbb{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbb{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}
$$

$$
\mathbb{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{2.12}
$$

where $\mathbb{I}$, $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{Z}$ are the Pauli operators, and H is the Hadamard operator. $\mathbb{I}$ is the identity operator that keeps the state intact. The operators $\mathbb{X}, \mathbb{Y}$, and $\mathbb{Z}$, respectively,

rotate (flip) around the x, y, and z axes of the Bloch sphere by π radians, so $\mathbb{X}|1\rangle = |0\rangle$, $\mathbb{Z}|+\rangle = |-\rangle$, and $H|1\rangle = |-\rangle$. Unlike many classical operations, quantum operations are self-inverse, meaning $H = H^{-1}$, so $H|+\rangle = |0\rangle$ and $H|0\rangle = |+\rangle$. Similarly, these operations could be applied to states with more than one qubit like an entangled state, $H|00\rangle = |++\rangle$, $H|11\rangle = |--\rangle$, and $H|++\rangle = |00\rangle$,

$$
\begin{aligned}
H|00\rangle &= \tfrac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\
&= \tfrac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = |++\rangle. \qquad (2.13)
\end{aligned}
$$

The new state obtained from Eq. (2.13), $|++\rangle$, is another entangled state which is just transformed from the computational base Z into the Hadamard base $\boldsymbol{X}$. Such quantum operation could be applied to one qubit of the entangled state but not to the other, creating a new separable (disentangled) state. Let us assume the Hadamard operation, $H^{(1)}$, is applied to the first qubit (indicated by the superscripts (1)) of the entangled state $|\phi^+\rangle$,

$$
\begin{aligned}
H^{(1)}\left|\phi^{+}\right\rangle &= \tfrac{1}{2}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\
&= \tfrac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = |W^{+}\rangle. \qquad (2.14)
\end{aligned}
$$

The state $|W^{+}\rangle = \frac{1}{\sqrt{2}}(|11\rangle + |01\rangle)$ is separable as it can be noticed that the second qubit is always 1 and no correlation with the first qubit; however, performing another Hadamard operation on the second qubit re-entangle the states again, $H^{(2)}\left|W^{+}\right\rangle = |\phi+\rangle$. The following are the possible states of performing a Hadamard operation on one of the entangled state's qubits,

$$
\begin{aligned}
H^{(1)}\left|\Phi^{\pm}\right\rangle &= |W^{\pm}\rangle = \tfrac{1}{\sqrt{2}}(|\phi^{\mp}\rangle \pm |\psi^{\pm}\rangle), \\
H^{(1)}\left|\Psi^{\pm}\right\rangle &= |X^{\pm}\rangle = \tfrac{1}{\sqrt{2}}(|\psi^{\mp}\rangle \pm |\phi^{\pm}\rangle), \\
H^{(2)}\left|\Phi^{\pm}\right\rangle &= |K^{\pm}\rangle = \tfrac{1}{\sqrt{2}}(|\phi^{\mp}\rangle + |\psi^{\pm}\rangle), \\
H^{(2)}\left|\Psi^{\pm}\right\rangle &= |D^{\pm}\rangle = \tfrac{1}{\sqrt{2}}(|\phi^{\pm}\rangle - |\psi^{\mp}\rangle). \qquad (2.15)
\end{aligned}
$$

2.7 Quantum Correlation Property

Entanglement Swapping (ES) is a technique that allows two quantum systems that do not interact directly to become entangled. The beauty of entanglement swapping can be summarized in the idea of having entangled qubits that have never interacted in the past.

Let us start by briefly reviewing how this ES-based protocol works. Assume that Alice and Bob share two shared-state, $|\Phi^+\rangle_{12}$ and $|\Phi^+\rangle_{34}$, respectively. qubits 1 and 3 belong to Alice, whereas qubits 2 and 4 belong to Bob. The shared-state is $|\Phi^+\rangle$. Without loss of generality, let us assume the following two scenarios:

- Let us assume that the initial state of the two states is in similar basis (computational basis Z), $|\Phi^+\rangle_{12} = |\Phi^+\rangle_{34} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$; therefore, the quantum state of the whole system containing the qubits 1, 2, 3 and 4 can be written as:

$$\begin{aligned} |\Psi\rangle_{1234} = |\Phi^+\rangle_{12} \otimes |\Phi^+\rangle_{34} &= \frac{1}{2}(|00\rangle_{12} + |11\rangle_{12}) \\ &\quad (|00\rangle_{34} + |11\rangle_{34}) \\ &= \frac{1}{2}(|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle) \end{aligned} \tag{2.16}$$

  Once Alice performs a Bell operator measurement on qubits 1 and 3, qubits 2 and 4 will be entangled together,

$$|\Psi\rangle_{\mathbf{13}24} = \frac{1}{2}(|\mathbf{00}00\rangle + |\mathbf{01}01\rangle + |\mathbf{10}10\rangle + |\mathbf{11}11\rangle). \tag{2.17}$$

  Both Alice and Bob will end up having the same entangled states with equal probability $\frac{1}{4}$. To illustrate, If Alice measures the entangled qubits (1 and 3) and finds 00, Bob entangled qubits (2 and 4) are 00 as well.

- Let us assume that the initial states of the shared state are in different basis, first state is in the computational basis (Z) $|\Phi^+\rangle_{12} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and the other one is in the Hadamard basis ($\boldsymbol{X}$) $|\Phi^+\rangle_{34} = \frac{1}{\sqrt{2}}(|{+}{+}\rangle + |{-}{-}\rangle)$; therefore, the quantum state of the whole system containing the qubits 1, 2, 3 and 4 can be written as:

$$|\Psi\rangle_{1234} \quad = |\Phi^+\rangle_{12} \otimes |\Phi^+\rangle_{34} \quad = \frac{1}{2}(|00\rangle_{12} + |11\rangle_{12})(|{+}{+}\rangle_{34} + |{-}{-}\rangle_{34}).$$

As defined in Eq.(2.9), $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|{+}{+}\rangle + |{-}{-}\rangle)$, The whole system containing the four entangled qubits $|\Phi^+\rangle_{12}$ and $|\Phi^+\rangle_{34}$ can be written as

$$\begin{aligned} |\Psi\rangle_{1234} &= \frac{1}{2}(|00\rangle_{12} + |11\rangle_{12})(|00\rangle_{34} + |11\rangle_{34}) \\ &= \frac{1}{2}(|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle). \end{aligned} \tag{2.18}$$

Once Alice performs a Bell operator measurement on qubits 1 and 3, qubits 2 and 4 will be entangled together.

$$|\Psi\rangle_{\mathbf{13}24} = \frac{1}{2}(|\mathbf{00}00\rangle + |\mathbf{01}01\rangle + |\mathbf{10}10\rangle + |\mathbf{11}11\rangle) \tag{2.19}$$

Similar to the previous case, Alice and Bob end up with similar states even though the initial states are on a different basis.

2.8 Concluding Remarks

This chapter is a brief introduction to quantum computing. I introduced the unique features of quantum mechanics: superposition phenomena, non-locality, probabilistic laws, and uncertainty. Then I introduced several theoretical topics such as Hilbert space, qubit, operators, entanglement, Bell states, and the correlation property.

Chapter 3

Related works

Secure communication over a public network requires the encryption of transmitted messages using an encryption key. The challenge of key management, or securely delivering the encryption keys to the intended recipients, can be addressed through the implementation of key establishment protocols. These protocols establish and distribute the encryption key among authorized parties in a secure manner, thereby preventing unauthorized access to or accidental disclosure of the transmitted information during the transmission process over the network.

The number of members intended to participate in a protocol session is a crucial aspect of the protocol architecture. While many protocols are designed for point-to-point communication between two members, there are situations where a large group of people, devices, processes, or systems want to communicate securely. This can be particularly challenging because the group membership may change over time, with members joining or leaving at any time. Designing key management schemes for such groups involves addressing complexity in membership changes, computation and storage overhead, message overhead, and the potential threat of quantum attacks.

Figure 3.1 illustrates the taxonomy of key establishment protocols for both two-member and group communication. This chapter provides a review of key establishment protocols

for both types of communication.

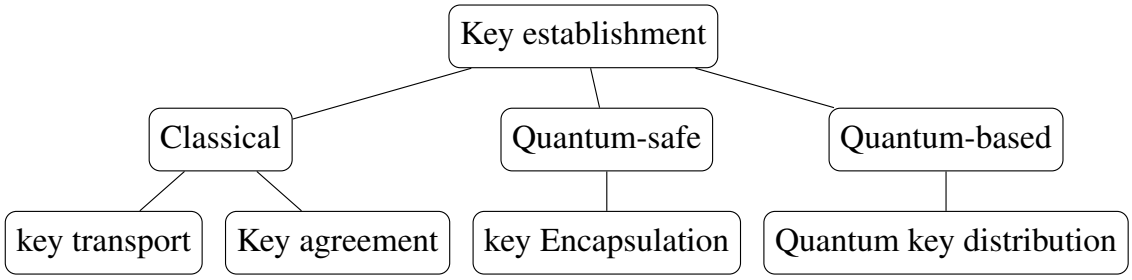


Figure 3.1: Taxonomy of key establishment solutions.

Before discussing group key establishment in detail, I will first provide an overview of key establishment in the traditional two-member scenario in Section 3.1. In Section 3.2, I will then examine the different types of group key establishment solutions, as well as their requirements and limitations.

3.1 Two-member key establishment

Key establishment techniques are methods of securely establishing a shared session key between two or more parties. The session key is a secret key that is used to encrypt and decrypt messages exchanged between the parties during a communication session. There are two main types of key establishment techniques: key transport and key agreement.

Key transport is a method of key establishment in which one party generates a session key and sends it secretly to the other party. The session key is used to encrypt and decrypt messages exchanged between the parties during a communication session. Key transport requires that the parties have a secure channel of communication for exchanging the key. There are several key transport protocols, including Kerberos and Secure Shell (SSH). In these protocols, the session key is encrypted using a pre-shared key or a public key before being transmitted to the other party. This helps to protect the session key from interception by an attacker, but it still requires the parties to have a secure channel of communication for exchanging the encrypted key.

Key agreement is a method of key establishment in which both parties jointly generate a session key based on shared information between them. The session key is used to encrypt and decrypt messages exchanged between the parties during a communication session. Key agreement ensures that the session key is new and cannot be predicted by either party, even if one of the parties is dishonest. There are several key agreement protocols, including the Diffie-Hellman key exchange and the Elliptic Curve Diffie-Hellman (ECDH) key exchange. In these protocols, the parties exchange information and use mathematical algorithms to generate a shared secret, which is then used as the session key. Because the key is not transmitted over an insecure channel, the key agreement is generally considered to be more secure than key transport.

The following subsections will examine key establishment solutions for two-member communication, which are organized according to the classification in Figure 3.1.

3.1.1 Classical solution

NIST has published recommendations for key management [14], including recommendations for key establishment techniques. According to these recommendations, two methods that are considered secure for key establishment are Discrete Logarithm (also known as Discrete-Log) and Integer Factorization.

- **Discrete-Log Schemes:** Discrete-Log is a mathematical method that involves finding the integer value of a logarithm. It is used in some public key cryptography algorithms, including the Diffie-Hellman key exchange and the El Gamal encryption system. NIST recommendation, [15], selected two specific schemes that are developed by the Accredited Standards Committee (ASC) which are: ANS X9.42 for Agreement of Symmetric Keys using Discrete Logarithm Cryptography [82], and ANS X9.63 for Key **Agreement** and Key **Transport** using Elliptic Curve Cryptography [22].

- **Integer-Factorization Schemes:** Integer Factorization involves finding the prime factors of a large composite integer. It is used in some public key cryptography algorithms, including the RSA algorithm. NIST recommendation, [16], selected specific schemes that are developed by the ASC, ANS X9.44 Key agreement using Integer Factorization Cryptography [79].

Overall, both Discrete-Log and Integer Factorization are considered secure methods for key establishment and are recommended by NIST for use in key management systems.

3.1.2 Quantum-safe

The potential of quantum computers to potentially break many current encryption algorithms has led to the development of numerous Key Encapsulation Mechanisms (KEMs) that are believed to be secure against quantum attacks. A KEM is a protocol that uses public key cryptography to establish a shared secret between two or more parties. There is concern that encrypted communication could be recorded and decrypted in the future using a quantum computer, so the use of quantum-secure KEMs is seen as a way to address this risk. However, there is also a risk that using novel, untested cryptosystems could weaken security against classical adversaries.

In response to the quantum threat, NIST has initiated a program to identify and standardize post-quantum algorithms for use in public-key encryption, key establishment, and digital signatures [2, 3]. As part of this process, NIST has held several rounds of competition to evaluate submitted algorithms. By the end of 2017, a total of 69 algorithms had been recognized as complete and suitable for the first round of evaluation, including 23 signature schemes and 59 encryption/KEM schemes [4]. In the second round, 28 schemes advanced, including nine signatures and 17 KEMs. In the third round, 15 schemes advanced, including six signatures and nine KEMs. In July 2022 [106], NIST announced the finalist algorithms for standardization, which included one KEM (Kyber) and three signatures (Dillithium, Falcon, and SPHINICS+). Four additional algorithms (McEliece,

HQC, BIKE, and SIKE) have been selected for the fourth round of evaluation.
KEM is a set of probabilistic algorithms ($KGen$, Enc, Dec) that are used to secure communication between two parties.

- The Key generation algorithm, $KGen$, produces a pair of keys - a secret key (sk) and a public key (pk).
- The encapsulation algorithm, Enc, takes the public key as input and produces a shared secret (k) and a public encapsulation.
- The decapsulation algorithm, Dec, takes the secret key and the public encapsulation as input and produces the shared secret k.

3.1.3 Quantum-based solutions

To protect against potential attacks from quantum computers on asymmetric cryptographic systems, QKD can be used as an alternative to traditional asymmetric key agreement techniques. QKD is a method for securely distributing a cryptographic key using the principles of quantum mechanics, which makes it resistant to attacks by classical computers. The first QKD protocol, called BB84, was introduced in 1984 [18]. It involves the use of two orthogonal bases for measuring quantum states, such as the vertical and horizontal polarization of photons. In 1991, Artur Ekert proposed an entanglement-based variant of QKD called E91 [51]. In this protocol, Alice and Bob share a maximally entangled state and measure its polarization at different angles. By publicly comparing the measurement results and angular orientations used, they can generate a raw key and detect any potential eavesdroppers. Overall, QKD allows two legitimate parties, Alice and Bob, to establish a secret key over a distance in a secure manner that is resistant to eavesdropping.

3.1.3.1 Main QKD Protocols types

There are two main types of QKD protocols: prepare-and-measure protocols, which are similar to BB84 [18], and entanglement-based protocols, which are similar to E91 [51].

Prepare-and-measure protocols: Prepare-and-measure protocols, such as BB84, involve the sender (Alice) preparing and sending a sequence of quantum states to the receiver (Bob). The receiver then measures the states and communicates the measurement results back to the sender. These types of protocols are used to measure unknown quantum states and can be used to detect eavesdropping, as well as determine how much data may have been intercepted.

In BB84, Alice encodes a secret key in the polarization of a string of photons and sends them to Bob. Polarization is a property of light waves that describes the direction in which the wave is oscillating. Linear polarization has two states: rectilinear and diagonal. Rectilinear polarization is further divided into horizontal and vertical states, while diagonal polarization is divided into diagonal and anti-diagonal states. These states can be represented using quantum states, such as $|H\rangle$ and $|V\rangle$ for horizontal and vertical rectilinear polarization, and $|D\rangle$ and $|A\rangle$ for diagonal and anti-diagonal diagonal polarization, respectively.

Alice randomly selects a string of bits and a string of bases (rectilinear or diagonal) of equal length. She then transmits a photon for each bit with the corresponding polarization through an optical fiber (or other channels that allow the transmission of photons) to Bob. Bob randomly chooses a basis for each photon and measures its polarization. If Bob selects the same basis as Alice for a particular photon, he will correctly determine the bit that Alice intended to share. If not, he will get a random bit. Alice and Bob then communicate over a classical public channel to compare their results. Bob tells Alice the bases he used to measure each photon, and Alice informs Bob which bases he used correctly to measure the encoded bits. They remove the encoded and measured bits that were measured in different bases, leaving them with an identical string of bits known as

the shifted key. This key can then be used for secure communication between Alice and Bob.

To check for the presence of an eavesdropper, Alice and Bob can randomly select a few bits from the sifted key and compare them over the classical public channel. If the bits are the same, it is less likely that an eavesdropper is present. If the bits are different, it is more likely that an eavesdropper is present and has intercepted the quantum channel.

Entanglement-based protocols: Entanglement-based protocols involve the use of quantum states in which two objects are linked together, forming a combined quantum state. In the E91 protocol, a single photon source is used to produce entangled photons in singlet states, given by the quantum state

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle). \tag{3.1}$$

One photon from each entangled pair is sent to Alice, and the other is sent to Bob.

They then randomly select measurement bases and measure the entangled photons they received. The measurement results are then communicated over a classical public channel, and Alice and Bob compare the correlations between their measurements. If the correlations between their measurements violate a certain mathematical limit, known as Bell's inequality, it is evidence that an eavesdropper (Eve) may be present and has intercepted the quantum channel. In this case, Alice and Bob discard the data and start the key generation process again. If the correlations do not violate the inequality, Alice and Bob can use the correlated measurement results to generate a shared secret key.

3.1.3.2 Entanglement-swapping-based

Both QKD classes, prepare-measure and entanglement-based, suffer from side-channel attacks and the use of entanglement-swapping helps in building a side-channel-free QKD [27, 75].

Entanglement swapping is a property of quantum entanglement that allows two quantum

systems that do not interact with each other to become entangled. It has been used in several QKD protocols. It helps in building a side-channel-free QKD [27, 75]. Several experimental studies have been conducted to investigate the entanglement swapping phenomenon using discrete variables (e.g., [135, 149]), continuous variables (e.g., [138]), or even a hybrid approach (e.g., [137]). For example, the authors of [135] achieved a fidelity of 84.9 ± 3.6%, which indicates a violation of the CHSH Bell inequality by more than two standard deviations. Similarly, the swapped state in [137] violated the CHSH Bell inequality by more than four standard deviations, and the authors concluded that the obtained entangled states could be directly used for QKD.

Early entanglement-based QKD protocols relied on the use of alternative measurements, in which Alice and Bob randomly alternate between several measurement bases (e.g., the Ekert protocol [51]). This approach has also been used in several entanglement-swapping-based QKD protocols [58, 59, 131, 161]. However, there are two main drawbacks to using alternative measurements. First, only a small portion of the transmitted quantum states is used to generate the key, while the rest are used for detecting eavesdropper interference, which reduces the key bit rate. Second, a quantum memory is required to store the quantum state sequences until Alice (or Bob) informs the other party about the indices of the grouped qubits and the basis on which the measurement should be performed, which makes such solutions impractical for devices with limited resources [58, 59, 161]. On the other hand, the advantage of such solutions is that they can differentiate between bit errors (related to device malfunctions or state incoherence) and quantum bit errors (related to Eve's interference).

In contrast to many other protocols, several entanglement-swapping-based protocols that do not require alternative measurements have been proposed in the literature, with the first being presented in Cabello's work [36]. This innovative idea sparked the development of successful attacks on the Cabello protocol by other authors, such as [163]. As a result, Cabello published a revised version of his protocol that included the suggestion to use the

Hadamard operation to enhance security [37]. In this revised protocol, Alice applies the Hadamard operation to the first qubit of a randomly prepared shared state, sends the other qubit to Bob, and subsequently informs Bob whether he needs to perform the Hadamard operation in the second half. The Hadamard operation cannot be performed on each state, as Eve can conduct an undetected collective attack through entanglement-swapping between the traveling qubits and their ancilla state,

$$|\Psi\rangle = H\,|\phi^{+}\rangle\, Alice\,|P^{+}\rangle\, Eve. \tag{3.2}$$

This results in Alice, Bob, and Eve sharing a maximally entangled state ($|P^{+}\rangle$). Similar to the approach outlined in [130], Cabello suggested that Alice perform the Hadamard operation on her half of a Bell state and send the other half to Bob, informing him whether she applied the operation or not. To ensure the security of the protocol, the Hadamard operation is only applied to random states. The authors of [125] demonstrated that the Chong protocol [85] is susceptible to a simplified collective attack and proposed a modified version that incorporates the benefits of the Hadamard operator in the same way as Cabello's revised protocol. The Hadamard operation must also be applied to random states in this modified version to ensure security [85].

An alternative type of entanglement-swapping QKD protocol, described in [166], involves Alice generating N Bell states and dividing the qubits of these states into two sequences. Alice keeps one sequence for herself and sends the other to Bob. Bob selects a group of qubits at random and performs Bell operator measurements on them in pairs, then informs Alice of the measurement results of the chosen qubits. Alice compares her results to Bob's by conducting Bell operator measurements on the corresponding qubits. If Alice's measurements match Bob's, she considers Bob to be legitimate and the communication can proceed. However, this protocol is vulnerable to attack by Eve, who can intercept each qubit and conduct an entanglement-swapping-based attack between the sent qubit and a prepared ancilla state. There are two main reasons why this approach is not secure. First,

Eve does not conduct any measurements during the security check stage because it is not secret, so no correlation error is introduced. Second, once Alice (or Bob) publicly announces the indices of the two qubits they have chosen to conduct Bell measurements on, Eve can choose the same qubits and follow the same procedure. As previously noted, some entanglement-swapping-based protocols are insecure due to a lack of randomization that confuses the eavesdropper [163], or due to the use of simple operators such as the bit-flip operator, which has been proven to be insecure [125]. Other protocols are inefficient because they increase the number of exchanged qubits, thereby increasing the attack surface [50]. Some protocols are not suitable for use as a conference key distribution due to the need for intermediate trusted nodes [50]. Increasing the attack surface by sending multiple qubits or introducing intermediate nodes gives Eve a greater opportunity to conduct a successful collective attack or reduces the efficiency of the protocol.

3.1.3.3 Devices Independence

QKDs can provide secure communication, but the detectors used in these systems are not always fully trusted and may be vulnerable to attacks. Many quantum hacking algorithms target the receiver's detectors [64], and there are potential vulnerabilities in real-world implementations due to flaws in QKD devices. One approach to addressing these security gaps is Device-Independent QKD (DI-QKD) [87], which does not rely on any assumptions about the internal workings of the QKD device and instead relies on the violation of Bell's inequality to provide security. In DI-QKD, the quantum devices are treated as black boxes, and no assumptions are made about the actual circuits that produce the outputs given the inputs in terms of Hilbert space, operators, or states. While DI-QKD offers a promising approach, it is currently unrealistic due to technical hurdles related to the hardware. DI-QKD also requires very high-quality detectors in order to identify loopholes in Bell tests [92]. A more practical approach is Measurement-Device-Independent QKD (MDI-QKD) [128], which does not make any assumptions about the measurement devices (detectors)

used in QKD protocols, even if the adversary can control or manipulate them [113]. Unlike other device-independent systems, MDI-QKD does not require nearly perfect detectors or a qubit amplifier. There have been significant efforts to conduct experiments on MDI-QKD in labs [143], field tests [141], and over networks [142]. MDI-QKD protocols involve a third party, Charlie, performing the measurement and sending the results through classical messages.

However, deploying MDI-QKD can be more challenging than deploying typical point-to-point QKD. One difficulty is the requirement for the interaction of two separate and indistinguishable photons, which can be challenging to achieve due to the need for the photons to arrive at Charlie's measurement device simultaneously while remaining indistinguishable in all degrees of freedom [23]. Another issue is the limited detection rate at Charlie's measurement devices, which is at most 50% of the photon detection rate assuming linear optics, and this limits the secret key rate. In addition, the finite keys analysis for MDI-QKD is less efficient than regular QKD, requiring a substantially larger postprocessing block size [23]. Finally, MDI-QKD relies on Charlie's messages, and if Charlie delays sending the results, this can make the protocol impractical for real-time applications.

Security can be ensured with fewer assumptions by relying on loose assumptions in which only the receiver's measurement device is not characterized. Some researchers have focused on the option of using uncharacterized detectors to bridge the gap between the practicality of prepare-and-measure schemes such as BB84 QKD and the enhanced security of MDI-QKD [23, 158]. These protocols are called detector-device-independent (DDI-QKD). Other researchers have focused on considering the receiver measurement device, not just the detectors, as an uncharacterized device. These protocols are called either one-sided-device-independent (1SDI-QKD) [26, 39, 154], or receiver-device-independent (RDI-QKD) [76]. Xin et al. (2020) recently published a 1SDI-QKD scheme [154] that ignores the parameter estimation stage in which Eve's presence is detected. However, this scheme discards about 75% of the protocol rounds and each round requires classical mes-

sages. Ioannou et al. (2022) recently proposed an RDI-QKD scheme [76] that operates in a prepare-and-measure fashion. However, the efficiency of this protocol is low because it discards more than half of the results.

3.2 Group Key distribution

Secure group communication is necessary for a variety of applications such as multimedia conferencing and multi-user interactive computation. In order for group communication to be secure, the group members must establish a shared secret key, called a group key. There are two main approaches to establishing a group key: group key distribution and group key agreement. In group key distribution, a trusted third party, such as a centralized node, is responsible for selecting the group key and distributing it to all group members. In group key agreement, all group members collaborate to generate the group key. There are three main types of solutions for group key establishment: classical, quantum-safe, and quantum-based. These solutions are illustrated in Figure (3.1). In the following subsections, I will discuss these three types of group key establishment solutions in more detail.

3.2.1 Classical solutions

Classical solutions for group key establishment can be divided into three main categories: pairwise key management, logical key hierarchy, and broadcast secret. These categories are described in the following subsections:

3.2.1.1 Pairwise key management:

One example of a pairwise key management scheme is the Group Key Management Protocol (GKMP) [70]. In GKMP, all messages sent by group members are encrypted using a TEK, which is known by the sender and the receivers. To ensure that all group members

have the TEK, the key server establishes a secure channel with each member and uses this channel to distribute the TEK.

To establish this secure channel, the key server shares a KEK with each group member. The KEK is used to encrypt the TEK during its distribution. When a new member, m_i, joins the group, the key server provides them with a KEK_i. The TEK is then refreshed by the key server and sent to the new member, m_i, encrypted using KEK_i, and sent to the old members encrypted using their KEK keys. The TEK is refreshed periodically and distributed to the members using their assigned KEK keys. The re-keying process in GKMP requires $O(n)$ re-key messages.

A variation GKMP is the minimal key storage scheme [86], which allocates a unique KEK to each group member. In this scheme, each member of the multicast group must store two keys: their KEK and the common Traffic Encryption Key (TEK). When a change in group membership occurs, the key server (KS) encrypts the new TEK individually with the KEKs of the remaining members. The message overhead for re-keying in this scheme is high, at $O(n)$. To minimize the amount of key storage required, the KS uses a pseudo-random function g_r with a random seed r. The KEK key of member i is given by $KEK_i = g_r(i)$. This allows the KS to avoid storing the KEKs of all n members; it only needs to store the TEK and the random number r.

3.2.1.2 Logical Key Hierarchy:

One of the most commonly used schemes in key management is the Logical Key Hierarchy (LKH) [150]. In this scheme, a key server maintains a tree of keys, with each node in the tree representing a KEK and each leaf representing a secret key that is assigned to a single member. Members hold a set of keys, including their own leaf secret key and all of the $KEKs$ corresponding to the nodes on the path from the root to their leaf. In a binary tree, each member stores at most $1 + log2(n)$ keys, where n is the total number of members. The re-keying process in LKH requires $log(n)$ re-key messages, as opposed to

$O(n)$ in the GKMP scheme.

Another tree-based key management scheme is called One-way Function Trees (OFT) [12]. OFT reduces the number of re-key messages to log_2 (n) by having members calculate the $KEKs$ rather than the key server. Each node's KEK is computed using its children's $KEKs$ using a one-way function g and the formula:

$$k_i = f(g(k_{left(i)}), g(k_{right(i)})). \tag{3.3}$$

The new version of the key called the blinded key $k^{'}$, is obtained by applying the function g to the old version of the key k. Members maintain a set of keys including their leaf key, their blinded sibling key, and the blinded sibling $KEKs$ of their ancestors.

3.2.1.3 Broadcast Secret:

Broadcast Secret, also known as secure lock, is a type of key management scheme that involves the distribution of a new key to a group of recipients through a single broadcast message. This approach reduces the need for secure channels to exchange keys and minimizes message overhead. In group key management, re-keying is typically accomplished using a secure lock. This approach has the advantage of being more efficient and requiring less overhead than other methods of key exchange.

There are several methods for creating secure locks, including the use of the Chinese Remainder Theorem, polynomial-based functions, and linear systems of equations. These methods are as follows:

Chinese Remainder Theorem: The Chinese Remainder Theorem (CRT) is a mathematical theorem that allows you to find the solution to a system of simultaneous congruences. It states that, if you have a system of linear congruences with coprime moduli, then there exists a unique solution modulo the product of the moduli. This theorem can be used to create secure locks because it allows you to create a system of equations that can only be solved if you know a certain piece of information, such as a secret key.

CRT has been used in the past to generate secure locks [45]. However, CRT has some limitations, including high computation costs and large secure lock messages. CRT works by selecting n pairwise relatively prime positive integers, $M_1, ..., M_n$, and another n integers, $K_1, ..., K_n$. A system of congruences is then formed as follows:

$$x \equiv K_1 \pmod{M_1}$$

...

$$x \equiv K_n \pmod{M_n}$$

, where M_i is a prime number known only to the system and member, K_i is the shared key, and x is the unique solution that is sent to all members in an open channel. Once the secure lock, x, is received, a member can extract the shared key K_i using the equation $K_i = X \pmod{M_i}$. While CRT has many advantages, it can also be limited by its high computation costs, large secure lock messages as the number of members increases, and its vulnerability to quantum attacks due to its reliance on the factorization problem.

There have been several other CRT-based key management schemes proposed in the literature. For example, [94] presented an asymmetric group key agreement based on CRT and the NTRU cryptosystem, and [84, 132] used hybrid CRT with LKH to construct a tree where each member holds a key and a modulus. [165] proposed two CRT-based schemes, CRGK and FCRGK, and [147] utilized CRT for distributing and updating group keys in vehicular ad hoc networks. [67] applied the generalized CRT to reduce communication costs while maintaining a high level of security.

Polynomial-based: Polynomial-based locks use polynomial functions to generate keys. These functions can be very difficult to reverse, making it difficult for someone to guess the key without knowing the specific polynomial being used. A polynomial-based scheme was first proposed by [127] for secret sharing. In [112], an intra-group polynomial-based key management scheme was proposed, let us call it *Piao*. The Group Controller (GC)

sends each member (m_i) a secret key k_i, $i = 1, ..., n$, through a secure channel. The GC generates a polynomial, P_L, over a finite field F, and broadcasts it through a public channel:

$$P_L = (x - K_i)(x - K_2)...(x - Kn) + GK \tag{3.4}$$

where n represents the number of group members, and GK is the group key generated by the GC. Once the i^{th} member receives the polynomial P_L, it computes the group key GK as:

$$GK = P_L(K_i), ; i = 1, ..., n. \tag{3.5}$$

However, several attacks on the Piao scheme have been proposed in the literature, such as [91] and [80], which can undermine trust in polynomial-based key management schemes.

Linear system of equations: In [140], a secure lock system called *Tang* is proposed using a linear system of equations. Each member generates a secret vector, A_i, and sends it to a KS. The KS combines all the secret vectors into a matrix called A and solves the linear system $A \times X = B$, where B is the shared secret key, known as the TEK.

Re-keying is typically accomplished by exchanging a single message containing the secure lock. This approach has the advantage of being more efficient and requiring less overhead than other methods of key exchange. However, this approach has several shortcomings:

1. To ensure that the columns of matrix A are linearly independent, a trial and error process is used. If they are not linearly independent, the KS must request new secret vectors from the members, which increases the number of messages exchanged.

2. The requirements for a linear system of equations to have a unique solution are not considered, despite their importance.

3. As the number of members increases, the matrix A becomes larger, and the computation cost increases.

4. A re-key event, such as when a member joins or leaves, can be disruptive as it

requires renewing both the member secret vectors and the TEK. This can affect the linear dependency of the entire system and may require all members to generate new secret vectors.

5. The storage space required increases with the number of members, as each vector has n values and each value is b bytes. No cost optimization or analysis for key storage is provided.

6. The proposed scheme has not been studied for large systems (at most seven members) and no security analysis is provided.

3.2.2 Quantum-safe solutions

With the advent of quantum computers, the security of traditional cryptographic methods has become questionable. In response, researchers have developed several post-quantum Group Key Establishment (GKE) techniques, including lattice-based and isogeny-based approaches, aimed at providing secure group communication. However, despite the promise of these techniques, their implementation and efficiency still pose significant challenges that must be effectively addressed for them to be widely adopted and utilized in practice.

- Lattice cryptography is a mathematical framework for developing cryptographic algorithms based on lattice theory. The security of lattice-based cryptographic methods is dependent on the difficulty of solving certain lattice problems, including the shortest vector problem, the closest vector problem, and the learning with errors (LWE) problem [99]. Several GKE protocols in the literature are static such as [8, 73, 107, 136]. Other lattice-based QKE schemes are not efficient as they require sending a substantial amount of broadcast messages such as [8, 46, 54], they need at least $2n$ broadcast messages where n represents the number of group members. Lattice cryptography is a mathematical framework used for the creation of cryptographic algorithms based on lattice theory. The security of these methods is based on

the complexity of solving certain lattice problems such as the shortest vector problem, the closest vector problem, and the learning with errors (LWE) problem. In the literature, several lattice-based QKE protocols have been proposed, with some being static, as in [8, 73, 107, 136]. However, some lattice-based QKE schemes are not efficient due to the large number of broadcast messages required, as demonstrated in references [8, 46, 54]. These schemes necessitate a minimum of $2n$ broadcast messages per group, where n represents the number of members. This high number of messages can increase network overhead and make the system vulnerable to distributed denial of service (DDoS) attacks.

- Isogeny-based post-quantum cryptography is an advanced field of cryptography that endeavors to secure communication against potential threats posed by quantum computers. The security of this system is founded on the mathematical principles of elliptic curve isogenies.

  Isogeny-based group key exchange (GKE) protocols have been introduced as a post-quantum secure method for group communication. In the literature, several GKE protocols have been proposed by Burmester and Desmedt [31, 32] and Steiner et al. [133] based on the two-party Diffie-Hellman (DH) protocol. However, these DH-based protocols are not secure against quantum computer attacks. To address this vulnerability, the authors of [57] proposed a GKE protocol based on the Supersingular Isogeny Key Exchange (SIDH) algorithm, which modifies the Broadcast GKA Protocol introduced by Burmester and Desmedt. It is important to note that there are fundamental differences between DH and SIDH when constructing a GKE protocol. In SIDH, the ephemeral secrets are chosen from different spaces, whereas in DH, the secrets are chosen from the same space. Furthermore, the shared secret space in SIDH is $j(E_{AB}) \in \mathbb{F}_{p^2}$, while in DH, it is $g^{r_A r_B} \in \mathbb{Z}_p$.

  Isogeny-based algorithms, such as the two-party key encapsulation algorithm SIKE GKE, have suffered from several security vulnerabilities as highlighted in studies

such as [48, 55, 97]. These vulnerabilities raise concerns regarding the trustworthiness of isogeny-based algorithms.

The current state of lattice-based and isogeny-based approaches in GKE presents several limitations such as security vulnerabilities, lack of flexibility, or inefficiency. In this study, I aim to address three crucial concerns in the field of GKE: quantum security, minimizing communication overhead, and supporting dynamic group membership changes. Group key establishment is a complex process that often requires a significant number of exchanged messages.

3.2.3 Quantum-based solutions

Quantum-based group key establishment techniques, also known as group-based QKD, conference quantum key agreement, multiparty quantum key agreement, and multimember key distribution, are protocols that allow multiple users to establish a common secret key for secure communication across a network. These protocols can be referred to as group-based QKD for simplicity throughout this thesis.

One of the early proposals for group-based QKD protocols used multipartite correlations in the n-member entangled GHZ state, represented by the equation:

$$|GHZ_n\rangle = \frac{1}{\sqrt{2}}(|00\ldots0\rangle + |11\ldots1\rangle), \tag{3.6}$$

where $|0\rangle$, $|1\rangle$ is the Z-basis, made up of the Pauli operator's eigenstates. The GHZ state has perfect correlation when all members measure in the Z-basis, making it suitable for conference key agreement protocols. However, this perfect correlation can only be achieved for $n \geq 3$. In [34], proposals for protocols using the GHZ state to establish a conference key between three members were presented and security against individual attacks was proven, however, noise resistance was not taken into account. Another study, [43], demonstrated security in conference key agreement using the distillation of GHZ

states.

A multipartite variant of the BB84 protocol was also proposed in [66], where members only need to measure in two bases, the Z-basis and the $\boldsymbol{X}$-basis. The security analysis is based on the smooth entropy uncertainty relation [146]. However, this protocol relies on the assumption that the pre-shared secret key is protected, which is a significant limitation. Another group-based QKD protocol is a multipartite version of the six-state QKD protocol [52]. The members conduct measurements in three bases $\boldsymbol{X}$, Y, and Z. The key generation rounds consist of measurements in the Z-basis, while the $\boldsymbol{X}$-basis and Y-basis are employed in fewer rounds in the test rounds to evaluate the information accessible to a prospective eavesdropper.

Many group-based QKD protocols also utilize entanglement-swapping, which allows for the entanglement of quantum systems without direct interaction among them. This technique was first introduced in the context of group-based QKD by Bose et al. [25]. Since then, several papers have been published on this topic, including [34, 89, 100, 153, 164].

3.2.3.1 Group-based QKD challenging issue

The security of QKD protocols is based on the use of random actions performed by legitimate members. These random actions can include different measurement bases, different unitary operators, or different states (normal or decoy states). When the legitimate members disagree on the chosen actions, the results are useless as they are random and the presence of an eavesdropper (Eve) cannot be detected. By discarding the disagreement, the eavesdropper can be detected by comparing an appropriate proportion of the agreed-upon measurement results. This demonstrates the importance of randomization in QKD protocols for multi-member scenarios, as it ensures that the legitimate members are able to detect the presence of an eavesdropper through the comparison of their agreed-upon measurement results.

For example, random selection is required for the N-BB84 protocol [66]. Let us consider

Table 3.1: Three members random actions: Randomly selecting to measure qubits on one of the basis $\boldsymbol{X}$, Z, or Y.

Alice	Bob	Charlie	Inferred key bit
Z	Z	Z	000 or 111
Z	Z	$\boldsymbol{X}$	Random
Z	$\boldsymbol{X}$	Z	Random
Z	$\boldsymbol{X}$	$\boldsymbol{X}$	Random
$\boldsymbol{X}$	Z	Z	Random
$\boldsymbol{X}$	Z	$\boldsymbol{X}$	Random
$\boldsymbol{X}$	$\boldsymbol{X}$	Z	Random
$\boldsymbol{X}$	$\boldsymbol{X}$	$\boldsymbol{X}$	000 or 111

the scenario in which three members, Alice, Bob, and Charlie, share a GHZ entangled state of three particles, represented by the equation:

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle\, ACB \pm |111\rangle\, ACB), \tag{3.7}$$

where Alice holds qubit A, Bob holds qubit B, and Charlie holds qubit C. They must measure their qubits on the same basis, either Z or $\boldsymbol{X}$, to share the same key bit. Table 3.1 shows all the possibilities of measuring on a random basis by Alice, Bob, and Charlie. It is clear that their inferred key bit is random most of the time unless they use the same basis. The randomness in the inferred key bit is a result of the disagreement between the members, which will be discarded. The probability of agreement among the legitimate members in the scenario we described is quite small, at $2/8$.

It's also worth noting that the results of the agreed-upon measurements are typically divided into two parts. One part is used to detect the presence of an eavesdropper, by comparing an appropriate proportion of the agreed-upon measurement results. The other part is used as the sifted key, which is further processed by error correction and privacy amplification to generate the final secret key. This division of the results ensures that even if an eavesdropper intercepts the qubits or performs measurements on them, they will not be able to infer the correct key due to the randomness in the key.

This approach of using random actions for group-based QKD is not effective due to the de-

crease in the probability of agreement between members as the group size increases. This decrease is exponential and is caused by the randomness of the chosen actions. Group-based QKD protocols typically rely on sending a large number of qubits as a finite number is not sufficient for larger groups. These protocols are usually only proven to be functional for small groups such as three or four members.

The probability of agreement between n members is given by

$$P_{agr} = p_1^n + p_2^n, \tag{3.8}$$

where p_1 and p_2 are the probabilities of performing the random actions, and $p_2 = 1 - p_1$.

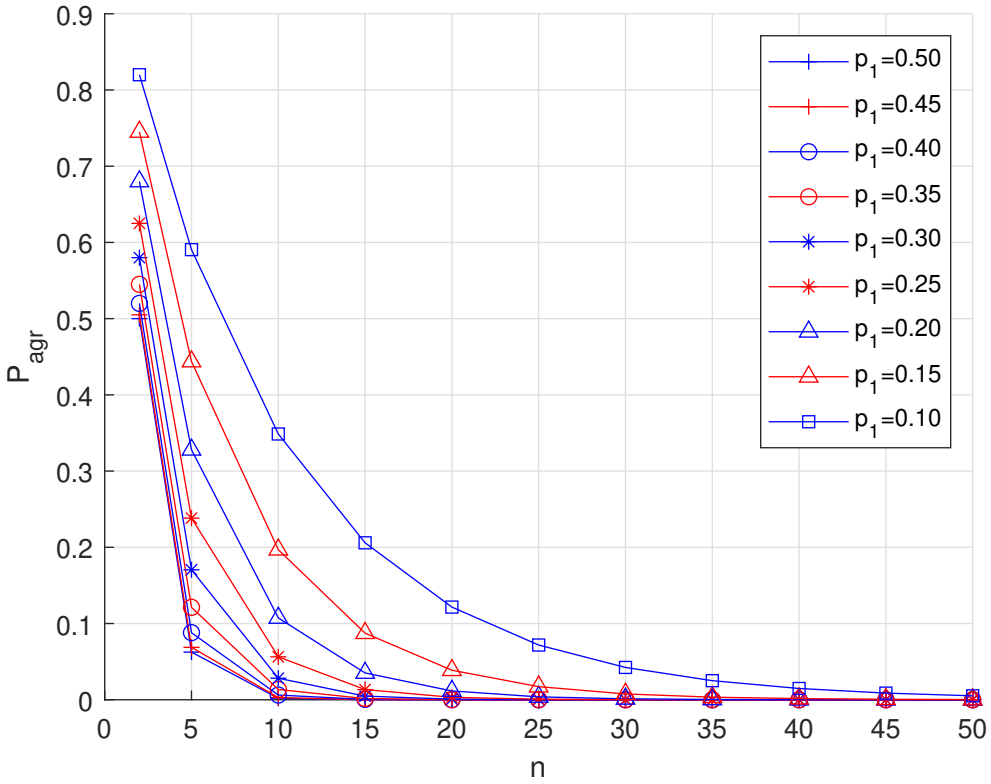


Figure 3.2: Probability of agreement, P_{agr}, as the number of members increases with respect to the random action selection probability p_1.

Figure 3.2 illustrates the relationship between P_{agr} and the number of members n, as well as the random action selection probability p_1. As the number of members increases, the

agreement probability drops significantly. The figure also highlights that p_1 plays a crucial role in determining the efficiency of the protocol. When $p_1 = 0.5$, which represents a fair probability, the agreement probability drops dramatically, for example, when $n = 10$, $P_{agr} = 0.002$, while when $p_1 = 0.2$, a biased probability, $P_{agr} = 0.107$

It is important to note that not all the agreement measurements obtained in the sifting step ultimately contribute to the final secret key. Additional steps, such as parameter estimation, error correction, information reconciliation, and privacy amplification, must be performed to further process and secure the key. These steps each consume a fraction of the sifted key, with the parameter estimation step typically consuming the largest portion. In order to further simplify the analysis, let us consider that the final secret key is equivalent to the sifted key and that the desired key length is 256 bits for the symmetric encryption algorithm AES. Additionally, it is assumed that a single qubit is sent in each protocol round to obtain a single secret classical bit, although in practice, several qubits are typically sent per round. Therefore, the number of qubits, R_N^{group}, that must be sent to obtain a 256-classical-bit key is given by

$$R_N^{group} = \frac{1}{p_1^n + p_2^n} * 256. \tag{3.9}$$

It is evident from the equation that R_N^{group} increases exponentially due to the exponential growth of $\frac{1}{p_1^n + p_2^n}$ as the number of members increases.

