Eur. Phys. J. D **18**, 161–170 (2002)
DOI: 10.1140/epjd/e20020020

THE EUROPEAN
PHYSICAL JOURNAL D
EDP Sciences

Shor and Preskill's and Mayers's security proof for the BB84 quantum key distribution protocol

D. Mayers[a]

NEC Research Institute, Maharishi University of Management, 4 Independence way, Princeton NJ-08540, USA

Received 14 July 2001

Abstract. We review two security proofs for the BB84 quantum key distribution protocol: Mayers's security proof and the more recent proof of Shor and Preskill. We focus on the basic principles and the intuition in Mayers's proof instead of technical details. We present a variation on Shor's and Preskill's proof which is convenient for purpose of comparison. We explain the connection between these two proofs.

PACS. 03.65.Ta Foundations of quantum mechanics; measurement theory – 42.50.Dv Nonclassical field states; squeezed, antibunched, and sub-Poissonian states; operational definitions of the phase of the field; phase measurements – 89.70.+c Information science

1 Introduction

Many proofs of security for the BB84 quantum key distribution (QKD) protocol were proposed, but two of them have a special interest. The author's proof established the unconditional security of QKD much before any other proof confirmed the same result. Moreover, as we shall see, it still provides today one the strongest security result for QKD. The proof of Shor and Preskill is definitely interesting because it brings a new light on the security issue, with connection with quantum error-correction, the notion of classical privacy amplification *versus* quantum privacy amplification, etc. It is very interesting to see these connections.

Before we discuss the details of the protocol, the proofs and these interesting connections, let us describe the general model for quantum protocols that we use. We use the circuit model, which is well-known, so we will put the emphasis on what is specific to quantum protocols, beginning by the transmission gates. A transmission gate which we represent by a triangle (for example, see Fig. 1), acts on two registers, one for the sender and another one for the receiver. Before the transmission, only the register of the sender contains useful information, the other one is in the fixed state $|0\rangle$. The transmission gate swaps the content of these two registers. Only the sender's register is shown before the gate and only the receiver register is shown after the gate (but formally both registers exist all the time). Furthermore, if the transmitted qbit is a *classical-quantum qbit* (*i.e.*, a qbit that represents a classical bit), the transmission gate copies this qbit before the swapping.

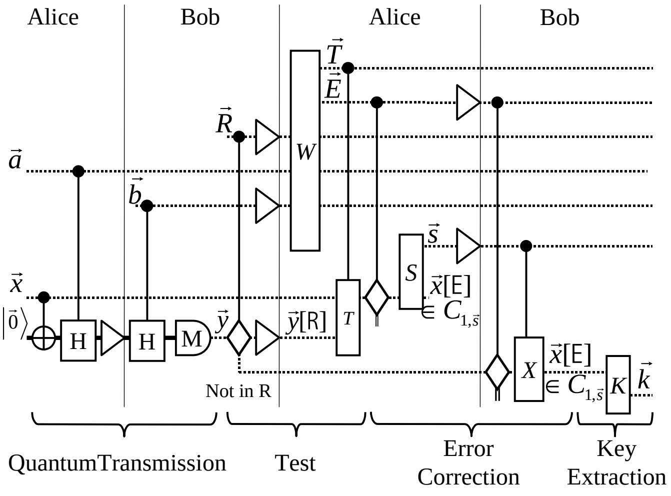

Fig. 1. The BB84 QKD protocol.

To copy a qbit, a CNOT is executed between this register used as the control qbit and a fresh target qbit initially in state $|0\rangle$. This copy operation is required to guarantee the classical behavior of the qbit against a dishonest receiver. It is also necessary because the sender might still need the classical bit later: the fresh target qbit represents the classical bit on the sender's side. We will call such a copy a CNOT-measurement gate, though it is just an ordinary CNOT gate, to remind us that the gate is useful to store a result.

Classical computation is done with CNOT and NOT gates on classical-quantum qbits. The only gates that connect classical-quantum qbits and ordinary qbits are

[a] e-mail: mayers@research.nj.nec.com

CNOT-measurement gates that correspond to measurements on the ordinary qbits, and in this case the classical-quantum qbit is a fresh qbit initially in state $|0\rangle$. To create a random classical bit with distribution $p(0)$, $p(1)$ one creates a fresh qbit, a classical-quantum qbit, in the state $\sqrt{p(0)}|0\rangle + \sqrt{p(1)}|1\rangle$. (Note that non random bits correspond to the case $p(0) = 1$ or $p(0) = 0$.) It should be clear to the reader that no CNOT-measurement gate is needed on any classical-quantum qbit until it is transmitted. The required CNOT-measurement gate is already included as part of the transmission gate. The details are provided in Appendix A.

Now, we describe the circuit of the BB84 quantum key distribution (QKD) protocol. We separated this circuit in four phases:

(1) the quantum transmission;
(2) the test;
(3) error-correction;
(4) and key extraction (but only Bob's key extraction is shown in the circuit of Fig. 1).

We begin by the quantum transmission. Alice picks two random strings $\mathbf{x}, \mathbf{a} \in \{0,1\}^N$, prepares the state $|\mathbf{x}\rangle$ using CNOT gates, applies an Hadamar transformation for each i with $a[i] = 1$, and sends the state to Bob. Next, Bob picks a random string $\mathbf{b} \in \{0,1\}^N$, applies an Hadamar transform on the photons at each position i with $b[i] = 1$, and measures the photons in the computational bases to obtain the string $\mathbf{y}$. Now, we describe the test. Bob picks a random string $\mathbf{R} \in \{0,1\}^N$. He announces the string $\mathbf{b}$ and $y[i]$ for every i with $R[i] = 1$. In the circuit W, Alice computes the string $\mathbf{T}$ defined *via* $T[i] = R[i]\cdot(a[i]\oplus b[i]\oplus 1)$ and the string $\mathbf{E}$ defined *via* $E[i] = (R[i]\oplus 1)\cdot(a[i]\oplus b[i]\oplus 1)$. The positions i with $T[i] = 1$ are then used for the test. In the circuit T, if the number of errors, *i.e.* positions i with $T[i] = 1$ and $x[i] \neq y[i]$ is greater than a number $d > 0$ fixed in advance in the protocol, the protocol aborts and the key is set to be the null string. Now, we describe error-correction. In the circuit S, Alice computes r bits of parity $s[j] = \oplus_{i\in E} F[j,i]x[i]$ where F is a $r \times |E|$ parity check matrix for an error-correcting code C_1, and the columns of F are indexed by $i \in E$. The circuit W contains a register $s[j]$ initially in state $|0\rangle$ for every j and one CNOT gate, with $s[j]$ as the target qbit and $x[i]$ as the control qbit, for every pair of positions (j,i) for which $F[j,i] = 1$. The string $\mathbf{s}$ is called a syndrome of the parity check matrix F. She sends the syndrome to Bob and, in the circuit X, Bob uses it to correct the errors in $\mathbf{y}[E]$ and thus obtains $\mathbf{x}[E]$. The circuit X needs only to contain a classical computation on the syndrome $\mathbf{s}$ which determines which positions must be corrected *via* a NOT operator. In practice, this is not efficient, but it is enough to prove the security of this variation on the protocol since as explained in [3] the more efficient variations are not less secure. The string $\mathbf{g}[E]$, now shared by Alice and Bob, is in the coset $C_{1,s} = \{\alpha \in \{0,1\}^E | F \odot \alpha = s\}$ of the code C_1. Finally, for key extraction, using a circuit K, Alice and Bob compute the key bits $k[j] = \oplus_{i\in E} K[j,i]x[i]$, where K is a $m \times |E|$ matrix picked uniformly at random. (Only the key extraction executed by Bob is shown in Fig. 1).

