11 Further topics around encryption

One of the main goals of cryptography is to enable secure communication. In Chapter **??** we discovered the quantum one-time pad, which Alice can use to perfectly hide an n-qubit quantum state ρ using a $2n$-bit classical key k. This scheme has perfect correctness (given k it is possible to perfectly recover ρ from its encryption) and perfect secrecy (without knowledge of k no information at all can be obtained about ρ from its encryption). The only drawback is the necessity to share a large classical secret key. In Chapter **??** and following we developed quantum protocols for secure key distribution which precisely address the question of generating such a key. Problem solved?

Not quite. First of all, there are many topics of further interest in quantum cryptography. Some of these were explored in the previous chapters; moreover, every day researchers identify new, creative ideas of using quantum information to achieve certain tasks securely. Second, the question of encryption itself has many facets, most of which we have not had a chance to explore in depth. For example, what about this long key, is there any chance that we could make it shorter? After all, two rounds of QKD (and more considering all the extra rounds for testing) for each qubit that we want to encrypt, it is a lot of effort. Could we not reuse the key, or extend it in some way? And what about the requirement that Alice and Bob need to share the same key: what if I want to communicate with a group of friends—how do we all get the same key? In this chapter we start by looking at the possibility of shortening the key used for quantum encryption. We will see an impossibility result, and then open the door towards the fascinating world of computational security. Finally we will discuss a new possibility for quantum encryption, which is known as *certified deletion*: this is the possibility for the encrypter of a secret to request that the ciphertext is provably and irrevocably *erased*!

11.1 The key length requirements for secure quantum encryption

We start by investigating the need for a large classical key to achieve secure quantum encryption. In Chapter **??** we discovered Shannon's theorem, which states that a perfectly secure classical encryption scheme requires keys of length at least as long as the messages. Let's first see how we can extend this result to the quantum case. For this we need a definition of perfect security for quantum encryption.

Definition 11.1.1 *A quantum encryption scheme for n-qubit messages is specified by two families of quantum maps $\mathrm{Enc}_k : (\mathbb{C}^2)^{\otimes n} \mapsto (\mathbb{C}^2)^{\otimes m}$ and $\mathrm{Dec}_k : (\mathbb{C}^2)^{\otimes m} \mapsto (\mathbb{C}^2)^{\otimes n}$, where the index k ranges over some set of keys $\mathcal{K}$. The scheme is said to be*

- Perfectly correct *if for any n-qubit state ρ and any key $k \in \mathcal{K}$, $\mathrm{Dec}_k \circ \mathrm{Enc}_k(\rho) = \rho$.*
- Perfectly secure *if there exists an m-qubit state σ_0 such that for any n-qubit state ρ,*

$$\frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathrm{Enc}_k(\rho) \;=\; \sigma_0 \;. \tag{11.1}$$

In this definition the meaning of the correctness requirement is clear: encryption followed by decryption with the same key should return the initial quantum state. Observe that, for our definition to be as general as possible, we allow encryption schemes that increase the number of qubits of the message; in contrast, the quantum one-time pad has $m = n$.

For security, the requirement is that when the key k is chosen uniformly at random in $\mathcal{K}$ and hidden from the adversary, then the encrypted state is independent of the message: the state σ_0 is some fixed state that may depend on the scheme but not on ρ. Intuitively a scheme satisfying this condition would indeed deserve the name "perfectly secure", and we will justify this intuition later. Note that the quantum one-time pad does satisfy the condition (11.1), with $\sigma_0 = \frac{1}{2^n}\mathbb{I}$.

Exercise 11.1.1 Call a quantum encryption scheme *super-perfectly secure* if there exists a state σ_0 such that for all quantum states ρ_{AE} where A is n qubits and E is arbitrary,

$$\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \big(\mathrm{Enc}_k \otimes \mathbb{I}_E\big)(\rho_{AE}) \;=\; \sigma_0 \otimes \rho_E \;.$$

Clearly this definition is at least as strong, and in particular if E is empty we recover Definition 11.1.1. Intuitively "super-perfect security" is meant to guard against a situation in which the malicious Eve has some information correlated with Alice's message. In this case we want to make sure that the ciphertext no longer contains any such correlation.

Show carefully that any perfectly secure scheme is automatically super-perfectly secure.

Let's show that the definition of perfect security in Definition 11.1.1 indeed requires long keys.

Theorem 11.1.1 *Let* $(\mathrm{Enc}, \mathrm{Dec})$ *be a perfectly correct and secure quantum encryption scheme for* n *qubits. Then* $|\mathcal{K}| \geq 2^{2n}$.