It is hypothesized that for a sufficiently large number of members n, where $n > \Theta$ and Θ is the threshold, using multiple instances of two-member QKD instead of a single group-based QKD protocol will improve security and efficiency. To empirically test this hypothesis, we can consider a centralized scenario where there is one Alice and $n - 1$ Bobs. Let us assume we are trying to obtain a 256-classical-bit key. In the group-based approach, the number of required qubits is given by Equation 3.9. On the other hand, if we assume that Alice initiates $n - 1$ quantum channels with each Bob, the number of required qubits,

$R_N^{2member}$, can be given by

$$R_N^{2member} = n \times \frac{1}{p_1^2 + p_2^2} * 256. \tag{3.10}$$

To determine which approach is more beneficial, we can find the intersection point of the two equations by solving Equation 3.11 for different values of n where $n \in 2, 3, 4, ..., \infty$,

$$\frac{1}{p_1^n + p_2^n} = \frac{n}{p_1^2 + p_2^2}, \tag{3.11}$$

$$\frac{1}{p_1^n + p_2^n} - \frac{n}{p_1^2 + p_2^2} = 0. \tag{3.12}$$

Let us assume that $p_1 = 0.5$, by substituting p_1 into Equation 3.11 and simplifying, we find that,

$$n = 2^{n-2}. \tag{3.13}$$

The only positive solution to this equation is $n = 4$, as illustrated in Figure 3.3. The obtained value 4 can be considered as the threshold Θ. This result validates the hypothesis

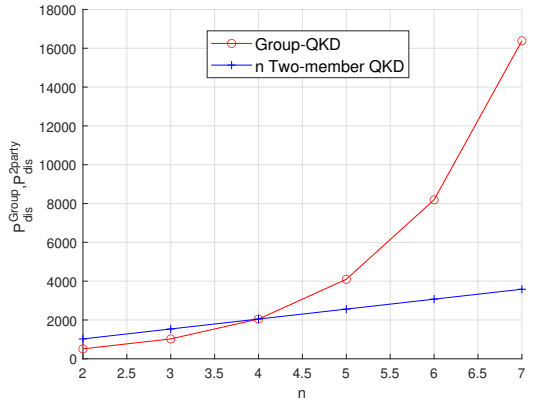


Figure 3.3: Comparison of the number of required qubits for group-based QKD and multiple instances of two-member QKD for various values of n, assuming a probability of successful measurement $p_1 = 0.5$.

that for $n > 4$, running multiple instances of two-member QKD is more beneficial than a single group-based QKD protocol. However, it is also dependent on the value of p_1.

Figure 3.4 illustrates the same comparison for different probability values.

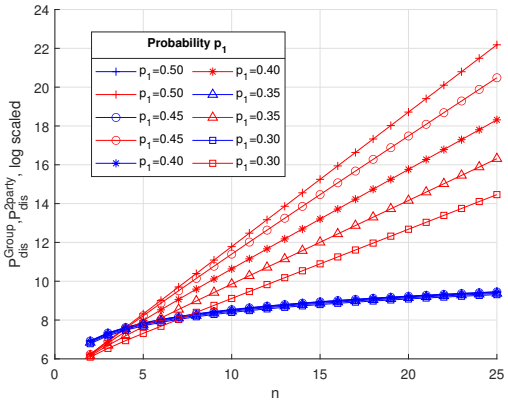


Figure 3.4: A log-scaled illustration of the number of required qubits for the group-based QKD approach in red and the n-copy of the two-party QKD approach in blue, plotted against the number of members.

To generalize the results, we can numerically solve Equation 3.12 for different values of p_1 using a simple numerical method such as the bisection method. This can give us an idea of the optimal number of members n for group-based QKD for a given value of p_1, as illustrated in Figure 3.5. It is worth noting that p_1 cannot be too small otherwise Eve's presence may not be detected, the lowest value of p_1 that is used by some biased protocols such as [148] is typically equal to 0.15, i.e. $p_1 \in 0.5, ..., 0.15$.

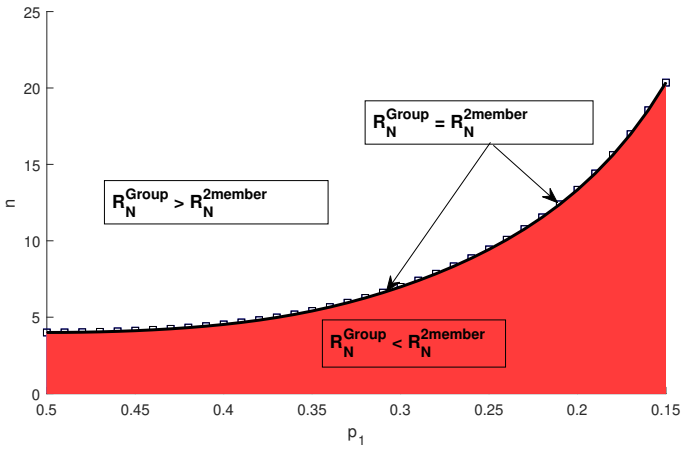


Figure 3.5: A visualization that shows the largest number of members for which group-QKD is more efficient than running multiple instances of two-member QKD, obtained by solving Equation 3.12 numerically using the bisection method.

Figure 3.5 shows an illustration of the results of solving Equation 3.12 numerically using the bisection method to determine the maximum number of members for which group-based QKD is beneficial. The area in red represents the region where group protocols are more efficient, the area in white represents the region where multiple instances of two-member QKD protocols are more efficient and the black line represents the point of intersection where both methods have the same level of efficiency. As can be seen, the random-based group-based QKD approach is only efficient for a limited number of members. For example, if we assume $p_1 = 0.15$, the maximum number of members for which group-based QKD is beneficial is around 20. This indicates that as the number of members increases, the group-based QKD approach becomes less efficient when compared to multiple instances of two-member QKD protocols.

To address this issue, various strategies have been proposed in the literature. The next subsection will explore some of these methods in more detail.

3.2.3.2 Available solutions

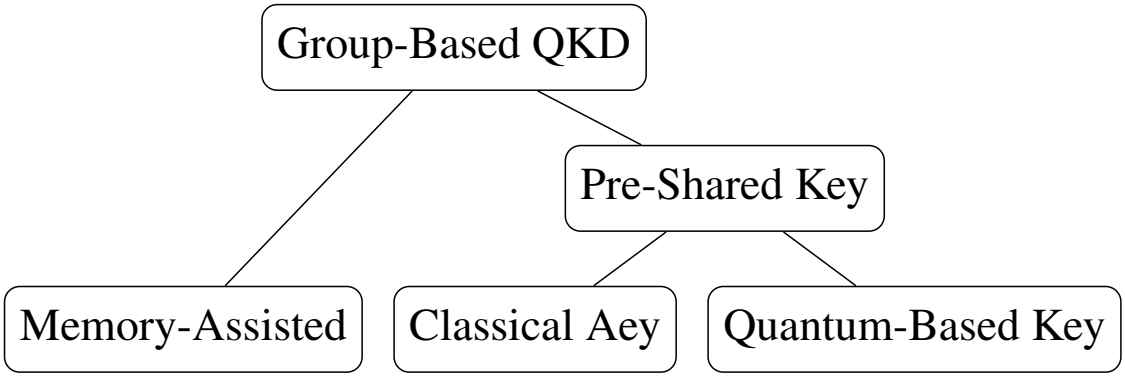


Figure 3.6: Solutions for addressing the inefficiency of the random-based group-based QKD approach.

Group-based QKD protocols that aim to improve the inefficiency of random-based group-based QKD can be categorized into two main approaches: memory-assisted and pre-shared key. Memory-assisted protocols involve one member making a random choice

and informing the others, while pre-shared key protocols rely on the prior existence of a secret shared among the participants.

Memory-assisted: In this approach, one member chooses the random action (such as measurement basis, unitary operator, or decoy states) and informs the other members about the chosen action. The recipients do not perform any random actions but rather store the received qubits and wait for the sender to inform them about the correct action. This technique increases the efficiency of the protocols dramatically, as the disagreement is minimized or eliminated. This approach have been studied extensively in [47, 60, 90, 100, 139, 153, 167].

Pre-shared: In the pre-shared key approach, the participants rely on the pre-existence of a secret that is shared between them. The shared secret tells each member about the random action required on the received qubits. This approach has been studied extensively in [30, 42, 52, 69, 88, 96, 116, 160]. This technique increases the efficiency of the protocols dramatically as the disagreement is eliminated. There are two types of these protocols: those that rely on a classical-based pre-shared key and those that rely on a quantum-based pre-shared key.

- **Classical-based pre-shared key**: Some protocols rely on the presence of a pre-shared key that is shared classically as in [30, 42, 52, 69, 96].
- **Quantum-based pre-shared key**: Other protocols generate the pre-shared key by running the quantum protocol in two different modes: a normal, inefficient random-based mode used to establish the pre-shared key and a normal, efficient pre-shared mode. This approach has been studied in several protocols such as [78, 144, 157, 159].

These approaches can increase the efficiency of group-based QKD protocols by minimizing or eliminating disagreement.

3.3 Concluding Remarks

In conclusion, this chapter has provided a detailed overview of the various key establishment techniques that have been proposed in the literature. The chapter began by discussing two-member key establishment techniques, where we looked at classical solutions, quantum-safe solutions, and quantum-based solutions in detail. We presented the main types of QKD protocols and discussed the use of entanglement-swapping-based and device-independent methods. Additionally, we also looked at the challenges and current trends in the field of two-member key establishment.

In conclusion, this chapter has provided a detailed overview of the various key establishment techniques proposed in the literature. The chapter began by discussing two-member key establishment techniques, where I examined classical solutions, quantum-safe solutions, and quantum-based solutions in detail. I presented the main types of QKD protocols and discussed the use of entanglement-swapping-based and device-independent methods. Additionally, I also explored the challenges and current trends in the field of two-member key establishment.

Next, I shifted the focus to group key distribution and presented the various classical solutions such as pairwise key management, logical key hierarchy, and broadcast secret. I also discussed the quantum-safe solutions for group key distribution and presented the current research on quantum-based solutions for group key distribution. I highlighted the challenges and open research questions in this area and provided an overview of the available solutions. I found that quantum-safe solutions are not yet mature enough, and much work is needed to develop efficient solutions. Quantum-based solutions were found to be impractical and inefficient.

Overall, the chapter has shown that key establishment is a critical aspect of secure communication and there is a significant amount of ongoing research in this area. The current research aims to develop efficient, scalable, and resilient key establishment protocols that

can be used in both two-member and group key distribution scenarios. It is clear that while classical solutions have been well-studied, there is still much work to be done in the field of quantum-safe and quantum-based fields. In the following chapter, Chapter 4, I will summarize the research gaps and present the research methodology for future work in this field.

Chapter 4

Research Gaps and Methodology

In this chapter, I identify and discuss research gaps and their causes n the field of quantum-safe and quantum-based key establishment solutions for two-member and group scenarios. In particular, the limitations of quantum-safe and quantum-based solutions for both two-member and group have been identified. Additionally, I identify other research gaps that researchers have neglected, such as Key metadata security, delay management, and frequent joining/re-keying. Furthermore, I present the thesis objectives and goals and present the research methodology, which includes the proposed system conditions, design objectives, design requirements, grouping criteria, security parameters, implementation, as well as test and evaluation criteria.

4.1 Research gaps

As discussed in the previous chapter, the key establishment can be classified into three main categories: classical, quantum-safe, and quantum-based. Exploring the state of the art indicates that there are still many open issues in two-member and group key establishment that demand efficient approaches, see Figure (4.1), for a summary of key establishment limitations.

As I work to address the challenges presented by quantum computing, two potential solu-

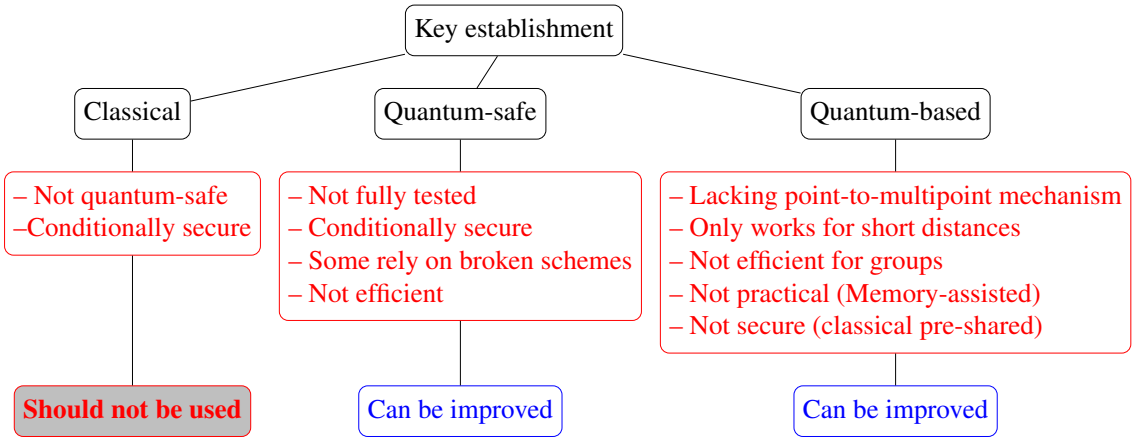


Figure 4.1: A summary of key establishment limitation.

tions stand out for their potential impact: those that are quantum-safe and those that are quantum-based. While both approaches offer promise, they each have their own unique set of challenges and limitations. Before we proceed with improving them, let's briefly summarize their current shortcomings.

1. **Classical (Two-member):** Classical solutions for two-member key establishment are not resistant to attacks by a quantum computer because they rely on mathematical problems that can be solved more efficiently on a quantum computer. Some examples of classical solutions for two-party key establishment include:

    - Diffie-Hellman key exchange: This is a widely-used method for establishing a shared secret key between two parties, but it is vulnerable to a quantum computer attack known as the quantum simulation attack.
    - RSA: This is a popular method for public key encryption, but it is vulnerable to a quantum computer attack known as Shor's algorithm which can be used to factorize large numbers and break the encryption.
    - Elliptic curve cryptography: This is another popular method for public key encryption, but it is also vulnerable to a quantum computer attack known as the quantum point multiplication attack.

These examples illustrate how classical solutions for two-party key establishment are not quantum resistant because they rely on mathematical problems that can be solved more efficiently on a quantum computer. To achieve quantum resistance, it is necessary to use quantum-safe or quantum-based solutions.

2. **Classical (group):** The classical schemes, also known as pre-quantum schemes, are cryptographic methods that have been widely used in the past but have become increasingly vulnerable to quantum computing attacks. As quantum computers continue to advance, it is becoming increasingly important to replace these classical schemes with quantum-safe alternatives. A summary of the limitations of classical approaches can be found in Table 5.2. This comparison includes several factors such as the main principle of the scheme, the scheme's ability to ensure forward secrecy and backward secrecy, resistance to quantum computing attacks, and vulnerability to other types of attacks such as collision attacks, brute force attacks, man-in-the-middle attacks, and birthday attacks.

   Table 4.1: Summary of the main attacks on group key management schemes.

	GKMP [70]	LKH [150]	CRT [45]	Polynomial [112]
Brute force attack	Yes	Yes	Yes	Yes
Quantum attacks	Yes	Yes	Yes	Yes
Collision attack	No	Yes	No	No
Man-in-the-middle	Yes	Yes	No	No
Birthday attacks	No	Yes	No	No

3. **Quantum-safe (Two-member):** The security of quantum-safe schemes is not absolute. It is always possible that a new mathematical breakthrough or the development of a more powerful quantum computer could potentially make it easier to solve the underlying mathematical problem. As a result, it is important to continue to research and develop new quantum-safe schemes as the field of cryptography and quantum computing evolves.

   Several side-channel attacks have been discovered in the 4^{th} round of NIST post-

quantum competition. For example, Kyper, which NIST considers to be the new standard, has been found to be vulnerable to several side-channel attaches, including Fault Attacks [111, 120, 121], Simple Power analysis [81, 156], Electromagnetic Attacks [120, 121, 156], Template Attacks [103], Single-Trace Attack [110], and Cold-Boot Attacks [9]. This highlights the need for further testing and evaluation of these schemes to ensure their security and robustness against potential attacks.

4. **Quantum-safe (group):** Quantum-safe group key establishment is a method for securely establishing a shared secret key among a group of members in a way that is resistant to attacks from quantum computers. There are several limitations to current quantum-safe group key establishment approaches:

    - One limitation of LWE-based schemes is that they may not be efficient as the number of group members increases and each member must transmit a larger matrix in each round, which can be resource-intensive. In some cases, the size of these messages may become so large that they have the same impact as that of a DDoS attack, disrupting the operation of the system and limiting the scalability of LWE-based schemes for use in larger groups.
    - Several schemes have been proposed that are based on the SIDH. However, authors of [119] proposed an attack that breaks the SIDH-based n-member group key exchange of [11], which was first introduced as GSIDH in [57].
    - Key metadata management is usually neglected. Key metadata is defined as information associated with a particular key that is explicitly recorded and managed by the key management scheme.
    - One of the challenges in the field of secure communication is the development of schemes that provide strong security while minimizing the need for trust in external entities. However, current quantum-safe group key establishment approaches often rely on the use of multiple trusted entities, which can intro-

duce additional vulnerabilities. One of the limitations of these approaches is that the probability of compromise increases as the number of trusted entities increases. This is because each additional entity introduces another potential point of failure, increasing the overall risk of the system. As such, it is important to strive for solutions that minimize the number of trusted entities, in order to reduce the risk of compromise.

- The *1-affect-n* issue in group key establishment refers to the fact that any change in membership of the group can have an impact on the security of the entire group, as it requires the regeneration and redistribution of the TEK. If this process is not properly managed, it can result in the need to change the member's keying materials more frequently and increase the number of communication messages required. It is important to implement robust key management protocols to effectively handle changes in membership and maintain the security of the group's communication. The current state of the art lacks a novel solution that effectively minimizes the impact of the *1-affect-n* phenomenon.

5. **Quantum-based(two-member):** QKD is a method of securely distributing cryptographic keys using the principles of quantum mechanics, which provides unconditional security for the key distribution process. This means that the security of the key distribution is not based on any computational assumptions or unproven mathematical assumptions, and is instead based on the fundamental laws of quantum mechanics. It is important to consider the balance between security and efficiency when selecting a QKD protocol, as it is generally desirable to have a protocol that is both secure and efficient. Despite QKD security advantages, there are several challenges to implementing QKD, including limited range, lack of scalability, and the high cost of building new infrastructure.

- The critical challenge is achieving long-distance communication, which is hindered by high channel loss and decoherence, with a detection rate of only 30% at a distance of 1000 km and a transmission loss of approximately 90% for every 50 km traveled in optical fibers [1].
- Another challenge in implementing QKD is scalability. Some QKD protocols, such as BB84, have a key rate of only 25%. This means that their ability to generate a key bit is limited, and as the number of members increases, the possibility of obtaining a key bit may vanish. This scalability issue presents a significant challenge in terms of the practicality and usefulness of QKD.
- Some QKD protocols may be less efficient and more vulnerable to attack than others due to the number of exchanged qubits. As the number of exchanged qubits increases, the attack surface of a QKD system also increases. This refers to the extent to which the system is exposed to potential attacks, such as attempts to intercept and compromise key information. The larger the attack surface, the more vulnerable the system is to attacks.
- The use of intermediate nodes in QKD can make the protocol less practical for several reasons. One reason is the potential vulnerability introduced by the intermediate nodes. If an intermediate node is compromised, it may be possible for an attacker to intercept and compromise the key information, undermining the security of the communication. This can make the QKD protocol less practical and useful. Another reason why the use of intermediate nodes can make QKD less practical is that it may require the deployment of a large number of intermediate nodes. This can be inefficient and impractical in establishing group keys, as it may require a significant investment in hardware and infrastructure. Additionally, the large number of intermediate nodes may increase the attack surface of the QKD system, making it more vulnerable to attacks. Overall, the use of intermediate nodes in QKD can introduce vulnerabilities

and reduce the practicality of the protocol.

6. **Quantum-based(group):** One of the main characteristics of QKD is that it is a point-to-point communication mechanism, which means that it is typically used to establish a secure connection between two specific parties. The point-to-point nature of QKD makes it well-suited for certain applications, such as establishing secure communication between two parties who need to exchange sensitive information.

   This is in contrast to point-to-multipoint communication, which involves establishing a connection between a single sender and multiple receivers. QKD is less practical for applications where multiple parties need to communicate with each other simultaneously, as it is generally not possible to transmit quantum states (qubit) to multiple recipients simultaneously. This is due to the impossibility of cloning quantum states. Additionally, quantum states are inherently fragile and are easily disturbed by the environment.

   To address this challenge, researchers have explored the development of group-based QKD protocols, which allow multiple parties to establish a shared secret key using quantum communication techniques. There are three main types of group-based QKD protocols, namely, random-based, memory-assisted, and pre-shared key based. Chapter 3 discusses these types. The identified limitation of each type are as follows:

   - **Random-based limitations:** Random-based QKD protocols are not typically efficient, particularly for groups with a large number of participants. As the group size increases, the probability of agreement between participants decreases, and the number of exchanged qubits required to establish a shared secret key also increases. This can make it challenging to use random-based QKD protocols in large groups, as it can be resource-intensive and may not be practical in certain situations.

- **Memory-assisted limitations:** One major limitation of memory-assisted quantum key distribution (QKD) protocols is the impracticality of storing a large number of qubits required for each member of the group. The fragility of quantum states and the need for environmental control make storing qubits challenging, which can hinder the implementation of memory-assisted QKD protocols on a large scale.

  In addition to the above limitation, there are several challenges associated with these protocols:

  (a) Slow speed due to the need for classical messages and quantum memory.

  (b) Quantum channel loss, which occurs when photons carrying quantum information are lost during transmission over long distances due to absorption, scattering, or other factors.

  (c) Decoherence, which is the loss of quantum information when a quantum system interacts with its environment. Maintaining the integrity of quantum states during storage is a significant challenge due to decoherence.

  (d) Technological demands and error-prone maintenance of a large number of coherent quantum states over an extended period.

- **Pre-shared key:** A type of group-based QKD protocol that relies on the existence of a pre-shared key in order to establish a shared secret key among the members of the group. The pre-shared key may be generated using classical methods or quantum methods.

  If the pre-shared key is generated using classical methods, it may be a security risk as it may be vulnerable to quantum attacks. This can undermine the security of the QKD protocol, as the shared secret key that is established using the pre-shared key may also be compromised.

  If the pre-shared key is generated using quantum methods, it may be generated using either random-based QKD or memory-assisted QKD. Both of these types

of QKD protocols have their own limitations, which can impact the scalability and practicality of the QKD protocol.

Overall, the use of a pre-shared key in QKD protocols can present both security and efficiency challenges, depending on how the pre-shared key is generated.

There are several features that researchers have neglected in the development of group-based QKD protocols. These features include:

1. **Key metadata:** Key metadata refers to information about the key, such as its size, format, constraints, parameters, and usage history. Ensuring the security of key metadata is important, as it can be used to infer information about the key itself or to identify patterns in the way the key is used. Researchers have neglected to consider the security of key metadata in many group-based QKD protocols, which can leave the key vulnerable to attack.

2. **Group management:** Group management is a crucial aspect of group-based key establishment protocols. It involves controlling access to a group, including deciding who can be added or removed from the group and who can coordinate these operations. Ideally, group admission control should be based on a local group policy that is independent of the key establishment protocol being used. However, it is also necessary to have a central point of control for group membership operations, such as adding or excluding members, to coordinate and synchronize these actions and prevent chaos. The responsibility for this role can be changed as needed and is largely a matter of policy.

   When membership changes occur, it is necessary to update the session keys accordingly and implement auxiliary procedures to reflect these changes. There are two types of membership changes: member addition and member exclusion. Both require the following security measures:

- Old group keys used in previous sessions must not be disclosed to new members. This means that a new member must not have access to or be able to calculate keys used before they joined the group.
- New keys must be kept secure and out of reach of unauthorized group members. This means that members who have been excluded from the group in the past should not have access to new keys unless they are re-admitted to the group.

Bulk membership changes, including bulk addition (such as the addition of multiple new members to an existing group, or the merger of two groups) and bulk exclusion (such as the exclusion of multiple members at the same time, or the partition of a group into smaller groups) also need to be considered. While the actual protocols for handling these bulk changes may differ from those for single-member changes, the key security requirements (key independence) remain the same.

It is worth noting that, in some cases, it may be desirable to forgo the requirement of key independence in order to allow late-joining participants to catch up with ongoing group communication. This decision should be made based on local group policy.

3. **Join/Leave/Evict delays:** In group communication, messages are encrypted using a single key that is known to all group members. In order to provide forward-secrecy, the key should be renewed whenever a new device joins the group. Similarly, in order to provide backward-secrecy, the key should be renewed whenever a device leaves or is evicted from the group.

   However, it is not always possible for all utility systems to be instantly aware of a join, leave, or evict event. There may be a delay before the event is known to all group members. To mitigate the effects of this delay, it is necessary to establish a maximum threshold of time after which a join, leave, or evict event is known to all group members.

Let T_{in} be the join delay and T_{out} be the leave/evict delay. If at time t_0 a join event is issued, it will take effect at time t, such that $t_0 < t \leq t_0 + T_{in}$. Similarly, if at time u a leave/evict event is issued, it will take effect at v, such that $u < v \leq u + T_{out}$. This ensures that the key is renewed in a timely manner and that the group communication remains secure.

4. **Re-keying delay:** In a group communication, when a decision is made to replace the old key k with its new version $k^{'}$, each group member should receive a notification. This is necessary in order to ensure that all group members are using the most up-to-date key for encrypting and decrypting messages.

   However, due to large number of members in the group, it can take some time for the re-keying message to be delivered to all group members. This time is referred to as the re-keying delay, which is denoted as T_{rek}. In order to calculate the re-keying delay, the time elapsed to complete the delivery of the re-keying message to each group member ($\Delta_1, \Delta_2, ..., \Delta_n$) is considered, and the maximum delay is taken as the re-keying delay: $T_{rek} = \max\Delta_1, \Delta_2, ..., \Delta_n$.

   Overall, the re-keying delay is an important factor to consider in group key establishment, as it can affect the security and effectiveness of communication. It is important to ensure that the re-keying message is delivered in a timely manner in order to maintain the security of the communication.

5. **Re-keying Notifications loss:** Using the Internet as a way of communication increases the probability of packet loss, which can affect the security and effectiveness of group communication. When a group re-keying notification is sent using a connection-less protocol such as UDP, it is cumbersome to initiate a TCP connection between the sender and all recipients. However, this means that there is a probability that the notification will not be received by some of the recipients.

   As a result, some group members may continue to use the old key k, while others are

using the new key k'. This can compromise the security of the communication, as messages encrypted with the old key may be vulnerable to attack. The notification loss probability may be higher in networks with high churn, where a large number of smart devices frequently join or leave the network. This can affect key management and increase the risk of communication being compromised [24].

Overall, it is important to carefully consider the impact of packet loss and high churn on the security and effectiveness of group communication in order to ensure the safety and integrity of the communication.

6. **Frequent joining/re-keying:** We assume that all group members receive the re-keying notification, but with a delay. In an ideal state, a device would continue to communicate with the utility using the current key k until the crypto-period expires or a new key k' is received. However, in the case of a change in group membership (such as a device joining or leaving the group, or being evicted), the group key should be changed to ensure forward and backward secrecy. High churn, or frequent joining or re-keying events, can make it difficult to maintain the ideal state, as the need for frequent re-keying can shorten the crypto-period. This increases the probability that encrypted messages will not be delivered with the correct key. When a notification Nx is issued at time $t = t_{nfy}$ by system SY_i, the probability that Nx is notified to all members and can be decrypted using k is one ($p = 1$), unless a join or re-key event occurs. These events can decrease the probability p to almost zero due to their delay.

   The probability that a device can decrypt a received notification Nx at time t during the joining/re-keying process can be represented as the number of recipients that can decrypt Nx divided by those who cannot. This probability can be divided into two parts: one related to joining, $f(t)$, and one related to re-keying, $g(t)$. In the following, these probabilities will be illustrated.

- **Joining:** There is a possibility that a message, designated as Nx, has been encrypted using a group key, referred to as k, while the node(s) that have joined the group have the updated version of the key, k'. In this case, the newly joined members will not be able to decrypt Nx using k'. This issue is caused by the delay in joining the group, denoted as T_{in}. The shorter the delay, the fewer devices will receive unencrypted notifications during the time period $0 \leq t \leq T_{in}$. The probability function $f(t)$ varies in proportion to the number of recipients that have received the unencrypted notifications. The rate of change of this function is given by $df(t)/dt = \alpha(t)(N - f(t))$, where $\alpha(t) = sf(t)/N$, and s is the rate at which devices join the group. Solving this ordinary differential equation with the initial condition $f(0) = 1$ yields the following result:

$$f(t) = \frac{1}{1 + (N-1) \times e^{-st}} \quad t \in [0, T_{in}], \tag{4.1}$$

where N is the total number of devices in the group. It is assumed that the larger the network, the higher the joining rate. Therefore, the total number of devices, N, affects the delay function for joining the group. When the number of devices is large enough, the probability of decrypting a notification becomes zero. If a device joins the group after or at the same time as a notification is issued ($t_{in} \geq t_{nfy}$), there is a chance that it will receive the notification due to network delays and the broadcast nature of the notification. However, the device's probability of decrypting Nx is zero ($f(0) = 0$) because it does not have the old key k. On the other hand, the probability of decrypting any notifications issued after T_{in} is one ($f(T_{in}) = 1$) because the sender and receiver both use the updated key k'. It is clear that $f(t)$ is an increasing function because $f(0) = 0$ and $f(T_{in}) = 1$.

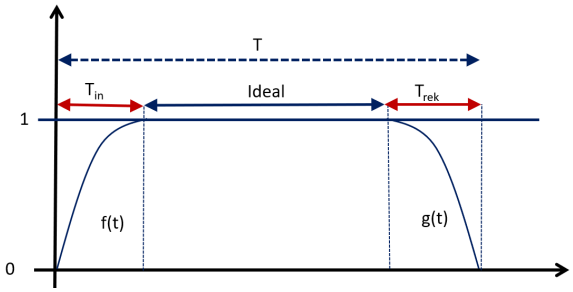


Figure 4.2: The probabilistic model of the frequent re-keying problem.

- **re-keying:** Using a similar process as the one used for frequent joining events, we can find $g(t)$ as follows:

$$g(t) = 1 - \frac{1}{1 + (N-1) \times e^{-rt}} \quad t \in [0, T_{rek}], \tag{4.2}$$

where r is the rate of re-keying. Notifications that are likely to be un-decrypted are those that are encrypted using the old key k and issued at a time $0 < t < T_{rek}$. Before the re-keying process starts, the probability of receiving a sent notification with the current key (k) is one ($g(0) = 1$). On the other hand, notifications that are issued at $t \geq T_{rek}$ will not be able to be decrypted using the old key k, meaning the probability is zero ($g(T_{rek}) = 0$) because the sender and receiver both have the updated key k'. As a result, it is clear that $g(t)$ is a decreasing function.

If there is no delay, i.e. $T_{rek} = T_{in} = 0$, each notification will be sent with the right key. Therefore, the decryption probability is one, $d_p = 1$. By applying the total probability theorem to find the area under the curve of Figure.(4.2), d_p, could be calculated in Eq.(4.4).

$$d_p = \frac{1}{T}\left(\int_0^{T_{in}} f(t)d_p t + \int_{T_{in}}^{T-T_{rek}} 1 d_p t + \int_0^{T_{rek}} g(t)d_p t\right) \tag{4.3}$$

$$d_p = \frac{T - T_{in} - T_{rek}}{T} + \frac{1}{T}\left(\int_0^{T_{in}} f(t)d_p t + \int_0^{T_{rek}} g(t)d_p t\right) \tag{4.4}$$

7. **Computation and communication efficiency:** When evaluating the feasibility and scalability of protocols, it is important to consider their costs in terms of computation and communication. The most computationally expensive processes in assessing the computational cost of protocols are modular exponentiations and other cryptographic operations such as signature creation and verification, public-key encryption, etc. Two key elements are considered when it comes to communication complexity: the number of messages transmitted and the number of rounds during the protocol execution. Messages sent by multiple participants at the same time are considered to be transmitted in the same round. "Simple rounds" are defined as rounds in which each member can only send and receive one message per round, while "synchronous rounds" are defined as required rounds in which each member may send multiple messages at the start of the round.

   In a study by Becker and Willie [17], lower limits on the communication complexity of contributory key agreement protocols were examined in two models: one with and one without a broadcast channel. In the model without broadcasting, each message is counted separately. The following results were reported:

   - Secure lock-based protocols:
     - Total number of messages $\geq n$
     - Total number of simple rounds $\geq \lceil log_2 n \rceil$
     - Total number of synchronous rounds 1
   - Other protocols:
     - Total number of messages $\geq 2n^2$
     - Total number of simple rounds $\geq \lceil log_2 n \rceil$

   where n is the number of group members. Secure lock-based protocols have a lower message complexity compared to other protocols, as they need to send fewer broadcast messages. Secure lock-based protocols require at least n messages, at least

$log2(n)$ simple rounds, and at least 1 synchronous round, while other protocols require at least $2n^2$ messages and at least $log2(n)$ simple rounds.

4.2 Research methodology

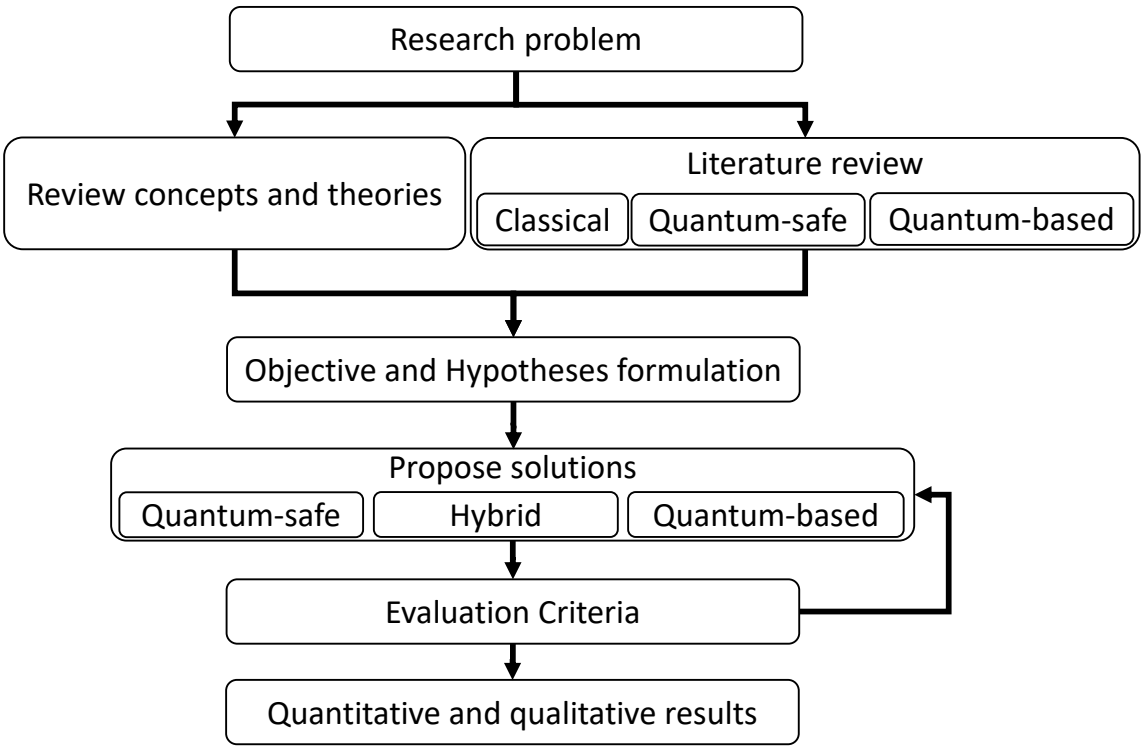


Figure 4.3: Research methodology.

In this research, I aim to address the challenges and limitations of current key establishment protocols. To do this, I have adopted a research methodology that includes the following, as shown in Figure (4.3):

1. Define the research problem: This involves clearly stating the challenges and limitations, as well as the neglected features.

2. Review the concepts and theories: Understanding the fundamental principles of quantum-safe cryptography and quantum-based cryptography is important for designing and evaluating key establishment protocols. This involves reviewing the

concepts and elements of quantum information theory, such as quantum states, quantum operations, and quantum entanglement.

3. Review the literature: This will involve conducting a thorough review of the existing research on QKD and group-based QKD protocols, including both quantum-safe and quantum-based approaches. This could help to identify any gaps in the literature and inform the direction of the research.

4. Formulate hypotheses and design objectives: Based on the identified challenges and neglected features, I formulate hypotheses about potential solutions and design objectives for addressing these challenges and filling these gaps.

5. Proposed schemes: In this step, I propose and design new protocols to address the identified challenges and gaps. This includes providing security proofs and demonstrating the performance of the proposed schemes through simulation or experimental results.

6. Evaluation criteria: In order to evaluate the proposed schemes, it is important to establish clear evaluation criteria, such as security, efficiency, scalability, and practicality.

4.2.1 Objective

The main objectives of this research are to 1) propose an effective and scalable quantum-safe key distribution protocol for two-member or group communication, 2) devise a practical and efficient quantum key distribution scheme, and 3) propose a hybrid quantum-based/quantum-safe group key management system that combines the benefits of both approaches. The effectiveness and significance of the proposed systems will be assessed through various criteria.

4.2.2 Research Hypotheses

Based on a review of the existing literature and the identified gaps in group-based key distribution research related to quantum-safe and quantum-based approaches, I formulate the research hypotheses (H1- H3) for the proposed work.

Hypothesis 1 (H1)**.** *QKD may offer unconditional security but it is hypothesized to not support group key distribution*

Hypothesis 2 (H2)**.** *For a sufficiently large number of members n, where $n > \Theta$, it is hypothesized that implementing multiple instances of two-member QKD, rather than a single group-based QKD protocol, will result in a higher level of security and efficiency.*

Hypothesis 3 (H3)**.** *It is hypothesized that group-QKD and group-quantum-safe alone are not suitable migration solutions for future group key establishment and that a hybrid solution is to be implemented as the initial step for migration.*

I investigate the dimension of the group-QKD issue and demonstrated the validity of hypotheses H1 and H2 in Section 3.2.3.1. Specifically, I provide proof for H2, which pertains the practicality of group-based QKD compared to n-QKD protocol.

In Section 3.2.2, I examine quantum-safe solutions for group key distribution and identify their limitations, security issues, and potential attacks.

To address the claim of hypothesis H3, I propose a hybrid solution that combines QKD and quantum-safe key distribution to facilitate a smooth migration. Specifically, QKD is used on a small scale to generate a random but secure secret key for each member. The quantum-safe protocol is then utilized to broadcast a secure lock that can only be unlocked by legitimate members with the correct secret keys. This process allows the group to discover the TEK, which serves as the group's secret key.

The key server shares the TEK with all valid recipients to ensure confidentiality and integrity during a session. Once an encrypted message is received, it can only be decrypted

by a valid recipient who possesses the TEK. The number of recipients is not fixed, as new members can join and others may leave. When the number of members is two, the process is referred to as a two-member key establishment. When there are more than two members, it is referred to as group key establishment. Both types of key establishments should consider certain security requirements or properties.

4.2.3 Quantum-safe evaluation

The proposed work will be evaluated using the following security properties and performance criteria.

4.2.3.1 Security properties

The key server shares TEK, with all valid recipients to guarantee confidentiality and integrity during a session. Once an encrypted message is received, which is encrypted by TEK, only the valid recipient with the TEK can decrypt the received message. The number of recipients is not fixed as new members can join and leave. To ensure the confidentiality and integrity of a message, several security properties should be considered.

Property 1. Forward-Secrecy: Forward-secrecy ensures that if an eavesdropper gains access to a consecutive set of previous group keys, they will not be able to uncover the following group keys [83]. When a member leaves the group, the key server generates a new secret key, $TEK^{'}$, and sends it only to the remaining members. This prevents a leaving member from being able to decrypt messages encrypted with the new key, $TEK^{'}$, as they do not have access to it. This helps to protect the confidentiality of future communications.

Property 2. Backward-Secrecy: Backward-secrecy ensures that if an eavesdropper gains access to a consecutive set of group keys, they will not be able to uncover the preceding group keys [83]. When a new member joins the group, the key server generates a new secret key, $TEK^{'}$, and sends it to all current members, including the new member. The new

member will not be able to decrypt previously exchanged messages, because it only knows the new key, $TEK^{'}$. This helps to protect the confidentiality of past communication.

Property 3. Collusion freedom: Collision freedom is a property that ensures that a group of members who have left cannot deduce the current secret key, TEK, through collusion. This helps to protect the confidentiality of communication by preventing individuals who are no longer part of the group from working together to deduce the key and decrypt messages.

Property 4. Key metadata security: Key metadata refers to information about a cryptographic key, including its characteristics, constraints, parameters, and intended use. Key metadata may also include the identity of a member associated with the key. While key metadata is essential for key management, it is not used in cryptographic algorithms. Key metadata is private and its disclosure can compromise the security of the secret key, TEK. Therefore, it is important to protect key metadata to ensure the confidentiality of communication.

Property 5. Key independence: The secret key, TEK, is replaced by a new key, $TEK^{'}$, whenever there is a change in membership. The key management scheme should ensure that adversaries cannot use the old key, TEK, to discover the new key, $TEK^{'}$. Key independence means that the disclosure of one key, TEK, does not compromise the security of another key, $TEK^{'}$. It is important for the keys (TEK and $TEK^{'}$) to be independent of each other in order to protect the confidentiality of communication.

Property 6. Guessing/Brute-force resistance: It is theoretically possible to guess or brute-force a cryptographic key by trying all possible combinations until the correct one is found. A brute-force resistant key is one that has two main properties: first, the probability of guessing a valid secret key, TEK, is very small; second, carrying out a brute-force attack would take a very long time, potentially years even with the most powerful computers. These properties help to ensure the security of communication by making it difficult

for an attacker to guess or brute-force the key.

Property 7. Quantum-resistant: Cryptographic systems that are quantum-resistant are designed to be resistant to attacks by quantum computers. This can be achieved through the use of cryptographic algorithms and protocols that are specifically designed to be resistant to quantum attacks. The use of quantum-resistant cryptographic systems is important for ensuring the security of communication in the face of advances in quantum computing. As quantum computers become more powerful and widespread, it is likely that they will be able to break many of the cryptographic systems that are currently in use, making it important to transition to quantum-resistant systems in order to ensure the continued security of communication.

Property 8. Membership inference: In the context of group key establishment, a membership inference attack could be used to determine whether an individual was a member of a group that used a certain key to encrypt its communication. Membership inference attacks can be particularly problematic in situations where the membership of a group is intended to be private or confidential. For example, if a group of individuals is using a key to communicate about sensitive or controversial topics, an attacker who is able to determine the membership of the group may be able to use that information to target or expose the group's members.

4.2.3.2 Performance Criteria

The proposed schemes will be evaluated with respect to the following performance criteria:

- **Computation complexity:** The computation cost of the proposed scheme is quantified in terms of the number of Arithmetic Operations Over A Finite Field ($\mathbb{F}_q OP$) [6]. Each $\mathbb{F}_q OP$ corresponds to a mathematical operation performed by the central processing unit. The execution time of $\mathbb{F}_q OPS$ depends on the capabilities of the

machine on which it runs, but it is typically measured in microseconds. It is important to evaluate the computation cost of both the key server and the members in order to assess the overall performance of the proposed scheme.

- **Storage cost:** Memory usage is an important performance metric in cryptography, as it can impact the efficiency and scalability of a protocol. This metric represents the amount of memory required by the server and each member to execute the proposed protocol. Evaluating the memory usage of the server and members can provide insights into the resource demands of the protocol and help to identify any potential bottlenecks or inefficiencies. It is important to ensure that the memory usage of the protocol is reasonable and does not excessively burden the server or members in order to maintain good performance.

- **Message overhead:** Efficient communication is an important consideration in group key establishment protocols, as it can impact the performance and scalability of the protocol. One aspect of communication efficiency is the number of messages that are required to complete the re-keying process. Reducing the number of messages required in the rekeying process can help to improve the performance and scalability of the protocol, and make it more practical for use in real-world applications.

- **Latency:** Latency, also known as delay, refers to the time it takes to complete a particular operation or protocol. Group key establishment protocols that have high latency may be perceived as slow or cumbersome by users, which can negatively impact their adoption and use. There are several factors that can contribute to the latency of a group key establishment protocol. These can include the complexity of the protocol itself, the computational resources required to execute it, the amount of communication required between participants, and the overall performance of the system. Reducing the latency of a group key establishment protocol can help to improve the user experience and make the system more practical for use in real-

world applications.

- **Scalability:** Scalability is an important consideration in group key establishment protocols, as it determines the ability of the system to handle an increasing number of members. A group key establishment protocol is considered scalable if it can support a large number of members without a significant increase in the time or resources required to establish keys.

4.2.4 Quantum-based evaluation

The proposed QKD protocols will be evaluated using the following security properties and performance criteria.

4.2.4.1 Security definitions

QKD's security is assessed in comparison to a flawless key distribution method in which Alice and Bob share a real random secret key. QKD would consider a key (K) to be a perfect key if it is a random bit string whose value is fully independent of Eve's knowledge. The deviation ε from a perfect key can be used to determine the security of K which is formulated in terms of security definitions that were proposed in [122]. Let S_A and S_B represent Alice's and Bob's bit strings, respectively.