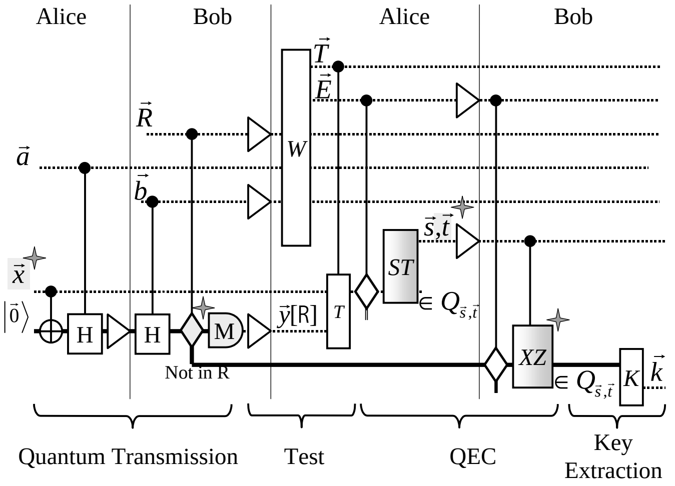

Fig. 2. The BB84-CSS QKD protocol.

Now, let us consider Shor and Preskill's proof. To simplify the analysis we consider a variation on the proof. We will reduce the security of the BB84 protocol to another protocol which we call the BB84-CSS protocol. The circuit for this protocol is shown in Figure 2. The protocol is a variation on what Shor and Preskill [2] call the Modified Lo-Chau protocol. It has the advantage of being closer to the BB84 protocol. There are only a few differences between the BB84-CSS and the BB84 protocols which we now describe. Stars are used in Figure 2 to show where these differences occur. First, in the BB84-CSS protocol we replace each classical-quantum random bit $x[i]$ by an ordinary quantum qbit in the state $(|0\rangle + |1\rangle)/\sqrt{2}$. As explained before, this would only affect the behavior of an eventual transmission gate (but there is none in this case). Second, in the BB84-CSS protocol, Bob only measures the tested photons (*i.e.* at positions i with $R[i] = 1$) and obtain $\mathbf{y}[R]$. The other photons are not measured. Third, in the BB84-CSS protocol, in addition and after the computation of the syndrome $\mathbf{s}$, Alice computes an extra syndrome $\mathbf{t}$ for phase flip error-correction. We will discuss how $\mathbf{t}$ is defined later. Fourth, Bob uses the syndrome $\mathbf{t}$ to do phase flip error-correction after bit flip error-correction.

The proof proceeds in two steps. First we show that, if the BB84-CSS protocol is secure, the BB84 protocol is also secure. This is the reduction. Second, we show that the BB84-CSS protocol is secure. We now proceed with the first step. The concept of reduction is standard in cryptography. Here we use the most simple kind of reduction. We show that for any attack against the BB84 protocol, there exists a corresponding attack that is as successful against the BB84-CSS protocol. Clearly, if we next show that there is no successful attack against the BB84-CSS protocol, then we have that there is no successful attack against the BB84 protocol. Note that Eve needs only two circuits (not shown in Fig. 1): one during the quantum transmission, and a final one on the extra registers on her side after she learned the bases. The attack on the BB84-CSS protocol is simply that Eve uses the exact same two circuits as against the BB84 protocol, and ignores the extra syndrome $\mathbf{t}$.

So, for Eve's circuits fixed, we want to show that Eve is not less successful in the BB84-CSS protocol than in the BB84 protocol. We will pass from the BB84 protocol to the BB84-CSS protocol in three steps such that at each step Eve has no less information in the new protocol than in the previous one. At each step, we add or remove an operation which could equivalently be executed at the end of the protocol after Eve's final circuit, and therefore executing or not this operation makes no difference: the operation does not influence the behavior of the reminder of the protocol.

First, we replace the classical-quantum qbits $x[i]$ by ordinary quantum qbit. This only means that eventual transmission gates will not CNOT-measure them anymore before the transmission, but these bits are not transmitted anyway (see Fig. 1), and so it makes no difference at all. Second, we take out Bob's measurements on the photons which are not tested. These measurements are like CNOT-measurement gates in which the measured photons are the control qbits. Later on, these photons are only used again in the circuit X for bit flip error-correction and the circuit K for key extraction (see Fig. 1). The circuit X needs only to use NOT gates on these photons. Clearly, these CNOT-measurement operations commute with these NOT operations modulo a final NOT operation on their target qbits which are not used anyway (see Appendix A). Similarly, these CNOT-measurement operations commute with circuit K for key extraction because in both cases the photons are only used as control qbits. Every thing else that occur later in the protocol is done on different registers, and thus commute with these CNOT-measurement operations. So these CNOT-measurement operations can be equivalently executed after every thing that occur later in the protocol. Third, we add the computation and announcement of $\mathbf{t}$ by Alice together with phase flip error-correction by Bob. Phase flip error-correction is the last step before key extraction and Eve's final measurement. It will not influence Eve's final measurement because Eve does it on a different system, but we need to check that it does not influence key extraction. This is not hard to check. In principle, phase flip errors can be corrected using the conditional phase flip (also denoted σ_3) operator which is the NOT operator in the Hadamar basis (see lemma 1). The syndrome $\mathbf{t}$ only indicates on which photons to execute this conditional phase shift operator. In practice, this is not efficient, but who cares since this is just part of a proof. Key extraction is done in the computational basis. Clearly, conditional phase shift operators do not influence the computation of the key by Bob and they could be done later in the protocol (or not at all). Now, we need to check that the computation and the announcement of the syndrome $\mathbf{t}$ can also be moved at the very end of the protocol. We will see later (see lemma 1) that the computation of the syndrome $\mathbf{t}$ (by Alice) commutes with the computation of the key $\mathbf{k}$ (by Alice). Every thing else that occurs later in the protocol is done on a different system and ignores $\mathbf{t}$ (see Fig. 1 and recall that Eve's ignore the extra syndrome $\mathbf{t}$). So, the computation-announcement of $\mathbf{t}$ commutes with every operation that occurs later in the current protocol. This concludes the proof.

Clearly, in view of the fact that we minimized as much as possible the differences between the BB84-CSS protocol and the BB84 protocol, the above reduction is as simple as it can be. In the original proof of Shor and Preskill, they used the modified Lo-Chau protocol which is interesting for an historical reason, but unfortunately the matching with the BB84 protocol is not as good as with the BB84-CSS protocol.

Now, we proceed with the second step, the proof that the BB84-CSS is secure. In the BB84 case, the strings of the coset $C_{1,s}$ are so far away from one to another (in the Hamming distance) that, if the number of errors in Bob's string is small, Bob can recover Alice's string. If we think of these classical strings as classical distribution over the coset $C_{1,s}$, we can say that, after error-correction, Alice and Bob respective distribution of strings are perfectly correlated. In the BB84-CSS case, the situation is similar, except that instead of only a classical correlation, they obtain a perfect entanglement between their respective quantum coset $Q_{s,t}$, a subspace of the state space of the n photons in E which we will defined later. (By "in E" we mean at positions i with $E[i] = 1$.) So, after quantum error correction and decoding, Alice and Bob share almost perfect Φ^+ pairs. Therefore, the extracted key is private, even if Eve receives from Alice the additional syndrome for phase flip error correction. This is the basic idea of the BB84-CSS security proof. This is not the most innovative part of the proof of Shor and Preskill because in a way it is only the statement that quantum error-correction works in a particular model of errors, the one that is forced by the test. In fact, this part of the proof was essentially skipped in the original paper of Shor and Preskill [2] and was done later for the perfect apparatus case in [5]. We will not cover this part of the proof except for a brief discussion in connection with Mayers's (the author's) proof.

However, it is interesting to understand the concepts of quantum error-correction in a new context. Besides, we also need to provide two lemma that are used in the proof of the reduction. We start by considering the notion of classical privacy amplification over classical error-correction (used in Mayers's original proof and hidden in Shor's and Preskill's proof), and then we go into the related concept of CSS codes and entanglement purification.

Classical privacy amplification together with error correction can be done with a $(r + m) \times n$ parity check matrix G for a linear code C_2. We assume that all rows of G are linearly independent in F_2^n. (Here F_2^n is the n dimensional vector space over the finite field $F_2 = \{0, 1\}$ with standard multiplication and addition modulo 2.) The r first rows at the bottom of G form a parity check matrix H_1 for a larger code C_1, which corresponds to the linear code used in the BB84 protocol. We have $\{0\} \subset C_2 \subset C_1 \subset F_2^n$. The syndrome $H_1 \cdot \mathbf{x} = \mathbf{s}$ is used to correct errors in the string $\mathbf{x}$. Here the dot "$\cdot$" represents the standard matrix multiplication (with a sum modulo 2). The top m

rows form a matrix K used to extract the key which is given by $\mathbf{k} = K \cdot \mathbf{x}$, as in the BB84 protocol.