Proof For simplicity in the proof we consider a scheme that encrypts n qubits into n qubits (i.e. $m = n$) and that is such that $\sigma_0 = \frac{1}{2^n}\mathbb{I}$. The general argument is a little more technical but leads to the same bound. Let $k = 1, \ldots, N$ index the possible keys, let p_k be the probability that the k-th key is chosen, and let U_i be the encoding unitary on key k, i.e. $\mathrm{Enc}_k(\rho) = U_k \rho U_k^\dagger$. (This is where we use that the scheme encrypts n qubits into n qubits; in general we would have to consider arbitrary channels.) We also define

$$\mathrm{Enc}(\rho) = \frac{1}{N}\sum_{k=1}^{N} \mathrm{Enc}_k(\rho) = \frac{1}{N}\sum_{k=1}^{N} U_k \rho U_k^\dagger \;.$$

Our goal is to show that necessarily $N \geq 2^{2n}$. So let's suppose for contradiction that $N < 2^{2n}$. The main observation is that when averaged over a random key the encoding map has exactly the same behavior as the one-time pad

$$\mathcal{E}(\rho) = \frac{1}{2^{2n}}\sum_{(k_1',k_2')} X^{k_1'} Z^{k_2'} \rho \big(X^{k_1'} Z^{k_2'}\big)^\dagger \;=\; \frac{1}{2^n}\mathbb{I}\,,$$

where k_1', k_2' range over n-bit strings and $X^{k_1'}$ denotes an X operator on the qubits associated with entries of k_1' that are equal to 1; similarly for $Z^{k_2'}$. We see that perfect security (together with our simplifying assumption that $\sigma_0 = \frac{1}{2^n}\mathbb{I}$) requires that $\mathrm{Enc}(\rho) = \mathcal{E}(\rho)$ for all ρ. From this it is possible to show, using a notion of unicity of the Kraus decomposition of a quantum channel (see Box **??**–we skip the details; to show it, consider the effect of encrypting one half of a maximally entangled state using either scheme) that there must exist an $2^{2n} \times 2^{2n}$ unitary matrix A such that for each k,

$$\sqrt{p_k} U_k \;=\; \sum_{(k_1',k_2')} A_{k,(k_1',k_2')} \frac{1}{\sqrt{2^{2n}}} X^{k_1'} Z^{k_2'} \;.$$

Here the indexing for the rows of A makes sense because we assumed that $N \leq 2^{2n}$. Using that the matrices $X^{k_1'} Z^{k_2'}$ are orthonormal with respect to the normalized trace inner product $\langle A, B\rangle \mapsto \frac{1}{2^n}\operatorname{Tr}(A^\dagger B)$ we can compute

$$\begin{aligned} p_k &= \frac{1}{2^n}\big\langle \sqrt{p_k}U_k, \sqrt{p_k}U_k\big\rangle \\ &= \frac{1}{2^{2n}} \sum_{(k_1',k_2')} \left|A_{k,(k_1',k_2')}\right|^2 \\ &\leq \frac{1}{2^{2n}} \ , \end{aligned}$$

where the second line uses orthonormality and the last line uses that the rows of A have euclidean norm at most 1 since A is unitary. So

$$1 = \sum_{k=1}^{N} p_k \ \leq \ \frac{N}{2^{2n}} \ ,$$

from which we deduce that $N \geq 2^{2n}$, as desired. □

So perfectly secure quantum encryption schemes require long keys! It seems like Alice and Bob will have to bite the bullet, and exchange long DVD's full of random bits to encrypt their quantum message. Or do they? Given Theorem 11.1.1, the only option is to modify our definition of perfectly secure encryption. Can we relax it in a way that it remains meaningful, but allows shorter keys? In the next subsections we describe two ideas on how this can be done. First, we can relax the notion that encryptions of distinct messages are perfectly indistinguishable. We discuss this possibility in the next section. Second, we can weaken our security requirement to only require that encryptions of distinct messages look similar to "reasonably powerful" adversaries. We discuss this in Section 11.1.2. Finally, we will conclude by giving a quick overview of the notion of *public-key* encryption.

11.1.1 Approximate encryption

As we saw, the requirement of perfect security inevitably imposes long keys. But do we really need perfect security? Suppose for example that we could construct a scheme that satisfies the weaker condition that for any ρ,

$$\Big\| \frac{1}{|\mathcal{K}|} \sum_{k\in\mathcal{K}} \operatorname{Enc}_k(\rho) - \sigma_0 \Big\|_1 \ \leq \ \varepsilon \ , \tag{11.2}$$

where ε is some very small quantity. By the interpretation of the trace distance, this would immediately imply that no adversary, given either $\operatorname{Enc}_k(\rho_0)$ or $\operatorname{Enc}_k(\rho_1)$, would be able to distinguish these two states with an advantage larger than ε. If ε is, say, 2^{-80}, this seems pretty safe; the adversary would have to see 2^{80} copies of our encryptions to reliably distinguish the messages that they encrypt. A scheme which satisfies the weaker requirement (11.2) instead of (11.1) is called an *ε-approximate encryption scheme*. Does there exist ε-approximate encryption schemes with short keys?