Definition 4.2.1 (Correctness)**.** The protocol is said to be $\varepsilon_{correct}$ if

$$Pr[S_A \neq S_B] \leq \varepsilon_{correct}, \tag{4.5}$$

that is, the probability (Pr) that Alice's and Bob's keys are not identical is not greater than $\varepsilon_{correct}$.

Definition 4.2.2 (Secrecy)**.** In a cryptographic protocol, secrecy refers to the level of protection against an adversary holding a quantum system E. A protocol is considered ε_{secret}

secure if the joint state of the system and the adversary satisfies:

$$\frac{1}{2}\left\|\rho_{AB} - \tau_K \otimes \rho_E\right\|_1 \leq \varepsilon_{secret}, \tag{4.6}$$

where $\|.\|_1$ is the trace norm, τ_K is the mixed state of the key and ρ_{AB} represents the joint state of the protocol, and ε_{secret} is the maximum tolerated failure probability, in which failure means that the adversary might have gained some knowledge.

Definition 4.2.3 (Security)**.** A protocol is considered to be ε_{secure} if

$$\varepsilon_{correct} + \varepsilon_{secret} \leq \varepsilon_{secure}. \tag{4.7}$$

The protocol is considered secure if the combined value of the failure probabilities for correctness and secrecy is less than or equal to ε_{secure}, otherwise, it aborts.
To ensure the security of the protocol, it is crucial to provide security proofs that adhere to the established security definitions.

4.2.4.2 Eavesdropping strategies evaluation

To ensure the robustness of a QKD protocol, it is essential to carefully evaluate a range of potential attack strategies. This may include Intercept–resend attack, a measure-resend Attack, an entangle-measure attack, a collective attack, etc.

- **Intercept–Resend Attack:** In this attack, an eavesdropper intercepts the transmitted qubits, measures them on a computational basis to gain knowledge, and sends newly cloned qubits to participants. According to the no-cloning theorem, Eve cannot clone the sent qubit without destroying it. On the other hand, Eve cannot get any useful information about the shared key, because the intercepted qubit does not contain any useful data.

- **Measure-Resend Attack** In this attack, Eve intercepts the transmitted qubits to Bob

and measures them, and resends the measured qubits back to Bob. This attack is not effective as the transmitted qubits are not information carriers.

- **Entangle-Measure Attack:** Eve intercepts the transmitted qubit to Bob and entangles it with her ancilla state (prepared, say, in the state $|E\rangle$) by performing unitary operators. Eve then transmits the traveling qubit to Bob. Finally, Eve measures the auxiliary qubit in her hands to learn more about the shared key.

- **Collective Attack:** The essential concept behind a collective attack is as follows: the adversary Eve attempts to find a multi-qubit state that retains the correlation between the two legitimate parties (Alice and Bob). She also provides new qubits to distinguish between Alice's and Bob's measurement findings. If Eve manages to find such a state, she can remain unnoticed throughout her intervention and obtain the shared key.

- **Denial-of-Service (DOS) attack:** Eve may execute a unitary operation on each qubit traveling from Alice to Bob before allowing them to continue their journey. Although Eve is unable to obtain any useful information, she causes the protocol to be denial-of-service (DOS). Eve's attempt, on the other hand, is going to be discovered once Alice and the channel security is checked by comparing their keys' similarity percentage.

- **Double-blinding attack:** A double-blinding attack is an attack that is intended for entanglement-based QKD protocols. The idea of this attack is that Eve launches a blinding attack on both sides (Alice and Bob) [5]. This attack is for those protocols in which a central node, let us say, Charlie, sends the qubits of the entangled state to the legitimate parties (Alice and Bob). Eve does not measure the intercepted qubits; she blocks them instead and replaces them with another properly prepared entangled ancilla state. This attack won't affect the protocol as it has minimized the attack surface to the minimum by sending a single qubit that is not enough for Eve

to execute this attack.

4.2.4.3 Performance Criteria

Several performance metrics are commonly used to evaluate the efficiency of QKD protocols. These include:

- **Key generation rate:** This metric measures the speed at which a secure key is generated by the QKD protocol. It is usually expressed in bits per second or key per second. A higher key generation rate means that the protocol is able to generate a secure key more quickly.
- **Key generation distance:** This metric measures the maximum distance over which the QKD protocol can generate a secure key. It is usually expressed in kilometers. A higher key generation distance means that the protocol can be used to generate a secure key over a longer distance. This is important for applications such as secure communication over long distances.
- **Key generation efficiency:** This metric measures the efficiency of key generation, taking into account factors such as the number of photons transmitted and received and the level of noise in the system. It is usually expressed as a percentage. A higher key generation efficiency means that the protocol is able to generate a secure key with fewer resources, such as fewer photons.
- **Key generation secrecy:** This metric measures the level of secrecy achieved by the QKD protocol. It is usually expressed as the amount of information an attacker could learn about the generated key. A higher key generation secrecy means that the protocol is able to generate a key that is more difficult for an attacker to learn about.
- **Key generation reliability:** This metric measures the reliability of the QKD protocol. It is usually expressed as a percentage of successful key generation attempts. A

higher key generation reliability means that the protocol is able to generate a secure key more consistently, resulting in fewer failed attempts.

- **Key generation scalability:** The scalability of a QKD protocol refers to its ability to handle an increasing volume of key generation requests, as well as an expanding network of participants. In order to maintain efficient performance as the demand for secure keys grows, a QKD protocol should be able to handle larger key sizes, higher key generation rates, and a greater number of participants without compromising the security or reliability of the generated keys.

- **Key generation overhead:** The additional resources (e.g., time, energy, hardware) required to implement the QKD protocol, beyond those required for conventional key distribution methods.

4.3 Concluding Remarks

In conclusion, this chapter presented an overview of the research gaps and methodology for this thesis. I first identified and classified the research gaps in the field of key establishment, particularly focusing on classical, quantum-safe, and quantum-based solutions for two-member and group scenarios. I also highlighted the shortcomings of existing protocols and identified areas such as key metadata security, delay management, and frequent joining/re-keying that have been neglected by previous researchers. To address these challenges, I have formulated research hypotheses and adopted a research methodology that comprises the following objectives: 1) propose an effective and scalable quantum-safe key distribution protocol for two-member or group communication, 2) devise practical and efficient QKD protocols for two-member communication, and 3) propose a hybrid quantum-based/quantum-safe group key establishment that combines the benefits of both approaches. The proposed systems will be evaluated using various criteria, including security properties and performance metrics such as key generation rate, key generation

distance, key generation efficiency, key generation secrecy, and key generation reliability. Furthermore, I have identified potential attack strategies to evaluate the robustness of the proposed schemes. The proposed work aims to address the challenges and limitations of current key establishment protocols and contribute to the advancement of this field of research.

Chapter 5

Quantum Safe Group key Establishment

In this chapter, I propose new quantum-safe protocols for secure key establishment. My first contribution is KeyShield, a scalable and quantum-safe system. I also propose an improved version of KeyShield, called KeyShield2, which addresses vulnerabilities identified in the original protocol. Both protocols are designed to facilitate group key establishment and use a single broadcast message in an open channel for re-keying.

5.1 KeyShield Protocol

Group communication is encrypted using a single key, TEK, that is known to all the group members (n). Each member should have the latest version of TEK in a secure way. I propose a new key management scheme called KeyShield to address the issue discussed in Section 3, as a solution to the identified research gaps in Section 4.1. KeyShield finds a unique integer solution to a linear system of equations with integer coefficients over a finite filed $\mathbb{F}pr$, where pr is a prime number. KeyShield's main advantages are as follows: it requires minimum message overhead, small memory, and little computation; it is quantum-safe, and it is effective against brute force attacks. In KeyShield, the

members receive a secure lock in a public channel. Each member will get the TEK by opening the secure lock using their own secret vector. Equation (5.1) shows a linear system $A \times X = B$, as a key distribution scheme. X represents the secure lock that will be shared in public, B is a vector of the shared TEK, and A is the secret matrix where each row $A_i = [a_{i,1}...a_{i,n}]$ is assigned to a member. Once a member has its own secret vector A_i, the Key Center generates the TEK and solves the system. Then, the solution, X, is sent in a single message to everyone in an open channel. Each member finds the shared TEK using (5.2).

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix} \times \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} TEK \\ \vdots \\ TEK \end{bmatrix} \tag{5.1}$$

$$[a_{i,1} \cdots a_{i,n}] \times \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = TEK \tag{5.2}$$

The KeyShield system, as depicted in Fig. 5.1, comprises of the following seven steps:

1. The key server collects the public keys ($[pk_1...pk_n]$) of all members using a quantum-safe cryptographic system such as NTRU [72].

2. The key server generates the secret vectors ($[A_1...A_n]$) for each member.

3. The key server encrypts the secret vectors ($Enc_{pk_i}(A_i)$) using each member's public key and sends them to the members.

4. Each member uses its private key (sk_i) to decrypt the received message and extract their own secret vector (A_i).

5. The key server generates the TEK and solves the linear system.

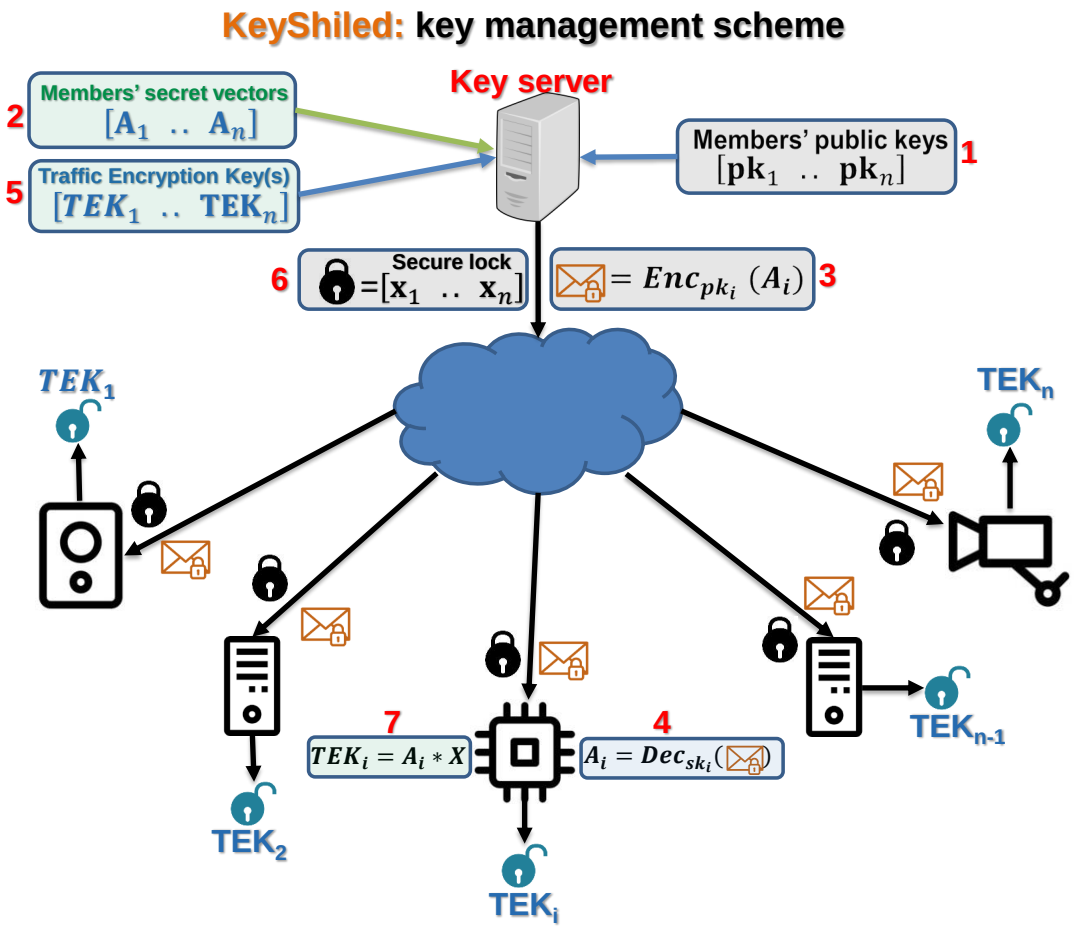


Figure 5.1: KeyShiled system model.

6. The key server sends the solution of the linear system ($[x_1...x_n]$) in a broadcast message.

7. Each member uses their own secret vector to find the TEK by computing $TEK = A_i \times X$.

5.1.1 KeyShield System conditions

A linear system of equations can have three possible outcomes: no solution, a unique solution, or infinitely many solutions. The KeyShield system requires a unique solution for its key distribution scheme. In order to ensure a unique solution, the system must meet certain conditions.

Firstly, the secret coefficients matrix A should be a square matrix, linearly independent, invertible, non-singular, and have a non-zero determinant. Additionally, the system $A \times X = B$ should be non-homogeneous (i.e. $B \neq 0$), have no equivalent solutions, be consistent, and be neither under-determined nor over-determined.

It's important to note that these conditions are based on a set of general mathematical definitions that apply to any system of equations that has a unique solution.

5.1.2 Design objectives

The primary objective of the design is to ensure confidentiality and integrity during a session by allowing the sender to share a cryptographic key, TEK, with all authorized recipients. Only authorized recipients with the TEK will be able to decrypt the received message. Additionally, the design should be flexible enough to accommodate changes in the number of recipients as new members may join or leave the group. Such key management is called group key management, which should take into account some specific security requirements or properties.

5.1.2.1 Banded linear System

An $n \times n$ matrix, A, is a band matrix if all its elements are zero outside a diagonal zone whose range is determined by the constants lower bandwidth b_L and upper bandwidth b_U.

$$a_{i,j} = 0 \text{ if } i > j + b_L \text{ or } i < j - b_U; \quad b_L, b_U > 1 \tag{5.3}$$

A matrix is called a banded matrix if its bandwidth is reasonably small. A tridiagonal matrix is a banded matrix with $b_U = b_L = 1$ as in Fig. 5.2. To make KeyShield secure against a brute-force attack, we need to make sure the matrix bandwidth is greater than one, b_L & $b_U > 1$, and the secret vector, A_i, bit-length is greater than the bit-length of the shared key TEK. By doing so, we make sure that launching a brute force attack on the TEK is easier than doing it on A_i. A security analysis regarding the effects of the bit length is presented in Section 5.1.2.4.

In a banded matrix, not all the rows (vectors) have the same number of non-zero entries. It means that the maximum number of non-zero entries is q, where $q = b_L + b_U + 1$, and the minimum number of non-zero entries is $b_L + 1$ or $b_U + 1$. This would make the attacker try to guess a secret vector, A_i, which has fewer non-zero entries, $A_i \times X = TEK$. As KeyShiled is a linear system, $A \times X = B$, the content of the B vector is used to address this possible attack. Instead of having $B = [TEK, TEK, ..., TEK]$, KeyShiled B content becomes

$$B = [[r_1, ..., r_{b_L}], [TEK, .., TEK], [r_{n-b_U}, ..., r_n]] \tag{5.4}$$

where r is a random number. B's first (b_L) and the last (b_U) entries are filled with useless random numbers, and the entries in the middle are filled with the actual traffic key TEK. The secret vector that has less than q non-zero entities becomes a useless random number. The concept of secret vector bit-length, A_L, is discussed in detail in the following. A_L defines the total bit-length of the q non-zero entries of the secret vector. For example,

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \ddots & & & & \vdots \\ 0 & a_{3,2} & a_{3,3} & a_{3,4} & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & a_{n-2,n-3} & a_{n-2,n-2} & a_{n-2,n-1} & 0 \\ \vdots & & & & \ddots & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

Figure 5.2: Banded matrix of bandwidth=1 ($b_L = b_U = 1$).

if we consider each entry, $a_{i,j}$, to have an eight-bit length ($a_L = 8$), then the number of non-zero entries can be calculated using the following equation:

$$q = \lceil \frac{A_L}{a_L} \rceil. \tag{5.5}$$

5.1.2.2 Members Grouping

Let us assume that the ground set, $\mathcal{V}$, contains all the members.

$$\mathcal{V} = \{m_1, ..., m_n\}. \tag{5.6}$$

Giving each member a secret vector would not increase the security level; however, it will (1) maximize the computation cost in solving the linear system; (2) minimizes the communication overhead for key-updating; and (3) maximizes the number of stored keys. In Section 5.1.4.4 a security analysis has been provided to show the effects of grouping. To balance the system, the ground set, $\mathcal{V}$, is partitioned into s logical (disjoint) subsets , $\{V_1 \cup V_2 ... \cup V_s\}$,

$$V_i, V_j \subset \mathcal{V} : V_i \cap V_j = \{\emptyset\}; \forall\, i \neq j;\ 1 \leq \ \ i, j \leq s. \tag{5.7}$$

Each su member forms a subset called V_i,

$$V_i = \{m_1, ..., m_{su}\}. \tag{5.8}$$

Instead of assigning a secret vector, A_i, to a single member,m_i, it would be assigned to a subset of members,V_i. Each member of the subset V_i will receive the same secret vector, A_i. The members of each subset are chosen randomly, and members do not have access to the list of members. The linear system size becomes $s \times s$ instead of $n \times n$. The key center solves the linear system and sends the solution, X, to everyone. Each member will use its secret vector, A_i, to obtain the shared key, $A_i \times X = TEK$.

The number of subsets, s, is an important factor as it balances the time required to solve the system and the number of re-keying messages. Having a large number of subsets increases the size of the linear system, which leads to a higher number of $\mathbb{F}_q OPS$. On the other hand, having a small number of subsets increases the messages overhead as the number of each subset increases. The size of the subset, su, and the number of subsets, s, are inversely proportional to each other with the proportionality constant n.

$$s = *\frac{n}{su} \tag{5.9}$$

This equation represents the number of subsets, by rounding up the division of the total number of members by the number of members in each subset. We can determine the optimal number of subsets, s, by considering two main factors: the total number of $\mathbb{F}_q OP$ operations and the total number of messages.

- According to [49], the total number of $\mathbb{F}_q OP$ operations required to solve a banded linear system with size s is given by:

$$\mathbb{F}_q OPS = 2s(b_L \times b_U + b_L + b_U) \tag{5.10}$$

where b_L and b_U are the lower and upper bandwidths of the system, respectively. If $b_L = b_U$, we can simplify the formula to:

$$\mathbb{F}qOPS = 2s(bLU^2 + 2b_{LU}) \tag{5.11}$$

Here, $\mathbb{F}qOPS$ represents the computation cost, measured in terms of the number of $\mathbb{F}qOP$ operations, each of which corresponds to a single mathematical operation executed on the CPU. Note that the actual execution time of $\mathbb{F}_qOPS$ depends on the capabilities of the machine and is typically measured in microseconds.

- The number of messages required in the re-keying process is presented in (5.12),

$$Messages = 2su + 1 \tag{5.12}$$

where $2su$ is the number of messages required to send the secret vector A_i to the subset members and one broadcast message to inform all the system entities about obtaining the new TEK. When a decision is made to replace the old secret vector A_i with its new version $A_i^{'}$, each subset member should receive a notification. The new secret vector message takes time to be delivered to all the subset members, let us call the required time the re-keying delay. Let m_i denote the i^{th} subset member; su denotes the number of group members; and Δ_i represents the time elapsed to complete the delivery of $A_i^{'}$ to m_i. In this case, the delay, which we can call T_{rek}, take the maximum delay, such that T_{rek} = max$\{\Delta_1, \Delta 2, ..., \Delta_n\}$. Increasing the number of subset members, su, increases the delay. The message delay is the summation of processing time on the sender side, propagation time, storage delays caused by intermediate devices like switches and bridges, and receiving time. If an acknowledgment is required, the message cost is doubled.

The main unit that unifies the two factors, computation cost and messages overhead, is

time. As discussed earlier, the lower the number of subsets, the faster the system is solved; however, a lower number of subsets increases the messages overhead, which increases the delay. Thus, the objective is to balance these two factors to find the minimum time required to solve the system and to complete the secret vector distribution. A weight, w, should be multiplied by the number of required messages to show their actual effect in the re-keying process.

To optimize the number of subsets, I develop a cost function considering both computation cost and message overhead:

$$Cost = (2su + 1) \times w + 2s({b_{LU}}^2 + 2b_{LU}), \tag{5.13}$$

where $(2su + 1)$ represents message overhead, $2s({b_{LU}}^2 + 2b_{LU})$ is the computation cost, and w is the weight of the messages. The cost function can be rewritten as follows:

$$\begin{aligned} Cost &= (2su + 1) \times w + 2s({b_{LU}}^2 + 2b_{LU}) \\ &= (2\frac{n}{s} + 1) \times w + 2s({b_{LU}}^2 + 2b_{LU}) \\ &= 2nw\frac{1}{s} + w + 2s({b_{LU}}^2 + 2b_{LU}). \end{aligned} \tag{5.14}$$

To determine the optimal cost, it is necessary to differentiate the cost function with respect to s and then set the resulting derivative to zero:

$$\begin{aligned} \frac{\partial Cost}{\partial s} &= -\frac{2nw}{s^2} + 2({b_{LU}}^2 + 2b_{LU}) \\ 0 &= \frac{-2nw}{s^2} + 2({b_{LU}}^2 + 2b_{LU}) \\ s^2 &= \frac{2nw}{2({b_{LU}}^2 + 2b_{LU})}. \end{aligned} \tag{5.15}$$

The following formula can be derived to obtain the optimal number of subsets:

$$s = \sqrt{\frac{nw}{({b_{LU}}^2 + 2b_{LU})}} \quad n > \frac{w}{({b_{LU}}^2 + 2b_{LU})} \tag{5.16}$$

Equation 5.16, where only the positive root is considered, is a general solution for determining the optimal number of subsets.

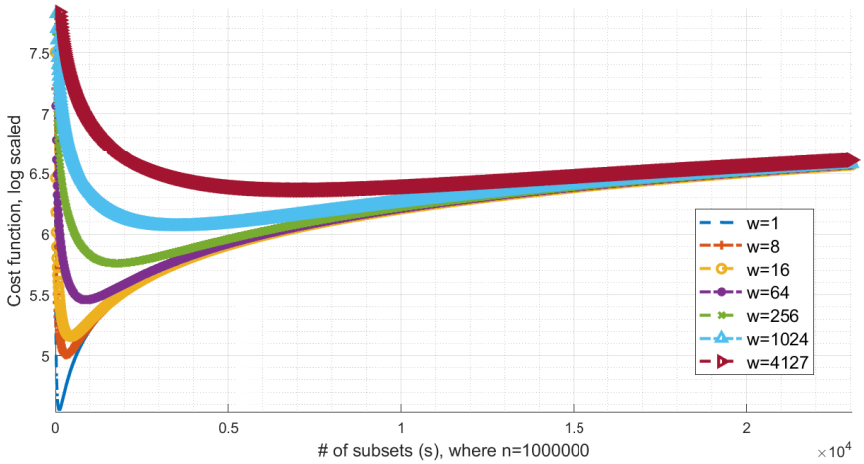


Figure 5.3: Finding the optimal number of subsets, s, theoretically where $n = 1000000$,$b_{LU} = 8$, w varies

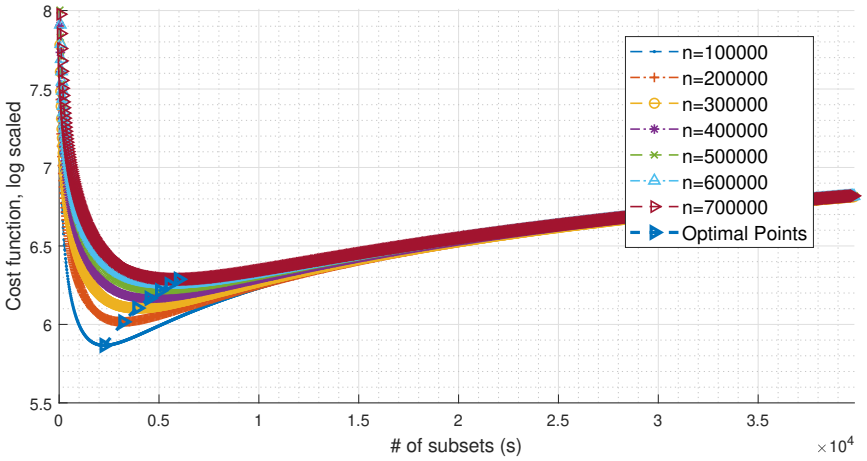


Figure 5.4: Finding the optimal number of subsets, s, theoretically where $w = 1$, $b_{LU} = 8$, and n varies

The optimal number of subsets can be determined by considering the total number of $\mathbb{F}_qOP$ operations and the total number of messages required. The total number of $\mathbb{F}_qOP$ operations required to solve a banded linear system with size n is $\mathbb{F}_qOPS = 2s(b_L \times b_U + b_L + b_U)$, where b_L and b_U are the lower and upper bandwidths, respectively. When $b_L = b_U$, the total number of $\mathbb{F}_qOP$ operations is $\mathbb{F}_qOPS = 2s(b_{LU}^2 + 2b_{LU})$, where s

represents the system size and b_{LU} is the common lower and upper bandwidth.

The number of messages required in the re-keying process is presented in equation (5.12), $Messages = 2su + 1$, where $2su$ is the number of messages required to send the secret vector A_i to the subset members and one broadcast message to inform all the system entities about obtaining the new TEK.

To balance these two factors and find the minimum time required to solve the system and complete the secret vector distribution, a cost function is developed. The cost function is given by equation (5.13), $Cost = (2su + 1) \times w + 2s(b_{LU}^2 + 2b_{LU})$, where $(2su + 1)$ is messages overhead, $(2s(b_{LU}^2 + 2b_{LU}))$ is the computation cost, and w is the weight of the messages.

The value of w is determined by the key server after examining the average time required to send a message (Δ) and the time required to execute a mathematical operation ($T_{\mathbb{F}_qOPS}$), $w = \frac{\Delta}{T_{\mathbb{F}_qOPS}}$. As an example, if the $T_{\mathbb{F}_qOPS}$ of executing a mathematical operation on MATLAB and Core i3 machine is equal to $(0.011ms)$ and according to (***testmy.net***) the average network latency between my location (Fredericton, NB, Canada) and (Toronto, Canada) is $(45.4ms)$, then in terms of milliseconds, sending one message weight is $w = 4,127.27$.

The cost function is visualized in Fig. 5.3 and 5.4, where it can be seen that increasing the messages overhead weight, w, increases the number of subsets. The optimal number of subsets is given by equation (5.16).

5.1.2.3 Re-keying

In a group communication, a single key is used to encrypt the communication among group members. When a membership change occurs, such as a new member joining or a member leaving, the key should be renewed to ensure backward- and forward secrecy. This also means that any information related to the key that was known to the affected member, such as the secret vector A_i, must also be updated. Notifications should be sent

to all members to replace the old key with the new version, and only members of the affected subset V_i should be notified of the new secret vector $A_i^{'}$.

In the literature, the new vector is typically required to be linearly independent of the entire system of n vectors. However, in my approach, only a small number of vectors need to be linearly independent of the new vector due to the use of a banded system. The number of columns affected is equal to the number of non-zero elements in each row, which is $q = b_L + b_U + 1$. The number of rows affected varies depending on the index of the affected row, with the maximum number being $h = 2q - 1$. The affected matrix size is ($\mathbf{H} \times q$), which can be calculated using equation (5.19).

$$q = b_L + b_U + 1 \tag{5.17}$$

$$h = 2q - 1 \tag{5.18}$$

$$\begin{cases} i < q & \mathbf{H} = h - x \quad ; x = q - i \\ (i \geq q)\ \&\ ((n-i) \geq q) & \mathbf{H} = h \\ ((n-i) < q) & \mathbf{H} = h - x \quad ; x = q - (n-i) \end{cases} \tag{5.19}$$

The matrix rows affected by a change in vector A_i start at $max(1, i - 2b_{LU})$ and end at $min(n, i + 2b_{LU})$, while the matrix columns affected start at $max(1, i - b_{LU})$ and end at $min(n, i + b_{LU})$. This is clearly shown in Figure 5.5. This affected block can be referred to as $\mathcal{C}$, with a maximum number of rows h and a maximum number of columns q. Since in a banded matrix, each vector A_i has q non-zero values and $n - q$ zeros. Therefore, the vector $\mathcal{C}_j$ is simply a copy of the non-zero values of vector A_i.

A more efficient method for ensuring the linear independence of the new secret vector $A_i^{'}$ is to use the properties of linear independence. If $\mathcal{C}$ is a linearly independent matrix and $\mathcal{C}_j^{'}$ is obtained by adding a scalar value to a row or column of $\mathcal{C}$, then $\mathcal{C}$ remains linearly independent. This approach avoids the limitations of trial and error, such as limited test

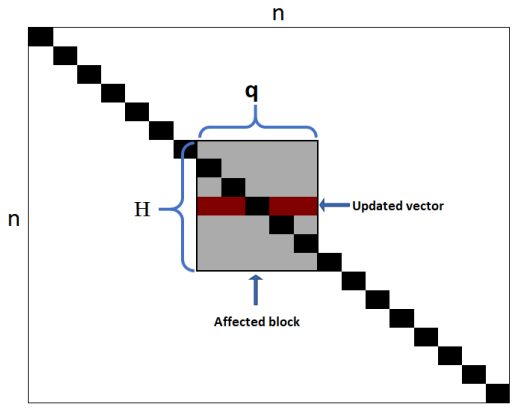


Figure 5.5: Maximum size of the affected blocks resulting from updating the secret vector A_i.

opportunities, time consumption, and wasted resources. Equation (5.20) illustrates this concept by showing that the summation of a linearly independent matrix remains linearly independent.

$$\mathcal{C}_{j,z}{}' = \sum_{L=1}^{\mathbf{H}} \mathcal{C}_{L,z} \quad ; z = [1, ..., q] \tag{5.20}$$

5.1.2.4 KeyShiled security parameter

In this study, we introduce A_L as the security parameter that represents the desired bit-strength of the KeyShield building block (A_i). If KeyShield is used for distributing keys for the AES-256 algorithm, then the resulting TEK will have a bit length of 256. The bit length of the secret vector A_i should be greater than the bit length of TEK, but not excessively so. Beyond a certain point, increasing the difficulty of A_i does not significantly improve security, but it does increase computational and storage requirements. Therefore, selecting an appropriate value of A_L that balances security and efficiency is important for the overall performance of the system. In a system with 10k members, for example, the linear system size would be ($10k \times 10k$). If a_i, j is just one bit, then A_i would be $10k$ bits in length. However, all this complexity, computation, and storage may be unnecessary if the main secret is only 128 bits. Two solutions address this issue: First, members grouping, where KeyShield assigns a secret vector (A_i) to a subset of members while maintaining

the same security level. Second, instead of using a full secret matrix A, KeyShield uses a banded matrix while keeping the difficulty of guessing a matrix row (A_i) higher than the cost of guessing the shared key (TEK).

The secret vector, A_i, in Section 5.1.2.1 comprises of q non-zero random numbers, each with a bit length of a_L. This results in a total bit length of $A_L = a_L \times q$, which is greater than the bit length of the shared key TEK_L ($A_L \geq TEK_L$). To crack KeyShield, an attacker would need to guess a single A_i vector, with a bit length of A_L, making it much more difficult than guessing the main secret (TEK). To ensure security, each entry ($a_{i,j}$) should have a minimum bit length of a_L,

$$a_L \geq * \frac{TEK_L}{q}, \tag{5.21}$$

where TEK_L is the bit length of TEK and q is the number of non-zeros values in the

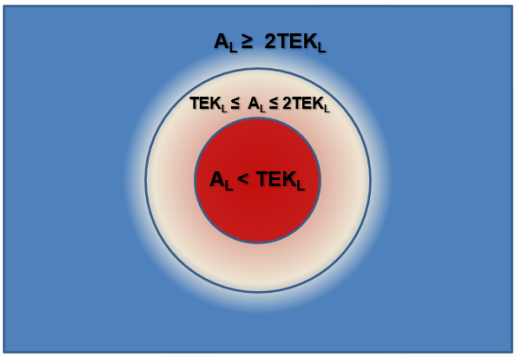


Figure 5.6: Security parameter (A_L) visualisation.

vector A_i. Equation (5.21) shows the lower bound of a_L shown in Figure 5.6.

- If a_L is in the red area, the secret vector bit length A_L is weaker than the shared TEK_L ($A_L \leq TEK_L$), making it easier for an attacker to use a brute-force attack on A_i than on TEK.
- When $A_L = TEK_L$, both A_L and TEK have the same level of difficulty. This defines the lower bound of a_L.

- If a_L is in the white area, the A_L bit length is in the safe area as $TEK_L \leq A_L \leq 2TEK_L$. $A_L = TEK_L$ defines the lower bound and $A_L = 2TEK_L$ defines the upper bound. In this case, an attacker might find it easier to use a brute-force attack on the shared TEK.

- If a_L is in the blue area, the A_L bit length is larger than twice that of the shared TEK. This area is bounded by $A_L = 2TEK_L$. In this case, resources are wasted with no additional security benefits as it is almost impossible to break them. Increasing the bit length beyond $2TEK_L$ is unnecessary.

The KeyShield scheme is designed to be strong, secure, and efficient, and not to be the weakest point in the security system.

5.1.3 KeyShield's Indistinguishability

The linearity of the system contributes to KeyShield's indistinguishability under CPA (Chosen Plaintext Attack) and CCA (Chosen Ciphertext Attack).

- **IND-CPA:** In an IND-CPA (Indistinguishability under Chosen-Plaintext Attack) scenario, let us assume that the attacker has the ability to choose plaintexts ($TEKs$) and obtain their corresponding ciphertexts (Xs). The attacker can choose two keys, say TEK_1 and TEK_2, and the key server will generate two secure locks, X_1 and X_2, by solving the linear system and sharing the resulting secure locks. Due to the randomness injected into the secret matrix A and the vector B (see Equation 5.4), the attacker should not be able to distinguish between X_1 and X_2. This property is crucial for ensuring that the KeyShield scheme is secure against chosen-plaintext attacks, where an attacker may be able to obtain some information about the secret key by observing the resulting ciphertexts.

- **IND-CCA:** In an IND-CCA (Indistinguishability under Chosen-Ciphertext Attack) scenario, let us assume that the attacker has the ability to choose ciphertexts (X)

and obtain their corresponding plaintexts (TEK). To obtain the TEK given the X, we need to multiply a secret vector, let's say A_i, with X: $A_i \times X = TEK$. This operation can be done by any member who has a secret vector. The attacker can choose two secure locks, say X_1 and X_2, and send them to a member who uses their secret vector to open the secure locks and send the resulting TEK_1 and TEK_2 to the attacker.

However, the attacker should not be able to distinguish between TEK_1 and TEK_2 without knowing the secret vector A_i. There are infinitely many vectors that satisfy the equation $A_j \times X = TEK$, making it impossible to determine the exact value of A_i using either classical or quantum efficient algorithms. To illustrate, assume $X = [2\ 5\ 3]$ and $TEK = 28$, finding A_j that satisfies the equation $A_j \times X = TEK$ would be an infinite set of vectors that would give us 28 when multiplied by the assumed X vector such as $[3\ 2\ 4]$, $[10\ 1\ 1]$, $[0\ 5\ 1]$, and so on.

5.1.4 Security analysis

In the following analysis, I consider two types of attackers:

1. An insider attacker whose goal is to disclose cryptographic keys or violate the forward/backward secrecy.

2. An outsider attacker who possesses publicly known information, such as the secure lock X, is attempting to guess the TEK or store messages for future use.

We analyze the following security properties of KeyShield for these two types of attackers.

5.1.4.1 Forward secrecy

As the number of members is not fixed, any member can join at any time. KeyShield ensures that the new member should not learn the previously used TEK keys and the secret vectors A. Once a member joins, a re-keying process starts to refresh the old TEK

and to regenerate a new secret vector for the affected set of members. Thus, the joined members do not know the old TEK and the secret vector A_i.

5.1.4.2 Backward secrecy

As the number of members is not fixed, any member can leave at any time. KeyShield assures that the left/evicted member should not learn the future used TEK keys and the secret vectors A. Once a member is left/evicted, a re-keying process starts to refresh the old TEK and regenerate a new secret vector for the affected set of members. Thus, the left members do not know the future TEK and the secret vector A_i.

5.1.4.3 Guessing/Brute-force resistance

Guessing or brute-forcing a cryptographic key is theoretically possible. However, a key with strong brute-force resistance exhibits two primary properties: first, the probability of correctly guessing a valid TEK is extremely low; second, even with the most powerful computers, a brute-force attack should require a significant amount of time to succeed, potentially taking years. KeyShield employs A_L as the security parameter to counter brute-force attacks effectively. Although guessing the secret vector could break the system, the secret vector's bit length, A_L, is always greater than the traffic key's bit length, TEK_L. Ensuring $A_L \geq TEK_L$ increases the difficulty of guessing the secret vector compared to guessing the TEK itself.

To further illustrate this point, the attacker's problem can be formulated as finding A_i such that:

$$A_i \times X = TEK$$

where $A_i \in \mathbb{F}_q^{(1\times p)}$ is the unknown secret vector. Since the matrix A is composed of secret row vectors distributed securely to the members, the attacker has no information about A_i. Therefore, the attacker has to search for A_i in a space of size pr^n. If pr is large enough,

the search space becomes enormous, making the attacker's problem computationally infeasible.

To quantify the difficulty of the attacker's problem, let's consider the probability that the attacker can guess the correct A_i in a single attempt. Since there are pr^n possible values for A_i, the probability of a successful guess is:

$$\frac{1}{pr^n} \tag{5.22}$$

This probability is extremely small when pr is large, making it practically impossible for the attacker to guess the correct A_i in a single attempt. As the attacker has no other information about A_i or TEK, their chances of finding the correct values do not improve over multiple attempts.

However, in this scenario, the attacker is not restricted to finding a single, specific vector. In fact, any vector present in the middle of the matrix A will suffice. Furthermore, the complexity of the problem increases due to the potential for recombination: if the attacker can accurately predict a portion of multiple secret vectors and appropriately recombine them, they can successfully solve the equation. In this case, each correctly guessed portion of a secret vector can be considered a piece of a larger puzzle that leads to the solution.

If there are n vectors and an attacker needs to guess at least 2 pieces to solve the equation, then there are $\binom{n}{2}$ combinations of picking 2 pieces out of n. If the attacker can guess more than 2 pieces, then there are $\binom{n}{3}$ combinations of picking 3 pieces, and so on, up to $\binom{n}{n}$.

The total number of possible combinations is the sum of these, which is $2^n - n - 1$ (since we're excluding the cases where the attacker guesses only 1 piece or no pieces at all).

Therefore, the probability of an attacker randomly guessing a correct combination that solves the equation is then:

$$P = \frac{2^n - n - 1}{p^n}. \tag{5.23}$$

5.1.4.4 Grouping security analysis

In group key management, two main things are shared, namely, the traffic encryption key (TEK) and the Key Encryption Key (KEK). The KEK is used to protect the TEK. A message that contains the TEK is encrypted using the KEK key. Using KeyShield, there is no need to encrypt the TEK; a secure lock is distributed instead. Then, each member uses its secret vector A_i to extract the shared TEK, $TEK = A_i \times X$. The secret vector A_i is used to extract the shared TEK the same way the KEK is used in the literature. With this knowledge in hand, we have two secrets to share: a primary one, which is the TEK; and a second one, which is the A_i or KEK in the literature. Let us consider a situation where we want to share a secret TEK with member(s). There will be three possible scenarios:

- **(n) secret and (n) member:** Each member is responsible for its own secret and the disclosure of its secret won't affect others as each member has a different secret. In this scenario, the presence of a weak/vulnerable member won't affect others.
- **(s) secret and (s) group of members:** There will be (s) secrets for (s) group (one secret for each group). A secret TEK_i is assigned to the group members. The disclosure of a secret TEK_i affects all the members of the group. In this scenario, the presence of a weak/vulnerable member affects the group.
- **One secret and (n) members:** Every member has a copy of the same secret (TEK). In this scenario, the presence of a weak/vulnerable member affects everyone.

By looking at the group effects of holding secret(s), it can be concluded that the larger the number of members holding a secret, the higher the probability the secret will be exposed. This conclusion is always true except in KeyShield because its shared secret is not primary (TEK). This work involves the sharing of a secondary secret (A_i), which is utilized to protect and extract the primary secrets ($TEKs$). Similar to sharing the primary secret

TEK, single A_i can be shared among all the members, su different A_i shared among su group of members or n different A_i shared with n member. Let us consider the third scenario where a single is a secret (TEK) shared with (n) member. As the primary secret (TEK) is shared with (n) members, sharing the secondary secret A_i to a single member or a group of users won't make the security level any worse. To clarify this claim, let us consider the same scenarios above but for sharing the secondary secret A_i:

- **(n) secret vector and (n) member:** Each member is given a separate secret vector A_i. Assume that a member (i) is weak/vulnerable, therefore its A_i will be used to extract the primary secret TEK.

- **(s) secret vector and (s) group of members:** Assume s different secret vectors A_i are shared among s different groups of members. This means each group member receives the same A_i. Assume that a member (i) is weak/vulnerable, therefore its group A_i will be used to extract the primary secret TEK.

- **Single secret vector and (n) member:** A single secret vector A_i is shared among all the members. This means each member receives the same A_i. Assume that a member (i) is weak/vulnerable; therefore its copy of the A_i will be used to extract the primary secret TEK.

As can be clearly understood, having a single A_i or n different A_i won't make any difference in terms of security. As A_i is used to protect TEK, therefore a single KEK_i is enough to extract the primary secret TEK. On the contrary, having more than a single primary secret TEK makes a big difference in terms of security as the disclosure of TEK_i won't affect TEK_i.

5.1.4.5 Collusion freedom

KeyShield ensures that any subset of members that leave should not be able to learn the future used TEk or A_i through collusion. None of what the former members know/have

information about has been used to regenerate the new $TEK^{'}$ or A_i.

5.1.4.6 Birthday attack

KeyShield won't be affected by this type of attack as no hash function is used. Unlike LKH, it depends on the effectiveness of hash functions.

5.1.4.7 Membership inference

KeyShield is not vulnerable to this type of attack. Members have been grouped to gain some time and to ease the management. As discussed earlier regarding the security analysis of grouping, the disclosure of a secret vector compromises the TEK. A set of members has the same secret A_i; however, knowing the group or the group members has no meaning and no security issue. None of what the group members know/knew is used in generating the secret vector; therefore, knowing the group members provides no clue to compromise the system.

5.1.4.8 Quantum-safe

The issue with current popular encryption schemes is that they rely on solving one of three mathematical problems: the factorization problem, the discrete logarithm problem, or the elliptic-curve discrete logarithm problem [20]. However, these problems can be easily solved by running Shor's algorithm on a powerful quantum computer [20, 129]. KeyShield relies on a mathematical problem that has no known classical or quantum-efficient algorithms for extracting the KeyShield shared key (B).

The problem of solving a Linear System of Equations classically is well-studied in mathematics, and it is known that in most cases, there are infinitely many solutions or no solutions at all. The goal of KeyShield is to find a unique solution for X given A and B, while keeping A and B secret. The problem of solving linear systems is well-studied in mathematics, and it is known that in most cases, there are infinitely many solutions or

no solutions at all. However, KeyShield ensures having a unique solution for X when A and B are known. Solving such a linear system efficiently is a hard problem, as it is closely related to the well-known problem of Matrix Inversion, which is known to be computationally hard.

While there has been recent progress in developing quantum algorithms for solving linear systems of equations, such as the HHL algorithm proposed by Aram Harrow, Avinatan Hassidim, and Seth Lloyd in 2009 [71], these algorithms are not yet efficient enough to be considered a practical solution. The HHL algorithm was first implemented experimentally in 2013 [38], and it has received significant attention in recent years [28, 40, 44, 134, 151, 155]. However, it is important to note that these quantum algorithms, including the HHL algorithm, are not able to provide an exact solution to the linear system of equations. Instead, they provide an approximation of the solution [101]. The KeyShield system utilizes a linear system of equations that is not sparse and requires an exact solution. Therefore, the quantum linear system of equations algorithms, such as the HHL algorithm, cannot solve the KeyShield linear system, let alone crack it. KeyShield uses a mathematical problem that has no known classical or quantum-efficient algorithms for solving it and this makes it a strong candidate for use in post-quantum cryptography.

The security of KeyShield relies on the fact that there are currently no known classical or quantum-efficient algorithms that can efficiently solve this solving Linear System when A and B are known. In order to break the system, an attacker would need to either find A and B, or find a way to efficiently solve the system of equations. However, finding A and B would require either compromising the members of the system or accessing their secret information, which is infeasible in practice. There are several scenarios in which an attacker may try to break the system:

1. If a member is an attacker or is compromised, both the secret vector A_i and the TEK are disclosed. In this case, any group key management system would be broken. However, KeyShield mitigates this by changing the secret vectors and the

shared key TEK.

2. Let's assume that the attacker has access to the old TEK and secret vector A_i of a left/evicted/compromised member. In this case, the system will regenerate only the secret vector A_i and the TEK, while the other secret vectors will remain unchanged. As the attacker knows the publicly available vector X and the number of secret vectors, n, they may try to find a new secret vector A_j that could be used to extract the new key $TEK^{'}$. This would be done by finding a vector A_j that satisfies the equation $A_j \times X = TEK$, and then computing the new $TEK^{'}$ by $A_j \times X^{'} = TEK^{'}$.

