

Improved Keyrates for Quantum Key Distribution from two-way Classical Communication

by

Devashish Tupkary

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Physics (Quantum Information)

Waterloo, Ontario, Canada, 2022

© Devashish Tupkary 2022

Author's Declaration

This thesis consists of material all of which I authored or co-authored : see Statement of Contributions included in this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

All results in this thesis are effectively my own. Beyond the guidance and advisement of my advisor, Norbert Lütkenhaus, I would like to acknowledge contributions by Ernest Tan to the technical parts of Sec. [4.1.3](#). Those proofs emerged during an active discussion with him.

Abstract

Quantum Key Distribution (QKD) promises information-theoretic security of secret keys. An important metric for QKD protocols is the asymptotic keyrate. We propose and analyze several new classical processing protocols for QKD, that improve the maximum asymptotic keyrate for qubit-based BB84 and six-state protocols. These protocols only modify the classical stage of Quantum Key Distribution protocols, and hence can be implemented with relative ease. These protocols extend and generalize the work of [39]. We also compute keyrates for the protocol in [19], which until now had only been analyzed for individual attacks. Some of the protocols we study do not implement IID blockwise processing. We provide a rigorous analysis of their keyrate which is novel and interesting.

Acknowledgements

First and foremost, I would like to thank my advisor, Norbert Lütkenhaus for his continuous support, guidance, time and general positive attitude. I am very grateful to him for giving me the opportunity to do a Master's degree in his group.

This thesis greatly benefited from discussions with Thomas van Himbeeck, particularly during the time when in-person meetings were rare. I would like to thank him for taking an interest in this topic. I am grateful to Ernest Tan for teaching me most of what I know about smoothed min-entropies. Without him, parts of this thesis would have been far more cumbersome to write. I found discussions with the members of the Optical Quantum Communications Theory group to be extremely valuable. I thank all its members for being amazing individuals to work with. In particular, I benefited greatly from discussions with Shlok Nahar, Lars Kamin and John Burniston.

I would like to thank Shlok Nahar, Ernest Tan, Sayan Gangopadhyay and Norbert Lütkenhaus for providing valuable feedback on early drafts of this thesis.

I was lucky to be constantly surrounded by some amazing and wonderful people in Waterloo. I thank the residents of 66 Albert Street, Brett, Sayan, Pritam and Kartik for making the house feel like a home. Finally, I thank my sister Kalyani and my parents for being an irreplaceable source of love and support.

Table of Contents

List of Figures	ix
1 Introduction	1
2 Background	4
2.1 Quantum Information Theory	4
2.1.1 Hilbert Spaces	4
2.1.2 Linear Operator Theory	5
2.1.3 Quantum Mechanics	7
2.1.4 Measures and Entropies	8
2.1.5 Miscellaneous	9
2.2 Quantum Key Distribution	10
2.2.1 A generic QKD protocol	10
2.2.2 Source Replacement Scheme	11
2.2.3 Security framework	12
2.2.4 BB84 and Six-state protocols	14
3 Protocols for Classical Processing in QKD - 1	16
3.1 Preliminaries	16
3.2 Compressed Partition Announcements (CPA)	19
3.2.1 Protocol description	19

3.2.2	Keyrate calculations for protocol with IID blockwise processing . .	20
3.2.3	Keyrate analysis	22
3.2.4	Results	23
3.3	Sequential Compressed Partition Announcements (SCPA)	26
3.3.1	Protocol description	27
3.3.2	Keyrate analysis	28
3.3.3	Results	29
4	Protocols for Classical Processing in QKD - 2	31
4.1	Recursive Protocol	31
4.1.1	Protocol description	32
4.1.2	IID Structure in the Recursive Protocol	33
4.1.3	Technical tools	35
4.1.4	Keyrate analysis	38
4.1.5	Choosing l_i	41
4.1.6	Final Keyrate expression	42
4.1.7	Results	43
4.2	AD+CPA	45
4.2.1	Protocol Description	45
4.2.2	Keyrate Analysis	46
4.2.3	Results	49
5	Conclusion	51
	References	52
	APPENDICES	56
A	Symmetry : Reduction to Bell-Diagonal states	57

B	Linear Algebra	64
C	Numerical details for CPA keyrate calculations	67
C.1	Choosing a linear basis	67
C.2	Computing Cost of Error Correction	68
C.2.1	Cost of correcting Partitions	69
C.2.2	Cost of correcting key	69
C.3	Computing Eve's Information on the Key	69
C.3.1	When partitions match	70
C.3.2	When partitions don't match	71
C.4	Calculations for AD partitions	73
D	Numerical details for SCPA keyrate calculations	75
D.1	Choosing a linear basis	75
D.2	Computing probability of rounds	76
D.3	Computing Cost of Error Correction	77
D.4	Computing Eve's Information on the Key	77
E	Numerical details for Recursive keyrate calculations	79
E.1	Eve's information on the key	79
E.1.1	Terminal round	82
F	Numerical details for AD+CPA keyrate calculations	83
F.1	AD Round	83
F.2	CPA Round	84
F.2.1	Eve's information when parities match	84
F.2.2	Eve's information when parities don't match	86

List of Figures

3.1	An example of our CPA protocol for $H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$	20
3.2	Six-state keyrate for the CPA protocol with $H = [1, 1, 1]$. We find that the CPA keyrate is higher than both our benchmarks for some range of QBER.	25
3.3	BB84 keyrate for the CPA protocol with $H = [1, 1, 1]$. We find that the CPA keyrate is higher than both our benchmarks for some range of QBER.	25
3.4	BB84 keyrate for the CPA protocol with $H = [1, 1, 1, 1]$. We find that the CPA keyrate is higher than both our benchmarks for some range of QBER.	26
3.5	An example of our SCPA protocol for $H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$	27
3.6	Six-state keyrate for the SCPA protocol with $H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. We find that the SCPA keyrate is higher than both our benchmarks for some range of QBER.	30
3.7	BB84 keyrate for the SCPA protocol with $H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. We find that the SCPA keyrate is higher than both our benchmarks for some range of QBER.	30
4.1	An example of the Recursive protocol for $t = 3$	32
4.2	Six-state keyrate for the recursive protocol with 4 rounds. We find improvements over our benchmarks for a large range of QBER.	44
4.3	BB84 keyrate for the recursive protocol with 4 rounds. We find improvements over our benchmarks for a large range of QBER.	44

4.4	An example of the AD+CPA protocol	46
4.5	Six-state keyrate for the AD+CPA protocol with $b = 2$. We find improvements to our benchmarks over a wide range of QBER.	50
4.6	BB84 keyrate for the AD+CPA protocol with $b = 2$. We find improvements to our benchmarks over a wide range of QBER.	50
E.1	Schematic of recursive protocol	80

Chapter 1

Introduction

Quantum key Distribution (QKD) [3, 13] is a cryptographic protocol that enables two parties, Alice and Bob to share an information-theoretically secure key using a completely insecure quantum channel, and an authenticated classical channel. Unlike classical key sharing protocols, whose security depend upon the computation assumptions on the adversary Eve, and/or on the computational hardness of certain mathematical problems, keys generated via QKD protocols can be shown to be information-theoretically secure [30, 28, 36, 20]. The proof for QKD protocol security is based only upon the properties of quantum mechanics, and assumptions on the functioning of local devices. This is not the only reason why QKD protocols are relevant today. In the case of classical key-sharing protocols, all the transcripts and communication can be copied and stored indefinitely until faster computers, or mathematical breakthroughs in algorithms allow future technology to break the encryption. Thus, classical protocols cannot guarantee security against future advancements in cryptanalysis. In contrast, keys generated using QKD are as secure in the future as they are at the moment of the generation. This property of QKD makes it an attractive protocol to have in our cryptographic arsenal.

QKD protocols typically involve Alice preparing some signal states and sending them to Bob, who performs measurements on them. After this stage, both Alice and Bob hold classical data pertaining to the states sent and measurement outcomes obtained respectively. In the parameter estimation step, they publicly announce a fraction of the data to determine whether or not to abort or proceed with the protocol. If they chose to continue with the protocol, Alice and Bob now possess classical data that they wish to transform into a shared secret key. In general, they may perform any sequence of operations to do this job. It is this classical protocol, which Alice and Bob implement to transform their classical data to a secret key, that is the topic of study in this thesis.

The asymptotic secret key rate, i.e the fraction of secret bits generated per signal sent, is an important metric for the analysis of QKD protocols. Many techniques such as advantage distillation [28, 22, 33], or noisy preprocessing [28], or others [16, 32, 15, 38, 25] have been proposed over the years to improve QKD keyrates. However, there is still a gap between the maximum known keyrates for these protocols, and the known upper bounds to keyrates [23, 24, 8, 27]. While noisy processing does provide small improvement in keyrates, advantage distillation typically yields very low keyrates (although it enables key generation at high QBER). Thus, it is natural to ask whether further improvements are possible.

In this thesis we tackle this question by inventing new classical protocols that improve upon the best known asymptotic key rates for qubit-based BB84 [3] and six-state [6] protocols. Our protocols involve two-way classical communication [1, 17, 2]. Since we propose to change only the classical stage of the protocol, keeping the signal states and measurements the same, our protocols are relatively easy to implement. Our main contribution is using two ideas, generalized from Watanabe et al [39], in new ways. First, Alice and Bob can “partition” their data into different subsets of the partition depending upon some predecided rule. If they do so, then conditioned on them obtaining data belonging to the same subset, their data is better correlated and more of it can be mapped to the key. If they do not obtain data that belongs to the same subset of the partition, then their data is poorly correlated, and it may be better to discard it. Second, Alice and Bob can determine which of the above case is true by performing error-correction on the partition information rather than revealing it openly. In typical cases, this leads to less information leakage to Eve and more key.

Lietzen et al [19], also building upon the work of Watanabe et al, recently proposed a new protocol to increase QKD key rates that has so far only been analyzed in the case of individual attacks. In this thesis, we are able to compute (asymptotic) key rates for this protocol for collective attacks. This protocol is tricky to analyze, since it does not implement IID blockwise processing, which is usually the case in generic QKD protocols. I.e, it does not divide all the signals into blocks and then implement the same operation on every block. We provide a rigorous analysis of this protocol that is novel and interesting. Our approach may have applications in analyzing other protocols without IID blockwise processing as well.

One significant problem in analyzing the protocols we propose is that one has to consider block sizes of more than one signals for classical processing. This means that if the protocol involves blocks of b signals, which each block processed identically, then describing Eve’s

information on the key requires modelling Eve as possessing at least a 4^b dimensional state¹. This requires us to diagonalize large matrices for key rate computations. We use several simple linear algebra tricks that allow us to vastly reduce this computational cost for our protocols for Bell-diagonal states. For the purposes of key rate calculations, we restrict our attention to Bell-diagonal states shared between Alice and Bob, and argue that this restriction can be done without loss of generality. For the entirety of this thesis, we work in the IID collective attack scenario [29], where Eve can have arbitrary quantum memory and attacks each signal in the same way with a fresh probe. In the asymptotic regime, this can be shown to be equivalent to coherent attacks using techniques such as the quantum de Finetti theorem [28] and the post-selection technique [7]².

This thesis is structured as follows. In Chapter 2, we give a brief overview of the basic concepts of quantum mechanics, quantum information theory, and QKD required to understand this thesis. In Chapter 3, we present two of our new protocols, and their key rate analysis followed by results showing improvements over our benchmarks. In Chapter 4, we present one new protocol and another protocol from [19], along with their key rate analysis and results. The first two protocols can be analyzed relatively easily, since they implement IID blockwise processing (see Sec. 3.2.2). The last two cannot, and we present some technical arguments needed to compute keyrates for these protocols in Sec. 4.1.2 and Sec. 4.1.3. This discussion might be of relevance even outside of this thesis. In Chapter 5, we present some concluding remarks and discuss some open questions. The appendices allow us to reduce the computational cost of computing our keyrates. In Appendix A, we show that without loss of generality, for all the protocols in this thesis, it can be assumed that Alice and Bob share a Bell-diagonal states. In Appendix B, we explain some simple linear algebra tricks we use for the keyrate calculations. In Appendices C, D, E and F, we explain the keyrate computations for all our protocols in detail.

¹Eve can be assumed to be 4 dimensional for each signal for the qubit-based protocols.

²The caveat here is that the protocols must satisfy the permutation-invariant condition, which can anyway be artificially enforced, albeit in an expensive manner, by performing a random permutation of all the rounds.

Chapter 2

Background

In this chapter, we will present the relevant background necessary to understand this thesis. We will also review definitions and setup our notation. The discussion of entropies is primarily taken from [34], whereas the rest of the material is from [40].

2.1 Quantum Information Theory

2.1.1 Hilbert Spaces

To each system we wish to describe in quantum information, we will associate a complex Euclidean space. To do so, let an *alphabet* be a finite and nonempty set, whose elements we call *symbols*. For any alphabet Σ , let $\mathbb{C}^\Sigma$ denote the set of all functions from Σ to the complex numbers $\mathbb{C}$. This set $\mathbb{C}^\Sigma$ forms a vector space of dimension $|\Sigma|$ over $\mathbb{C}$, where addition and multiplication are defined as :

1. Addition : for vectors $u, v \in \mathbb{C}^\Sigma$ and $a \in \Sigma$, $(u + v)(a) = u(a) + v(a)$.
2. Scalar Multiplication : For $u \in \mathbb{C}^\Sigma$ and $a \in \Sigma$ and scalar $c \in \mathbb{C}$, $(cu)(a) = cu(a)$.

This vector space is a *complex Euclidean space*. We use $\mathbb{C}^n$ to denote $\mathbb{C}^\Sigma$ where $\Sigma = \{1, 2, \dots, n\}$.

Definition 2.1 (Inner product space). Let F denote a field that is either the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$. An inner product space is a real or complex vector space V with an operation

$$\langle \cdot, \cdot \rangle : V \otimes V \rightarrow \mathbb{F}$$

such that the following conditions hold for all $x, y, z \in V$ and all scalars $c \in \mathbb{F}$:

1. Linearity :

$$\begin{aligned}\langle x, cy \rangle &= c \langle x, y \rangle \\ \langle x, y + z \rangle &= \langle x, y \rangle + \langle x, z \rangle\end{aligned}$$

2. Conjugate Symmetry :

$$\langle x, y \rangle = \langle y, x \rangle^*$$

3. Positive-definiteness : If x is not zero, then

$$\langle x, x \rangle > 0$$

One can then consider the following *inner product* $\langle u, v \rangle = \sum_{a \in \Sigma} u(a)^* v(a)$ over the complex Euclidean space. It is easy to verify that $\mathbb{C}^\Sigma$ is a complex inner-product space. For finite dimensions, this space can be shown to be complete in the norm and hence is a valid *Hilbert space*¹. Henceforth, for any system A , we will refer to its the Hilbert space associated with it via $\mathcal{H}_A$. The standard basis of $\mathbb{C}^\Sigma$ is an orthonormal basis given by $\{e_a : a \in \Sigma\}$, where $e_a(b) = 1$ if $a = b$ and 0 otherwise, Given n Hilbert spaces $\mathcal{H}_A = \mathbb{C}^\Sigma, \mathcal{H}_B = \mathbb{C}^\Gamma$, one can construct the bigger Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B = \mathbb{C}^{\Sigma \times \Gamma}$ via $(u \otimes v)(a_1, a_2) = u(a_1)v(a_2)$.

2.1.2 Linear Operator Theory

The space of linear operators from $\mathcal{H}_A = \mathbb{C}^\Sigma$ to $\mathcal{H}_B = \mathbb{C}^\Gamma$ will be denoted by $L(\mathcal{H}_A, \mathcal{H}_B)$. We will also use $L(\mathcal{H}_A)$ to denote $L(\mathcal{H}_A, \mathcal{H}_A)$. To each operator $X \in L(\mathcal{H}_A, \mathcal{H}_B)$, one can associate a matrix M_X after fixing a basis via :

$$M_X(a, b) = \langle e_a, X e_b \rangle$$

We will refer to both the matrix M_X and the operator X interchangeably from now on. The sum of the diagonal entries of an operator $X \in L(\mathcal{H}_A)$ is defined to be the trace of the operator :

$$\text{Tr}(X) = \sum_{a \in \Sigma} X(a, a)$$

¹In general, a Hilbert space is any inner-product space which is complete in the norm defined by the inner-product. For infinite dimensions, this requirements is non-trivial. For finite dimensions, all the Hilbert spaces are isomorphic to complex Euclidean spaces $\mathbb{C}^d$. In this thesis, we will only be concerned with finite dimensional Hilbert spaces.

Definition 2.2 (Adjoint of an operator). The operator $X^\dagger \in L(\mathcal{H}_B, \mathcal{H}_A)$ is the uniquely determined operator satisfying

$$\langle u, Av \rangle = \langle A^\dagger u, v \rangle$$

for all $u \in \mathcal{H}_A$ and all $v \in \mathcal{H}_B$.

We will need the following class of operators as we move forward :

Definition 2.3 (Important classes of Linear operators). Given $\mathcal{H}_A$, we define the following subsets of $L(\mathcal{H}_A)$ as follows :

1. Normal Operators :

$$\{X \in L(\mathcal{H}_A) : XX^\dagger = X^\dagger X\}$$

2. Hermitian Operators :

$$\text{Herm}(\mathcal{H}_A) = \{X \in L(\mathcal{H}_A) : X = X^\dagger\}$$

Every Hermitian operator is Normal.

3. Positive Semidefinite Operators :

$$\text{P}(\mathcal{H}_A) = \{X \in L(\mathcal{H}_A) : X = Y^\dagger Y, Y \in L(\mathcal{H}_A)\}$$

Every positive semidefinite operator is Hermitian.

4. Density Operators :

$$\text{D}(\mathcal{H}_A) = \{X \in \text{Pos}(\mathcal{H}_A) : \text{Tr}(X) = 1\}$$

5. Subnormalized Positive operators :

$$\text{P}_{\leq}(\mathcal{H}_A) = \{X \in \text{Pos}(\mathcal{H}_A) : \text{Tr}(X) \leq 1\}$$

Given $\mathcal{H}_A$ and $\mathcal{H}_B$, one can denote the set of linear maps from $L(\mathcal{H}_A)$ to $L(\mathcal{H}_B)$ as $T(\mathcal{H}_A, \mathcal{H}_B)$, with $T(\mathcal{H}_A, \mathcal{H}_A) = T(\mathcal{H}_A)$. These maps can be classified as follows :

Definition 2.4 (Important classes of Linear Maps).

1. Positive Maps :

$$\{\Phi \in T(\mathcal{H}_A, \mathcal{H}_B) : \forall X \in \text{Pos}(\mathcal{H}_A), \Phi(X) \in \text{Pos}(\mathcal{H}_B)\}$$

2. Completely Positive (CP) Maps :

$$\{\Phi \in T(\mathcal{H}_A, \mathcal{H}_B) : \forall \mathcal{H}_C, \forall X \in \text{Pos}(\mathcal{H}_A \otimes \mathcal{H}_C), (\Phi \otimes I)(X) \in \text{Pos}(\mathcal{H}_B \otimes \mathcal{H}_C)\}$$

Definition 2.5 (Adjoint Map). Given a map $\Phi \in T(\mathcal{H}_A, \mathcal{H}_B)$, the adjoint map denoted by $\Phi^\dagger$ is uniquely defined by

$$\langle \Phi(X), Y \rangle = \langle X, \Phi^\dagger(Y) \rangle \quad \forall X \in L(\mathcal{H}_A), \forall Y \in L(\mathcal{H}_B)$$

Definition 2.6 (Trace Norm). The trace norm of an operator $X \in L(\mathcal{H}_A)$ is given by

$$\|A\|_1 = \text{Tr}(\sqrt{X^\dagger X})$$

where the operator $\sqrt{P}$ corresponds to the unique positive operator that satisfied $\sqrt{P}\sqrt{P} = P$.

2.1.3 Quantum Mechanics

In general, we are interested in describing how quantum states are prepared, evolved, transformed and measured. In this section, we will see how quantum states, and operations on them, are represented.

Definition 2.7 (Quantum states).

- The state of a system A can be represented by a density matrix $\rho_A \in D(\mathcal{H}_A)$. A state ρ_A is said to be *pure* if it can be written as $\rho_A = uu^\dagger$ for unit vector $u \in \mathcal{H}_A$.
- A state is said to be *classical* if it is diagonal in the standard basis : i.e, if $\rho_A = \sum_i p(i) |e_i\rangle \langle e_i|$. A state $\rho \in D(\mathcal{H}_A \otimes \mathcal{H}_B)$ is said to be a CQ-state if it can be written as $\rho = \sum_i p(i) |e_i\rangle \langle e_i|_A \otimes \rho_B^{(i)}$.

Definition 2.8 (Measurements). A measurement M is represented by a positive-operator valued measure (POVM) $\{\Gamma_i\}_{i \in \Omega} \subset \text{Pos}(\mathcal{H}_A)$ such that $\sum_{i \in \Omega} \Gamma_i = \mathbb{I}$. On measuring M , the probability of obtaining outcome i is given by $p(i) = \text{Tr}(\Gamma_i \rho)$. The positivity and normalization conditions on the POVMs ensure that $p(i)$ is a valid normalized probability distribution.

Definition 2.9 (Partial Trace). Given a state $\rho_{AB} \in D(\mathcal{H}_A \otimes \mathcal{H}_B)$, the state on A is given by the partial trace of ρ_{AB} over B given by

$$\text{Tr}_B(\rho_{AB}) = \rho_A = \sum_j (\mathbb{I}_A \otimes \langle b_j |) \rho_{AB} (\mathbb{I}_A \otimes |b_j\rangle)$$

Definition 2.10 (Purifications). The purification of a state $\rho_A \in D(\mathcal{H}_A)$ is any vector $|\psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$ that satisfied $\text{Tr}_B[|\psi\rangle \langle \psi|_{AB}] = \rho_A$. Purifications are equivalent up to unitaries on the purifying space $\mathcal{H}_B$.

2.1.4 Measures and Entropies

We have the following notion of distance between states.

Definition 2.11 (Generalized Trace Distance). For $\rho, \tau \in P_{\leq}(\mathcal{H}_A)$, the generalized trace distance is a metric, and defined to be $\Delta(\rho, \tau) := \frac{1}{2}\|\rho - \tau\|_1 + \frac{1}{2}|\text{Tr}(\rho - \tau)|$. It reduces to the usual trace distance when $\rho, \tau \in D(\mathcal{H}_A)$.

Definition 2.12 (Generalized Fidelity and Purified Distance). For $\rho, \tau \in P_{\leq}(\mathcal{H}_A)$, the generalized fidelity is given by

$$F_*(\rho, \tau) := \left(\text{Tr}|\sqrt{\rho}\sqrt{\tau}| + \sqrt{(1 - \text{Tr} \rho)(1 - \text{Tr} \tau)} \right)^2$$

The purified distance is a metric and is given by

$$P(\rho, \tau) = \sqrt{1 - F_*(\rho, \tau)}$$

These quantities were first proposed in [35]. The two distance measures are related via inequalities analogous to Fuchs-van de Graaf inequalities.

Lemma 2.13 (Lemma 3.17 from [34]). For $\rho, \tau \in P_{\leq}(\mathcal{H}_A)$, the following inequalities hold :

$$\Delta(\rho, \tau) \leq P(\rho, \tau) \leq \sqrt{2\Delta(\rho, \tau)} \quad (2.1)$$

Definition 2.14 (von Neumann entropy). Let $\rho_A \in \text{Pos}(\mathcal{H}_A)$. The von Neumann entropy of ρ_A is defined as

$$S(A)_{\rho_A} = \sum_a -\lambda_a \log(\lambda_a) \quad (2.2)$$

where λ_a is the a th eigenvalue of ρ_A ². The conditional entropy of A given B is written as $S(A|B)_{\rho_{AB}} = S(AB)_{\rho_{AB}} - S(A)_{\rho_A}$.

Definition 2.15 (Min-entropy). Let $\rho_{AB} \in P_{\leq}(\mathcal{H}_{AB})$. The min entropy of A conditioned on B of the state ρ_{AB} is defined as

$$H_{\min}(A|B)_{\rho} := \sup_{\sigma_B \in P_{\leq}(\mathcal{H}_B)} \sup\{\lambda \in \mathcal{R} : \rho_{AB} \leq 2^{-\lambda} I_A \otimes \sigma_B\} \quad (2.3)$$

²It is understood that $0 \log 0 = 0$

The min-entropy quantifies the amount of randomness present in the system A , given access to system B . Another useful way to think about this quantity is as follows : Suppose we receive the state ρ_B^x with probability $p(x)$, and are asked to guess X . This situation can be described as a density matrix $\rho_{XB} = \sum_x p(x) |x\rangle \langle x|_X \otimes \rho_B^x$. The most general strategy for guessing involves choosing POVMs P_x corresponding to each guess, and measuring the system B . Then, the probability of correct guess is given by $p_g(\rho_{XB}, \{P_x\}) = \sum_x p(x) \text{Tr}(P_x \rho_B^x)$. The *optimal* guessing probability is then given by $p_g(G|B) = 2^{-H_{\min}(X|B)}$. This has a nice operational interpretation of an adversary trying to guess X given access to certain states conditioned on X .

Definition 2.16 (ϵ ball). Let $\rho \in P_{\leq}(\mathcal{H}_A)$ and let $0 \leq \epsilon < \sqrt{\text{Tr}(\rho)}$. We define the ϵ -ball of states in $P_{\leq}(\mathcal{H}_A)$ around ρ as

$$B^\epsilon(A; \rho) := \{\tau \in P_{\leq}(\mathcal{H}_A) : P(\tau, \rho) \leq \epsilon\} \quad (2.4)$$

Definition 2.17 (smoothed min-entropy). Let $\rho_{AB} \in P_{\leq}(\mathcal{H}_{AB})$. The ϵ -smooth min-entropy of A conditioned on B of the state ρ_{AB} is defined as

$$H_{\min}^\epsilon(A|B)_\rho := \max_{\bar{\rho}_{AB} \in B^\epsilon(AB; \rho)} H_{\min}(A|B)_{\bar{\rho}} \quad (2.5)$$

Here we have chosen to define the smoothed min-entropy via smoothing with respect to the purified distance, rather than the trace distance as is done in [28]. We do this since our results require the use of certain theorems that are proven only for the purified distance version of the smoothed min-entropy. The interested reader can look at [34] and references therein for more information.

Lemma 2.18 (Lemma 6.17 from [34]). Let $\epsilon \in [0, 1)$ and let $\rho \in D(\mathcal{H}_{AYB})$ be classical on Y , where d_Y is the dimension of Y . Then,

$$H_{\min}^\epsilon(A|YB)_\rho \geq H_{\min}^\epsilon(A|B)_\rho - \log d_Y \quad (2.6)$$

This lemma essentially states that the decrease in the amount of uncertainty about A that might result from revealing the classical register Y is at most the log of the dimension of Y .

2.1.5 Miscellaneous

Theorem 2.19 (Chernoff-Hoeffding theorem). Let $X_1 \dots X_n$ be IID random variables taking values in $\{0, 1\}$, and let $p = E[X_1]$ and $\epsilon > 0$. Then

$$\Pr \left(\frac{\sum_i X_i}{n} \geq p + \epsilon \right) \leq e^{-D(p+\epsilon||p)n} \quad (2.7)$$

where $D(x||y)$ is the Kullback–Leibler divergence and is given by

$$D(x||y) = x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}$$

This theorem states that the probability of the sum of a large number of random variables being significantly less than the expected value is very small. For our work, we do not care about the scaling of the bound and only require the property $D(x||y) \geq 0$ with equality iff $x = y$.

2.2 Quantum Key Distribution

A QKD protocol seeks to establish a shared secret key between two parties, Alice and Bob. The key is required to be secret from any eavesdropper Eve. To do this, Alice and Bob are given two channels : A completely insecure quantum channel, and an insecure and authenticated classical channel. Note that if the classical channel is not authenticated, Eve can simply impersonate Bob and no key generation is possible. At a high level, QKD works because Alice and Bob can announce a part of their data in order to determine how much information Eve could have stolen from the quantum signals. If this is too high, Alice and Bob choose to abort the protocol. If not, Alice and Bob can implement further steps to lower Eve’s information on the key. We describe the typical stages in the QKD protocol below³.

2.2.1 A generic QKD protocol

1. **Signal preparation, Transmission** : Alice prepares a large number of entangled states ρ_{AB} and sends B to Bob.

2. **Measurement** : Alice and Bob perform measurements on their part of the entangled states to obtain some public information (typically basis choice, detection events) and private information.

3. **Parameter Estimation** : Alice and Bob select some fraction of their data for parameter estimation. They openly announce both their private and public information to obtain the probability distribution $p(a, b)$, where a, b are Alice and Bob’s measurement

³We describe an entanglement based (EB) protocol here. We will later show how prepare and measure (PM) protocols can be reduced to EB protocols for the purposes of security proofs

outcomes ⁴. Alice and Bob accept or abort the protocol at this stage depending on their observed $p(a, b)$.

4. **Sifting** : Alice and Bob sift through their signals and discard a part of their data depending on the public announcements.

5. **Key Map** : Alice implements a *keymap* which maps her raw data to the *raw key*. This process may or may not involve communication with Bob. It is this stage of QKD protocols that is the topic of this thesis.

6. **Error correction (EC) and Privacy amplification (PA)** : Alice sends error correction information about her raw key to Bob. Bob will use this information to produce a guess for Alice's key. Additionally, there may be an error verification step performed, where Alice and Bob compute hashes to verify (with high probability) whether their strings are equal. Finally, Alice and Bob will implement privacy amplification by applying a two-universal hash function onto their raw keys.

2.2.2 Source Replacement Scheme

We have described the EB (entanglement-based) QKD protocol above. In practise, instead of creating an entangled state ρ_{AB} and then measuring her subsystem, Alice will simply prepare and transmit signals from a set of predecided signals. The latter is known as the Prepare and Measure (PM) protocol. The source replacement scheme [4, 10] allows us to reduce all PM protocols to EB protocols, in such a way that any valid key rate for the EB protocol is also a valid key rate for the PM protocol.