A simple idea for constructing an approximate encryption scheme is to start with the quantum one-time pad but only use a subset of all the possible keys. Let $\mathcal{K} \subseteq \{0,1\}^{2n}$ denote a subset. How small can we find a $\mathcal{K}$ such that the equation (11.2) holds, where Enc_k is the quantum one-time pad? It turns out that the answer is, roughly, $\log|\mathcal{K}| = n + O(\log n) + O(\log(1/\varepsilon))$. That is, by considering approximate encryption and even asking for an ε that is almost exponentially small in n, we can get a savings of a factor 2 in the key length, bringing it down to almost the same length as the *classical* one-time pad.

So how do we choose the keys? Without going into details, it is possible to show that a randomly

chosen set $|\mathcal{K}|$ of this size will work. Moreover, there also are explicit constructions of small sets of keys that will work, using techniques from the area of classical error-correcting codes. Finally, the size $|\mathcal{K}| \approx 2^n$ is optimal, i.e. no smaller set will give security.

Quiz 11.1.1 *Imagine applying the quantum one-time pad to a single qubit $|\psi\rangle$, such that we choose only one of three possible keys, corresponding to doing nothing ($k = 1$), applying an X operation ($k = 2$) or a Z operation ($k = 3$). In other words, we never apply XZ. Under this scheme, an encryption of $|0\rangle\langle 0|$ and an encryption of $|1\rangle\langle 1|$ have trace distance*

a) 0
b) $\frac{1}{3}$
c) $\frac{1}{2}$
d) 1

There is an important caveat to keep in mind with approximate encryption, which is explored in the next exercise.

Exercise 11.1.2 Suppose given a message $|m\rangle$ that is chosen uniformly at random in $\{0,1\}^n$, but such that an adversary Eve holds a copy of $|m\rangle$. That is, we imagine that the initial state of Alice and the eavesdropper is $|\Phi\rangle = 2^{-n} \sum_{m\in\{0,1\}^n} |m\rangle_A |m\rangle_E$. Show that if Alice encrypts her message using the quantum one-time pad, then the ciphertext is completely unknown to Eve, i.e.

$$\frac{1}{2^{2n}} \sum_{(k_1,k_2)} \big(\mathrm{Enc}_{(k_1,k_2)} \otimes \mathbb{I}_E\big)\big(\, |\Phi\rangle \langle\Phi| \,\big) \;=\; \rho_A \otimes \rho_E \;, \tag{11.3}$$

for some density matrices ρ_A and ρ_E that you will determine.

Exercise 11.1.3 Imagine now that Alice decides to encrypt her message using an *approximate* encryption scheme, such that the total number of keys is $K \leq \frac{1}{2}2^{2n}$. Show that in this case it must be the case that the trace distance between the left-hand side and the right-hand side in (11.3) is at least some constant.

The exercises show that while approximate encryption may be good enough to encode messages that are in tensor product with the environment (the adversary), one has to be careful that it is *not* sufficient to destroy *correlations*, as an adversary that has some quantum correlation with Alice's message before encryption may retain some quantum correlation with the ciphertext after encryption. This is in contrast to perfect encryption, which as you showed in Exercise 11.1 always perfectly destroys all correlations.

A saving of a factor 2 in the key length might not seem worth the trouble. To save more, we consider a further relaxation of the security definition, to *computational* security.

11.1.2 Computational Security

We already met the idea that security can be based on computational, as opposed to physical, assumptions twice in this book — first in Chapter **??** when discussing authenticated channels, and then in Chapter **??** when discussing computationally secure commitments based on the notion of pseudo-random generator. Formally, computational security can be defined in a similar way as physical security, through an appropriate *security game*. Let's see in more detail how this can be done for the case of encryption. Towards this we introduce a game between a *challenger*, Charlie, and an *adversary*, Eve. In the game, the challenger always plays honestly, while the adversary tries to win as best she can by optimizing its strategy.

1 Charlie generates parameters for the encryption scheme, i.e. he selects a key $k \in \mathcal{K}$ uniformly at random.
2 Eve prepares a quantum state ρ_{ME} of her choice, where M is a register the size of a plaintext message, and E is a quantum register that Eve keeps to herself. Eve sends the part of the quantum state in register M to Charlie.
3 Charlie selects a uniformly random $c \in \{0,1\}$. If $c = 0$ then Charlie encrypts M:

$$\rho'_{CE} = (\mathrm{Enc}_k \otimes \mathbb{I}_E)(\rho_{ME}) \ ,$$

and sends register C, now containing the ciphertext, to Eve. If $c = 1$ then Charlie first replaces the contents of M by a "dummy" message $|0\rangle\langle 0|_M$, encrypts the dummy message into register C, and sends C back to Eve. In this case,

$$\rho'_{CE} = (\mathrm{Enc}_k \otimes \mathbb{I}_E)(|0\rangle\langle 0|_M \otimes \rho_E) \ .$$

4 Eve produces a guess $d \in \{0,1\}$ and sends it to Charlie.
5 Charlie declares that the adversary has won if and only if $d = c$.