The associated CSS code $Q = Q(C_1, C_2)$ is a subspace of C^{2^n} protected against errors in a small number of qbits. The quantum states

$$\Psi(x,z) = \frac{1}{|C_2|^{1/2}} \sum_{w \in C_2} (-1)^{w \cdot z} |x + w\rangle \qquad (1)$$

are called "string-states" and will be used to define the CSS code. Let H_2 be any $(n-m-r) \times n$ parity check matrix for $C_2^{\perp}$, the dual of C_2. The following proposition is helpful to understand CSS quantum error-correcting code. It is not hard to prove using techniques described in [3], so we only state it.

Proposition 1. *If we rewrite the string-states* $\Psi(\mathbf{x}, \mathbf{z})$ *in the new basis states* $|\mathbf{w}\rangle_H$ *(obtained from the old basis states* $|\mathbf{w}\rangle$ *by an Hadamar transform* H *on all* n *qbits), we get:*

$$\Psi(\mathbf{x},\mathbf{z}) = \frac{(-1)^{\mathbf{z}\cdot\mathbf{x}}}{|C_2^{\perp}|^{1/2}} \sum_{\mathbf{w} \in C_2^{\perp}} (-1)^{\mathbf{w}\cdot\mathbf{x}} |\mathbf{z} + \mathbf{w}\rangle_H. \qquad (2)$$

To our knowledge this proposition was never used before. Using proposition 1, we now prove the following lemma.

Lemma 1. *The three circuits which respectively computes* s, k *and* t *pairwise commute, and the string-states are in one-to-one correspondence with the* 2^n *triplets* $(\mathbf{s}, \mathbf{k}, \mathbf{t})$.

Remark. In accordance with this lemma, whenever it is convenient, the string-states will be denoted $\Psi(\mathbf{s}, \mathbf{k}, \mathbf{t})$ instead of $\Psi(\mathbf{x}, \mathbf{z})$.

Proof of lemma 1. For the first part of the lemma, it will be sufficient to show that each of these three circuits defines an orthogonal measurement which is diagonal in the basis of string-states. For the circuits S and K which classically compute s and k, the argument is simply that, because C_2 is the linear code with parity check matrix G, every basis state $|\mathbf{x} + \mathbf{w}\rangle$ in formula (1) encodes the same syndrome-key pair $(\mathbf{s}, \mathbf{k}) = G \odot (\mathbf{z} + \mathbf{w})$. So, when these circuits act on any given string-state, they only return the encoded $\mathbf{s}$ and $\mathbf{k}$, and leave invariant the string-state. The collapse operation associated with the outcome $(\mathbf{s}, \mathbf{k})$ is the projection on the span of the string-states that encode $\mathbf{s}$ and $\mathbf{k}$. For the circuit T, the argument is similar except that we use formula (2) which uses the new basis $\{|w\rangle_H\}$. (We did not mention it before, but $\mathbf{t}$ is computed in the same manner as $\mathbf{s}$ but using the new basis $\{|w\rangle_H\}$.) The second part of the lemma is easily obtained from the first part using the fact that there are 2^n different string-states. This concludes the proof.

We will try to understand quantum error-correction with the help of the previous lemma. However, note that what is going on in error-correction is not entirely obvious even in the classical case. For example, it is interesting to note that one only needs to know the syndrome $\mathbf{s}$ of the received string, not the entire string, to determine which positions must be flipped back. The situation is very much the same in the quantum case, except that the classical code (which is a set of strings) is replaced by a CSS code (which is a subspace generated by a set of string-states). As we will see, the string-states of a CSS code are far away (in the Hamming distance) for both x and z.

We recall from classical error correcting code that $C_{\mathbf{s}} = \{\mathbf{x} \in F_2^n \mid H_1 \cdot \mathbf{x} = \mathbf{s}\}$ is called a coset of the code associated with H_1, and the particular coset with $\mathbf{s} = \mathbf{0}$ is the linear code itself. By analogy, the "quantum coset" $Q_{\mathbf{s},\mathbf{t}} = Q_{\mathbf{s},\mathbf{t}}(C_1, C_2)$ of a CSS quantum code with syndromes $\mathbf{s}$ and $\mathbf{t}$ is defined *via*

$$Q_{\mathbf{s},\mathbf{t}} = \mathrm{Span}\{\Psi(\mathbf{x},\mathbf{z}) \,|\, H_1 \cdot \mathbf{x} = \mathbf{s} H_2 \cdot \mathbf{z} = \mathbf{t}\} \qquad (3)$$

and the CSS code Q itself is the particular case where $\mathbf{s} = \mathbf{0}$ and $\mathbf{t} = \mathbf{0}$. As mentioned in [2], the same CSS code Q is obtained if we use $(C_2^{\perp}, C_1^{\perp})$ instead of (C_1, C_2), but the basis of string-states are different. We will not use this alternative representation.

Note that the only degree of freedom left in the string-states of a coset $Q_{\mathbf{s},\mathbf{t}}$ is the outcome of a measurement of the key $\mathbf{k}$ (using the circuit K). So $\mathbf{s}$ and $\mathbf{t}$ specify one of the $2^{r+(n-m-r)}$ cosets and each state $|\mathbf{k}\rangle$ is encoded in the state $\Psi(\mathbf{s}, \mathbf{k}, \mathbf{t})$ in this coset. The string-states in the CSS code $Q_{0,0}$ are called the *codewords* of the CSS codes.

For *entanglement purification* of $\rho \approx (\phi^+)^{\otimes n}$, Alice measures the syndromes $\mathbf{s} = H_1 \cdot \mathbf{x}$ and $\mathbf{t} = H_2 \cdot \mathbf{z}$ and announces them to Bob. If we had exactly $\rho = (\phi^+)^{\otimes n}$, the residual state would be the exact encoding

$$2^{-m} \sum_{\mathbf{k} \in \{0,1\}^m} \Psi(\mathbf{s},\mathbf{k},\mathbf{t}) \otimes \Psi(\mathbf{s},\mathbf{k},\mathbf{t}) \in Q_{\mathbf{s},\mathbf{t}} \otimes Q_{\mathbf{s},\mathbf{t}}$$

of $2^{-m} \sum_{\mathbf{k} \in \{0,1\}^m} |\mathbf{k}\rangle \otimes |\mathbf{k}\rangle$. As we will see, if the number of bit flips and phase flips errors is small, Alice and Bob can recover this exact encoding using quantum-error correction on Bob's side, and eventually they can decode it on their respective side to obtain $\mathbf{k}$.

Hereafter, we only consider quantum error-correction on Bob's side. For the intuition, we briefly review the classical case. A non zero syndrome $\mathbf{s}$ corresponds to an error $\mathbf{u}$ such that $H \odot \mathbf{u} = \mathbf{s}$. The initial string is $\mathbf{x}$ with $H \odot \mathbf{x} = \mathbf{0}$, and the final string is $\mathbf{x} \oplus \mathbf{u}$. Note that the error transformation is only *a posteriori* defined. It is defined as the translation that sends the actual initial state to the actual final state. It is not necessarily the actual error process. The actual error process could define a different translation for every initial string. In fact, usually the error mechanism is probabilistic and there is no fixed translation associated with an initial state. If we have a good error-correcting code, for most error transformation $\mathbf{u}$ in the error model (in probabilistic sense), there is no other transformation $\mathbf{u}'$ also in this error model that defines the same syndrome $\mathbf{s}$. In other words, except with a small probability, the syndrome should tell you what is the way to undo the error. For example, if the minimal distance of the code is d and the error transformations $\mathbf{u}$ are guarantee to have weight smaller than $\lfloor d/2 \rfloor$, then for every syndrome $\mathbf{s}$ there is at most one error-transformation $\mathbf{u}$