Let the PM protocol be such that Alice prepares the signal state $|\psi_i\rangle_{A'}$ with probability p_i . By the source-replacement scheme, it is equivalent for Alice to prepare the entangled state

$$|\psi\rangle_{AA'} = \sum_i p_i |i\rangle_A |\psi_i\rangle_{A'} \quad (2.8)$$

and send A' to Bob. If $\mathcal{E}$ is the quantum channel that describes Eve's attack, then the joint state between Alice and Bob is given by

$$\rho_{AB} = (I_A \otimes \mathcal{E})(|\psi\rangle \langle \psi|_{AA'}) \quad (2.9)$$

If Alice performs a project measurement on A , then Bob receives the following state with probability p_i ,

$$\rho_B^i = \frac{1}{p_i} \text{Tr}[(|i\rangle \langle i| \otimes I_B) \rho_{AB}] = \mathcal{E}(|\psi_i\rangle \langle \psi_i|) \quad (2.10)$$

⁴In any finite-size protocol, we only have a frequency distribution rather than a probability distribution. However, since this thesis is concerned with asymptotic key rates, we are able to obtain exact statistics.

which is identical to the one Bob would have received in the PM scheme. If in the original PM scheme, Alice and Bob obtained $p(i, j)$ as the probability distribution, then it is easy to see that our ρ_{AB} will also have to satisfy

$$\text{Tr}[(|i\rangle \langle i| \otimes \Gamma_j^B) \rho_{AB}] = p(i, j) \quad (2.11)$$

where $\{\Gamma_j^B\}$ is the set of Bob's POVMs. However, it is not enough to consider all states ρ_{AB} with the above constraints as the set of possible states Alice and Bob share, since a separable state $\sum_i p_i |i\rangle \langle i| \otimes |\psi_i\rangle \langle \psi_i|$ satisfies the above equation and yields zero keyrate. One has to additionally argue that since Eve cannot affect Alice's subsystem (since it never leaves her lab), $\text{Tr}_B(\rho_{AB}) = \text{Tr}_{A'}(|\psi\rangle \langle \psi|_{AA'})$

2.2.3 Security framework

Definition 2.20 (Security of a Key). A key is ϵ -secure (sometimes referred to as ϵ -secret) with respect to an adversary Eve if the state ρ_{KE} , where K is Alice's key register and E is *all* of Eve's information satisfies

$$\frac{1}{2} \|\rho_{KE} - \tau_K \otimes \rho_E\|_1 \leq \epsilon$$

where τ_K is the perfectly mixed state on K , and ρ_{KE} is the output of the protocol conditioned on not aborting⁵.

Since the trace distance is a measure of distinguishability of states, the above statement states that the key is ϵ -secure if it cannot be distinguished from the one obtained from the ideal protocol except with probability ϵ .

Definition 2.21 (Security of QKD protocol). A QKD protocol is ϵ -secure if for all input states, the output state $\rho_{K_A K_B E}$ satisfies

$$\frac{1}{2} \|\rho_{K_A K_B E} - \tau_{K_A K_B} \otimes \rho_E\|_1 \leq \epsilon$$

where $\tau_{K_A K_B} = \frac{1}{|K_A|} \sum_k |kk\rangle \langle kk|$, and $\rho_{K_A K_B E}$ be the output of the protocol conditioned on not aborting.

⁵Alternatively, one could let always let ρ_{KE} be the output of the protocol, with the understanding that in case of abort, the key is mapped to a trivial symbol $\perp$

In general the raw keys obtained from measurements are not secure. However, Alice and Bob can use two-universal hashing to get rid of Eve's correlations with the key.

Theorem 2.22 (Leftover Hashing Lemma, [36]). Let $\rho \in P_{\leq}(\mathcal{H}_{ZE})$ and be a classical n -bit register on Z and $\epsilon \in [0, \sqrt{\text{Tr}(\rho)})$. Let $\mathcal{F}$ be a two-universal family of hash functions from n bits to l bits, and let K be a classical l -bit register. Let U_F and U_K be the perfectly mixed states on $|\mathcal{F}|$ and $|K|$. Let $\mathcal{E}$ denote the map that applies the hash function randomly chosen from $\mathcal{F}$ to Z to obtain the register K . Let $\omega_{KZEF} := \mathcal{E}(\rho_{ZE} \otimes U_F)$, then we have

$$\frac{1}{2} \|\omega_{KZEF} - U_Z \otimes \omega_{EF}\|_1 \leq 2^{-\frac{1}{2}(H_{\min}^{\epsilon}(Z|E)_{\rho} - l + 2)} + 2\epsilon$$

In this theorem Z should be interpreted as the raw key register, E as all of Eve's side information and K is the final key register. The trace distance measures how far the final key is from the *perfect* key. Note that in this theorem ρ represents the state of *all* our signals. Denoting the smoothing parameter with ϵ_s and the final security parameter with ϵ , the above theorem gives us a condition on l for obtaining ϵ -secure key :

$$\begin{aligned} 2^{-\frac{1}{2}(H_{\min}^{\epsilon_s}(Z|E)_{\rho} - l + 2)} + 2\epsilon_s &\leq \epsilon \\ l &\leq H_{\min}^{\epsilon_s}(Z|E)_{\rho} + 2 + 2\log(\epsilon - 2\epsilon_s) \end{aligned} \quad (2.12)$$

The above statement can be interpreted as follows : If one desires ϵ security, then it can be guaranteed by choosing a smoothing parameter $\epsilon_s \leq \epsilon/2$, and an l that satisfies the above equation. The asymptotic keyrate is therefore simply given by

$$K \geq \lim_{n \rightarrow \infty} \frac{H_{\min}^{\epsilon_s}(Z|E)_{\rho} + 2 + 2\log(\epsilon - 2\epsilon_s)}{n} = \lim_{n \rightarrow \infty} \frac{H_{\min}^{\epsilon_s}(Z|E)_{\rho}}{n} \quad (2.13)$$

This is difficult to compute for large n , unless the state $\rho = \sigma^{\otimes n}$ is known to be IID. In that case, the following theorem can be used to compute the key rate

Theorem 2.23 (Asymptotic equipartition property [12]). For any $\sigma \in D(\mathcal{H}_{AB})$, then for any $\epsilon \in (0, 1)$, we have

$$H_{\min}^{\epsilon}(A^{\otimes n}|B^{\otimes n})_{\sigma^{\otimes n}} > nH(A|B)_{\sigma} - \sqrt{n}(2\log(1 + 2\dim(A)))\sqrt{\log \frac{2}{\epsilon^2}}$$

The main property that we will need for asymptotic keyrates is $\lim_{n \rightarrow \infty} \frac{H_{\min}^{\epsilon}(A^{\otimes n}|B^{\otimes n})_{\sigma^{\otimes n}}}{n} > H(A|B)_{\rho}$.

2.2.4 BB84 and Six-state protocols

In this section we will describe the BB84 and six-state protocols that are studied in this thesis.

1. Signal transmission (PM scheme) : Alice chooses the basis Z with probability p_Z and sends the states $|0\rangle, |1\rangle$ with equal probability. Alice chooses the basis X with probability p_X and sends the states $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}, |-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ with equal probability. For the six-state protocol, Alice chooses an additional basis Y with probability p_Y and sends the states $|+i\rangle = \frac{|0\rangle+i|1\rangle}{\sqrt{2}}, |-i\rangle = \frac{|0\rangle-i|1\rangle}{\sqrt{2}}$ with equal probability.

1. Signal transmission (EB scheme) : Alice prepares the Bell state $|\phi_+\rangle = \frac{|00\rangle_{AB}+|11\rangle_{AB}}{\sqrt{2}}$ and sends the qubit B to Bob. Alice measures her system A in the basis X with probability p_X , and basis Z with probability p_Y . For the six-state protocol, Alice chooses an additional basis Y with probability p_Y . The POVMs for each basis are given by $F_X = \{|+\rangle\langle+|, |-\rangle\langle-|\}, F_Y = \{|+i\rangle\langle+i|, |-i\rangle\langle-i|\}, F_Z = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$. For each outcome, the total information is split into public information (basis choice) and a 1-bit private information (outcome given basis choice).

2. Bob's Measurements : Bob's measurements are identical to Alice's. Bob measures his system B in the X basis with probability p_X , and Z basis with probability p_Y . For the six-state protocol, Bob chooses an additional Y basis with probability p_Y . Bob flips the Y outcomes, since Y outcomes are anti correlated for $|\phi_+\rangle$.

3 Parameter Estimation : Alice and Bob will announce the measurement results of some test rounds. They will obtain the probability distribution $p(a, b)$. Alice and Bob will then compute the QBERs (quantum-bit-error-rates) in each of the the basis Q_X, Q_Z and Q_Y (Q_Y for six-state only). They will choose whether or not to abort the protocol.

4 Sifting : For the remaining rounds, Alice and Bob will announce their basis choice and sift through their data to discard all rounds where Alice and Bob chose different basis.

Remark In this thesis, for simplicity we will use asymmetric basis choice with the Z basis chosen for key generation. Therefore, all X and Y measurement rounds will be used in parameter estimation, and p_Z can be set arbitrary close to one ⁶.

6. Key Map : Alice and Bob will then implement a protocol that results in Alice implementing a keymap on her data to obtain a bit string. For example, Alice could simply map measurements $|0\rangle\langle 0|$ to 0, and $|1\rangle\langle 1|$ to 1. In this thesis, we will study improved ways of performing this map.

6. EC and PA : Alice sends error correction information (if required). Alice and Bob implement privacy amplification.

⁶Essentially this allows us to do PE by sacrificing a negligible amount of signals to basis-mismatch

In this thesis we look at various classical methods for Alice and Bob to implement the keymap. We will show that some of these methods are lead to higher asymptotic secret key rate. We will work under the IID collective attack assumption, where we assume that Eve chooses a fresh probe to attack each new signal Alice sends to Bob. Eve is allowed unlimited quantum memory, and can perform any valid operation on her system and the signal, as long at it is the same for all the signals.

Chapter 3

Protocols for Classical Processing in QKD - 1

In this chapter, we will look at two new classical protocols proposed in this thesis. These classical protocols describe how Alice and Bob transform their measurement results into a secret key. We first setup our notation and describe the setting in which we propose and study these classical protocols. Then, we describe the simplest of our new classical processing protocols, which is a generalization of the protocol from Watanabe et al [39]. Next, we will turn to its key rate analysis and results. We will then move on to the second classical processing protocol, and explain the protocol itself, its key rate analysis and results.

3.1 Preliminaries

We analyze our proposed classical protocols for qubit-based BB84 and six-state protocols in the asymptotic regime, with asymmetric basis choice. This means that the signal preparation, transmission and measurement phase of our protocols is exactly identical to the qubit-based BB84 and six-state protocols. We use only the Z basis for key generation. Using the equivalence between prepare-and-measure and entanglement-based schemes, we assume the latter from now on. Thus, our protocol consists of Alice and Bob receiving a state and performing measurements in the standard X , Z (and Y for six-state) basis on it. We assume that Bob flips his Y outcomes, since the Y outcomes in the entanglement-based scheme are anti correlated. This is done purely for convenience, so that the “error”

in Y basis corresponds to Alice and Bob obtaining *different* outcomes. In the asymptotic regime with collective attacks, parameter estimation allows Alice and Bob to know the exact statistics of the states they share.

For all the protocols we consider, we can show that one can assume that a Bell-diagonal state is shared between Alice and Bob. This can be argued similar to the approach in [39, 14], and is done in Appendix A for all our protocols. Note that this Reduction to Bell-diagonal states is only used at the last step of the keyrate calculation, when we are required to compute the conditional entropy. In this case, the state shared between Alice and Bob can be written as

$$\rho_{AB} = \lambda_0 |\phi_+\rangle \langle \phi_+| + \lambda_1 |\phi_-\rangle \langle \phi_-| + \lambda_2 |\psi_+\rangle \langle \psi_+| + \lambda_3 |\psi_-\rangle \langle \psi_-|, \quad (3.1)$$

where the λ_i s are related to QBERs via, $Q_Z = \lambda_2 + \lambda_3$, $Q_X = \lambda_1 + \lambda_3$, $Q_Y = \lambda_1 + \lambda_2$ and the Bell states are given by

$$\begin{aligned} |\phi_+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, & |\phi_-\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\ |\psi_+\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}}, & |\psi_-\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}} \end{aligned} \quad (3.2)$$

We can assume Eve holds a purification of the form

$$|\psi\rangle_{ABE} = \sqrt{\lambda_0} |\phi_+\rangle |e_0\rangle + \sqrt{\lambda_1} |\phi_-\rangle |e_1\rangle + \sqrt{\lambda_2} |\psi_+\rangle |e_2\rangle + \sqrt{\lambda_3} |\psi_-\rangle |e_3\rangle \quad (3.3)$$

where $|e_i\rangle$ are orthonormal basis vectors for Eve's system. Let the set of states that satisfy QBER constraints be given by Γ . Let Γ_{Bell} be the set of states satisfying QBER constraints and which are Bell-diagonal between Alice and Bob.

After Alice and Bob perform their measurements (in the Z basis since that is the only one used to generate the key), the state is given by

$$\rho_{XYE} = \sum_{x,y \in \{0,1\}} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \rho_E^{x,y} \quad (3.4)$$

where X and Y denote Alice and Bob's classical registers storing measurement results and $\rho_E^{x,y} = \text{Tr}(|x\rangle \langle x| \otimes |y\rangle \langle y| \otimes I_E |\psi\rangle \langle \psi|_{ABE})$ is the state of Eve's system conditioned on

Alice and Bob measuring x, y . Eve's conditional states are given by

$$\begin{aligned}
\rho_E^{x,y} &= |e_{x,y}\rangle \langle e_{x,y}| \\
|e_{0,0}\rangle &= \sqrt{\frac{\lambda_0}{2}} |e_0\rangle + \sqrt{\frac{\lambda_1}{2}} |e_1\rangle \\
|e_{1,1}\rangle &= \sqrt{\frac{\lambda_0}{2}} |e_0\rangle - \sqrt{\frac{\lambda_1}{2}} |e_1\rangle \\
|e_{0,1}\rangle &= \sqrt{\frac{\lambda_2}{2}} |e_2\rangle + \sqrt{\frac{\lambda_3}{2}} |e_3\rangle \\
|e_{1,0}\rangle &= \sqrt{\frac{\lambda_2}{2}} |e_2\rangle - \sqrt{\frac{\lambda_3}{2}} |e_3\rangle
\end{aligned} \tag{3.5}$$

Observe that since the space spanned by $\{|e_{0,0}\rangle, |e_{1,1}\rangle\}$ is orthogonal to the space spanned by $\{|e_{0,1}\rangle, |e_{1,0}\rangle\}$, Eve can be assumed to always know the value of $x \oplus y$. This is because her conditional states are block-diagonal in $x \oplus y$.

We wrote the above equations for measurements undertaken on a single ρ_{AB} shared in one round. However, our classical protocols will require Alice and Bob to process blocks of n -bit each in some manner. Thus, one can easily generalize the above discussion to n bits, and write

$$\begin{aligned}
\rho_{XYE} &= \sum_{x,y \in F_2^n} p(x,y) |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \rho_E^{x,y} \\
\rho_E^{x,y} &= \bigotimes_{i=1}^n \rho_E^{x_i, y_i} = |e_{x,y}\rangle \langle e_{x,y}| \\
|e_{x,y}\rangle &= \bigotimes_{i=1}^n |e_{x_i, y_i}\rangle.
\end{aligned} \tag{3.6}$$

In general, the information Eve has on the key is completely a function of the conditional states above. However, for an n bit block, these conditional states are 4^n dimensional positive matrices, and are thus in general difficult to diagonalize and compute entropies on. However, one can make use of the fact that $\rho_E^{x,y}$ is orthogonal to $\rho_E^{x',y'}$ if $x \oplus y \neq x' \oplus y'$ to obtain a block-diagonal matrix that is simpler to diagonalize. This and other such tricks are widely used to complete the last step of the key rate calculations in this thesis, and are collected in Appendix B.

3.2 Compressed Partition Announcements (CPA)

This protocol ¹ extends and generalizes the ideas from [39]. At a high-level, the protocol works as follows: For each block, Alice and Bob divide implement a partition on their data, classifying each block of data into subsets of the partition depending on some of its parities. Thus, each subset can be labelled by parity strings. They then communicate and determine the blocks where they obtained the same parity string (i.e, subset of the partition) vs the blocks where they did not. Alice can then choose to map more of her bits to the key when she knows that her and Bob's data have matching parity strings, since their data conditioned on parities matching is better correlated.

One typical way for Alice and Bob to determine their parity string match/mismatch status is for Alice to announce to Bob (and Eve) her parity string and for Bob to respond with a match/mismatch status. This does the job but drastically lowers keyrate since the announcement leaks the entire parity information to Eve.

In [39], the authors note that in general, Alice and Bob's parities are correlated. Hence, Alice can choose to send error correction information for her partitions to Bob, and Bob will be able to decode her parity strings and reply with a match/mismatch answer. Typically, especially for lower error rates, this error correction based approach leaks much less information to Eve than the direct announcement of parity strings. In [39] the authors proposed a specific protocol involving error-correction of parities, and partitions determined by parities of two-bit blocks. In this thesis, we have extended this to partitions determined by an arbitrary set of parities of the block.

3.2.1 Protocol description

- Alice and Bob divide their raw data into blocks of n bits X and Y .
- Alice and Bob use a predecided $(n - k) \times n$ parity check matrix H to compute their $(n - k)$ bit parity strings $P_X = HX$ ($P_Y = HY$) for each n bit block.
- Alice sends error correction information for Bob to decode P_X for all n bit blocks. Bob sends feedback $F = 0$ if $P_X = P_Y$, and $F = 1$ if $P_X \neq P_Y$, for each n bit block.
- Alice maps P_X to the key register Z if $F = 1$ (parity string did not match) and X if $F = 0$ (parity string match).
- Alice and Bob perform error correction (EC) and privacy amplification (PA).

See Fig. 3.1 for an example of the CPA protocol for $H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$. It is useful

¹CPA has no relation to chosen-plaintext attacks

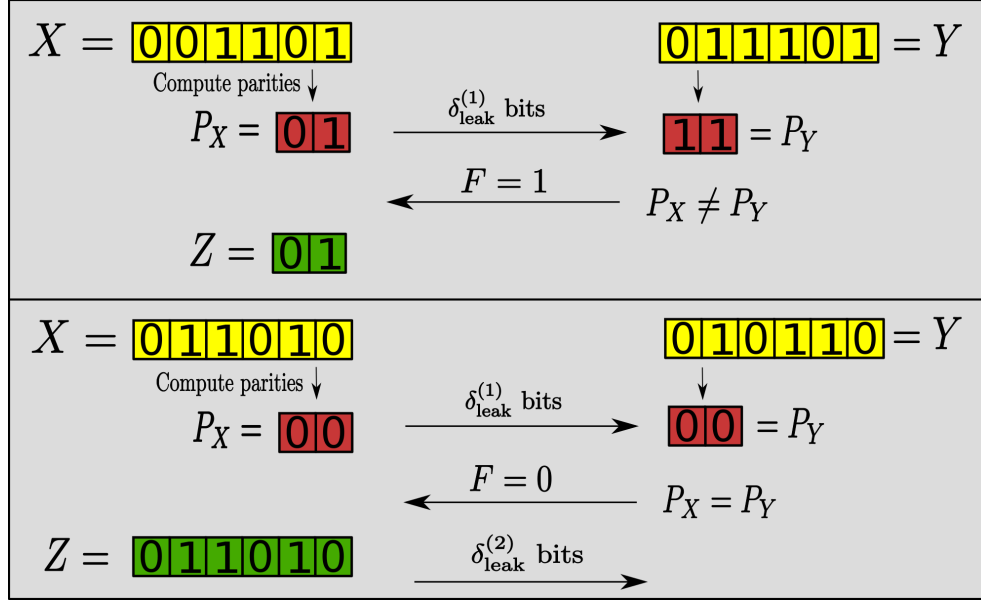


Figure 3.1: An example of our CPA protocol for $H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$

to think of this protocol as advantage distillation (AD) with “improved” announcements, as we are able to obtain some key out of the parity string P_X . In usual AD protocols, this P_X is announced directly, and hence cannot be used in the key. In fact, the above protocol with P_X announced directly is exactly equivalent to the AD protocol.

3.2.2 Keyrate calculations for protocol with IID blockwise processing

In the usual version of QKD classical post-processing, Alice and Bob make some announcements (announcing for example basis choice), sift and discard signals, Alice implements a keymap, and then proceeds to do error correction. The error-correction step always occurs after the end of the keymap². For a protocol that follows the above steps in that order, and for IID collective attacks, the key rate can be shown to be equal to the so called

²Although error correction is just another announcement, it is treated differently from announcements of basis choice in our security proof

Devetak-Winter formula [11]:

$$K \geq \min_{\rho \in \Gamma} S(Z|E) - S(Z|Y) \quad (3.7)$$

where Z is Alice's key register, Y is all of Bob's information, and E is all of Eve's information (including all announcements other than error correction), and Γ is the set of feasible states that satisfy the constraints obtained from parameter estimation. The first term can be thought of as a measure of Eve's uncertainty about the key, and the second term represents the number of bits leaked during error correction to Eve. The sifting can be modelled as Alice mapping the key to $\perp$ or "discard".

The CPA protocol involves an error correction step *prior* to the keymap (when parity string P_X is corrected), in addition to an error correction step after the keymap. Thus it is not obvious whether the above formula applies, even when counting information leakage during both error-correction steps. It is instructive to go over the steps involved in the keyrate calculations in some detail. We will find this useful in later protocols as well, when we encounter non-IID keymaps.

Let N be the total number of signals, and let $\vec{Z}$ denote the total key register, while $\vec{E}$ denotes Eve's quantum system, $\vec{C}$ denotes *all* classical communication revealed during *all* error correction, and $\vec{F}$ denotes feedback from Bob. We split up the classical communication into separate registers because they are handled differently. $\vec{F}$ is blockwise announcement whereas $\vec{C}$ "mixes" up the blocks and we are forced to bound its effect in a different way.

From Eq. 2.13, to compute an achievable asymptotic keyrate K , we wish to compute a lower bound on

$$\lim_{N \rightarrow \infty} H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{F}\vec{C})/N.$$

Our first step is to split off the information leaked during error correction via

$$H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{F}\vec{C}) \geq H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{F}) - \log(\dim(\vec{C})) \quad (3.8)$$

(Lemma 6.17 [34], Lemma 2.18), where $\dim(\vec{C})$ is the dimension of $\vec{C}$. We stress that the above statement is valid for any classical $\vec{C}$, regardless of when and how it is computed and revealed in our protocol. $\log(\dim(\vec{C}))$ can be estimated by counting the number of bits required for error correction. One can use the Slepian-Wolf theorem for the optimal value for each step of error correction in the protocol [31].

Now we are left with lower bounding $H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{F})$, for which we use the fact that the output state of our protocol is obtained by *IID block-wise processing of IID input states*.

Let n be the blocksize of processing, i.e Alice and Bob process each block of n signals identically and separately, according to some map $\mathcal{E}_{(AB)^n \rightarrow ZF}^{\otimes N/n}$ which acts on all the N/n blocks. Since this map acts on the state $\rho_{QBE}^{\otimes N}$, the result of the above operation is also an IID state with N/n copies given by $\rho_{ZEF}^{\otimes N/n} = (\mathcal{E}_{(AB)^n \rightarrow ZF}^{\otimes N/n} \otimes I) \rho_{ABE}^{\otimes N}$. In this case, one can use the AEP property (Theorem 2.23) to compute the asymptotic keyrate using

$$\lim_{N \rightarrow \infty} \frac{H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{F})_{\rho_{ZEF}^{\otimes N/n}}}{N} \geq \frac{1}{n} \frac{S(Z|EF)_{\rho_{ZEF}}}{N/n} = \frac{S(Z|EF)_{\rho_{ZEF}}}{n} \quad (3.9)$$

If we let δ_{leak} be the number of bits leaked during error correction *per block* in $\vec{C}$, then the keyrate is given by

$$K = \min_{\rho \in \Gamma} \frac{S(Z|EF)_{\rho_{ZEF}} - \delta_{\text{leak}}}{n} \quad (3.10)$$

where we take the minimum keyrate over all states that can lead to Alice and Bob's observations. Note that here we see that *all* classical side-information is modelled and assumed to be known to Eve, but in *two different ways*. For blockwise communication (eg. $\vec{F}$), we explicitly give the register to Eve and compute its effect on the conditional entropy, whereas for communication that "mixes" all blocks (eg. error correction communication $\vec{C}$), we bound its effect by subtracting the number of bits. This will be a common feature of all our analysis for all our protocols.

3.2.3 Keyrate analysis

The CPA protocol implements blockwise IID processing. Therefore the asymptotic keyrate is given by an slightly modified version of Eq. 3.10 :

$$K_{\text{CPA}} = \min_{\rho \in \Gamma} \frac{S(Z|EF) - \delta_{\text{leak}}^{(1)} - \delta_{\text{leak}}^{(2)}}{n} \quad (3.11)$$

where Z is the key register, E is Eve's quantum system, F is the feedback from Bob to Alice for each n bit block. The $\delta_{\text{leak}}^{(1)}$ term is the number of bits per block sent from Alice to Bob for correcting the partition strings, and is equal to $S(P_X|Y)$ by the Slepian Wolf theorem. The $\delta_{\text{leak}}^{(2)}$ term is the number of bits per block sent from Alice to Bob for correcting the raw key, *after partitions have been corrected*, and is equal to $S(Z|Y P_X)$. These terms together count all the bits leaked during error-correction steps in the protocol. Thus, the key rate is given by

$$K_{\text{CPA}} = \min_{\rho \in \Gamma} \frac{1}{n} \left(S(Z|EF) - S(Z|Y P_X) - S(P_X|Y) \right) \quad (3.12)$$

If p_{match} is the probability of partitions matching ($F = 0$), for the keymap Alice implements we have

$$\begin{aligned} S(Z|EF) &= p_{\text{match}}(S(X|E, F = 0) + (1 - p_{\text{match}})S(P_X|E, F = 1)) \\ S(Z|YP_X) &= S(Z|YP_X F) = p_{\text{match}}S(X|YP_X, F = 0) + (1 - p_{\text{match}})S(P_X|YP_X, F = 1) \end{aligned} \quad (3.13)$$

where we have used that F is a function of Y, P_X . Since $S(P_X|YP_X, F = 1) = 0$, the keyrate can be rewritten as

$$\begin{aligned} K_{\text{CPA}} = \min_{\rho \in \Gamma} \frac{1}{n} &\left(p_{\text{match}}(S(X|E, F = 0) - S(X|YP_X, F = 0)) \right. \\ &\left. + (1 - p_{\text{match}})S(P_X|E, F = 1) - S(P_X|Y) \right) \end{aligned} \quad (3.14)$$

At this point, for minimizing the above function, we can without loss of generality restrict our attention to Bell-diagonal states satisfying our QBER constraints (see Appendix A). Therefore, our keyrate is given by

$$\begin{aligned} K_{\text{CPA}} = \min_{\rho \in \Gamma_{\text{Bell}}} \frac{1}{n} &\left(p_{\text{match}}(S(X|E, F = 0) - S(X|YP_X, F = 0)) \right. \\ &\left. + (1 - p_{\text{match}})S(P_X|E, F = 1) - S(P_X|Y) \right) \end{aligned} \quad (3.15)$$

More details about how the above expressions are actually computed can be found in Appendix C.

3.2.4 Results

We compute keyrates for a number of choices of the partition matrix H . We compare our keyrates against two benchmarks. “Watanabe” refers to the keyrate obtained from [39], which is simply a special case of our CPA protocol with $H = [1, 1]$. We also compare our keyrates against the best repetition code advantage distillation protocol at that error rate, optimized over all possible block sizes.

Note that in [39], it is argued that the keyrate of the CPA protocol cannot be lower than the corresponding protocol where Alice announces P_X directly. Therefore, the keyrate for a protocol where Alice announces P_X is a valid lower bound on K_{CPA} . For $H = [1, 1]$, this means that the repetition-code advantage distillation keyrate [28] with blocksize equal to 2, is a valid lower bound on the Watanabe keyrate. In [39] it was shown that the

keyrate obtained from Eq. 3.15 is sometimes lower than this lower bound, and in such cases the repetition-code AD keyrate should be treated as the CPA keyrate. Since we have repetition-code advantage distillation keyrates as our benchmark anyway, we plot the Watanabe keyrate as the one evaluated by Eq. 3.15.

However, this leads us to the interesting question of why Eq. 3.15 yields a lower keyrate than the keyrate for the protocol with direct announcement of parity strings. The answer is the difference in proof technique. To put it simply, announcements that can be modelled as IID announcement (say F or P_X) for each block of data can be “included” in Eve’s information explicitly, by computing $S(Z|EP_X)$ instead of $S(Z|E)$. Thus when P_X is announced directly, P_X as a register can be given to Eve. However, announcements that “mix” together all the blocks (such as error correction) are treated by simply subtracting the number of bits of the announcement from the smooth min-entropy term. The above paragraph illustrates that the latter method is *not* tight.

For both six-state and BB84 protocols, we test the keyrate from the CPA protocol for all H upto $n = 5$, for equal QBERs in all basis. We find that only a handful of choices lead to improvements over both our benchmarks. We plot the three scenarios where we obtain an improvement in Figs. 3.2, 3.3, 3.4. Our final keyrate expression is such that H is equivalent upto permutation of rows (due to IID input), and permutations of columns and row additions (both of which implement a permutation on the key register). Therefore such choices are not treated as separate choices of H . Exhaustive search over all values of H for higher values of n is not feasible, although we can compute keyrates for an individual H with n as high as 10.

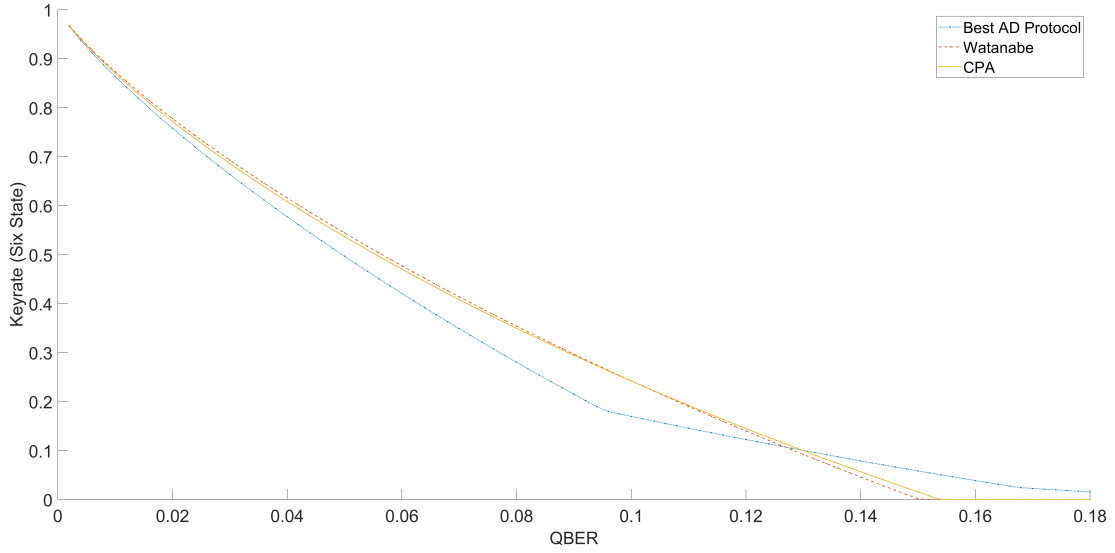


Figure 3.2: Six-state keyrate for the CPA protocol with $H = [1, 1, 1]$. We find that the CPA keyrate is higher than both our benchmarks for some range of QBER.