To understand this "game" let's first go through it in the case where Eve chooses the register E to be empty. Let ρ_M be the state prepared by Eve at step 2. Then, in case $c = 0$ the state sent back to the adversary at step 3 is precisely $\frac{1}{|\mathcal{K}|}\sum_k \mathrm{Enc}_k(\rho)$, while if $c = 1$ it is $\frac{1}{|\mathcal{K}|}\sum_k \mathrm{Enc}_k(|0\rangle\langle 0|)$. As we saw in Chapter **??**, by definition of the trace distance Eve's maximum success probability to distinguish these two states is exactly

$$\frac{1}{2} + \frac{1}{2}\left\| \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \mathrm{Enc}_k(\rho) - \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \mathrm{Enc}_k(|0\rangle\langle 0|) \right\|_1 \ .$$

We see that if Enc is ε-approximate secure in the sense of (11.2) then by the triangle inequality the adversary's success probability is at most $\frac{1}{2} + \varepsilon$. Conversely, if the adversary's success probability is at most $\frac{1}{2} + \varepsilon$ then by defining $\sigma_0 = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \mathrm{Enc}_k(|0\rangle\langle 0|)$ we get that (11.2) holds with right-hand side 2ε. In other words, the two statements

(a) The scheme Enc is ε-approximate secure,
(b) Eve's maximum success probability in the game defined above, when she is restricted to not use any E and Charlie plays honestly, is at most ε,

are equivalent (up to a factor 2 in the ε's). Using such a "security game", as opposed to an equation that needs to be satisfied, is an intuitive yet mathematically rigorous (and fun!) way of making a security definition.

The same reasoning applies when E is included, and corresponds to a strengthening of (11.2) which takes into account correlations; this strengthening is motivated by the discussion at the end of the previous section, which shows that the definition is indeed strictly stronger (see Exercise 11.1.3).

Remark 11.1.2 *The choice of encrypting $|0\rangle\langle 0|_M$ in case $c = 1$ in the definition of the security game is arbitrary. The point is that whatever message the adversary chooses to give to the challenger, they should not be able to distinguish an encryption of it from an encryption of some fixed message, such as $|0\rangle\langle 0|_M$. While we could have directly required that encryptions of 0 are indistinguishable from encryptions of 1, by letting the adversary choose the message we give her more power and hence obtain a potentially stronger notion of security (for example, when there are more than two possible plaintexts).*

exercise 11.1.4 Show that a quantum encryption scheme $(\mathrm{Enc}, \mathrm{Dec})$ is super-perfectly secure (in the sense of Exercise 11.1) if and only if the maximum success probability for the adversary in the

security game is exactly $\frac{1}{2}$. *[Don't forget to show both directions of the "if and only if"! The "if" part requires more work, so start with "only if"]*

The security game gives us a different way to think about the security definition for quantum encryption. Now we can ask the following question: what if we only care about certain types of adversaries, and we only need to be secure against them? We can then specialize the security game, and only ask about the success probability of the class of adversaries we care about. In a way we already did this when we considered two versions of the security definition, when E is required to be empty or not. And in the previous section we saw that if E is empty, and we allow success probabilities up to $\frac{1}{2} + \varepsilon$, then we can use shorter keys. So, placing restrictions on the adversary can help us create more efficient schemes! Of course, we always have to remember that the scheme is only secure up to the security definition, not any further.

Some other restrictions we could consider on the adversary is to have a bounded quantum memory, similarly to the model considered in Chapter **??**. Another alternative would be to imagine that the adversary only has a certain amount of computational time to invest in breaking the scheme. This amount of time should of course be allowed to be be much bigger than the space or time it takes to honestly encrypt and decrypt, but perhaps it is not infinite either (because who has infinite time?).

Investigating these questions can lead to very large improvements in the performance of encryption schemes as well as other cryptographic primitives. For example, it is possible to show that secure encryption against computationally bounded adversaries is possible with a key length that is only logarithmic, as opposed to linear, in the length of the message (for long enough messages)! Unfortunately, introducing such schemes would take us far beyond the scope of this book. We give a very informal description here and refer you to the chapter notes for pointers on where to learn more.