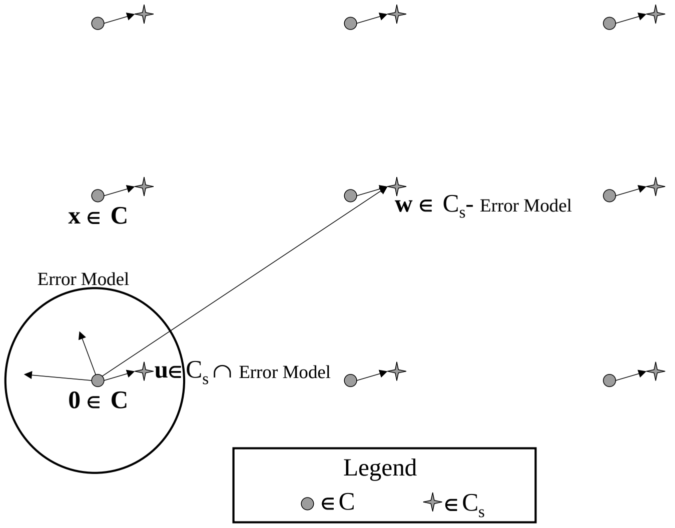

Fig. 3. The error model Vs code and cosets.

in the error model that has this syndrome. For simplicity, we will analyse this non probabilistic case. In Figure 3, the dots are codewords and the stars are strings in the coset $C_{\mathbf{s}}$. By definition, the coset $C_{\mathbf{s}}$ is obtained *via* a translation $\mathbf{x} \mapsto \mathbf{x} \oplus \mathbf{u}$ on the linear code where $H \odot \mathbf{u} = \mathbf{s}$. Error transformations (*i.e.* translations) on the codewords correspond to vectors which are represented by arrows in Figure 3. If we restrict the error transformation to the error model, only one such error transformation can map a codeword into a string in $C_{\mathbf{s}}$ because the strings in this coset are far away one to the other. This error transformation corresponds to the unique vector $\mathbf{u}$ with weight smaller than $d/2$ in the coset $C_{\mathbf{s}}$. So, the syndrome $\mathbf{s}$ uniquely determines this transformation, and that is sufficient for error-correction. The actual physical error mechanism might generate a different transformation for each initial string, and the syndrome does not tell us what is this actual error process, but we don't care about that.

Now, we analyse the quantum case. We recall that the standard Pauli matrices σ_1, σ_3 and σ_2 correspond respectively to bit flip error (the NOT operation), phase flip error (conditional phase shift) and both. For an error string $\mathbf{e} \in \{0,1\}^n$, and an error operator σ on single qbit (*e.g.* $\sigma = \sigma_1$ or $\sigma = \sigma_3$) we define $\sigma^{\mathbf{e}} = \otimes_{i=1}^{n} \sigma^{e[i]}$. In the quantum case, we do not have the distinction between *a posteriori* defined and actual error mechanism because the actual state could be a superposition of all possible string-states in the code. In exchange, we have that the most general transformation, a generalised collapse operation A, has to be linear. Every generalised collapse operation A on the state of the n photons can be expressed as a linear combination of basic error operations on n photons of the form $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z}$ where $\mathbf{e}_X, \mathbf{e}_Z \in \{0,1\}^n$. Of course, we must add a condition on the possible error transformation A. For perfect error-correction, a natural condition is that, for every basic error operation $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z}$ of the linear combination, the weights of $\mathbf{e}_X$ and $\mathbf{e}_Z$ are both below $\lfloor d/2 \rfloor$ where d is the minimal distance of the code. As we will discuss later this condition is more restrictive than necessary for perfect error-correction, but for simplicity we will adopt it. As previously discussed, a condition of this kind should be forced with high precision by the test, but we will not cover this part of the proof except for a brief discussion later in the paper. However, we would like to mention in parenthesis that the test, which is executed in the computational basis $|x[T]\rangle$ in T, is used to bound both the σ_1 errors in the bases $\{|x[E]\rangle\}$ and the σ_3 errors in the basis $\{|z[E]\rangle_H\}$ in E.

As in the classical case, if we know which basic error transformation $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z}$ needs to be undone, then it is easy to undo it. If the initial state is

$$\Psi = \sum_{\mathbf{k} \in \{0,1\}^m} \alpha_{\mathbf{k}} \Psi(\mathbf{0}, \mathbf{k}, \mathbf{0}) \in Q_{\mathbf{0},\mathbf{0}},$$

the final state is

$$A\Psi = \sum_{\mathbf{k}} \sum_{\mathbf{e}_X, \mathbf{e}_Z} \alpha_{\mathbf{k}} \beta_{\mathbf{e}_X, \mathbf{e}_Z} \sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z} \Psi(\mathbf{0}, \mathbf{k}, \mathbf{0})$$

where $\alpha_{\mathbf{k}}$ and $\beta_{\mathbf{e}_X, \mathbf{e}_Z}$ are possibly complex coefficients. Note that the states $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z} \Psi(\mathbf{0}, \mathbf{k}, \mathbf{0})$ belong to cosets of the code $Q_{\mathbf{0},\mathbf{0}}$. If we measure the pair of syndromes $(\mathbf{s}, \mathbf{t})$, we project on the components $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z} \Psi(\mathbf{0}, \mathbf{k}, \mathbf{0})$ of $A\Psi$ such that

$$\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z} \Psi(\mathbf{0}, \mathbf{k}, \mathbf{0}) \in Q_{\mathbf{s},\mathbf{t}}.$$

However, by definition of the coset $Q_{\mathbf{s},\mathbf{t}}$, if this property (which determines the non vanishing components) is true for one $\mathbf{k}$, it must be true for all $\mathbf{k} \in \{0,1\}^m$. So, the above property is independent of $\mathbf{k}$ and can be rewritten $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z} \Psi(\mathbf{0}, \mathbf{0}) = \Psi(\mathbf{e}_X, \mathbf{e}_Z) \in Q_{\mathbf{s},\mathbf{t}}$ which is the same as $H_1 \odot \mathbf{e}_X = \mathbf{s}$ and $H_2 \odot \mathbf{e}_Z = \mathbf{t}$. Furthermore, in our simple error model, we also have that the weight of both $\mathbf{e}_X$ and $\mathbf{e}_Z$ is smaller than d. These two properties are simply the conjunction of the corresponding classical condition for bit flip error correction and phase flip error correction separately. Because the string-states in $Q_{\mathbf{s},\mathbf{t}}$ are far away in the Hamming distance of both x and z, at most one error transformation $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z}$ has these two properties. So, after we have obtained the pair of syndromes $(\mathbf{s}, \mathbf{t})$, we know that we need to correct this particular error transformation. The above discussion can be summarized in the following lemma.

Lemma 2. *Given that the minimal distance of the linear codes defined by the parity check matrix H_1 and H_2 of a CSS code is d in both cases. Given that the possible collapse operations A are linear combinations of error transformations $\sigma_1^{\mathbf{e}_X} \sigma_3^{\mathbf{e}_Z}$ where the weights of $\mathbf{e}_X$ and $\mathbf{e}_Z$ are both smaller than $\lfloor d/2 \rfloor$. Bit flip (phase flip) error-correction can be done via the transformation $\sigma_1^{\mathbf{e}_X}$ ($\sigma_3^{\mathbf{e}_Z}$) in which the error string $\mathbf{e}_X$ ($\mathbf{e}_Z$) is the unique string of weight smaller than $\lfloor d/2 \rfloor$ that respects $H_1 \odot \mathbf{e}_X = \mathbf{s}$ ($H_2 \odot \mathbf{e}_Z = \mathbf{t}$).*