   However, this problem is not as easy as it seems, as there are infinitely many solutions that satisfy the equation $A_j \times X = TEK$, thus there are no classical nor quantum efficient algorithms that can find the exact required A_j with 100% confidence. To illustrate, assume $X = [2\ ;5\ 3]$ and $TEK = 28$, finding A_j that satisfies the equation $A_j \times X = TEK$ would be an infinite set of vectors that would give us 28 when multiplied by the assumed X vector such as $[3\ 2\ 4]$, $[10\ 1\ 1]$, $[0\ 5\ 1]$, and so on.

3. Assuming an attacker knows the values of X and B, and uses the equation $A = B \times X^{-1}$ to find the $n \times n$ matrix A, it is important to note that this equation does not yield a unique solution. This is because X and B are vectors of size $1 \times n$, while A is a matrix of size $n \times n$. In other words, there are more unknowns than equations, which results in an undetermined linear system with infinitely many solutions.

   In more detail, a linear system with unknowns greater than the number of equations is referred to as an undetermined system, and it has infinitely many solutions [29]. In the case where we have n equations and $n \times n$ variables, n variables can be determined while the remaining $n^2 - n$ variables are considered free variables. These free variables can take any value, and once chosen, the n equations are used to determine the values of the non-free variables.

Therefore, the attacker will not be able to extract the unique content of A using this system, except in the case where n = 1, in which the system will have a unique solution.

4. Let us assume an attacker knows the value of X and wants to determine the secret vector A_i and the shared key TEK that satisfies the equation $A_i \times X = TEK$. In this scenario, the number of unknowns, A_i and TEK, is greater than the number of knowns, which is only X. Therefore, it is impossible for the attacker to determine the secret vector A_i or the shared key TEK with the information they have. They have no means to verify their guesses or narrow down the possibilities.

Quantum linear systems algorithms: Solving a linear system of equations is the task of returning a vector x satisfying $A \times X = B$ by taking a given matrix A and vector B. Aside from cracking a linear system of equations, there is no efficient quantum algorithm to solve a linear system of equations. A new quantum algorithm for solving linear systems equations, the HHL algorithm, has been proposed in 2009 by Aram Harrow, Avinatan Hassidim, and Seth Lloyd [71]. The implementation of this algorithm was first demonstrated in 2013 [38]. As a linear system of equations have a vast number of applications, an efficient quantum-based linear system solver has received significant attention recently [28, 40, 44, 134, 151, 155]. According to a recent study published in Nature, this algorithm does not give the exact solution; instead, it provides an estimate [101]. Several quantum linear systems of equation solvers have been proposed; however, they solve the system of equations in the same way as the HHL algorithm does. The HHL algorithm only works if the linear system is sparse and has a low condition. KeyShield system of equations is not sparse and is well conditioned and requires an exact solution. Therefore, the quantum linear system of equations algorithms, like the HHL algorithm, cannot solve the KeyShiled linear system, not to mention cracking it.

KeyShield flexibility :

AES is a flexible quantum-safe system because it can adapt to a quantum attack by increasing its key size [108]. Similarly, KeyShield can also increase its security by increasing the length of the secret vectors, the size of the system, and the bandwidth of the system. Additionally, several security layers such as hashing or encrypting the secret vectors and the shared key TEK can also be added to further increase the difficulty of a quantum computer solving the problem.

When compared to one of the proven quantum-resistant techniques called Learning With Errors (LWE) [95], it can be concluded that KeyShield is a quantum-safe key management scheme. The LWE system is based on the equation $A \times X + e = B$, where A and B are publicly known, X is the secret, and e is a noise. Without the noise e, the equation becomes $A \times X = B$, which is a simple linear system with only one unknown, X, which can be easily solved. However, by adding noise, it becomes more difficult to extract the exact values of X. On the other hand, KeyShield keeps A and B secret and publishes X over an open channel, making it extremely difficult for an attacker to break, as most of the variables are unknown. When there are more unknowns than knowns, there will be infinitely many solutions.

5.1.5 Comparison with Related Works

In this section, I compare the proposed solution with existing ones in the literature. I evaluate various factors, such as the main principle of the scheme, its ability to provide forward and backward secrecy, quantum-safe, scalability, message overhead, computation complexity, and storage. These comparisons are presented in Table (5.1). Additionally, Table 5.2 shows the various key establishment attacks that the proposed solution is able to defend against. KeyShield and Tang share similar concepts and many of the features listed in Table 5.1, with the exception of key server computation cost and storage. In terms of attacks, the main difference is that Tang is vulnerable to collision attacks between member secret vectors. Both KeyShield and Tang are based on the difficulty of breaking a

Table 5.1: A comparison between KeyShiled and literature works

	KeyShield	GKMP	LKH	**Secure lock**	Piao	Tang
Principle adopted	**Linear system**	Encryption	Tree	CRT	Polynomial	**Linear system**
Forward/Backward Secrecy	**yes**	No	yes	yes	Yes	yes
Strong encryption needed	**no**	yes	Yes	Yes	No	No
Efficient for very large group	**yes**	No	Yes	No	Yes	No
Scalable	**Yes**	No	No	Yes	Yes	Yes
Number of re-keying messages	$O(1)$	No	$O(log2(n))$	$O(1)$	$O(1)$	$O(1)$
Member computation complexity	$O(1)$	$O(1)$	$O(log2(n))$	$O(1)$	$O(1)$	$O(1)$
KS computation cost	$O(s)$	$O(n)$	$O(2log2(n))$	$O(n)$	$O(n)$	$O(n)$
KS Key storage	s	n	n	n	$n+1$	n
Member key storage	2	2	$log2(n)$	2	2	3

Table 5.2: Summary of the main key management attacks

	KeyShield	GKMP	LKH	Secure Lock	Piao	Tang
Brute force attack	No	Yes	Yes	Yes	Yes	No
Quantum safe	**Yes**	No	No	No	No	Yes
Collision attack	No	No	Yes	No	No	Yes
Man-in-the middle	No	Yes	Yes	No	No	No
Birthday attacks	No	No	Yes	No	No	No

linear system, but KeyShield outperforms Tang for several reasons, including the use of a banded linear system, smaller block of vectors affected by the re-keying process $(su : q)$ rather than $(n : n)$, considering a set of members as one, and not evaluating the number of bits of each secret vector A_i in Tang, which results in fewer data storage.

KeyShield has several advantages over other proposed schemes in the literature. It assures forward and backward secrecy requires no encryption functions, is scalable for large membership changes, has $O(1)$ re-keying messages, $O(n)$ computation cost for the key server, and $O(1)$ computation cost for each member, and requires only two keys to be

stored on the member device and n keys stored in the key server. Additionally, KeyShield can be solved in parallel using the SPIKE algorithm [114], unlike Tang's linear system-based scheme. My implementation is in C++ and utilizes the GMP library [65] for exact computation. I have conducted experiments to verify the correct distribution of the TEK for various scenarios using the GMP library and OMNET++ network simulator.

Table 5.3: The main differences between KeyShield and Tang

	KeyShield	Tang
Type of LS	Banded	Normal
Members grouping	yes	No
Secure A_i number of bits	Formalized	Not specified
Number of $\mathbb{F}_q OPS$	$s({b_{LU}}^2 + 2b_{LU})$	$n^3 + 2n^2$
Prone to linearly dependency	no	yes
linearly dependency check	not required	required
linearly dependency affected block	(H, q)	(n, n)
can be solved in parallel	Yes	No

5.1.6 Discussion

Proper key management is crucial for the effective use of cryptography in ensuring security. KeyShield is a proposed key management scheme that balances various factors, including forward/backward secrecy, minimal trust, guessing resistance, quantum resistance, scalability, computational cost, storage overhead, messages overhead, and entity management.

The use of a linear system of equations as a key establishment protocol in the literature suffers from high computational complexity. KeyShield addresses this issue through two main solutions: first, member grouping, in which a secret vector is assigned to a set of members instead of a single member while maintaining the same level of security; second, the use of a banded matrix instead of a full secret matrix, which keeps the difficulty of

guessing a matrix row (secret vector) higher than the cost of guessing the shared key (TEK).

KeyShield's operations are supported by mathematical analysis and proofs when necessary. It opens a new door for utilizing the difficulty of finding a unique solution to an underdetermined linear system of equations, thus more research and innovation in this direction are needed.

5.2 KeyShield2 Protocol

KeyShield is a protocol that utilizes a unique method for solving a linear system of equations over a finite field $\mathbb{F}_{pr}$, where pr is a prime number. In KeyShield, members receive a secure lock in a public channel. They are able to obtain the TEK (encryption key) by applying their own secret vector to unlock the secure lock. Equation (5.1) shows the linear system $A \times X = B$ used as a key distribution mechanism in this protocol. X represents the secure lock that is shared in a public channel, B is a vector of the temporary encryption key (TEK) that is obtained by unlocking the lock, and A is a secret banded matrix. Each row of the matrix, $A_i = [a_{i,1} \dots a_{i,n}]$, is referred to as the secret vector and is assigned to a particular member. The members are able to obtain the TEK by applying their own secret vector to unlock the secure lock, as represented by the linear system $A_i \times X = TEK$.

KeyShield has a number of potential use cases:

- Its primary function is to deliver a single key to all members of a group. The key center generates the TEK and fills the vector B with the generated key, $B = [TEK, .., TEK]$.
- KeyShield can also be used to deliver unicast private keys to each member of the group. To do this, the keys are simply placed in the B vector, $B = [K_A, K_B, .., K_Z]$.
- KeyShield can be utilized to distribute keys to subgroups within a larger group. For example, if the key server decides to divide the members into three subgroups,

a single KeyShield system can be used to distribute three different keys to each subgroup, $B = [[K_1^{G1}, ..., K_k^{G1}], [K_1^{G2}, ..., K_k^{G2}], [K_1^{G3}, ..., K_k^{G3}]]$.

In summary, KeyShield is a protocol that allows for the secure distribution of keys to members of a group. The key server generates the desired key(s) and solves a system of equations using the secret vectors of each member and the generated key(s). The solution, X, is then broadcast to all members in an open channel. Each member is able to retrieve the shared key(s) by multiplying their secret vector A_i by the solution vector X, see Equation 5.2. Whether the key being distributed is a private key, a group key, or a subgroup key, all members follow the same procedure, $A_i \times X$, to extract the key.

KeyShield utilizes a banded system to minimize computation costs but has some vulnerabilities. In this system, a secret vector A_i will have leading and trailing zeros, with the only non-trivial entries being a_{ji} with $b_L \leq i \leq n - b_U$, where $m = n - b_U - b_L$ is the maximum effective row width. To further increase security, KeyShield injects random entries at the beginning and end of the TEK vector. This helps to protect against attacks on the system and ensure the security of the key distribution process,

$$B = [[r_1, ..., r_{b_L}], [TEK, .., TEK], [r_{n-b_U}, ..., r_n]], \tag{5.24}$$

$$B = [[r_1, ..., r_{b_L}], [K_A, K_B, .., K_Z], [r_{n-b_U}, ..., r_n]] \tag{5.25}$$

$$B = [[r_1, ..., r_{b_L}], [K_1^{G1}, ..., K_k^{G1}], [K_1^{G2}, ..., K_k^{G2}], [K_1^{G3}, ..., K_k^{G3}], [r_{n-b_U}, ..., r_n]]. \tag{5.26}$$

This means that each member m_i only needs to consider the window $X_j^{[b_L, n-b_U]}$ of the secure lock. The number of members in each window is limited, and the various windows $X_j^{[b_L, n-b_U]}, X_k^{[b_L, n-b_U]}$ will not be scalar multiples of one another. To facilitate this process,

KeyShield defines the $[b_L, n - b_U]$-truncation of a vector as follows:

$$\overline{X_j}^{[b_L, n-b_U]} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}. \tag{5.27}$$

5.2.1 KeyShield's Cryptanalysis

KeyShield has been subjected to cryptanalysis by Nicholas Touikan [1]. This analysis identifies several potential attacks on the protocol, which are primarily related to the use of the banded system and insufficient randomness.

In the following sections, I provide a comprehensive analysis of the vulnerabilities identified in KeyShield. I also examine the potential impact of these vulnerabilities on the security of the protocol and propose potential countermeasures to address these issues. It should be noted that these vulnerabilities have been addressed in the improved version of the protocol, KeyShield2.

5.2.1.1 Scalar product vulnerability

For this vulnerability, let us will assume that membership does not change. This vulnerability called *scalar product vulnerability*, can be exploited to use the secure locks X_j in order to gain information on the TEK_j, which in some cases will be sufficient for an adversary to completely recover TEK_j. The TEK-ratio, $X_j = \frac{TEK_j}{TEK_i} X_i$, can be easily computed from two secure locks X_i and X_j, which are obtained by solving the linear

[1] Touikan is an Associate Professor at the Department of Mathematics and Statistics of the University of New Brunswick.

system of equations $A \times X = B$ for two different $TEKs$. If

$$X_j = \begin{bmatrix} x_{1j} \\ \vdots \\ x_{nj} \end{bmatrix}, X_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{nk} \end{bmatrix} \tag{5.28}$$

then $\frac{TEK_j}{TEK_k} = \frac{x\ell j}{x_{\ell k}}, \ell = 1, \ldots, n$. We summarize this as follows.

Proposition 1. Without the inclusion of random elements, an adversary can determine the ratio $\frac{TEK_i}{TEK_j}$ directly from the secret locks X_i and X_j. In this case, if an adversary gains access to a single TEK_i, they can determine all other $TEKs$.

5.2.1.2 Un-obfuscation

The most obvious way to remove the scalar product vulnerability is to inject random rows into the TEK vector and the matrix A, this is actually done using banded linear systems. The secure lock X_j is constructed by solving,

$$\begin{bmatrix} R_1 \\ \vdots \\ R_{b_L} \\ \hline A_1 \\ \vdots \\ A_m \\ \hline R_{m+b_L+1} \\ \vdots \\ R_{n,} \end{bmatrix} X_j = \begin{bmatrix} r_1 \\ \vdots \\ r_{b_L} \\ \hline TEK_j \\ \vdots \\ TEK_j \\ \hline r_{m+b_L+1} \\ \vdots \\ r_n \end{bmatrix}, \tag{5.29}$$

where R_i and r_i are random row vectors and numbers, respectively. As a result of the injected random entries, the secure locks X_j are not scalar multiples of each other.

Considering the vector space $W = \mathbb{F}^n$, let us define the member subspace $W_m = \mathbb{F}^m = \text{span}\,(A_1, \ldots, A_m)$ and the random subspace $R_u = \mathbb{F}^u == \text{span}\,(R_1, \ldots, R_m)$ such that $W = W_m + R_u$, where $u = n - m$. The list of vectors $(A_1, \ldots, A_m)$ in W_m are considered the basis if and only if every vector $v \in W_m$ can be written uniquely in the form

$$v = \sum_{i=1}^{m} a_i A_i, \tag{5.30}$$

where $a_i \in \mathbb{F}$.

The secure lock X_j rises to a linear mapping $L_{X_j} : W \to \mathbb{F}$. When this mapping is restricted to the member subspace, different secure keys we get

$$L_{X_j}\big|_{W_m} = \lambda \cdot L_{X_k}|_{W_m}\,, \tag{5.31}$$

for some $\lambda \in \mathbb{F}$.

A vector space W_m in $\mathbb{F}^m$ can be spanned by many different bases, including the standard basis $\{(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)\}$, a basis consisting of the columns of an invertible $m \times m$ matrix A, denoted $(A_1, \ldots, A_m)$, or any other basis. Suppose an adversary has obtained a basis $\{e_1, e_2, \ldots, e_m\}$ for a subspace W_m of $\mathbb{F}^m$. Then they can find a linear map L_{W_m} such that $range\ L_{W_m} = W$. L_{W_m} helps to map the basis $\{e_1, \ldots, e_m\}$ of $\mathbb{F}^n$ to a basis for the entire space $\mathbb{F}^n$ This results in a new basis $\{e_1, \ldots, e_n\}$ for $\mathbb{F}^n$. Then, any vector $v \in \mathbb{F}^n$ can be written as a linear combination of the basis vectors:

$$v = \sum_{i=1}^{n} a_i e_i. \tag{5.32}$$

Therefore, the dot products of the linear mappings L_{X_i} and L_{W_m}, $L_{X_i} \circ L_{W_m}$, allow the adversary to map directly from $\mathbb{F}^m$ to $\mathbb{F}$, $L_{X_i} \circ L_{W_m} : W_m \to \mathbb{F}$. This can be summarized as follows:

Proposition 2. Given the obfuscated KeyShield protocol, if an adversary knows the mem-

ber subspace, then they reduce the problem to the unobfuscated KeyShield. In particular, they can find TEK ratios.

If an adversary is able to gain access to a single TEK_i, it may be possible to compromise all other TEKs in the unobfuscated case.

5.2.1.3 Fake secret vector attack

If an adversary has access to multiple pairs of secure locks (X_j) and their corresponding shared keys (TEKj), they may be able to construct a fake secret vector (A_{fake}) that allows them to unlock future secure locks. This is possible due to the fact that the dimension of the nullspace (kernel) of any linear map L_{X_j} from the vector space $\mathbb{F}^n$ to the base field $L_{X_j} : \mathbb{F}^n \to \mathbb{F}$ is equal to the dimension of the vector space minus one, $\dim(\ker(L)) = \dim(\mathbb{F}^n) - 1$. Since scalar multiples of linear maps have the same nullspace, for all j the intersection of the nullspace of L_{X_j} and the member subspace W_m is constant, K_m,

$$\ker\left(L_{X_j}\right) \cap W_m = K_m, \tag{5.33}$$

where $K_m \subset W_m$ is a fixed subspace of dimension $m - 1 = \dim(W_m) - 1$. Now the dimension of $\ker L_{X_j}$ is equal to $n - 1$, $\dim\left(\ker L_{X_j}\right) = n - 1$, meaning that these are codimension 1 subspaces of $\mathbb{F}^n$. In general, the intersection of ks codimension 1 subspaces $S_1, \ldots, S_{ks}$ has dimension $n - ks$ for $ks < n$, i.e.

$$\dim\left(\bigcap_{r=1}^{ks} S_{ks}\right) = n - ks. \tag{5.34}$$

However, in this case, the codimension 1 subspaces $\ker\left(L_{X_j}\right)$ have the additional constraint that they all contain K_m. Given secure locks $X_1, \ldots, X_N$ for any $N \geqslant n - m + 1$, with high probability the intersection of these subspaces will be K_m, i.e.

$$\bigcap_{j=1}^{N} \ker\left(L_{X_j}\right) = K_m. \tag{5.35}$$

This can be summarized as the following proposition:

Proposition 3. Given $N \geqslant n - m + 1$ secure locks $X_1, \ldots, X_N$, an adversary can, with high probability, compute the codimension 1 subspace K_m of the member space W_m.

Consider the scenario in which an adversary has secure locks $X_1, \ldots, X_N$ with $N \geqslant n-m$ and corresponding $TEKs$ $\{TEK_1, \ldots, TEK_N\}$. In this case, the adversary can solve for A_{fake} in the system of equations

$$\begin{cases} A_{\text{fake}} \cdot X_1 = TEK_1 \\ \qquad \ldots \\ A_{\text{fake}} \cdot X_N = TEK_N \end{cases} \tag{5.36}$$

The solution is not unique, as all legitimate secret vectors A_i will also solve this equation. However, we have

$$A_{\text{fake}} = A_i + v, v \in K_m. \tag{5.37}$$

Since $K_m \subset W_m$, it follows that $A_{\text{fake}} \in W_m$. Moreover, due to the scalar product property, we have that for all subsequent secure locks $X_{j'}$, $A_{\text{fake}} X_{j'} = TEK_{j'}$, thus compromising all subsequent TEKs. We can express A_{fake} as a linear combination of the secret vectors $A_1, \ldots, A_m$:

$$A_{\text{fake}} = \alpha_1 A_1 + \cdots + \alpha_m A_m, \tag{5.38}$$

where $\alpha_1 + \cdots + \alpha_m = 1$.

We summarize this discussion as follows:

Proposition 4. Given the hypothesis of Proposition 3, if an adversary also gains access to the corresponding $TEKs$ $\{TEK_1, \ldots, TEK_N\}$ then the adversary can determine the member space and compute all subsequent $TEKs$.

Corollary 1. Once sufficiently many $TEKs$ are sent, any member m_i can compute the member subspace W_m.

It is worth noting that although member m_i can compute the subspace $\mathrm{span}(A_1, \ldots, A_m)$, it is still information-theoretically impossible for them to determine the other secret vectors.

5.2.1.4 Updating secret vector Vulnerability

Suppose that the member m_1 is removed and replaced by a member m_1' with new secret vector A_1'. Suppose furthermore, we are using an obfuscated KeyShield scheme. Corollary 1 immediately gives:

Proposition 5. If the new secret vector A_1' is a linear combination of the old secret vectors, i.e $A_1' \in \mathrm{span}\,(A_1, \ldots, A_m)$ then the member subspace W_m remains invariant. In particular, the ex-member m_1 can still determine the ratios of future TEKs, $TEK_2', \ldots, TEK_N'$

5.2.1.5 Window Vulnerability

In the case where a banded linear system is used, a secret vector A_i will have many leading and trailing zeroes the only nontrivial entries being a_{ji} with $t_i \leqslant i \leqslant b_j$, where $b_j - t_i = w$ is something the maximum effective row width. This means that the member m_i only needs to care about the window $X_j^{[t_i,b_i]}$ of the secure lock.

Furthermore, because randomness is injected at the beginning and end of the TEK vector as in equation (6) of [AdGL20], the various windows $X_j^{[t_i,b_i]}, X_k^{[t_i,b_i]}$ will not be scalar multiples of one another. Let us define the $[t, b]$-truncation of a vector to be

$$\overline{X_j}^{[t,b]} = \begin{bmatrix} x_t \\ x_{t+1} \\ \vdots \\ x_b \end{bmatrix}$$

which is a vector of length $b - t$. I don't have the energy to prove this, but I'm pretty sure this is true. The idea is that the truncated secure locks become random mappings, but whose kernels always avoid the truncated secret vector; thus sufficiently many of them (along with the TEKs) will pick out the secret vector exactly.

Conjecture 1. (Window attack). In a KeyShield scheme that uses a banded system with a maximum effective row with w, either no randomization is used, in which case the scalar product property holds. If randomization is used then for $N \geqslant w$, with high probability the system of equations

$$\begin{cases} Y \cdot \overline{X_1}^{[t_i,b_i]} = \text{ TEK } K_1 \\ \qquad\qquad \cdots \\ Y \cdot \overline{X_N}^{[t_i,b_i]} = TEK_N \end{cases}$$

will admit the unique solution $Y = \overline{A_i}^{[t_i,b_i]}$. In particular, any member m_j can determine the secret vector of any other member because they know the TEKs.

The use of banded linear systems where each member receives a vector with a small window populated with numbers and the rest are zeros, can create a vulnerability that an attacker may exploit. Specifically, an attacker may be able to compromise the security of the system by taking advantage of the possibility that a member's window may not change if it is small.

5.2.2 Countermeasures

In order to protect against these potential vulnerabilities, I now propose several countermeasures to ensure the security of the KeyShield protocol. These measures aim to prevent an adversary from compromising the system or obtaining sensitive information. It is important to note that the updated version of the protocol, called KeyShield2, will incorporate these countermeasures to ensure the security of the system.

Countermeasure 1. To secure the system against potential attacks, it is advisable to masquerade the banded system. This means that the attacker will not be able to exploit the smaller window in attempting to break the system. Recall the basic equation:

$$AX_j = B_j.$$

The server will solve the leaner system and send the secure lock (X_j) to every member. Each member then obtains the shared TEK through,

$$A_i \times X_j = TEK.$$

The matrix A is banded, meaning that it has many leading and trailing zeroes with only m no-zeros entries. To maintain the simplicity and speed benefits of using a banded system, we can apply a preconditioner P to mask it. The preconditioner is assumed to be well-conditioned, fully ranked, and totally random. This leads to the following preconditioned system:

$$(\bar{A}P^{-1})(PX_j) = B_j \tag{5.39}$$

$$\bar{A}Z_j = B_j, \tag{5.40}$$

where $\bar{A} = AP^{-1}$ is the masked secret matrix and $Z_j = PX_j$ is the masked secure lock. To implement this countermeasure, the key server performs the following actions:

1. Given that the matrix A and the preconditioner P are available, the key server computes the masked matrix $\bar{A}$, which is not banded. The server then sends the vectors of $\bar{A}$ to the members as their secret vectors (i.e., the secret vector $\bar{A}_i$ is assigned to member i instead of A_i from the banded matrix).

2. The key server selects the shared key TEK and prepares the B vector.

3. The server solves the simple banded system $A_i \times X_j = B$.

4. The server computes the masked secure lock $Z_j = PX_j$ and broadcasts it.

Each member obtains the share TEK through

$$\bar{A}_i \times Z_j = TEK.$$

This technique hides the structure of the main matrix and prevents the sliding window attack.

Countermeasure 2. To increase the security of the KeyShield protocol, it is important to incorporate randomness into the system and to ensure the secrecy of the $TEKs$ even after they have been used. Before masking, the system can be represented as in Equation 5.29, which contains two separated sections that are fully random and are replaced every time there is a membership change. However, this may not be sufficient as they are not within the member area $[A_1, \ldots, A_m]$, which would change the member subspace window. To further enhance security, it is recommended to insert several random vectors within the middle area of the matrix A. These vectors can be placed randomly and should be replaced every time there is a membership change.

Countermeasure 3. In the re-keying process, it is recommended that the secret vector A_i be replaced with a new, totally random, vector $A_i{}'$. The summation procedure used in KeyShield to maintain linear independence is not secure, as it preserves the member subspace, making it easier for an attacker to target. Instead, the new secret vector should be generated as a totally random vector, while still maintaining the independence conditions. This will help to improve the security of the system.

Countermeasure 4. It is important to note that KeyShield originally relied on the NTRU algorithm for distributing the private secret vectors, but NIST did not include NTRU in the fourth round of its competition and recommended replacing NTRU with the Kyber

algorithm, which is currently the only winner of the competition. This will ensure that KeyShield is using the most secure and up-to-date algorithm for distributing private secret vectors.

Countermeasure 5. One effective countermeasure to prevent the compromise of the secret vectors is to update them frequently, even in the absence of any changes in membership. This countermeasure can be implemented by setting a maximum limit on the number of times a secret vector can be used. Specifically, the secret vector A_i should not be used for more than n times, and the maximum number of times a secret vector can be used to solve the linear system and obtain secure locks X is limited to $n - 1$. By enforcing this countermeasure, the risk of an attacker obtaining the secret vectors and compromising the security of the system is significantly reduced.

5.2.3 Security

KeyShield2 employs A_L as the security parameter. Although guessing the secret vector could break the system, the secret vector's bit length, A_L, is always greater than the traffic key's bit length, TEK_L. Ensuring $A_L \geq TEK_L$ increases the difficulty of guessing the secret vector compared to guessing the TEK itself. To further illustrate this point, the attacker's problem can be formulated as finding A_i such that:

$$A_i \times X = TEK$$

where $A_i \in \mathbb{F}_{pr}^{(1\times n)}$ is the unknown secret vector. Since the matrix A is composed of secret row vectors distributed securely to the members, the attacker has no information about A_i. Therefore, the attacker has to search for A_i in a space of size pr^n. If pr is large enough, the search space becomes enormous, making the attacker's problem computationally infeasible. To quantify the difficulty of the attacker's problem, let's consider the probability that the attacker can guess the correct A_i in a single attempt. Since there are pr^n possible

values for A_i, the probability of a successful guess is:

$$\frac{1}{pr^n} \tag{5.41}$$

This probability is extremely small when pr is large, making it practically impossible for the attacker to guess the correct A_i in a single attempt. As the attacker has no other information about A_i or TEK, their chances of finding the correct values do not improve over multiple attempts.

5.2.4 Indistinguishability

The linearity of the system contributes to KeyShield2's indistinguishability under CPA and CCA.

- **IND-CPA:** In the scenario of indistinguishability under Chosen-Plaintext Attack (IND-CPA), the attacker has the ability to choose plaintexts ($TEKs$) and obtain their corresponding ciphertexts (Xs). The attacker selects two keys, say TEK_1 and TEK_2, and the key server generates two secure locks, X_1 and X_2, by solving the linear system and sharing the resulting secure locks. Because the secret matrix A and vector B have randomness injected into them (see Equation 5.29 and Countermeasure 2), the attacker should not be able to distinguish between X_1 and X_2. This property is critical in ensuring that the KeyShield2 scheme is secure against chosen-plaintext attacks, where an attacker may gain some information about the secret key by observing the resulting ciphertexts.

  To show mathematically how an adversary cannot distinguish between X_1 and X_2 due to the randomness in A and B, we can make the following arguments:

  1. **Uniform Distribution of Non-Zero Elements in $\bar{A}$ and and all elements in P:** The uniform distribution of non-zero elements in $\bar{A}$ and all elements

in P ensures that no specific pattern or relationship can be established by an adversary between different instances of A, and hence between X_1 and X_2. The randomness in these elements contributes to the unpredictability of A.

2. **Statistical Distance:** The statistical distance might not be zero due to the banded structure of $\bar{A}$. However, it can still be negligible if the size of the finite field $\mathbb{F}_{pr}$ is large enough, and if the number of non-zero elements in each row of $\bar{A}$ is relatively high.

The statistical distance between two distributions D_1 and D_2 is defined as:

$$\Delta(D_1, D_2) = \frac{1}{2} \sum_{x \in \mathbb{F}_{pr}^n} |D_1(x) - D_2(x)| \tag{5.42}$$

Due to the randomness in A and B, and given that they are uniformly distributed over $\mathbb{F}pr$, the distributions D_1 and D_2 will also be uniformly distributed over $\mathbb{F}pr^n$. $\mathbb{F}pr^n$ represents a finite field of size pr^n, where pr is a prime number and n is the dimension of the field. The size of the finite field, denoted as $|\mathbb{F}pr^n|$, is equal to pr^n. This implies that for any $x \in \mathbb{F}_{pr}^n$, we have:

$$D_1(x) = D_2(x) = \frac{1}{|\mathbb{F}_{pr}^n|} = \frac{1}{pr^n} \tag{5.43}$$

Therefore, the statistical distance between D_1 and D_2 is:

$$\Delta(D_1, D_2) = \frac{1}{2} \sum_{x \in \mathbb{F}pr^n} |D_1(x) - D_2(x)| \tag{5.44}$$

If the statistical distance is negligible, D_1 and D_2 (the distributions of X_1 and X_2) will be virtually indistinguishable, and an adversary cannot distinguish between X_1 and X_2 based on the resulting ciphertexts.

- **IND-CCA:** In an IND-CCA scenario, an attacker chooses ciphertexts (X) and obtains corresponding plaintexts ($TEKs$) using the secret vector A_i: $A_i \times X = TEK$. The attacker sends two ciphertexts, X_1 and X_2, to a member, who calculates TEK_1 and TEK_2 and sends them back. For KeyShield2 to be secure against chosen-ciphertext attacks, the attacker must not be able to distinguish between TEK_1 and TEK_2 without knowing A_i, as infinitely many vectors can satisfy $A_j \times X = TEK$.

  The IND-CCA security game involves a challenger and an adversary. The adversary has access to a decryption oracle $\mathcal{O}_{\text{Dec}}$, which takes a ciphertext X' and returns the corresponding plaintext TEK' using a chosen member's secret vector A_i.

  1. The adversary selects two ciphertexts, X_1 and X_2, and sends them to the challenger.
  2. The challenger randomly chooses $b \in 0, 1$ and provides $TEK_b = A_i \times X_b$.
  3. The adversary guesses b based on information from $\mathcal{O}_{\text{Dec}}$ and the ciphertexts.

  Let b' be the adversary's guess. The adversary's advantage is defined as

$$\text{Adv}_{\text{IND-CCA}} = |\Pr[b' = b] - \frac{1}{2}|. \tag{5.45}$$

  We want to show that this advantage is negligible. Since there exists a many-to-one mapping between A_j and the resulting TEK, the adversary cannot uniquely link a TEK to its corresponding X, and therefore, cannot determine A_i with a non-negligible advantage over random guessing.

  As the adversary cannot distinguish between TEK_1 and TEK_2 without knowing the secret vector A_i, their advantage in the IND-CCA security game is negligible, demonstrating that KeyShield2 is secure against chosen-ciphertext attacks.

5.2.5 Performance

We have assessed KeyShield2's efficiency by examining a variety of performance metrics. These factors encompass: the core mathematical problem; the nature of the protocol (whether it is static or dynamic, i.e., accommodating membership alterations with members joining and departing at any moment); the quantity of protocol rounds demanded; the total number of broadcast messages involved; and the count of peer-to-peer messages needed. Table 5.4 offers a comparative overview, highlighting KeyShield2's effectiveness as a dynamic protocol that requires merely a single message to update the key.

Table 5.4: A comparative analysis of efficiency parameters between recent quantum-safe protocols and KeyShield2.

Protocol	Year	Core	Type	Rounds	#Broadcast	#P2P
[56]	2019	Isogeny	Static	1	n	0
[8]	2019	Lattice	static	3	$2n+1$	0
[109]	2019	Compiler (KEM)	Static	3	n	$2n$
[46]	2020	Lattice	Dynamic	3	$2n+1$	0
[63]	2020	Compiler (KEM)	Static	2	n	n^2-n
[54]	2020	Lattice	Static	4	$2n$	$2n$
[53]	2022	Lattice	Static	4	$2n$	$2n$
[107]	2022	Compiler (KEM)	Static	4	$2n$	$2n$
Ours	Now	lattice-Linear System	Dynamic	1	1	n

5.2.6 Discussion

KeyShield2 is an updated version of KeyShield that addresses the vulnerabilities identified in the original protocol. The recommended countermeasures do not significantly impact the performance or efficiency of the protocol. The key server in KeyShield2 solves a banded linear system, which allows for faster computation. Additionally, KeyShield2 requires additional computation for the following tasks:

- Masquerading the secret vectors with a simple matrix multiplication of AP^{-1}, which is done on a large scale for the whole matrix A at the beginning of the protocol.

- Masking a new secret vector $\bar{A}_i P^{-1}$ with a simple multiplication process during membership changes.

- Computing the TEK with a simple multiplication and addition operation, $TEK = A_i \times X_j$. In KeyShield2, this is done with the full system size of n rows, whereas in the original KeyShield, it was done with a smaller window size of m rows.

Overall, the key server and members in KeyShield2 perform simple multiplication and addition operations, and the extra computation cost is not significant.

5.3 Concluding remarks

In conclusion, in this chapter, I have proposed two quantum-safe protocols, KeyShield and KeyShield2, for secure group key establishment. KeyShield is a novel key management scheme that balances various factors, including forward/backward secrecy, minimal trust, guessing resistance, quantum resistance, scalability, computational cost, storage overhead, messages overhead, and entity management. It addresses the high computational complexity of using a linear system of equations as a key establishment protocol through member grouping and the use of a banded matrix.

KeyShield2 is an improved version of the KeyShield protocol that addresses vulnerabilities found in the original protocol. The recommended countermeasures do not negatively impact the performance or efficiency of the protocol. KeyShield2 includes a faster key server that solves a banded linear system, as well as additional computation for masquerading secret vectors, masking new secret vectors during membership changes, and computing the TEK. The extra computation cost is not significant, and the key server and members in KeyShield2 mainly perform simple multiplication and addition operations.

The proposed protocols offer new possibilities for leveraging the challenge of finding a unique solution for an underdetermined linear system of equations. Further research and innovation in this direction are necessary.

Chapter 6

Quantum-Based Key Establishment

In this chapter, I will discuss the QKeyShield and DGL22 protocols in detail. These protocols utilize the principles of quantum mechanics to establish a shared secret key and are designed to resist common attacks. Both QKeyShield and DGL22 are RDI-QKD protocols, meaning that the key generation process does not rely on the specific characteristics of the measurement devices used by Bob. I will provide a comprehensive overview of the specifications of both protocols. Then I will discuss their similarities and distinctions in detail.

6.1 QKeyShield Protocol

In a scenario where an organization wants to establish secret keys with its customers, there may be a risk that customers' measuring devices, which could be built by an adversary, are low-cost and unsafe. To address this, I propose using a type of quantum key distribution called RDI-QKD, in which the measuring device and entanglement sources used by the organization (Alice) are trusted, but the customers' measuring devices (Bob) are not. To ensure security, Alice and Bob use both a quantum channel for transmitting a swapping qubit and a classical channel for information reconciliation and privacy amplification. The proposed protocol, called QKeyShield, is outlined in detail in Figure 6.1. Bob's measure-

ment device is uncharacterized and protected from any information leakage to the outside. The system contains six qubits: $(1, 2, A, B)$ are prepared by Alice and $(3, 4)$ are prepared by Bob. The straight line that connects the qubits represents their entanglement. The double straight line links the qubits on which BSM is performed. Alice (Bob) randomly performs Hadamard on qubit A (B) and performs a bit-flip operation when required on qubit 2 (4). The following steps outline the protocol:

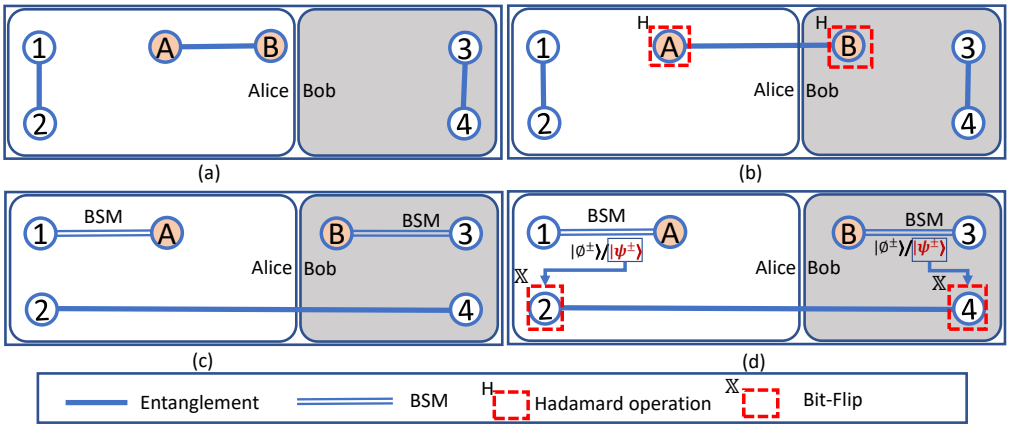


Figure 6.1: QKeyShield Protocol. The diagram illustrates the steps of the QKeyShield protocol for establishing secret keys with customers.

1. **Hadamard operation probability :** Alice and Bob choose a number $0 < p \leq \frac{1}{2}$ whose value is publicly revealed. p represents the probability of performing Hadamard operation and $(1 - p)$ the probability of performing the identity operator. Assume q_{total} is the total number of transmitted qubits. Bob receives a series of q_{total} qubits from Alice. The value of p is set in such a way that $q_{total}(p^2 - \lambda) = n_1 = \Omega(logq_{total})$, where λ is a small positive number (i.e., the error due to statistical fluctuations) chosen by Alice and Bob. n_1 and n_2 are the numbers of test samples chosen from the subsets where Alice and Bob both perform the Hadamard operator or identity operator, respectively.

2. **Initial states preparation:** Alice prepares two entangled states, $|\phi^+\rangle_{12}$ and $|\phi^+\rangle_{AB}$, and Bob prepares an entangled state, $|\phi^+\rangle_{34}$, see Figure 6.1(a). The initial quantum

state of the whole system containing the qubits $1, 2, A, B, 3$, and 4 can be written as:

$$\begin{aligned} |\Psi\rangle_{12AB34} &= |\phi^+\rangle_{12} \otimes |\phi^+\rangle_{AB} \otimes |\phi^+\rangle_{34} \\ &= \frac{1}{2\sqrt{2}}(|00\rangle_{12} + |11\rangle_{12}) \otimes (|00\rangle_{AB} + |11\rangle_{AB}) \\ &\otimes (|00\rangle_{34} + |11\rangle_{34}), \end{aligned} \tag{6.1}$$

$$\begin{aligned} |\Psi\rangle_{12AB34} = \frac{1}{2\sqrt{2}}(&|000000\rangle + |000011\rangle + |001100\rangle \\ &+ |001111\rangle + |110000\rangle + |110011\rangle \\ &+ |111100\rangle + |111111\rangle). \end{aligned} \tag{6.2}$$

3. **Preparing Swapping state:** Alice produces a maximally entangled Bell state $|\phi^+\rangle_{AB}$.

4. **Sending Swapping qubit :** Alice keeps the qubit (A) of the maximally entangled state $|\phi^+\rangle_{AB}$ for herself and sends the qubit (B) to Bob, see Figure 6.1.(b).

5. **Performing Hadamard operation:** Alice (Bob) chooses between the two operators, Hadamard, and identity, to perform on her (his) qubit A (B) with probabilities p and $(1-p)$, respectively. If both qubits are simultaneously subjected to the Hadamard operation, the initial state $H^{(1)}H^{(2)}|\phi^+\rangle_{AB} = |\phi^+\rangle_{AB}$ remains intact. However, if the operations are performed successively, with Alice going first, the state becomes $|W^+\rangle\, AB$, as shown in Figure 6.1.(b). Bob then receives qubit B and applies the Hadamard operation, undoing Alice's operation and returning the state to $|\phi^+\rangle\, AB$. Alice randomly chooses to perform a Hadamard operation on her qubit A before of after sending qubit B to Bob. If she performs the Hadamard operation after sending, Eve intercepts qubit B of the state $|\phi^+\rangle_{AB}$. If she performs the Hadamard operation before sending, Eve intercepts qubit B of the state $|W^+\rangle_{AB}$. Whether this operation is conducted before or after sending; it won't affect the measurement outcomes; however, it secures the transmission of the qubit B. The security benefit of

Alice's random decision is discussed in Section 6.1.5.1.

6. **Bell state measurement (BSM):** Qubits $1, 2$ and A belong to Alice, and qubits $B, 3$ and 4 belong to Bob. Alice and Bob perform BSM on qubits 1 and A, and B and 3, respectively. Conducting BSM results in projecting the remaining qubits (2 and 4) belonging to Alice (Bob), into one of the Bell states $|\phi^{\pm}\rangle_{24}$ or $|\psi^{\pm}\rangle_{24}$,

$$
\begin{aligned}
|\Psi\rangle_{\mathbf{1AB3}24} = \frac{1}{2}\Big[\Big(& |\phi^+\rangle_{1A}|\phi^+\rangle_{B3}|\phi^+\rangle_{24} + |\phi^+\rangle_{1A}|\phi^-\rangle_{B3}|\phi^-\rangle_{24} \\
& + |\phi^-\rangle_{1A}|\phi^+\rangle_{B3}|\phi^-\rangle_{24} + |\phi^-\rangle_{1A}|\phi^-\rangle_{B3}|\phi^+\rangle_{24}\Big) \\
& + \Big(|\phi^+\rangle_{1A}|\psi^+\rangle_{B3}|\psi^+\rangle_{24} + |\phi^+\rangle_{1A}|\psi^-\rangle_{B3}|\psi^-\rangle_{24} \\
& + |\phi^-\rangle_{1A}|\psi^+\rangle_{B3}|\psi^-\rangle_{24} + |\phi^-\rangle_{1A}|\psi^-\rangle_{B3}|\psi^+\rangle_{24}\Big) \\
& + \Big(|\psi^+\rangle_{1A}|\psi^+\rangle_{B3}|\phi^+\rangle_{24} - |\psi^+\rangle_{1A}|\psi^-\rangle_{B3}|\phi^-\rangle_{24} \\
& + |\psi^-\rangle_{1A}|\psi^+\rangle_{B3}|\phi^-\rangle_{24} - |\psi^-\rangle_{1A}|\psi^-\rangle_{B3}|\phi^+\rangle_{24}\Big) \\
& + \Big(|\psi^+\rangle_{1A}|\phi^+\rangle_{B3}|\psi^+\rangle_{24} - |\psi^+\rangle_{1A}|\phi^-\rangle_{B3}|\psi^-\rangle_{24} \\
& + |\psi^-\rangle_{1A}|\phi^+\rangle_{B3}|\psi^-\rangle_{24} - |\psi^-\rangle_{1A}|\phi^-\rangle_{B3}|\psi^+\rangle_{24}\Big)\Big].
\end{aligned}
\tag{6.3}
$$

Knowing their own BSM results, Alice and Bob could determine which Bell state they are sharing $|\phi^{\pm}\rangle_{24}$, or $|\psi^{\pm}\rangle_{24}$. They can decide their desired states before starting the protocol, and we do not assume such information is secret. If they choose $|\phi^{\pm}\rangle_{24}$ as their desired states, they will share the same secret bit "0" or "1" when measuring qubits 2 and 4. Thus, they have to agree on a mechanism that makes sure their shared state is always $|\phi^{\pm}\rangle_{24}$.