The main idea behind computational security is to postulate that a certain computational problem is *hard on the average* and construct an encryption scheme such that the only case when an adversary can win in the security game is if the adversary also has the ability to solve an instance of the computational problem that is related to the key. A typical computational problem used in classical cryptography is the problem of factoring. However, since this problem is not hard for quantum computers, it is not a good problem on which to base quantum encryption schemes. Instead some other computational problems have been used, including problems related to error-correcting codes (finding a minimum-weight codeword) and integer lattices (finding a closest lattice vector). Security of the scheme is proven by *reduction*: we show that if Eve is an adversary who succeeds in the security game with too large probability, say more than $\frac{1}{2} + \varepsilon$, then the same Eve could be used to break the computational problem. So if we assume that the latter is hard, then the scheme is secure.[1]

Beyond efficiency savings, computational security can also lead to conceptually different schemes. The most important family of schemes beyond private-key encryption is called *public-key* encryption, and we briefly discuss it in the next section.

11.1.3 Public-key encryption

The encryption schemes we have considered so far all work under the same natural premise: that Alice and Bob should both share the same key, and that this key can be used to either encrypt or decrypt. This is natural because since decryption is the inverse of encryption, it makes sense to define both operations from the same piece of information, the key. Hence such schemes are usually called "private-key" quantum encryption schemes, where the term "private" refers to the fact that the scheme is only secure as long as the same key k remains private to the sender and receiver.

[1] Sometimes we think of this as a "win-win" notion: *either* the scheme is secure, *or* someone has found a new algorithm for a hard problem!

A striking observation that revolutionized cryptography in the 1990s is that encryption and decryption do not need to be treated symmetrically. From the security standpoint, it is crucial that only the authorized party is able to decrypt a ciphertext, but it is a priori not a problem that anyone would be able to encrypt.[2] From the mathematical standpoint there is also a natural asymmetry: some functions are easy to compute in one direction, but hard in the other. A prime example is multiplication (easy), whose inverse is factoring (hard). The idea of public-key cryptography is to leverage this mathematical asymmetry to implement an encryption scheme such that anyone can encrypt, but only the trusted user can decrypt.

More precisely, in *public-key encryption* the keys come in pairs (sk, pk) where sk is a *secret key* and pk is a *public key*. As their name indicates, the secret key is meant to be kept private (e.g. only Bob has it) while the public key can be made publicly available (Bob can publish it on his website, or include it as part of his email signature). Encryption can be done using only pk, while decryption requires sk. Therefore, the security game is exactly the same as the one in the previous section, except that the challenger generates a pair (sk, pk), keeps sk to themselves, and gives pk to the challenger. Public-key encryption schemes necessarily rely on computational security (because from an information-theoretic point of view, the public key pk uniquely specifies a private key sk, and so such a scheme can always be broken in principle), and can be implemented using similar assumptions to the ones discussed in the previous section.[3]

Since we are able to use quantum key distribution to distribute private keys, do we really need public-key cryptography? There are many reasons why we do. First of all, implementing QKD remains technologically challenging. It is already hard when there are two parties; however, a major benefit of public-key cryptography is the ability for *any* user to publicly send a message to Bob by encrypting using his public key, while having the guarantee that only Bob will be able to decrypt the messages; in this sense the asymmetry helps. This use case is also relevant for quantum cryptography and so we expect that both private-key schemes, for which encryption and decryption of a single message is typically very fast, and public-key schemes, which require higher encryption and decryption times but are much more efficient in settings where there are many users, since ciphertexts can be "re-used," can co-exist.

11.2 Encryption with certified deletion

Even when Alice only wants to transmit classical messages to Bob, there may be a use for encryption using quantum ciphertexts. Suppose for example that Alice has some private information, such as the details of a future financial transaction that she wishes to have executed. Alice could share a key with her bank, encrypt the transaction details using a classical one-time pad, and send the resulting ciphertext to the bank. The bank can then use its own copy of the key to decrypt the information and execute the transaction. However, imagine now that this is a timed transaction: Alice would like to be able to send the ciphertext to the bank, and then tell the bank to store the ciphertext and wait for her next signal. A few days later Alice can decide that she wants to proceed with her transaction, in which case she would tell the bank to "go ahead", i.e. decrypt and execute. But what if Alice has changed her mind? Wouldn't it be nice if there was a special "delete" signal that she could send to request that the bank *delete* her ciphertext, without having learned any information at all about Alice's aborted transaction?

Another example is with data protection. Nowadays, we are constantly asked to submit personal information, such as our address or birth date, to some websites. In certain countries the websites are legally

[2] One might nevertheless worry about the possibility of "spoofing" a message. This problem is solved by *authentication*, which is a separate technique that can be combined with encryption (see Section **??**.)

[3] In general, the mathematical assumptions required to implement a public-key encryption scheme tend to be more demanding than the assumptions required to implement private-key encryption, because public-key encryption requires more structure.

required to delete this information shortly after the transaction has been completed. But do they actually do this? How could one check that they have not kept a copy in some secret vault of theirs?