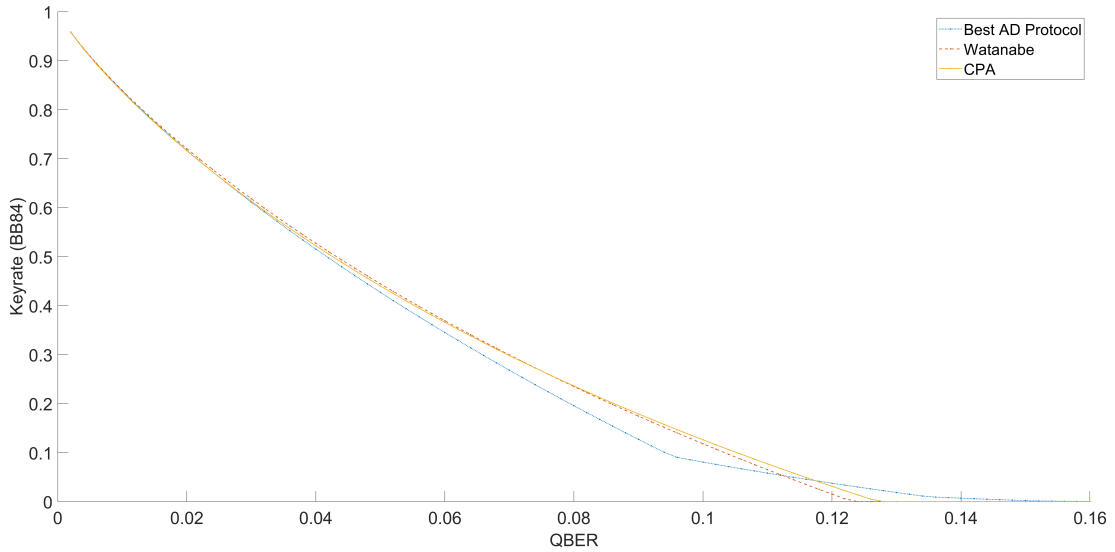


Figure 3.3: BB84 keyrate for the CPA protocol with $H = [1, 1, 1]$. We find that the CPA keyrate is higher than both our benchmarks for some range of QBER.

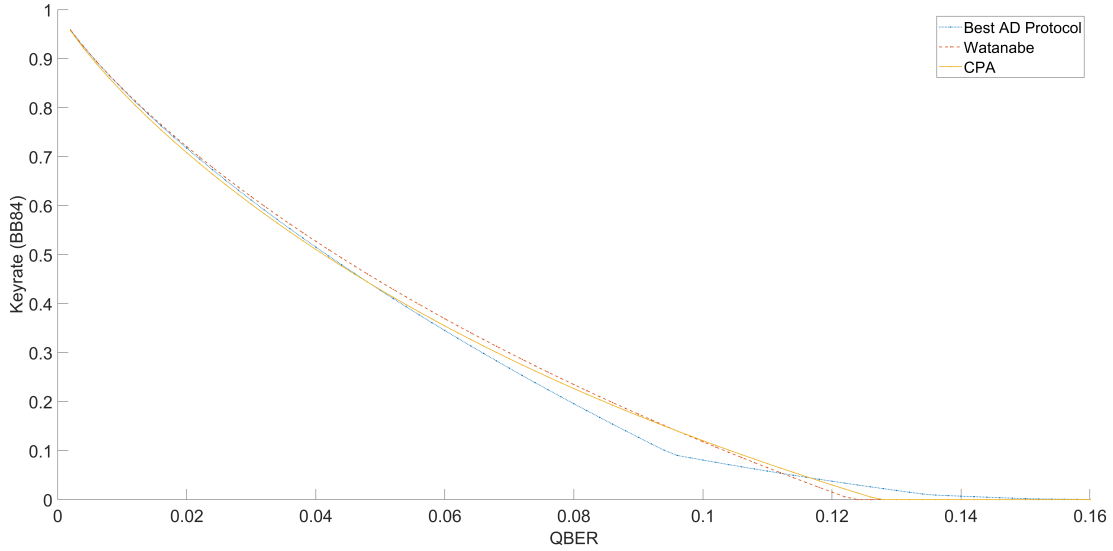


Figure 3.4: BB84 keyrate for the CPA protocol with $H = [1, 1, 1, 1]$. We find that the CPA keyrate is higher than both our benchmarks for some range of QBER.

We test H corresponding to repetition codes for block sizes upto 40, for which an analytic answer is possible (see Appendix C). We find no improvement over the Watanabe benchmark for the six-state protocol.

3.3 Sequential Compressed Partition Announcements (SCPA)

In the previous protocol we considered, for each block Alice and Bob first determined if their data belongs to the same subset of partition or not. This was done by comparing their parity strings. Instead of Alice announcing her parity string directly, Alice sent error correction information to Bob. Bob would then announce whether the parity strings matched or not. In this process Alice and Bob would correct the entire $n - k$ bit parity string for all the blocks in one go.

In the next protocol, instead of correcting the entire parity string in one go, Alice and Bob correct them bit-by-bit. And they stop correcting as soon as an error is found i.e, when it is known that the partitions are different. This translates to a lower number of

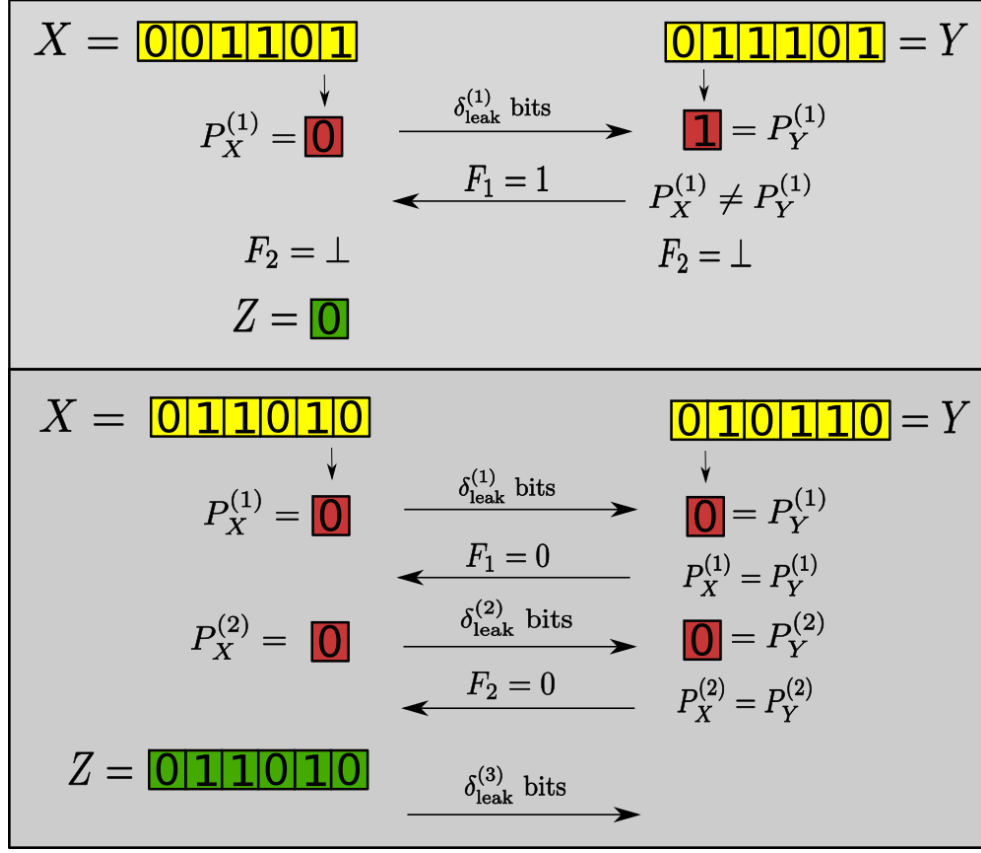


Figure 3.5: An example of our SCPA protocol for $H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$.

bits communicated during error correction, but also leads to a shorter key. Thus there is a trade-off, which in some cases leads to a higher overall keyrate.

3.3.1 Protocol description

- Alice and Bob divide their data into n bit blocks X and Y .
- Alice and Bob use a pre-decided $n - k \times n$ parity check matrix H to compute their parity strings $P_X = HX$ ($P_Y = HY$) for each n bit block. Then, for round j , with j going from 1 to $n - k$, Alice does the following :
 - a) Alice sends error correction information for the j th bit of all blocks whose previous partition bits had no error.

b) Bob provides feedback $F_j = P_X^{(j)} \oplus P_Y^{(j)}$ to Alice for each of these blocks. For remaining blocks (i.e, blocks which had some error in previous parity bits), Alice and Bob set $F_j = \perp$.

c) All corrected parity bits are mapped to the key. Alice and Bob move to the $j + 1$ th round.

- For the blocks where all parity bits were without error, Alice maps the entire block to the key (instead of just the parity bits) and performs EC.
- For blocks where an error is found in the parity string, only the part of the parity string already corrected is mapped to the key.
- Alice and Bob implement PA.

Since Alice and Bob only correct blocks where previous bits had no errors, the cost of error correction is lower than CPA. However, the length of the raw key is also lower. Thus there is a trade-off. See Fig. 3.5 for an example of the SCPA protocol for $H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$.

One can view this protocol as first performing an error detection step over the partitions. At the end of the error detection step, Alice and Bob are left with two types of blocks : those where an error was detected (where they map only the corrected parity bits to the key), and those where no error was detected (where they map the whole block to the key).

3.3.2 Keyrate analysis

Since this protocol also implements IID block-wise processing, our analysis from Sec. 3.2.2 is applicable here. The asymptotic keyrate of the protocol will then be given by,

$$K_{\text{SCPA}} = \min_{\rho \in \Gamma} \frac{S(Z|EF) - \sum_{j=1}^{n-k} \delta_{\text{leak}}^{(j)} - \delta_{\text{leak}}^{(\perp)}}{n} \quad (3.16)$$

where Z is the key register, E is Eve's quantum system and F is the feedback from Bob to Alice for each n bit block. $\delta_{\text{leak}}^{(j)}$ is the number of error correction bits sent per block from Alice to Bob in the j th round (to correct the j th parity bit), and $\delta_{\text{leak}}^{(\perp)}$ is the number of error correction bits sent after the last round is completed. We can again show that it is enough to restrict the above minimization to Bell-diagonal states (see Appendix A). Therefore,

$$K_{\text{SCPA}} = \min_{\rho \in \Gamma_{\text{bell}}} \frac{S(Z|EF) - \sum_{j=1}^{n-k} \delta_{\text{leak}}^{(j)} - \delta_{\text{leak}}^{(\perp)}}{n} \quad (3.17)$$

Now, we can simplify the above expression as follows. For each block, let N_R be the number of the round where the first error is found ($N_R = \perp$ if no error is found in any round for

that block). Let p_j be the probability that $N_R = j$, for $j \in \{1, 2, \dots, n - k, \perp\}$. Since Eve can be assumed to know the location of errors, she can also be assumed to know F and N_R . Since F and N_R are known to Eve, we can write

$$\begin{aligned} S(Z|EF) &= S(Z|EN_R) \\ &= \sum_{l=1}^{n-k} p_l S(P_X^{(1)} \dots P_X^{(l)} | E, N_R = l) + p_\perp S(X|E, N_R = \perp) \end{aligned} \quad (3.18)$$

Note that F and N_R are equivalent in that $N_R = j \Leftrightarrow F_i = 0 \ \forall i < j, \ F_j = 1, \ F_i = \perp \ \forall i > j$.

The error-correction cost can be computed in a similar way. In each round j , where $1 \leq j \leq n - k$, Alice sends $S(P_X^{(j)} | Y P_X^{(1)} \dots P_X^{(j-1)} F_1 = 0, \dots, F_{j-1} = 0)$ bits per block *for only those blocks whose previous parities had no error*. The probability of a block having no error until round j is given by $1 - \sum_{i=1}^{j-1} p_{j-1}$. In the last round, Alice sends $S(X | Y P_X, F_1 = 0, \dots, F_{n-k} = 0)$ bits per block, and a given block reaches this round with probability $p_\perp$. Therefore, the number of bits per block sent during error correction in each round can be written as

$$\begin{aligned} \delta_{\text{leak}}^{(j)} &= (1 - p_1 - \dots p_{j-1}) S(P_X^{(j)} | Y P_X^{(1)} \dots P_X^{(j-1)}, F_1 = 0, \dots, F_{j-1} = 0) \\ \delta_{\text{leak}}^{(\perp)} &= p_\perp S(X | Y P_X, F_1 \dots F_{n-k} = 0) \end{aligned} \quad (3.19)$$

More details about how the above expressions are actually computed can be found in Appendix D.

3.3.3 Results

We test all choices of H for n up to 3. For both BB84, and six-state, we find only one choice (upto permutation of columns) improves over our benchmarks. Due to collective attack assumption, H is equivalent upto permutation of columns. The results are plotted in Figs. 3.6, 3.7. Again, it is difficult to do an exhaustive search over all choices of H for higher n , even though the keyrate for any individual H up to $n = 10$ can be computed. We did compute keyrates for a number of values of H for large n , however we did not obtain any interesting results there.

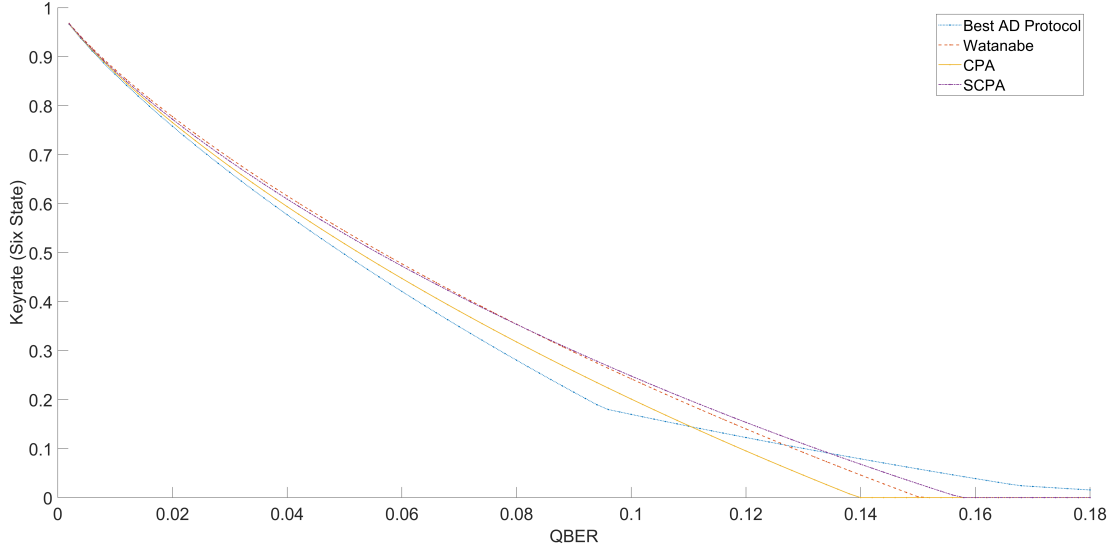


Figure 3.6: Six-state keyrate for the SCPA protocol with $H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. We find that the SCPA keyrate is higher than both our benchmarks for some range of QBER.

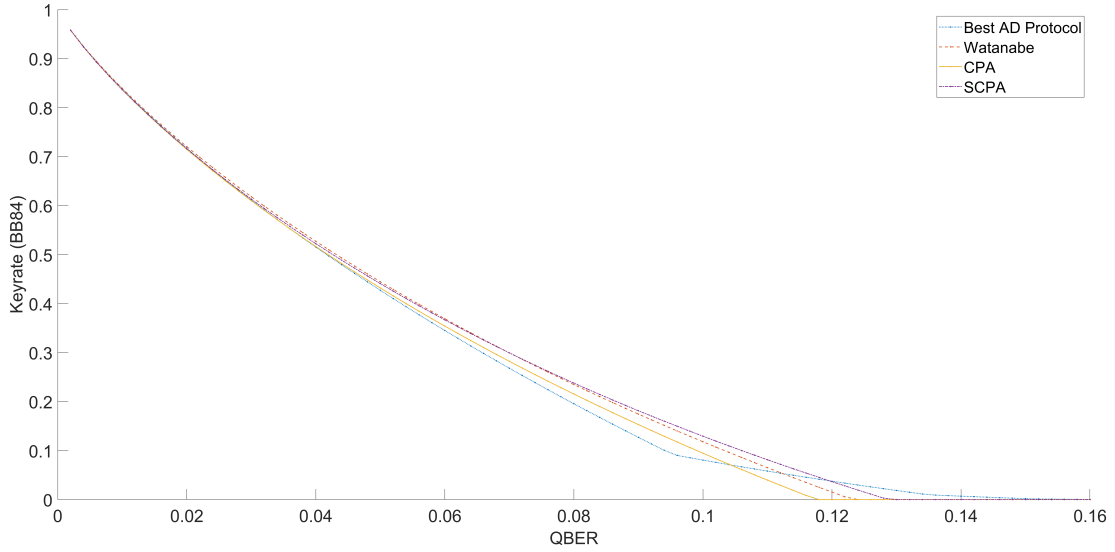


Figure 3.7: BB84 keyrate for the SCPA protocol with $H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. We find that the SCPA keyrate is higher than both our benchmarks for some range of QBER.

Chapter 4

Protocols for Classical Processing in QKD - 2

In this chapter, we will look at two new classical protocols proposed in this thesis. For each of these protocols, we will first explain the protocol description, followed by the computation of keyrates and results. Both these protocols cannot be easily fitted into the IID blockwise processing framework, and thus we need to invent a fresh approach to compute their keyrates. The first of these protocols was proposed and studied by Lietzen et al [19], with a weaker security proof against individual attacks, where Eve is forced to measure her registers after the signals are transmitted. We are able to provide a full security proof and compute the keyrates for collective attacks.

4.1 Recursive Protocol

One can build a new protocol by using the CPA protocol as a building block and applying it recursively. We start with the CPA protocol with $H = [1, 1]$, where the parity string is the parity of two-bit blocks, i.e, the Watanabe protocol [39]. Each two bit block x_1, x_2 can be viewed as the parity bit $x_1 \oplus x_2$, and the ‘remaining’ bit x_1 . One can view this protocol as one which always maps the parity bits to the key, but maps the remaining bit to the key depending on feedback from Bob.

In the Recursive protocol, Alice always maps the parity bits of two-bit blocks to the key (after sending error-correction information for it), and receives Bob’s feedback. However, depending on Bob’s feedback, instead of mapping the remaining bit to the key directly,

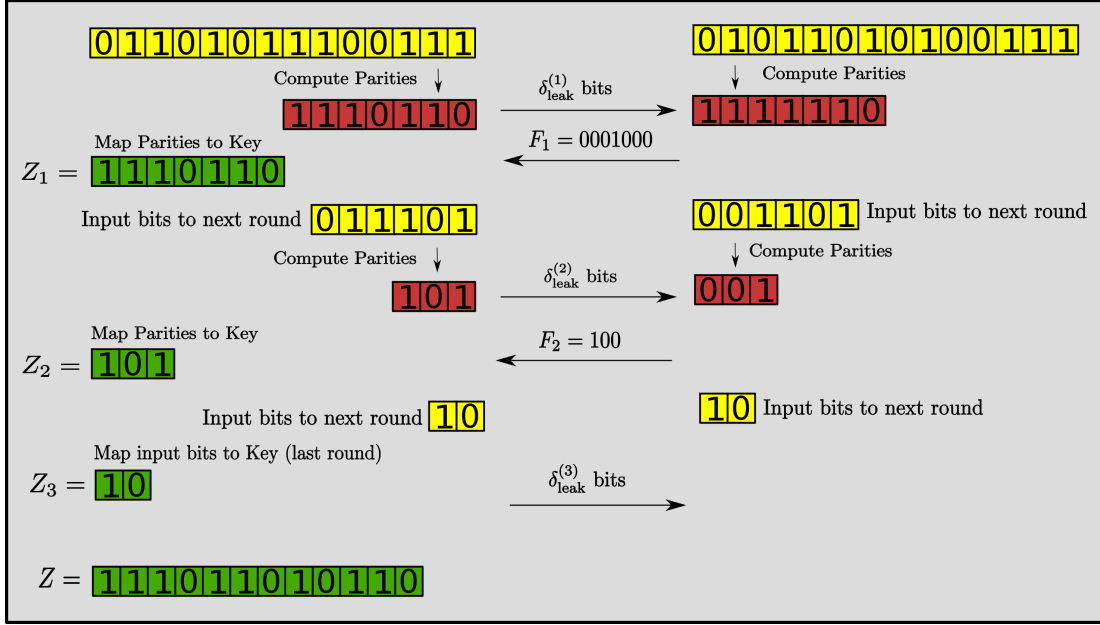


Figure 4.1: An example of the Recursive protocol for $t = 3$.

Alice instead “promotes” it to the next round of the Recursive protocol. Thus, the remaining bits in each block whose parity matched are put together into a new string to be run through the next round.

This protocol was first studied in [19]. However their analysis assumes Eve measures her quantum register before the classical protocol starts, while we compute keyrates in the general setting without this assumption.

4.1.1 Protocol description

- For round j starting from 1 to $t - 1$, Alice and Bob perform the following operations
 - a) Alice and Bob receive a set of input bits from the previous round, that they divide into smaller blocks of size 2. For the first round, the input bits are the raw data.
 - b) Alice computes the parity of each of the blocks, and sends error correction information to Bob. Bob decodes Alice’s parities.
 - c) For each of these blocks, Bob provides feedback to Alice. Bob replies with $F_i^j = 0$ if the i th block in the j th round has matching parities, and $F_i^j = 1$ if not.
 - d) Alice maps all the corrected parities to the key.

e) For the blocks whose parity matched, Alice and Bob promote the first bit of the block as the input bits to the next round.

- In the terminal round t , Alice maps all input bits to the key.
- Alice and Bob do EC and PA.

Note that setting $t = 2$ recovers the Watanabe protocol. See Fig. 4.1 for a schematic of the protocol for $t = 3$ rounds.

4.1.2 IID Structure in the Recursive Protocol

Let N be the total number of signals, and let $\vec{Z}$ denote the total key register, while $\vec{E}$ denotes Eve's quantum system, $\vec{C}$ denotes *all* classical communication revealed during error correction, $\vec{F}$ denotes feedback from Bob, and $\vec{A}$ denotes announcements made by Alice. We split up the classical communication this way because they are analyzed differently. For instance $\vec{C}$ is split off by counting the number of bits leaked in it. The rest are not. Note that the Recursive protocol does not require us to have a $\vec{A}$ register, since all of Alice's communication can be modelled in $\vec{C}$. However we will include it in our analysis since it is required in the next protocol we consider.

One can easily see that for this protocol, it is *not* the case that the final state $\rho_{\vec{Z}\vec{E}\vec{F}\vec{A}} = \sigma_{\vec{Z}\vec{E}\vec{F}\vec{A}}^{\otimes l_{\text{key}}}$, for some l_{key} or $\sigma_{\vec{Z}\vec{E}\vec{F}\vec{A}}$. The final state is not IID, and therefore one cannot directly utilize the discussion from Sec. 3.2.2, as is usually done in the analysis of QKD keyrates.

We do have *some* structure in the Recursive protocol. All tests are identical, and all blocks once created are treated identically. However, in each round there is a “re-blocking” of bits into new blocks. These new blocks are not fixed a priori, but depend upon the outcome of tests. Additionally, since the number of blocks passing the test is a random variable, apart from the first round, every other round generates a variable length of the key.

Our approach is as follows : Our task is to compute $H_{\min}^e(\vec{Z}|\vec{E}\vec{A}\vec{F})$. Let $\vec{W} = \vec{X} \oplus \vec{Y}$. Note that since $\vec{F}$ is a function of $\vec{W}$, $H_{\min}^e(\vec{Z}|\vec{E}\vec{A}\vec{F}) \geq H_{\min}^e(\vec{Z}|\vec{E}\vec{A}\vec{W}\vec{F}) = H_{\min}^e(\vec{Z}|\vec{E}\vec{A}\vec{W})$. This means that giving $\vec{W}$ to Eve instead of $\vec{F}$ can only increase her power. Thus the keyrate for a protocol that announces $\vec{W}$ is a valid lower bound keyrate for a protocol that announces $\vec{F}$. Note that $\vec{W}$ is a register that does not physically exist at any point in the protocol. However, giving it to Eve makes our task easier, since it is an IID announcement that we have swapped for the non-IID announcement $\vec{F}$. We will later show that this step does not lead to any reduction in the key rate by our approach.

Let $\vec{Z}_i$ denote the key generated during round i , i.e, the parities of the blocks generated in round i (unless i is the terminal round, in which case $\vec{Z}_i$ is simply equal to the input to round i). Since the number of bits of key generated in round i is not fixed, we assume that Alice pads the key generated with $\perp$ s to make $\vec{Z}_i$ a constant length register¹. Similarly, we also assume that $\vec{A}$ is also padded to maintain constant length. Therefore let us concern ourselves with the structure of the state $\rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W}}$, which will be necessary to compute our key rate. Our state at the start of the classical processing protocol can be written as $\rho_{XYEW}^{\otimes N}$. We will start by considering the part of the key generated in round i . Let L_i be the classical register denoting the number of bits of key in $\vec{Z}_i$. In general, $\rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W}} = \sum_l \text{Prob}(L_i = l) \rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W} | L_i = l}$.

If $L_i = l$, we know there are l bits mapped to the key in the i th round. Each of these key bits (assuming i is not the terminal round), is the parity of some two bits input to round i , which can be seen from step (a) of our protocol description. Each of these two bits in turn came from some two bit block of round $(i - 1)$, for a total of 4 bits input to round $i - 1$. Thus, each bit in $\vec{Z}_i$ is obtained from the processing of *some* 2^i bit block of the input signals. Although we do not know a priori the location of the signals that make up this block, we will see that the locations do not matter. Each bit in $\vec{Z}_i$ is generated from a *disjoint* set of 2^i signals. Let us call these the “primary” blocks for round i . These primary blocks are blocks of signals having errors $\hat{W}^{(i)}$ that pass all the parity matching test up to round i , to generate 1 bit of key in round i . Therefore each 2^i bit block generates 2^{i-1} bits of key in the first round, 2^{i-2} bits in the second round, ..., 2 bits of key in round $i - 1$, and 1 bit of key in round i . Grouping these blocks together, we can rewrite our starting state as

$$\rho_{\vec{X} \vec{Y} \vec{E} \vec{W} | L_i = l} = \rho_{\hat{X}^{(i)} \hat{Y}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(1)} \otimes \rho_{\hat{X}^{(i)} \hat{Y}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(2)} \cdots \otimes \rho_{\hat{X}^{(i)} \hat{Y}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(l)} \otimes \rho_{\vec{X} \vec{Y} \vec{E} \vec{W}}^{\text{rest}, L_i = l} \quad (4.1)$$

where $\rho_{\hat{X}^{(i)} \hat{Y}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(k)}$ is the state on a k th primary block conditioned on $\hat{W}$ passing all parity tests up to round i ². The remaining signals (which do not generate a bit of key in round i) are grouped together in $\rho_{\vec{X} \vec{Y} \vec{E} \vec{W}}^{\text{rest}, L_i = l}$. Note that we have grouped our signals together in such a way that each group of signals (except the last one) generates 1 bit of

¹We require registers to be constant length to have well-defined states and entropies

²The superscript i is necessary here, since the size of the registers $\hat{X}^{(j)}$ depends upon j , which is the round where the block generates 1 bit of key. Strictly speaking, the length of the register $\vec{X}$ also depends on the value of l and j . However, we do not choose to make this explicit since it leads to some very cumbersome notation, and does not affect our final result.

key in round i . After the implementation of the key map, the final state will therefore be

$$\rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W} | L_i=l} = \rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{A}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(1)} \otimes \rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{A}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(2)} \dots \otimes \rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{A}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(l)} \otimes \rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W}}^{\text{rest}, L_i=l} \quad (4.2)$$

where $\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{A}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(k)}$ is the state obtained after processing the k th primary block that generated 1 key bit in round i . $\rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W}}^{\text{rest}, L_i=l}$ is the state obtained from the processing of signals not present in any primary block and includes any padding required to maintain fixed size of registers.

Now, note that due to the IID collective attack assumption, $\rho_{\hat{X}^{(i)} \hat{Y}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{(k)} = \rho_{\hat{X}^{(i)} \hat{Y}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}$ is the same state for all primary blocks. Moreover, Alice implements the *exact same* keymap on each of her primary blocks. Therefore all $\rho_{\hat{Z}_i^{(i)} \hat{A}^{(i)} \hat{E}^{(i)}}^{(k)}$ are in fact identical. Then, one can write :

$$\rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W}, L_i=l} = \rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{A}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{\otimes l} \otimes \rho_{\vec{Z}_i \dots \vec{Z}_1 \vec{A} \vec{E} \vec{W}}^{\text{rest}, L_i=l} \quad (4.3)$$

where $\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{A}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{\otimes l}$ is the result of the keymap operations on $\rho_{\hat{X}^{(i)} \hat{Y}^{(i)} \hat{E}^{(i)} \hat{W}^{(i)}}^{\otimes l}$. Here note that $\vec{X}$ and $(\hat{X}^{\otimes l}, \bar{X})$ are just different permutations of the same register. Finally, the same argument presented above applies even when $i = t$ is the terminal round, except the number of signals in the primary block is now 2^{t-1} .

Crucially, we now have an IID structure that is conditioned on the number of bits of key (or blocks generated) in each round, which is itself not a fixed number. This structure can be exploited to compute the keyrate. At a high level, our approach is as follows:

- 1) Use a chain rule to rewrite the entropy term into terms that describe the key from each round separately. This allows us to utilize the IID structure naturally.
- 2) Show that the event where too few bits of key are generated in any round is a low-probability-event and can be ignored (via a suitable change to the smoothing parameter).
- 3) Show that the case where there are at least so many bits in each round can be analyzed by assuming the worst case scenario.

4.1.3 Technical tools

In this section we will state and prove some results that we use to compute keyrates via the approach described above. The first result below will be used for Step 1 above.

Theorem 4.1 (Chain rule for smooth-min-entropy). Let $\rho \in P_{\leq}(\mathcal{H}_{ABC})$, and $\epsilon \in [0, 1)$. Then

$$H_{\min}^{(6.2^{t-2}-2)\epsilon}(A_t A_{t-1} \dots A_1 | C)_{\rho} \geq \sum_{i=1}^t H_{\min}^{\epsilon}(A_i | A_{i-1} \dots A_1 C)_{\rho} + (t-1) \log(1 - \sqrt{1 - \epsilon^2}) \quad (4.4)$$

Proof : Let $\rho \in P_{\leq}(ABC)$, and $\epsilon, \epsilon', \epsilon'' \in [0, 1)$ with $\epsilon > \epsilon' + 2\epsilon''$, and $\delta = \epsilon - \epsilon' - 2\epsilon''$. Then, Theorem 6.16 from [34] states

$$H_{\min}^{\epsilon}(AB|C)_{\rho} \geq H_{\min}^{\epsilon'}(A|BC)_{\rho} + H_{\min}^{\epsilon''}(B|C)_{\rho} + \log(1 - \sqrt{1 - \delta^2}) \quad (4.5)$$

To get to the chain rule we want, choose $\delta = \epsilon' = \epsilon$. This lets us write

$$H_{\min}^{2\epsilon+2\epsilon''}(AB|C)_{\rho} \geq H_{\min}^{\epsilon}(A|BC)_{\rho} + H_{\min}^{\epsilon''}(B|C)_{\rho} + \log(1 - \sqrt{1 - \epsilon^2}) \quad (4.6)$$

Repeated application of this equation, starting with $\epsilon' = \epsilon$, allows us to write

$$H_{\min}^{(6.2^{t-2}-2)\epsilon}(A_t A_{t-1} \dots A_1 | C)_{\rho} \geq \sum_{i=1}^t H_{\min}^{\epsilon}(A_i | A_{i-1} \dots A_1 C)_{\rho} + (t-1) \log(1 - \sqrt{1 - \epsilon^2}) \quad (4.7)$$

We will use the next theorem for Step 2.