7. **Desired state preparation:** If they choose $|\phi^{\pm}\rangle_{24}$ as their desired state, Alice(Bob) should perform a quantum bit-flip to her(his) unmeasured qubit $2(4)$ only if her(his) own measurement is $|\psi^{\pm}\rangle_{1A}(|\psi^{\pm}\rangle_{B3})$,

$$|b\rangle_{2(4)} = \begin{cases} |\bar{b}\rangle_{2(4)}, & if \quad |\psi\rangle_{1A(3B)} = |\psi^{\pm}\rangle \\ \\ |b\rangle_{2(4)}, & otherwise, \end{cases} \tag{6.4}$$

where $|b\rangle_{2(4)}$ represents Alice's(Bob's) qubit 2(4) and the bar above $|b\rangle$ indicates its logical negation. By doing so, Alice and Bob collaboratively, decentrally, and without classical communication prepare the desired state between them, see Figure 6.1.(d) and Table 6.1. The system described in Equation (6.3) can be found in one of 16 possible outcomes upon measurement. These possibilities are grouped into 4 possibilities based on the obtained states by both Alice and Bob:$|\phi^{\pm}\rangle_{1A}|\phi^{\pm}\rangle_{B3}$, $|\phi^{\pm}\rangle_{1A}|\psi^{\pm}\rangle_{B3}$, $|\psi^{\pm}\rangle_{1A}|\psi^{\pm}\rangle_{B3}$, and $|\psi^{\pm}\rangle_{1A}|\phi^{\pm}\rangle_{B3}$. The default shared stated can be either $|\phi^{\pm}\rangle_{24}$ or $|\psi^{\pm}\rangle_{24}$. Using Equation (6.4), Alice (Bob) calculates the desired state. The final shared state is not always $|\phi^{+}\rangle_{AB}$, it might have a phase shift $|\phi^{-}\rangle_{AB}$; however, the probability of measuring the state in the computational basis (either 0 or 1) remains unchanged.

Table 6.1: Desired state preparation illustration.

Possible system states	Obtained state $\|\psi\rangle_{24}$	Alice's operator	Bob's operator	**Desired** state $\|\psi\rangle_{24}$
$\Big(\|\phi^{+}\rangle_{1A}\|\phi^{+}\rangle_{B3}\|\phi^{+}\rangle_{24} + \|\phi^{+}\rangle_{1A}\|\phi^{-}\rangle_{B3}\|\phi^{-}\rangle_{24}$ $+ \|\phi^{-}\rangle_{1A}\|\phi^{+}\rangle_{B3}\|\phi^{-}\rangle_{24} + \|\phi^{-}\rangle_{1A}\|\phi^{-}\rangle_{B3}\|\phi^{+}\rangle_{24}\Big)$	$\|\phi^{\pm}\rangle_{24}$	-	-	$\|\phi^{\pm}\rangle_{24}$
$\Big(\|\phi^{+}\rangle_{1A}\|\psi^{+}\rangle_{B3}\|\psi^{+}\rangle_{24} + \|\phi^{+}\rangle_{1A}\|\psi^{-}\rangle_{B3}\|\psi^{-}\rangle_{24}$ $+ \|\phi^{-}\rangle_{1A}\|\psi^{+}\rangle_{B3}\|\psi^{-}\rangle_{24} + \|\phi^{-}\rangle_{1A}\|\psi^{-}\rangle_{B3}\|\psi^{+}\rangle_{24}\Big)$	$\|\psi^{\pm}\rangle_{24}$	-	$\mathbb{X}\|b\rangle_4$	$\|\phi^{\pm}\rangle_{24}$
$\Big(\|\psi^{+}\rangle_{1A}\|\psi^{+}\rangle_{B3}\|\phi^{+}\rangle_{24} - \|\psi^{+}\rangle_{1A}\|\psi^{-}\rangle_{B3}\|\phi^{-}\rangle_{24}$ $+ \|\psi^{-}\rangle_{1A}\|\psi^{+}\rangle_{B3}\|\phi^{-}\rangle_{24} - \|\psi^{-}\rangle_{1A}\|\psi^{-}\rangle_{B3}\|\phi^{+}\rangle_{24}\Big)$	$\|\phi^{\pm}\rangle_{24}$	$\mathbb{X}\|b\rangle_2$	$\mathbb{X}\|b\rangle_4$	$\|\phi^{\pm}\rangle_{24}$
$\Big(\|\psi^{+}\rangle_{1A}\|\phi^{+}\rangle_{B3}\|\psi^{+}\rangle_{24} - \|\psi^{+}\rangle_{1A}\|\phi^{-}\rangle_{B3}\|\psi^{-}\rangle_{24}$ $+ \|\psi^{-}\rangle_{1A}\|\phi^{+}\rangle_{B3}\|\psi^{-}\rangle_{24} - \|\psi^{-}\rangle_{1A}\|\phi^{-}\rangle_{B3}\|\psi^{+}\rangle_{24}\Big)$	$\|\psi^{\pm}\rangle_{24}$	$\mathbb{X}\|b\rangle_2$	-	$\|\phi^{\pm}\rangle_{24}$

8. **Single Qubit Measurement (SQM):** Alice and Bob share one of the maximally entangled states $|\phi^{\pm}\rangle_{24} = \frac{1}{\sqrt{2}}(|00\rangle_{24} \pm |11\rangle_{24})$. Alice(Bob) performs an SQM on qubit 2 (4) on a computational basis because no alternative measurement is required.

They both get either 00 or 11.

9. **Repeat:** Alice and Bob repeat steps 1–8 until they get a sufficiently long string. For convenience, we consider protocol steps 1–8 as one round.

10. **Sifting:** Alice and Bob will independently follow the previous protocol steps without sharing their measurement results or the operators used with each other. Once Bob receives the q_{total} qubits, they begin the sifting phase. During this phase, Alice and Bob reveal the operators they used for each qubit through a public channel. Any results obtained from qubits where different operators were used are discarded, while the remaining results are retained for further analysis.

11. **Error estimation:** Alice and Bob split the approved results into two subsets based on the performed unitary operator (Hadamard or Identity). They then select a predetermined number of samples, say n_1, at random from the subset where they both conduct Hadamard operation, and compare their findings publicly. The estimated Quantum Bit Error Rate (QBER) on Hadamard operator $e_1 = \frac{r_1}{n_1}$ is determined by the number of mismatches r_1. Similarly, Alice and Bob choose a fixed number of instances, say n_2, at random from the subset where they both perform identity operations and compare their measurement findings publicly. $e_2 = \frac{r_2}{n_2}$ is the expected error rate based on the number of mismatches r_2. They require

$$e_1, e_2 < e_{max} - \lambda_e, \tag{6.5}$$

where e_{max} is a specified maximum allowable error rate and λ_e is a small positive value. According to Shor–Preskill [130], e_{max} is around 11%. If these two separate limitations are met, they proceed to the next step. Otherwise, they abort.

12. **Information reconciliation and privacy amplification:** Upon successful completion of the error estimation step, we have left with $(N - n_1 - n_2)$ measurement

results, referred to as the raw key. Alice and Bob then use classical algorithms to perform information reconciliation and privacy amplification on the raw key, resulting in a final secret key. The focus of this protocol is on the generation of the raw key. Discussions of privacy amplification algorithms, which are based on classical algorithms, fall outside the scope of this thesis on quantum key generation.

6.1.1 Error correction

Alice and Bob will share the maximally entangled state $|\phi^{\pm}\rangle$ 24 by following the protocol steps. However, due to decoherence, measurement errors, or other quantum noise, the sifted key may have some erroneous values as discussed in [77]. To combat this, they choose a probability p for performing the Hadamard operation and $(1-p)$ for performing the identity operator. Typically, p is chosen to be less than $\frac{1}{2}$ as the biased approach is followed. A fraction of the measurements where they both perform Hadamard is used for eavesdropper check and the remaining fraction, $1-p$, of measurements, are used to generate the key. QKeyShield always measures along the computational base. The final shared state might have a phase flip $|\phi^{+}\rangle\, AB$ or $|\phi^{-}\rangle_{AB}$; however, the probability of measuring the state on the computational basis(0 or 1) is unchanged. The phase-flip error rate does not affect the final measurement. The bit-flip error rate is given by:

$$e^{bit-flip} = pe_1 + (1-p)e_2 \tag{6.6}$$

where e_1 and e_2 are the QBERs when Alice and Bob both employ the Hadamard and identity operations, respectively.

The simplest classical error correction code is the three-bit code in which each bit is duplicated three times: $0 \rightarrow 000$ and $1 \rightarrow 111$. However, this method is not effective in a quantum channel because the no-cloning theorem prohibits the duplication of a single qubit. To overcome this limitation, a different method can be used, such as the use of

a GHZ state consisting of three maximally entangled qubits ($|P\rangle = \frac{1}{2}(|000\rangle + |111\rangle)$). Instead of copying the qubits, they are created and entangled in such a way that they hold the same value when measured. The GHZ state code can detect and correct a single error. For generalization, a state of Nq maximally entangled qubits can be called a Cat state, $|Cat\rangle$.

$$|Cat\rangle = \frac{1}{\sqrt{2}}(\prod_{j=1}^{Nq} |b_j\rangle \pm \prod_{j=1}^{Nq} |\bar{b_j}\rangle), \tag{6.7}$$

where $b \subset \{0, 1\}$ and the bar above b indicates its logical negation. Nq represents the number of qubits in the state. Alice and Bob can replace their private states $|\phi^+\rangle$ 12 and $|\phi^+\rangle$ 34 respectively with a Cat state $|Cat\rangle$. They can decide the number of repetitive qubits, Nq, to use for error correction.

Alice and Bob perform BSM between qubit A and one of her/his cat state qubits $|b_j\rangle$, resulting in a new entangled pair, $|SW\rangle$ A and $|SW\rangle$ B respectively. This BSM projects all the remaining qubits of Alice and Bob, which are $(2 \times (Nq - 1))$, into an entangled state, denoted as $|\omega\rangle$,

$$|\omega\rangle = \frac{1}{\sqrt{2}}(\prod_{j=1}^{2Nq-2} |b_j\rangle \pm \prod_{j=1}^{2Nq-2} |\bar{b_j}\rangle). \tag{6.8}$$

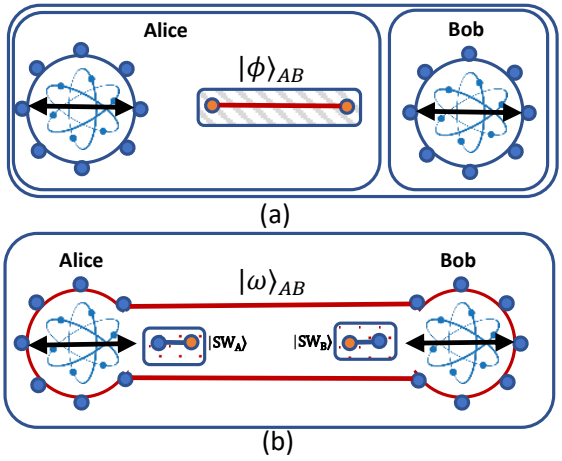


Figure 6.2: QKeyShield protocol. (a) Initial state where Alice and Bob prepare a *cat* state consisting of N_q qubits. Alice also prepares another entangled state ($|\phi\rangle$ AB) and sends qubit B to Bob. (b) Alice and Bob perform a BSM on $|SWA\rangle$ and $|SW_B\rangle$ which results in the creation of a larger *cat* state (qubits linked in red).

Alice(Bob) measures the state $|SW_A\rangle$ ($|SW_B\rangle$) to determine if a bit-flip operation on the remaining qubits ($|Cat\rangle^{Nq-1}$) is required,

$$|Cat\rangle = \begin{cases} |\bar{Cat}\rangle^{Nq-1}, & if \quad |SW\rangle_{A(B)} = |\psi\pm\rangle \\ \\ |Cat\rangle^{Nq-1}, & otherwise. \end{cases} \tag{6.9}$$

Assuming that $Nq - 1 = 3$, Alice (Bob) measures three qubits and maps the resulting code words into a single bit, with $000, 001, 010, 100 \rightarrow 0$ and $011, 101, 110, 111 \rightarrow 1$. This mapping can decrease the probability of a mismatch between Alice and Bob.

6.1.2 Time-reversed QKeyShield

The time-reversal scenario where the single-qubit measurement is conducted before the swapping operation (BSM) has first appeared in [21] and its security has been proven in [75]. Interestingly, QKeyShiled can also be implemented in a time-reversal fashion, see Figure 6.3. This is because BSM operations commute with Alice's and Bob's single-qubit measurements. As a result, the measurements might be reversed in sequence. That is, Alice and Bob do not need to wait for the BSM results to measure half of their Bell states, but they can measure them beforehand. This converts the original QKeyShiled protocol into an analogous prepare-and-measure technique, in which the unmeasured qubits 1 (3) of Alice(Bob) can be considered as BB84 states. Similar to step (6) of the QKeyShield protocol, Alice (Bob) performs BSM between qubit A (B) and her(his) qubit 1 (3). It's worth noting that BSM provides no information about the individual bit values that are obtained when measuring qubits $2(4)$; however, it helps in preparing the desired shared classical bit, detecting Eve's intervention, and testing Bob's device credibility. Similar to step (7), Alice (Bob) performs a **classical** bit-flip to her(his) obtained bit only if her(his) BSM result is $|\psi^{\pm}\rangle_{1A}(|\psi^{\pm}\rangle_{B3})$. Most significantly, Alice and Bob follow the same error

estimation strategy used in the original QKeyShield.

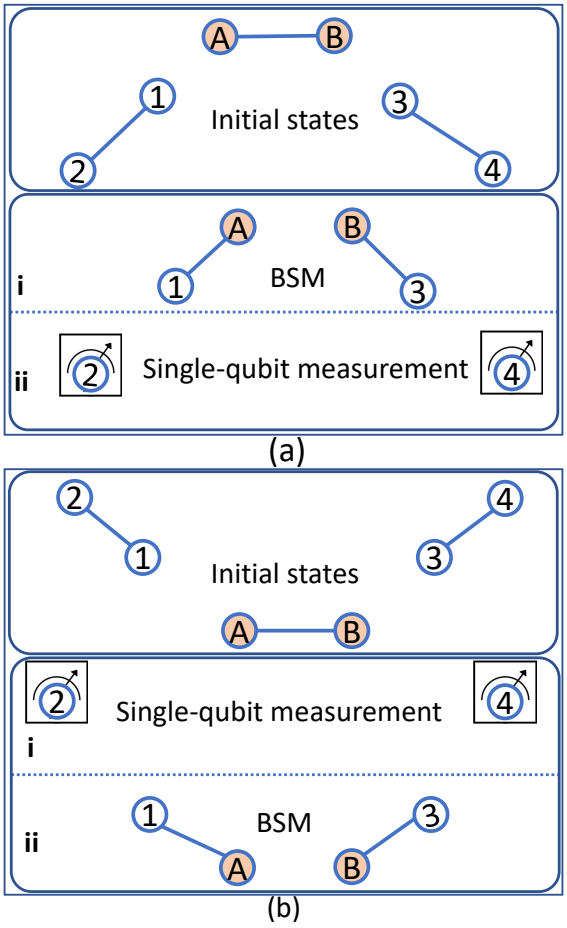


Figure 6.3: (a) QKeyShield (b) Time-reversed QKeyShield.

Using the time-reversed mode, the information is obtained beforehand, which provides another layer of security. It helps in securing the protocol against detector blinding attacks. This version and the original version are almost similar; therefore, throughout the rest of this thesis, I only mention QKeyShield.

6.1.3 Key rate

QKeyShield protocol aims to minimize the number of discarded results during key generation. The percentage of used qubits in key generation, denoted as η, is given by

$$\eta = \frac{q_k}{q_{total}}, \tag{6.10}$$

where q_k is the number of used qubits and q_{total} is the total number of transmitted qubits. q_{total} is used interchangeably with N, where N represents the total number of the protocol rounds. QKeyShield minimizes the number of discarded results as it adopts the use of a biased selection approach along with adequate error analysis; therefore, the probability of a qubit to used in the key generation is high. The number of discarded measurement results (bits) can be given by

$$\eta_{dis} = 2 \times p \times (1-p) \times N, \tag{6.11}$$

where p represents the probability of performing the Hadamard operation and is bounded by $0 < p \leq 0.5$; and $(1-p)$ represents the probability of performing the identity operator. After discarding the mismatched measurements, we divide the remaining results between error estimation and the sifted key. As discussed earlier, two test sets, n_1 and n_2, of length $(1-p) \times N$ are taken from both measurements. Thus, the total number of bits used for error estimations read

$$\eta_{ES} = 2 \times (1-p) \times N. \tag{6.12}$$

The remaining measurements results are the sifted key and can be given by

$$\eta_{sif} = (p^2 - (1-p)^2) \times N. \tag{6.13}$$

After the error estimation step, Alice and Bob are left with M measurement results, $M = N - n_1 - n_2$. They perform local error correction, discussed in Section 6.1.1, to form their raw key, denoted K_A^M and K_B^M, respectively. Then Bob performs a one-way error correction over the public channel to compare his key to Alice's key. This error correction procedure reveals b_{leak} bits of information. The maximum tolerable probability of $K_A^M \neq K_B^M$ after error correction is denoted as ε_{EC}. To find out if the final raw keys are corrected, Alice computes a hash h_A of length $\lceil \log(\frac{1}{\varepsilon_{EC}}) \rceil$ from her raw key K_A^M. She sends the hash

function and h_A to Bob over a public channel. Then Bob computes h_B. If $h_A \neq h_B$, the protocol aborts. The total leaked information during the error correction is given by $b_{leak} + \lceil \log(\frac{1}{\varepsilon_{EC}}) \rceil \leq b_{leak} + \log(\frac{2}{\varepsilon_{EC}\varepsilon_{secure}^2})$.

After the error correction step, if it is successful, Alice and Bob perform privacy amplification to distill a shorter secret key. Privacy amplification is a technique used to extract a secret key of a smaller length from a larger key while ensuring that the key is secure against eavesdropping.

One way to perform privacy amplification is by using a two-universal hash function. A two-universal hash function is a type of hash function that is used to map an input key of arbitrary length to a shorter key of fixed length. This process ensures that the shorter key is highly dependent on the original key and thus provides a high level of security against eavesdropping. The two-universal property of this function means that for any two distinct inputs, the probability of them being mapped to the same output is very small. This property is important for privacy amplification because it ensures that even if an eavesdropper intercepts some of the keys, the chances of guessing the correct key are negligible.

In the QKeyShield protocol, Alice and Bob apply the same two-universal hash function to their error-corrected keys K_A^M and K_B^M of length n to obtain shorter keys S_A and S_B of length L. This process ensures that the shorter keys are highly dependent on the original keys and thus provide a high level of security against eavesdropping. The maximum tolerable probability of K_A^M = K_B^M after privacy amplification is denoted as ε_{PA}. The secret key length L satisfies

$$L \leq H_{min}^{\varepsilon}(K_A^M|E) - b_{leak} - \log_2(\frac{2}{\varepsilon_{EC}(\varepsilon_{secure})^2}), \tag{6.14}$$

where $H_{min}^{\varepsilon}(K_A^M|E)$ is the conditional smooth min-entropy of Alice's key and Eve's knowledge. Computing $H_{min}^{\varepsilon}(K_A^M|E)$ is a challenge as Eve information is not accessible to Alice and Bob; therefore, the correlation between K_A^M and K_B^M can be used to

bound the correlation between Alice and Eve [146], which gives

$$H^{\varepsilon}_{min}(K^M_A|E) \geq D_q n - H^{\varepsilon}_{max}(K^M_A|K^M_B), \tag{6.15}$$

where $H^{\varepsilon}_{max}(K^M_A|K^M_B)$ corresponds to the amount of information that Bob needs to reconstruct Alices' key K^M_A with ε error probability ($H^{\varepsilon}_{max}(K^M_A|K^M_B) = nH(e_{max})$); D_q corresponds to the preparation device quality and we assume that the source is ideal, that is, $D_q = 1$; e_{max} is a specified maximum allowable error rate; and $H(x)$ is the binary Shannon entropy, $H(x) = -xlog_2(x) - (1-x)log_2(1-x)$. The secret key length reads:

$$L \leq n(1 - H(e_{max})) - b_{leak} - \log_2(\frac{2}{\varepsilon_{EC}(\varepsilon_{secure})^2}). \tag{6.16}$$

According to the security definitions, If QKeyShield always aborts, it is still secure. Consequently, *completeness* is another crucial aspect that should be taken into account. It represents the protocol probability of not aborting, $1 - \varepsilon_{abort}$ for small ε_{abort}. Due to the biased approach, QKeyShield allows for significantly higher sifting efficiency. The sifted key is extracted from the measurements of the dominant operator, let's say the identity operator, and the measurements of the non-dominant operator are used for error estimation. Finlay, the secret key rate for finite N reads

$$\delta_{fin} = \omega(1 - \varepsilon_{abort})\frac{L}{q_{total}(\eta_{sif}, \eta_{ES})}, \tag{6.17}$$

where $q_{total}(\eta_{sif}, \eta_{ES}) = \eta_{sif} + \eta_{ES} + 2\sqrt{\eta_{sif}\eta_{ES}}$ represents the total number of required qubits that should be sent until η_{sif} sifted key bits and η_{ES} error estimation bits are collected; and ω is the experiments repetition rate, i.e. the inverse of the time required for a single experiment. When q_{total} is sufficiently large, the sifted key tend to infinity $\eta_{sif} \rightarrow \infty$, thus $L/q_{total}(\eta_{sif}, \eta_{ES}) \rightarrow 1$. For arbitrary security bound, $\varepsilon > 0$, the

formula of the asymptotic secret key rate can be given by

$$\delta = 1 - H(e_1) - H(e_2), \tag{6.18}$$

where $H(.)$ is the binary Shannon entropy; and e_1 and e_2 are the QBERs when Alice and Bob both employ the Hadamard and identity operation, respectively. This equation represents the upper bound on the secret key rate, which is only possible with ideal implantation.

6.1.4 Finite-key security proof

We remark that the security definitions listed in Section 4.2.4.1 are *composable*. That is to say, the security of the resulting combination can be deduced from the security of the individual components' proofs.

Lemma 6.1.1 (Security of QKeyShield)**.** QKeyShield protocol is ε_{total} secure , with $\varepsilon_{total} \geq \varepsilon_{EC} + \varepsilon_{secure} + \varepsilon_{PA}$

Proof. To begin, we show that QKeyShield is ε_{EC}. Alice and Bob obtained the hashes $h_A \neq h_B$ of length $\lceil \log_2(\frac{1}{\varepsilon_{EC}}) \rceil$ by performing a two-universal hash function on their raw keys. The probability of two hashes of length $\lceil \log_2(\frac{1}{\varepsilon_{EC}}) \rceil$ of two different inputs to coincide is small, $2^{-\lceil \log_2(\frac{1}{\varepsilon_{EC}} \rceil}$.

$$Pr[h_A = h_B, K_A^M \neq K_B^M] \leq Pr[h_A = h_B | K_A^M \neq K_B^M] \leq 2^{-\lceil \log_2(\frac{1}{\varepsilon_{EC}} \rceil} \leq \varepsilon_{EC}. \tag{6.19}$$

It is observed that when the protocol aborts, S_A and S_B always coincide, thus $Pr[S_A \neq S_B, h_A \neq h_B] = 0$. By employing this in Equation 6.19, we demonstrate the protocol is ε_{EC}:

$$Pr[S_A \neq S_B] \leq Pr[S_A \neq S_B, h_A = h_B] \leq Pr[K_A^M \neq K_B^M, h_A = h_B] \leq \varepsilon_{EC}. \tag{6.20}$$

To demonstrate the protocol's secrecy, the Quantum Leftover hashing lemma [123, 145] is used to give us the upper bound that follows,

$$\frac{1}{2}\left\|\rho_{AB}-\tau_K\otimes\rho_E\right\|_1 \leq \varepsilon_{secret}+\frac{1}{2}\sqrt{2^{L-H^{\varepsilon}_{min}(K_A^M|E_CE)}} \tag{6.21}$$

where L is the length of S_A after the privacy amplification and E represents the total information that Eve learned about K_A^M. This comprises her quantum system E_Q, information gained during Alice and Bob classical communication E_C, and the knowledge about the used hash function E_F, $E = E_qE_CE_F$. By employing the min entropy chain-rule [145] :

$$H^{\varepsilon}_{min}(K_A^M|E_CE) \quad \geq H^{\varepsilon}_{min}(K_A^M|E) - log_2|E_C| \tag{6.22}$$

where $log_2|E_C|$ represents Eve's gained knowledge during the error correction and is given by $log_2|E_C| = b_{leak} - \log_2(\frac{2}{\varepsilon_{EC}\varepsilon^2_{secure}})$. By substituting this in Equation 6.53,

$$H^{\varepsilon}_{min}(K_A^M|E_CE) \quad \geq \quad H^{\varepsilon}_{min}(K_A^M|E) \quad - \quad b_{leak} \quad - \quad \log_2(\frac{2}{\varepsilon_{EC}\varepsilon^2_{secure}}) \tag{6.23}$$

By inserting Equation 6.54 into 6.51 we obtain:

$$\frac{1}{2}\left\|\rho_{AB}-\tau_K\otimes\rho_E\right\|_1 \leq \varepsilon_{secret}+\frac{1}{2}\sqrt{2^{L-(H^{\varepsilon}_{min}(K_A^M|E)-b_{leak}-\log_2(\frac{2}{\varepsilon_{EC}\varepsilon^2_{secure}}))}} \tag{6.24}$$

$$\begin{aligned}\frac{1}{2}\left\|\rho_{AB}-\tau_K\otimes\rho_E\right\|_1 &\leq \varepsilon_{secret}+\frac{1}{2}\sqrt{2^{log_2(2\varepsilon_{PA})^2}} \\ &= \varepsilon_{secret}+\varepsilon_{PA}.\end{aligned} \tag{6.25}$$

The secrecy of QKeyShield has been demonstrated through rigorous proof in the preceding analysis. By combining it with an ε_{EC} proof, it has been shown that QKeyShield offers ε_{total} security, where $\varepsilon_{total} \geq \varepsilon_{EC} + \varepsilon_{secure} + \varepsilon_{PA}$. This suggests that the overall security

of QKeyShield is at least as strong as the total of its individual security measures. □

6.1.5 Eavesdropping strategies

In this section, I have delved deeply into the various attacks that are made possible by the principles of quantum mechanics, gaining a comprehensive understanding of their capabilities and potential implications.

6.1.5.1 Entangle-measure attack

Eve intercepts the transmitted qubit sent to Bob and entangles it with her ancilla state (prepared, say, in the state $|E\rangle$) by performing unitary operators. Eve then transmits the traveling qubit traveling to Bob. Finally, Eve measures the auxiliary qubit in her possession to learn more about the shared key. as Alice randomly chooses to perform a Hadamard operation on her qubit A before or after sending qubit B to Bob, the intercepted state by Eve is either $|\phi^+\rangle_{AB}$ or $|W^+\rangle_{AB}$. Eve's goal is to intercept the qubits sent from Alice to Bob and entangle them with her ancilla state, for example, in the state $|E\rangle$. Eve has no information about the intercepted state, so she uses the same unitary operator and the same ancilla state. Eve's unitary operation U_e performed on the composite system of the shared state can be written as:

$$U_{\mathrm{e}} |\phi^+\rangle |E\rangle = |\phi^+\rangle |E_{\phi^+}\rangle , \tag{6.26}$$

$$U_{\mathrm{e}} |W^+\rangle |E\rangle = |W^+\rangle |E_{W^+}\rangle , \tag{6.27}$$

where the states $|E_{\phi^+}\rangle$ and $|E_{W^+}\rangle$ are simply the states that Eve holds after the unitary transformation depending on the initial state that Alice has sent. We know that the initial states $|\phi^+\rangle$ and $|W^+\rangle$ are not orthogonal to each other; therefore, we can now compare the scalar product of the states on the right-hand side with the states on the left-hand side of

Equations 6.26 and 6.27.

$$\langle \phi^+ | W^+ \rangle \langle E | E \rangle = \langle \phi^+ | W^+ \rangle \langle E_{\phi^+} | E_{W^+} \rangle \tag{6.28}$$

The scalar product of Eve's initial states $\langle E|E\rangle$ is equal to one. If Eve does not disturb Alice's state, i.e $\langle \phi^+|W^+\rangle$ has not changed on the right-hand side, the scalar product of $\langle E_{\phi^+}|E_{W^+}\rangle$ has to be one as well, which directly implies that the two states have to be the same. That means, no matter which state Alice prepares, Eve always gets the same state. On the other hand, if Eve disturbs Alice's state, she can gain some information about Alice's state.

$$\langle \phi^+ | W^+ \rangle \langle E | E \rangle = \langle \acute{\phi}^+ | \acute{W}^+ \rangle \langle E_{\phi^+} | E_{W^+} \rangle \tag{6.29}$$

The more distinguishable Eve's states $|E_{\phi^+}\rangle$ and $|E_{W^+}\rangle$, the more disturbed the states $|\acute{\phi}^+\rangle$ and $|\acute{W}^+\rangle$. The scalar product $\langle \acute{\phi}^+ | \acute{W}^+ \rangle$ increases as $\langle E_{\phi^+}|E_{W^+}\rangle$ decreases. This implies that the more disturbance Eve introduces to the system, the more information she gains. Above, QKeyShield's resistance to the entangle-measure attack has been demonstrated.

6.1.5.2 Collective attack

The fundamental idea behind a collective attack is that the adversary, Eve, attempts to find a multi-qubit state that preserves the correlation between the two legitimate parties (Alice and Bob). She also inserts new qubits in order to differentiate between Alice's and Bob's measurement results. If Eve is successful in finding such a state, she can carry out her interference undetected and gain access to the shared key. As illustrated in Figure 6.4, Eve may prepare a multi-qubit complex state and attempt to launch a collective attack.
Before discussing Eve's attack, let us summarize the basic principles that Alice and Bob will follow in QKeyShield:

- First, Alice(Bob) performs Hadamard/identity operation on the share-state $|\phi^{\mp}\rangle_{AB}$ qubits based on a biased probability.

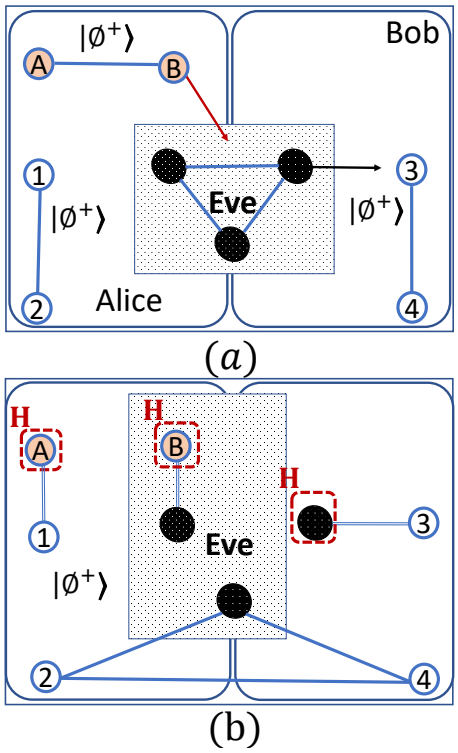


Figure 6.4: Illustration of Eve's attempt to carry out a collective attack strategy.

Table 6.2: Possible scenarios when an eavesdropper applies random operators, given that the eavesdropper's initial state is $|P^{\pm}\rangle_{DEF}$.

Alice	Eve	Bob	Shared state $\|\psi\rangle_{2E4}$	Inferred key	Alice & Bob use the same operators?	Results Correctness	Eve detected?
$\mathbb{I}$	$\mathbb{I}\mathbb{I}$	$\mathbb{I}$	$\|P^{\pm}\rangle$	000 or 111	Yes	*Correct*	No
$\mathbb{I}$	$\mathbb{I}\mathbb{I}$	H	$\|P^{\pm}\rangle$, $\|Q^{\pm}\rangle$	Random	No	Incorrect	Discarded
$\mathbb{I}$	$\mathbb{I}$H	$\mathbb{I}$	$\|P^{\pm}\rangle$, $\|Q^{\pm}\rangle$	Random	Yes	**Incorrect**	**Yes**
$\mathbb{I}$	$\mathbb{I}$H	H	$\|P^{\pm}\rangle$	000 or 111	No	Correct	Discarded
$\mathbb{I}$	H $\mathbb{I}$	$\mathbb{I}$	$\|P^{\pm}\rangle$, $\|S^{\pm}\rangle$	Random	Yes	**Incorrect**	**Yes**
$\mathbb{I}$	H $\mathbb{I}$	H	$\|P^{\pm}\rangle$, $\|Q^{\pm}\rangle$, $\|R^{\pm}\rangle$, or $\|S^{\pm}\rangle$	Random	No	Incorrect	Discarded
$\mathbb{I}$	H H	$\mathbb{I}$	$\|P^{\pm}\rangle$, $\|Q^{\pm}\rangle$, $\|R^{\pm}\rangle$, or $\|S^{\pm}\rangle$	Random	Yes	**Incorrect**	**Yes**
$\mathbb{I}$	H H	H	$\|P^{\pm}\rangle$, or $\|S^{\pm}\rangle$	Random	No	Incorrect	Discarded
H	$\mathbb{I}\mathbb{I}$	$\mathbb{I}$	$\|P^{\pm}\rangle$, or $\|S^{\pm}\rangle$	Random	No	Incorrect	Discarded
H	$\mathbb{I}\mathbb{I}$	H	$\|P^{\pm}\rangle$, $\|Q^{\pm}\rangle$, $\|R^{\pm}\rangle$, or $\|S^{\pm}\rangle$	Random	Yes	**Incorrect**	Yes
H	$\mathbb{I}$H	$\mathbb{I}$	$\|P^{\pm}\rangle$, $\|Q^{\pm}\rangle$, $\|R^{\pm}\rangle$, or $\|S^{\pm}\rangle$	Random	NO	Incorrect	Discarded
H	$\mathbb{I}$H	H	$\|P^{\pm}\rangle$ or $\|S^{\pm}\rangle$	Random	Yes	**Incorrect**	Yes
H	H $\mathbb{I}$	$\mathbb{I}$	$\|P^{\pm}\rangle$	000 or 111	No	Correct	Discarded
H	H $\mathbb{I}$	H	$\|P^{\pm}\rangle$ or $\|Q^{\pm}\rangle$	Random	Yes	**Incorrect**	Yes
H	H H	$\mathbb{I}$	$\|P^{\pm}\rangle$ or $\|Q^{\pm}\rangle$	Random	No	Incorrect	Discarded
H	H H	H	$\|P^{\pm}\rangle$	000 or 111	Yes	*Correct*	No

- Next, Alice(Bob) measures the state $|\psi\rangle_{1A}(|\psi\rangle_{3B})$ and performs a quantum bit-flip on the unmeasured quibit $2(4)$ if the measurement results are $|\psi^{\pm}\rangle$.

- The goal for Alice and Bob is to end up sharing the maximally entangled state

$|\phi^{\pm}\rangle_{24}$.

Eve's attack aims to make the system end up in a state $|P^{\pm}\rangle_{2E4} = \frac{1}{\sqrt{2}}(|000\rangle \pm |111\rangle)$, where 2 is Alice's qubit, 4 is Bob's qubit, and E is Eve's qubit. A MATLAB-based simulator of QKeyShield has been developed to investigate Eve's possible collective attack scenarios. In the first scenario, as illustrated in Figure 6.4, Eve intercepts the qubit sent to Bob, prepares a GHZ state ($|P^{\pm}\rangle$), and performs a Hadamard/identity operation and a Bell operator measurement on the intercepted qubit. She then conducts a quantum bit-flip on the unmeasured qubits and sends one of them to Bob after applying a Hadamard/identity operation on it. As a result, the shared state between Alice and Bob is disrupted. Eve's attempt to gain information disrupts the shared state of the legitimate parties.

This attack can also be performed on the time-reversed QKeyShield with minor modifications. The presence of Eve will impact the results of Alice's and Bob's BSM, which in turn affects the final shared bits due to the bit-flip operations.

In an effort to find the optimal multi-qubit state that preserves the correlation between Alice and Bob, Eve can choose to use any of the GHZ states (such as $|P^{\pm}\rangle$, $|Q^{\pm}\rangle$, $|R^{\pm}\rangle$, and $|S^{\pm}\rangle$) as her initial state. However, regardless of the initial state chosen, the results of the legitimate parties (Alice and Bob) will always be disrupted.

Table 6.2 shows all possible scenarios when an eavesdropper applies random operators, given that the eavesdropper's initial state is $|P^{\pm}\rangle_{DEF}$. Alice applies either H or I operator on her qubit A with probabilities p and $(1-p)$, respectively, before sending qubit B to Bob. Eve intercepts qubit B and applies either H or I operator on it with probabilities p_1 and p_2, respectively, before sending qubit F to Bob. Bob then applies either H or I operator on the received qubit F with probabilities p and $(1-p)$, respectively. As can be seen, half of the results are discarded due to Alice and Bob using different operators, while the other half is where they use the same unitary operator. However, in the latter case, the presence of Eve affects 75% of the results. Although these affected results can be used for detecting Eve, only a small portion of them are used for key generation. Figure 6.5 illustrates the

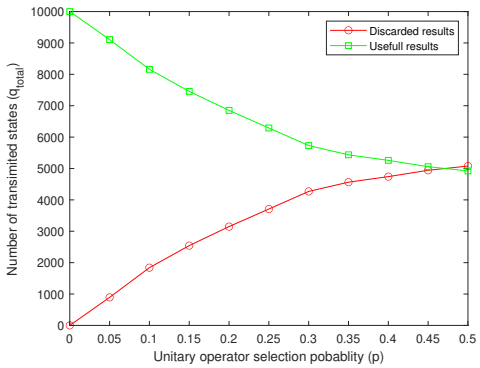


Figure 6.5: Comparison of discarded results to useful ones when $q_{total} = 10000$ and the selection probability varies from $0 \leq p \leq \frac{1}{2}$. Fair random-based protocols like RDI-QKD [76] discard a minimum of 50% of the results.

theoretical number of useful and discarded results when varying the selection probability for a biased approach. Using a fair operator selection probability of $p = 0.5$ results in 50% of the results being discarded. On the other hand, a completely biased protocol with $p = 0$ will not discard any results, but Eve cannot be detected. By using a biased operator selection probability, the number of discarded results is minimized. To detect Eve while using a biased probability, a certain number of test samples must be chosen from subsets where both parties perform the Hadamard operator or identity operator. These test samples, n_1 and n_2, should be at least of the order of $\Omega(log_2 q_{total})$. The use of bias operator selection probability minimizes the number of results that will be discarded. To detect Eve while using biased probability, we need n_1 and n_2 test samples chosen from the subsets where they both perform the Hadamard operator or identity operator, respectively. The number of test samples n_1 and n_2 should be at least of order $\Omega(log_2 q_{total})$.

6.1.5.3 Coherent attack

The Coherent attack is a type of eavesdropping strategy in which an attacker, commonly referred to as Eve, creates a global auxiliary system that interacts with all qubits transmitted through the channel [152]. This interaction is achieved through the application of a global unitary operator, which allows Eve to manipulate the transmitted qubits in a coher-

ent manner. The idea behind this attack is that Eve can gain access to more information by manipulating the entire system as a whole, rather than targeting individual qubits. This approach is particularly effective when the transmitted qubits are highly correlated, as Eve can exploit these correlations to gain more information about the transmitted state.

In contrast, QKeyShield is a protocol that is specifically designed to be resistant to coherent attacks. The protocol utilizes a series of independent qubits that are transmitted through the channel, with each qubit being completely independent of the others. Furthermore, all of the measurements in QKeyShield are performed on each round in a completely independent manner. This is because no classical messages are required between Alice and Bob during the protocol, which makes it much more difficult for Eve to gather information about the transmitted state.

Another important feature of QKeyShield that makes it resistant to coherent attack is the fact that it is an entanglement-swapping-based protocol. This means that Eve is not able to interact directly with the information-carrying qubits, as they are entangled with other qubits in the system. As a result, Eve would not be able to gain any additional information about the transmitted state by performing a coherent attack.

In summary, while the Coherent attack is a powerful strategy for an attacker to gain information about transmitted qubits, it is not effective against QKeyShield. The protocol's use of independent qubits, independent measurements and entanglement-swapping all work together to make it resistant to this type of attack, making it more secure than other protocols that might be vulnerable to this type of attack.

6.1.5.4 Detector blinding attack

Entanglement-swapping has been utilized in the past for building a side-channel-free QKD [27, 75]. The entanglement-swapping dual teleportation channel serves as an ideal Hilbert space filter. QKeyShield is an entanglement-swapping-based protocol that allows Alice's and Bob's qubits that do not interact directly to become entangled. Unlike BB84 [18]

and BBM92 [19], the qubit sent by Alice to Bob is not an information carrier; however, it is used to help Alice and Bob to establish secret entangled qubits, that are the information carrier. Bob's device performs three main operations before measuring the secret qubit. It starts by randomly performing a Hadamad operation on the received qubit. Then it conducts BSM. After that, a bit-flip operation is performed on the secret qubit if required. Finally, the secret qubit is measured. A blinding attack will affect the BSM, which will increase the error rate when measuring the secret qubit. Moreover, the time-reversed QKeyShield is more robust against detector blinding attacks as the information is extracted in advance and Eve's input cannot manipulate them.

6.1.6 Protocol Efficiency

QKeyShield is an efficient QKD protocol, as it effectively balances multiple efficiency metrics including communication efficiency, resource efficiency, key-rate efficiency, and sifting-time efficiency. The analysis includes computation of communication efficiency, resource efficiency, and sifting efficiency for each round of the protocol (steps 1-8).

6.1.6.1 Communication efficiency (σ)

To calculate the communication efficiency of the protocol, I utilize the definition proposed by Cabello in [35], given by the following equation:

$$\sigma = \frac{b_s}{q_t + \acute{b_t}}, \tag{6.30}$$

where b_s represents the expected number of secret bits obtained by Alice(Bob), q_t is the number of qubits exchanged, and $\acute{b_t}$ represents the number of classical bits sent. To be realistic, I consider it to be the number of classical messages sent.

The sent classical messages for the purpose of sifting are not included in this communication efficiency (only the messages that are used in each round for inferring the key).

Table 6.3: Communication efficiency σ of different entanglement-swapping-based QKD protocols along with the most recent related protocols.

Protocol	b_s	q_t	$b_t^{'}$	**Communication efficiency (σ)**
QKeyShield	1	1	0	1
[37]	2	2	1	0.67
[50]	4	5	3	0.5
[59]	2	2	1	0.67
[162]	1	2	1	0.33
[131]	2	2	4	0.33
[68]	2/1	2	2	0.312
[118]	1	2	1	0.33
[51]	1	2	0	0.5
[93]	1	2	1	0.33
[7]	1	1	1	0.5
[117]	1	1	1	0.5
[41]	1	3	1	0.25
[76]	0.5	1	0	0.5

Communication efficiency σ comparison was introduced in [35] where several protocols were compared, such as Cabello, Ekert, Bennett (BB84), etc. The proposed QKeyShield has outperformed Cabello's protocol because every transmission of a single qubit can generate one classical bit of the shared key with local measurements and without the use of classical communication. Thus, the QKeyShield protocol is 100% efficient in terms of σ. The high discard rate of [76] allows for a maximum of 0.5 percent of acquiring a classical bit, similar to the case of the B92 protocol reported in [35].

6.1.6.2 Resources efficiency (Υ)

Some QKD protocols claim 100% efficiency using the communication efficiency metric (σ), but neglect to consider the large quantum memory or multiple relay nodes required. To accurately evaluate the resource efficiency of each protocol in obtaining a single classical bit, I have included multiple criteria, such as the number of required Bell states ($C1$), time delay in each round ($C2$), number of performed BSMs ($C3$), number of quantum memory cells required ($C4$), number of used relay nodes ($C5$), and number of detectors

in measurements ($C6$).

The proposed resource efficiency evaluation is comprehensive and general, taking into account different criteria. However, it does not consider all aspects of a QKD protocol, such as operations like Hadamard and bit-flip. To determine the resource efficiency of each protocol, a multicriteria decision problem is solved using the Analytic Hierarchy Process (AHP) method. A numerical scale from 1 to 8 is used for pairwise comparison to establish preferences between criteria, where 8 indicates one criterion is highly more significant than the other and 1 shows both criteria are equally significant. The significance of one criterion in relation to another is given by the reciprocal,

$$C_{ij} = \frac{1}{C_{ji}}, \tag{6.31}$$

where C_{ij} represents the pairwise comparison between the criterion C_i and C_j. The value 1/8 suggests that one criterion is extremely less significant than the other. The weight of each criterion is given by

$$W_i = \sum_{i=1}^{5} C_i, \tag{6.32}$$

while the normalized weight is given by

$$\bar{W}_i = \frac{W_i}{\sum_{i=1}^{5} W_i}. \tag{6.33}$$

This allows us to arrive at the final resource efficiency of each protocol, by taking into account all of the criteria that are included in the evaluation process.