Classically both of these tasks are impossible to realize securely. If the bank can either decrypt or delete then it can also do both: simply copy the ciphertext, decrypt one copy and use the other copy to "prove" deletion. Similarly, there is no way to prevent a website from storing a copy of any classical information we send to to it.

However, if the information is quantum then by the no-cloning principle it is no longer clear that this strategy can be applied. So, can certified deletion be realized by using quantum information? Let's start by introducing a security definition that captures this new property. Afterward we'll investigate a scheme that satisfies the definition.

11.2.1 Security definition

For simplicity let's focus on defining security for schemes that encrypt a single classical bit at a time. As for a standard encryption scheme there should be a key generation procedure, together with encryption and decryption procedures. In addition to these we now include two additional procedures, the procedure Del that given a ciphertext "deletes" it and obtains a "proof of deletion" π, and a "verification of deletion" procedure VerDel that checks that deletion has been produced correctly. Here is a more formal definition. (The first time you read the definition you can ignore the role of the "security parameter" λ. This quantifies the security of the scheme with respect to deletion, and we'll explain it later.)

Definition 11.2.1 *Let $\lambda \geq 1$ be an integer. A* quantum encryption scheme with certified deletion *with security $2^{-\lambda}$ is specified by the following procedures:*

1. *A classical probabilistic key generation procedure $k \leftarrow \mathrm{Gen}()$ that generates a secret key k.*[4]
2. *An encryption procedure $(c, dk) \leftarrow \mathrm{Enc}_k(m)$ that given a plaintext $m \in \{0,1\}$ returns a ciphertext c together with a "deletion key" dk (that may depend on m). The ciphertext c may contain a quantum component, that is, it can be a CQ state.*
3. *A decryption procedure $m \leftarrow \mathrm{Dec}_k(c)$.*
4. *A deletion procedure $\pi \leftarrow \mathrm{Del}(c)$ that given a ciphertext c produces a classical proof of deletion π. Note that Del does not require the key k.*
5. *A verification of deletion procedure $v \leftarrow \mathrm{VerDel}_k(dk, \pi)$ that takes as input a deletion key dk and a deletion proof π and returns a bit $v \in \{0,1\}$, where 1 stands for "accept" and 0 stands for "reject".*

Let's now discuss security requirements. First, as usual the scheme is called *perfectly correct* if for every key k and plaintext m, $\mathrm{Enc}_k(m) = (c, dk)$ implies that $\mathrm{Dec}_k(c) = m$. In addition, we add the requirement that for any proof of deletion that is generated by the correct deletion procedure, $\pi = \mathrm{Del}(c)$, it holds that $\mathrm{VerDel}_k(dk, \pi) = 1$.

For security, we first require perfect security for the encryption scheme, i.e. condition (11.1).[5] What about deletion? Informally, we would like that for any "adversary" holding a ciphertext c, if the adversary successfully "proves deletion" then it becomes impossible for them to recover the plaintext m associated with c, even if they are later given the key (of course, if they don't have the key, then encryption security guarantees that they can't recover m). There are many quotes in this sentence! To formalize the intuition we introduce a security game. This game is of the same type as the one in Section 11.1.2, and once again

[4] Recall that the notation $X \leftarrow \textsc{Proc}(Y)$ means that we use the variable X to denote the outcome of running the procedure $\textsc{Proc}$ on input Y. If Y is omitted, then it means that PROC takes no input.

[5] More generally, we could consider a weaker security notion of the kind considered in the first part of this chapter. For simplicity, and because we can, we focus on the stronger notion.

it is played between a honest *challenger* Charlie and a possibly malicious *adversary* Eve. The idea is that a scheme will be called a *certified deletion* (encryption scheme) with security λ if and only if no adversary can win in the game with probability much larger than $2^{-\lambda}$.

1 Charlie selects a key $k \leftarrow \mathrm{Gen}()$.
2 Eve prepares an arbitrary quantum state ρ_{ME}, where M is a classical register the size of a plaintext message and E is a quantum register that the adversary keeps to herself. Eve sends register M to Charlie.
3 Charlie selects a $c \in \{0,1\}$ uniformly at random.
   - If $c = 0$ then Charlie encrypts M:
   $$\rho'_{CDE} = (\mathrm{Enc}_k \otimes \mathbb{I}_E)(\rho_{ME}) \ ,$$
   and sends register C, now containing the (possibly quantum) ciphertext, to Eve. Charlie keeps register D, which contains the (classical) deletion key dk.
   - If $c = 1$ then Charlie first replaces the contents of M by a "dummy" message $|0\rangle\langle 0|_M$, encrypts the dummy message into register C, and sends C back to Eve. As before, Charlie keeps register D.
4 Eve sends a "proof of deletion" $\pi \in \{0,1\}^\lambda$ to Charlie.
5 Charlie sends the secret key k to Eve.
6 Eve produces a guess $d \in \{0,1\}$.
7 Charlie declares that Eve has won if and only if $d = c$ *and* $\mathrm{VerDel}_k(dk, \pi) = 1$.