Theorem 4.2. Let $\rho = (1-p)\rho_1 + p\rho_2$, where $\rho, \rho_1, \rho_2 \in D(AB)$. Then, $H_{\min}^{\epsilon+\sqrt{2p}}(A|B)_{\rho} \geq H_{\min}^{\epsilon}(A|B)_{\rho_1}$

Proof : $\Delta(\rho, \rho_1) \leq p$. Therefore $P(\rho, \rho_1) \leq \sqrt{2p}$. Let $H_{\min}^{\epsilon}(A|B)_{\rho_1} = H_{\min}(A|B)_{\sigma_1}$, with $P(\sigma_1, \rho_1) \leq \epsilon$. Therefore, $P(\rho, \sigma_1) \leq \epsilon + \sqrt{2p}$ via the Triangle inequality. Therefore, $H_{\min}^{\epsilon+\sqrt{2p}}(A|B)_{\rho} \geq H_{\min}(A|B)_{\sigma_1} = H_{\min}^{\epsilon}(A|B)_{\rho_1}$.

In some sense since ρ and ρ_1 are states that are close to each, one can replace ρ with ρ_1 while paying a small price in the smoothing parameter.

Theorem 4.3. Let $\rho_{ABY} \in D(ABY)$, where Y is classical. Let $\rho = \sum_y \text{Prob}(Y = y) \rho_{AB}^{(y)} \otimes |y\rangle \langle y|$. Then $H_{\min}^{\sqrt{2\epsilon}}(A|BY)_{\rho} \geq \min_y H_{\min}^{\epsilon}(A|B)_{\rho^{(y)}}$.

Proof : We start from the equivalent statement for min entropy (Eq. 6.25 from [34])

$$H_{\min}(A|BY)_{\rho} = -\log \left(\sum_y \text{Pr}(Y = y) 2^{-H_{\min}(A|B)_{\rho_y}} \right) \quad (4.8)$$

The above expression directly implies

$$H_{\min}(A|BY)_\rho \geq -\log \left(2^{-\min_y H_{\min}(A|B)_{\rho_y}} \right) = \min_y H_{\min}(A|B)_{\rho_y} \quad (4.9)$$

Now, let $H_{\min}^\epsilon(A|B)_{\rho_y} = H_{\min}(A|B)_{\sigma_y}$, with $\Delta(\rho_y, \sigma_y) \leq P(\rho_y, \sigma_y) \leq \epsilon$. Let $\sigma = \sum_y \text{Prob}(Y = y) \sigma_y \otimes |y\rangle \langle y|$. Then,

$$\begin{aligned} \Delta(\rho, \sigma) &= \frac{1}{2} \left\| \sum_y \text{Prob}(Y = y) (\rho_y - \sigma_y) \otimes |y\rangle \langle y| \right\|_1 + \frac{1}{2} \left| \text{Tr} \left(\sum_y \text{Prob}(Y = y) (\rho_y - \sigma_y) \otimes |y\rangle \langle y| \right) \right| \\ &= \frac{1}{2} \sum_y \text{Prob}(Y = y) \|\rho_y - \sigma_y\|_1 + \frac{1}{2} \left| \text{Tr} \left(\sum_y \text{Prob}(Y = y) (\rho_y - \sigma_y) \otimes |y\rangle \langle y| \right) \right| \\ &\leq \sum_y \text{Prob}(Y = y) \frac{1}{2} \|\rho_y - \sigma_y\|_1 + \text{Prob}(Y = y) \frac{1}{2} |\text{Tr}(\rho_y - \sigma_y)| \\ &= \sum_y \text{Prob}(Y = y) \Delta(\rho_y, \sigma_y) \\ &\leq \epsilon \end{aligned} \quad (4.10)$$

Therefore $P(\rho, \sigma) \leq \sqrt{2\epsilon}$. This allows us to obtain the result.

$$H_{\min}^{\sqrt{2\epsilon}}(A|BY)_\rho \geq H_{\min}(A|BY)_\sigma \geq \min_y H_{\min}(A|B)_{\sigma_y} = \min_y H_{\min}^\epsilon(A|B)_{\rho_y} \quad (4.11)$$

Lemma 4.4. For $\rho_1 \in P_{\leq}(\mathcal{H}_{A_1 B_1})$ and $\rho_2 \in P_{\leq}(\mathcal{H}_{A_2 B_2})$, we have

$$H_{\min}(A_1 A_2 | B_1 B_2)_{\rho_1 \otimes \rho_2} \geq H_{\min}(A_1 | B_1)_{\rho_1} + H_{\min}(A_2 | B_2)_{\rho_2} \quad (4.12)$$

Proof : From the definition of min entropy, let

$$\begin{aligned} H_{\min}(A_1 | B_1)_{\rho_1} &= \sup_{\sigma_1 \in P_{\leq}(\mathcal{H}_{B_1})} \sup \{ \lambda_1 : 2^{-\lambda_1} I_{A_1} \otimes \sigma_1 \geq \rho_1 \} \\ H_{\min}(A_2 | B_2)_{\rho_2} &= \sup_{\sigma_2 \in P_{\leq}(\mathcal{H}_{B_2})} \sup \{ \lambda_2 : 2^{-\lambda_2} I_{A_2} \otimes \sigma_2 \geq \rho_2 \} \end{aligned} \quad (4.13)$$

Now, for any $\lambda_1, \lambda_2, \sigma_1, \sigma_2$ that satisfy the feasibility conditions in the above supremums, we have

$$\begin{aligned} 2^{-\lambda_1 - \lambda_2} I_{A_1} \otimes \sigma_1 \otimes I_{A_2} \otimes \sigma_2 &\geq 2^{-\lambda_1} I_{A_1} \otimes \sigma_1 \otimes \rho_2 \\ 2^{-\lambda_1} I_{A_1} \otimes \sigma_1 \otimes \rho_2 &\geq \rho_1 \otimes \rho_2 \end{aligned} \quad (4.14)$$

Therefore, summing the two inequalities we obtain $2^{-\lambda_1 - \lambda_2} I_{A_1} \otimes I_{A_2} \otimes \sigma_1 \otimes \sigma_2 \geq \rho_1 \otimes \rho_2$. The statement follows from the definition of the min-entropy.

Lemma 4.5. For $\rho_1 \in D(\mathcal{H}_{A_1 B_1})$ and $\rho_2 \in D(\mathcal{H}_{A_2 B_2})$, where A_2 is a classical register, we have

$$H_{\min}^\epsilon(A_1 A_2 | B_1 B_2)_{\rho_1 \otimes \rho_2} \geq H_{\min}^\epsilon(A_1 | B_1)_{\rho_1} \quad (4.15)$$

Proof: Let $H_{\min}^\epsilon(A_1 | B_1)_{\rho_1} = H_{\min}(A_1 | B_1)_{\sigma_1}$, for some $\sigma_1 \in P_{\leq}(\mathcal{H}_{A_1 B_1})$ such that $P(\rho_1, \sigma_1) \leq \epsilon$. Then note that

$$\begin{aligned} \sqrt{F_*(\rho_1 \otimes \rho_2, \sigma_1 \otimes \rho_2)} &= \text{Tr}|\sqrt{\rho_1 \otimes \rho_2} \sqrt{\sigma_1 \otimes \rho_2}| + \sqrt{(1 - \text{Tr}(\rho_1 \otimes \rho_2))(1 - \text{Tr}(\sigma_1 \otimes \rho_2))} \\ &= \text{Tr}|\sqrt{\rho_1 \sigma_1} \otimes \rho_2| + \sqrt{(1 - \text{Tr}(\rho_1))(1 - \text{Tr}(\sigma_1))} \\ &= \text{Tr}|\sqrt{\rho_1 \sigma_1}| + \sqrt{(1 - \text{Tr}(\rho_1))(1 - \text{Tr}(\sigma_1))} \\ &= \sqrt{F_*(\rho_1, \sigma_1)} \end{aligned} \quad (4.16)$$

since ρ_1 and ρ_2 are normalized. Therefore $P(\rho_1 \otimes \rho_2, \sigma_1 \otimes \rho_2) = P(\rho_1, \sigma_1) \leq \epsilon$. Therefore, we have

$$\begin{aligned} H_{\min}^\epsilon(A_1 A_2 | B_1 B_2)_{\rho_1 \otimes \rho_2} &\geq H_{\min}(A_1 A_2 | B_1 B_2)_{\sigma_1 \otimes \rho_2} \\ &\geq H_{\min}(A_1 | B_1)_{\sigma_1} + H_{\min}(A_2 | B_2)_{\rho_2} \\ &\geq H_{\min}(A_1 | B_1)_{\sigma_1} \\ &= H_{\min}^\epsilon(A_1 | B_1)_{\rho_1} \end{aligned} \quad (4.17)$$

where we have used Lemma 4.4 in the second line, along with the fact that $H_{\min}(A_2 | B_2)_{\rho_2} \geq 0$ since ρ_2 is classical on A_2 .

We now turn to computing the keyrate for the Recursive protocol.

4.1.4 Keyrate analysis

Let N be the total number of signals, and let $\vec{Z}_i$ be the total key generated in round i , including padding. Let $\vec{W} = \vec{X} \oplus \vec{Y}$. We let $\vec{E}$, $\vec{A}$, $\vec{F}$ and $\vec{C}$ denote Eve's quantum system, Alice's announcements, Bob's feedback and error-correction information respectively. To compute the asymptotic keyrate, we seek a lower bound on $\lim_{N \rightarrow \infty} \frac{H_{\min}^\epsilon(\vec{Z} | \vec{E} \vec{A} \vec{W} \vec{C})}{N}$.

As usual, we split off the error-correction information using Lemma 6.18 from [34] (or Lemma 2.18). We can write

$$H_{\min}^\epsilon(\vec{Z} | \vec{E} \vec{A} \vec{W} \vec{C}) \geq H_{\min}^\epsilon(\vec{Z} | \vec{E} \vec{W} \vec{A}) - \log(\dim(\vec{C})) \quad (4.18)$$

Let

$$\lim_{N \rightarrow \infty} \frac{\log(\dim(\vec{C}))}{N} = \delta_{\text{leak}}. \quad (4.19)$$

δ_{leak} can be estimated by counting the number of bits leaked during error correction. We now wish to obtain a lower bound on $H_{\min}^\epsilon(\vec{Z}|\vec{E}\vec{W}\vec{A})$. Let there be t rounds in the Recursive protocol. The first term in Eq. 4.18 can be split up using Theorem 4.1, to obtain

$$H_{\min}^{(6.2^{t-2}-2)\epsilon}(\vec{Z}_t\vec{Z}_{t-1}\dots\vec{Z}_1|\vec{E}\vec{W}\vec{A})_\rho \geq \sum_{i=1}^t H_{\min}^\epsilon(\vec{Z}_i|\vec{Z}_{i-1}\dots\vec{Z}_1\vec{E}\vec{W}\vec{A})_\rho + (t-1)\log(1-\sqrt{1-\epsilon^2}) \quad (4.20)$$

The last term is finite and vanishes when we divide by N and take the $N \rightarrow \infty$ limit. We would now like to bound each of the terms $H_{\min}^\epsilon(\vec{Z}_i|\vec{Z}_{i-1}\dots\vec{Z}_1\vec{E}\vec{W}\vec{A})_\rho$, where $\vec{Z}_i$ represents the fixed length string mapped to the key in round i . Since $\vec{Z}_i$ consists of a variable number of key bits, it is not obvious how to compute the above smoothed min-entropy. One *could* use Theorem 4.3, and compute the smoothed min-entropy for the worst case value of the number of bits in $\vec{Z}_i$. However, we would then trivially obtain 0, since in the worst case $\vec{Z}_i$ contains zero key bits because all parities in the previous round contained errors.

To avoid this problem, we choose to improve the worst case value of the number of bits in $\vec{Z}_i$ as follows. Let L_i be the number of bits in $\vec{Z}_i$. Then the state conditioned on a value of L_i (as explained in the previous section) looks like

$$\rho_{L_i=l} = \rho_{\hat{Z}_i^{(i)}\dots\hat{Z}_1^{(i)}\hat{E}^{(i)}\hat{W}^{(i)}\hat{A}^{(i)}}^{\otimes l} \otimes \rho_{\vec{Z}_{i-1}\dots\vec{Z}_i\vec{E}\vec{W}\vec{A}}^{(l)} \quad (4.21)$$

Now, we can divide the full state into two parts as

$$\begin{aligned} \rho_{\vec{Z}_i\dots\vec{Z}_1\vec{E}\vec{W}\vec{A}} &= \sum_l \Pr(L_i=l) \rho_{\hat{Z}_i^{(i)}\dots\hat{Z}_1^{(i)}\hat{E}^{(i)}\hat{W}^{(i)}\hat{A}^{(i)}}^{\otimes l} \otimes \rho_{\vec{Z}_{i-1}\dots\vec{Z}_i\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle\langle l| \\ &= \sum_{l \geq l_i} \Pr(L_i=l) \rho_{\hat{Z}_i^{(i)}\dots\hat{Z}_1^{(i)}\hat{E}^{(i)}\hat{W}^{(i)}\hat{A}^{(i)}}^{\otimes l} \otimes \rho_{\vec{Z}_{i-1}\dots\vec{Z}_i\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle\langle l| \\ &\quad + \sum_{l < l_i} \Pr(L_i=l) \rho_{\hat{Z}_i^{(i)}\dots\hat{Z}_1^{(i)}\hat{E}^{(i)}\hat{W}^{(i)}\hat{A}^{(i)}}^{\otimes l} \otimes \rho_{\vec{Z}_{i-1}\dots\vec{Z}_i\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle\langle l| \\ &= (1-\alpha)\rho_{L_i \geq l_i} + \alpha\rho_{L_i < l_i} \end{aligned} \quad (4.22)$$

where $\rho_{L_i \geq l_i} = \sum_{l \geq l_i} \frac{\Pr(L_i=l)}{(1-p)} \rho_{\hat{Z}_i^{(i)}\dots\hat{Z}_1^{(i)}\hat{E}^{(i)}\hat{W}^{(i)}\hat{A}^{(i)}}^{\otimes l} \otimes \rho_{\vec{Z}_{i-1}\dots\vec{Z}_i\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle\langle l|$, and $\rho_{L_i < l_i} = \sum_{l < l_i} \frac{\Pr(L_i=l)}{p} \rho_{\hat{Z}_i^{(i)}\dots\hat{Z}_1^{(i)}\hat{E}^{(i)}\hat{W}^{(i)}\hat{A}^{(i)}}^{\otimes l} \otimes \rho_{\vec{Z}_{i-1}\dots\vec{Z}_i\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle\langle l|$, and $\alpha = \Pr(L_i < l_i)$. For a large number of signals, we will later show how to choose an l_i such that the probability of $L_i \leq l_i$ is small. Since it is a low-probability event, we can then replace ρ with ρ_1 in our smoothed min-entropy calculations, with a small increase of the smoothing parameter.

Let us suppose that $\alpha \leq \alpha_i$ is known. Using Theorem 4.2, and the fact that the smoothed min entropy is an increasing function of the smoothing parameter, we have

$$H_{\min}^{\epsilon}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A})_{\rho} \geq H_{\min}^{\epsilon - \sqrt{2\alpha_i}}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A})_{\rho_{L_i \geq l_i}} \quad (4.23)$$

The entropy on $\rho_{L_i \geq l_i}$ is difficult to evaluate due to the variable number of bits in $\vec{Z}_i$. However, we can choose the worst case as a valid lower bound. Finally, we have using Theorem 4.3,

$$\begin{aligned} H_{\min}^{\epsilon - \sqrt{2\alpha_i}}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A})_{\rho_1} &= H_{\min}^{\epsilon - \sqrt{2\alpha_i}}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A} L_i)_{\rho_{L_i \geq l_i}} \\ &\geq \min_{\{l: l \geq l_i\}} H_{\min}^{\epsilon_i}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A})_{\rho_{L_i=l}} \end{aligned} \quad (4.24)$$

where $\epsilon_i = (\epsilon - \sqrt{2\alpha_i})^2/2$. Next, we use Lemma 4.5 and Eq. 4.21 to write to write

$$H_{\min}^{\epsilon_i}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A})_{\rho_{L_i=l}} \geq H_{\min}^{\epsilon_i}(\hat{Z}_i^{(i) \otimes l} | \hat{Z}_{i-1}^{(i) \otimes l} \dots \hat{Z}_1^{(i) \otimes l} \hat{E}^{(i) \otimes l} \hat{W}^{(i) \otimes l} \hat{A}^{(i) \otimes l})_{\rho^{\otimes l}} \quad (4.25)$$

This trivially leads to

$$\min_{\{l: l \geq l_i\}} H_{\min}^{\epsilon_i}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A})_{\rho_{L_i=l}} \geq \min_{\{l: l \geq l_i\}} H_{\min}^{\epsilon_i}(\hat{Z}_i^{(i) \otimes l} | \hat{Z}_{i-1}^{(i) \otimes l} \dots \hat{Z}_1^{(i) \otimes l} \hat{E}^{(i) \otimes l} \hat{W}^{(i) \otimes l} \hat{A}^{(i) \otimes l})_{\rho^{\otimes l}} \quad (4.26)$$

Again, using 4.5, we find that the above minimum is obtained at $l = l_i$. If N is the number of signals sent, the next step is to choose l_i/N to be a constant in such a way that $\epsilon_i > 0$ ³. We will show that such a choice of l_i exists in the next section. Then,

$$\begin{aligned} &\lim_{N \rightarrow \infty} \frac{H_{\min}^{\epsilon}(\vec{Z}_i | \vec{Z}_{i-1} \dots \vec{Z}_1 \vec{E} \vec{W} \vec{A})_{\rho}}{N} \\ &\geq \lim_{N \rightarrow \infty} \frac{l_i}{N} \frac{H_{\min}^{\epsilon_i}(\hat{Z}_i^{(i) \otimes l_i} | \hat{Z}_{i-1}^{(i) \otimes l_i} \dots \hat{Z}_1^{(i) \otimes l_i} \hat{E}^{(i) \otimes l_i} \hat{W}^{(i) \otimes l_i} \hat{A}^{(i) \otimes l_i})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}}}}{l_i} \\ &= \frac{l_i}{N} \lim_{l_i \rightarrow \infty} \frac{H_{\min}^{\epsilon_i}(\hat{Z}_i^{(i) \otimes l_i} | \hat{Z}_{i-1}^{(i) \otimes l_i} \dots \hat{Z}_1^{(i) \otimes l_i} \hat{E}^{(i) \otimes l_i} \hat{W}^{(i) \otimes l_i} \hat{A}^{(i) \otimes l_i})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}}}}{l_i} \\ &\geq \frac{l_i}{N} S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}}} \end{aligned} \quad (4.27)$$

³Since smoothing with a negative ϵ is not well defined

where we use the asymptotic equipartition property from Theorem 2.23 for the last inequality. Finally, combining the above equations, we arrive at

$$\lim_{N \rightarrow \infty} \frac{H_{\min}^{(6.2^{t-2}-2)\epsilon}(\vec{Z}_t \dots \vec{Z}_1 | \vec{E} \vec{W} \vec{A})_\rho}{N} \geq \sum_{i=1}^t \frac{l_i}{N} S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}}} \quad (4.28)$$

4.1.5 Choosing l_i

For the above argument to work, we need to choose the lower bound l_i to be a constant fraction of N , such that $\Pr(L_i \leq l_i) \leq \alpha_i$ can be upper bounded. We want to choose an l_i that is slightly smaller than the expected value of L_i . Let p_i be the probability of parity matching in round i and let δ_i be some small positive number. Then, we set $l_1 = \frac{N}{2}$, $l_i = \frac{l_{i-1}}{2}(p_{i-1} - \delta_i)$ for $1 < i < t$ and $l_t = l_{t-1}(p_{t-1} - \delta_{t-1})$. Trivially, $\Pr(L_1 \geq N/2) = 1$. Note that L_i is exactly equal to the number of parities matching in round $i-1$. Therefore, for $1 < i < t$ we obtain

$$\begin{aligned} \Pr(L_i \geq l_i) &= \Pr\left(\frac{\# \text{ parities matching in round } i-1}{l_{i-1}/2} \geq p_{i-1} - \delta_{i-1}\right) \\ &\geq \Pr\left(\frac{\# \text{ parities matching in round } i-1}{l_{i-1}/2} \geq p_{i-1} - \delta_{i-1} \mid L_{i-1} \geq l_{i-1}\right) \Pr(L_{i-1} \geq l_{i-1}) \end{aligned} \quad (4.29)$$

When $L_i \geq l_i$, we can write

$$\begin{aligned} \Pr(L_i \geq l_i) &\geq \Pr\left(\frac{\# \text{ parities matching in round } i-1}{L_{i-1}/2} \geq p_{i-1} - \delta_{i-1} \mid L_{i-1} \geq l_{i-1}\right) \Pr(L_{i-1} \geq l_{i-1}) \\ &\geq (1 - e^{-D(p_{i-1} - \delta_{i-1} \parallel p_{i-1})l_{i-1}/2}) \Pr(L_{i-1} \geq l_{i-1}) \end{aligned} \quad (4.30)$$

where $D(a \parallel b)$ is the Kullback–Leibler divergence. Here we have used the Chernoff bound from theorem 2.19. Similarly, for the terminal round,

$$\begin{aligned} \Pr(L_t \geq l_t) &= \Pr\left(\frac{\# \text{ parities matching in round } t-1}{l_{t-1}} \geq p_{t-1} - \delta_{t-1}\right) \\ &\geq \Pr\left(\frac{\# \text{ parities matching in round } t-1}{l_{t-1}} \geq p_{t-1} - \delta_{t-1} \mid L_{t-1} \geq l_{t-1}\right) \Pr(L_{t-1} \geq l_{t-1}) \\ &\geq (1 - e^{-D(p_{t-1} - \delta_{t-1} \parallel p_{t-1})l_{t-1}}) \Pr(L_{t-1} \geq l_{t-1}) \end{aligned} \quad (4.31)$$

Therefore, for any choice of positive δ_i s, we have $\lim_{N \rightarrow \infty} \alpha_i = 0$. Thus, we can always choose an N large enough that, $\epsilon_i = (\epsilon - \sqrt{2\alpha_i})^2/2$ is always positive. Now, using Eqs. 4.28, and our choice of l_i we can write

$$\begin{aligned}
\lim_{N \rightarrow \infty} \frac{H_{\min}^{(6 \cdot 2^{t-2}-2)\epsilon}(\vec{Z}_t \dots \vec{Z}_1 | \vec{E} \vec{W} \vec{A})_\rho}{N} &\geq \sum_{i=1}^t \frac{l_i}{N} S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}}} \\
&= \sum_{i=1}^{t-1} \frac{1}{2^i} \left(\prod_{j=1}^{i-1} p_{j-1} - \delta_{j-1} \right) S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}}} \\
&\quad + \frac{1}{2^{t-1}} \left(\prod_{j=1}^{t-1} p_{j-1} - \delta_{j-1} \right) S(\hat{Z}_t^{(t)} | \hat{Z}_{t-1}^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{W}^{(t)} \hat{A}^{(t)})_{\rho_{\hat{Z}_t^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{W}^{(t)} \hat{A}^{(t)}}}
\end{aligned} \tag{4.32}$$

for any choice of $\delta_i \geq 0$. We can write

$$\begin{aligned}
\lim_{N \rightarrow \infty} \frac{H_{\min}^{(6 \cdot 2^{t-2}-2)\epsilon}(\vec{Z}_t \dots \vec{Z}_1 | \vec{E} \vec{W} \vec{A})_\rho}{N} &\geq \\
&\sum_{i=1}^{t-1} \frac{1}{2^i} \left(\prod_{j=1}^{i-1} p_{j-1} \right) S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}}} \\
&\quad + \frac{1}{2^{t-1}} \left(\prod_{j=1}^{t-1} p_{j-1} \right) S(\hat{Z}_t^{(t)} | \hat{Z}_{t-1}^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{W}^{(t)} \hat{A}^{(t)})_{\rho_{\hat{Z}_t^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{W}^{(t)} \hat{A}^{(t)}}}
\end{aligned} \tag{4.33}$$

The coefficients of the conditional entropies are simply the expected length of the key mapped in that round as a fraction of the number of input signals. The conditional entropies can be interpreted as a measure of Eve's uncertainty about the key in that round.

4.1.6 Final Keyrate expression

We will now turn our attention to the computation of δ_{leak} . Here we use the fact that correcting a string of L bits with error rate e , requires $Lh(e)$ bits to be sent from Alice to Bob, where h is the binary entropy function.

Let e_j be the error rate in the bits in round j , then $e_1 = Q_Z$. The error rate of bits in the next round can be shown to be given by $e_{j+1} = \frac{e_j^2}{e_j^2 + (1-e_j)^2}$. The probability of parity matching in round i is given by $p_i = e_i^2 + (1-e_i)^2$, and the parities have an error rate of

$2e_i(1 - e_i)$. Therefore, correcting parities in round i requires the leakage of $h(2e_i(1 - e_i))$ bits per parity, while correcting the key from the last round requires $h(e_t)$ bits per bit of key. Therefore, assuming optimal error correction, the δ_{leak} term can be estimated⁴ via

$$\delta_{\text{leak}} = \sum_{i=1}^{t-1} \frac{1}{2^i} \left(\prod_{j=1}^{i-1} p_{j-1} \right) h(2e_i(1 - e_i)) + \frac{1}{2^{t-1}} \left(\prod_{j=1}^{t-1} p_{j-1} \right) h(e_t) \quad (4.34)$$

Thus, using Eqs. 4.33 and 4.34, we have

$$\begin{aligned} K_{\text{Recursive}} = \min_{\rho \in \Gamma} & \left(\sum_{i=1}^{t-1} \frac{1}{2^i} \left(\prod_{j=1}^{i-1} p_{j-1} \right) S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{W}^{(i)} \hat{A}^{(i)}} \right. \\ & \left. + \frac{1}{2^{t-1}} \left(\prod_{j=1}^{t-1} p_{j-1} \right) S(\hat{Z}_t^{(t)} | \hat{Z}_{t-1}^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{W}^{(t)} \hat{A}^{(t)})_{\rho_{\hat{Z}_t^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{W}^{(t)} \hat{A}^{(t)}} - \delta_{\text{leak}} \right) \end{aligned} \quad (4.35)$$

Without loss of generality, we can restrict our attention to Bell-diagonal states in the above minimization (see Appendix. A). Since for Bell-diagonal states, $\hat{E}$ is already block diagonal in $\hat{W}$, and Eve anyway knows the location of errors, we do not need to include the W register. Therefore, we obtain

$$\begin{aligned} K_{\text{Recursive}} = \min_{\rho \in \Gamma_{\text{bell}}} & \left(\sum_{i=1}^{t-1} \frac{1}{2^i} \left(\prod_{j=1}^{i-1} p_{j-1} \right) S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{A}^{(i)}} \right. \\ & \left. + \frac{1}{2^{t-1}} \left(\prod_{j=1}^{t-1} p_{j-1} \right) S(\hat{Z}_t^{(t)} | \hat{Z}_{t-1}^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{A}^{(t)})_{\rho_{\hat{Z}_t^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{A}^{(t)}} - \delta_{\text{leak}} \right) \end{aligned} \quad (4.36)$$

We refer the reader to Appendix E for details on how to compute the above expression.

4.1.7 Results

We find small improvements over benchmarks for both six-state and BB84 (Fig 4.2, 4.3). Since the error rate is roughly squared in each round of the Recursive protocol, the error rate of the bits in later rounds is already extremely small. Therefore, the bits in later

⁴We can make the error correction computation more rigorous by conditioning on the number of blocks in each round. We choose to do it in a simpler way since it should be clear how to do this from the above subsection. Additionally, the number of bits leaked during error-correction can simply be ‘counted’ in any actual experiment.

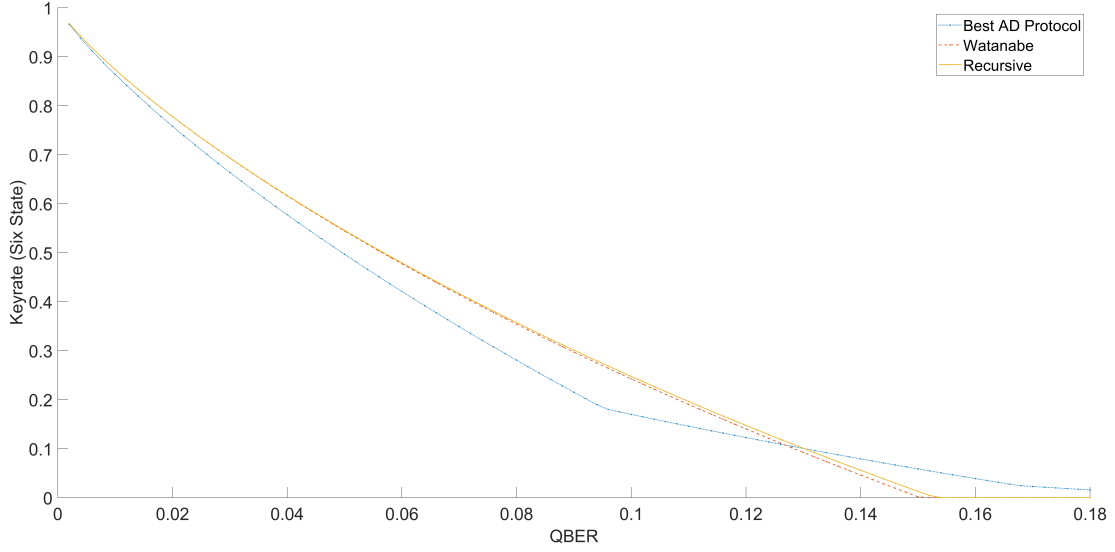


Figure 4.2: Six-state keyrate for the recursive protocol with 4 rounds. We find improvements over our benchmarks for a large range of QBER.

