10 Security from physical assumptions

Week 9, Lecture 9.2,Lecture 1: The noisy storage model

As we saw in the previous chapter, security for two-party cryptography is difficult to achieve. Even given the ability to use quantum communication, we still cannot hope to achieve the security conditions of general two-party cryptography, including $1-2$-oblivious transfer and bit commitment. Yet, two-party cryptography includes many interesting challenges we would like to solve on a daily basis! Due to the interest in solving this challenge, we may be willing to bend our principles somewhat and limit ourselves to obtain security guarantees under some assumption about the adversary. The idea is that we'd show security only for "reasonable" adversaries, where "reasonable" is still a very large class of adversaries that we expect might arise in real life.

Classically, the most commonly used kind of assumption on the adversary is a *computational assumption*. This assumption is two-fold: first, we assume that a specific problem, such as factoring a large integer, requires a large amount of computational resources to be solved. Second, we assume that the adversary has a relatively limited amount of computational resources available - namely, an amount that is insufficient to solve the difficult problem within a practically interesting time frame. An example of a computational assumption is the pseudorandom generator assumption which we used to construct a secure bit commitment protocol in Section **??**.

An important limitation of computational assumptions is that they tend to not stand the test of time. For example, once we build a quantum computer, any assumption based on the hardness of factoring will become vacuous, because the Shor algorithm provides an efficient way to factor large numbers on a quantum computer. Moreover, security can often be broken retroactively: if we build a quantum computer tomorrow, most two-party protocols that have been executed to date can lose their security, as long as the adversary has stored a copy of the protocol transcript in their classical memory, such as a flash drive. Clearly, for high security applications this is quite undesirable.

In this chapter we present an alternative path to base security in challenging settings. The main idea is to make physical, rather than computational, assumptions. Importantly, we would like that security only requires the assumption of interest to be valid during the execution of the protocol. If someone builds better equipment tomorrow, good for them, but we don't want that to compromise our cryptographic interactions from today! This guarantee allows us to use physical assumptions that are motivated by technological challenges that may be overcome in the future, but which may be reasonable to assume today given mankind's present technological abilities.

How is it possible to obtain long-lasting security under an assumption that is valid today, but may become invalid tomorrow? Intuitively, the idea is that even temporary physical assumptions can lead to a permanent lack of information for the adversary that prevents them from ever breaking the protocol in the future. It is interesting to note that technologically motivated physical assumptions can also enable key exchange or two-party cryptography using only classical communication (see the chapter notes). Indeed, we already saw an example of such an assumption in Section **??**. There, Alice and Bob were able to generate secure key using only classical communication, as long as the eavesdropper is limited in their ability to listen in to their communication channel.

When looking for technologically motivated assumptions, it is useful to keep in mind the perpetual conflict we face when designing cryptographic protocols: on the one hand, the protocol should be secure, that is, whatever assumptions we impose on the adversary should be sufficient to protect the honest parties. On the other hand, however, we of course want the protocol to be correct. That is, the honest parties should be able to execute it correctly. When considering good technological assumptions to make, this conflict translates into a desire to design a protocol that is technologically *easy* to execute for the honest parties, but at the same time technologically *infeasible* to break for the adversary. The gap between the resources needed to execute the protocol and the resources needed to break the protocol should be as large as possible; if it is too small then we may need to look for a better security assumption.

10.1 The noisy storage model

At present, our abilities to store large amounts of quantum information without errors and for a long time are extremely limited. In quantum systems with a quantum memory capable of accessing quantum communication, i.e. quantum memory systems with an optical interface, the state of the art at the time of writing is that storage times of the order of seconds have been observed for a small number of qubits. Moreover, the a transfer of qubits into such a memory is typically itself already lossy.