Table (6.4) illustrates the pairwise comparison of two scenarios. In the first scenario, it is assumed that all criteria are of equal importance. However, in terms of time delay and technological challenges, some criteria may be more costly than others. For instance, protocols that cause time delays can be slower in establishing a shared key, while the use of quantum memory can make the protocol less efficient due to current technology

Table 6.4: Example of pairwise comparison: two scenarios were considered: 1) all the criteria are "equal significant" to each other, and 2) some criteria are "more significant" than others ($C4 > C2 > C5 > C1, C3, C6$).

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	(W)	$(\bar{W})$
Scenario(1)							W^{eq}	$\bar{W^{eq}}$
$C1$	1	1	1	1	1	1	6	0.16
$C2$	1	1	1	1	1	1	6	0.16
$C3$	1	1	1	1	1	1	6	0.16
$C4$	1	1	1	1	1	1	6	0.16
$C5$	1	1	1	1	1	1	6	0.16
$C6$	1	1	1	1	1	1	6	0.16
Total							36	
Scenario(2)							W^{var}	$\bar{W^{var}}$
$C1$	1	0.25	1	0.125	0.5	1	2.88	0.06
$C2$	4	1	4	0.5	2	4	11.5	0.23
$C3$	1	0.25	1	0.125	0.5	1	2.88	0.06
$C4$	8	2	8	1	2	8	21	0.47
$C5$	2	0.5	2	0.5	1	1	6	0.1
$C6$	1	0.25	1	0.125	1	1	6	0.066
Total							65.38	

Table 6.5: Resources efficiency of different entanglement-swapping-based QKD protocols per each obtained classical bit.

Protocol	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	Υ **:** $\bar{W} = \bar{W^{eq}}$	Υ **:** $\bar{W} = \bar{W^{var}}$
QKeyShield	6	0	3	0	0	2	0.93	**0.97**
[37]	3	1	1.5	1	0	2	0.94	0.93
[50]	3	1	2	0	0.75	4	0.92	0.95
[59]	4	1	2	4	0	4	0.89	0.85
[131]	2	1	1	1	0	4	0.92	0.90
[68]	4	1	2	0	0	4	0.92	0.95
[118]	4	1	1	2	1	4	0.90	0.89
[162]	4	1	1	0	1	4	0.92	0.95
[117]	1	1	1	2	0	4	0.93	0.91
[7]	1	1	2	2	0	4	0.92	0.90
[154]	2	2	1	2	1	4	0.91	0.89
[41]	3	3	1	0	1	4	0.91	0.92
[76]	0.5	3	0	0	0	4	**0.95**	0.95

limitations. Similarly, the use of intermediate nodes (relay nodes) can indicate that the protocol is slower and more expensive. From these perspectives, it is not reasonable to assume that all the criteria are of equal importance.

The proposed resource efficiency evaluation is general and allows for the design of different scenarios that match specific case study requirements.In the second scenario presented in Table (6.4), the importance of criteria is weighted differently. Specifically, time delay, quantum memory, and relay nodes are deemed more significant than other criteria. Therefore, it is assumed that $C4$ is more significant than $C2$, $C2$ is more significant than $C5$, $C5$ is more significant than $C1$ and $C3$, and $C1$, $C3$, and $C6$ are equally significant to one another. The normalized weights obtained from the pairwise comparison are used to evaluate different entanglement-swapping-based QKD protocols, as seen in Table 6.5.

The protocol resource efficiency, Υ_j, is given by:

$$\Upsilon_j = 1 - \left(\frac{\sum_{i=1}^{5} C_{ki} * \bar{W}_i}{\sum_{k=1}^{a}\sum_{i=1}^{5} C_{ki} * \bar{W}_i}\right) \tag{6.34}$$

where C_{ki} represents the value of the criterion Ci of the protocol j, and a represents the total number of compared protocols. The values of C_{ki} are given by: $C_{k1} = \frac{\#qubits}{b_s}$; $C_{k2} = 0, 1$ where 0 means that Alice and Bob do not wait for each other to proceed in measuring their qubits, and 1 means that Alice (Bob) waits for Bob's (Alice's) results to proceed in measuring her(his) qubit; $C_{k3} = \frac{\#BSMs}{b_s}$; $C_{k4} = \frac{\#quantum_memory_cells}{b_s}$; $C_{k5} = \frac{\#relay_nodes}{b_s}$; and C_{k5} equal the number of used detectors.

The first scenario in Table 6.5 where all the criteria are considered of equal importance, shows that the [76] protocol is the most efficient because it is a prepare-measure protocol that uses a single qubit and classical messages to obtain a classical bit. However, when considering other factors such as time delay and technological challenges, the efficiency of a protocol can change.

The second scenario in Table 6.5 takes into account these additional factors and shows that the QKeyShield protocol is the most efficient as it eliminates the need for costly cri-

teria such as delay time, classical messages, relay nodes, detectors, and quantum memory. Other protocols that suffer from a time delay can be considered infeasible for hard real-time applications. QKeyShield's resource efficiency makes it a suitable option for practical applications.

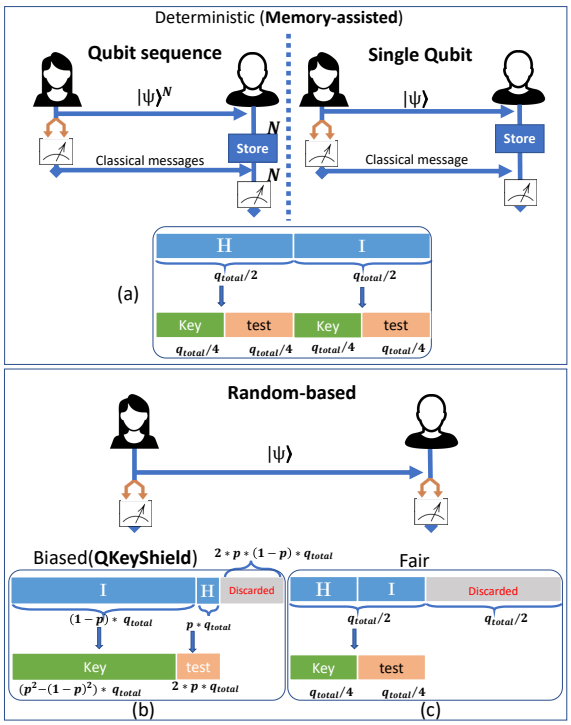


Figure 6.6: Comparison of expected sifting efficiencies between random-based and memory-assisted.

6.1.6.3 Key rate efficiency (δ_{fin})

In addition to minimizing communication overhead and increasing resource utilization, QKeyShield boasts a high key rate efficiency. It achieves this by minimizing the number of discarded results through its adoption of a biased selection approach and appropriate error analysis. This results in a higher probability of a qubit being used in key generation. QKD protocols can be divided into two main categories: random-based and memory-assisted, as shown in Figure 6.6(a-c). (a) Several protocols, such as [37, 59, 118, 131], use a deterministic approach with the aid of quantum memory and classical messages,

resulting in 50% of the obtained bits ending up in the sifted key. (b) In contrast, the QKeyShield protocol utilizes a biased approach, leading to a decrease in the proportion of bits with operator conflicts from 50% to $2p(1-p) \times q_{total}$ and an increase in the proportion of bits that end up in the sifted key to $(p^2-(1-p)^2) \times q_{total}$. (c) A fair approach, commonly used in traditional QKD protocols such as [68, 93, 162] and in recent protocols like [76], results in only 25% of the obtained bits ending up in the sifted key due to the probability of performing the Hadamard operator being $p = 0.5$.

Random-based protocols involve the legitimate parties randomly choosing the measurement basis or unitary operator. QKeyShield is a biased random-based protocol, whereas other protocols in this category may be fair. Some protocols measure in a different basis ($\boldsymbol{X}$ or Z), while others randomly choose to send decoy or normal states [148]. QKeyShield randomly chooses different operators (H or I) but with the same probability of selection.

On the other hand, memory-assisted protocols involve one party randomly choosing the action and informing the other party. In this approach, Bob does not perform the Hadamard operator randomly but instead stores the received qubits and waits for Alice to inform him if a Hadamard operation is required, similar to Cabello's protocol. With the aid of classical messages and quantum memory, memory-assisted protocols often result in a time delay and only 50

As shown in Figure 6.6, QKeyShield allows for a significantly higher sifted key length and can achieve an efficiency close to 100% when q_{total} is sufficiently large. Figure 6.7 presents a numerical comparison of the finite secret key rate for three approaches: memory-assisted protocols [7, 117], random-based protocols with fair selection ($p = 0.5$) [41], and the biased approach used by QKeyShield where p is set based on q_{total} to provide the highest efficiency for the three scenarios. QKeyShield's flexibility in choosing p improves the protocol's key rate efficiency. However, not all biased approaches perform well all the time, such as the based decoy QKD [148], which does not mention the probabilities used for choosing between signal states, decoy pulses, or vacuum pulses.

It is worth noting that only a biased memory-assisted protocol can achieve a higher key rate than QKeyShield.

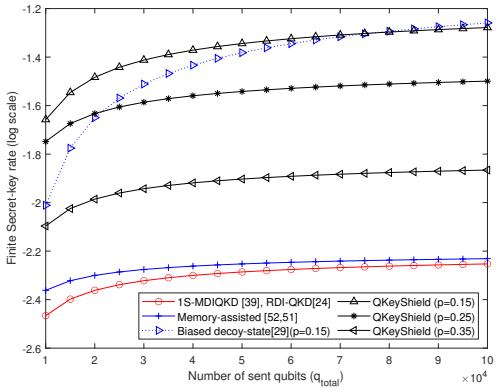


Figure 6.7: Finite Secret-key rate $\log_2(\delta_{fin})$ of the three categories shown in Figure 6.6. Parameters: $\omega = 1$, $e_{max} = 0.11$, $b_{leak} = 1.05h(e_{max})$, $\varepsilon_{EC} = 10^{(-10)}$, and $\varepsilon_{abort} = 10^{-1}$. The value of p of QKeyShield is set in such a way that $q_{total}(p^2 - \lambda) = n_1 = \Omega(logq_{total})$, where λ is a small positive number (i.e., the error due to statistical fluctuations) chosen by Alice and Bob.

6.1.6.4 Key establishment delay (T_{sif})

Many real-time applications are sensitive to delays in packet delivery, making it essential for key distribution protocols to minimize delay. The total delay for the establishment of the sifted key can be represented by the following equation:

$$T_{sif} = N(q_t \times t_{qm} + \acute{b_t} \times t_{cm}) + 2t_{cm} \tag{6.35}$$

where t_{qm} is the propagation time of the quantum messages, t_{cm} is the propagation time for the classical messages, and $2t_{cm}$ represents the delay caused by the two messages sent by Alice and Bob to inform each other of their random choices. As shown in Figure 6.8, QKeyShield is the fastest protocol as it eliminates the need for classical messages and minimizes quantum messages to the bare minimum. It is assumed that $t_{qm} = 10^{-2}$ms and $t_{cm} = 10^{-1}$ ms.

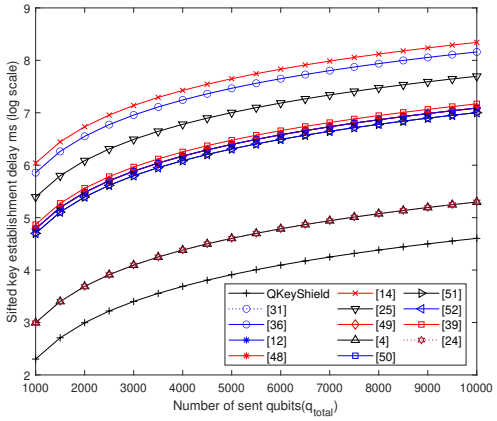


Figure 6.8: Sifted key establishment delay.

6.1.6.5 Detectors efficiencies

Most DI-QKD protocols require highly efficient detectors. DI-QKD requires the detector quality/efficiency to be $D_q > 91.1\%$ [26]. On the other hand, 1SDI-QKD allows the use of less efficient detectors, $D_q > 65.9\%$ [26]. That means we can use the current arbitrary low-quality detectors. This feature makes RDI-QKD protocols practical in situations in which Bob's measurement device is not trusted. Figure 6.14 shows the key rate along with the channel disturbance as a function of the channel loss. We assume that the channel is a fibre link. The visibility of the signal can be given by:

$$Vis = \frac{\mu 10^{-\alpha F_L/10} D_q}{\mu 10^{-\alpha F_L/10} + 2P_e} \tag{6.36}$$

where α represents the fibre attenuation $\alpha = 0.2$, P_e is the probability of an error count per clock cycle $P_e = 8.5 \times 10^{-7}$, [61], μ is the average number of photons leaving Alice's device, and D_q is Bob's detector efficiency. The fidelity can be given by:

$$Fid = \frac{1 + 3Vis}{4}. \tag{6.37}$$

The channel disturbance is $D_r = 1 - Fid$. In the event that Alice and Bob find out the

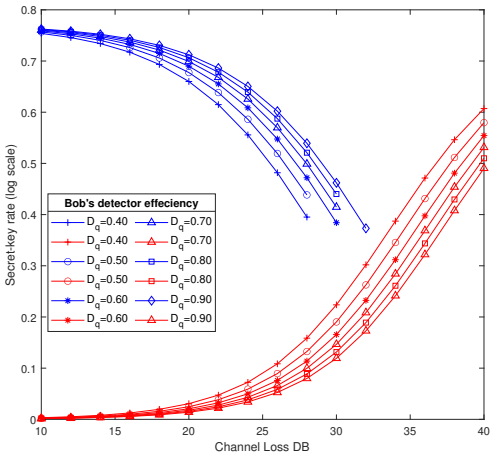


Figure 6.9: An illustration of the channel disturbance (D_r) effects on the key rate. The key rates are depicted in blue and the channel disturbance rates are depicted in red. The parameters are as inFigure 6.7 except Bob's detector efficiency is $D_q \in \{0.6, 0.7, 0.8, 0.9\}$, $q_{total} = 10^5$, and the fibre channel loss is 0.2/km.

channel fidelity is insufficient based on the results of their testing, they abort and restart the protocol. They proceed if they are confident that the fidelity is high.

6.1.7 Discussion

In this work, I considered a scenario where an organization wants to establish secret keys with its customers. The organization may invest a significant amount of money to create reliable measuring devices and place them in a secure environment, but the customers on the other end of the channel might have low-cost detectors that are placed in isolated areas. To address this challenge, I proposed an RDI-QKD protocol called QKeyShield. The entanglement source and Alice's measuring device are trusted/characterized but Bob's measuring device is not.

We found that QKeyShield is an efficient and secure RDI-QKD protocol. Despite the fact that entanglement-swapping-based protocols use extra Bell states and BSM, they allow for performing local error correction, defending against detector blinding attacks, and having two modes of the protocol, namely, QKeyShield and the time-reversed QKeyShield.

These two modes allow us to utilize the features of both entanglement-based and prepare-measure-based protocols. The time-reversal mode of QKeyShield provides another layer of security. That is, Alice does not need to wait for her BSM results and Bob's BSM results to measure half of her Bell states, but she can measure them beforehand. This converts QKeyShield into a prepare-measure protocol; however, the sent qubits are just to detect Eve's presence and to test Bob's measurement device.

We found that QKeyShield is more efficient than prior protocols in terms of communication efficiency, resource efficiency, sifting efficiency, key establishment delay, and detection efficiency. It eliminates the need for classical messages, quantum memory, and relay nodes. It minimizes the number of discarded results, the number of required detectors, the required detector quality/efficiency, the number of exchanged qubits, and the key establishment delay. QKeyShield increases the key rate, which makes it a practical QKD protocol.

Finally, the security of QKeyShield is assured by well-established quantum features such as the no-cloning theorem, non-locality, or the monogamy of entanglement. I have proved the protocol's security in general and its security against entangle-measure attacks. It is secure against the existing attacks allowed by quantum mechanics, which has been demonstrated by exploring all the attacks that are allowed by quantum mechanics. It ensures that Eve's intervention is detectable through the use of several measures. The trusted party, Alice, randomly chooses to perform a Hadamard operation on her qubit A before sending qubit B to Bob or after sending. The security of the QKeyShield protocol depends on the probability of performing the Hadamard (identity) operator. The probability of performing the Hadamard operation (p) is chosen in such a way that, $n_1, n_2 \geq \Omega(log_2 q_{total})$, where n_1 and n_2 are the numbers of test samples chosen from the subsets where they both perform the Hadamard operator or identity operator, respectively. Two error rates, e_1 and e_2, are obtained from the test samples to evaluate the untrustworthiness of Bob's devices and Eve's intervention. Due to the randomization probability, p, Eve does not know which

operator has been used by both Alice and Bob; hence, any eavesdropping attack will affect the correlation between Alice's and Bob's measurement results. In the absence of Eve, Alice's and Bob's measurement outcomes when they use the same operator should exhibit deterministic correlations. If different operators are used by Alice and Bob, their measurement results will not be correlated.

6.2 DGL22 Protocol

In this work, I propose a new QKD protocol, called DGL22, which aims to provide secure and efficient key distribution for real-world applications. The protocol is built upon the idea of using multiple measurement devices to detect and defend against eavesdropping attacks and detector blinding.

The protocol is designed for a scenario where an organization wants to establish secret keys with its customers. The organization may invest a significant amount of money to create reliable measuring devices and place them in a secure environment, but the customers on the other end of the channel might have low-cost detectors that are placed in isolated areas. The proposed protocol, DGL22, is an RDI-QKD protocol that relies on the assumption that Bob has two measurement devices, namely, BSM and SQM, and that Alice has only SQM device. Bob's BSM device inputs are coming from the quantum channel, which might be manipulated by Eve to blind the detectors. However, the other measurement device, SQM, is not connected to the quantum channel to avoid any detector blinding attacks.

The protocol also makes use of two kinds of channels, quantum and classical. Alice and Bob trust the entanglement source and the random operation choice $\{I, H\}$. In the protocol, Alice sends a series of N qubits to Bob. Figure 6.10 shows a visual representation of the proposed DGL22 protocol. The grey areas in the diagram represent the uncharacterized devices, with Bob's SQM device being isolated from the quantum channel for added

security. The red dashed line highlights the isolation of the SQM device. The trustworthiness of Bob's quantum gates (H and $\mathbb{X}$) is assumed. The protocol begins with Alice and Bob preparing two $|\phi\rangle$ states and generating a random number R. If $R < p$, where p is the probability of conducting a Hadamard operation, the Hadamard operation is performed. The protocol then proceeds with the subsequent steps as outlined. The enumerated steps that follow describe the DGL22 protocol in detail:

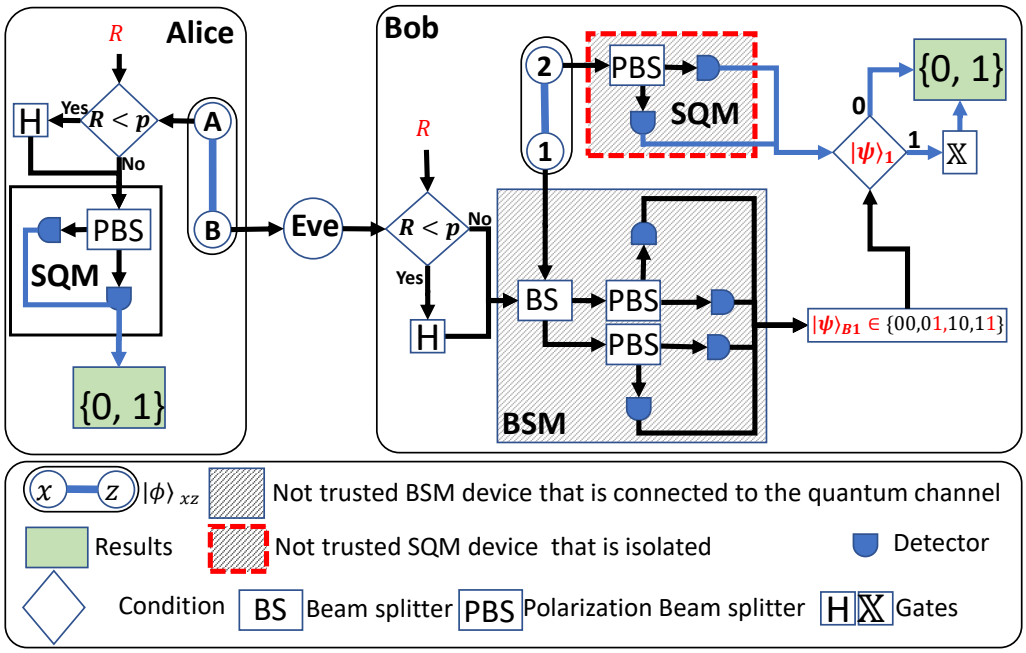


Figure 6.10: Visual representation of the proposed DGL22 protocol.

1. **Initialization:** Alice and Bob choose a value for $0 < p < \frac{1}{2}$, which is made publicly known. The probability of conducting the Hadamard operation is defined by p, while the probability of performing the identity operation is defined by $(1 - p)$. The value of p is chosen such that $N(p^2 - \delta) = n_1 = \Omega(logN)$, where δ is a small positive number (i.e., the error caused by statistical fluctuations). The number of test samples picked from the groups where they both perform the Hadamard operation or the identity operation, respectively, is n_1 and n_2.

2. **State preparation:** A Bell state is produced by Alice's (Bob's) trusted entangle-

ment source, $|\phi^+\rangle_{AB}$ ($|\phi^+\rangle_{12}$). The system containing the qubits A, B, 1, and 2:

$$\begin{aligned} |\Psi\rangle_{AB12} &= \frac{1}{2}(|00\rangle_{AB} + |11\rangle_{AB}) \otimes (|00\rangle_{12} + |11\rangle_{12}) \\ &= \frac{1}{2}(|0000\rangle_{AB12} + |0011\rangle_{AB12} + |1100\rangle_{AB12} + |1111\rangle_{AB12}). \end{aligned} \tag{6.38}$$

3. **Qubit sending:** Alice sends the qubit B of the generated Bell state $|\phi^+\rangle_{AB}$ to Bob while keeping qubit A for herself, as illustrated in Figure 6.10.

4. **Performing Hadamard operation:** Alice (Bob) chooses at random whether to perform the Hadamard or identity operation on her (his) qubit A or B with probabilities p and $(1-p)$, respectively. Once both legitimate parties have completed the Hadamard operation, the state $H^{(1)}H^{(2)}\,|\phi^+\rangle\,AB = |\phi^+\rangle\,AB$ will remain intact. If the Hadamard operation is performed by Alice first we get: $H^{(1)}\,|\phi^+\rangle\,AB = |W^+\rangle\,AB$. Once Bob performs the Hadamard operation we get, $H^{(2)}\,|W^+\rangle\,AB = |\phi^+\rangle\,AB$. It makes no difference whether this step is performed concurrently or separately; nonetheless, it secures the protocol. Alice randomly performs the Hadamard operation on her qubit A before or after sending qubit B.

5. **SQM:** Alice and Bob share one of the maximally entangled states $|\phi^\pm\rangle\,AB = \frac{1}{\sqrt{2}}(|00\rangle\,AB \pm |11\rangle\,AB)$. Alice holds qubit A, while Bob holds qubit B and his maximally entangled state $|\phi^+\rangle\,12 = \frac{1}{\sqrt{2}}(|00\rangle\,12 + |11\rangle\,12)$. Because no alternative measurement is required, both Alice and Bob perform SQM on their respective qubits in a computational basis. Alice's SQM device is trusted while Bob's is not. As Alice and Bob measure their qubits before they become entangled, there are four possible outcomes of the measurements: $00, 01, 10,$ and 11. At this point, their results are not yet correlated.

6. **BSM:** Bob sends his qubit 1 and the received qubit B to the not-trusted BSM device. BSM operations are $(\mathbb{I} \otimes CNOT \otimes \mathbb{I})$ and $(\mathbb{I} \otimes H \otimes \mathbb{I} \otimes \mathbb{I})$, the system's four qubit

states becoming

$$|\Psi\rangle_{AB12} = \frac{1}{2}(|0000\rangle + |0100\rangle + |0011\rangle + |0111\rangle + |1010\rangle - |1110\rangle + |1001\rangle - |1101\rangle). \tag{6.39}$$

The BSM results of the qubits B and 1 could be in one of four values: $\{00, 01, 10, \text{or } 11\}$ with the first bit representing the qubit B ($|\psi\rangle_B$) and the second bit representing the qubit 1 ($|\psi\rangle_1$).

7. **Desired state preparation:** The protocol is working in time-reversal fashion, see Figure 6.11. Bob starts by measuring his qubit 2. Then he sends B and 1 to the BSM device. A trusted classical unit is responsible for performing a conditional transformation on qubit 2 outputs based on BSM's output. If the classical bit obtained from measuring qubit 1 ($|\psi\rangle_1$) is one, Bob flips operation the classical bit obtained from measuring qubit 2($|\psi\rangle_2$).

$$|\Psi\rangle_{AB12} = \frac{1}{2}(|0000\rangle + |0010\rangle + |0100\rangle + |0110\rangle + |1001\rangle + |1011\rangle - |1101\rangle - |1111\rangle). \tag{6.40}$$

By doing so, Alice and Bob share the same results, i.e. either both get 0 ($|00\rangle_{A2}$) or both get 1($|11\rangle_{A2}$). By reordering the system's qubits we get,

$$\begin{aligned} |\Psi\rangle_{B1A2} = \frac{1}{2}(&|00\rangle_{B1}(|00\rangle + |11\rangle)_{A2} + |01\rangle_{B1}(|00\rangle + |11\rangle)_{A2} \\ &+ |10\rangle_{B1}(|00\rangle - |11\rangle)_{A2} + |11\rangle_{B1}(|00\rangle - |11\rangle)_{A2}). \end{aligned} \tag{6.41}$$

The state $|\psi\rangle_{A2}$ is either $|\phi^+\rangle_{A2}$ or $|\phi^-\rangle_{A2}$. As we only measure on a computational basis, the probabilities of obtaining 00 or 11 are not affected by the phase change.

8. **Repeat:** Alice and Bob repeat the protocol steps 1–7 until they get a sufficiently long string. Let us consider steps 1–7 as one round for convenience.

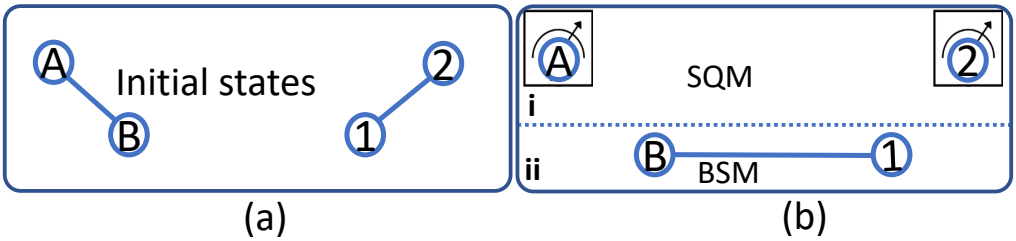


Figure 6.11: An illustration of the time-reversed fashion of DGL22. a) Alice and Bob's initial states. b) Alice and Bob start measuring qubits A and 2 then Bob performs BSM on quibits B and 1.

9. **Sifting:** Alice and Bob will proceed with the prior protocol steps without disclosing the measurement findings or the operations employed. When Bob receives the N-qubits, the sifting step begins. Alice and Bob use the public channel to expose the operations that are being employed for each qubit. When they utilise different operations, the results are used to perform a randomization test. The remaining results are saved for further examination to see whether they may be utilised for key generation.

10. **Error estimation:** Alice and Bob divide the key generation's results into two groups based on the unitary operation (Hadamard or Identity). They then choose a predetermined number of samples, say n_1 at random from the results group where they both do Hadamard operation, and publicly compare their results. The amount of mismatches r_1 determines the expected error rate $e_1 = \frac{r_1}{n_1}$. Similarly, the n_2 number of instances are chosen from the results group where they both apply the identity operation. The amount of mismatches r_2 determines the expected error rate $e_2 = \frac{r_2}{n_2}$. They require

$$e_1, e_2 < e_{max} - \delta_e, \tag{6.42}$$

where e_{max} denotes the maximum permissible error rate and δ_e denotes a small positive value. e_{max} is around 11% according to [130]. They proceed if these two different constraints are fulfilled. Otherwise, they will abort the protocol.

11. **Information reconciliation and privacy amplification:** If the protocol passed the error estimation test, the remaining results $(N - n_1 - n_2)$ are the raw key. The final secret key is obtained by Alice and Bob by conducting information reconciliation and privacy amplification using classical methods.

6.2.1 Security analysis

A secret key (K) is considered ideal if its bit string is completely unknown to Eve. The K's security is described through security definitions presented in [122].

Definition 6.2.1. Let K_A and K_B denote Alice's and Bob's key bit strings, respectively. A QKD protocol is said to be $\varepsilon_{correct}$ if

$$P_{AB}[K_A \neq K_B] \leq \varepsilon_{correct}, \tag{6.43}$$

where P_{AB} denotes the probability that the shared secret key is not correct. To consider the key correct, P_{AB} should be less than $\varepsilon_{correct}$.

Definition 6.2.2. A protocol is ε_{secret} if the system state in the presence of Eve who owns a quantum state E is

$$\frac{1}{2} \left\| \rho_{AE} - \tau_K \otimes \rho_E \right\|_1 \leq \varepsilon_{secret}, \tag{6.44}$$

where ρ_{AE} represents the correlation of Alice's and Eve's information; $\|.\|_1$ denotes the trace norm; ρ_{AE} is the correlation between Alice's and Eve's total knowledge; τ_K represents the fully mixed state of the generated key; and ρ_E is Eve's generic state. That is, ρ_{AB} is close to the perfect key by ε_{secret} where ε_{secret} is the failure probability threshold. The protocol failure indicates that Eve gets enough information about the key.

Definition 6.2.3. If DGL22 is both $\varepsilon_{correct}$ and ε_{secret}, it is considered to be ε_{secure} [115], with

$$\varepsilon_{correct} + \varepsilon_{secret} \leq \varepsilon_{secure}. \tag{6.45}$$

DGL22 guarantees that a key is generated if $\varepsilon_{correct} + \varepsilon_{secret} \leq \varepsilon_{secure}$, otherwise it aborts.

Assuming no attacks, the measurement results of Alice and Bob when they utilize the same operation should have deterministic correlations. If Alice and Bob operate different operations, their results will be uncorrelated. As a result, Eve must use H or I operations to ensure that her results correlate with the generated key. Fortunately, the eavesdropper has no idea which operation Alice and Bob picked. Thus, any attempt would break the security definitions. DGL22 is a ε_{secure} protocol since it has enough test results from each group.

6.2.1.1 Finite-key security

The security definitions provided previously are *composable*. That is, the protocol security of can be derived from the security of its component proofs.

Lemma 6.2.1 (DGL22's Security)**.** DGL22 is ε_{total} secure which can be represented by the simplified version of error propagation, $\varepsilon_{total} \geq \varepsilon_{EC} + \varepsilon_{secure} + \varepsilon_{PA}$

Proof. The ideal system that we would like to have is the one that gives a secret key to Alice and Bob, but gives nothing to the adversary. On the other hand, a real system is one that gives a secret key to Alice and Bob, but leaks some information to the adversary. The real should be ε_{total} close to the ideal [115]. A protocol is ε_{total} secure if

$$\frac{1}{2} \left\| \rho_{real} - \rho_{ideal} \right\|_1 \leq \varepsilon_{total} \tag{6.46}$$

$$(1 - \varepsilon_{abort}) \frac{1}{2} \left\| \rho_{ABE} - \tau_K \otimes \rho_E \right\|_1 \leq \varepsilon_{total} \tag{6.47}$$

where ρ_{ABE} represents the correlation between the generated key and Eve's information; and ε_{abort} is the probability of aborting the protocol. Eq. 6.47 helps to distinguish how close the real system should be from the ideal system. For simplicity, let $\gamma = (1 - \varepsilon_{abort})$.

From the triangle inequality, we get

$$\begin{aligned}\gamma\frac{1}{2}\left\|\rho_{ABE}-\tau_K\otimes\rho_E\right\|_1 \leq \frac{1}{2}\left\|\rho_{ABE}-\bar{\rho}_{ABE}\right\|_1 \\ +\frac{1}{2}\left\|\bar{\rho}_{ABE}-\tau_K\otimes\rho_E\right\|_1,\end{aligned} \tag{6.48}$$

where $\bar{\rho}_{ABE}=\frac{1}{Tr(\rho_{ABE})}\rho_{ABE}$. Let us take the first part, $\frac{1}{2}\left\|\rho_{ABE}-\bar{\rho}_{ABE}\right\|_1$. The system E appears in both terms which means Eve affects both Alice's and Bob's states. We can quantify this by only comparing Alice's and Bob's states and removing Eve's state,

$$\begin{aligned}\gamma\frac{1}{2}\left\|\rho_{ABE}-\tau_K\otimes\rho_E\right\|_1 &\leq \frac{1}{2}\left\|\rho_{AB}-\bar{\rho}_{AB}\right\|_1 \\ &\quad+\frac{1}{2}\left\|\bar{\rho}_{ABE}-\tau_K\otimes\rho_E\right\|_1. \\ &\leq P_{AB}[K_A\neq K_B] \\ &\quad+\frac{1}{2}\left\|\bar{\rho}_{ABE}-\tau_K\otimes\rho_E\right\|_1 \\ &\leq \varepsilon_{EC}+\frac{1}{2}\left\|\bar{\rho}_{ABE}-\tau_K\otimes\rho_E\right\|_1\end{aligned} \tag{6.49}$$

We demonstrate that the first half represents the protocol is ε_{EC}. Let us now focus on the second part, $\frac{1}{2}\left\|\bar{\rho}_{ABE}-\tau_K\otimes\rho_E\right\|_1$. We can see that system B is a copy of A, hence we can throw it,

$$\begin{aligned}\frac{1}{2}\left\|\rho_{ABE}-\tau_K\otimes\rho_E\right\|_1 \leq P_{AB}[K_A\neq K_B] \\ +\frac{1}{2}\left\|\bar{\rho}_{AE}-\tau_K\otimes\rho_E\right\|_1\end{aligned} \tag{6.50}$$

The second part, $\frac{1}{2}\left\|\bar{\rho}_{AE}-\tau_K\otimes\rho_E\right\|_1$, represents the protocol security ε_{secure}, see definition 6.2.2. The protocol's secrecy can be demonstrated by utilizing the Quantum Leftover hashing lemma [123, 145] to give us the upper bound,

$$\frac{1}{2}\left\|\bar{\rho}_{AE}-\tau_K\otimes\rho_E\right\|_1 \leq \varepsilon_{secret}+\frac{1}{2}\sqrt{2^{L-H^{\varepsilon}_{min}(K^M_A|E_CE)}} \tag{6.51}$$

where E is Eve's total knowledge of K_A^M. E is given by,

$$E = E_q E_C E_F \tag{6.52}$$

where E_q is Eve's quantum system, E_C is the information gained error correction, and E_F is the knowledge about the hash function. Using the min-entropy chain-rule [145],

$$H_{min}^{\varepsilon}(K_A^M|E_C E) \quad \geq H_{min}^{\varepsilon}(K_A^M|E) - log_2|E_C| \tag{6.53}$$

where $log_2|E_C| = b_{leak} - \log_2(\frac{2}{\varepsilon_{EC}(\varepsilon_{secure})^2})$. By inserting this in Equation 6.53,

$$\begin{aligned} H_{min}^{\varepsilon}(K_A^M|E_C E) \geq H_{min}^{\varepsilon}(K_A^M|E) - b_{leak} \\ - \log_2(\frac{2}{\varepsilon_{EC}(\varepsilon_{secure})^2}) \end{aligned} \tag{6.54}$$

By substituting Equation 6.54 into 6.51 we obtain:

$$\leq \varepsilon_{secret} \quad + \frac{1}{2}\sqrt{2^{L-(H_{min}^{\varepsilon}(K_A^M|E) - b_{leak} - \log_2(\frac{2}{\varepsilon_{EC}(\varepsilon_{secure})^2}))}} \tag{6.55}$$

$$\begin{aligned} \frac{1}{2}\left\|\rho_{AE} - \tau_K \otimes \rho_E\right\|_1 &\leq \varepsilon_{secret} + \frac{1}{2}\sqrt{2^{log_2(2\varepsilon_{PA})^2}} \\ &= \varepsilon_{secret} + \varepsilon_{PA}. \end{aligned} \tag{6.56}$$

The secrecy of DGL22 has been demonstrated above. Therefore, the combination of the secrecy proof with an ε_{EC} proof allows for the establishment of DGL22's security, where $\varepsilon_{total} \geq \varepsilon_{EC} + \varepsilon_{secure} + \varepsilon_{PA}$. □

6.2.2 Sophisticated attacks security analysis

There are several ways that Eve might try to break the protocol. Hereafter, I explore the security of the protocol against Eve's sophisticated attacks.

6.2.2.1 Detector blinding attack

This attack is similar to a measure-resend attack, except that Eve is blinding Bob's detector before sending the results. Once Bob's detector is blind, it only clicks if the signal power is above a threshold D_{th}. A bright pulse that Eve sends to Bob in accordance with the results of her measurement would, deterministically, give Eve the same result as Alice's (Bob's) if their basis is the same and no result if their basis is different. In this attack, Eve intercepts the transmitted qubits and measures them, then resends the classical results (0 or 1) to Bob. DGL22 is equipped with two measurement devices: BSM and SQM. Eve might try to blind the BSM device as it is connected to the quantum channel. Interestingly, DGL22 does not require alternative measurement; it requires performing Hadamard and identity operations randomly instead. If Bob is not performing Hadamard operations, Eve might blind the BSM detector. Let us say Eve is sending a classical bit of 0 and neither Eve nor Bob performs a Hadamard operation. Bob's system, which contains qubits B, 1, and 2 will be

$$|\psi\rangle_{B12} = |0\rangle_B \otimes |\phi^+\rangle_{12} = \frac{1}{2}(|000\rangle_{B12} + |011\rangle_{B12}). \tag{6.57}$$

Bob sends B and 1 qubits to the BSM device, which performs ($CNOT \otimes \mathbb{I}$) and ($H \otimes \mathbb{I} \otimes \mathbb{I}$),

$$|\psi\rangle_{B12} = \frac{1}{2}(|000\rangle_{B12} + |100\rangle_{B12} + |011\rangle_{B12} + |111\rangle_{B12}). \tag{6.58}$$

If the value of qubit 1 obtained from the BSM is one, as in $|0\mathbf{1}1\rangle_{B12} + |1\mathbf{1}1\rangle_{B12}$, Bob performs bit-flip operation on qubit 2 results obtained from the SQM,

$$|\psi\rangle_{B12} = \frac{1}{2}(|000\rangle_{B12} + |100\rangle_{B12} + |010\rangle_{B12} + |110\rangle_{B12}). \tag{6.59}$$

If Eve sends 1, the system will be

$$|\psi\rangle_{B12} = \frac{1}{2}(|011\rangle_{B12} - |111\rangle_{B12} + |001\rangle_{B12} - |101\rangle_{B12}) \tag{6.60}$$

As we can see, Bob's final obtained bit is always similar to Eve's sent classical bit. However, if Eve sends a classical bit 0 and Bob performs a Hadamard operation, the system will start as

$$|\psi\rangle_{B12} = |\psi\rangle_{B} \otimes |\phi^{+}\rangle_{12} = \frac{1}{2}(|000\rangle_{B12} + |011\rangle_{B12} + |100\rangle_{B12} + |111\rangle_{B12}). \tag{6.61}$$

The final state reads,

$$|\psi\rangle_{B12} = \frac{1}{2}(|000\rangle_{B12} + |100\rangle_{B12} + |010\rangle_{B12} + |110\rangle_{B12}) + |011\rangle - |111\rangle + |001\rangle - |101\rangle). \tag{6.62}$$

As we can see, Bob's final obtained bit is totally random, which proves the infeasibility of such an attack.

6.2.3 Protocol performance

The proposed protocol achieves both security and practicality. I consider several efficiency metrics, communication, sifting, and key-establishment delay.

Communication efficiency σ **:**To compute the protocol communication efficiency, the

definition proposed by Cabello in [35] is used,

$$\sigma = \frac{b_s}{q_t + b_t} \ , \tag{6.63}$$

where b_s represents the expected number of secret bits obtained by Alice (Bob), q_t is the number of qubits exchanged, and b_t represents the number of classical messages sent. The number of classical messages can be given by

$$b_t = b_t^{Bm} + b_t^{Am} \tag{6.64}$$

where b_t^{Bm} is the number of classical messages required before the measurements and b_t^{Am} is the number of classical messages required after the measurement for the sifting and error estimation purposes. If we do not include the messages delivered for the purpose of sifting (only the messages that are used in each round for inferring the key), DGL22, [158], and [76] are 100% efficient as no classical messages are required before the measurement. After the measurement and for sifting purposes, two messages are enough to convey all the chosen random operators by both Alice and Bob. DGL22 outperforms other protocols as it achieves the highest efficiency, 0.33. This is due to minimizing the number of sent qubits and classical messages. This comparison considers the messages sent before and during the sifting process. The messages required for error correction, information reconciliation, and privacy amplification are not included as classical algorithms are used for these purposes.

Key establishment time: The key establishment time is the total time that the protocol takes to finish establishing a key. The total time is a summation of several delays,

$$T = N(q_t \times t_{qm} + b_t^{B_m} \times t_{cm}) + b_t^{A_m} \times t_{cm} \tag{6.65}$$

where t_{qm} is the propagation time of a quantum message; t_{cm} is the propagation time for a

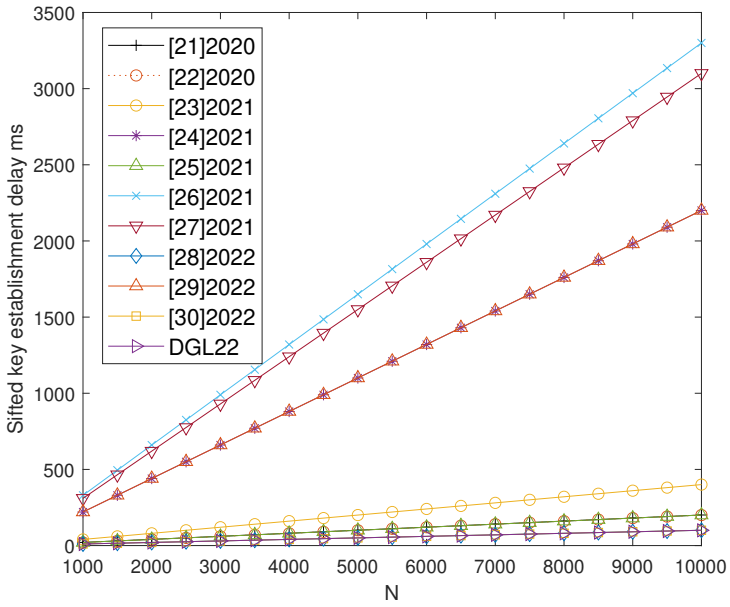


Figure 6.12: A numerical comparison between DGL22 and several recent protocols in terms of key establishment delay. $t_{qm} = 10^{-2}$ms and $t_{cm} = 10^{-1}$ ms are assumed.

Table 6.6: Communication efficiency, σ, of DGL22 in comparison with the most recent related protocols. This table considers the messages sent before and during the sifting process. The messages required for error correction, information reconciliation, and privacy amplification are not included.

Protocol	Year	q_t	$b_t^{B_m}$	b_t^{Am}	**Communication efficiency** (σ)
[105]	2020	2	0	4	0.16
[104]	2020	2	0	2	0.25
[162]	2021	4	0	2	0.16
[118]	2021	2	2	1	0.20
[7]	2021	2	0	2	0.25
[41]	2021	3	3	0	0.25
[74]	2021	1	3	0	0.25
[158]	2022	1	0	3	0.25
[117]	2022	2	2	2	0.16
[76]	2022	1	0	3	0.25
DGL22	Now	1	0	2	**0.33**

classical message; and $b_t^{A_m} \times t_{cm}$ is a delay caused by the messages that are sent by Alice and Bob for the sifting purpose. Figure 6.12 shows the Key establishment time efficiency of DGL22.

Sifting efficiency (S_η): Most popular QKD protocols discard a large portion of the measurements due to the alternative measurements' disagreement. QKD protocols may be divided into two primary categories: random-based protocols and memory-assisted protocols. In random-based protocols, Alice and Bob pick random actions, such as the measurement basis. Then, they discard the results if they choose different actions. Random-based protocols can be fair or biased. The biased random-based protocol reduces the number of discarded results and minimizes the number of results that are used for Eve's presence detection. In the memory-assisted QKDs, Alice and Bob store the qubits in quantum memory. Then, Alice, for example, will select the random actions and send them to Bob. The memory-assisted protocols do not discard results as there is no disagreement; however, 50% of the results are used for Eve's presence detection. DGL22 is a biased random-based protocol. Because DGL-22 employs biassed selection probability (p) in conjunction with proper error analysis, the number of results used as raw keys has increased. As a result, there's a good chance that a qubit will be used in key generation, which means that S_η will be maximized. Table 6.6 compares protocols that are either random-based or memory-assisted. Figure 6.13 shows a numerical comparison between DGL22 and other recent protocols. This comparison shows the efficiency of these protocols in minimising the proportion of discarded results and the proportion of results required for Eve's presence detection. We used the same simulation parameters for all of the compared protocols. s can be seen, DGL22's flexibility in choosing p, allows for achieving a high key rate while maintaining security. As illustrated in Figure 6.13, DGL22 with a low probability p necessitates a high N to achieve a high key rate. Low p and N actively demonstrate that DGL22 cannot generate a sufficient number of results for error estimation. The parameter values used in the simulation are: $\omega = 1$, $e_{max} = 0.11$, $b_{leak} = 1.05h(e_{max})$, $\varepsilon_{EC} = 10^{(-10)}$, and $\varepsilon_{abort} = 10^{-1}$. The value of p of DGL22 is set in such a way that $N(p^2 - \lambda) = n_1 = \Omega(logN)$, where λ is a small positive number (i.e., the error due to statistical fluctuations) chosen by Alice and Bob.