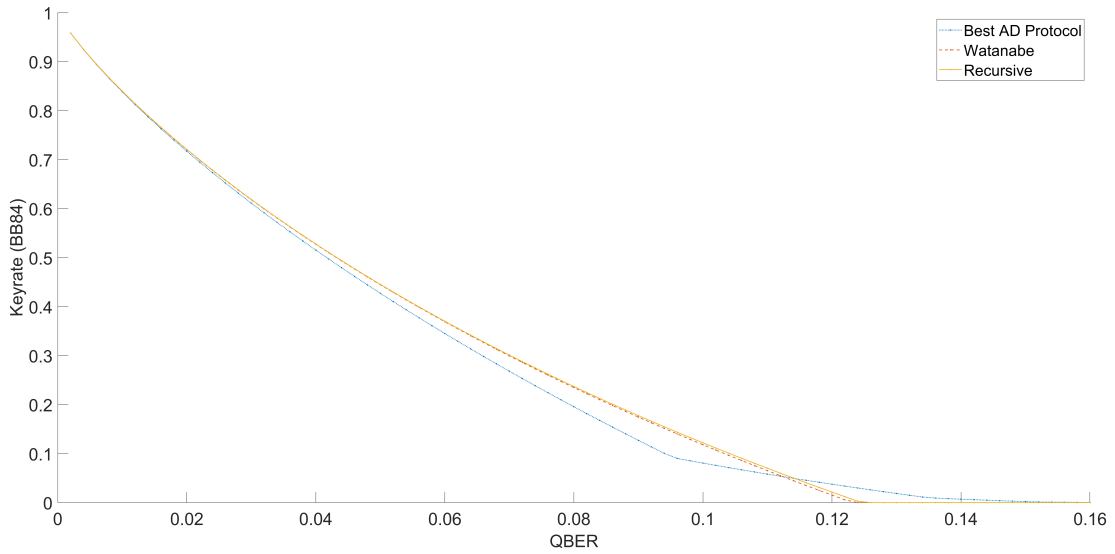


Figure 4.3: BB84 keyrate for the recursive protocol with 4 rounds. We find improvements over our benchmarks for a large range of QBER.

rounds can be corrected with minimal error correction cost. Since the number of bits are also more than halved in each round, the effect of latter rounds on the total keyrate decreases. Thus we find that beyond a small number of rounds, the keyrate does not change significantly with the number of rounds. Note that setting the number of rounds to 2 recovers the Watanabe protocol. In [19], the number of rounds were chosen to be dependent on QBER in a manner that is meaningful only in the case of individual attacks.

In our analysis, we chose to give Eve access to $\vec{W}$ instead of just $\vec{F}$. This approach allowed us to replace a more complicated announcement with a simpler IID announcement, while still computing a lower bound on the secure key rate. We note here that this step turns out to be tight for our analysis, and does not lead us to lose keyrate. One can show this by considering our final keyrate expression in Eq. 4.36. It can be shown that it is exactly the same expression that one would arrive at, if we gave Eve access to *neither* $\vec{F}$ *nor* $\vec{W}$, and Alice implemented the same keymap. Since this key rate is an upper bound on the keyrate when Eve is given access to $\vec{F}$, we see that our choice did not lead us to lose keyrate.

4.2 AD+CPA

All the previous protocols we studied did not yield improvements in the medium to high QBER range. Since we know repetition-code AD is capable of producing keys at high QBER [28], one can combine repetition-code AD and the CPA protocol, as we do below. In the next and final protocol we study, Alice and Bob first do a round of repetition-code advantage-distillation on their data, which ends with some bits being selected for future rounds and some bits discarded. Next, Alice and Bob perform the simplest non-trivial CPA protocol with $H = [1, 1]$ on the remaining bits.

4.2.1 Protocol Description

- Alice and Bob divide their data into b bit blocks. For each of these b bit blocks, Alice and Bob perform the advantage distillation protocol as follows

- a) Let H_{AD} be the $(b-1) \times b$ parity check matrix associated with the linear code space spanned by a single basis vector $(1111\dots)_b$ times, and let $x = x_1 \dots x_b$ be the b bit block. For every b bit block, Alice announces $\tilde{A} = H_{AD}x$ publicly, and Bob compares it with $H_{AD}y$, replying with $F_{AD} = 0$ if there is match and $F_{AD} = 1$ if there is not. This is exactly equivalent to the repetition code advantage distillation studied in [28].

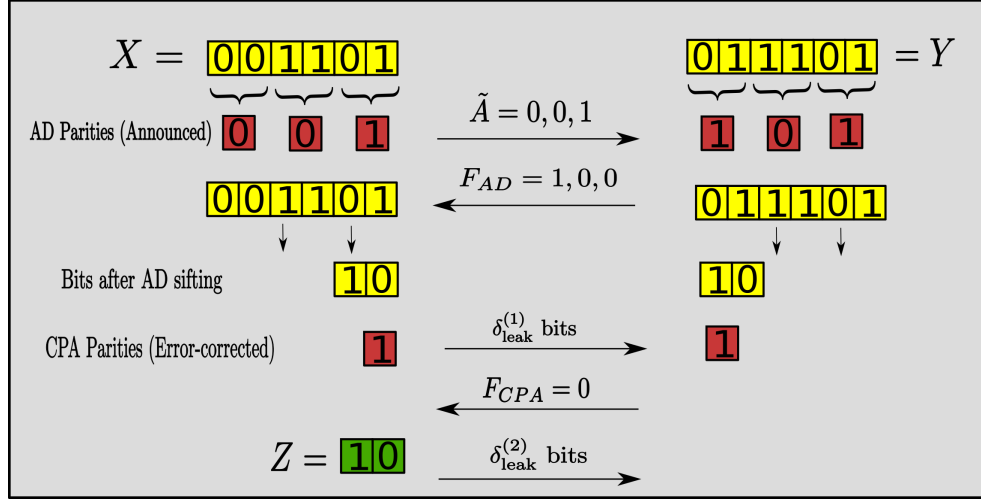


Figure 4.4: An example of the AD+CPA protocol

b) Alice and Bob keep the first bit of each block that is accepted ($F_{AD} = 0$). The remaining blocks are thrown away.

- Alice and Bob now have bit strings which have been filtered through AD protocol. They divide up these bits into blocks of two.
- Alice and Bob perform a CPA protocol on these two bit blocks, as follows

a) Alice computes the parity of the two bits in each block, and sends error correction to Bob. Bob guesses Alice's parity, and for each two-bit block responds with $F_{CPA} = 0$ if the parity matches, and $F_{CPA} = 1$ if they do not.

b) For every two-bit block, if $F_{CPA} = 0$ Alice maps both bits to the key, else she maps the parity to the key. We assume that Alice will pad her key with $\perp$ s to maintain constant length.

- Alice and Bob perform EC and PA
- See Fig. 4.4 for an example of this protocol for $b = 2$.

4.2.2 Keyrate Analysis

Our protocol does not implement IID blockwise processing, since Alice and Bob re block their data after the AD round. Thus, we must turn to the approach from Sec. 4.1.2. It is best to view this protocol as having two rounds : AD and CPA, and the entire key is generated from the CPA round. In this case, let L_1 be the number of b bit blocks generated in the AD round, and let L_2 be the number of two-bit blocks generated in the CPA round.

Each CPA block generates either 1 or 2 bits of raw key depending upon feedback from Bob. Note that the entire key is generated from round 2, therefore $\vec{Z} = \vec{Z}_2$.

For sake of completeness we repeat the arguments from Sec. 4.1.2. Let N be the total number of signals, and let $\vec{Z}$ denote the total key register, while $\vec{E}$ denotes Eve's quantum system, $\vec{C}$ denotes *all* classical communication revealed during error correction, $\vec{F}$ denotes feedback from Bob, and $\vec{A}$ denotes announcements made by Alice. Our task is to compute $H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{A}\vec{F})$. Let $\vec{W} = \vec{X} \oplus \vec{Y}$. Note that since $\vec{F}$ is a function of $\vec{W}$, $H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{A}\vec{F}) \geq H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{A}\vec{W}\vec{F}) = H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{A}\vec{W})$

The protocol only generates a key in round 2. Therefore let us concern ourselves with the structure of the state $\rho_{\vec{Z}\vec{A}\vec{E}\vec{W}}$, where it is understood that $\vec{Z}$ is infact the key generated in round 2. We assume that Alice pads $\vec{Z}$ and $\vec{A}$ to maintain constant lengths. Our state at the start of the classical processing protocol can be written as $\rho_{XYEW}^{\otimes N}$. Each block generated in the CPA round is obtained from some *disjoint* set of $2b$ signals, processed identically. We refer to such a set of signals as a “primary” block⁵. Since we assume IID collective attacks, the input state can be written as

$$\rho_{\vec{X}\vec{Y}\vec{E}\vec{W}|L_2=l} = \rho_{\hat{X}\hat{Y}\hat{E}\hat{W}}^{\otimes l} \otimes \rho_{\vec{X}\vec{Y}\vec{E}\vec{W}}^{\text{rest}, L_2=l} \quad (4.37)$$

where $\rho_{\hat{X}\hat{Y}\hat{E}\hat{W}}^{(k)}$ is the state on a k th primary block conditioned on $\hat{W}$ passing all parity tests of the AD round. The remaining signals (which do not generate a block in the CPA round) are grouped together in $\rho_{\vec{X}\vec{Y}\vec{E}\vec{W}}^{\text{rest}, L_2=l}$. Note that we have grouped our signals together in such a way that each group of signals except the last consists of $2b$ signals and generates 1 *block* in the CPA round (not one key bit). After the keymap, we obtain

$$\rho_{\vec{Z}\vec{A}\vec{E}\vec{W}, L_2=l} = \rho_{\hat{Z}\hat{A}\hat{E}\hat{W}}^{\otimes l} \otimes \rho_{\vec{Z}\vec{A}\vec{E}\vec{W}}^{\text{rest}, L_i=l} \quad (4.38)$$

where $\rho_{\hat{Z}_i \dots \hat{Z}_1 \hat{A} \hat{E} \hat{W}}^{\otimes l}$ is the result of the keymap operations on $\rho_{\hat{X}\hat{Y}\hat{E}\hat{W}}^{\otimes l}$, and $\rho_{\vec{Z}\vec{A}\vec{E}\vec{W}}^{\text{rest}, L_i=l}$ denotes the result of keymap operations on the rest of the signals and the result of padding.

⁵We do not need to separately define a primary block of each round of this protocol, unlike the protocol studied previously. This is because the entire key in the current protocol is generated in round 2, and therefore we are concerned with finding primary blocks that generate one block in round 2

Now, we can divide the full state into two parts as

$$\begin{aligned}
\rho_{\vec{Z}\vec{E}\vec{W}\vec{A}L_2} &= \sum_l \Pr(L_i = l) \rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}^{\otimes l} \otimes \rho_{\vec{Z}\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle \langle l| \\
&= \sum_{l \geq l_2} \Pr(L_2 = l) \rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}^{\otimes l} \otimes \rho_{\vec{Z}\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle \langle l| \\
&\quad + \sum_{l < l_2} \Pr(L_2 = l) \rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}^{\otimes l} \otimes \rho_{\vec{Z}\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle \langle l| \\
&= (1 - \alpha) \rho_{L_2 \geq l_2} + \alpha \rho_{L_2 < l_2}
\end{aligned} \tag{4.39}$$

where $\rho_{L_2 \geq l_2} = \sum_{l \geq l_2} \frac{\Pr(L_2=l)}{(1-p)} \rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}^{\otimes l} \otimes \rho_{\vec{Z}\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle \langle l|$, and $\rho_{L_2 < l_2} = \sum_{l < l_2} \frac{\Pr(L_2=l)}{p} \rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}^{\otimes l} \otimes \rho_{\vec{Z}\vec{E}\vec{W}\vec{A}}^{(l)} \otimes |l\rangle \langle l|$, and $\alpha = \Pr(L_2 < l_2)$. For a large number of signals, we will later show how to choose an l_2 such that the probability of $L_2 \leq l_2$ is small. Since it is a low-probability event, we can then replace ρ with ρ_1 in our smooth-min-entropy calculations, with a small increase of the smoothing parameter. Let us suppose that $\alpha \leq \alpha_2$ is known. Using Theorem 4.2, we have

$$H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{W}\vec{A})_{\rho} \geq H_{\min}^{\epsilon - \sqrt{2\alpha_2}}(\vec{Z}|\vec{Z}\vec{E}\vec{W}\vec{A})_{\rho_{L_2 \geq l_2}} \tag{4.40}$$

The entropy on $\rho_{L_2 \geq l_2}$ is difficult to evaluate due to the variable number of bits in $\vec{Z}_2$. However, we can choose the worst case as a valid lower bound. Finally, we have using Theorem 4.3,

$$\begin{aligned}
H_{\min}^{\epsilon - \sqrt{2\alpha_2}}(\vec{Z}|\vec{E}\vec{W}\vec{A})_{\rho_1} &= H_{\min}^{\epsilon - \sqrt{2\alpha_2}}(\vec{Z}|\vec{E}\vec{W}\vec{A}L_i)_{\rho_{L_2 \geq l_2}} \\
&\geq \min_{\{l: l \geq l_2\}} H_{\min}^{\epsilon_2}(\vec{Z}|\vec{E}\vec{W}\vec{A})_{\rho_{L_2=l}}
\end{aligned} \tag{4.41}$$

where $\epsilon_2 = (\epsilon - \sqrt{2\alpha_2})^2/2$. Next, we use Lemma 4.5 and Eq. 4.21 to write

$$\begin{aligned}
\min_{\{l: l \geq l_2\}} H_{\min}^{\epsilon_2}(\vec{Z}|\vec{E}\vec{W}\vec{A})_{\rho_{L_2=l}} &\geq \min_{\{l: l \geq l_2\}} H_{\min}^{\epsilon_2}(\hat{Z}^{\otimes l}|\hat{Z}^{\otimes l}\hat{E}^{\otimes l}\hat{W}^{\otimes l}\hat{A}^{\otimes l})_{\rho^{\otimes l}} \\
&= H_{\min}^{\epsilon_2}(\hat{Z}^{\otimes l_2}|\hat{E}^{\otimes l_2}\hat{W}^{\otimes l_2}\hat{A}^{\otimes l_2})_{\rho^{\otimes l_2}}
\end{aligned} \tag{4.42}$$

If N is the number of signals sent, the next step is to choose l_2/N to be a constant in such

a way that $\epsilon_2 > 0$. To do this, let $l_1 = N/b$, and let $l_2 = \frac{N(p_1 - \delta_1)}{2b}$. Therefore we have

$$\begin{aligned}
\Pr(L_2 \geq l_2) &= \Pr\left(L_2 \geq \frac{N(p_1 - \delta_1)}{2b}\right) \\
&= \Pr\left(\frac{\# \text{ AD blocks passing test}}{N/b} \geq p_1 - \delta_1\right) \\
&\geq 1 - e^{-D(p_1 - \delta_1 \| p_1)N/b} \\
\Pr(L_2 \leq l_2) &\leq \alpha_2 = e^{-D(p_1 - \delta_1 \| p_1)N/b}
\end{aligned} \tag{4.43}$$

where $D(a||b)$ is the Kullback–Leibler divergence. Here we have used the Chernoff bound from Theorem 2.19. Again, since $\lim_{N \rightarrow \infty} \alpha_2 = 0$, for large N , $\epsilon - \sqrt{2\alpha_2} \geq 0$. Therefore, finally we obtain

$$\begin{aligned}
\lim_{N \rightarrow \infty} \frac{H_{\min}^{\epsilon}(\vec{Z}|\vec{E}\vec{W}\vec{A})_{\rho}}{N} &\geq \frac{l_2}{N} \lim_{l_2 \rightarrow \infty} \frac{H_{\min}^{\epsilon_2}(\hat{Z}^{\otimes l_2}|\hat{E}^{\otimes l_2}\hat{W}^{\otimes l_2}\hat{A}^{\otimes l_2})_{\rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}^{\otimes l_2}}}{l_2} \\
&\geq \frac{p_1 - \delta_1}{2b} S(\hat{Z}|\hat{E}\hat{W}\hat{A})_{\rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}}
\end{aligned} \tag{4.44}$$

for any choice of $\delta_1 > 0$.

The error rate in the bits filtered via AD is given by $\tilde{e} = \frac{e^b}{e^b + (1-e)^b}$. The δ_{leak} (number of bits leaked per input signal) contribution can be computed using

$$\delta_{\text{leak}} = \frac{p_1}{2b} h(2\tilde{e}(1 - \tilde{e})) \tag{4.45}$$

Thus the asymptotic keyrate is given by

$$K_{\text{AD+CPA}} = \min_{\rho \in \Gamma} \frac{p_1}{2b} S(\hat{Z}|\hat{E}\hat{W}\hat{A})_{\rho_{\hat{Z}\hat{E}\hat{W}\hat{A}}} - \delta_{\text{leak}} \tag{4.46}$$

Again, without loss of generality we can restrict to bell-diagonal states where W register is redundant. Therefore,

$$K_{\text{AD+CPA}} = \min_{\rho \in \Gamma_{\text{bell}}} \frac{p_1}{2b} S(\hat{Z}|\hat{E}\hat{A})_{\rho_{\hat{Z}\hat{E}\hat{A}}} - \delta_{\text{leak}} \tag{4.47}$$

4.2.3 Results

We find improvements in keyrates are obtained over benchmarks. We plot the most significant ones here, which occurs for an AD blocksize of 2 (see Figs 4.5, 4.6). The improvements for higher values of b are far smaller.

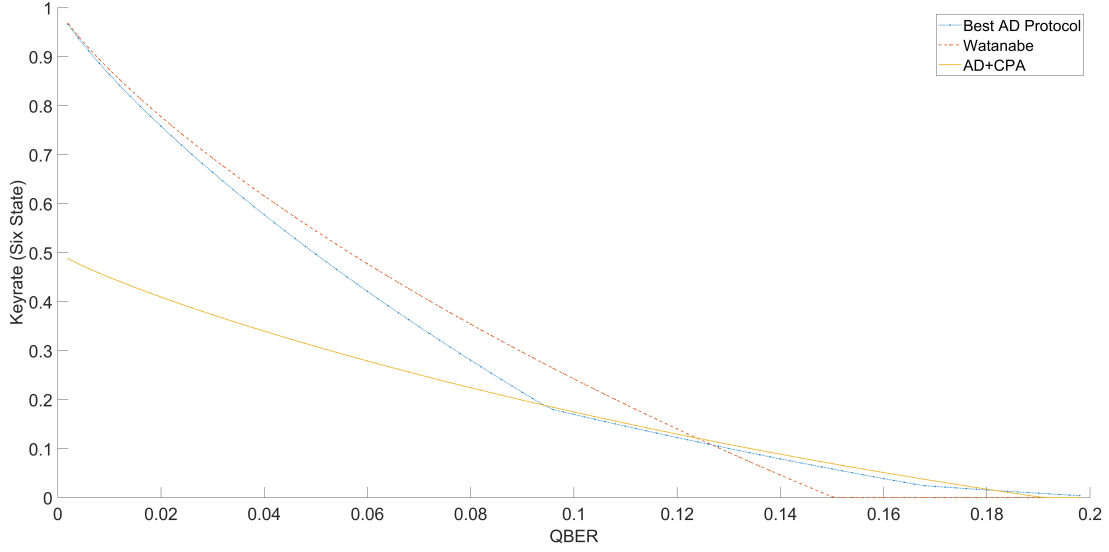


Figure 4.5: Six-state keyrate for the AD+CPA protocol with $b = 2$. We find improvements to our benchmarks over a wide range of QBER.

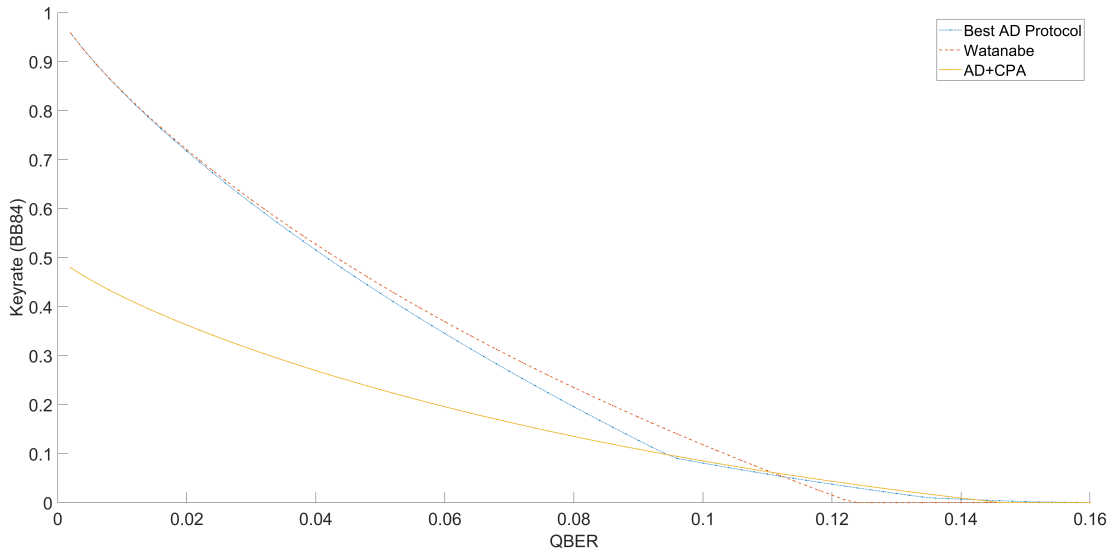


Figure 4.6: BB84 keyrate for the AD+CPA protocol with $b = 2$. We find improvements to our benchmarks over a wide range of QBER.

Chapter 5

Conclusion

In summary, we have proposed new classical protocols that improve the asymptotic key rates for BB84 and six-state protocols. Our protocols use partitioning of data, where Alice and Bob first determine their joint partition status using two-way classical communication. Alice can then decide further actions depending on whether their data belongs to the same subset, or different subsets of the partition. Our protocols also implement error correction instead of direct announcements, which significantly reduces the information leakage to Eve in many cases. Our modifications provide improvements to keyrates and are relatively easy to implement, since they only affect the classical stage of QKD protocols.

We also studied examples of classical processing protocols that do not implement IID blockwise processing. Analysis of such protocols is lacking in the literature. We presented a rigorous analysis of such protocols and their keyrate computations, which is novel and interesting.

Some natural directions for further work include finite size analysis of the protocols, and analysis of more realistic QKD setups involving weak coherent pulses and threshold detectors with the classical protocols proposed in this thesis. If the results of this thesis are taken as a guide, then heuristically, protocols on blocksizes of 3 already yield some significant improvements in keyrates. For such blocksizes, it is possible to use the numerical framework for keyrate computations from [41]. Another important direction for further work is to investigate better bounds to information leakage due to error correction, since our results and those of [39] provide examples where the current method is not tight.

References

- [1] A. Acín, J. Bae, E. Bagan, M. Baig, Ll. Masanes, and R. Muñoz-Tapia. Secrecy properties of quantum channels. *Physical Review A*, 73(1), January 2006.
- [2] Joonwoo Bae and Antonio Acín. Key distillation from quantum channels using two-way communication protocols. *Physical Review A*, 75(1), January 2007.
- [3] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, December 2014.
- [4] Charles H. Bennett, Gilles Brassard, and N. David Mermin. Quantum cryptography without bell’s theorem. *Phys. Rev. Lett.*, 68:557–559, Feb 1992.
- [5] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, Nov 1996.
- [6] Dagmar Bruß. Optimal eavesdropping in quantum cryptography with six states. *Physical Review Letters*, 81(14):3018–3021, October 1998.
- [7] Matthias Christandl, Robert König, and Renato Renner. Postselection technique for quantum channels with applications to quantum cryptography. *Physical Review Letters*, 102(2), January 2009.
- [8] Matthias Christandl and Andreas Winter. “squashed entanglement”: An additive entanglement measure. *Journal of Mathematical Physics*, 45(3):829–840, March 2004.
- [9] T. M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 2006.
- [10] Marcos Curty, Maciej Lewenstein, and Norbert Lütkenhaus. Entanglement as a precondition for secure quantum key distribution. *Phys. Rev. Lett.*, 92:217903, May 2004.

- [11] Igor Devetak and Andreas Winter. Distillation of secret key and entanglement from quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 461(2053):207–235, January 2005.
- [12] Frédéric Dupuis, Omar Fawzi, and Renato Renner. Entropy accumulation. *Communications in Mathematical Physics*, 379(3):867–913, September 2020.
- [13] Artur K. Ekert. Quantum cryptography based on bell’s theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.
- [14] Agnes Ferenczi and Norbert Lütkenhaus. Symmetries in quantum key distribution and the connection between optimal attacks and optimal cloning. *Phys. Rev. A*, 85:052310, May 2012.
- [15] Cong Jiang, Zong-Wen Yu, Xiao-Long Hu, and Xiang-Bin Wang. Higher key rate of measurement-device-independent quantum key distribution through joint data processing. *Physical Review A*, 103(1), January 2021.
- [16] O. Kern and J.M. Renes. Improved one-way rates for BB84 and 6-state protocols. *Quantum Information and Computation*, 8(8&9):756–772, September 2008.
- [17] Sumeet Khatri and Norbert Lütkenhaus. Numerical evidence for bound secrecy from two-way postprocessing in quantum key distribution. *Physical Review A*, 95(4), April 2017.
- [18] B. Kraus, N. Gisin, and R. Renner. Lower and upper bounds on the secret-key rate for quantum key distribution protocols using one-way classical communication. *Phys. Rev. Lett.*, 95:080501, Aug 2005.
- [19] Jari Lietzen, Roope Vehkalahti, and Olav Tirkkonen. A two-way qkd protocol outperforming one-way protocols at low qber. pages 1106–1111, 06 2020.
- [20] Hoi-Kwong Lo. Proof of unconditional security of six-state quantum key distribution scheme, 2001.
- [21] Xiongfeng Ma, Chi-Hang Fred Fung, Frédéric Dupuis, Kai Chen, Kiyoshi Tamaki, and Hoi-Kwong Lo. Decoy-state quantum key distribution with two-way classical postprocessing. *Physical Review A*, 74(3), September 2006.
- [22] U.M. Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

- [23] Tobias Moroder, Marcos Curty, and Norbert Lütkenhaus. One-way quantum key distribution: Simple upper bound on the secret key rate. *Phys. Rev. A*, 74:052301, Nov 2006.
- [24] Tobias Moroder, Marcos Curty, and Norbert Lütkenhaus. Upper bound on the secret key rate distillable from effective quantum correlations with imperfect detectors. *Physical Review A*, 73(1), January 2006.
- [25] Gláucia Murta, Filip Rozpedek, Jérémy Ribeiro, David Elkouss, and Stephanie Wehner. Key rates for quantum key distribution protocols with asymmetric noise. *Physical Review A*, 101(6), June 2020.
- [26] Michael A. Nielsen. *Quantum computation and quantum information - 10. ed.* Cambridge University Press, 2010.
- [27] M. Piani. Relative entropy of entanglement and restricted measurements. *Physical Review Letters*, 103(16), October 2009.
- [28] Renato Renner. Security of quantum key distribution, 2005.
- [29] Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J. Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81:1301–1350, Sep 2009.
- [30] Peter W. Shor and John Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Phys. Rev. Lett.*, 85:441–444, Jul 2000.
- [31] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, 1973.
- [32] Graeme Smith, Joseph M. Renes, and John A. Smolin. Structured codes improve the bennett-brassard-84 quantum key rate. *Physical Review Letters*, 100(17), April 2008.
- [33] Ernest Y.-Z. Tan, Charles C.-W. Lim, and Renato Renner. Advantage distillation for device-independent quantum key distribution. *Physical Review Letters*, 124(2), January 2020.
- [34] Marco Tomamichel. *Quantum Information Processing with Finite Resources*. Springer International Publishing, 2016.

- [35] Marco Tomamichel, Roger Colbeck, and Renato Renner. Duality between smooth min- and max-entropies. *IEEE Transactions on Information Theory*, 56(9):4674–4681, 2010.
- [36] Marco Tomamichel and Anthony Leverrier. A largely self-contained and complete security proof for quantum key distribution. *Quantum*, 1:14, July 2017.
- [37] Alexander Vitanov, Frederic Dupuis, Marco Tomamichel, and Renato Renner. Chain rules for smooth min- and max-entropies. *IEEE Transactions on Information Theory*, 59(5):2603–2612, May 2013.
- [38] Karl Vollbrecht and Frank Verstraete. Interpolation of recurrence and hashing entanglement distillation protocols. *Physical Review A*, 71(6), June 2005.
- [39] Shun Watanabe, Ryutaroh Matsumoto, Tomohiko Uyematsu, and Yasuhito Kawano. Key rate of quantum key distribution with hashed two-way classical communication. *Physical Review A*, 76(3), September 2007.
- [40] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.
- [41] Adam Winick, Norbert Lütkenhaus, and Patrick J. Coles. Reliable numerical key rates for quantum key distribution. *Quantum*, 2:77, July 2018.

APPENDICES

Appendix A

Symmetry : Reduction to Bell-Diagonal states

In this appendix, we present an argument that allows us to restrict our attention to Bell-diagonal states shared between Alice and Bob. We will utilize the symmetry of measurements, parameter estimation and the keymap to show that for any state σ_{AB} shared between Alice and Bob that satisfies the constraints from observations, there exists a Bell-diagonal state $\hat{\sigma}_{AB}$ that also satisfied the constraints and yields a lower (or equal) keyrate. Thus without loss of generality, we can compute the worst keyrate over all Bell-diagonal states satisfying out constraints. The argument we present here is similar to [39, 14], however it is more general since our protocols involve more complicated communication between Alice and Bob. In particular, since our protocols involve multiple rounds of key generation and error correction, we argue our case for every single term that appears in the keyrate equations 3.12, 3.16, 4.35, 4.46. separately.

This reduction to Bell-diagonal states vastly simplifies keyrate calculations, since a normalized Bell-diagonal states is parameterized by just three parameters. For six-state, the three QBERs fix the state entirely, while for BB84 we are left with one parameter to compute the minimum over. The reduction to the Bell-diagonal states is also what allows us to utilize the tools from Appendix B, and to argue that Eve can be assumed to know the location of errors in Alice and Bob's raw data.

Our Protocol

Let us recall the measurements performed by Alice and Bob for the protocols we consider. Alice and Bob receive (a large number of copies of) the state σ_{AB} and measure their respective systems in the X and Z basis. For the six-state protocol, the measure in an additional basis Y . The choice of basis is done randomly and independently by Alice and Bob, with the probability of choosing the Z basis being arbitrarily close to 1.