We will go into more details *via* an *example of a CSS code.* Let us consider the parity check matrix

$$G = \begin{array}{c} 1\,1\,1\,1\,1\,1\,1 \\ \hline 1\,0\,0\,1\,1\,1\,0 \\ 0\,1\,0\,0\,1\,1\,1 \\ 0\,0\,1\,1\,1\,0\,1 \end{array}$$

for a linear code C_2. Here the top matrix is $K = 1111111$, a single row, so the key is a single bit. The parity check matrix H_1 for the larger code C_1 consists of the three rows at the bottom. We have that C_1 is a $[7, 4, 3]$ code:

$$C_1 = \begin{array}{l} \{\underline{0000000}\ 0001011\ 0010110\ \underline{0011101} \\ \ \underline{0100111}\ 0101100\ 0110001\ \underline{0111010} \\ \ 1000101\ \underline{1001110}\ \underline{1010011}\ 1011000 \\ \ 1100010\ \underline{1101001}\ \underline{1110100}\ 1111111\} \end{array}$$

and $C_2 = \{x \in C_1 \,|\, K \cdot x = 0\}$, *i.e.*, C_2 contains all the codewords that we underlined in C_1. In this particular case, it turns out that $C_2 = C_1^\perp$ or equivalently $C_2^\perp = C_1$. So we take $H_2 = H_1$. Let $\mathbf{0} = 0000000$ and $\mathbf{1} = 1111111$. The CSS code $Q_{0,0}$ contains the two codewords

$$\begin{aligned} \Psi(\mathbf{0}, \mathbf{0}) = {} & \frac{1}{2\sqrt{2}} \\ & \times (|0000000\rangle + |0011101\rangle + |0100111\rangle + |0111010\rangle \\ & \quad + |1001110\rangle + |1010011\rangle + |1101001\rangle + |1110100\rangle) \end{aligned}$$

and

$$\begin{aligned} \Psi(\mathbf{1}, \mathbf{0}) = {} & \frac{1}{2\sqrt{2}} \\ & \times (|1111111\rangle + |1100010\rangle + |1011000\rangle + |1000101\rangle \\ & \quad + |0110001\rangle + |0101100\rangle + |0010110\rangle + |0001011\rangle) \cdot \end{aligned}$$

Consider the new basis states $|\mathbf{w}\rangle_H$ which are obtained from the old basis states $|\mathbf{w}\rangle$ by an Hadamar transform H on all n qbits. To illustrate formula (2), we note that it allows us to rewrite the string-states $\Psi(\mathbf{x}, \mathbf{z})$ in the new basis states $|\mathbf{w}\rangle_H$:

$$\begin{aligned} \Psi(\mathbf{1}, \mathbf{0}) = {} & \frac{1}{4} \\ & \times (|000000\rangle_H + |0011101\rangle_H + |0100111\rangle_H + |0111010\rangle_H \\ & + |1001110\rangle_H + |1010011\rangle_H + |1101001\rangle_H + |1110100\rangle_H \\ & - |0001011\rangle_H - |0010110\rangle_H - |0101100\rangle_H - |0110001\rangle_H \\ & - |1000101\rangle_H - |1011000\rangle_H - |1100010\rangle_H - |1111111\rangle_H) \cdot \end{aligned}$$

The error model describes the error transformations A which can possibly occur before error-correction. For error correction, first Bob finds out the syndromes $H_1 \cdot \mathbf{x} = \mathbf{s}$ and $H_2 \cdot \mathbf{z} = \mathbf{t}$. The final state after the measurement of the syndromes $\mathbf{s}$ and $\mathbf{t}$ necessarily belongs to $Q_{\mathbf{s},\mathbf{t}}$. Many transformations $e_{\mathbf{s},\mathbf{t}}$ can map a state initially in $Q_{0,0}$ into a state of $Q_{\mathbf{s},\mathbf{t}}$, but there should exist only one such a transformation modulo a global phase that is consistent with the error model. This must be a property of the error model whatever this model is. The key point is that $\mathbf{s}$ and $\mathbf{t}$ must be enough information to find out this unique transformation.

Let $\sigma_X^{\hat{e}_i}$, $X \in 1, 2, 3$, denote the application of σ_X at position i. In the $C_1 = [7, 4, 3]$ example, we assume that the error model implies that, for every given $(\mathbf{s}, \mathbf{t})$, the valid error transformations have the form $\sigma_1^{\hat{e}_i} \sigma_3^{\hat{e}_j}$, where i and j are not necessarily distinct. As mentioned before, linear combinations of these basic transformations are also allowed. Again, such an error-model and its properties needs to be forced by the test. We do not cover this part of the security proof. Now, consider the initial codeword $\Psi(0001011, 0000000) \in Q_{0,0}$. Alice announces the initial syndromes $\mathbf{s} = 000$ and $\mathbf{t} = 000$. Assume that the final syndromes obtained by Bob are $\mathbf{s} = 011$ and $\mathbf{t} = 101$. Because the minimal distance of both C_1 and $C_2^\perp$ is 3, it can be verified that the only valid transformation $e_{011,101}$ that can reach a state in $Q_{011,101}$ from a state in $Q_{000,000}$ corresponds to the transformation $\sigma_1^{(4)} \sigma_3^{(7)} = \mathsf{I} \otimes \mathsf{I} \otimes \mathsf{I} \otimes \sigma_1 \otimes \mathsf{I} \otimes \mathsf{I} \otimes \sigma_3$. Thus we know that the final codeword is $\Psi(0000011, 0000001) \in Q_{011,101}$, but Bob can undo the error without knowing this final codeword. Also, as we mentioned before, the syndrome $\mathbf{s}$ alone tell us at which position i to execute σ_1. Also, we see in this example that whether or not we undo σ_3, the phase flip error, will not affect the final key.

Now, we would like to discuss some connections between classical and quantum privacy amplification. For classical privacy amplification over classical error-correction, the minimal weight of $C_2^\perp - C_1^\perp$ must be large, not the minimal distance of $C_2^\perp$ [3]. Consider the toy example

$$G = \begin{array}{c} 1\,1\,1\,1\,1 \\ \hline 1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0 \end{array} \cdot$$

Here $K = [11111]$ and

$$C_2^\perp - C_1^\perp = \{11111, 01111, 10111, 00111\}$$

with minimal weight $d_Z = 3$. Eve can learn up to $d_Z - 1 = 2$ bits at given positions in a string v and she will learn nothing about $K \cdot v$, even if she has the parity bits from the two rows at the bottom. This suggests that in general the minimal weight of $C_2^\perp - C_1^\perp$ must be large for privacy. In general, Eve can obtain more complicated kinds of information and general proof for privacy amplification are difficult. New techniques were used in [3,6]. In particular, it was realised and proven that we obtain a large minimal weight for $C_2^\perp - C_1^\perp$ if we pick the matrix K at random over any parity check matrix H_1, thus allowing the use of an efficient linear code C_1 for error-correction. Similarly, for classical error-correction (followed by privacy implication), it is not the minimal distance of C_1 which must be large, but the minimal weight of $C_1 - C_2$. This is because errors in C_2 do not affect the final key.

It is interesting to note that the corresponding concept is also true for quantum error correction: a more detailed analysis [2] shows that in a CSS code it is sufficient that the minimal weight of $C_1 - C_2$ and $C_2^\perp - C_1^\perp$ are large.

We remark that we can obtain an efficiently decodable code C_1 and another code $C_2 \subset C_1$ with a large minimal weight for $C_1 - C_2$ (as required for bit flip error correction) together with a large minimal weight for $C_2^\perp - C_1^\perp$ [3]. However, we do not know how to have $C_2^\perp$ efficiently decodable at the same time. Nevertheless, Shor's and Preskill's reduction is still practical because the code $C_2^\perp$ for phase flip error-correction is only used to prove privacy in the BB84-CSS protocol; it is not needed in the BB84 protocol [2]. Another remark is only Bob does error correction in the BB84 protocol and thus also in the BB84-CSS protocol, but in practice there will be errors on both sides. This is why Shor and Preskill must assume that the Φ^+ source is perfect. Errors occur on both sides, but since the Φ^+ pairs are perfect, errors on Alice's side are equivalent to errors on Bob's side, and thus we can assume that there is no error on Alice's side.

Now, let us analyse in more details the security of the BB84 protocol in the context of imperfect apparatus. As for many other security proofs, Shor and Preskill's proof assumes that there is a known upper bound on the level of imperfection in the apparatus used. The version of their proof in [2] assumes that the apparatus are perfect, that is, this upper bound is in fact zero. However, whether the upper bound is zero or strictly positive, in both cases, this bounded error assumption is problematic. For example, let us assume that we obtain the system from some manufacturer. The manufacturer is perhaps not malicious, but he might be negligent. How do you know for sure that the bound is respected? This is the untrusted apparatus issue. Mayers's original proof [3] has the advantage to have addressed the untrusted measuring apparatus issue. Later, an approach to deal with an untrusted source in the context of untrusted measuring apparatus was described by Mayers and Yao [8].