These technological limitations motivate the assumption that the abilities of the adversary to store quantum information are limited. The *noisy storage model* assumes that during finite waiting times Δt introduced in the protocol, the adversary is limited in their ability to store quantum information (see Figure 10.1). Specifically, to keep information during such waiting times, the adversary is limited to using a quantum memory whose action on the quantum data is modeled by some quantum channel $\mathcal{F}$. The fact that $\mathcal{F}$ is not necessarily the identity represents the fact that there may be losses in the memory. Other than this limitation the adversary remains all powerful: they may perform arbitrary quantum operations, including arbitrary encoding and decoding procedures before and after the waiting time, and store an unlimited amount of classical information. In particular, before and after the waiting time, the adversary is allowed to have an arbitrary faithful quantum memory. Since proofs of security in the noisy storage model only require the adversary's memory to be limited during the waiting time Δt, even if tomorrow we can build better quantum memories then security can nevertheless not be broken retroactively.

A simple example of a channel $\mathcal{F}$ we might use to model the adversary's memory corresponds to a quantum memory that is error-free, but limited in size to only q qubits for some integer q. That is, $\mathcal{F} = \mathbb{I}^{\otimes q}$. This special case is also known as the *bounded quantum storage* model. In general, one typically assumes $\mathcal{F}$ to be of the form $\mathcal{F} = \mathcal{N}^{\otimes q}$, that is, the quantum memory is both bounded in size as well as noisy. Here, $\mathcal{N}$ may be a noisy one qubit channel, such as depolarizing noise. Taking $\mathcal{F}$ to be of this form is appealing to analyze since one understands general properties of such channels to store quantum information from quantum information theory.

10.2 1-2 Oblivious Transfer in the noisy storage model

Week 9, Lecture 9.3,Lecture 1: A protocol for oblivious transfer

From the previous chapter we know that any two-party protocol can be constructed by combining multiple copies of a 1-2 oblivious transfer protocol, making this primitive a fundamental building block

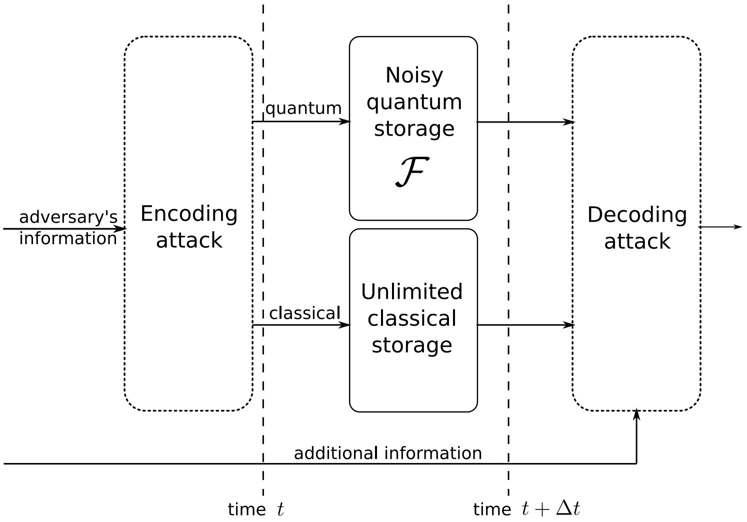

Fig. 10.1 Noisy storage model: During waiting times Δt introduced in the protocol, the adversary is required to store any quantum information using a quantum memory modeled by the channel $\mathcal{F}$. Other than this requirement the adversary may be arbitrary, including having access to a perfect large scale quantum computer to encode and decode the quantum information before and after using the memory $\mathcal{F}$.

for two-party cryptography. We have also seen that despite the attempt to construct OT protocols that make use of quantum communication, no unconditionally secure protocols can exist. Nevertheless we made a valiant attempt, Protocol **??** from Section **??**, which we unfortunately observed could easily be broken by a malicious Bob by using, for example... a large quantum memory. We thus know that our earlier protocol cannot be secure without some form of assumption on the power of the adversary. Could the protocol be secure against a memory-bounded malicious Bob? Let's give it a shot.