Given that they measure in all three (or two) basis, Alice and Bob could use the entire probability table they generate accross all measurement outcomes and basis choices to place constraints on Eve's attack, i.e, use coarse-grained constraints. However, in such situations we are unable to argue symmetry [14]. Therefore we utilize fine-grained constraints, where Alice and Bob first sift and through out all rounds where the basis did not match, and then compute Q_X, Q_Z on test rounds (and Q_Y for six-state). Therefore the sifted state after measurement for a given basis is given by

$$\rho_{\text{sifted}}^b = \sum_{x,y} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \text{Tr} \left[(|x, b\rangle \langle x, b| \otimes |y, b\rangle \langle y, b| \otimes I) |\psi\rangle \langle \psi| \right] \quad (\text{A.1})$$

where $|\psi\rangle$ is a purification of σ_{AB} and $|x, b\rangle \langle x, b|$ denotes the projector POVM with outcome x and basis choice b . Note that we include Bob's bit flips for Y measurements inside $|y, b\rangle$ here. The probability distribution for measurement outcomes in the given basis is given by

$$P_b(x, y) = \text{Tr} \left[(|x, b\rangle \langle x, b| \otimes |y, b\rangle \langle y, b|) \sigma_{AB} \right] \quad (\text{A.2})$$

The course grained QBER's lead to the constraints $Q_b = P_b(x \oplus y = 1)$, which is the probability of error given both Alice and Bob measured in the same basis b .

Symmetry and Twirling Map

Let $|\phi_i\rangle$ denote the Bell-states and let the $\langle \phi_i | \rho_{AB} | \phi_i \rangle = \lambda_i$. Further, let σ_i be the Pauli operators with $(\sigma_0, \sigma_1, \sigma_2, \sigma_3) = (I, \sigma_X, \sigma_Y, \sigma_Z)$. Additionally, for $s \in S = \{0, 1, 2, 3\}$, we define

$$\sigma_{AB}^{(s)} = [\sigma_s \otimes \sigma_s] \sigma_{AB} [\sigma_s \otimes \sigma_s]^\dagger \quad (\text{A.3})$$

We then define the "twirled state" [14, 5] given by

$$\hat{\sigma}_{AB} = \frac{1}{4} \sum_{s \in S} \sigma_{AB}^{(s)} = \sum_{i=0}^3 \lambda_i |\phi_i\rangle \langle \phi_i| \quad (\text{A.4})$$

which can be shown to always be Bell-diagonal [14, 5]. We can now generalize the above claims to n qubit states, where the input state to our protocol is $\sigma_{AB}^{\otimes n}$, and

$$\begin{aligned}\sigma_{A^n B^n}^{(s)} &= \bigotimes_{i=1}^n \sigma_{AB}^{(s_i)} \\ \hat{\sigma}_{AB}^{\otimes n} &= \sum_{s \in S^n} \frac{1}{4^n} \sigma_{A^n B^n}^{(s)}\end{aligned}\tag{A.5}$$

where s is now a vector of length n whose entries belong to S . We will now show that given any $\sigma_{AB}^{\otimes n}$ that satisfied the QBER constraints, the twirled state $\hat{\sigma}_{AB}^{\otimes n}$ a) satisfies the same constraints as the original state, and b) has a lower (or equal) keyrate. Since the twirled states are Bell-diagonal, it is enough to consider attacks where Eve shares a Bell-diagonal state.

Symmetry in Constraints

The probability distribution for a given basis and state is given by

$$\begin{aligned}P_b(x, y) &= \text{Tr} \left[(|x, b\rangle \langle x, b| \otimes |y, b\rangle \langle y, b|) \sigma_{AB}^{\otimes n} \right] \\ P_b^{(s)}(x, y) &= \text{Tr} \left[(|x, b\rangle \langle x, b| \otimes |y, b\rangle \langle y, b|) \sigma_{A^n B^n}^{(s)} \right] \\ P_b^{(s)}(x, y) &= \text{Tr} \left[(|x \oplus e_{(s,b)}, b\rangle \langle x \oplus e_{(s,b)}, b| \otimes |y \oplus e_{(s,b)}, b\rangle \langle y \oplus e_{(s,b)}, b|) \sigma_{AB}^{\otimes n} \right]\end{aligned}\tag{A.6}$$

where P_b is the probability distribution obtained when measurements are done on $\sigma_{AB}^{\otimes n}$, while $P_b^{(s)}$ is the distribution obtained when measured on $\sigma_{A^n B^n}^{(s)}$. The non-trivial equality in the last step follows from the fact that Pauli operators acting on any of our measurement POVMs either leave the POVM unchanged, or flip the outcome. These flips are included are $e_{(s,b)}$. Here, note the *same* $e_{(s,b)}$ appears in both Alice and Bob's outcomes, and therefore leaves $x \oplus y$ unchanged. Therefore, we have

$$P_b^{(s)}(x, y) = P_b(x \oplus e_{(s,b)}, y \oplus e_{s,b})\tag{A.7}$$

The above equation is valid for measurements on any number of systems. For symmetry in parameter estimation, we only need single-copy version of the above statement : $P_b^{(s)}(x \oplus y = 1) = P_b(x \oplus y = 1) = Q_b$. Thus σ_{AB} , $\sigma_{AB}^{(s)}$, and the twirled state $\hat{\sigma}_{AB}$ produce identical QBER's.

Symmetry in IR

We will show that the twirled state cannot have a lower error correction cost than the original state. Note that in the Recursive protocol (Sec. 4.1) and AD+CPA protocol (Sec. 4.2), the error correction cost is simply the binary entropy of the error rate of the bits to be corrected, which depends only on Q_Z . In these cases, the condition of equal error-correction cost is trivially satisfied, since twirling does not change the QBERs. In remaining protocols, our statistics satisfy

$$P^{(s)}(x, y) = P(x \oplus e_s, y \oplus e_s) \quad (\text{A.8})$$

We will drop the b subscript here, and assume it is Z , since all the key generation rounds are measured in the Z basis. Let F be a register whose value is some deterministic function of $x \oplus y$. F will represent feedback from Bob to Alice in our protocols. Let $\alpha(x \oplus y)$ be function that computes this value. Then obviously we have,

$$\begin{aligned} P^{(s)}(X = x, Y = y, F = \alpha(x \oplus y)) &= P(X = x \oplus e_{s,b}, Y = y \oplus e_{s,b}, F = \alpha(x \oplus y)) \\ P^{(s)}(X = x, Y = y, F = f) &= P(X = x \oplus e_{s,b}, Y = y \oplus e_{s,b}, F = f) \end{aligned} \quad (\text{A.9})$$

Let $L_1(X)$ and $L_2(X)$ be any linear functions of X . Notice that our the expressions that appear in δ_{leak} are always of the form $H(L_1(X)|YL_2(X), F \in \Omega)_P$. This can be interpreted as follows : assuming that Bob knows $L_2(X)$ and Y and used it to compute F , and that F is in some specified set, send error-correction information to correct $L_1(X)$. The expression is evaluated on $P(X, Y)$. Since measuring x, y on $\sigma_{AB}^{(s)}$ is equivalent to measuring $x \oplus e_s, y \oplus e_s$ on $\sigma_{AB}^{\otimes n}$, we have

$$\begin{aligned} P^{(s)}(L_1(X) = L_1(x), L_2(X) = L_2(x), Y = y, F = f) &= \\ P(L_1(X) = L_1(x) \oplus e_s^{(1)}, L_2(X) = L_2(x) \oplus e_s^{(2)}, Y = y \oplus e_s, F = f) \end{aligned} \quad (\text{A.10})$$

where $e_s^{(1)} = L_1(e_s)$ and $e_s^{(2)} = L_2(e_s)$. Using this, we can write

$$\begin{aligned} H(L_1(X)|YL_2(X), F \in \Omega)_{P^{(s)}} &= \sum_{f \in \Omega} p(F = f) H(L_1(X)|YL_2(X), F = f)_{P^{(s)}} \\ &= \sum_{f \in \Omega} p(F = f) H(L_1(X)|YL_2(X), F = f)_P \\ &= H(L_1(X)|YL_2(X), F \in \Omega)_P \end{aligned} \quad (\text{A.11})$$

Now, consider the statistics $\hat{P}$ obtained from the twirled state.

$$\hat{P}(X, Y) = \frac{1}{4^n} \sum_s P^{(s)}(X, Y) \quad (\text{A.12})$$

and

$$\hat{P}(L_1(X), L_2(X), Y, F) = \frac{1}{4^n} \sum_s P^{(s)}(L_1(X), L_2(X), Y, F) \quad (\text{A.13})$$

By concavity of conditional entropy [26],

$$\begin{aligned} H(L_1(X)|Y L_2(X), F \in \Omega)_{\hat{P}} &\geq \frac{1}{4^n} \sum_s H(L_1(X)|Y L_2(X), F \in \Omega)_{P^{(s)}} \\ &= \frac{1}{4^n} \sum_s H(L_1(X)|Y L_2(X), F \in \Omega)_P \\ &= H(L_1(X)|Y L_2(X), F \in \Omega)_P \end{aligned} \quad (\text{A.14})$$

i.e, the twirled state has a higher EC cost.

Symmetry in Eve's information on the Key

We now consider the objective function that determines Eve's information on the key. We will show that the twirled state leads to Eve having more knowledge of the key than the original state. Again, we drop the b and assume that for the key generation rounds, Alice and Bob measure in Z basis and share the state

$$\rho_{XYE} = \sum_{x,y \in F_2^n} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \text{Tr} \left[(|x\rangle \langle x| \otimes |y\rangle \langle y| \otimes I_E) |\psi\rangle \langle \psi| \right] \quad (\text{A.15})$$

where $|\psi\rangle$ is a purification of $\sigma_{AB}^{\otimes n}$. After measurements, F (or W) is some announcement completely determined by location of errors, given by $\alpha(x \oplus y)$. Let $\tilde{A}$ be an announcement that is linear in X , given by $L_3(X)$. If there is no such announcement then L_3 can be identically $\perp$. Note that in case of the Recursive protocol (see Sec 4.1, Eq. 4.36), for round i , $\hat{Z}_{j < i}$ can be treated as an announcement and absorbed into $\tilde{A}$. Note that for Eq. 4.35 and Eq. 4.46, F here should be interpreted as W (both are linear functions of location of errors). Thus the above formulation applies to all terms with Eve in our keyrate expressions. The state then looks like

$$\rho_{XY\tilde{A}FE} = \sum_{x,y \in F_2^n} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes |L_3(x)\rangle \langle L_3(x)| \otimes |\alpha(x \oplus y)\rangle \langle \alpha(x \oplus y)| \otimes \rho_E^{x,y} \quad (\text{A.16})$$

where $\rho_E^{x,y} = \text{Tr} \left[(|x\rangle \langle x| \otimes |y\rangle \langle y| \otimes I_E) |\psi\rangle \langle \psi| \right]$. We can rewrite this as

$$\rho_{XY\tilde{A}FE} = \sum_f \sum_{\substack{x,y \in F_2^n \\ \alpha(x \oplus y) = f}} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes |L_3(x)\rangle \langle L_3(x)| \otimes |f\rangle \langle f| \otimes \rho_E^{x,y} \quad (\text{A.17})$$

Alice implements a *linear* keymap depending on the value of F . Let us call the keymap function K_f conditioned on $F = f$. Therefore, the $Z\tilde{A}FE$ state is then given by

$$\begin{aligned}\rho_{ZY\tilde{A}FE} &= \sum_f \sum_{\substack{x,y \in F_2^n \\ \alpha(x \oplus y) = f}} |K_f(x)\rangle \langle K_f(x)| \otimes |y\rangle \langle y| \otimes |L_3(x)\rangle \langle L_3(x)| \otimes |f\rangle \langle f| \otimes \rho_E^{x,y} \\ \rho_{Z\tilde{A}FE} &= \sum_f \sum_{x \in F_2^n} |K_f(x)\rangle \langle K_f(x)| \otimes |L_3(x)\rangle \langle L_3(x)| \otimes |f\rangle \langle f| \otimes \left(\sum_{\substack{y \in F_2^n \\ \alpha(x \oplus y) = f}} \rho_E^{x,y} \right) \quad (\text{A.18})\end{aligned}$$

The $S(Z|\tilde{A}FE)$ expression is evaluated on this state. Next, let us consider what happens when we move to the twirled state. Let $|\psi^{(s)}\rangle = \bigotimes_{i=1}^n (\sigma_{s_i} \otimes \sigma_{s_i} \otimes I_{E_1}) |\psi\rangle$ be a purification of $\rho_{AB}^{(s)}$. For this state running through the above protocol, we can compute the state of Alice-Bob-Eve at each step as follows :

$$\begin{aligned}\rho_{XY\tilde{A}FE}^{(s)} &= \sum_{x,y \in F_2^n} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes |L_3(x)\rangle \langle L_3(x)| \otimes |\alpha(x \oplus y)\rangle \langle \alpha(x \oplus y)| \otimes \rho_E^{x \oplus e_s, y \oplus e_s} \\ \rho_{XY\tilde{A}FE}^{(s)} &= \sum_{x,y \in F_2^n} |x \oplus e_s\rangle \langle x \oplus e_s| \otimes |y \oplus e_s\rangle \langle y \oplus e_s| \otimes |L_3(x \oplus e_s)\rangle \langle L_3(x \oplus e_s)| \\ &\quad \otimes |\alpha(x \oplus y)\rangle \langle \alpha(x \oplus y)| \otimes \rho_E^{x,y} \\ \rho_{X\tilde{A}FE}^{(s)} &= \sum_f \sum_{x \in F_2^n} |x \oplus e_s\rangle \langle x \oplus e_s| \otimes |L_3(x) \oplus L_3(e_s)\rangle \langle L_3(x) \oplus L_3(e_s)| \\ &\quad \otimes |f\rangle \langle f| \otimes \left(\sum_{\substack{y \in F_2^n \\ \alpha(x \oplus y) = f}} \rho_E^{x,y} \right) \\ \rho_{Z\tilde{A}FE}^{(s)} &= \sum_f \sum_{x \in F_2^n} |K_f(x) \oplus K_f(e_s)\rangle \langle K_f(x) \oplus K_f(e_s)| \otimes |L_3(x) \oplus L_3(e_s)\rangle \langle L_3(x) \oplus L_3(e_s)| \\ &\quad \otimes |f\rangle \langle f| \otimes \left(\sum_{\substack{y \in F_2^n \\ \alpha(x \oplus y) = f}} \rho_E^{x,y} \right) \quad (\text{A.19})\end{aligned}$$

Therefore $S(Z|E\tilde{A}F)_\rho = S(Z|E\tilde{A}F)_{\rho^{(s)}}$, since the states are the same upto a permutation on the $Z, \tilde{A}$ registers. Now, consider

$$|\hat{\psi}\rangle_{ABEE'E''} = \sum_s \frac{1}{\sqrt{4^n}} |\psi^{(s)}\rangle_{ABE} |s\rangle_{E'} |s\rangle_{E''} \quad (\text{A.20})$$

This is a purification of the twirled state, and it can be easily verified that $\text{Tr}_{EE'E''} [|\hat{\psi}\rangle \langle \hat{\psi}|] = \hat{\sigma}_{AB}^{\otimes n}$. Let $\mathcal{E}$ be the operation that maps the $A^{\otimes n} B^{\otimes n}$ register to $Z\tilde{A}F$ register. Therefore, applying this on the twirled state, we find that the state is given by

$$\hat{\rho}_{ZFE E'E''} = (\mathcal{E}_{A^{\otimes n} B^{\otimes n} \rightarrow Z\tilde{A}F} \otimes I_{EE'E''}) (|\hat{\psi}\rangle \langle \hat{\psi}|) \quad (\text{A.21})$$

By strong subadditivity of quantum entropy, we have $S(Z|\tilde{A}FEE'E'')_{\hat{\rho}} \leq S(Z|\tilde{A}FEE')_{\hat{\rho}}$. And we have

$$\begin{aligned} \rho_{Z\tilde{A}FEE'} &= \frac{1}{4^n} \sum_s (\mathcal{E}_{A^{\otimes n} B^{\otimes n} \rightarrow Z\tilde{A}F} \otimes I_{EE'} (|\psi^{(s)}\rangle \langle \psi^{(s)}| \otimes |s\rangle \langle s|)) \\ &= \frac{1}{4^n} \sum_s \rho_{Z\tilde{A}FE}^{(s)} \otimes |s\rangle \langle s|_{E'} \\ S(Z|\tilde{A}FEE')_{\hat{\rho}} &= \sum_s \frac{1}{4^n} S(Z|\tilde{A}FE)_{\rho_{Z\tilde{A}FE}^{(s)}} \\ &= S(Z|\tilde{A}FE)_{\rho_{Z\tilde{A}FE}} \end{aligned} \quad (\text{A.22})$$

Thus, we have $S(Z|\tilde{A}FEE'E'')_{\hat{\rho}} \leq S(Z|\tilde{A}FE)_{\rho_{Z\tilde{A}FE}}$. Thus for any $\sigma_{AB}^{\otimes n}$, the twirled state $\hat{\sigma}_{AB}^{\otimes n}$ is a) Bell-diagonal, b) satisfies constraints, and c) yields a lower keyrate. Thus the optimal attack can be assumed to be Bell-diagonal.

Appendix B

Linear Algebra

The results of this section are helpful in the last step of computing keyrates in Eqs. [3.15](#), [3.17](#), [4.36](#), [4.47](#). They reduce the computational requirements.

Gram-Schmidt Technique

In order to actually compute the keyrate, we will need to compute the von Neumann entropy of states. Typically, we will find that we need to compute $S(M)$ during our keyrate calculations, and M is a sum of t rank-1 operators, $M = \sum_{i=1}^t |\alpha_i\rangle \langle \alpha_i|$. Without loss of generality we can assume that these rank-1 operators cannot be broken down into two different sets of rank-1 operators that are orthogonal to each other. If that were so, M would have been block diagonal and we would diagonalize those blocks separately.

In the situations we encounter, the number of rank-1 projectors t is much smaller than the dimension of $|\alpha_i\rangle$. In this case, it is computationally easier to first express M on the t dimensional subspace on which it is non-zero, and then compute the entropy on this smaller $t \times t$ matrix. We will find that we actually only need to know the value of the Gram Matrix of $|\alpha_i\rangle$ s to diagonalize M .

To express M on the t dimensional subspace on which it is non-zero, we construct a new orthogonal basis $|\beta_i\rangle$ from $|\alpha_i\rangle$ via the Gram-Schmidt Process [\[26\]](#). This can be done as follows :

$$|\beta_i\rangle = |\alpha_i\rangle - \sum_{j=1}^{i-1} \frac{\langle \beta_j, \alpha_i \rangle}{\langle \beta_j, \beta_j \rangle} |\beta_j\rangle \quad (\text{B.1})$$

The above equations directly lead to

$$\begin{aligned}\langle \alpha_i, \beta_j \rangle &= \langle \alpha_i, \alpha_j \rangle - \sum_{k=1}^{j-1} \frac{\langle \alpha_i, \beta_k \rangle \langle \beta_k, \alpha_j \rangle}{\langle \beta_k, \beta_k \rangle} \\ \langle \beta_j, \beta_j \rangle &= \langle \alpha_j, \alpha_j \rangle - \sum_{k=1}^{j-1} \frac{\langle \alpha_j, \beta_k \rangle \langle \beta_k, \alpha_j \rangle}{\langle \beta_k, \beta_k \rangle}\end{aligned}\tag{B.2}$$

Since M is non-zero only on the space spanned by the $|\beta_i\rangle$ s, we can write

$$\begin{aligned}M &= \sum_{i,j=1}^t \frac{|\beta_i\rangle \langle \beta_i|}{\langle \beta_i, \beta_i \rangle} M \frac{|\beta_j\rangle \langle \beta_j|}{\langle \beta_j, \beta_j \rangle} \\ M &= \sum_{i,j=1}^t \frac{|\beta_i\rangle}{\sqrt{\langle \beta_i, \beta_i \rangle}} \left[\sum_{k=1}^t \frac{\langle \beta_i, \alpha_k \rangle \langle \alpha_k, \beta_j \rangle}{\sqrt{\langle \beta_i, \beta_i \rangle} \langle \beta_j, \beta_j \rangle} \right] \frac{\langle \beta_j|}{\sqrt{\langle \beta_j, \beta_j \rangle}} \\ M'_{ij} &= \sum_{k=1}^t \frac{\langle \beta_i, \alpha_k \rangle \langle \alpha_k, \beta_j \rangle}{\sqrt{\langle \beta_i, \beta_i \rangle} \langle \beta_j, \beta_j \rangle}\end{aligned}\tag{B.3}$$

Diagonalizing M is the same as diagonalizing the matrix M'_{ij} . In fact, we do not even need to write down M to compute M'_{ij} . Just the Gram Matrix $\langle \alpha_i, \alpha_j \rangle$ is enough. We do not need to store large dimensional vectors $|\alpha_i\rangle$ and $|\beta_i\rangle$. This can be seen easily from the following algorithm

Computing M'_{ij}

- 1 : compute $\langle \alpha_i, \alpha_j \rangle$
- 2 : set $\langle \alpha_i, \beta_1 \rangle = \langle \alpha_i, \alpha_1 \rangle$ and $\langle \beta_1, \beta_1 \rangle = \langle \alpha_1, \alpha_1 \rangle$
- 3 : for $j = 2$ to t
- 4 : compute $\langle \alpha_i, \beta_j \rangle$ and $\langle \beta_j, \beta_j \rangle$ from Eq. B.2, $\forall i$
- 5 : compute M'_{ij} from Eq. B.3 using $\langle \alpha_i, \beta_j \rangle$ and $\langle \beta_j, \beta_j \rangle$

M'_{ij} is a $t \times t$ matrix, regardless of the dimension of $|\alpha_i\rangle$. If $|\alpha_i\rangle$ is a d dimensional vector, then direct diagonalization of M takes about $O(d^3)$ time. In contrast, the above method requires atmost $O(t^3d)$ to compute all the inner-product tables, and then atmost $O(t^3)$ to diagonalize M_{ij} for a total time cost of $O(t^6d)$. We will repeatedly be invoking the above trick to complete the final step of our keyrate calculations. Therefore we summarize the results of this section along with a few other tools in the following lemmas. It might be useful to revise our notation from Sec 3.1 to have the right context for these statements.

Lemma B.1. Let M be a sum of t rank-1 operators, $M = \sum_{i=1}^t |\alpha_i\rangle \langle \alpha_i|$, such that $|\alpha_i\rangle$ cannot be separated into two sets of vectors orthogonal to the other set. Let G be the Gram Matrix of $|\alpha_i\rangle$ denoted by $G_{ij} = \langle \alpha_i, \alpha_j \rangle$. Let M' denote the $t \times t$ matrix whose i, j entries are given by Eqs. B.2, B.3. Then $S(M) = S(M')$, and $S(M)$ is a deterministic function of G_{ij} .

Proof : From the above section.

Lemma B.2. Two operators $M = \sum_{i=1}^t |\alpha_i\rangle \langle \alpha_i|$, and $\tilde{M} = \sum_{i=1}^t |\gamma_i\rangle \langle \gamma_i|$ will have the same entropy if their Gram matrix is identical. I.e, $\langle \alpha_i, \alpha_j \rangle = \langle \gamma_i, \gamma_j \rangle \implies S(M) = S(\tilde{M})$.

Proof : From the above section.

Lemma B.3. Let $S_x, S_y \subset F_2^n$, and let $\rho = \sum_{x \in S_x, y \in S_y} \rho_E^{x,y}$, where $\rho_E^{x,y}$ as defined in Eq. 3.6, then

$$S(\rho) = \sum_{e \in F_2^n} S\left(\sum_{\substack{x \in S_x, y \in S_y \\ x \oplus y = e}} \rho_E^{x,y}\right) \quad (\text{B.4})$$

Proof : Follows from the fact that the support of $\rho_E^{x,y}$ and $\rho_E^{x',y'}$ is orthogonal if $x \oplus y \neq x' \oplus y'$.

Lemma B.4. Let $S_x \subset F_2^n$, and let $\rho^\alpha = \sum_{x \in S_x} \rho_E^{x \oplus \alpha, x \oplus \alpha \oplus e}$ for any $\alpha, e \in F_2^n$, where $\rho_E^{x,y}$ as defined in Eq. 3.6, then $S(\rho^\alpha) = S(\rho^{\alpha'}) = S(\rho^{\alpha=0})$.

Proof : Since $\rho_E^{x,y} = |e_{x,y}\rangle \langle e_{x,y}|$, the Gram Matrix corresponding to ρ^α can be written as $G_{x,x'}^\alpha = \langle e_{(x \oplus \alpha, x \oplus \alpha \oplus e)}, e_{(x' \oplus \alpha, x' \oplus \alpha \oplus e)} \rangle$. However, since $\langle e_{x,y}, e_{x',y'} \rangle = \langle e_{x \oplus \alpha, y \oplus \alpha}, e_{x' \oplus \alpha, y' \oplus \alpha} \rangle$ (from Eq. 3.6), $G_{x,x'}^\alpha$ is independent of α . Since $S(\rho^\alpha)$ is a function of its Gram matrix G^α , and G^α is independent of α , so is $S(\rho^\alpha)$ (from Lemma B.1).

Appendix C

Numerical details for CPA keyrate calculations

In this section we explain how we compute the keyrate expression [3.12](#).

$$K_{\text{CPA}} = \min_{\rho \in \Gamma_{\text{Bell}}} \frac{1}{n} \left(p_{\text{match}} (S(X|E, F=0) - S(X|Y P_X, F=0)) \right. \\ \left. + (1 - p_{\text{match}}) S(P_X|E, F=1) - S(P_X|Y) \right) \quad (\text{C.1})$$

where p_{match} is the probability of parity string matching. The computation is extremely simple, albeit a bit tedious and cumbersome. The most important aspect of the computation is choosing the right linear basis to express Alice and Bob's data in. We compute the expression inside the minimization for an arbitrary Bell-diagonal state. For six-state, the QBER's and normalization fix the state entirely. For BB84, the keyrate calculation involves a simple minimization over one parameter.

C.1 Choosing a linear basis

It is crucial to choose the correct basis to express our data, which belongs to the set of n bit strings $F_2^n = \{0, 1\}^n$ ¹. We would like a basis that makes the action of the partitioning clear. To do so we turn to linear codes. Our partition matrix H can be thought of as

¹This is a basis for classical bit strings, not a Hilbert space

a parity check matrix for a k dimensional linear code C . Let $C = \text{Span}\{c_1, c_2, \dots, c_k\}$. We complete this basis by adding $n - k$ basis elements, via $D = \text{Span}\{d_1, d_2, \dots, d_{n-k}\}$. Then $C \oplus D = \{0, 1\}^n$, since it is obtained by spanning the n linearly independent basis elements. Now, any $x \in F_2^n$ can be written as $x = d_x \oplus c_x$ for $d_x \in D, c_x \in C$ and vice-versa. Here, the d_x part determines the parity string (i.e the subset of the partition), while the c_x part determines the “index within the subset”. This follows from the fact that $H(c_x \oplus d_x) = Hd_x$. Parity string match is equivalent to $Hx = Hy \iff d_x = d_y$. Therefore the CCQ state at the start of the classical stage of the QKD protocol is

$$\begin{aligned} \rho_{XYE} &= \sum_{x,y \in \{0,1\}^n} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \rho_E^{x,y} \\ &= \sum_{\substack{c_x, c_y \in C \\ d_x, d_y \in D}} |c_x \oplus d_x\rangle \langle c_x \oplus d_x| \otimes |c_y \oplus d_y\rangle \langle c_y \oplus d_y| \otimes \rho_E^{c_x \oplus d_x, c_y \oplus d_y} \end{aligned} \quad (\text{C.2})$$

Note that x is equivalent to the combination $c_x d_x$, since there is a one-to-one correspondence between the two. We will thus use both these terms interchangeably.

C.2 Computing Cost of Error Correction

We first compute the probability of obtaining matching partitions. One can see that this happens when $Hx = Hy \iff d_x = d_y$. Thus,

$$\begin{aligned} p_{\text{match}} &= \sum_{\substack{c_x, c_y \in C \\ d_x, d_y \in D}} \delta_{d_x d_y} \text{Tr}[\rho_E^{c_x \oplus d_x, c_y \oplus d_y}] \\ &= \sum_{\substack{c_x, c_y \in C \\ d_x \in D}} \left(\frac{\lambda_0 + \lambda_1}{2} \right)^{n-w(c_x \oplus c_y)} \left(\frac{\lambda_2 + \lambda_3}{2} \right)^{w(c_x \oplus c_y)} \\ &= 2^{n-k} \sum_{c_x, c_y \in C} \left(\frac{\lambda_0 + \lambda_1}{2} \right)^{n-w(c_x \oplus c_y)} \left(\frac{\lambda_2 + \lambda_3}{2} \right)^{w(c_x \oplus c_y)} \\ &= \sum_{c \in C} (\lambda_0 + \lambda_1)^{n-w(c)} (\lambda_2 + \lambda_3)^{w(c)} \end{aligned} \quad (\text{C.3})$$

where $w(x)$ is the hamming weight of x .

C.2.1 Cost of correcting Partitions

With $x = c_x \oplus d_x$ and $y = c_y \oplus d_y$, we have

$$P(c_x d_x, c_y d_y) = \left(\frac{\lambda_0 + \lambda_1}{2} \right)^{n-w(c_x \oplus c_y \oplus d_x \oplus d_y)} \left(\frac{\lambda_2 + \lambda_3}{2} \right)^{w(c_x \oplus c_y \oplus d_x \oplus d_y)} \quad (\text{C.4})$$

where $P(x, y) = P(c_x d_x, c_y d_y)$ is the probability table for Alice and Bob's outcomes. Since $P_X = H(c_x \oplus d_x) = H d_x$, and permutations don't change the entropies, we have

$$P(H d_x, c_y d_y) = \sum_{c_x \in C} \left(\frac{\lambda_0 + \lambda_1}{2} \right)^{n-w(c_x \oplus c_y \oplus d_x \oplus d_y)} \left(\frac{\lambda_2 + \lambda_3}{2} \right)^{w(c_x \oplus c_y \oplus d_x \oplus d_y)} \quad (\text{C.5})$$

Then, $S(P_X|Y) = S(P_X Y) - S(Y) = S(P(d_x, c_y d_y)) - n$.

C.2.2 Cost of correcting key

We compute $S(X|Y P_X, F = 0)$. Let $P(x, y|F = 0)$ be the normalized probability distribution of outcomes between Alice and Bob conditioned on partitions match.

$$\begin{aligned} P(c_x d_x, c_y d_y|F = 0) &= \delta_{d_x d_y} \frac{\text{tr}[\rho_E^{c_x \oplus d_x, c_y \oplus d_y}]}{p_{\text{match}}} \\ &= \frac{\delta_{d_x d_y} \left(\frac{\lambda_0 + \lambda_1}{2} \right)^{n-w(c_x \oplus c_y)} \left(\frac{\lambda_2 + \lambda_3}{2} \right)^{w(c_x \oplus c_y)}}{p_{\text{match}}} \end{aligned} \quad (\text{C.6})$$

$S(X|Y P_X, F = 0)$ can be computed from the probability distribution.

C.3 Computing Eve's Information on the Key

We will use the fact that $S(Z|E)$ evaluated on $\rho_{ZE} = \sum_z p(z) \rho_E^z$ is given by

$$S(Z|E)_{\rho_{ZE}} = S(Z) + \sum_z p(z) S(\rho_E^z) - S(\rho_E) \quad (\text{C.7})$$

Thus, computing the above expression is essentially an exercise in computing Eve's conditional states depending on the key register, i.e, ρ_E^z .