Consider formula (3) in Shor's and Preskill paper: $F \equiv \langle (\Phi^+)^{\otimes m} | \rho' | (\Phi^+)^{\otimes m} \rangle \geq \mathrm{Tr}(\Pi\, \rho)$. In this formula, ρ is a state of n pairs of photons that are used to encode $m < n$ pairs of photons. The density operator ρ' is the state of these m pairs after quantum error-correction and decoding. The measurement operators Π correspond to a test on the number of phase and bit flip errors in ρ. The above inequality states that $\mathrm{Tr}(\Pi\, \rho) \geq 1 - \epsilon$, a constraint that is forced by the test on the number of phase and bit flip errors, is sufficient to obtain a corresponding level of fidelity $F \geq 1 - \epsilon$ between the decoded state ρ' and m perfect Bell states $(\Phi^+)^{\otimes m}$. However, to obtain this inequality (*via* quantum error correction) we needs a constraint on ρ, not on $\Pi \rho$. Therefore, my guess is that to work out the imperfect case we will need some upper bound on the imperfection level in Π so that the value of $\mathrm{Tr}(\Pi\, \rho)$ means something on ρ. The problem is that such an upper bound is not available if the manufacturer is negligent – it is hard to put an upper bound at this level.

How Mayers's proof addresses this issue? The complementary principle states that, if a measurement operator provides a lot of information about Alice's bits when Alice uses the conjugate bases (*i.e.* swaps the diagonal and rectilinear bases), the same measurement provides little information when Alice uses the original bases. Accordingly, Mayers's proof considers a variation on the protocol where Bob uses the conjugate bases for the untested bits so that, as required, he would obtain a lot of information had Alice used these conjugate bases. Eve and Bob together execute a refinement of this measurement that necessarily provides even more information. So, in accordance with the complementary principle, Eve and Bob together, and thus Eve alone, have little information when Alice use the original bases. This provides a bound on Eve's information, but on a key that exists only on Alice's side. (Bob uses the wrong bases.) Fortunately, Bob keeps his untested bits private, so this key and Eve's information about it are the same as in the original protocol, and that's sufficient in the proof. The untrusted apparatus issue is taken care of because the proof depends only on the fact that Bob obtains a lot of information when Alice uses the conjugate bases. This is forced by the test in the protocol, whatever is Bob's apparatus.

To understand Mayers's proof in more details, one essentially needs to understand the complementary principle and how the proof uses it. We already explained in the above discussion how the complementary principle is used. Here, we explain how to obtain it. There are many possible variations on the complementary principle. Mayers's proof includes the proof of two possible variations on this principle, one for an exact (*i.e.*, ideal) case and one for the inexact (*i.e.* realistic) case. These two variations fit well with the classical privacy amplification technique used in the BB84 protocol. In the remainder of this document, we will only focus on the exact case.

We recall that in the complementary principle we need to quantify how much information is provided by Bob's measurement operator when Alice uses the flipped bases. Bob must also use these flipped bases because otherwise he will not have a lot of information. However, here this is not the main point since we will assume that the test was successful in verifying that Bob's measurement provides a lot of information. The complementary principle says that, because this measurement provides a lot of information when Alice uses the flipped bases, then this same measurement provide little information when Alice uses the original bases. The fact that Bob's measurement provides a lot of information when Alice uses the flipped bases is expressed in Mayers's proof in the so-called small sphere property $\mathcal{S}$. The strong small sphere property is the exact case. As explained before, this property must be given in terms of a fictive preparation where Alice and Bob use flipped bases.

Some notations. The strings of flipped bases on Alice's and Bob's side are denoted $\tilde{a}$ and $\tilde{b}$, respectively. Since, these strings are actually string of bits, here we refer to a convention that the rectilinear basis and the diagonal basis are respectively associated with the bits 0 and 1.

So $a[i] = 1$ refers to the diagonal basis and $a[i] = 0$ refers to the rectilinear basis. For every $\alpha \in \{0,1\}^E$ and $\theta \in \{0,1\}^E$, we denote by $\Psi(\alpha, \theta)$ the state that encodes the string of bits α into the photons in E using the string of bases θ. We recall that by "in E" or "on E" we mean in or on the set of photons at positions i with $E[i] = 1$, but sometimes E directly refers to the positions i with $E[i] = 1$. For any α, we denote by $|\alpha\rangle = \Psi(\alpha, \tilde{b}[E])$ the state that encode the string α in the photons in E using the flipped bases $\tilde{b}[E] = \tilde{a}[E]$.

Definition 1. *Consider any state $\tilde{\phi}$ in the state space for the photons in E. We say that $\tilde{\phi}$ has the strong small sphere property with radius d'' if whenever $\alpha \in \{0,1\}^E$ does not lie strictly inside the sphere of radius d'' around $\mathbf{y}[E]$ (in the Hamming distance), we have that $\langle\tilde{\phi}|\alpha\rangle = 0$.*

Remark. In accordance with the basic intuition that is explained above, we want to show that Bob receives a lot of information when Alice uses the string of conjugate bases $\tilde{a}[E] = \tilde{b}[E]$ on E. The strong small sphere property (strong ssp) says that Bob has a lot of information because it puts an upper bound on $d_E(x, y)$, the number of errors in Bob's string y restricted at E. The strong ssp says that, given that Alice's initial state is encoded in the string of conjugate bases $\tilde{b}$, the outcome associated with $|\tilde{\phi}\rangle\langle\tilde{\phi}|$ ensure that the number of errors is strictly smaller than d'', being implicit here that $|\tilde{\phi}\rangle\langle\tilde{\phi}|$ is a measurement operator that returns the outcome $\mathbf{y}[E]$.

2 An example

The goal here is to illustrate the strong small sphere property. This example should not be considered as an illustration of the proof, only the strong small sphere property is illustrated. We consider a simple kind of attacks where Eve-Bob announced the string of bases $\tilde{b}[E] = ++\ldots++$ on E at the beginning, but Eve-Bob cheated and actually measured in the flipped string of bases $\tilde{b}^*[E] = \times\times+\ldots++$: the bases for the two first positions i with $\mathbf{E}[i] = 1$ have been flipped with respect to the bases $\tilde{b}[E]$. (Eve-Bob can obtain such a situation with a significant probability by flipping few bases at random.) Let us assume that the outcome on E is $\mathbf{y}[E] = 00\ldots0$. We will see that the state $\Psi(\mathbf{y}[E], \tilde{b}^*[E])$ has the strong small sphere property with radius 3. The associated "bra" operation is

$$\begin{aligned}&\langle\,\Psi(\mathbf{y}[E], \tilde{b}^*[E])\,| = \\ &\quad 1/2(\langle 000\ldots0| + \langle 010\ldots0| + \langle 100\ldots0| + \langle 110\ldots0|)\cdot\end{aligned}$$

There are only 4 strings $\alpha \in \{0,1\}^E$ on E such that

$$|\langle\,\Psi(\mathbf{y}[E], \tilde{b}^*[E])\,|\,\alpha\rangle| \neq 0.$$

These are the 4 strings that label the 4 components of $\langle\,\Psi(\mathbf{y}[E], \tilde{b}^*[E])\,|$. These four strings lie strictly inside a sphere of radius 3 around $\mathbf{y}[E]$. So, the state $\Psi(\mathbf{y}[E], \tilde{b}^*[E])$ has the strong small sphere property with radius 3. This concludes the example.

The strong small sphere property is too strong to be a property of the actual collapse operation executed by Eve-Bob on the photons in E. This property cannot be obtained, not even probabilistically. It corresponds to the ideal requirement that the test on E passes with probability exactly 1 given that this collapse operation occured. Nevertheless, this ideal situation is sufficient to explain the basic mechanism of the proof. The next lemma says that if a state $|\phi\rangle$ has the strong small sphere property then the associated collapse operation provides no information at all about the final key. This lemma combines together privacy amplification and the complementary principle in an intricated manner. We already explained the lemma in connection with the complementary principle. Privacy amplification enters into the picture because we directly consider the density matrix associated with the final key. We emphasis that the approach in which one first obtains a bound on some kind of information (such as the collision information) about Alice's raw key $x[E]$ and then separately use standard privacy amplification techniques [4] to obtain a much smaller bound on the final key didn't succeed thus far in quantum cryptography.