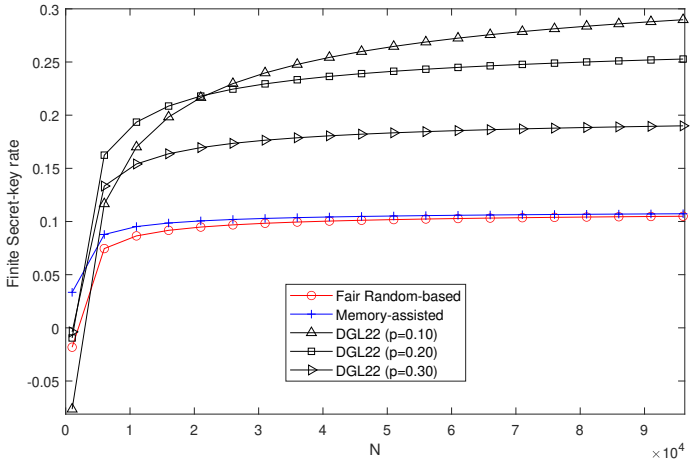


Figure 6.13: Finite Secret-key rate of DGL22 compared with several recent protocols: Fair random-based protocols [104, 105, 158, 162], and [76]; and memory-assisted protocols [7, 74, 118], and [117].

Receiver's detectors efficiency: Usually, DI-QKDs need high-quality detectors, i.e. $D_q > 91.1\%$ [26]; however, 1SDI-QKD can use less efficient detectors, $D_q > 65.9\%$ [26], that are currently available detectors in the market. Figure 6.14 shows the key rate as a function of Bob's detector efficiency and the channel loss. Bob has several detectors, and we assume their quality is similar. The received qubits' visibility can be given by:

$$Vis = \frac{\mu 10^{-\alpha F_L/10} D_q}{\mu 10^{-\alpha F_L/10} + 2P_e} \tag{6.66}$$

where α is fibre's attenuation $\alpha = 0.2$, P_e is the probability of an error count per clock cycle $P_e = 8.5 \times 10^{-7}$, [61], μ is the average number of sent photons by Alice, and D_q is Bob's detector efficiency. The channel fidelity can be determined by:

$$Fid = \frac{1 + 3Vis}{4}. \tag{6.67}$$

where Vis represents the visibility. The channel disturbance is

$$D_r = 1 - Fid. \tag{6.68}$$

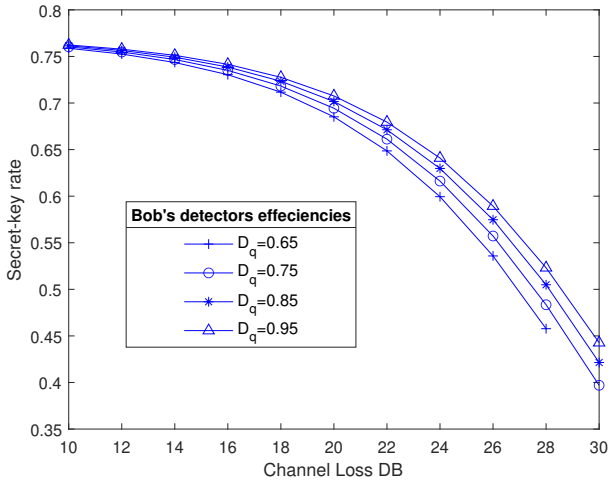


Figure 6.14: An illustration of the Bob's detector efficiency, D_q, effects on the key rate. The parameters are as in Figure 6.13 except Bob's detector efficiency is $D_q \in \{0.65, 0.75, 0.85, 0.95\}$, $N = 10^5$, and the fibre channel loss , α, is 0.2/km.

In the event that Alice and Bob find the channel fidelity insufficient based on the results of their testing, they abort and restart the protocol. They proceed if they are confident that the fidelity is high.

6.2.4 Discussion

This work proposed a practical, secure, and efficient QKD protocol, called DGL22. DGL22 is an entanglement-based RDI-QKD. The proposed protocol relies on the fact that Bob has two measurement devices, namely, BSM and SQM, while Alice has only an SQM device. Bob's SQM device is isolated to protect it from any detector-blinding attacks. Bob's device is a time-reversal teleportation machine that will teleport the received qubit B to his qubit 2. Bob starts by sending his qubit 2 to the not-trusted SQM device. Then he sends B and 1 to the BSM device. A trusted classical unit is responsible for performing a conditional transformation on SQM's outputs based on BSM's outputs. Alice (Bob) chooses at random whether to perform a Hadamard or identity operation on qubit A (B). DGL has integrated several measures to assure its security. It uses two measurement devices, works on time reversal, and requires performing a Hadamard gate at random. General security

proof of DGL has been provided. It is demonstrated that DGL22 is capable of resisting the detector blinding attack, as well as all known attacks permitted by quantum mechanics. DGL22 is easy to implement using the current state-of-the-art technology as it relies on fewer assumptions compared to DI-QKD protocols. It is more effective and secure than conventional QKDs while also being simpler to build than MDI-QKD. DGL22 is an efficient RDI-QKD protocol that does not require classical messages per round. DGL22 achieves high communication efficiency (0.33 bit/qubit) and low key establishment delay. DGL22 achieves a high key rate due to adopting the biased random-based approach in selecting the unitary operators, namely, Hadamard and identity. DGL22 does not require high-quality detectors.

6.3 QKeyShield vs DGL22

In this section, I will delve into the similarities and distinctions between the QKeyShield and DGL22 protocols. Both protocols utilize the principles of quantum mechanics to establish a shared secret key and have been specifically designed to resist common attacks such as eavesdropping and detector blinding. Both QKeyShield and DGL22 are RDI-QKD protocols In RDI-QKD protocols, the key generation process does not rely on the specific characteristics of the measurement devices used by Bob. Both protocols also employ a trusted entanglement source, which is controlled by Alice, and utilize a quantum channel for qubit transmission and a classical channel for error estimation, information reconciliation, and privacy amplification. Furthermore, both protocols utilize a biased random-based approach in selecting unitary operators, specifically the Hadamard and identity operators, to enhance the security and efficiency of the key generation process. They are relatively easy to implement using current state-of-the-art technology.

However, it should be noted that DGL22 also includes additional measures to further enhance its security and efficiency, such as isolating Bob's SQM measurement device to

prevent any potential side-channel attacks, operating in time reversal mode, and using teleportation to improve the efficiency of the key generation process. These added features set DGL22 apart from QKeyShield, making it a more robust and efficient RDI-QKD protocol. Table 6.7 shows the main differences between them in terms of their modes of operation, entanglement swapping, teleportation, initial state preparation, measurement devices, and the total number of qubits and detectors required.

Table 6.7: Comparison of key differences between QKeyShield and DGL22 RDI-QKD protocols

Feature	QKeyShield	DGL22
Modes	Normal and time-reversal	Only time-reversal
Entanglement swapping	Required	Not used
Teleportation	Not used	Used
Initial state preparation	3 entangled pairs	2 entangled pairs
Total Number of qubits	6	4
Total Number of detectors	8	6

As we can see from Table 6.7, DGL22 operates only in time-reversal mode, unlike QKeyShield which operates in both normal and time-reversal mode. DGL22 does not use entanglement swapping, instead, it uses teleportation which is more efficient as it only requires one entangled pair. DGL22 also uses fewer qubits and detectors, and the initial state is prepared using fewer entangled pairs.

It is clear that DGL22 has several advantages over QKeyShield in terms of efficiency, complexity, assumptions, robustness, scalability and cost.

- **Efficiency:** DGL22 requires fewer qubits, entangled pairs, detectors, and measurement devices, making it a more efficient protocol. This means that DGL22 uses fewer resources to establish the shared secret key, resulting in a higher key generation rate.

- **Complexity:** DGL22 is designed to be more complex than QKeyShield, requiring specific design for the inner circuits. This complexity is necessary to achieve the added security and efficiency benefits of DGL22.

- **Assumptions:** DGL22 requires two measurement devices SQM and BSM for Bob and only SQM for Alice, and a trusted classical unit for Bob. It assumes that Bob's SQM device is isolated from the quantum channel to prevent any possibility of a detector blinding attack. This added isolation and use of multiple measurement devices improves the security of the protocol.

- **Robustness:** DGL22 is designed to be more robust than QKeyShield. This is achieved through the use of added security measures such as isolation of Bob's SQM measurement device, operation in time reversal mode, and the use of teleportation.

- **Scalability:** DGL22's design allows for better scalability by requiring fewer resources. This means that DGL22 can be easily implemented in larger-scale systems without incurring a significant increase in cost or complexity.

- **Implementation cost:** DGL22 requires less devices and detectors than QKeyShield. This means that DGL22 has a lower cost of implementation, making it more accessible to a wider range of users.

In summary, DGL22 is a more efficient, complex, robust, scalable, and cost-effective RDI-QKD protocol in comparison to QKeyShield. The added security and efficiency benefits of DGL22 make it a valuable alternative to QKeyShield for secure communication systems.

6.4 Concluding Remarks

In this chapter, I have discussed the QKeyShield and DGL22 protocols in detail. Both protocols are Receiver-Device-Independent QKD protocols that utilize the principles of quantum mechanics to establish a shared secret key. I have provided a comprehensive overview of the specifications of both protocols.

Both protocols employ a trusted entanglement source, use a quantum channel for qubit transmission and a classical channel for error estimation, information reconciliation, and

privacy amplification. Furthermore, both protocols utilize a biased random-based approach in selecting unitary operators to enhance the security and efficiency of the key generation process.

However, it should be noted that DGL22 also includes additional measures to further enhance its security and efficiency, such as isolating Bob's SQM measurement device to prevent any potential side-channel attacks, operating in time reversal mode, and using teleportation to improve the efficiency of the key generation process. These added features set DGL22 apart from QKeyShield, making it a more robust and efficient RDI-QKD protocol.

In conclusion, QKeyShield and DGL22 are both promising RDI-QKD protocols, but DGL22 has several advantages over QKeyShield in terms of security and efficiency. Therefore, DGL22 is a more suitable choice for practical implementation. As the field of quantum communication continues to evolve, it will be exciting to see how these protocols and others like them are developed and applied in real-world scenarios.

Chapter 7

Hybrid Quantum-Safe/Quantum-based Key Establishment

In this thesis, I propose several protocols for secure key establishment, which include both quantum-safe and quantum-based approaches. I begin by introducing the quantum-safe protocols, KeyShield and KeyShield2, which are designed for group key establishment. These protocols address the high computational complexity issue of the use of a linear system of equations as a key establishment protocol in the literature. KeyShield does this by using member grouping, in which a secret vector is assigned to a set of members instead of a single member while maintaining the same level of security, and by using a banded matrix instead of a full secret matrix, which keeps the difficulty of guessing a matrix row (secret vector) higher than the cost of guessing the shared key, TEK. KeyShield's operations are supported by mathematical analysis and proofs when necessary, and it opens a new door for utilizing the difficulty of finding a unique solution to an underdetermined linear system of equations, thus more research and innovation in this direction are needed. KeyShield2 is an improved version of KeyShield that addresses the vulnerabilities identified in the original protocol. The recommended countermeasures do not significantly impact the performance or efficiency of the protocol. The key server in KeyShield2 solves

a banded linear system, which allows for faster computation.

We then present the quantum-based protocols, QKeyShield and DGL22, which are tailored for two-member key establishment. Both protocols rely on trusted devices at Alice's side and untrusted devices at Bob's side, and they eliminate the need for Alice and Bob to pick between alternative measurements, thus increasing the rate of generated bits. QKeyShield is based on entanglement-swapping, while DGL22 is based on quantum teleportation. QKeyShield and DGL22 also minimize the attack surface by allowing Alice to send a single qubit of an entangled state (called the shared state) to Bob, and by integrating several measures to protect the sent qubit. Security proofs and analyses have been provided to show that the proposed protocols can securely and effectively allow Alice and Bob to agree on a secret key.

One of the main challenges in the secure key establishment is the need to migrate from current classical systems to more secure quantum systems, while still maintaining security during the transition period. Both QKD and quantum-safe key distribution are approaches that aim to address this challenge, but they each have their own limitations. QKD technology is currently limited to relatively short distances, while quantum-safe protocols in the literature may not be secure or scalable. The proposed quantum-safe and scalable group key establishment protocol, KeyShield2, relies on the Kyber algorithm for distributing secret vectors. While Kyber is a quantum-safe key encapsulation algorithm, it is not yet a standard and is susceptible to various side-channel attacks. Given these limitations, it is natural to consider a hybrid approach that combines the strengths of both QKD and quantum-safe key distribution. Such an approach could potentially offer the benefits of both technologies while minimizing their respective limitations. For example, a hybrid approach could use QKD technology to establish secure keys over short distances, while relying on quantum-safe protocols for key establishment over longer distances. This could provide a more flexible and scalable solution for secure key establishment during the transition period from classical to quantum systems.

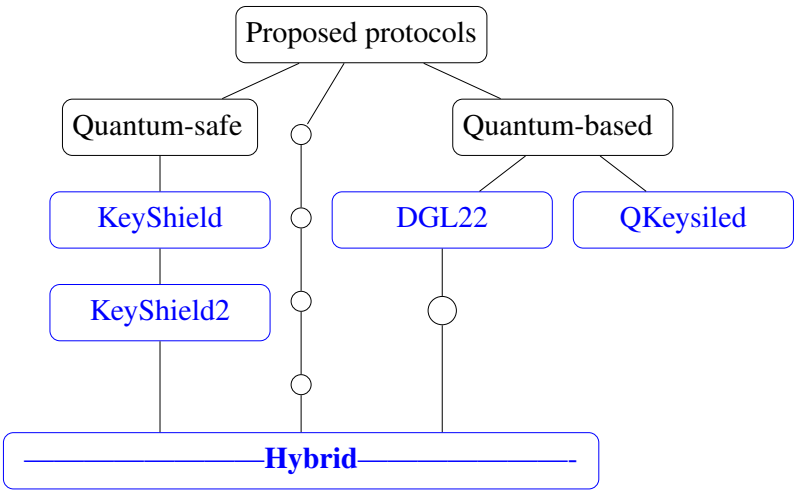


Figure 7.1: The proposed protocols in this thesis.

In the proposed hybrid approach, I aim to combine the two-member quantum-based solution (DGL22) with the group quantum-safe distribution solution (KeyShield2), as shown in Figure 7.1. This hybrid approach aims to facilitate group key establishment and offer a more flexible and scalable solution during the transition period from classical to quantum systems.

7.0.1 System model

In the context of secure key establishment, the main goal is to establish a traffic encryption key (TEK) that each legitimate member can use securely to encrypt and decrypt messages. To achieve this goal, the system must be able to establish the TEK in a secure and reliable manner, regardless of the specific communication channels being used. A hybrid key establishment system is one that combines elements from both quantum and classical technologies in order to achieve a more secure and flexible solution. In a hybrid key establishment system, there are two types of communication channels: quantum channels and classical channels. Quantum channels are those that use principles of quantum mechanics to transmit information, while classical channels use traditional methods such as electrical or optical signals. Each member of the system should be able to communi-

cate through both types of channels in order to take advantage of the strengths of each. For example, a member may use a quantum channel for secure two-member key establishment over short distances, and a classical channel for group key distribution over longer distances. This flexibility allows the hybrid system to adapt to different communication scenarios and requirements.

7.0.1.1 Quantum Channels

In a key establishment system, the primary goal is to securely establish a shared key between two parties, typically referred to as Alice and Bob. A quantum channel is a communication channel that uses quantum mechanics to transmit information. In the context of key establishment, Alice and Bob are the two parties at the ends of the quantum channel. In some cases, Alice may initiate a quantum-based protocol, such as DGL22, to establish a shared key with Bob. DGL22 is just one example of a quantum-based protocol that can be used for key establishment. There are many other quantum-based protocols that can be used for this purpose as well.

In a group key establishment scenario, multiple parties are involved and a central party known as the root node or key server initiates the process and establishes the network topology or the connections between the parties. The root node's role is to initiate the key establishment process and ensure that all parties can securely establish a shared key, known as $SDEK$, which is used to encrypt and decrypt data.

To create the network topology, the root node may divide the members into logical subsets with a group representative device deployed in an optimal location for each subset, as shown in Figure 7.2. The set of members, represented by V, is divided into s logical subsets, with each subset containing su members. Members located near each other geographically are placed into the same subset. If a group representative is too far from the root node for a direct connection, intermediate nodes may be used to extend the connection.

During the key establishment process, the root node acts like Alice when running the DGL22 protocol with its children nodes, while the members act like Bob. The group representatives and intermediate nodes act as both Alice and Bob, behaving as Bob when dealing with a parent node closer to the root node and as Alice when dealing with a child node closer to the members. Each node runs multiple copies of the DGL22 protocol with its child nodes to establish an $SDEK$. The group representatives store the members' $SDEK$, while the root node only stores the group representatives' $SDEK$. Intermediate nodes use quantum channels to generate a key, but these keys are only used to secure the transmission of the representatives' keys to the root node.

In the key establishment process, the nodes from the root down to the representatives form a chain. The upper node is referred to as the parent node, while the lower node is referred to as the child node. The intermediate nodes act as a chain to deliver the group representative key (GQK_i) to the root node. As shown in Figure 7.3, the root node uses the key k_4 as GQK. Each node has two keys - for example, the intermediate node j has the keys k_i and k_{i+1}. The key k_i is generated in collaboration with the parent node, while k_{i+1} is generated in collaboration with the child node.

The intermediate nodes use symmetric encryption to encrypt k_{i+1} using the key k_i, result-

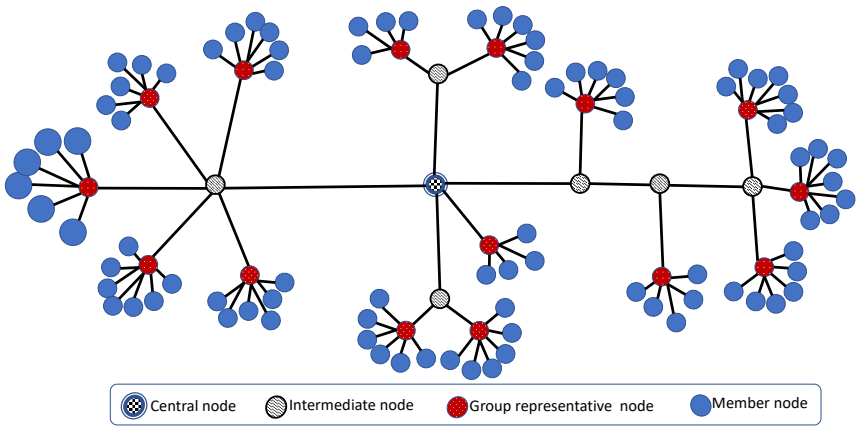


Figure 7.2: A random network topology of the members, with quantum channels represented by links. These channels are used to establish Symmetric Data Encryption Keys ($SDEKs$).

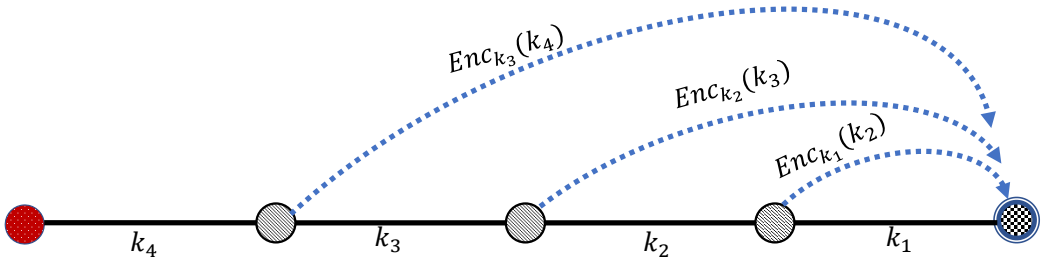


Figure 7.3: Diagram of chained nodes for delivering the group representative key to the root.

ing in the encrypted key $Enc_{k_i}(k_{i+1})$. This encrypted key is then sent to the root node using a classical channel. The root node is the first node in the chain and has the key k_1. To find the final key, the root node uses the following process:

$$GQK = Dec_{(Dec_{(Dec_{k_1}(\bar{k_2}))}(\bar{k_3}))}(\bar{k_4}). \tag{7.1}$$

Where Dec represents decryption using the corresponding key.

This process is only done at the beginning of the protocol and is the main purpose of the quantum channels. $GQKs$ have a lifetime, known as the cryptoperiod, which can be several years as recommended by NIST [13].

7.0.1.2 Classical Channels

Classical channels are used for two main purposes in the key establishment process. First, they are used to distribute the KeyShield2 secret vectors to the group representatives and members as shown in Figure 7.4. The key server sends the secret vectors to the group representatives, encrypted using their $SDEKs$ (GQK). The group representatives then forward the secret vectors to the members, encrypted using the members' $SDEKs$ ($MQKs$).

Second, the key server uses a classical channel to broadcast the KeyShield secure lock X, as shown in Figure 7.5.

Once the keys, $GQKs$ and $MQKs$, are established using the quantum channel, the pro-

tocol follows the steps outlined in Figure 7.6.

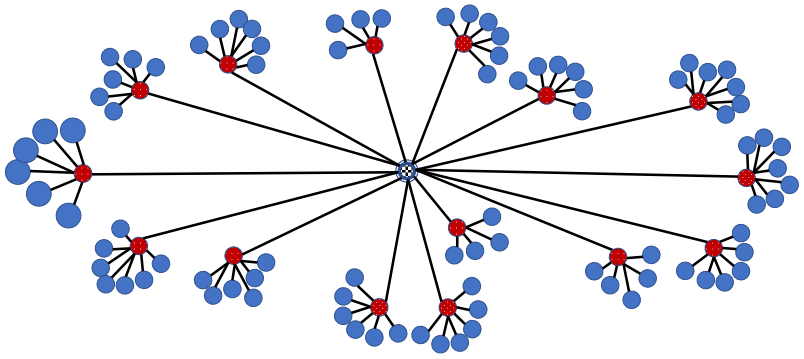


Figure 7.4: A network topology showing the use of classical secure channels (represented by links) to transmit encrypted secret vectors from the key server to group representatives, who then forward the vectors to the members. No intermediate nodes are required in this process.

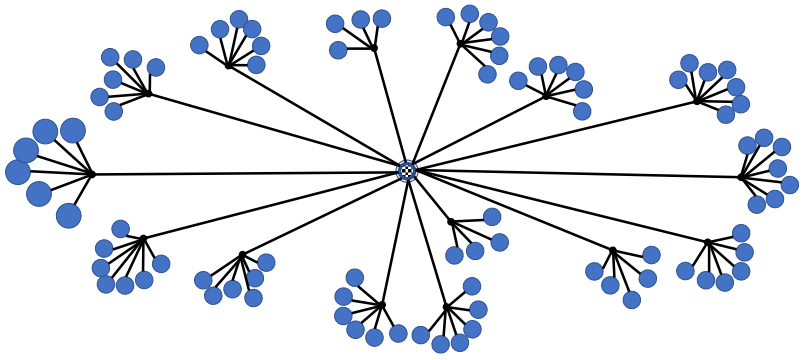


Figure 7.5: A network topology showing the use of classical channels (represented by links) to broadcast the secure lock X to all members. No intermediate nodes or representative nodes are required in this process.

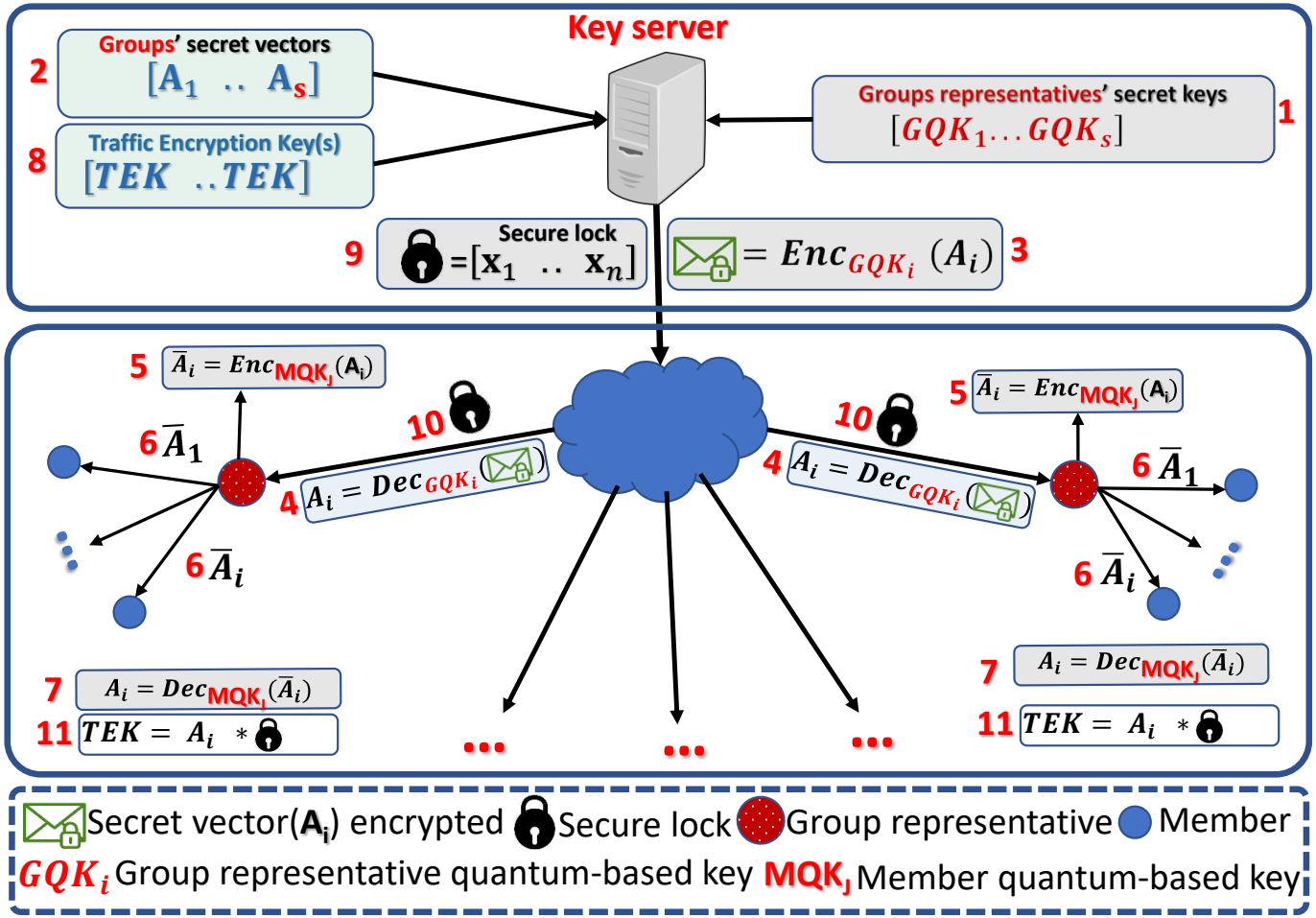


Figure 7.6: Proposed hybrid protocol.

As previously mentioned, the set of members (V) is divided into s logical subsets, with each subset containing su members. Instead of assigning a secret vector (A_i) to a single member (m_i), it is assigned to a subset of members (V_i). Each subset has a representative, referred to as m_{ij}, who has several responsibilities:

- Acting as a gateway between the members and the key server.
- Behaving as Bob when running the DGL22 protocol with the key server or the intermediate node to establish a key for the subset.
- Acting as Alice when running the DGL22 protocol with each member to establish a secret key with each member.

The representative (m_{ij}) initiates the DGL22 protocol with each member within its range to establish a $SDEK$. The key server will then send the subset secret vector (A_i) to the group representatives, encrypted using their $SDEKs$ (GQK). The group representative (m_{ij}) becomes Alice for the subset and encrypts the secret vector using the members' $SDEKs$ (obtained through the DGL22 protocol) and sends the encrypted secret vectors to each member.

Only members with an up-to-date key with the representative will be able to decrypt the message and obtain the correct secret vector, A_i. The key server then chooses the group TEK, solves the linear system, and broadcasts the secure lock, X, to all members. Each member will use their secret vector, A_i, to compute the shared key, $A_i \times X = TEK$.

7.0.2 Hybridizing benefits

There are several benefits to combining quantum-safe algorithms with quantum-based ones when transitioning to a quantum system. These benefits include:

1. **QKD limitations:** The currently available QKD technology is the most secure option for generating a group secret key, but it has limitations. For example, it is not practical for use over long distances or for large groups of users.

2. **Quantum-safe protocols limitations:** Quantum-safe protocols that have been proven to be secure may not be efficient or scalable, which can make them impractical for use in certain situations. They may also not have mechanisms in place for securing all key metadata, which can leave the keys vulnerable to attacks.

3. **Assurance of cryptographic strength:** New, supposedly quantum-safe algorithms have received less analysis by the cryptographic community than classical algorithms, which means there is less assurance that they are indeed safe. This lack of assurance will remain until these algorithms have undergone more years of analysis and testing.

4. **Forward compatibility:** By combining quantum-safe and quantum-based algorithms, it is possible to maintain security while still being able to take advantage of the advancements in QKD technology as they become available. This allows for a smooth transition to using QKD over long distances when it becomes practical.

5. **Backwards compatibility:** For parties that have already migrated to quantum-based systems, combining quantum-safe algorithms with quantum-based ones allows them to continue relying on quantum-safe alone at any time.

6. **Message overhead:** Combining quantum-safe and quantum-based algorithms minimize the message overhead required for key establishment. Only two classical broadcast messages are needed to achieve the rekeying process.

7. **Computation cost:** The computation cost is decreased when combining quantum-safe and quantum-based algorithms because the process involves solving a smaller linear system where each row is a secret vector for a group of members.

8. ***1-affects-n* phenomenon:** The "*1-affects-n*" phenomenon occurs when a single membership change in a group affects all other members. This can be addressed by formalizing the key timeline from generation to when all members commit to

a new key, which minimizes the impact of membership changes. Formalizing the key timeline involves establishing a clear process and timeline for generating, distributing, and updating cryptographic keys, and for members committing to using a new key. It helps minimize the impact of membership changes and disruptions, maintains key security, and ensures a consistent understanding of the key exchange process among group members.

7.0.3 Hybrid Protocol Security

The group communication system features a robust security mechanism, beginning with secure key establishment, as depicted in Figure 7.7. The hybrid protocol combines quantum-based and quantum-safe technologies to ensure strong security. The quantum-based protocol, DGL22, generates the $SDEKs$ necessary for securing the secret vector distribution of the quantum-safe protocol, KeyShield2. However, DGL22 is not utilized as a standalone application; rather, it serves as a tool to enhance the security of the hybrid protocol. KeyShield2 is responsible for securely distributing group keys and extending over

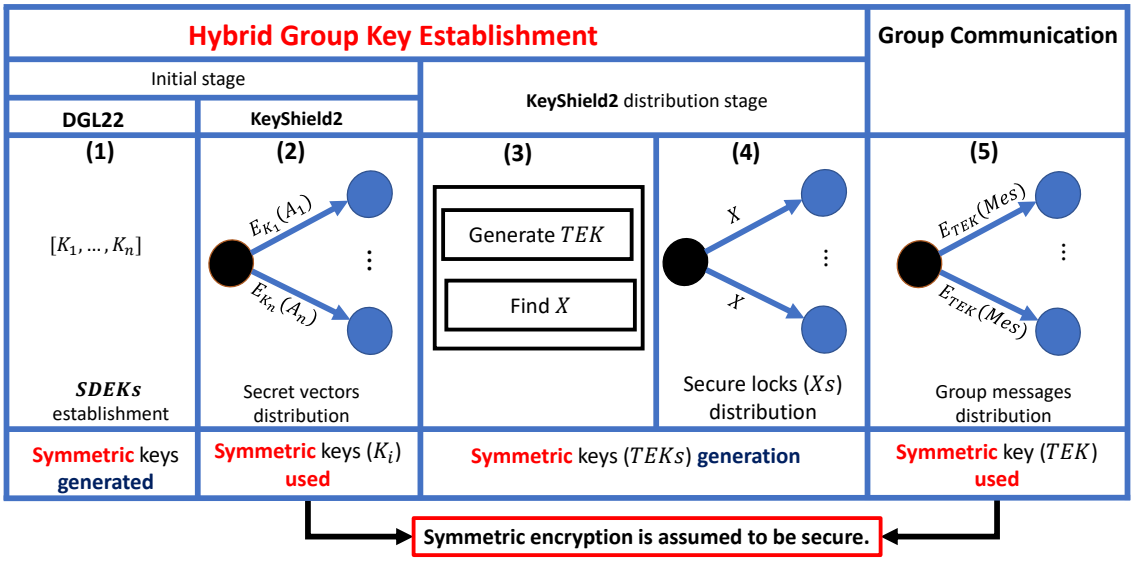


Figure 7.7: Components of the group communication system. The process sequence (1-5) begins with the proposed hybrid group key establishment followed by communication. Key servers are depicted as black circles while blue circles represent members.

long distances using classical communication channels in a broadcast manner.

The DGL22 and KeyShield2 protocols exhibit robust security features. DGL22, a QKD protocol, can detect any eavesdropping attempts, ensuring the confidentiality of exchanged information. In general, QKD protocols share this valuable property. KeyShield2 demonstrates indistinguishability under CPA and CCA scenarios, as discussed in Section 5.2.4. In IND-CPA, the attacker chooses two keys and receives corresponding secure locks. The randomness in the secret matrix and vector prevents the attacker from distinguishing between secure locks, securing KeyShield2 against chosen-plaintext attacks. In IND-CCA, the attacker sends two ciphertexts to a member who uses their secret vector to obtain related plaintexts. The attacker, without knowledge of the secret vector, cannot distinguish between plaintexts due to the infinite number of vectors satisfying the multiplication equation. This feature ensures KeyShield2's security against chosen-ciphertext attacks. Both protocols provide essential security measures for maintaining the confidentiality and integrity of communication.

The security of the hybrid protocol is crucial and must withstand attacks in the real world. To ensure this security, the concept of composability is employed, meaning that the protocol is designed to be equivalent to an ideal model of the task, capturing all security requirements. By doing this, the protocol can securely perform the intended task in the real world, even when faced with potential attacks.

The composition of the hybrid system consists of DGL22 (denoted as Q), KeyShield2 (denoted as G), and symmetric encryption (denoted as S). To understand the concept of composability in the context of the proposed hybrid protocol, consider the keys produced by the QKD protocol Q^{real}, which are later used by symmetric encryption system S^{real} to distribute the secret vectors of the quantum-safe protocol G^{real}, specifically the group key distribution protocol KeyShield2. Then, the key distributed by G^{real}, TEK, is used by symmetric encryption system S^{real} to encrypt the group's normal messages. Assuming that Q^{real} is π_1-secure, G^{real} is π_2-secure, and S^{real} is π_3-secure (meaning it

is π_3-indistinguishable from the ideal protocol S^{ideal}). π_1, π_2, and π_3 represent the maximum failure probabilities of Q^{real}, S^{real}, and G^{real}, respectively. The maximum failure probability of the composed system and is given by:

$$\pi = \pi_1 + \pi_2 + \pi_3, \tag{7.2}$$

It is important to note that symmetric encryption is used twice in the system. It is employed outside the scope of the hybrid protocol, meaning it is used to secure group communication. KeyShield2 protocol also utilizes symmetric encryption, in which the secret vectors are encrypted using the $SDEKs$ produced by DGL22 and distributed to the members. When the perfect key is used, symmetric encryption such as (AES-GCM) is secure and indistinguishable from a perfect encryption system [98]. Thus, we can say that $\pi_3 = 0$. Therefore, when a symmetric encryption system is combined with an π_1-secure QKD protocol Q^{real}, the resulting protocol is π_1-secure. Similarly, when a symmetric encryption system is combined with an π_2-secure quantum-safe protocol G^{real}, the resulting protocol is π_2-secure. While it is assumed that symmetric encryption is secure due to established standards, it falls outside the scope of this research. Hence, the focus of hybrid protocols is on the remaining components, Q^{real} and G^{real}. Therefore, the maximum failure probability of the composed system is given by

$$\pi = \pi_1 + \pi_2. \tag{7.3}$$

When a secure and indistinguishable symmetric encryption scheme is used for both securing the secret vector distribution and group communication, the resulting hybrid protocol is π-secure. That is, it distributes group keys that are π-indistinguishable from uniform randomness.

Proof. In previous discussions, it was established that π_1 and π_2 represent the acceptable failure probabilities for Q^{real} and G^{real}, respectively. When the two protocols are com-

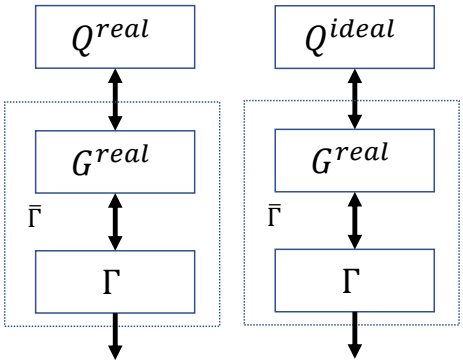


Figure 7.8: Combining the old distinguisher Γ with G^{real} yields a new distinguisher $\bar{\Gamma}$.

bined, the overall failure probability should be less than π, which is obtained by adding π_1 and π_2 together. Now, for the sake of argument, let's assume that the hybrid protocol $Q^{real} \circ G^{real}$ is not π-indistinguishable from $Q^{ideal} \circ G^{ideal}$. In this case, there must be a distinguisher (Γ) that produces results conforming to:

$$Pr[Q^{real} \circ G^{real}] - Pr[Q^{ideal} \circ G^{ideal}] > \pi. \tag{7.4}$$

We can consider the possibility that the real protocol G^{real} could be combined with either Q^{real} or a perfect QKD protocol Q^{ideal}. Subsequently, we can utilize the same Γ to distinguish between $Q^{real} \circ G^{real}$ and $Q^{real} \circ G^{ideal}$. Since Q^{real} is identical in both instances, we can regard it as part of an expanded distinguisher $\bar{\Gamma}$, which handles either Q^{real} or Q^{ideal}, as depicted in Figure 7.8.

As previously stated, Q^{real} is π_1-secure, so we find:

$$Pr[Q^{real} \circ G^{real}] - Pr[Q^{real} \circ G^{ideal}] \leq \pi_1. \tag{7.5}$$

Likewise, G^{real} is π_2-indistinguishable from G^{ideal}. If the G protocol might potentially be paired with a perfect QKD Q^{ideal}, we can use the same Γ to differentiate between the combinations $Q^{ideal} \circ G^{real}$ and $Q^{ideal} \circ G^{ideal}$. This brings us to the following conclusion:

$$Pr[Q^{ideal} \circ G^{real}] - Pr[Q^{ideal} \circ G^{ideal}] \leq \pi_2. \tag{7.6}$$

Combining equations 7.5 and 7.6, we obtain:

$$Pr[Q^{real} \circ G^{real}] - Pr[Q^{ideal} \circ G^{ideal}] \leq \pi_1 + \pi_2. \tag{7.7}$$

Equation 7.7 contradicts equation 7.4, which proves the composability of the hybrid protocol. □

7.0.4 Re-keying

In group communication, a single key, TEK, is used to encrypt messages and is known to all members of the group. When a membership change occurs, such as the joining or leaving of a member, it is necessary to renew the key to provide both backward- and forward secrecy. This is because the key renewal ensures that any information related to the group's TEK and secret vector, A_i, that was known to the changed member, m_j, is updated. Upon a membership change, all group members, n, should be notified to replace the old TEK with the new version, $TEK^{'}$, while only the subset representative, V_i, should be notified of the updated secret vector,$A_i^{'}$. The representative then distributes the updated secret vector, $A_i^{'}$, to the legitimate members using their shared data encryption keys, $SDEKs$.

7.0.5 Grouping

Grouping refers to the process of dividing members into smaller subsets based on shared characteristics. In the context of this work, members are grouped based on their geographical location. This facilitates efficient organization and management of the group and reduces the number of required intermediate nodes.

A group representative is a device that is responsible for managing the group of members within its proximity. It is placed within a distance R of all the members in its group and serves as a central point of communication for these members. Group representatives

are used to facilitate communication between members and to ensure the security of the group's key exchange process.

Intermediate nodes are devices that are placed at locations where distance extension is required in order to meet the QKD distance constraint. They are placed within a distance R of the nearest group representative and serve as a relay point for communication between nodes that are too far apart to communicate directly. The precise locations and the number of intermediate nodes are determined during the planning phase and are deployed as needed to facilitate communication between members.

7.1 Performance

In this evaluation section, a comparison is made between the performance of the proposed hybrid protocol and that of KeyShield2. As previously mentioned, the initial stage of the two protocols differs, with KeyShield2 using a post-quantum algorithm, Kyber, to establish secure channels for generating symmetric keys with each group of members, while the hybrid approach uses DGL22, a quantum-based protocol, to generate symmetric keys with each group of members. This difference in the key establishment mechanism makes the hybrid protocol slower than KeyShield2 in the initial stage. This is due to the distance limitations of the QKD technology used in the DGL22, which can be affected by the transmission loss of the quantum signal over long distances and the use of intermediate nodes.

However, in the distribution stage, both the hybrid and KeyShield2 protocols solve the linear system and broadcast a secure lock in a single message. This makes them similar in terms of performance in the distribution stage.

Overall, while the hybrid protocol may be slower than KeyShield2 in the initial stage, it provides a more flexible and scalable solution during the transition period from classical to quantum systems. By combining the strengths of both quantum-safe and quantum-

based protocols, it allows for secure key establishment over short and long distances. The use of symmetric encryption allows for secure communication even in the presence of a quantum computer and makes it a suitable choice for the hybrid protocol. Further research and innovation in this direction are needed to continue to improve the performance and security of the proposed hybrid protocol.

7.2 Concluding Remarks

In conclusion, in this chapter, I have presented a hybrid protocol for a secure key establishment that combines the strengths of both quantum-safe and quantum-based protocols. The hybrid protocol uses DGL22, an RDI-QKD protocol, for establishing pairwise secure channels and generating symmetric keys, $SDEKs$, for secure communication. On the other hand, KeyShield2, a quantum-safe protocol, is used to broadcast a group traffic data encryption key, TEK. Both protocols operate independently and do not interact with one another. They are used for different purposes, with DGL22 being used for pairwise key establishment and KeyShield2 being used for group key establishment.

The hybrid protocol provides a way to combine the benefits of QKD and quantum-safe key distribution, which can help to overcome the limitations of each individual approach. For example, QKD technology is currently limited to relatively short distances, while quantum-safe protocols in the literature may not be secure or scalable. The hybrid protocol can use QKD technology to establish secure keys over short distances while relying on quantum-safe protocols for key establishment over longer distances. This allows for a more flexible and scalable solution for secure key establishment during the transition period from classical to quantum systems.

The use of symmetric encryption serves as a bridge between the two protocols, as the keying materials required by KeyShield2 are encrypted using the symmetric keys generated by DGL22. Symmetric encryption is resistant to attacks by quantum computers, with the

level of resistance increasing with longer key lengths. This makes it a suitable choice for the hybrid protocol as it allows for secure communication even in the presence of a quantum computer.

In terms of performance, the hybrid protocol is similar to KeyShield2 with minimal differences. The initial stage of both protocols differs, with KeyShield2 using a classical key KEM algorithm (Kyber) to establish secure channels for generating symmetric keys with each group of members, while the hybrid approach uses DGL22 to generate symmetric keys with each group of members. This makes the hybrid protocol slower than KeyShield2 in the initial stage due to the distance limitations and the use of intermediate nodes. However, in the distribution stage, both the hybrid and KeyShield2 protocols solve the linear system and broadcast a secure lock in a single message, making them similar in this regard.

In summary, the proposed hybrid protocol presents a new approach to secure key establishment that combines the strengths of both quantum-safe and quantum-based protocols. It provides a more flexible and scalable solution for secure key establishment during the transition period from classical to quantum systems. The use of symmetric encryption allows for secure communication even in the presence of a quantum computer and makes it a suitable choice for the hybrid protocol. Further research and innovation in this direction are needed to continue to improve the performance and security of the proposed hybrid protocol.

Chapter 8

Conclusion

The focus of this thesis is to explore methods of securely delivering encryption keys to authorized parties, in order to mitigate the risk of unauthorized access to sensitive information. Through an in-depth examination of current techniques and the development of new solutions, this thesis aims to provide a comprehensive understanding of the challenges and opportunities in securing encryption key distribution.