C.3.1 When partitions match

We compute $S(X|E, F=0)$. This happens when $Hx = Hy \iff d_x = d_y$ and . Therefore

$$\rho_{XYE|F=0} = \sum_{\substack{c_x, c_y \in C \\ d_x \in D}} |c_x \oplus d_x\rangle \langle c_x \oplus d_x| \otimes |c_y \oplus d_x\rangle \langle c_y \oplus d_x| \otimes \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_x}}{p_{\text{match}}} \quad (\text{C.8})$$

where the conditional states $\rho_E^{x,y}$ are determined by Eq. 3.6. After the keymap, we have

$$\begin{aligned} \rho_{ZE|F=0} &= \sum_{\substack{c_x, c_y \in C \\ d_x \in D}} |c_x \oplus d_x\rangle \langle c_x \oplus d_x| \otimes \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_x}}{p_{\text{match}}} \\ &= \sum_{\substack{c_x \in C \\ d_x \in D}} \frac{1}{2^n} |c_x \oplus d_x\rangle \langle c_x \oplus d_x| \otimes \sum_{c_y \in C} \frac{2^n \rho_E^{c_x \oplus d_x, c_y \oplus d_x}}{p_{\text{match}}} \\ \rho_{E|F=0}^{c_x \oplus d_x} &= \sum_{c_y \in C} \frac{2^n \rho_E^{c_x \oplus d_x, c_y \oplus d_x}}{p_{\text{match}}} \\ \rho_{E|F=0} &= \sum_{\substack{c_x, c_y \in C \\ d_x \in D}} \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_x}}{p_{\text{match}}} \end{aligned} \quad (\text{C.9})$$

Note that $\rho_{E|F=0}^{c_x \oplus d_x}$ above is a sum of 2^k orthogonal projectors, since $c_x \oplus d_x \oplus c_y \oplus d_y = c_x \oplus c_y$ is different for each of the projectors in the sum. Therefore, the non-zero eigenvalues of $\rho_{E|F=0}^{c_x \oplus d_x}$ are $\{(\lambda_0 + \lambda_1)^{n-w(c)}(\lambda_2 + \lambda_3)^{w(c)} / p_{\text{match}} \forall c \in C\}$. Since this is independent of the value of d_x or c_x , we have ²

$$\sum_{c_x d_x} \frac{1}{2^n} S(\rho_{E|F=0}^{c_x \oplus d_x}) = S\left(\left\{\frac{(\lambda_0 + \lambda_1)^{n-w(c)}(\lambda_2 + \lambda_3)^{w(c)}}{p_{\text{match}}} \forall c \in C\right\}\right) \quad (\text{C.10})$$

²This is simply a special case of the use of our Lemmas B.1, B.3, B.4.

To compute $S(\rho_{E|F=0})$, we write

$$\begin{aligned}
\rho_{E|F=0} &= \sum_{\substack{c_x, c_y \in C \\ d_x \in D}} \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_x}}{p_{\text{match}}} \\
&= \sum_{c \in C} \sum_{x \in F_2^n} \frac{\rho_E^{x, x \oplus c}}{p_{\text{match}}} \\
&= \sum_{c \in C} \bigotimes_{i=1}^n \frac{1}{p_{\text{match}}} \left(\sum_{x_i=0,1} \rho_E^{x_i, x_i \oplus c_i} \right)
\end{aligned} \tag{C.11}$$

Note that each value of c in the above expression determines a block, and therefore each of the blocks can be diagonalized separately as

$$\begin{aligned}
S\left(\frac{1}{p_{\text{match}}} \sum_{x_i=0,1} \rho_E^{x_i, x_i \oplus c_i}\right) &= S\left(\frac{1}{p_{\text{match}}} \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix}^{\otimes(n-w(c))} \otimes \begin{pmatrix} \lambda_2 & 0 \\ 0 & \lambda_3 \end{pmatrix}^{\otimes w(c)}\right) \\
&= \frac{1}{p_{\text{match}}} \left\{ (\lambda_0 + \lambda_1)^{n-w(c)-1} (\lambda_2 + \lambda_3)^{w(c)} S(\{\lambda_0, \lambda_1\}) (n - w(c)) + \right. \\
&\quad \left. (\lambda_0 + \lambda_1)^{n-w(c)} (\lambda_2 + \lambda_3)^{w(c)-1} S(\{\lambda_2, \lambda_3\}) w(c) \right\} + (\lambda_0 + \lambda_1)^{n-w(c)} (\lambda_2 + \lambda_3)^{w(c)} \frac{\log(p_{\text{match}})}{p_{\text{match}}} \\
S(\rho_{E|F=0}) &= \sum_{c \in C} S\left(\frac{1}{p_{\text{match}}} \sum_{x_i=0,1} \rho_E^{x_i, x_i \oplus c_i}\right)
\end{aligned} \tag{C.12}$$

where we have used the fact that $S(\rho_1 \otimes \rho_2) = \text{Tr}(\rho_1)S(\rho_2) + \text{Tr}(\rho_2)S(\rho_1)$. Thus

$$S(X|E, F=0) = n + \sum_{c_x d_x} \frac{1}{2^n} S(\rho_{E|F=0}^{c_x \oplus d_x}) - S(\rho_{E|F=0}) \tag{C.13}$$

can be computed using Eq. C.10 and Eq. C.12.

C.3.2 When partitions don't match

We compute $S(P_X|E, F=1)$. This happens when $Hx \neq Hy \iff d_x \neq d_y$. Therefore

$$\rho_{XYE|F=1} = \sum_{\substack{c_x, c_y \in C \\ d_x, d_y \in D \\ d_x \neq d_y}} |c_x \oplus d_x\rangle \langle c_x \oplus d_x| \otimes |c_y \oplus d_y\rangle \langle c_y \oplus d_y| \otimes \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}} \tag{C.14}$$

After the keymap, we have

$$\begin{aligned}
\rho_{ZE|F=1} &= \sum_{\substack{c_x, c_y \in C \\ d_x, d_y \in D \\ d_x \neq d_y}} |H(c_x \oplus d_x)\rangle \langle H(c_x \oplus d_x)| \otimes \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}} \\
&= \sum_{d_x \in D} |Hd_x\rangle \langle Hd_x| \otimes \sum_{\substack{c_x, c_y \in C \\ d_y \neq d_x \in D}} \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}}
\end{aligned} \tag{C.15}$$

For the purposes of computing the entropies required for the keyrate calculation, we can replace Hd_x with d_x , since this is equivalent to a one-to-one relabelling of the key. Therefore, we have

$$\begin{aligned}
\rho_{ZE|F=1} &= \sum_{d_x \in D} \frac{1}{2^{n-k}} |d_x\rangle \langle d_x| \otimes \sum_{\substack{c_x, c_y \in C \\ d_y \neq d_x \in D}} \frac{2^{n-k} \rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}} \\
\rho_{E|F=1}^{d_x} &= \sum_{\substack{c_x, c_y \in C \\ d_y \neq d_x \in D}} \frac{2^{n-k} \rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}} \\
\rho_{E|F=1} &= \sum_{\substack{c_x, c_y \in C \\ d_x, d_y \in D \\ d_x \neq d_y}} \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}}
\end{aligned} \tag{C.16}$$

One can rewrite these expressions as

$$\begin{aligned}
\rho_{E|F=1}^{d_x} &= \sum_{\substack{c_x, c_y \in C \\ d_y \neq d_x \in D}} \frac{2^{n-k} \rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}} \\
&= \sum_{\substack{c \in C, d \in D \\ d \neq 0}} \sum_{c_x \in C} \frac{2^{n-k} \rho_E^{d_x \oplus c_x, d_x \oplus c_x \oplus d \oplus c}}{1 - p_{\text{match}}}
\end{aligned} \tag{C.17}$$

First, let us consider $\sum_{d_x \in D} \frac{1}{2^{n-k}} S(\rho_{E|F=1}^{d_x})$. This entropy can be simplified and computed via Lemmas B.3, B.4 to obtain

$$\sum_{d_x \in D} \frac{1}{2^{n-k}} S(\rho_{E|F=1}^{d_x}) = S(\rho_{E|F=1}^{d_x=0}) \tag{C.18}$$

Similary, $\rho_{E|F=1}$ can be written as

$$\begin{aligned}\rho_{E|F=1} &= \sum_{\substack{c_x, c_y \in C \\ d_x, d_y \in D \\ d_x \neq d_y}} \frac{\rho_E^{c_x \oplus d_x, c_y \oplus d_y}}{1 - p_{\text{match}}} \\ &= \sum_{\substack{c \in C, d \in D \\ d \neq 0}} \sum_{x \in \{0,1\}^n} \frac{\rho^{x, x \oplus c \oplus d}}{1 - p_{\text{match}}}\end{aligned}\tag{C.19}$$

and note that each c and d determines a new block. Now, we can repeat the procedure explained earlier,

$$\begin{aligned}S\left(\sum_{x \in \{0,1\}^n} \frac{\rho^{x, x \oplus c \oplus d}}{1 - p_{\text{match}}}\right) &= S\left(\frac{1}{p_{\text{match}}} \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix}^{\otimes (n-w(c \oplus d))} \otimes \begin{pmatrix} \lambda_2 & 0 \\ 0 & \lambda_3 \end{pmatrix}^{\otimes w(c \oplus d)}\right) \\ &= \frac{1}{1 - p_{\text{match}}} \left\{ (\lambda_0 + \lambda_1)^{n-w(c \oplus d)-1} (\lambda_2 + \lambda_3)^{w(c \oplus d)} S(\{\lambda_0, \lambda_1\}) (n - w(c \oplus d)) + \right. \\ &\quad \left. (\lambda_0 + \lambda_1)^{n-w(c \oplus d)} (\lambda_2 + \lambda_3)^{w(c \oplus d)-1} S(\{\lambda_2, \lambda_3\}) w(c \oplus d) \right\} \\ &\quad + (\lambda_0 + \lambda_1)^{n-w(c \oplus d)} (\lambda_2 + \lambda_3)^{w(c \oplus d)} \frac{\log(1 - p_{\text{match}})}{1 - p_{\text{match}}} \\ S(\rho_{E|F=1}) &= \sum_{\substack{c \in C, d \in D \\ d \neq 0}} S\left(\sum_{x \in \{0,1\}^n} \frac{\rho^{x, x \oplus c \oplus d}}{1 - p_{\text{match}}}\right)\end{aligned}\tag{C.20}$$

Finally, from Eq. C.18 and Eq. C.20

$$S(Z|E, F=1) = n - k + \sum_{d_x \in D} \frac{1}{2^{n-k}} S(\rho_{E|F=1}^{d_x}) - S(\rho_{E|F=1})\tag{C.21}$$

We have now computed all the expressions needed to compute Eq. 3.15.

C.4 Calculations for AD partitions

In this section we look at the results for AD partitions, where $k = 1$ and c_1 is the vector of all 1s. This corresponds to the situation where x and $\bar{x}$ ($\bar{x}$ is the string obtained by

flipping all bits of x) are assigned to one partition. H is a $n - 1 \times n$ matrix. Analytic results are possible since the largest block we have to diagonalize is 2×2 , where the Gram Schmidt procedure can be done by hand. Using the results of the previous section, it is straightforward to compute the following quantities in this case.

$$\begin{aligned}
p_{\text{match}} &= (\lambda_0 + \lambda_1)^n + (\lambda_2 + \lambda_3)^n \\
\tilde{e} &= \frac{(\lambda_2 + \lambda_3)^n}{(\lambda_0 + \lambda_1)^n + (\lambda_2 + \lambda_3)^n} \quad (\text{sifted error rate of the one bit}) \\
\sum_z p(z) S(\rho_{E|F=0}^z) &= h(\tilde{e}) \quad (\text{when partitions match}) \\
S(\rho_{E|F=0}) &= \frac{n(\lambda_0 + \lambda_1)^{n-1}}{p_{\text{match}}} S(\{\lambda_0, \lambda_1\}) + \frac{n(\lambda_2 + \lambda_3)^{n-1}}{p_{\text{match}}} S(\{\lambda_2, \lambda_3\}) + \log(p_{\text{match}}) \\
S(\rho_{E|F=1}) &= \frac{n(1 - (\lambda_0 + \lambda_1)^{n-1})}{1 - p_{\text{match}}} S(\{\lambda_0, \lambda_1\}) + \frac{n(1 - (\lambda_2 + \lambda_3)^{n-1})}{1 - p_{\text{match}}} S(\{\lambda_2, \lambda_3\}) \\
&\quad + \log(1 - p_{\text{match}})
\end{aligned} \tag{C.22}$$

The two remaining terms are somewhat more complicated to write down, and require the following expressions to be written concisely

$$\begin{aligned}
f_k &= (\lambda_0 + \lambda_1)^{n-k} (\lambda_2 + \lambda_3)^k + (\lambda_0 + \lambda_1)^k (\lambda_2 + \lambda_3)^{n-k} \\
g_k^+ &= \frac{(\lambda_0 + \lambda_1)^{n-k} (\lambda_2 + \lambda_3)^k}{2(1 - p_{\text{match}})} \left(1 + \frac{(\lambda_2 - \lambda_3)^k (\lambda_0 - \lambda_1)^{n-k}}{(\lambda_2 + \lambda_3)^k (\lambda_0 + \lambda_1)^{n-k}} \right) \\
g_k^- &= \frac{(\lambda_0 + \lambda_1)^{n-k} (\lambda_2 + \lambda_3)^k}{2(1 - p_{\text{match}})} \left(1 - \frac{(\lambda_2 - \lambda_3)^k (\lambda_0 - \lambda_1)^{n-k}}{(\lambda_2 + \lambda_3)^k (\lambda_0 + \lambda_1)^{n-k}} \right)
\end{aligned} \tag{C.23}$$

which gives us

$$\begin{aligned}
S(P_X|Y) &= -\frac{1}{2} \sum_{k=0}^n \binom{n}{k} f_k \log(f_k) \\
\sum_z p(z) S(\rho_{E|F=1}^z) &= \sum_{k=1}^{n-1} \binom{n}{k} S(\{g_k^+, g_k^-\})
\end{aligned} \tag{C.24}$$

Appendix D

Numerical details for SCPA keyrate calculations

In this section we explain how we compute the keyrate expression [3.17](#),

$$K_{\text{SCPA}} = \min_{\rho \in \Gamma_{\text{bell}}} \frac{S(Z|EF) - \sum_{j=1}^{n-k} \delta_{\text{leak}}^{(j)} - \delta_{\text{leak}}^{(\perp)}}{n} \quad (\text{D.1})$$

which we have simplified to :

$$\begin{aligned} S(Z|EF) &= \sum_{l=1}^{n-k} p_l S(P_X^{(1)} \dots P_X^{(l)} | E, N_R = l) + p_{\perp} S(X|E, N_R = \perp) \\ \delta_{\text{leak}}^{(j)} &= (1 - p_1 - \dots p_{j-1}) S(P_X^{(j)} | Y P_X^{(1)} \dots P_X^{(j-1)}, F_1 = 0 \dots F_{j-1} = 0) \\ \delta_{\text{leak}}^{(\perp)} &= p_{\perp} S(X | Y P_X, F_1 \dots F_{n-k} = 0) \end{aligned} \quad (\text{D.2})$$

We compute the above expressions for Bell-diagonal states shared between Alice and Bob. For six-state, the QBER's and normalization fix the state entirely. For BB84, the keyrate calculation involves a simple minimization over one parameter.

D.1 Choosing a linear basis

We need a convenient basis to write Alice and Bob's data in, one that makes the action of our classical processing clear. Let the parity check matrix given by $H = [h_1; h_2 \dots; h_{n-k}]^T$

where h_i is the i th row of H . Consider the case there the first error was found in round j , i.e when $N_R = j$. Therefore $h_i.x = h_i.y$ for $i < j$, and $h_j.x \neq h_j.y$. Let C_j be the codespace corresponding to the first j parities matching (i.e the codespace corresponding to the parity check matrix consisting of the first j rows of H). Let C_{j-1} be the codespace corresponding to the first $j - 1$ parities matching (i.e the codespace corresponding to the parity check matrix consisting of the first $j - 1$ rows of H). C_j and C_{j-1} are spanned $n - j$ and $n - j + 1$ basis elements respectively.

Now, any element of C_j is also an element of C_{j-1} . Therefore C_{j-1} can be constructed by adding exactly one basis element to C_j . Let $\tilde{D}_j$ be the space spanned by this basis element. Thus $C_j \oplus \tilde{D}_j = C_{j-1}$. And let D_j be the space spanned by basis elements that complete C_{j-1} . Therefore $D_j \oplus C_{j-1} = F_2^n$. Therefore we now have

- 1) C_j made of $n - j$ basis elements, which is the part of x and y strings that may or may not be different. This represents the fact that we only know information about the first j parities.
- 2) $\tilde{D}_j$ made of 1 basis element, which is the part of x and y that has to be different. This represents the fact that the j th parity is different.
- 3) D_j made of $j - 1$ basis elements, which is the part of x and y that has to be the same. This represents the fact that $j - 1$ parities are the same.

Since $\{0, 1\}^n = C_j \oplus \tilde{D}_j \oplus D_j$, for the equations below always assume that $x = c_x \oplus d_x \oplus \tilde{d}_x$, where $c_x \in C_j$, $d_x \in D_j$, $\tilde{d}_x \in \tilde{D}_j$.

To see the the spaces we have constructed have the above properties, note that the condition for the first $j - 1$ parities matching is given by $h_i.(x \oplus y) = 0$ for $i < j$, which implies $h_i.(d_x \oplus d_y) = 0$ which requires $d_x = d_y$. The condition that the j th parity has an error translates then translates to $h_j.(x \oplus y) = h_j.(\tilde{d}_x \oplus \tilde{d}_y) \neq 0$, or $\tilde{d}_x \neq \tilde{d}_y$.

Thus, $N_R = j \iff d_x = d_y, \tilde{d}_x \neq \tilde{d}_y$.

Our choice of basis is slightly modified in the extreme cases of error in 1st partition bit and no error. In the former case, we do not have D_1 , but only C_1 with $n - 1$ basis elements, and $\tilde{D}_1$ with one basis element. In the latter, we do not have $\tilde{D}_{n-k}$, but only C_{n-k} and D_{n-k} . In fact, the case where partition bits are all equal is exactly identical to the case in CPA protocol, and we do not have to redo those calculations.

D.2 Computing probability of rounds

After measurements, the state of the protocol is given by $\rho_{XYE} = \sum_{x,y \in \{0,1\}^n} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \rho_E^{x,y}$. We will first compute the probabilities p_j of $N_R = j$. If $j \in \{1 \dots n - k\}$, then

p_j is the probability of $j - 1$ parity bits matching and the j th parity bit having an error. This can be easily computed via

$$\begin{aligned}
p_j &= \sum_{\substack{c_x, c_y \in C_j \\ d_x = d_y \in D_j \\ \tilde{d}_x \neq \tilde{d}_y \in \tilde{D}_j}} \text{tr}[\rho_E^{c_x \oplus d_x \oplus \tilde{d}_x, c_y \oplus d_y \oplus \tilde{d}_y}] \\
&= 2^{j-1} \sum_{\substack{c_x, c_y \in C_j \\ \tilde{d}_x \neq \tilde{d}_y \in \tilde{D}_j}} \text{tr}[\rho_E^{c_x \oplus \tilde{d}_x, c_y \oplus \tilde{d}_y}]
\end{aligned} \tag{D.3}$$

Note that the probability of *reaching* round j , is given by the probability of $N_R \geq j$ or the probability of the first $j - 1$ parities matching. This can be computed via

$$\begin{aligned}
\text{Prob}(N_R \geq j) &= 2^{j-1} \sum_{\substack{c_x, c_y \in C_j \\ \tilde{d}_x \tilde{d}_y \in \tilde{D}_j}} \langle e_{c_x \oplus \tilde{d}_x, c_y \oplus \tilde{d}_y}, e_{c_x \oplus \tilde{d}_x, c_y \oplus \tilde{d}_y} \rangle \\
&= 1 - p_{n-k} - \dots p_{j+1}
\end{aligned} \tag{D.4}$$

D.3 Computing Cost of Error Correction

The cost of error correction in the final round, corresponding to all partition bits matching is identical to the value in the CPA protocol. Therefore let $j \in \{1, 2, \dots, n-k\}$, and let us consider the error correction cost for correcting parities in this round. The probability of reaching round j is given by $\text{Prob}(N_R \geq j)$ and is computed above. The probability distribution of the data at this point is given by $P_{N_R \geq j}(d_x \tilde{d}_x c_x, d_y \tilde{d}_y c_y) = \delta_{d_x, d_y} \langle e_{xy}, e_{xy} \rangle / \text{Prob}_{N_R \geq j}$, where $x = c_x \oplus d_x \oplus \tilde{d}_x$ and $y = c_y \oplus d_y \oplus \tilde{d}_y$.

Therefore, $\delta_{\text{leak}}^{(j)} = S(P_X^{(j)} | Y P_X^{(1)} \dots P_X^{(j-1)}) = S(P_{N_R \geq j}(d_x \tilde{d}_x, d_y \tilde{d}_y c_y)) - S(P_{N_R \geq j}(\tilde{d}_x, d_y \tilde{d}_y c_y))$. This can be simplified further, by noting that the probability distribution is blockdiagonal in d_x .

$\delta_{\text{leak}}^{(\perp)}$ is identical to the corresponding term in the CPA protocol.

D.4 Computing Eve's Information on the Key

We will again use the fact that $S(Z|E)$ evaluated on $\rho_{ZE} = \sum_z p(z) \rho_E^z$ is given by

$$S(Z|E)_{\rho_{ZE}} = S(Z) + \sum_z p(z) S(\rho_E^z) - S(\rho_E) \tag{D.5}$$

Thus, computing the above expression is essentially an exercise in computing Eve's conditional states depending on the key register, i.e, ρ_E^z . Again note that if $N_R = \perp$, the term is identical to the corresponding term in the CPA protocol.

We compute $S(Z|E, N_R = j) = (P_X^{(1)} \dots P_X^{(j)}|E, N_R = j)$. To do this, we first compute the state of the protocol conditioned on $N_R = j$,

$$\rho_{XYE|N_R=j} = \sum_{\substack{c_x, c_y \in C_j \\ d_x = d_y \in D_j \\ \tilde{d}_x \neq \tilde{d}_y \in \tilde{D}_j}} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \frac{\rho_E^{x,y}}{p_j} \quad (\text{D.6})$$

In the keymap, $Z = H(1 : j) * X = H(1 : j) * (d_x \oplus \tilde{d}_x)$ is mapped to the key, where $H(1 : j)$ denotes the matrix constructed from the first j rows of H . Since entropies are invariant under relabellings, we can replace $H(1 : j) * (d_x \oplus \tilde{d}_x)$ with $(d_x \oplus \tilde{d}_x)$ in the key register without changing the entropy. This follows from the one-to-one map between the two. Now, to compute Eve's information on the key, we repeat the same procedure as before to get,

$$\begin{aligned} \rho_{ZE|N_R=j} &= \sum_{\substack{c_x, c_y \in C_j \\ d_x = d_y \in D_j \\ \tilde{d}_x \neq \tilde{d}_y \in \tilde{D}_j}} |d_x \oplus \tilde{d}_x\rangle \langle d_x \oplus \tilde{d}_x| \otimes \frac{\rho_E^{c_x \oplus d_x \oplus \tilde{d}_x, c_y \oplus d_y \oplus \tilde{d}_y}}{p_j} \\ \rho_{E|N_R=j}^{d_x \tilde{d}_x} &= \sum_{\substack{e_c \in C_j \\ \tilde{e}_d \in \tilde{D}_j, \tilde{e}_d \neq 0}} \sum_{c_x \in C_j} \frac{2^j \rho_E^{c_x \oplus d_x \oplus \tilde{d}_x, c_x \oplus d_x \oplus \tilde{d}_x \oplus e_c \oplus e_d}}{p_j} \\ \rho_{E|N_R=j} &= \sum_{\substack{e_c \in C_j \\ \tilde{e}_d \neq 0 \in \tilde{D}_j}} \sum_{x \in \{0,1\}^n} \frac{\rho_E^{x, x \oplus e_c \oplus e_d}}{p_j} \end{aligned} \quad (\text{D.7})$$

The entropies of the above states can be computed using Lemmas [B.3](#), [B.4](#). Thus, $p_j S(Z|E, N_R = j)$ can be computed for any value of j . We have now computed all the expressions needed to evaluate Eq [3.17](#).

Appendix E

Numerical details for Recursive keyrate calculations

In this section we explain how we compute the keyrate expression Eq. 4.36

$$K_{\text{Recursive}} = \min_{\rho \in \Gamma_{\text{bell}}} \left(\sum_{i=1}^{t-1} \frac{1}{2^i} \left(\prod_{j=1}^{i-1} p_{j-1} \right) S(\hat{Z}_i^{(i)} | \hat{Z}_{i-1}^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{A}^{(i)})_{\rho_{\hat{Z}_i^{(i)} \dots \hat{Z}_1^{(i)} \hat{E}^{(i)} \hat{A}^{(i)}}} \right. \\ \left. + \frac{1}{2^{t-1}} \left(\prod_{j=1}^{t-1} p_{j-1} \right) S(\hat{Z}_t^{(t)} | \hat{Z}_{t-1}^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{A}^{(t)})_{\rho_{\hat{Z}_t^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)} \hat{A}^{(t)}}} - \delta_{\text{leak}} \right) \quad (\text{E.1})$$

δ_{leak} and p_j has already been computed in Sec. 4.1. Therefore, in this appendix we focus on the computation of the conditional entropy.

E.1 Eve's information on the key

For round j , we wish to compute $S(\hat{Z}_j^{(j)} | \hat{Z}_{j-1}^{(j)} \dots \hat{Z}_1^{(j)} \hat{E}^{(j)} \hat{A}^{(j)})_{\hat{\rho}_{\hat{Z}_j^{(j)} \dots \hat{Z}_1^{(j)} \hat{E}^{(j)} \hat{A}^{(j)}}}$. The idea is to look at bits at round j , and construct the ‘history’ or ‘tree’ of bits from previous rounds which led to this round. Ultimately, we know that the state above is obtained from the processing of some 2^j signals from round 1. Let j be a non-terminal round. Therefore, the bit stored in $\hat{Z}_j^{(j)}$ is the parity of two input rounds to round j . Let us call this bit a_j . Therefore the block in round j generating this bit would be $(x_j, x_j \oplus a_j)$.

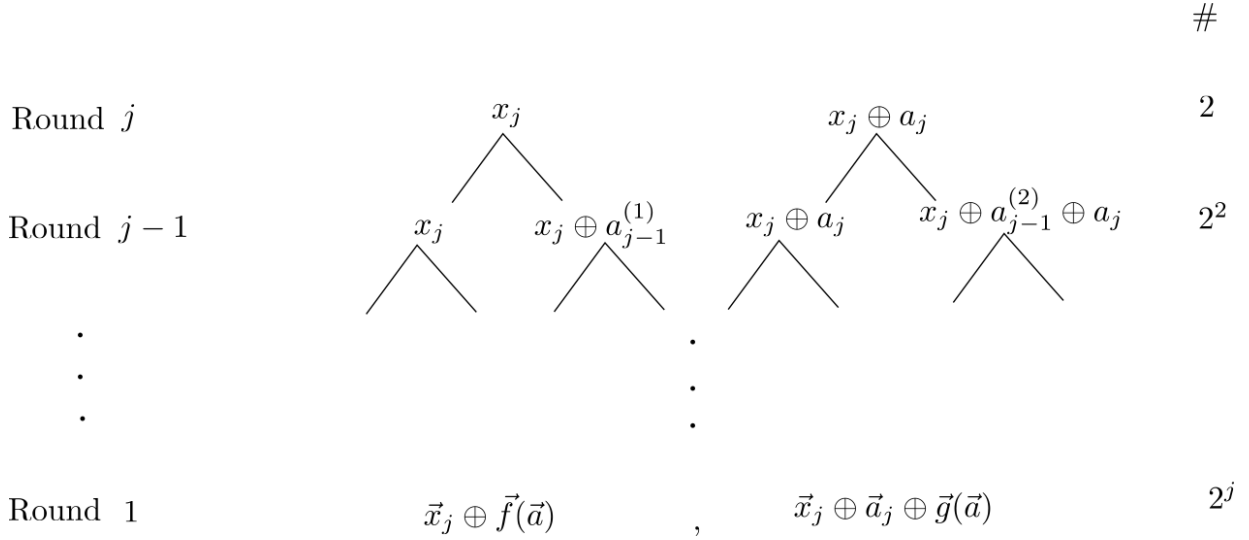


Figure E.1: Schematic of recursive protocol

Each of these bits in the block came from a block of two bits at round $j - 1$. Let those blocks be $(x_j, x_j \oplus a_{j-1}^{(1)})$ and $(x_j \oplus a_j, x_j \oplus a_j \oplus a_{j-1}^{(2)})$. Thus the bit in $\hat{Z}_j$ comes from 4 bits in round $j - 1$, each of which is obtained from 8 bits in round $j - 2$, and so on until we get to 2^j bits in round 1. Let the sequence of parities that appear in this tree by $\vec{a}$ (parities at round $j - 1$ and below). We will see later that only the length of $\vec{a}$ is important. See Fig. E.1 for a schematic diagram.

Therefore, each input bit in round j is a result of processing a block of 2^{j-1} input bits to round 1. Let the input block to x_j be given by $\vec{x}_j \oplus \vec{f}(\vec{a})$, and let the input block to $x_j \oplus a_j$ be given by $\vec{x}_j \oplus \vec{a}_j \oplus \vec{g}(\vec{a})$. Here we use $\vec{x} \oplus \vec{f}(\vec{a})$ to represent a string of bits given by $x_j \oplus \vec{f}(\vec{a})_1, x_j \oplus \vec{f}(\vec{a})_2, \dots, x_j \oplus \vec{f}(\vec{a})_{2^{j-1}}$ ¹.

Let the corresponding bits to $x_j, x_j \oplus a_j$ on Bob's side by $y_j, y_j \oplus b_j$. Now, although in this round and this block, the parity of Alice and Bob (a_j, b_j) can be different, we know that the parities in *all previous blocks must have matched*. This means, that the strings of bits on Bob's side that lead to this block at round j are given by $\vec{y}_j \oplus \vec{f}(\vec{a})$ and $\vec{y}_j \oplus \vec{b}_j \oplus \vec{g}(\vec{a})$. The fact that the *same* $\vec{f}(\vec{a})$ and $\vec{g}(\vec{a})$ appear on both Alice and Bob's side is a result of the fact that all previous parities in the history of each bit in round j were the same for Alice and Bob.

¹i.e, $\vec{x}$ when x is a single bit is x repeated in every position of of the bit string

Another way to think of the above claim is as follows. Each bit of input in round j is a result of the processing of 2^{j-1} signals. During this processing, $2^{j-1} - 1$ parities are generated and mapped to the key in earlier rounds. Therefore, the 2^{j-1} possible bits that make up the input block can be thought of as 1 bit plus $2^{j-1} - 1$ parity bits. This picture is useful, because we know that the parities on Alice and Bob's side are identical.