Lemma 3. *For every key $\mathbf{k} \in \{0,1\}^m$ and syndrome $\mathbf{s} \in \{0,1\}^r$, consider the density matrix*

$$\tilde{\rho}_{\mathbf{s},\mathbf{k}} \overset{\text{def}}{=} |C_{\mathbf{k},\mathbf{s}}|^{-1} \sum_{\alpha \in C_{\mathbf{k},\mathbf{s}}} |\Psi(\alpha, a[E])\rangle\langle\Psi(\alpha, a[E])|$$

where $C_{\mathbf{k},\mathbf{s}}$ is the set of string $\alpha \in \{0,1\}^D$ consistent with the key $\mathbf{k}$ and the syndrome $\mathbf{s}$, i.e., for which $F \bullet \mathbf{x}[E] = \mathbf{s}$ and $K \bullet \mathbf{x}[E] = \mathbf{k}$. Consider any state $\tilde{\phi}$ on the state space for the photons at positions i with $\mathbf{E}[i] = 1$. If $\tilde{\phi}$ has the strong small sphere property with radius $d'' \leq d_Z/2$, where d_Z was defined before as the minimal weight of $C_2^{\perp} - C_1^{\perp}$, then $\langle\tilde{\phi}\,|\,\tilde{\rho}_{\mathbf{k},\mathbf{s}}\,|\,\tilde{\phi}\rangle$ is independent of $\mathbf{k}$.

Note that an outcome that is equally likely to occur for every possible key $\mathbf{k}$ provides no information about the key. Therefore, lemma 2 is really an exact version of the complementary principle. We now prove this lemma.

Proof of lemma 3. We first do the case where $r = 0$ (no error-correction), $m = 1$ and the one row binary matrix K is $[11\ldots11]$. In this case, we have $d_Z = n_E$. Also, one can easily compute the density matrices $\tilde{\rho}_0$ and $\tilde{\rho}_1$ respectively associated with Alice's preparation for the photons in E when the key is $k = 0$ and $k = 1$. We recall that, for every $\alpha \in \{0,1\}^E$, we defined $|\alpha\rangle \overset{\text{def}}{=} \Psi(\alpha, \tilde{b}[E])$. One obtains that the matrix $[\Delta\tilde{\rho}]_{\{|\alpha\rangle\}}$ of $\Delta\tilde{\rho} \overset{\text{def}}{=} \tilde{\rho}_0 - \tilde{\rho}_1$ in Bob's basis $\{|\alpha\rangle\} \overset{\text{def}}{=} \{|\alpha\rangle\,|\,\alpha \in \{0,1\}^E\}$ is

$$[\Delta\tilde{\rho}]_{\{|\alpha\rangle\}} = 2^{1-n_E} \left(\begin{array}{c|c|c} \begin{matrix}0 & 1\\ 1 & 0\end{matrix} & & \\ \hline & \ddots & \\ \hline & & \begin{matrix}0 & 1\\ 1 & 0\end{matrix} \end{array}\right)$$

in which there are 0 everywhere except when indicated otherwise in the two by two blocks that are located on the diagonal. The indices $\alpha \in \{0,1\}^E$ for the rows in the matrix are ordered in such a way that any two indices $\alpha_1, \alpha_2 \in \{0,1\}^E$ which are at maximal Hamming distance n_E are always adjacent, and the same ordering is used for the indices $\alpha' \in \{0,1\}^E$ for the columns. The entries in the matrix are $\langle\alpha|\Delta\tilde{\rho}|\alpha'\rangle$, where $\alpha, \alpha' \in \{0,1\}^E$. We have that $\langle\alpha|\Delta\tilde{\rho}|\alpha'\rangle = 0$ unless $d(\alpha,\alpha') \geq d_Z = n_E$. The matrix $[\Delta\tilde{\rho}]_{\{|\alpha\rangle\}} = [\Delta\tilde{\rho}]^{(n_E)}_{\{|\alpha\rangle\}}$ can be obtained using the recurrence formula

$$\Delta\tilde{\rho}^{(n)} = (1/2)\Delta\tilde{\rho}^{(n-1)} \otimes (\tilde{\rho}^{(1)}_0 - \tilde{\rho}^{(1)}_1)$$

which can be obtained with some algebra using the formula

$$\tilde{\rho}^{(n)}_b = (1/2)[\tilde{\rho}^{(n-1)}_0 \otimes \tilde{\rho}^{(1)}_b + \tilde{\rho}^{(n-1)}_1 \otimes \tilde{\rho}^{(1)}_{\bar{b}}].$$

Now, we want to show that the probability of v is the same given both density matrices. So, we want to show that $\langle\tilde{\phi}\,|\,\Delta\tilde{\rho}\,|\tilde{\phi}\rangle = 0$. We have that

$$\langle\tilde{\phi}\,|\,\Delta\rho\,|\,\tilde{\phi}\rangle = \sum_{\alpha,\alpha'}\langle\tilde{\phi}\,|\,\alpha\rangle\langle\alpha\,|\,\Delta\tilde{\rho}\,|\,\alpha'\rangle\langle\alpha'\,|\,\tilde{\phi}\rangle\cdot$$

We show in two cases that every term in the sum is 0. Case 1: if $d(\alpha,\alpha') \geq d_Z = n_E$ then, because $\tilde{\phi}$ has the strong small sphere property with radius $d_Z/2$, either $\langle\tilde{\phi}\,|\,\alpha\rangle = 0$ or $\langle\alpha'\,|\,\tilde{\phi}\rangle = 0$. Case 2: if $d(\alpha,\alpha') < n_E$, then $\langle\alpha|\Delta\rho|\alpha'\rangle = 0$. This concludes the proof for the simple case where $m = 1$ and $r = 0$.

Now we do the proof for the general case where $m, r > 0$. The matrix $[\tilde{\rho}_{\mathbf{s},\mathbf{k}}]_{\alpha,\alpha'}$ is given by

$$(\hat{\rho}_{\mathbf{s},\mathbf{k}})_{\alpha,\alpha'} =$$
$$2^{-n_E}\begin{cases} 0 & \text{if } (\alpha\oplus\alpha') \notin C^{\perp}[G] \\ (-1)^{\lambda(\alpha\oplus\alpha')\bullet(\mathbf{s},\mathbf{k})} & \text{otherwise} \end{cases}$$

where

$$G = \begin{pmatrix} F \\ K \end{pmatrix},$$

$C^{\perp}[G]$ is the code that contains linear combinations of rows of G and λ is the coordinate function that when evaluated on any string $\alpha \in C^{\perp}[G]$ returns the string coordinate $\lambda(\alpha)$ such that $\lambda(\alpha)\bullet G = \alpha$. The computation is provided in Appendix B. By definition of d_Z if the weight of $(\alpha\oplus\alpha')$, which is the same as $d(\alpha,\alpha')$, is strictly smaller than d_Z, then $\lambda(\alpha\oplus\alpha')$ vanishes in its K-section. We obtain that, for (α,α') fixed, the sign of the entry $[\tilde{\rho}_{\mathbf{s},\mathbf{k}}]_{\alpha,\alpha'}$ depends only on s. Therefore, $d(\alpha,\alpha') < d_Z$ implies that $[\Delta\tilde{\rho}]_{\alpha,\alpha'} = [\tilde{\rho}_{\mathbf{s},\mathbf{k}}]_{\alpha,\alpha'} - [\tilde{\rho}_{\mathbf{s},\mathbf{k}'}]_{\alpha,\alpha'}$ vanishes. The remainder of the proof is identical the proof in the simple case, and this can be easily checked by the reader. This concludes the proof.

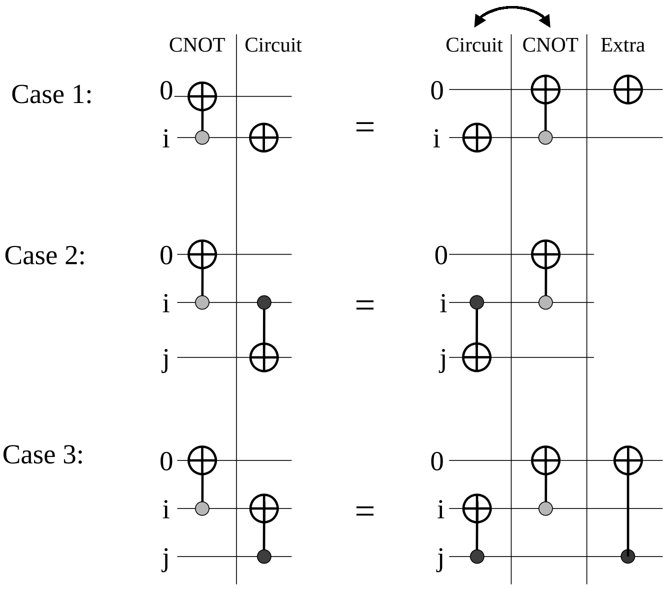

Fig. 4. Delaying a CNOT-measurement gate.