We have investigated two main categories of key establishment protocols: two-members key establishment protocols and group key establishment protocols. I have investigated three main subcategories for each category: classical solutions, quantum-safe solutions, and quantum-based solutions. Quantum computing poses threat to classical solutions which can be overcome with quantum-safe or quantum-based solutions. However, both quantum-safe or quantum-based solutions have their own unique set of limitations. I have organized and summarized their limitations and identified the research gaps.

Quantum-safe solutions are not secure by nature that a new mathematical breakthrough or the development of a more powerful quantum computer could potentially make it easier to solve the underlying mathematical problem. The limitations of the quantum-safe group key establishment include inefficiency as the number of group members increases, vulnerability to certain attacks, lack of attention to key metadata management, and reliance

on multiple trusted entities. Therefore, it is important to continue researching and developing new quantum-safe schemes as the field of cryptography and quantum computing evolves. Quantum-based solutions use the principles of quantum mechanics to provide unconditional security for key distribution. However, they have limitations in terms of range, scalability, cost, and vulnerability to attacks. Group-based QKD protocols have been developed to address point-to-multipoint communication limitations, but they also have their own limitations, such as inefficiency in large groups. Our analysis suggests that utilizing multiple instances of two-member QKD, rather than a single group-based protocol, can significantly enhance security and efficiency for a sufficient number of members above a certain threshold.

We have proposed two novel quantum-safe group key establishment protocols, KeyShield and KeyShield2. KeyShield balances various factors, including forward/backward secrecy, minimal trust, guessing resistance, quantum resistance, scalability, computational cost, storage overhead, messages overhead, and entity management. It addresses the high computational complexity of using a linear system of equations as a key establishment protocol through member grouping and the use of a banded matrix. KeyShield2 is an improved version that addresses vulnerabilities found in the original protocol without negatively impacting performance or efficiency.

Both protocols rely on the difficulty of finding a unique solution to an underdetermined linear system of equations for key establishment. They are scalable and offer a high level of security by using the impossibility of finding a unique solution. The initial keying materials that each member should have been shared in the initial stage through pairwise secure channels established using a post-quantum algorithm, Kyber. After that, the group key TEK is established using a single broadcast message (called secure lock) in an open channel, instead of establishing pairwise secure channels. Each member uses their keying materials to open the secure lock and obtain the shared key TEK.

We have proposed two Receiver-Device-Independent QKD protocols, namely, QKeyShield

and DGL22. Both protocols utilize the principles of quantum mechanics to establish a shared secret key and have been specifically designed to resist common attacks. Additionally, both protocols utilize a biased random-based approach in selecting unitary operators, specifically the Hadamard and identity operators, to enhance the security and efficiency of the key generation process. They are relatively easy to implement using current state-of-the-art technology. Several measures have been integrated to protect the sent qubit. Furthermore, security proofs and analyses have been provided showing that the proposed protocols can securely and effectively allow Alice and Bob to agree on a secret key.

Despite the similarities between QKeyShield and DGL22, the latter includes additional measures to further enhance its security and efficiency, such as isolating Bob's SQM measurement device to prevent any potential side-channel attacks, operating in time reversal mode, and using teleportation to improve the efficiency of the key generation process. They have distinct differences between them in terms of their modes of operation, entanglement swapping, teleportation, initial state preparation, measurement devices, and the total number of qubits and detectors required. Overall, DGL22's distinct features make it a more robust and efficient RDI-QKD protocol compared to QKeyShield.

Finally, a highly flexible and efficient hybrid group key establishment protocol that seamlessly integrates the advantages of both quantum-safe and quantum-based protocols has been proposed in this thesis. The proposed hybrid protocol utilizes DGL22, a state-of-the-art RDI-QKD protocol, to establish pairwise secure channels and generate symmetric keys, while leveraging KeyShield2, a robust quantum-safe protocol, for secure group key establishment. Both protocols operate independently and do not interact with one another, with DGL22 being utilized for pairwise key establishment and KeyShield2 for group key establishment.

This hybrid approach provides a flexible and scalable solution for secure key establishment during the transition period from classical to quantum systems, by utilizing the strengths of both quantum-safe and quantum-based protocols. The integration of DGL22,

an RDI-QKD protocol, for pairwise secure channels and symmetric key generation, and KeyShield2, a quantum-safe protocol, for group key establishment allows for a comprehensive and dynamic approach to secure key establishment. The use of QKD technology for pairwise key establishment over short distances and quantum-safe protocols for key establishment over longer distances addresses the limitations of individual approaches, making it a more versatile and practical solution. Additionally, the use of symmetric encryption serves as a bridge between the two protocols, ensuring secure communication even in the presence of a quantum computer, making the proposed hybrid protocol a secure and reliable choice for key establishment in a quantum-safe environment. Furthermore, the performance of the hybrid protocol is similar to KeyShield2, with minimal differences in the initial stage. The hybrid protocol is slower in the initial stage due to the distance limitations and the use of intermediate nodes, but it is similar to KeyShield2 in the distribution stage.

The proposed protocols are expected to be highly effective in securely establishing keys for real-world scenarios. It is anticipated that these protocols will gain widespread adoption and play a significant role in ensuring secure information transmission.

8.1 Future Work

The proposed protocols offer new avenues for the development of effective key establishment protocols. There are various research directions that can be explored to advance the current state of knowledge, including:

- Utilizing the difficulty of finding a unique solution to an underdetermined linear system of equations, as shown in the proposed quantum-safe protocols, and conducting further research and innovation in this area.
- Working towards submitting KeyShield2 for consideration as a future post-quantum group key establishment solution.

- Exploiting the challenge of discovering a single solution to an underdetermined linear system of equations, as demonstrated in proposed quantum-secure protocols, to transmit encrypted messages among group members instead of establishing keys.
- Investigating the use of quantum error-correcting codes to improve the robustness of quantum-based protocols against noise and other forms of interference.
- Exploring the utilization of various types of quantum states, including mixed states and coherent states, in quantum-based protocols has the potential to advance the development of more sophisticated quantum protocols.
- Investigating the use of satellite-based and free-space quantum communication, which could enable QKD over longer distances than is currently possible with fiber-optic systems.
- Developing more advanced modulation and multiplexing techniques to increase the amount of quantum information that can be transmitted over a given distance.
- Enhancing security and efficiency of hybrid key establishment protocols: The proposed hybrid protocol provides a flexible solution but can still be improved in terms of security and efficiency. Research could focus on finding new ways further to enhance the security and efficiency of the hybrid protocol.
- Investigating key metadata management: The limitations of quantum-safe group key establishment include a lack of attention to key metadata management, so future research could focus on addressing this issue to ensure that the management of encryption keys is secure and efficient.
- Evaluating the performance of the proposed protocols in real-world scenarios. Future research could focus on evaluating their performance in real-world scenarios to see how they perform in practical situations.

The primary objective of this thesis endeavor is to stimulate advanced investigations and facilitate significant advancements within the domain of secure key establishment techniques.

Chapter 9

Publications

1. Al-darwbi, M. Y., Ghorbani, A. A., Lashkari, A. H. (2020). **KeyShield**: A Scalable and Quantum-Safe Key Management Scheme. IEEE Open Journal of the Communications Society, 2, 87-101. (**Published**)

2. Al-Darwbi, M. Y., Ghorbani, A. A., Lashkari, A. H. (2022). **QKeyShield**: A Practical Receiver-Device-Independent Entanglement-Swapping-Based Quantum Key Distribution. IEEE Access, 10, 107685-107702. (**Published**)

3. Al-darwbi, M., Ghorbani, A., Lashkari, A. (2022). **DGL22**: A practical Quantum key distribution. Submitted to "International Journal of Information Security

4. **KeyShield2**: Comments on"KeyShield: A Scalable and Quantum-Safe Key Management Scheme". (To be submitted to "Journal of Information Security and Applications")

5. A practical and scalable **Hybrid** quantum-based and quantum-safe group Key establishment. (To be submitted)

6. Classical, quantum-safe, and quantum-based key establishment schemes: a survey. (**On going**)

References

[1] *Quantum key distribution: Awesome or pointless?*, https://cdn.ttgtmedia.com/rms/pdf/RHUL_Cobourne_Final.pdf, 2011.

[2] *Post-quantum cryptography*, https://csrc.nist.gov/projects/post-quantum-cryptography, 2017.

[3] *Post-quantum cryptography standardization*, https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization, 2019.

[4] *The future is now: Spreading the word about post-quantum cryptography*, https://www.nist.gov/blogs/taking-measure/future-now-spreading-word-about-post-quantum-cryptography, 2020.

[5] Guillaume Adenier, Masanori Ohya, Noboru Watanabe, Irina Basieva, and Andrei Yu Khrennikov, *Double blinding-attack on entanglement-based quantum key distribution protocols*, AIP Conference Proceedings, vol. 1424, American Institute of Physics, 2012, pp. 9–16.

[6] Gordon B. Agnew, Thomas Beth, Ronald C. Mullin, and Scott A. Vanstone, *Arithmetic operations in gf (2 m)*, Journal of Cryptology **6** (1993), 3–13.

[7] Michael Ampatzis and Theodore Andronikos, *Qkd based on symmetric entangled bernstein-vazirani*, Entropy **23** (2021), no. 7, 870.

[8] Daniel Apon, Dana Dachman-Soled, Huijing Gong, and Jonathan Katz, *Constant-round group key exchange from the ring-lwe assumption*, Post-Quantum Cryptography: 10th International Conference, PQCrypto 2019, Chongqing, China, May 8–10, 2019 Revised Selected Papers 10, Springer, 2019, pp. 189–205.

[9] Daniel Apon and James Howe, *Attacks on nist pqc 3rd round candidates*, IACR Real World Crypto Symposium, 2021.

[10] Reza Azarderakhsh, Dieter Fishbein, and David Jao, *Efficient implementations of a quantum-resistant key-exchange protocol on embedded systems*, Center of Applied Cryptographic Research (CACR), Waterloo, Canada, Tech. Rep. CACR **20** (2014), 2014.

[11] Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev, *Practical supersingular isogeny group key agreement.*, IACR Cryptol. ePrint Arch. **2019** (2019), 330.

[12] David Balenson, D McGrew, and A Sherman, *Key management for large dynamic groups: One-way function trees and amortized initialization*, Mar **15** (1999), 1–14.

[13] Elaine Barker, *Nist special publication 800-57 part 1 revision 5*, NIST, Tech. Rep **16** (2020).

[14] Elaine Barker, Elaine Barker, William Burr, William Polk, Miles Smid, et al., *Recommendation for key management: Part 1: General*, National Institute of Standards and Technology, Technology Administration . . . , 2006.

[15] Elaine Barker, Lily Chen, Sharon Keller, Allen Roginsky, Apostol Vassilev, and Richard Davis, *Recommendation for pair-wise key-establishment schemes using*

discrete logarithm cryptography, Tech. report, National Institute of Standards and Technology, 2017.

[16] Elaine Barker, Lily Chen, Dustin Moody, et al., *Recommendation for pair-wise key-establishment schemes using integer factorization cryptography*, US Department of Commerce, National Institute of Standards and Technology, 2014.

[17] Klaus Becker and Uta Wille, *Communication complexity of group key distribution*, Proceedings of the 5th ACM Conference on Computer and Communications Security, 1998, pp. 1–6.

[18] CH BENNET, *Quantum cryptography: Public key distribution and coin tossing*, Proc. of IEEE Int. Conf. on Comp., Syst. and Signal Proc., Bangalore, India, Dec. 10-12, 1984, 1984.

[19] Charles H Bennett, Gilles Brassard, and N David Mermin, *Quantum cryptography without bell's theorem*, Physical review letters **68** (1992), no. 5, 557.

[20] Daniel J Bernstein, *Introduction to post-quantum cryptography*, Post-quantum cryptography, Springer, 2009, pp. 1–14.

[21] Eli Biham, Bruno Huttner, and Tal Mor, *Quantum cryptographic network based on quantum memories*, Physical Review A **54** (1996), no. 4, 2651.

[22] Simon Blake-Wilson, *Ansi x9.63 overview key agreement and key transport using elliptic curve cryptography*.

[23] Alberto Boaron, Boris Korzh, Raphael Houlmann, Gianluca Boso, Charles Ci Wen Lim, Anthony Martin, and Hugo Zbinden, *Detector-device-independent quantum key distribution: Security analysis and fast implementation*, Journal of Applied Physics **120** (2016), no. 6, 063101.

[24] Vaios Bolgouras, Christoforos Ntantogian, Emmanouil Panaousis, and Christos Xenakis, *Distributed key management in microgrids*, IEEE Transactions on Industrial Informatics (2019).

[25] Sougato Bose, Vlatko Vedral, and Peter L Knight, *Multiparticle generalization of entanglement swapping*, Physical Review A **57** (1998), no. 2, 822.

[26] Cyril Branciard, Eric G Cavalcanti, Stephen P Walborn, Valerio Scarani, and Howard M Wiseman, *One-sided device-independent quantum key distribution: Security, feasibility, and the connection with steering*, Physical Review A **85** (2012), no. 1, 010301.

[27] Samuel L Braunstein and Stefano Pirandola, *Side-channel-free quantum key distribution*, Physical review letters **108** (2012), no. 13, 130502.

[28] Carlos Bravo-Prieto, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J Coles, *Variational quantum linear solver: A hybrid algorithm for linear systems*, arXiv preprint arXiv:1909.05820 (2019).

[29] Otto Bretscher, *Linear algebra with applications*, Prentice Hall Eaglewood Cliffs, NJ, 1997.

[30] Abudhahir Buhari, Zuriati Ahmad Zukarnain, Shamala K Subramaniam, Hisham Zainuddin, and Suhairi Saharudin, *A simple post quantum scheme for higher key rate multiparty quantum key distribution*, International Journal of Network Security & Its Applications **4** (2012), no. 5, 1.

[31] Mike Burmester and Yvo Desmedt, *A secure and efficient conference key distribution system*, Workshop on the Theory and Application of of Cryptographic Techniques, Springer, 1994, pp. 275–286.

[32] Burmester, Mike and Desmedt, Yvo, *A secure and scalable group key exchange system*, Information Processing Letters **94** (2005), no. 3, 137–143.

[33] Francesco Buscemi, *All entangled quantum states are nonlocal*, Physical review letters **108** (2012), no. 20, 200401.

[34] Adan Cabello, *Multiparty key distribution and secret sharing based on entanglement swapping*, arXiv preprint quant-ph/0009025 (2000).

[35] Adán Cabello, *Quantum key distribution in the holevo limit*, Physical Review Letters **85** (2000), no. 26, 5635.

[36] Adan Cabello, *Quantum key distribution without alternative measurements*, Physical Review A **61** (2000), no. 5, 052312.

[37] Ad'an Cabello, *Addendum to "quantum key distribution without alternative measurements"*, Physical Review A **64** (2001), no. 2, 024301.

[38] X-D Cai, Christian Weedbrook, Z-E Su, M-C Chen, Mile Gu, M-J Zhu, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan, *Experimental quantum computing to solve systems of linear equations*, Physical review letters **110** (2013), no. 23, 230501.

[39] Wen-Fei Cao, Yi-Zheng Zhen, Yu-Lin Zheng, Li Li, Zeng-Bing Chen, Nai-Le Liu, and Kai Chen, *One-sided measurement-device-independent quantum key distribution*, Physical Review A **97** (2018), no. 1, 012313.

[40] P Casares and M Martin-Delgado, *A quantum ip predictor-corrector algorithm for linear programming*, arXiv preprint arXiv:1902.06749 (2019).

[41] David Castañeda Valle, Luis Fernando Quezada, and Shi-Hai Dong, *Bell-ghz measurement-device-independent quantum key distribution*, Annalen der Physik **533** (2021), no. 9, 2100116.

[42] Yagnik Chatterjee, *Entanglement based multi-party quantum key distribution*, (2020).

[43] Kai Chen and Hoi-Kwong Lo, *Multi-partite quantum cryptographic protocols with noisy ghz states*, arXiv preprint quant-ph/0404133 (2004).

[44] Andrew M Childs, Robin Kothari, and Rolando D Somma, *Quantum algorithm for systems of linear equations with exponentially improved dependence on precision*, SIAM Journal on Computing **46** (2017), no. 6, 1920–1950.

[45] Guang-huei Chiou and Wen-Tsuen Chen, *Secure broadcasting using the secure lock*, IEEE Transactions on Software Engineering **15** (1989), no. 8, 929–934.

[46] Rakyong Choi, Dongyeon Hong, and Kwangjo Kim, *Constant-round dynamic group key exchange from rlwe assumption*, Cryptology ePrint Archive (2020).

[47] Nayana Das and Goutam Paul, *Secure multi-party quantum conference and xor computation*, arXiv preprint arXiv:2101.05560 (2021).

[48] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E Stange, *Improved torsion-point attacks on sidh variants*, Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III 41, Springer, 2021, pp. 432–470.

[49] James W Demmel, *Applied numerical linear algebra*, SIAM, 1997.

[50] Jian Dong and Jianfu Teng, *Quantum key distribution protocol of mesh network structure based on n+ 1 epr pairs*, Journal of Systems Engineering and Electronics **21** (2010), no. 2, 334–338.

[51] Artur K Ekert, *Quantum cryptography based on bell's theorem*, Physical review letters **67** (1991), no. 6, 661.

[52] Michael Epping, Hermann Kampermann, Dagmar Bruß, et al., *Multi-partite entanglement can speed up quantum key distribution in networks*, New Journal of Physics **19** (2017), no. 9, 093012.

[53] José Ignacio Escribano Pablos and María Isabel González Vasco, *Secure post-quantum group key exchange: Implementing a solution based on kyber*, IET Communications (2023).

[54] José Ignacio Escribano Pablos, María Isabel González Vasco, Misael Enrique Marriaga, and Ángel Luis Pérez del Pozo, *Compiled constructions towards post-quantum group key exchange: A design from kyber*, Mathematics **8** (2020), no. 10, 1853.

[55] Tako Boris Fouotsa and Christophe Petit, *A new adaptive attack on sidh*, Topics in Cryptology–CT-RSA 2022: Cryptographers' Track at the RSA Conference 2022, Virtual Event, March 1–2, 2022, Proceedings, Springer, 2022, pp. 322–344.

[56] Atsushi Fujioka, Katsuyuki Takashima, and Kazuki Yoneyama, *One-round authenticated group key exchange from isogenies*, Provable Security: 13th International Conference, ProvSec 2019, Cairns, QLD, Australia, October 1–4, 2019, Proceedings 13, Springer, 2019, pp. 330–338.

[57] Satoshi Furukawa, Noboru Kunihiro, and Katsuyuki Takashima, *Multi-party key exchange protocols from supersingular isogenies*, 2018 International Symposium on Information Theory and Its Applications (ISITA), IEEE, 2018, pp. 208–212.

[58] Gao Gan, *Quantum key distribution scheme with high efficiency*, Communications in Theoretical Physics **51** (2009), no. 5, 820.

[59] Gan Gao, *Quantum key distribution by comparing bell states*, Optics communications **281** (2008), no. 4, 876–879.

[60] Hang Gao and Ri-Gui Zhou, *Multi-party quantum key agreement protocol based on g-like states and χ+ states*, International Journal of Theoretical Physics **61** (2022), no. 2, 1–13.

[61] C Gobby, aZL Yuan, and AJ Shields, *Quantum key distribution over 122 km of standard telecom fiber*, Applied Physics Letters **84** (2004), no. 19, 3762–3764.

[62] Allan Goff, *Quantum tic-tac-toe: A teaching metaphor for superposition in quantum mechanics*, American Journal of Physics **74** (2006), no. 11, 962–973.

[63] María Isabel González Vasco, Ángel L Pérez del Pozo, and Rainer Steinwandt, *Group key establishment in a quantum-future scenario*, Informatica **31** (2020), no. 4, 751–768.

[64] Daniel Gottesman, H-K Lo, Norbert Lutkenhaus, and John Preskill, *Security of quantum key distribution with imperfect devices*, International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings., IEEE, 2004, p. 136.

[65] Torbjörn Granlund, *Gnu multiple precision arithmetic library*, http://gmplib. org/ (2010).

[66] Federico Grasselli, Hermann Kampermann, and Dagmar Bruß, *Finite-key effects in multipartite quantum key distribution protocols*, New Journal of Physics **20** (2018), no. 11, 113014.

[67] Cheng Guo and Chin-Chen Chang, *An authenticated group key distribution protocol based on the generalized chinese remainder theorem*, International Journal of Communication Systems **27** (2014), no. 1, 126–134.

[68] Fenzhuo Guo, Tailin Liu, Qiaoyan Wen, and Fuchen Zhu, *Quantum key distribution based on entanglement swapping between two bell states*, International Journal of Quantum Information **4** (2006), no. 05, 769–779.

[69] Lein Harn, Chingfang Hsu, and Zhe Xia, *Lightweight group key distribution schemes based on pre-shared pairwise keys*, IET Communications **14** (2020), no. 13, 2162–2165.

[70] Hugh Harney and Carl Muckenhirn, *Group key management protocol (gkmp) architecture*, Tech. report, 1997.

[71] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd, *Quantum algorithm for linear systems of equations*, Physical review letters **103** (2009), no. 15, 150502.

[72] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman, *Ntru: A ring-based public key cryptosystem*, International Algorithmic Number Theory Symposium, Springer, 1998, pp. 267–288.

[73] Hector B Hougaard and Atsuko Miyaji, *Group key exchange compilers from generic key exchanges*, Network and System Security: 15th International Conference, NSS 2021, Tianjin, China, October 23, 2021, Proceedings 15, Springer, 2021, pp. 162–184.

[74] Xi Huang, Shi-Bin Zhang, Yan Chang, Chi Qiu, Dong-Mei Liu, and Min Hou, *Quantum key agreement protocol based on quantum search algorithm*, International Journal of Theoretical Physics **60** (2021), no. 3, 838–847.

[75] Hitoshi Inamori, *Security of practical time-reversed epr quantum key distribution*, Algorithmica **34** (2002), no. 4, 340–365.

[76] Marie Ioannou, Pavel Sekatski, Alastair A Abbott, Denis Rosset, Jean-Daniel Bancal, and Nicolas Brunner, *Receiver-device-independent quantum key distribution protocols*, New Journal of Physics **24** (2022), no. 6, 063006.

[77] Gregg Jaeger, *Quantum information*, Springer, 2007.

[78] Li-Zhen Jiang, *Semi-quantum private comparison based on bell states*, Quantum Information Processing **19** (2020), no. 6, 1–21.

[79] Burt Kaliski, *Key management and ansi x9.44*, 2000.

[80] Abdel Alim Kamal, *Cryptanalysis of a polynomial-based key management scheme for secure group communication.*, IJ Network Security **15** (2013), no. 1, 68–70.

[81] Alexandre Karlov and Natacha Linard de Guertechin, *Power analysis attack on kyber*, Cryptology ePrint Archive (2021).

[82] Sharon Keller, *Ansi x9.42 agreement of symmetric keys using discrete logarithm cryptography*, 2000.

[83] Yongdae Kim, Adrian Perrig, and Gene Tsudik, *Simple and fault-tolerant key agreement for dynamic collaborative groups*, Proceedings of the 7th ACM Conference on Computer and Communications Security, 2000, pp. 235–244.

[84] Yi-Hsuan Kung and Hsu-Chun Hsiao, *Groupit: Lightweight group key management for dynamic iot environments*, IEEE Internet of Things Journal **5** (2018), no. 6, 5155–5165.

[85] Chong Li, Zhen Wang, Chunfeng Wu, He-Shan Song, and Ling Zhou, *Certain quantum key distribution achieved by using bell states*, International Journal of Quantum Information **4** (2006), no. 06, 899–906.

[86] Mingyan Li, Radha Poovendran, and C Berenstein, *Design of secure multicast key management schemes with communication budget constraint*, IEEE communications Letters **6** (2002), no. 3, 108–110.

[87] Charles Ci Wen Lim, Christopher Portmann, Marco Tomamichel, Renato Renner, and Nicolas Gisin, *Device-independent quantum key distribution with local bell test*, Physical Review X **3** (2013), no. 3, 031006.

[88] Song Lin, Ying Sun, Xiao-Fen Liu, and Zhi-Qiang Yao, *Quantum private comparison protocol with d-dimensional bell states*, Quantum information processing **12** (2013), no. 1, 559–568.

[89] Bin Liu, Fei Gao, Wei Huang, and Qiao-yan Wen, *Multiparty quantum key agreement with single particles*, Quantum information processing **12** (2013), no. 4, 1797–1805.

[90] Hao-Nan Liu, Xiang-Qian Liang, Dong-Huan Jiang, Yong-Hua Zhang, and Guang-Bao Xu, *Multi-party quantum key agreement protocol with bell states and single particles*, International Journal of Theoretical Physics **58** (2019), no. 5, 1659–1666.

[91] Niu Liu, Shaohua Tang, and Lingling Xu, *Attacks and comments on several recently proposed key management schemes.*, IACR Cryptol. ePrint Arch. **2013** (2013), 100.

[92] Hoi-Kwong Lo, Marcos Curty, and Bing Qi, *Measurement-device-independent quantum key distribution*, Physical review letters **108** (2012), no. 13, 130503.

[93] Hoi-Kwong Lo, Marcos Curty, and Bing Qi, *Measurement-device-independent quantum key distribution.*, Physical review letters **108 13** (2012), 130503.

[94] Xixiang Lv, Hui Li, and Baocang Wang, *Group key agreement for secure group communication in dynamic peer systems*, Journal of Parallel and Distributed Computing **72** (2012), no. 10, 1195–1200.

[95] Vadim Lyubashevsky, Chris Peikert, and Oded Regev, *On ideal lattices and learning with errors over rings*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2010, pp. 1–23.

[96] Xiyuan Ma, Chaonan Wang, Zexi Li, and Hongfeng Zhu, *Multi-party quantum key distribution protocol with new bell states encoding mode*, International Journal of Theoretical Physics **60** (2021), no. 4, 1328–1338.

[97] Luciano Maino and Chloe Martindale, *An attack on sidh with arbitrary starting curve*, Cryptology ePrint Archive (2022).

[98] David McGrew and John Viega, *The galois/counter mode of operation (gcm)*, submission to NIST Modes of Operation Process **20** (2004), 0278–0070.

[99] Daniele Micciancio and Oded Regev, *Lattice-based cryptography in post quantum cryptography, bernstein dj, buchmann j., dahmen e*, 2009.

[100] Shi-Qi Min, Hua-Ying Chen, and Li-Hua Gong, *Novel multi-party quantum key agreement protocol with g-like states and bell states*, International Journal of Theoretical Physics **57** (2018), no. 6, 1811–1822.

[101] Ashley Montanaro, *Quantum algorithms: an overview*, npj Quantum Information **2** (2016), no. 1, 1–8.

[102] Sasa Mrdovic and Branislava Perunicic, *Kerckhoffs' principle for intrusion detection*, Networks 2008-The 13th International Telecommunications Network Strategy and Planning Symposium, IEEE, 2008, pp. 1–8.

[103] Jianan Mu, Yixuan Zhao, Zongyue Wang, Jing Ye, Junfeng Fan, Shuai Chen, Huawei Li, Xiaowei Li, and Yuan Cao, *A voltage template attack on the modular polynomial subtraction in kyber*, 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 2022, pp. 672–677.

[104] Vankamamidi S Naresh, Moustafa M Nasralla, Sivaranjani Reddi, and Iván García-Magariño, *Quantum diffie–hellman extended to dynamic quantum group key agreement for e-healthcare multi-agent systems in smart cities*, Sensors **20** (2020), no. 14, 3940.

[105] Vankamamidi Srinivasa Naresh and Reddi Sivaranjani, *Multiparty quantum key agreement with strong fairness property.*, Comput. Syst. Sci. Eng. **35** (2020), no. 6, 457–465.

[106] NIST, *Status report on the third round of the nist post-quantum cryptography standardization process*, Available:https://csrc.nist.gov/publications/detail/nistir/8413/final, 2022, [Online; accessed 28-07-2022].

[107] José Ignacio Escribano Pablos, Misael Enrique Marriaga, and Ángel L Pérez del Pozo, *Design and implementation of a post-quantum group authenticated key exchange protocol with the liboqs library: A comparative performance analysis from classic mceliece, kyber, ntru, and saber*, IEEE Access **10** (2022), 120951–120983.

[108] Mark Pecen et al., *Quantum safe cryptography and security: An introduction, benefits, enablers and challenges, white paper*, European Telecommunications Standards Institute (2014).

[109] Edoardo Persichetti, Rainer Steinwandt, and Adriana Suárez Corona, *From key encapsulation to authenticated group key establishment—a compiler for post-quantum primitives*, Entropy **21** (2019), no. 12, 1183.

[110] Peter Pessl and Robert Primas, *More practical single-trace attacks on the number theoretic transform*, International Conference on Cryptology and Information Security in Latin America, Springer, 2019, pp. 130–149.

[111] Peter Pessl and Lukas Prokop, *Fault attacks on cca-secure lattice kems*, IACR Transactions on Cryptographic Hardware and Embedded Systems (2021), 37–60.

[112] Yanji Piao, JongUk Kim, Usman Tariq, and Manpyo Hong, *Polynomial-based key management for secure intra-group and inter-group communication*, Computers & Mathematics with Applications **65** (2013), no. 9, 1300–1309.

[113] Stefano Pirandola, Ulrik L Andersen, Leonardo Banchi, Mario Berta, Darius Bunandar, Roger Colbeck, Dirk Englund, Tobias Gehring, Cosmo Lupo, Carlo Ot-

taviani, et al., *Advances in quantum cryptography*, Advances in optics and photonics **12** (2020), no. 4, 1012–1236.

[114] Eric Polizzi and Ahmed Sameh, *Spike: A parallel environment for solving banded linear systems*, Computers & Fluids **36** (2007), no. 1, 113–120.

[115] Christopher Portmann and Renato Renner, *Cryptographic security of quantum key distribution*, arXiv preprint arXiv:1409.3525 (2014).

[116] Massimiliano Proietti, Joseph Ho, Federico Grasselli, Peter Barrow, Mehul Malik, and Alessandro Fedrizzi, *Experimental quantum conference key agreement*, Science advances **7** (2021), no. 23, eabe0395.

[117] Huawang Qin, Weihua Sun, and Wallace KS Tang, *Quantum secure direct communication based on single particles*, Optical and Quantum Electronics **54** (2022), no. 8, 1–11.

[118] Lingyu Qin, Ying Li, and Shibo Wang, *A key distribution protocol based on quantum entanglement swapping for unmanned surface vehicle*, Journal of Physics: Conference Series, vol. 1976, IOP Publishing, 2021, p. 012034.

[119] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E Stange, *Improved torsion-point attacks on sidh variants*, Annual International Cryptology Conference, Springer, 2021, pp. 432–470.

[120] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadhyay, *Drop by drop you break the rock-exploiting generic vulnerabilities in lattice-based pke/kems using em-based physical attacks*, Cryptology ePrint Archive (2020).

[121] Prasanna Ravi, Debapriya Basu Roy, Shivam Bhasin, Anupam Chattopadhyay, and Debdeep Mukhopadhyay, *Number "not used" once-practical fault attack on pqm4*

implementations of nist candidates, International Workshop on Constructive Side-Channel Analysis and Secure Design, Springer, 2019, pp. 232–250.

[122] R Renner and R Konig, *Second theory of cryptography conference tcc*, Lecture Notes in Computer Science **3378** (2005), 407–425.

[123] Renato Renner, *Security of quantum key distribution*, International Journal of Quantum Information **6** (2008), no. 01, 1–127.

[124] H. P. Robertson, *The uncertainty principle*, Phys. Rev. **34** (1929), 163–164.

[125] Stefan Schauer and Martin Suda, *A novel attack strategy on entanglement swapping qkd protocols*, International Journal of Quantum Information **6** (2008), no. 04, 841–858.

[126] Debashis Sen, *The uncertainty relations in quantum mechanics*, Current Science (2014), 203–218.

[127] Adi Shamir, *How to share a secret*, Communications of the ACM **22** (1979), no. 11, 612–613.

[128] Akshata Shenoy-Hejamadi, Anirban Pathak, and Srikanth Radhakrishna, *Quantum cryptography: key distribution and beyond*, Quanta **6** (2017), no. 1, 1–47.

[129] Peter W Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM review **41** (1999), no. 2, 303–332.

[130] Peter W Shor and John Preskill, *Simple proof of security of the bb84 quantum key distribution protocol*, Physical review letters **85** (2000), no. 2, 441.

[131] Daegene Song, *Secure key distribution by swapping quantum entanglement*, Physical Review A **69** (2004), no. 3, 034301.

[132] Ronggong Song, Larry Korba, and George OM Yee, *A scalable group key management protocol*, IEEE communications letters **12** (2008), no. 7, 541–543.

[133] Michael Steiner, Gene Tsudik, and Michael Waidner, *Diffie-hellman key distribution extended to group communication*, Proceedings of the 3rd ACM Conference on Computer and Communications Security, 1996, pp. 31–37.

[134] Yiğit Subaşı, Rolando D Somma, and Davide Orsucci, *Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing*, Physical review letters **122** (2019), no. 6, 060504.

[135] Qi-Chao Sun, Ya-Li Mao, Yang-Fan Jiang, Qi Zhao, Si-Jing Chen, Wei Zhang, Wei-Jun Zhang, Xiao Jiang, Teng-Yun Chen, Li-Xing You, et al., *Entanglement swapping with independent sources over an optical-fiber network*, Physical Review A **95** (2017), no. 3, 032306.

[136] Katsuyuki Takashima, *Post-quantum constant-round group key exchange from static assumptions*, International Symposium on Mathematics, Quantum Theory, and Cryptography, 2021, p. 251.

[137] Shuntaro Takeda, Maria Fuwa, Peter van Loock, and Akira Furusawa, *Entanglement swapping between discrete and continuous variables*, Physical review letters **114** (2015), no. 10, 100501.

[138] Nobuyuki Takei, Hidehiro Yonezawa, Takao Aoki, and Akira Furusawa, *High-fidelity teleportation beyond the no-cloning limit and entanglement swapping for continuous variables*, Physical review letters **94** (2005), no. 22, 220502.

[139] Run-Hua Tang, Cai Zhang, Dong-Yang Long, and Pan Wu, *Improvements on "multi-party quantum key agreement protocol with bell states and single particles"*, International Journal of Theoretical Physics **59** (2020), no. 5, 1623–1637.

[140] Shaohua Tang, Jintai Ding, and Yujun Liang, *A simple and efficient new group key management approach based on linear geometry.*, IACR Cryptol. ePrint Arch. **2011** (2011), 225.

[141] Yan-Lin Tang, Hua-Lei Yin, Si-Jing Chen, Yang Liu, Wei-Jun Zhang, Xiao Jiang, Lu Zhang, Jian Wang, Li-Xing You, Jian-Yu Guan, et al., *Field test of measurement-device-independent quantum key distribution*, IEEE Journal of Selected Topics in Quantum Electronics **21** (2014), no. 3, 116–122.

[142] Yan-Lin Tang, Hua-Lei Yin, Qi Zhao, Hui Liu, Xiang-Xiang Sun, Ming-Qi Huang, Wei-Jun Zhang, Si-Jing Chen, Lu Zhang, Li-Xing You, et al., *Measurement-device-independent quantum key distribution over untrustful metropolitan network*, Physical Review X **6** (2016), no. 1, 011024.

[143] Zhiyuan Tang, Kejin Wei, Olinka Bedroya, Li Qian, and Hoi-Kwong Lo, *Experimental measurement-device-independent quantum key distribution with imperfect sources*, Physical Review A **93** (2016), no. 4, 042308.

[144] Kishore Thapliyal, Rishi Dutt Sharma, and Anirban Pathak, *Orthogonal-state-based and semi-quantum protocols for quantum private comparison in noisy environment*, International Journal of Quantum Information **16** (2018), no. 05, 1850047.

[145] Marco Tomamichel, Charles Ci Wen Lim, Nicolas Gisin, and Renato Renner, *Tight finite-key analysis for quantum cryptography*, Nature communications **3** (2012), no. 1, 1–6.

[146] Marco Tomamichel and Renato Renner, *Uncertainty relation for smooth entropies*, Physical review letters **106** (2011), no. 11, 110506.

[147] Pandi Vijayakumar, Maria Azees, Arputharaj Kannan, and Lazarus Jegatha Deborah, *Dual authentication and key management techniques for secure data transmis-*

sion in vehicular ad hoc networks, IEEE Transactions on Intelligent Transportation Systems **17** (2015), no. 4, 1015–1028.

[148] Zhengchao Wei, Weilong Wang, Zhen Zhang, Ming Gao, Zhi Ma, and Xiongfeng Ma, *Decoy-state quantum key distribution with biased basis choice*, Scientific reports **3** (2013), no. 1, 1–4.

[149] Morgan M Weston, Sergei Slussarenko, Helen M Chrzanowski, Sabine Wollmann, Lynden K Shalm, Varun B Verma, Michael S Allman, Sae Woo Nam, and Geoff J Pryde, *Heralded quantum steering over a high-loss channel*, Science advances **4** (2018), no. 1, e1701230.

[150] Chung Kei Wong, Mohamed Gouda, and Simon S Lam, *Secure group communications using key graphs*, IEEE/ACM transactions on networking **8** (2000), no. 1, 16–30.

[151] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash, *Quantum linear system algorithm for dense matrices*, Physical review letters **120** (2018), no. 5, 050502.

[152] Yadong Wu, Jian Zhou, Xinbao Gong, Ying Guo, Zhi-Ming Zhang, and Guangqiang He, *Continuous-variable measurement-device-independent multipartite quantum communication*, Physical Review A **93** (2016), no. 2, 022325.

[153] Yiting Wu, Hong Chang, Gongde Guo, and Song Lin, *Multi-party quantum key agreement protocol with authentication*, arXiv preprint arXiv:2103.12252 (2021).

[154] Jun Xin, Xiao-Ming Lu, Xingmin Li, and Guolong Li, *One-sided device-independent quantum key distribution for two independent parties*, Optics Express **28** (2020), no. 8, 11439–11450.

[155] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan, *Variational algorithms for linear algebra*, arXiv preprint arXiv:1909.03898 (2019).

[156] Zhuang Xu, Owen Michael Pemberton, Sujoy Sinha Roy, David Oswald, Wang Yao, and Zhiming Zheng, *Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber*, IEEE Transactions on Computers (2021).

[157] Lili Yan, Shibin Zhang, Yan Chang, Guogen Wan, and Fan Yang, *Semi-quantum private comparison protocol with three-particle g-like states*, Quantum Information Processing **20** (2021), no. 1, 1–16.

[158] Yu-Guang Yang, Xin-Long Lv, Shang Gao, Yi-Hua Zhou, and Wei-Min Shi, *Detector-device-independent quantum key agreement based on single-photon bell state measurement*, International Journal of Theoretical Physics **61** (2022), no. 2, 1–16.

[159] Tian-Yu Ye and Chong-Qiang Ye, *Measure-resend semi-quantum private comparison without entanglement*, International Journal of Theoretical Physics **57** (2018), no. 12, 3819–3834.

[160] Chao-Hua Yu, Gong-De Guo, and Song Lin, *Quantum private comparison with d-level single-particle states*, Physica Scripta **88** (2013), no. 6, 065013.

[161] Hao Yuan, Jun Song, Lian-fang Han, Kui Hou, and Shou-hua Shi, *Improving the total efficiency of quantum key distribution by comparing bell states*, Optics communications **281** (2008), no. 18, 4803–4806.

[162] Chang-Yue Zhang and Zhu-Jun Zheng, *Entanglement-based quantum key distribution with untrusted third party*, Quantum Information Processing **20** (2021), no. 4, 1–20.

[163] Yong-Sheng Zhang, Chuan-Feng Li, and Guang-Can Guo, *Comment on "quantum key distribution without alternative measurements"*, Physical Review A **63** (2001), no. 3, 036301.

[164] Xing-Qiang Zhao, Nan-Run Zhou, Hua-Ying Chen, and Li-Hua Gong, *Multiparty quantum key agreement protocol with entanglement swapping*, International Journal of Theoretical Physics **58** (2019), no. 2, 436–450.

[165] Xinliang Zheng, Chin-Tser Huang, and Manton Matthews, *Chinese remainder theorem based group key management*, Proceedings of the 45th annual southeast regional conference, 2007, pp. 266–271.

[166] Nanrun Zhou, Lijun Wang, Lihua Gong, Xiangwu Zuo, and Ye Liu, *Quantum deterministic key distribution protocols based on teleportation and entanglement swapping*, Optics communications **284** (2011), no. 19, 4836–4842.

[167] Hongfeng Zhu, Zexi Li, Xueying Wang, and Liuyi Chen, *Multi-party quantum key agreement protocol for smart home environment*, International Journal of Theoretical Physics **60** (2021), no. 10, 3948–3960.

Mohammed Al-Darwbi, Ph.D.
Fredericton, NB, Canada

Education

Ph.D in Computer ScienceCanadian Institute for Cybersecurity-Faculty of Computer Science-University of New Brunswick,Sep-2018 – June-2023,Fredericton, NB, Canada

- Specialized in cybersecurity with focus on key establishment mechanisms

M.Sc. in Computer NetworksComputer Engineering department-King Fahd University of Petroleum and Minerals,Jan-2014 –Dec-2017,Dhahran, Saudi Arabia

- Conducted advanced research in Computer Networks during my Master's studies.

B.Sc in Computer ScienceFaculty of Computer Science and Information System- Dhamar University,Sep-2004 –May-2008,Dhamar, Yemen

Work Experiance

Post-Quantum Cryptography Researcher(Team Leader) Scotiabank - Canadian Institute for Cybersecurity collaboration project - University of New Brunswick ,Dec-2022–ongoing,Fredericton-NB, Canada

- Demonstrated leadership in assessing Quantum Readiness of Cryptography Standards, developing post-quantum migration protocols, and conducting Quantum Risk Assessment for the financial sector.

Post-Quantum Cryptography Researcher (Team Co-leader) Scotiabank - Canadian Institute for Cybersecurity collaboration project - University of New Brunswick ,Dec-2021–Nov-2022,Fredericton-NB, Canada

- Assisted in team coordination, roadmap development for quantum threat mitigation, and directed framework creation for post-quantum solutions evaluation.

[-1ex]page2sidebar

Cyber Kill-Chain Researcher (Team Leader) Scotiabank - Canadian Institute for Cybersecurity collaboration project - University of New Brunswick ,Apr-2021–Dec-2021,Fredericton-NB, Canada

- Coordinated a team to develop a Machine Learning-based scheme for cyber kill-chain assessment and prediction, alongside supervising the data collection process.

System Vulnerabilities Remediation Prioritization Researcher Scotiabank - Canadian Institute for Cybersecurity collaboration project - University of New Brunswick ,Nov-2020–Apr-2021,Fredericton-NB, Canada

- Developed a Random Forests-based scheme for risk-based vulnerabilities remediation prioritization and programmed a tool for optimized extraction of exploits from various databases.

Cyber Threat Intelligence Researcher Scotiabank - Canadian Institute for Cybersecurity collaboration project - University of New Brunswick ,Dec-2019–Nov-2020,Fredericton-NB, Canada

- Completed an in-depth study of cyber threat intelligence and developed a consumer-oriented evaluation scheme for Cyber Threat Intelligence providers.

Operational Technology Security Researcher A Siemens - Canadian Institute for Cybersecurity collaboration project - University of New Brunswick ,Sep-2018–Nov-2019, Fredericton-NB, Canada

comprehensive surveys and timelines of industrial cybersecurity attacks, and developed a Python-based Network Forensics Tool, NetFoT.

Software developer A software development Team Leader at Dadiah General Trading. 2011–2013Thamar, Yemen

- Coordinated team of five to build an International Trade Management System.

Research Projects
Lecturer Teaching Assistance - Thamar University, Yemen. 2008–2013,Thamar, Yemen

- Teaching labs and supervising students in their senior projects
- Assisting faculty with organizing exams and scheduling

Thesis Related Publications

1. Al-darwbi, M. Y., Ghorbani, A. A., Lashkari, A. H. (2020). **KeyShield**: A Scalable and Quantum-Safe Key Management Scheme. IEEE Open Journal of the Communications Society, 2, 87-101. (**Published**)
2. Al-Darwbi, M. Y., Ghorbani, A. A., Lashkari, A. H. (2022). **QKeyShield**: A Practical Receiver-Device-Independent Entanglement-Swapping-Based Quantum Key Distribution. IEEE Access, 10, 107685-107702. (**Published**)
3. Al-darwbi, M., Ghorbani, A., Lashkari, A. (2022). **DGL22**: A practical Quantum key distribution. Submitted to "International Journal of Information Security
4. **KeyShield2**: Comments on"KeyShield: A Scalable and Quantum-Safe Key Management Scheme". (To be submitted to "Journal of Information Security and Applications")
5. A practical and scalable **Hybrid** quantum-based and quantum-safe group Key establishment. (To be submitted)
6. Classical, quantum-safe, and quantum-based key establishment schemes: a survey. (**On going**)