What is the length of $\vec{a}$? . The answer is $\sum_{k=1}^{j-1} 2^k = 2^j - 2$. This means that in round j , to describe 2 bits and 1 parity, we have to consider 2^{2^j-2} number of possibilities for $\vec{a}$ ². Since we are discussing non-terminal rounds, we have

$$\rho_{\hat{Z}_j^{(j)} \hat{Z}_{j-1}^{(j)} \dots \hat{Z}_1^{(j)} E^{(j)}} = \sum_{\vec{a}, a_j, b_j, x_j, y_j} |a_j\rangle \langle a_j| \otimes |\vec{a}\rangle \langle \vec{a}| \otimes \frac{\rho_E^{\vec{x}_j \oplus \vec{f}(\vec{a}), \vec{y}_j \oplus \vec{f}(\vec{a})} \otimes \rho_E^{\vec{x}_j \oplus \vec{a}_j \oplus \vec{g}(\vec{a}), \vec{y}_j \oplus \vec{b}_j \oplus \vec{g}(\vec{a})}}{f_j} \quad (\text{E.2})$$

where f_j is the normalization factor, given by

$$\begin{aligned} f_j &= \text{Tr}[\rho_E^{\vec{x}_j \oplus \vec{f}(\vec{a}), \vec{y}_j \oplus \vec{f}(\vec{a})} \otimes \rho_E^{\vec{x}_j \oplus \vec{a}_j \oplus \vec{g}(\vec{a}), \vec{y}_j \oplus \vec{b}_j \oplus \vec{g}(\vec{a})}] \\ &= 2^{(2^j-2)} \sum_{a_j, b_j, x_j, y_j} \text{Tr}[\rho_E^{\vec{x}_j, \vec{y}_j} \otimes \rho_E^{\vec{x}_j \oplus \vec{a}_j, \vec{y}_j \oplus \vec{b}_j}] \end{aligned} \quad (\text{E.3})$$

where we have used the fact that the value of $\vec{a}$ does not matter in computing the trace in the above expression ³. To compute $S(\hat{Z}_j^{(j)} | \hat{E}^{(j)} \hat{Z}_1^{(j)} \dots \hat{Z}_{j-1}^{(j)})$, we need

$$\begin{aligned} \rho_{\hat{Z}_1^{(j)} \dots \hat{Z}_{j-1}^{(j)} \hat{E}^{(j)}}^{a_j} &= \frac{2}{f_j} \sum_{\vec{a}, b_j, x_j, y_j} |\vec{a}\rangle \langle \vec{a}| \otimes \rho_E^{\vec{x}_j \oplus \vec{f}(\vec{a}), \vec{y}_j \oplus \vec{f}(\vec{a})} \otimes \rho_E^{\vec{x}_j \oplus \vec{a}_j \oplus \vec{g}(\vec{a}), \vec{y}_j \oplus \vec{b}_j \oplus \vec{g}(\vec{a})} \\ \rho_{\hat{Z}_1^{(j)} \dots \hat{Z}_{j-1}^{(j)} \hat{E}^{(j)}} &= \frac{1}{f_j} \sum_{\vec{a}, a_j, b_j, x_j, y_j} |\vec{a}\rangle \langle \vec{a}| \otimes \rho_E^{\vec{x}_j \oplus \vec{f}(\vec{a}), \vec{y}_j \oplus \vec{f}(\vec{a})} \otimes \rho_E^{\vec{x}_j \oplus \vec{a}_j \oplus \vec{g}(\vec{a}), \vec{y}_j \oplus \vec{b}_j \oplus \vec{g}(\vec{a})} \\ S(\hat{Z}_j^{(j)} | \hat{E}^{(j)} \hat{Z}_1^{(j)} \dots \hat{Z}_{j-1}^{(j)}) &= S(\hat{Z}_j^{(j)}) + \sum_{a_j} p(a_j) S(\rho_{\hat{Z}_1^{(j)} \dots \hat{Z}_{j-1}^{(j)} \hat{E}^{(j)}}^{a_j}) - S(\rho_{\hat{Z}_1^{(j)} \dots \hat{Z}_{j-1}^{(j)} \hat{E}^{(j)}}) \end{aligned} \quad (\text{E.4})$$

Note that each $\vec{a}$ determines an orthogonal block in the above expressions, and there are $2^{(2^j-2)}$ possible $\vec{a}$ s. Moreover, the eigenvalues of the each of the block $|\vec{a}\rangle \langle \vec{a}| \otimes \rho_E^{\vec{x}_j \oplus \vec{f}(\vec{a}), \vec{y}_j \oplus \vec{f}(\vec{a})} \otimes \rho_E^{\vec{x}_j \oplus \vec{a}_j \oplus \vec{g}(\vec{a}), \vec{y}_j \oplus \vec{b}_j \oplus \vec{g}(\vec{a})}$ is independent of $\vec{a}$. Therefore, the total entropy is simply $2^{(2^j-2)}$ times the entropy of one block. We can simply set $\vec{a} = \vec{f}(\vec{a}) = \vec{g}(\vec{a}) = 0$ to compute the entropy of one block. This involves using Lemmas. [B.3](#), [B.1](#).

²Note that here we have clubbed together the history of parities for both bits. Although they are in fact separate, clubbing them together is an easier approach.

³Note here that we can relate f_j to the probability of all the previous parities matching

E.1.1 Terminal round

For the terminal round, the basic approach is the same as above. Suppose the last round is the t th round, and in this round, all input bits are directly mapped to the key.

Consider one bit on Alice's side, x_t . It come from a block of two bits in round $t - 1$, say $x_t, x_t \oplus a_{t-1}^{(1)}$. Each of *these* came from block of two bits in round $t - 2$, say $x_t, x_t \oplus a_{t-2}^{(1)}, x_t \oplus a_{t-1}^{(1)}, x_t \oplus a_{t-1}^{(1)} \oplus a_{t-2}^{(2)}$. Going further, we reach a sequence of 2^{t-1} bits at round 1, let that sequence be $\vec{x}_t \oplus \vec{f}(\vec{a})$, where $\vec{a}$ is the sequence of parities in rounds before round t . On Bob's side, we obtain y_t , and the sequence $\vec{y}_t \oplus \vec{f}(\vec{a})$. This happens because we know that all earlier parities had to match. How many parities show up in the history? The answer is $\sum_{k=0}^{t-2} 2^k = 2^{t-1} - 1$. The state which we consider at this stage is

$$\rho_{\hat{Z}_t^{(t)} \hat{Z}_{t-1}^{(t)} \dots \hat{Z}_1^{(t)} \hat{E}^{(t)}} = \sum_{x_j, y_j, \vec{a}} |x_j\rangle \langle x_j| \otimes |\vec{a}\rangle \langle \vec{a}| \otimes \frac{\rho_{\hat{x}_j \oplus \vec{f}(\vec{a}), \hat{y}_j \oplus \vec{f}(\vec{a})}}{f_t} \quad (\text{E.5})$$

Note that the ρ_E state is on 2^{t-1} bits. The normalization factor is given by

$$\begin{aligned} f_t &= \sum_{x_j, y_j, \vec{a}} \text{Tr}[\rho_E^{\vec{x}_j \oplus \vec{f}(\vec{a}), \vec{y}_j \oplus \vec{f}(\vec{a})}] \\ &= 2^{(2^{t-1}-1)} \sum_{x_j, y_j} \text{Tr}[\rho_E^{\vec{x}_j, \vec{y}_j}] \end{aligned} \quad (\text{E.6})$$

To compute $S(\hat{Z}_t^{(t)} | E \hat{Z}_1^{(t)} \dots \hat{Z}_{t-1}^{(t)})$, we need

$$\begin{aligned} \rho_{\hat{Z}_1^{(t)} \dots \hat{Z}_{t-1}^{(t)} \hat{E}^{(t)}}^{x_t} &= \frac{2}{f_t} \sum_{\vec{a}, y_j} |\vec{a}\rangle \langle \vec{a}| \otimes \rho_E^{x_j \oplus \vec{f}(\vec{a}), y_j \oplus \vec{f}(\vec{a})} \\ \rho_{\hat{Z}_1^{(t)} \dots \hat{Z}_{t-1}^{(t)} \hat{E}^{(t)}} &= \frac{1}{f_t} \sum_{\vec{a}, x_j, y_j} |\vec{a}\rangle \langle \vec{a}| \otimes \rho_E^{x_j \oplus \vec{f}(\vec{a}), y_j \oplus \vec{f}(\vec{a})} \end{aligned} \quad (\text{E.7})$$

$$S(\hat{Z}_t^{(t)} | \hat{Z}_1^{(t)} \dots \hat{Z}_{t-1}^{(t)} E^{(t)}) = S(\hat{Z}_t^{(t)}) + \sum_{x_t} p(x_t) S(\tilde{\rho}_{\hat{Z}_1^{(t)} \dots \hat{Z}_{t-1}^{(t)} \hat{E}^{(t)}}^{x_t}) - S(\rho_{\hat{Z}_1^{(t)} \dots \hat{Z}_{t-1}^{(t)} \hat{E}^{(t)}})$$

Again, we note that each $\vec{a}$ gets its own block in the above expressions, and there are $2^{(2^{t-1}-1)}$ such blocks. Moreover, the eigenvalues of the matrix $|\vec{a}\rangle \langle \vec{a}| \otimes \rho_E^{x_j \oplus \vec{f}(\vec{a}), y_j \oplus \vec{f}(\vec{a})}$ are independent of $\vec{a}$. Therefore, we can simply set $\vec{a} = \vec{f}(\vec{a}) = \vec{g}(\vec{a}) = \vec{0}$, and diagonalize that block only, in order to diagonalize the entire matrix.

Appendix F

Numerical details for AD+CPA keyrate calculations

In this section we explain how we compute the keyrate expression Eq. 4.47

$$K_{\text{AD+CPA}} = \frac{p_1}{2b} S(\hat{Z}|\hat{E}\hat{A})_{\hat{\rho}_{\hat{Z}\hat{E}\hat{A}}} - \delta_{\text{leak}} \quad (\text{F.1})$$

p_1 and δ_{leak} has already been computed in Sec. 4.2. Therefore, in this appendix we focus on the computation of the conditional entropy. We compute $S(\hat{Z}|\hat{E}\hat{F}_{\text{CPA}}\hat{A})_{\hat{\rho}_{\hat{Z}\hat{E}\hat{F}_{\text{CPA}}\hat{A}}} = S(\hat{Z}|\hat{E}\hat{A})_{\hat{\rho}_{\hat{Z}\hat{E}\hat{A}}}$, since $\hat{F}_{\text{CPA}}$ is known to Eve anyway .

F.1 AD Round

The sifting probability and sifted error rate for the AD protocol is given by $p_{\text{sift}} = e^b + (1 - e)^b$, and $\tilde{e} = e^b/p_{\text{sift}}$. In what follows we will separate the b bit string x into a one bit x_1 and $n - 1$ bit s_1 which contains the remaining bits of x . Let the AD announcement for the string $x_1 s_1$ be given by $A(x_1, s_1)$. We have $A(x_1, s_1) = A(\bar{x}_1, \bar{s}_1)$, where we use $\bar{x}$ to denote a string of bits obtained by flipping every bit of x . Then, the Alice-Bob-Eve state

of each bit after AD stage (conditioned on passing the AD filter) is

$$\begin{aligned}\rho_{XYE} = & \sum_{x_1, s_1} |x_1\rangle \langle x_1| \otimes |x_1\rangle \langle x_1| \otimes |A(x_1, s_1)\rangle \langle A(x_1, s_1)| \otimes \frac{\rho_E^{x_1 s_1, x_1 s_1}}{p_{\text{sift}}} \\ & + \sum_{x_1, s_1} |x_1\rangle \langle x_1| \otimes |\bar{x}_1\rangle \langle \bar{x}_1| \otimes |A(x_1, s_1)\rangle \langle A(x_1, s_1)| \otimes \frac{\rho_E^{x_1 s_1, \bar{x}_1 \bar{s}_1}}{p_{\text{sift}}}\end{aligned}\quad (\text{F.2})$$

The first term represents the case where announcements matched and Alice and Bob obtain the same bits, whereas the second term is the case where announcements matched and Alice and Bob obtain different bits. One can write the above state as

$$\begin{aligned}\rho_{XYE} = & \sum_{x_1} |x_1\rangle \langle x_1| \otimes |x_1\rangle \langle x_1| \sigma_E^{x_1, x_1} \\ & + \sum_{x_1} |x_1\rangle \langle x_1| \otimes |\bar{x}_1\rangle \langle \bar{x}_1| \sigma_E^{x_1, \bar{x}_1} \\ \sigma_E^{x_1, x_1} = & \sum_{s_1} |A(x_1, s_1)\rangle \langle A(x_1, s_1)| \otimes \frac{\rho_E^{x_1 s_1, x_1 s_1}}{p_{\text{sift}}} \\ \sigma_E^{x_1, \bar{x}_1} = & \sum_{s_1} |A(x_1, s_1)\rangle \langle A(x_1, s_1)| \otimes \frac{\rho_E^{x_1 s_1, \bar{x}_1 \bar{s}_1}}{p_{\text{sift}}}\end{aligned}\quad (\text{F.3})$$

Thus, if Stage 1 starts with N bits, it ends with (roughly) $p_{\text{sift}} N/b$ bits each with Alice and Bob, with the effective Alice-Bob-Eve state as described above.

F.2 CPA Round

Alice and Bob now implement the CPA protocol with blocksize=2 on $\rho_{\hat{X}\hat{Y}\hat{E}} = \sum_{x,y} |x\rangle \langle x| \otimes |y\rangle \langle y| \otimes \sigma_E^{x,y}$. Let the probability of parities matching be given by $p_m = \tilde{e}^2 + (1 - \tilde{e})^2$.

F.2.1 Eve's information when parities match

If parities match, the two bit state is given by

$$\begin{aligned}\rho_{\hat{X}\hat{Y}\hat{E}|\hat{F}_{\text{CPA}}=0} = & \sum_{x_1 x_2} |x_1 x_2\rangle \langle x_1 x_2| \otimes |x_1 x_2\rangle \langle x_1 x_2| \otimes \frac{\sigma_E^{x_1, x_1} \otimes \sigma_E^{x_2, x_2}}{p_m} \\ & + \sum_{x_1 x_2} |x_1 x_2\rangle \langle x_1 x_2| \otimes |\bar{x}_1 \bar{x}_2\rangle \langle \bar{x}_1 \bar{x}_2| \otimes \frac{\sigma_E^{x_1, \bar{x}_1} \otimes \sigma_E^{x_2, \bar{x}_2}}{p_m}\end{aligned}\quad (\text{F.4})$$

The first term is the case when parity matches and there are no errors, while the second term is the case when parity matches but both bits have errors. When parity matches, the keymap is identity, and $z_1 z_2 = x_1 x_2$ is mapped to the key. Therefore,

$$\rho_{\hat{Z}\hat{E}|\hat{F}_{\text{CPA}}=0} = \sum_{z_1 z_2} \frac{1}{4} |z_1 z_2\rangle \langle z_1 z_2| \otimes \left(\frac{4\sigma_E^{z_1, z_1} \otimes \sigma_E^{z_2, z_2}}{p_m} + \frac{4\sigma_E^{z_1, \bar{z}_1} \otimes \sigma_E^{z_2, \bar{z}_2}}{p_m} \right) \quad (\text{F.5})$$

The entropy of the conditional states can be computed as follows:

$$\begin{aligned} S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=0}^z) &= S\left(\frac{4\sigma_E^{z_1, z_1} \otimes \sigma_E^{z_2, z_2}}{p_m}\right) + S\left(\frac{4\sigma_E^{z_1, \bar{z}_1} \otimes \sigma_E^{z_2, \bar{z}_2}}{p_m}\right) \\ S(\sigma_{\hat{E}|\hat{F}_{\text{CPA}}=0}^{z_1 z_1}) &= -2^{n-1} \left(\frac{\lambda_0 + \lambda_1}{2}\right)^n \frac{1}{p_{\text{sift}}} \log\left(\frac{(\lambda_0 + \lambda_1)^n}{2^n p_{\text{sift}}}\right) \\ &= -\frac{(1 - \tilde{e})}{2} (n \log(\lambda_0 + \lambda_1) - n - \log(p_{\text{sift}})) \\ S(\sigma_{\hat{E}|\hat{F}_{\text{CPA}}=0}^{z_1 \bar{z}_1}) &= -2^{n-1} \left(\frac{\lambda_2 + \lambda_3}{2}\right)^n \frac{1}{p_{\text{sift}}} \log\left(\frac{(\lambda_2 + \lambda_3)^n}{2^n p_{\text{sift}}}\right) \\ &= -\frac{\tilde{e}}{2} (n \log(\lambda_2 + \lambda_3) - n - \log(p_{\text{sift}})) \end{aligned} \quad (\text{F.6})$$

This is because the eigenvalues of $\sigma_{\hat{E}|\hat{F}_{\text{CPA}}=0}^{z_1, z_1}$ are 2^{n-1} copies of $\frac{(\lambda_0 + \lambda_1)^n}{2^n p_{\text{sift}}}$, and the eigenvalues of $\sigma_{\hat{E}}^{z_1, \bar{z}_1}$ are 2^{n-1} copies of $\frac{(\lambda_2 + \lambda_3)^n}{2^n p_{\text{sift}}}$. Note that $\text{Tr}(\sigma_{\hat{E}}^{z_1, z_1}) = \frac{(\lambda_0 + \lambda_1)^n}{2 p_{\text{sift}}} = \frac{1}{2}(1 - \tilde{e})$, and $\text{Tr}(\sigma_{\hat{E}}^{z_1, \bar{z}_1}) = \frac{(\lambda_2 + \lambda_3)^n}{2 p_{\text{sift}}} = \frac{1}{2}\tilde{e}$. The first equation above follows from the observation that Eve's conditional states are orthogonal for different locations for errors. The second and third follow from the expressions for $\sigma_E^{x, y}$ states. The final result is given by

$$\begin{aligned} S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=0}^z) &= \frac{2}{p_m} (1 - \tilde{e}) S(\sigma_E^{z_1 z_1}) + \frac{2}{p_m} (1 - \tilde{e}) S(\sigma_E^{z_2 z_2}) + \frac{(1 - \tilde{e})^2}{4} S\left(\frac{4}{p_m}\right) \\ &\quad + \frac{2}{p_m} \tilde{e} S(\sigma_E^{z_1 \bar{z}_1}) + \frac{2}{p_m} \tilde{e} S(\sigma_E^{z_2 \bar{z}_2}) + \frac{\tilde{e}^2}{4} S\left(\frac{4}{p_m}\right) \\ &= -\frac{2(1 - \tilde{e})^2}{p_m} (n \log(\lambda_0 + \lambda_1) - n - \log(p_{\text{sift}})) - \frac{2\tilde{e}^2}{p_m} (n \log(\lambda_2 + \lambda_3) - n - \log(p_{\text{sift}})) \\ &\quad + \log(p_m/4) \\ &= 2(n + \log(p_{\text{sift}})) - \frac{2n}{p_m} (1 - \tilde{e})^2 \log(\lambda_0 + \lambda_1) - \frac{2n}{p_m} \tilde{e}^2 \log(\lambda_2 + \lambda_3) + \log\left(\frac{p_m}{4}\right) \end{aligned} \quad (\text{F.7})$$

To compute Eve's average state, we have

$$\begin{aligned}
\rho_{\hat{E}|\hat{F}_{\text{CPA}}=0} &= \sum_{z_1 z_2} \left(\frac{\sigma_E^{z_1 z_1} \otimes \sigma_E^{z_2 z_2}}{p_m} + \frac{\sigma_E^{z_1 \bar{z}_1} \otimes \sigma_E^{z_2 \bar{z}_2}}{p_m} \right) \\
&= \sum_{z_1, z_2} \left(\sum_{s_1} |A(z_1 s_1)\rangle \langle A(z_1 s_1)| \otimes \frac{\rho_E^{z_1 s_1, z_1 s_1}}{p_{\text{sift}}^2 p_m} \right) \left(\sum_{s_2} |A(z_2 s_2)\rangle \langle A(z_2 s_2)| \otimes \frac{\rho_E^{z_2 s_2, z_2 s_2}}{p_{\text{sift}}^2 p_m} \right) \\
&\quad + \sum_{z_1, z_2} \left(\sum_{s_1} |A(z_1 s_1)\rangle \langle A(z_1 s_1)| \otimes \frac{\rho_E^{z_1 s_1, \bar{z}_1 \bar{s}_1}}{p_{\text{sift}}^2 p_m} \right) \left(\sum_{s_2} |A(z_2 s_2)\rangle \langle A(z_2 s_2)| \otimes \frac{\rho_E^{z_2 s_2, \bar{z}_2 \bar{s}_2}}{p_{\text{sift}}^2 p_m} \right)
\end{aligned} \tag{F.8}$$

Now, we note that the two terms added above are separate orthogonal blocks and their entropies can be computed separately. Additionally, the announcement register gives us another block diagonal structure. Given any $z_1 s_1$, only $\bar{z}_1 \bar{s}_1$ can give the same announcement. This leads to multiple blocks, where each block represents one set of announcements, with 4 terms in each blocks : Given by $(z_1 s_1, z_2 s_2), (z_1 s_1, \bar{z}_2 \bar{s}_2), (\bar{z}_1 \bar{s}_1, z_2 s_2), (\bar{z}_1 \bar{s}_1, \bar{z}_2 \bar{s}_2)$. A simple counting argument shows that there will be 4^{n-1} such blocks. From Lemmas. B.3, B.1, it is easy to show that all these blocks have the same eigenspectrum, and therefore the same entropy, since they correspond to the same Gram Gatrix. Using this, we obtain

$$\begin{aligned}
S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=0}) &= 4^{n-1} \left\{ S \left((\rho_E^{z_1 s_1, z_1 s_1} + \rho_E^{\bar{z}_1 \bar{s}_1, \bar{z}_1 \bar{s}_1})(\rho_E^{z_2 s_2, z_2 s_2} + \rho_E^{\bar{z}_2 \bar{s}_2, \bar{z}_2 \bar{s}_2}) \frac{1}{p_{\text{sift}}^2 p_m} \right) \right. \\
&\quad \left. + S \left((\rho_E^{z_1 s_1, \bar{z}_1 \bar{s}_1} + \rho_E^{\bar{z}_1 \bar{s}_1, z_1 s_1})(\rho_E^{z_2 s_2, \bar{z}_2 \bar{s}_2} + \rho_E^{\bar{z}_2 \bar{s}_2, z_2 s_2}) \frac{1}{p_{\text{sift}}^2 p_m} \right) \right\}
\end{aligned} \tag{F.9}$$

In this way, $S(\hat{Z}|\hat{E}, \hat{F}_{\text{CPA}} = 0) = 2 + \sum_{z_1 z_2} p(z_1 z_2) S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=0}^{z_1 z_2}) - S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=0})$ can be computed.

F.2.2 Eve's information when parities don't match

The full state in this case is given by

$$\begin{aligned}
\rho_{\hat{X}\hat{Y}\hat{E}|\hat{F}_{\text{CPA}}=1} &= \sum_{x_1 x_2} |x_1 x_2\rangle \langle x_1 x_2| \otimes |\bar{x}_1, x_2\rangle \langle \bar{x}_1, x_2| \frac{\sigma_E^{x_1, \bar{x}_1} \otimes \sigma_E^{x_2, x_2}}{1 - p_m} \\
&\quad + \sum_{x_1 x_2} |x_1 x_2\rangle \langle x_1 x_2| \otimes |x_1 \bar{x}_2\rangle \langle x_1 \bar{x}_2| \frac{\sigma_E^{x_1, x_1} \otimes \sigma_E^{x_2, \bar{x}_2}}{1 - p_m}
\end{aligned} \tag{F.10}$$

Here the keymap is just the parity bit. Let the parity bit be a . Therefore, $x_2 = x_1 \oplus a$. Therefore the state after keymap is given by

$$\begin{aligned}\rho_{\hat{Z}\hat{E}|\hat{F}_{\text{CPA}}=1} &= \sum_a |a\rangle \langle a| \otimes \left(\sum_{x_1} \frac{\sigma_E^{x_1, \bar{x}_1} \otimes \sigma_E^{x_1 \oplus a, x_1 \oplus a} + \sigma_E^{x_1, x_1} \otimes \sigma_E^{x_1 \oplus a, \overline{x_1 \oplus a}}}{1 - p_m} \right) \\ \rho_{\hat{E}|\hat{F}_{\text{CPA}}=1}^a &= \sum_{x_1} \frac{2\sigma_E^{x_1, \bar{x}_1} \otimes \sigma_E^{x_1 \oplus a, x_1 \oplus a} + 2\sigma_E^{x_1, x_1} \otimes \sigma_E^{x_1 \oplus a, \overline{x_1 \oplus a}}}{1 - p_m}\end{aligned}\quad (\text{F.11})$$

Again we will utilize the fact that Eve's stats form different blocks depending on $\rho_E^{x,y}$ (location of errors) and the announcement register. This gives us

$$\begin{aligned}\rho_{\hat{E}|\hat{F}_{\text{CPA}}=1}^a &= \frac{2}{p_{\text{sift}}^2(1 - p_m)} \sum_{x_1} \left(\sum_{s_1} |A(x_1, s_1)\rangle \langle A(x_1, s_1)| \rho_E^{x_1 s_1, \bar{x}_1 \bar{s}_1} \right) \\ &\quad \left(\sum_{s_2} |A(x_1 \oplus a, s_2)\rangle \langle A(x_1 \oplus a, s_2)| \rho_E^{(x_1 \oplus a)s_1, (x_1 \oplus a)s_2} \right) \\ &\quad + \frac{2}{p_{\text{sift}}^2(1 - p_m)} \sum_{x_1} \left(\sum_{s_1} |A(x_1, s_1)\rangle \langle A(x_1, s_1)| \rho_E^{x_1 s_1, x_1 s_1} \right) \\ &\quad \left(\sum_{s_2} |A(x_1 \oplus a, s_2)\rangle \langle A(x_1 \oplus a, s_2)| \rho_E^{(x_1 \oplus a)s_1, \overline{(x_1 \oplus a)s_2}} \right)\end{aligned}\quad (\text{F.12})$$

Note that the two terms added together above are in orthogonal spaces and therefore can be diagonalized separately. Consider the first term : For a given x_1, s_1, s_2 term, we see $A(x_1, s_1)$ and $A(x_1 \oplus a, s_2)$ in the announcement registers. The only way to get the same announcement is $A(\bar{x}_1, \bar{s}_1) = A(x_1, s_1)$ and $A(\bar{x}_1 \oplus a, \bar{s}_2) = A(x_1 \oplus a, s_2)$, i.e the term obtained from $\bar{x}_1, \bar{s}_1, \bar{s}_2$. A simple counting argument shows that this means we will obtain 4^{n-1} blocks, with each block being 2×2 matrix. We will see that each block has the same eigenvalues from Lemmas B.3, B.1. A similar argument for the second term gives us

$$\begin{aligned}S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=1}^a) &= \\ 4^{n-1} S \left(\frac{2}{p_{\text{sift}}^2(1 - p_m)} (\rho_E^{x_1 s_1, \bar{x}_1 \bar{s}_1} \otimes \rho_E^{(x_1 \oplus a)s_2, (x_1 \oplus a)s_2} + \rho_E^{\bar{x}_1 \bar{s}_1, x_1 s_1} \otimes \rho_E^{\overline{(x_1 \oplus a)s_2}, \overline{(x_1 \oplus a)s_2}}) \right) \\ 4^{n-1} S \left(\frac{2}{p_{\text{sift}}^2(1 - p_m)} (\rho_E^{x_1 s_1, x_1 s_1} \otimes \rho_E^{(x_1 \oplus a)s_2, \overline{(x_1 \oplus a)s_2}} + \rho_E^{\bar{x}_1 \bar{s}_1, \bar{x}_1 \bar{s}_1} \otimes \rho_E^{\overline{(x_1 \oplus a)s_2}, (x_1 \oplus a)s_2}) \right)\end{aligned}\quad (\text{F.13})$$

Now we move to computing $S(\rho_{E|\hat{F}_{\text{CPA}}=1})$. We write

$$\begin{aligned}\rho_{\hat{E}|\hat{F}_{\text{CPA}}=1} &= \sum_{a,x_1} \frac{\sigma_E^{x_1,\bar{x}_1} \otimes \sigma_E^{x_1 \oplus a, x_1 \oplus a} + \sigma_E^{x_1,x_1} \otimes \sigma_E^{x_1 \oplus a, \overline{x_1 \oplus a}}}{1 - p_m} \\ &= \sum_{x_1 x_2} \frac{\sigma_E^{x_1,\bar{x}_1} \otimes \sigma_E^{x_2,x_2} + \sigma_E^{x_1,x_1} \otimes \sigma_E^{x_2,\bar{x}_2}}{1 - p_m}\end{aligned}\tag{F.14}$$

The two terms added together live in orthogonal spaces and their entropies can be computed separately. They will also give identical entropies. Therefore

$$\begin{aligned}S(\bar{\rho}_{\hat{E}}) &= 2S\left(\sum_{x_1 x_2} \frac{\sigma_E^{x_1,\bar{x}_1} \otimes \sigma_E^{x_2,x_2}}{1 - p_m}\right) \\ &= 2S\left(\frac{1}{p_{\text{sift}}^2(1 - p_m)} \sum_{x_1 x_2} \left(\sum_{s_1} |A(x_1, s_1)\rangle \langle A(x_1, s_1)| \otimes \rho_E^{x_1 s_1, \bar{x}_1 \bar{s}_1}\right) \right. \\ &\quad \left. \left(\sum_{s_2} |A(x_2, s_2)\rangle \langle A(x_2, s_2)| \otimes \rho_E^{x_2 s_2, x_2 s_2}\right)\right) \\ &= 2 \times 4^{n-1} S\left(\frac{1}{p_{\text{sift}}^2(1 - p_m)} (\rho_E^{x_1 s_1, \bar{x}_1 \bar{s}_1} + \rho_E^{\bar{x}_1 \bar{s}_1, x_1 s_1}) \otimes (\rho_E^{x_2 s_2, x_2 s_2} + \rho_E^{\bar{x}_2 \bar{s}_2, \bar{x}_2 \bar{s}_2})\right)\end{aligned}\tag{F.15}$$

where we have repeated our standard tricks using Lemmas B.3, B.1. In this way $S(\hat{Z}|\hat{E}, \hat{F}_{\text{CPA}} = 1) = 1 + \sum_z p(z) S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=1}^z) - S(\rho_{\hat{E}|\hat{F}_{\text{CPA}}=1})$ can be computed. Note that $S(\hat{Z}|\hat{E}) = S(\hat{Z}|\hat{E}\hat{F}_{\text{CPA}}) = \sum_f \text{Prob}(\hat{F}_{\text{CPA}} = f) S(\hat{Z}|\hat{E}, \hat{F}_{\text{CPA}} = f)$. Thus we have computed everything that is needed to evaluate our keyrate.