Recall that in 1-2 OT, Alice has two inputs $s_0, s_1 \in \{0,1\}^\ell$ and Bob has a single input $y \in \{0,1\}$. At the end of the protocol, Alice should return no output and Bob should obtain the string s_y. Here we recap essentially the same 1-2 OT protocol as in the previous chapter, except for two small modifications. First we make explicit a small waiting time Δt during which the noisy storage assumption is applied. Second, for reasons that will soon become clear we introduce a secure strong extractor $\mathrm{Ext} : \{0,1\}^{n/2} \times \{0,1\}^t \to \{0,1\}^\ell$ whose exact parameters we discuss later. First let's see the protocol.

Protocol 1 *Protocol for 1-2 OT in the noisy storage model. Alice has inputs $s_0, s_1 \in \{0,1\}^\ell$ and Bob has input $y \in \{0,1\}$.*

1 *Alice chooses a uniformly random string $x = x_1, \ldots, x_n \in \{0,1\}^n$ and $\theta = \theta_1, \ldots, \theta_n \in \{0,1\}^n$. She prepares BB'84 states $|x_j\rangle_{\theta_j}$ for $j = 1, \ldots, n$ and sends them to Bob.*
2 *If $y = 0$ then Bob measures all of the qubits in the standard basis. If $y = 1$, he measures in the Hadamard basis. He records the resulting outcome string $\tilde{x} = \tilde{x}_1, \ldots, \tilde{x}_n$.*
3 *Both parties wait time Δt. (Storage assumption is applied!)*
4 *Alice sends to Bob the string $\theta_1, \ldots, \theta_n$. Bob computes the set of indices $I = \{j \mid \theta_j = y\}$ where he measured in the same basis than Alice.*
5 *Alice chooses two independent random seeds $r_0, r_1 \in \{0,1\}^t$. She computes $k_0 = \mathrm{Ext}(x_+, r_0)$ and*

$k_1 = \mathrm{Ext}(x_\times, r_1)$, where x_+ is the substring of x which Alice encoded in the standard basis, and $x_\times$ is the substring where she used the Hadamard basis. Alice sends r_0 and r_1 to Bob.

6 *Alice sends to Bob $m_0 = s_0 \oplus k_0$ and $m_1 = s_1 \oplus k_1$, where $\oplus$ denotes the bit wise xor.*

7 *Bob computes $k = \mathrm{Ext}(x_I, r_y)$ and $s_y = k_y \oplus r_y$.*

We see that the honest parties need *no* quantum memory to execute this protocol. While we have for ease of explanation described the protocol in a way that may suggest that Alice first prepares all n qubits, and only then sends all of them to Bob, Alice and Bob can also execute the protocol with Alice preparing and transmitting only one qubit at a time, which Bob immediately measures upon receipt. This makes the comparison of the resources needed to execute the protocol (no quantum memory at all), to the resources needed to break the protocol (a large amount of quantum memory, as we will see) especially appealing.

Why does this protocol work? Let us first double check that the protocol is correct, that is, Bob actually obtains s_y in accordance with his choice bit $y \in \{0, 1\}$. Note that if there is no noise, then whenever $\theta_j = y$, we have $x_j = \tilde{x}_j$. That is, whenever Alice had encoded in the basis in which Bob measures, then Bob learns the corresponding element of Alice's bit string. This means that if Alice applies Ext to hash down the elements of the strings corresponding to the standard and Hadamard basis respectively, then Bob knows one of them perfectly. Since Alice sends him r_0 and r_1, he learns the correct k_y. In the protocol the string k_y is used as a key that encrypts s_y using one-time pad encryption, and so Bob can recover Alice's string s_y as well.