3 Conclusion

We have presented the essential idea of both Shor and Preskill's proof and Mayers's original proof. We have seen the interesting connections between these two proofs and classical privacy amplification. We have seen that Mayers's proof, despite the fact that it was proposed years before any other proof, still provides one of the strongest security result that is available for QKD.

Appendix A: Transmission gates

Here we want to show that the CNOT-measurement gate for a classical-quantum bit that is used in a transmission gate is sufficient: there is no need to CNOT-measure a classical-quantum bit before it is transmitted. Consider $n+1$ qbits at positions $0, 1, \ldots, n$ where positions 1 to n are for classical-quantum bits. First, we show that a CNOT-measurement gate with target qbit at position 0 and control qbit at position $i \in \{1, \ldots, n\}$ commute with any circuit of one or more CNOT or NOT gates in positions 1 to n (0 excluded) modulo extra CNOT gates also using position 0 as target qbit and possibly followed by a NOT gate at position 0. The proof can be done by induction on the number of gates, but here we will only prove the case where the circuit contains only one gate. The proof is by inspection of three cases illustrated in Figure 4:

– the circuit contains a NOT gate on the control qbit. In this case we simply have to add an extra NOT gate at position 0;
– the circuit contain a CNOT gate with control qbit at position i and target qbit at position $j > 0$, $j \neq i$. In this case, the two CNOT gates commute with no extra gate required;

– the circuit contain a CNOT gate with target qbit at position i and control qbit at position $j > 0$, $j \neq i$. In this case, one must add an extra CNOT gate with control qbit at position j and target qbit at position 0.

Therefore, we can delay the original CNOT-measurement gate until after the circuit if we follow it by these other CNOT and NOT gates. The new circuit execute the exact same unitary transformation. Since these CNOT and NOT gates are executed after the circuit they do not influence its classical computation. It may be important to execute them if their control qbit is transmitted, but this is taken care by the transmission gate.

Appendix B: The density matrices $\tilde{\rho}$

Consider a linear code $C[G] \subseteq \{0,1\}^n$ of dimension q and a coset $C[G,x]$ of this code (G is the parity check matrix and x is the syndrome). Here, we analyse the general situation where a string $\tilde{g}$ uniformly chosen at random in the coset $C[G,x]$ is sent from Alice to Bob using a fixed string of bases $a \in \{+,\times\}^n$. We want to find the matrix representation of the density operator

$$\tilde{\rho}_x = 2^{-q} \sum_{\tilde{g} \in C[G,x]} \widetilde{\Psi}(\tilde{g},a)\,\widetilde{\Psi}(\tilde{g},a)^\dagger$$

in the basis $\{\widetilde{\Psi}(\alpha,b) | \alpha \in \{0,1\}^n\}$ where $b = \bar{a}$. To apply this result to this paper, one must use

$$G = \begin{pmatrix} F \\ K \end{pmatrix},$$

$x = (s,k)$ and $q = n - r - m$ but the computation for the general case is the same. A key ingredient is that if a string g belongs to a code $C = C[G]$ for which G is the parity check matrix then we have $g = \lambda \bullet G^\perp$ where $\lambda \in \{0,1\}^{\dim C}$ and $G^\perp$ is a parity check matrix for the dual code. We will apply this principle twice, once with the code and once with its dual. We have

$$\tilde{\rho}_x = \frac{1}{|C|} \sum_{g \in C[G,x]} |w\rangle\langle w| \cdot$$

We will use the fact that in the conjugate basis we have

$$|w\rangle = 2^{-n} \sum_{t \in \{0,1\}^n} (-1)^{g \bullet t} |t\rangle \cdot$$

We obtain

$$\tilde{\rho}_x = \frac{2^{-n}}{|C|} \sum_{t,t',g \in C} (-1)^{g \bullet (t \oplus t')} |t\rangle\langle t'| \cdot$$

Let g_0 be any string in the coset $C[G,x]$. We will use the fact that the sum over $g \in C[G,x]$ can be replaced by a sum over $\gamma \in \{0,1\}^{\dim C}$ with the change of variable $g \mapsto (\gamma \bullet G^\perp) \oplus g_0$. We get

$$\tilde{\rho}_x = \frac{2^{-n}}{|C|} \sum_{t,t',\gamma \in \{0,1\}^{\dim C}} (-1)^{(g_0 \oplus \gamma \bullet G^\perp) \bullet (t \oplus t')} |t\rangle\langle t'| \cdot$$

After simple algebra, we get

$$\tilde{\rho}_x = \frac{2^{-n}}{|C|} \sum_{t,t'} (-1)^{g_0 \bullet (t \oplus t')} \underbrace{\sum_{\gamma \in \{0,1\}^{\dim C}} (-1)^{\gamma \bullet G^\perp \bullet (t \oplus t')}}_{k(t,t')} |t\rangle\langle t'| \cdot$$

Now, consider the coefficient $k(t,t')$. This coefficient vanishes if $G^\perp \bullet (t \oplus t') \neq 0$, that is, if $(t \oplus t') \notin C^\perp$. If $(t \oplus t') \in C^\perp$, we have $k(t,t') = |C|$. We obtain

$$\tilde{\rho}_x = 2^{-n} \sum_{t,t' | (t \oplus t') \in C^\perp} (-1)^{g_0 \bullet (t \oplus t')} |t\rangle\langle t'|$$

where we used $g_0 \bullet (t \oplus t') = (t \oplus t') \bullet g_0$. Now, we will use the fact that $(t \oplus t')$ is a string in $C^\perp$. We obtain that $t \oplus t' = \lambda(t \oplus t') \bullet G$ where $\lambda(t \oplus t')$ is the unique string with this property. The exponent $(t \oplus t') \bullet g_0$ becomes $\lambda(t \oplus t') \bullet G \bullet g_0 = \lambda(t \oplus t') \bullet x$, by definition of g_0. We obtain

$$\tilde{\rho}_x = 2^{-n} \sum_{t,t' | (t \oplus t') \in C^\perp} (-1)^{\lambda(t \oplus t') \bullet x} |t\rangle\langle t'|$$

or equivalently

$$\langle t | \tilde{\rho}_x | t' \rangle = 2^{-n} \begin{cases} (-1)^{\lambda(t \oplus t') \bullet x} & \text{if } (t \oplus t') \in C^\perp \\ 0 & \text{otherwise} \end{cases}.$$

References

1. C.H. Bennett, G. Brassard, Quantum cryptography: Public key distribution and coin tossing, *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, Bangalore, India, December 1984, pp. 175–179.
2. P.W. Shor, J. Preskill, Phys. Rev. Lett. **85**, 441 (2000).
3. D. Mayers, Quantum key distribution and string oblivious transfer in noisy channels, *Advances in Cryptology — Proceedings of Crypto'96* (Springer-Verlag, Berlin, 1996), pp. 343 – 357. A more recent version to be published in JACM, may 2001.
4. C.H Bennett, G. Brassard, J.-M Robert, SIAM J. Comp. **17**, 210 (1988).
5. D. Gottesman, J. Preskill, Phys. Rev. A **63** (2001); also in `quant-ph/0008046`.
6. A. Yao, Security of Quantum Protocols Against Coherent Measurements, *Proceedings of the 26th Symposium on the Theory of Computing*, June 1995, pp. 67 – 75.
7. H.-K. Lo, H.F. Chau, Science **283**, 2050 (1999); also in Los Alamos preprint archive `quant-ph/9803006`, March 1998.
8. D. Mayers, A. Yao, Quantum Cryptography with Imperfect Apparatus, *Proceedings of the 39th IEEE Conference on Foundations of Computer Science*, 1998.