We should convince ourselves that this game captures the intuition of the "deletion security" that we want for our encryption scheme. Compared to the game in Section 11.1.2, there are two key differences. First, Eve is asked for some additional information: at step 4, she has to return a "deletion proof" π. What we imagine here is that Charlie has asked for his ciphertext to be deleted, and Eve is supposed to comply by sending the proof π, which is checked in the last step. Now, the validity of this proof is *supposed* to guarantee that Eve has deleted the ciphertext. How do we check this? Here comes the second difference: at the next step, Charlie *reveals* the secret key k to the challenger! We say that Eve wins the game if, first of all her "deletion certificate" is accepted, and second she is able to discover which plaintext was encoded by Charlie. Note that if the deletion that is supposed to have happened at step 4 does not affect the ciphertext (or the ciphertext can be copied) then by correctness of the encryption scheme it is easy to win in this game, simply by decrypting c once k is given (and, for example, choosing $\rho_M = |1\rangle\langle 1|$ in step 2, so that the challenger can indeed distinguish between the cases $c = 0$ and $c = 1$). So, for any scheme that satisfies this definition, clearly there must be something interesting going on: decryption is possible before π is produced, but no longer after; which is exactly what we want.

The next exercise shows that in the security game it is essential that the key is revealed to Eve only *after* the proof of deletion has been obtained. Otherwise, there will always be an adversary that is able, given the key, to produce both a valid deletion certificate and a correct guess for the the bit c.

Exercise 11.2.1 Show that if steps 4 and 5 are inverted then for any perfectly correct certified deletion scheme there is an adversary that succeeds with probability 1 in the security game.

11.2.2 A construction

As we hinted earlier, the task of encryption with certified deletion is closely related to the notion of no-cloning. This is because if the ciphertext is cloneable then it will always be possible to win in the security game; hence the existence of an encryption scheme with certified deletion implies that there exists quantum states that cannot be cloned. Based on this observation, a natural idea for implementing

a certified deletion scheme would be to include an "uncloneable" component in our ciphertexts. For example, we could add a randomly generated Wiesner quantum money state (see Chapter **??**) to each ciphertext. This is a good idea but by itself it is unlikely to work, as we must somehow tie the part of the ciphertext that contains information about the plaintext to the "uncloneable" part.

We now introduce a scheme that does just that. To describe the scheme, we identify a string $\mathcal{I} \in \{0,1\}^\lambda$ with the subset $\mathcal{I} \subseteq \{1,\ldots,\lambda\}$ which is the list of positions at which $\mathcal{I} = 1$. We also recall the notation $|x\rangle_\theta = H^\theta |x\rangle$ for the BB'84 states, where $x, \theta \in \{0,1\}$.

1 The key space is $\mathcal{K} = \{0,1\} \times \{0,1\}^\lambda$. The key generation procedure returns a uniformly random $k = (u, \mathcal{I})$ such that $u \in \{0,1\}$ and $\mathcal{I}$ is a subset of $\{1,\ldots,\lambda\}$.
2 Given a message $m \in \{0,1\}$ and a key $k = (u,\mathcal{I})$, $\mathrm{Enc}_k(m)$ generates $x \leftarrow \{0,1\}^\lambda$ uniformly at random and returns the ciphertext $c = (c', |\phi\rangle)$ where $c' = m \oplus u \oplus_{i\in\overline{\mathcal{I}}} x_i$ and $|\phi\rangle = |x_1\rangle_{\mathcal{I}_1} \cdots |x_\lambda\rangle_{\mathcal{I}_\lambda}$, together with the deletion key $dk = x$.
3 Given a ciphertext $c = (c', |\phi\rangle)$ and a key $k = (u,\mathcal{I})$, Dec_k measures $|\phi\rangle$ in the standard basis to obtain a string y and returns $m = c' \oplus u \oplus_{i\in\overline{\mathcal{I}}} y_i$.
4 Given a ciphertext $c = (c', |\phi\rangle)$, Del measures $|\phi\rangle$ in the Hadamard basis to obtain a string z and returns $\pi = z$.
5 Given $k = (u,\mathcal{I})$, $\mathrm{VerDel}_k(\pi, dk)$ returns 1 if and only if $\pi_i = dk_i$ for all $i \in \mathcal{I}$.