Quiz 10.2.1 *In the protocol for oblivious transfer, honest Alice and Bob do not need any quantum memory. Does that mean that if Alice possessed an unbounded and noise-free quantum memory, then the protocol would no longer be secure against dishonest Alice?*

a) *Yes, the protocol would not be secure against cheating Alice in this case, that is she could then easily learn Bob's bit y.*

b) *No, even with such a quantum memory Alice would not be able to learn Bob's bit y because nowhere in the protocol is there any communication from Bob to Alice taking place.*

10.3 Security from quantum uncertainty

Week 9, Lecture 9.4,Lecture 1: Security from quantum uncertainty

Is Protocol 1 secure? Let us first check security against dishonest Alice. This is virtually identical to the argument from Chapter **??**. Recall that we only need to show that Alice cannot learn Bob's choice bit y. If you stare at the protocol description, it is clear that this is definitely the case: Bob never sends any information at all to Alice, from which she could learn anything about y. This idea can be formalized into a complete security proof (against dishonest Alice), which we omit.

The main difficulty is to show security against dishonest Bob. Recall from the definition that we need to show that, while Bob might learn one of Alice's two strings s_0 and s_1, there must always exist some $\bar{y}$ such that Bob learns (almost) nothing about $s_{\bar{y}}$.

As you can see, at step 5 of the protocol we have used an old "trick" which we already encountered in the analysis of quantum key distribution, namely the use of privacy amplification to reduce the amount of information that a potential adversary has about a certain string. Based on what we know about privacy amplification and extractors, to achieve our goal it will be sufficient to somehow ensure that the conditional min-entropy of either X_+ or $X_\times$, conditioned on all information available to Bob, is high. As long as we can establish a good lower bound on this, by selecting the parameters of the extractor appropriately we can make sure that from Bob's point of view the string k_0 or k_1 is ε-close to uniformly distributed, making him unable to recover the corresponding s_0 or s_1.

To summarize, our goal is now to show that there exists a $\bar{Y}$ such that the conditional min-entropy of X_+ for $\bar{Y} = 0$ or $X_\times$ for $\bar{Y} = 1$, conditioned on Bob's information, is high. Observe how here we started using a capital letter for $\bar{Y}$. You know that we use capital letters for random variables. We did this because, if you think about it, for a given malicious Bob we may not be able to identify a single string X_+ or $X_\times$ that is unknown to Bob. This is simply because Bob, being quantum, could use a quantum coin flip to govern his behavior in the protocol, and end in a superposition of a Bob who knows a little bit about X_+, with a Bob who knows a little bit about $X_\times$. Therefore, it is inevitable that the answer to the question "which string is unknown to Bob" not only depends on Bob, but may in fact be a "quantum" answer. We warned you that proofs of security for quantum protocols can be subtle! Let's see how to make it work.

10.3.1 Security in the bounded quantum storage model

For security we will focus on the special case of the bounded quantum storage model, where $\mathcal{F} = \mathcal{N}^{\otimes q}$ with $\mathcal{N} = \mathbb{I}$. The argument in this case is simpler; we will sketch how it can be generalized in the following subsection.

In order to examine the conditional min-entropy of the strings X_+ and $X_\times$, let us first consider if we can say anything at all about Bob's min-entropy about *the entire* string x. Clearly, if Bob could store all of Alice's qubits, then he could just measure the entire string in the correct basis after having learned it at step 4 and obtain the entire string x.

This means that security against Bob can never be achieved if the number q of qubits that Bob can store is $q \geq n$. Let us thus assume that $q < n$. In practice this means that if we assume that the adversary can store at most q qubits – for example we might take q to be many times the size of the largest quantum memory known to date – we will choose the parameter n in the protocol to be large enough to achieve security. Note that since we allow Bob to have an arbitrary quantum memory and computer *before* the waiting time, he can first store all of Alice's qubits and perform an arbitrary quantum operation on all of them. For example, he might measure some of them, resulting in some classical information K. However, he can then keep only q qubits in his quantum memory, which we denote by Q.

Since we are interested in Bob's min-entropy about the entire string X, we want to show a lower bound on the quantity

$$H_{\min}(X_+X_\times|\Theta, K, Q) ,$$

where we have written $X_+X_\times$ for the string X to remind ourselves that we are ultimately interested in the two portions corresponding to the two different bases. To make his guess, Bob can use the classical information K, his quantum memory Q, and the basis information Θ that Alice sends to him after the waiting time.

When we studied quantum key distribution we noticed that is it often much easier to show security against an adversary who is entirely classical. Here, the adversary, malicious Bob, has information that is partially classical (Θ and K) and partially quantum (Q). How can we get rid of Q? The intuition of course is that Q, being made of only q qubits, should contribute "only q" to Bob's information. How do we make this formal? If you remember our work in previous chapters you might notice that we already faced a similar issue a couple times, and we handled it using the powerful *chain rule* for the conditional min-entropy. To recall the chain rule, see Box **??**. In our context the chain rule gives us that

$$\begin{aligned} H_{\min}(X_+X_\times|\Theta,K,Q) &\geq H_{\min}(X_+X_\times|\Theta,K) - \log|Q| \;, \\ &= H_{\min}(X_+X_\times|\Theta,K) - q \;. \end{aligned}$$

What this shows is that Bob's limited quantum memory can indeed only contribute an additive q bits to his conditional entropy. Now, it remains to bound the information aquired by a Bob who has only Θ and K. How could we possibly analyze this?

Once more, let us think back to the ideas that we learned while studying quantum key distribution. We know that the conditional min-entropy has an interpretation as a guessing probability, and that guessing probabilities can be bounded by studying an associated guessing game. How did we reduce the analysis of the BB'84 QKD protocol to a guessing game? By purifying the protocol and introducing an equivalent formulation that indeed looked like a game. So let's do the same here.

Consider a purified version of the 1-2 OT protocol where at the first step, instead of Alice sending BB'84 states to Bob, Bob prepares n EPR pairs and sends the first qubit of each pair to Alice. As for QKD, let's even give more power to Bob and let him prepare any state that he wants, as long as there are n qubits that are sent to Alice. Having received the qubits from Bob, Alice chooses one of two random bases to measure each qubit and announces the basis choice to Bob. Bob, given his side information (any qubits that he kept to himself, as well as the strings Θ and K), has to guess Alice's string of outcomes.

This is precisely the bipartite guessing game from Chapter **??**! The only difference is that in Chapter **??** we called Bob "Eve," and only looked at the version where Bob/Eve sends a single qubit to Alice, not n as here. Also, in Chapter **??** we didn't explicitly write out the classical side information K—but because we allow Bob/Eve to prepare any bipartite quantum state he wants, he can as a special case keep an arbitrary string K of information about the state he prepared.

Applying the bound shown in Chapter **??**,[1] we obtain the familiar

$$H_{\min}(X_+X_\times|\Theta,K) = n\left(-\log\left(\frac{1}{2}+\frac{1}{2\sqrt{2}}\right)\right) \approx 0.22n \;.$$

Of course, what we really need is to make a statement about the different parts X_+ and $X_\times$. That is, we would like to show that there exists a $\bar{Y} \in \{+,\times\}$ such that Bob's entropy about $X_{\bar{Y}}$ is high. This is some form of *min-entropy splitting*: if the uncertainty about a string is high, there must be some half of it that is unknown. And indeed, this is true (but a bit harder to prove): there exists some register $\bar{Y}$ such that

$$H_{\min}(X_{\bar{Y}}|\Theta,K,\bar{Y}) \;\geq\; \frac{H_{\min}(X_+X_\times|\Theta,K)}{2} - 1 \;.$$

Note that here, as we expected, $\bar{Y}$ is a classical random variable that can be correlated with the side information held by Bob. It is also noteworthy that min-entropy splitting only works if K really is classical, which is why we first have to get rid of Q. Putting all the steps together we obtain

$$\begin{aligned} H_{\min}(X_{\bar{Y}}|\Theta,K,\bar{Y}Q) &\geq H_{\min}(X_{\bar{Y}}|\Theta,K,\bar{Y}) - q \\ &\geq \frac{H_{\min}(X_+X_\times|\Theta,K)}{2} - 1 - q \;. \end{aligned}$$

To conclude, we can employ the properties of randomness extraction to claim that Bob is ϵ-close to being

[1] In fact we need an n-qubit version of the bound from Section **??**. This can be done using the same ideas as those introduced to study the tripartite guessing game, and it leads to the same bound $p_{\text{succ}}^{\text{n rounds}} \leq \left(\frac{1}{2}+\frac{1}{2\sqrt{2}}\right)^n$.

ignorant about $X_{\bar{Y}}$ whenever

$$\begin{aligned}\ell &< H_{\min}(X_{\bar{Y}}|\Theta, K, \bar{Y}, Q) - O(\log 1/\varepsilon) - 1 \\ &\approx 0.11n - q - O(\log 1/\varepsilon) - 2\ .\end{aligned}$$

This means that whenever $q \lesssim 0.11n$, we can have security for some $\ell > 0$! Or, reading it the other way around, assuming a maximum q for the adversary tells us that we need to send roughly $n \approx q/0.11$ qubits in order to achieve security.

Quiz 10.3.1 *In the security proof against cheating Bob we encountered again the uncertainty game discussed in Chapter* **??**. *However, here we want to evaluate $P_{guess}(X|\Theta K)$ rather than just $P_{guess}(X|\Theta)$. That is, Bob can have some classical coin which determines what state he sends to Alice. Consider two states ρ_1 and ρ_2, such that in a deterministic game the guessing probability of Bob corresponding to each of those states is $P_{guess}(X|\Theta)_{\rho_1} = P_{guess}(X|\Theta_A)_{\rho_2} = p$. Depending on the outcome of the random coin K, Bob sends to Alice the first or the second state. Would the resulting guessing probability $P_{guess}(X|\Theta K)$ be smaller than, equal to, or bigger than p?*

a) $\longrightarrow P_{\boldsymbol{guess}}(X|\Theta K) = p$
b) $P_{guess}(X|\Theta K) > p$
c) $P_{guess}(X|\Theta K) < p$

10.3.2 General channels

Using a more sophisticated analysis it is possible to show that security can be achieved as long as $q \leq n - O(\log^2 n)$, which is essentially optimal. We thus see that security becomes possible by sending just a few more qubits than Bob can store. This result was obtained following a series of advanced ideas in quantum information theory that were motivated by the noisy storage model. While outside the scope of our book, we briefly explain the notion of *capacity* of a quantum channel that underpins these ideas through the simple example of the classical capacity of a perfect one-qubit channel.

Earlier we directly assumed that $\mathcal{F} = \mathcal{N}^{\otimes q}$, where $\mathcal{N} = \mathbb{I}$ is a perfect – noise free – one-qubit channel. Looking at Figure 10.1, it is intuitive that the problem of proving security in the noisy storage model is directly related to a problem at the heart of quantum information theory, namely the study of the *capacity* of the channel $\mathcal{F}$. Roughly speaking, the capacity of a channel quantifies how much information we may send through it (with some limited amount of error), provided that we can use the best possible error-correcting code imaginable. That is, the best possible encoding procedure to protect the information from the noise in the channel, and the best possible decoding procedure to recover it.

For sending some classical information $i \in \{0,1\}^m$ through the channel $\mathcal{F}$, an encoding scheme consists of a mapping $i \to \rho_i$ of the classical string i to some quantum state ρ_i. A decoding scheme corresponds to a measurement on the channel output $\mathcal{F}(\rho_i)$ with outcome $\hat{i}$ that correspond to a guess for i. The probability that a string i is correctly decoded may thus be written as $q_i = \mathrm{tr}\left(M_i \mathcal{F}(\rho_i)\right)$, where $\{M_i\}_i$ are the POVM operators corresponding to the decoding measurement. The quantity of interest when studying the classical capacity of a channel $\mathcal{F}$ is the average probability of recovering an unknown string i. This can be expressed as

$$P_{\mathrm{succ}}\left(\mathcal{F}, m\right) = \max \frac{1}{2^m} \sum_{i \in \{0,1\}^m} \mathrm{tr}\left(M_i \mathcal{F}(\rho_i)\right)\ ,$$

where the maximization is taken over all possible encodings and all possible decodings. In general, even for sending classical information, the capacity of most quantum channels is difficult to understand!

Luckily, for our simple example of $\mathcal{F} = \mathbb{I}^{\otimes q}$ it is easy to examine “how much“ information we can convey. We call $R = m/q$ the *rate* of sending classical information though $\mathcal{F}$. The *classical capacity* C is defined such that for $R \leq C$, there exists an error-correcting code to send information with $P_{\rm succ}(\mathcal{F}, qR) \to 1$ and for $R > C$ we have $P_{\rm succ}(\mathcal{F}, qR) \to 0$ as $q \to \infty$. That is, the capacity forms a sharp threshold for sending information! For our example with $m = qR$ we have

$$\begin{aligned} P_{\rm succ}\left(\mathcal{F}, qR\right) &= \max \frac{1}{2^{qR}} \sum_{i \in \{0,1\}^{qR}} \operatorname{tr}\left(M_i \rho_i\right) \ , \\ &\leq \frac{1}{2^{qR}} \sum_{i \in \{0,1\}^{qR}} \operatorname{tr}\left(M_i\right) \ , \\ &= \frac{1}{2^{qR}} \operatorname{tr}\left(\mathbb{I}\right) \ , \\ &= 2^{-qR} 2^{q} = 2^{-q(R-1)} \ , \end{aligned}$$

where the second line follows from $0 \leq \rho_i \leq \mathbb{I}$, and the third from $\sum_i M_i = \mathbb{I}$. Since we are considering a space of q qubits, we have $\operatorname{tr}(\mathbb{I}) = 2^q$. We thus see that in our case $C = 1$ forms a sharp threshold, since for $R > C$ we get that $P_{\rm succ} \to_{q\to\infty} 0$. That is, we can transmit no more than one classical bit per qubit of channel use. Moreover, clearly for $R \leq C$ we may indeed send one classical bit per qubit, e.g. by encoding it into the standard basis.

Quiz 10.3.2 *Consider the setting described in this section. How reliably can we reconstruct the input state at the output of the channel $\mathcal{F}$ in the regime where $R > 1$?*

a) There is no difference between $R > 1$ and $R \leq 1$, that is suitable encodings can allow us to reliably decode the input state at the output.

b) In the regime where $R > 1$, the set of encoding procedures that allow for reliable transmission of all the N qubits becomes restricted to encodings that satisfy certain specific conditions.

c) $\longrightarrow$ ***In this regime there exist no reliable encodings, that is for all encoding procedures, the trace distance between the input and the output state will be exponentially (in N, the number of input qubits that we want to store) close to 1.***

10.3.3 Working with noisy devices

Our protocol above assumes that the honest Alice and Bob can execute the protocol without errors. What if even the honest users are subject to errors? This is important for practical applications. Adapting the protocol to this situation can be done using the techniques we already learned in our study of quantum key distribution. Indeed, similar to the case of QKD when a small amount of errors occur on the channel, information reconciliation can be performed in order to ensure that Bob is able to correct for errors if he only receives noisy versions $\tilde{X}_+$ or $\tilde{X}_\times$ of Alice's string. Alice can then employ a classical error correcting code, and she will need to send Bob the error syndromes $e_+ = s_H(x_+)$ and $e_\times = s_H(x_\times)$ computed according to the error correcting code used, with parity check matrix H. Bob then proceeds to use the relevant syndrome to correct his noisy string $\tilde{x}_+$ to x_+ for $y = 0$, and $\tilde{x}_\times$ to $x_\times$ for $y = 1$. Correctness of the protocol then follows by the properties of the classical error correcting code used.

How about security? The security argument for the case that Alice is dishonest remains the same as before, so we only have to worry about dishonest Bob. If Bob is dishonest, we make a worst case assumption and assume that all noise is in fact due to Bob's attack and no other noise occurs in the transmission. In other words, we assume that if Bob is dishonest then he may also eliminate all other errors - for example those occurring during transmission – and he is only limited by the assumption on his noisy storage device. Evidently, this means that Bob could now also use the error-correcting information that Alice sends in order to correct errors in his noisy memory $\mathcal{F}$! Our task is hence to understand the min-entropy *conditioned* also on the syndrome information S:

$$H_{\min}(X|\Theta, K, Q, S)\ .$$

Luckily, we can again employ the chain rule to bound the reduction in min-entropy due to conveying this additional information to Bob, just as we did in the case of QKD. That is,

$$H_{\min}(X|\Theta, K, Q, S) \geq H_{\min}(X|\Theta, K, Q) - \log|S|\ ,$$

where $\log|S|$ corresponds the number of bits of syndrome information sent. Remember from the case of QKD that in the limit of large n, there exist error-correcting codes such that $\log|S| \approx nh(p)$, where p is the bit-flip error rate of each bit in x and $h(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy.

To conclude note that just as in the case of QKD, it may no longer be possible to achieve security for a particular storage assumption $\mathcal{F}$ once p gets too large. This is indeed very intuitive, since if p is too large, Alice has to send so much error-correcting information that dishonest Bob (whose only noise comes from $\mathcal{F}$) can use this information to correct the errors in this quantum memory $\mathcal{F}$ and therefore break the security of the protocol.

Quiz 10.3.3 *Consider a scenario where Alice's device is noisy, such that whenever she wants to prepare the states* $\{|0\rangle, |1\rangle\}$*, she actually prepares states* $|0'\rangle = \cos\epsilon\,|0\rangle + \sin\epsilon\,|1\rangle\,, |1'\rangle = \sin\epsilon\,|0\rangle - \cos\epsilon\,|1\rangle$ *for some small* $\epsilon > 0$*. Suppose that Bob is aware of this imperfection. In such a scenario is* $H_{\text{min}}(X|\mathit{Bob})$ *larger or smaller than with the perfect device?*

a) $\longrightarrow$ ***Larger***
b) Smaller

10.4 Chapter Notes

The classical bounded storage model was defined in [**?**]. A good introduction to the use of physical assumptions in classical communication can be found in [**?**]. Yet, not only is classical memory cheap and plentiful, the small gap between what classical parties need in order to implement the protocol ($\Omega(n)$ bits of classical memory) vs. what the adversary needs to break the protocols (typically, $O(n^2)$ bits of classical memory) is too small to make this assumption useful in general. Inspired by these classical results, the bounded quantum storage model was put forward in [**?**], and the more general noisy storage model in [**?**, **?**].

To learn more about protocols and their properties in the noisy storage model we refer to the review article [**?**] and references therein. The best bounds for protocols in the noisy storage model mentioned in this chapter were obtained in [**?**]. The min-entropy splitting lemma is due to Wullschleger [**?**].

One can in principle implement two-party protocols using the same equipment used to realize BB'84 QKD. Implementations have been reported for bit commitment [**?**] and oblivious transfer [**?**]. If you want to learn more about the analysis of quantum cryptography protocols in the presence of imperfections encountered in real-world systems, we encourage you to take a look at these papers.