To understand a scheme it is always useful to start by ignoring all the parts included to guarantee security and focus on checking that the scheme is correct. Let's do this. For any message $m \in \{0,1\}$, according to item 2. the associated ciphertext takes the form $c = (c', |\phi\rangle)$ where $c' = m \oplus u \oplus_{i\in\overline{\mathcal{I}}} x_i$ and $|\phi\rangle = |x_1\rangle_{\mathcal{I}_1} \cdots |x_\lambda\rangle_{\mathcal{I}_\lambda}$. When the decryption procedure measures $|\phi\rangle$ in the standard basis to obtain y, by definition we have that $y_i = x_i$ whenever $\mathcal{I}_i = 0$, because then $|x_i\rangle_{\mathcal{I}_i} = |x_i\rangle$. Since according to our notation $\mathcal{I}_i = 0$ is equivalent to $i \notin \mathcal{I}$, we get that

$$c' \oplus u \oplus_{i\in\overline{\mathcal{I}}} y_i \;=\; c' \oplus u \oplus_{i\in\overline{\mathcal{I}}} x_i \;=\; m \;.$$

So the scheme is perfectly correct. This correctness follows from the fact that, for decryption, the "$\mathcal{I}$" part of the private key tells us exactly which qubits of $|\phi\rangle$ were encoded in the standard basis and contain information that should be used for decryption.

The next step is to argue that the scheme is perfectly secure as an encryption scheme. To see this we can think of encryption as taking place in two steps. First the encrypter chooses a random x and $\mathcal{I}$ and returns the state $|\phi\rangle$. Clearly this is completely independent from the message and leaks no information whatsoever about it. Second the encrypter privately computes $m' = m \oplus_{i\in\overline{\mathcal{I}}} x_i$, which does depend on the message, and returns $m' \oplus u$ for a uniformly random u. Since adding u acts like a classical one-time pad, this part of the ciphertext is also perfectly secure and independent from $|\phi\rangle$. Hence for any single-bit message m it holds that

$$\frac{1}{|\mathcal{K}|} \sum_{k\in\mathcal{K}} \mathrm{Enc}_k(m) = \frac{1}{2}\mathbb{I} \otimes \sigma_0 \;,$$

where the first $\frac{1}{2}\mathbb{I}$ is the one-time padded m' and σ_0 represents a uniform mixture over all possible $|\phi\rangle$ (which you can check equals $2^{-\lambda}\mathbb{I}$, where the identity is over λ qubits). Therefore the scheme is perfectly secure.

It remains to show the certified deletion property! This requires more work, and we devote the next section to it.

Quiz 11.2.1 *Suppose that in the construction we choose $\lambda = 1$. Is the scheme still a perfectly secure encryption scheme?*

a) Yes
b) No

11.2.3 Proof of certified deletion property

We first consider the case where $\lambda = 1$. Let's rewrite the security game from Section 11.2.1, when specialized to our construction from the previous section, and the choice $\lambda = 1$. For reasons that will become clear later, we re-label $\mathcal{I}$ as $\theta \in \{0,1\}$. We obtain the following game.

1 Charlie selects $u \in \{0,1\}$ and $\theta \in \{0,1\}$ uniformly at random.
2 Charlie selects $c \in \{0,1\}$ uniformly at random. He selects $x \in \{0,1\}$ uniformly at random and creates $|\phi\rangle = |x\rangle_\theta$. If $\theta = 0$ Charlie sets $c' = c \oplus u \oplus x$. If $\theta = 1$ Charlie sets $c' = c \oplus u$. Charlie returns $c = (c', |\phi\rangle)$ to Eve.
3 Eve sends a deletion proof $\pi \in \{0,1\}$ to Charlie.
4 Charlie sends $k = (u, \theta)$ to Eve.
5 Eve produces a guess $d \in \{0,1\}$.
6 Charlie declares that Eve has won if and only if $d = c$ *and* (if $\theta = 1$ then $\pi = x$).

Remark 11.2.1 *In this description we did not let Eve choose the plaintext m and we also ignored the possibility for her to prepare a plaintext m that is correlated to some quantum information in register E. It is a good exercise to convince yourselves that both changes are without loss of generality, i.e. they do not reduce Eve's power. More formally, if any adversary can succeed in the earlier security game with probability ε then they can also succeed in this new security game with probability ε. Note that this simplification relies on the fact that we are considering a scheme that encrypts a single classical bit only.*

In the next step we are going to give more power to Eve. First of all, we will only check the condition that $d = c$ in the case when $\theta = 0$. Note that when $\theta = 0$, then Eve receives $c' = c \oplus u \oplus x$, and she also receives u at step 4. So the probability that she guesses c correctly is exactly the same as the probability that she guesses x (since she can convert from one to the other using $x = c \oplus (c' \oplus u)$, where she always has both c' and u). So, in this step we replace item 6 by

6 Charlie declares that Eve has won if and only if (if $\theta = 0$ then $d = x$) *and* (if $\theta = 1$ then $\pi = x$).

This new version of the game can only be easier for Eve. Finally, we observe that in this new version u no longer plays any role at all, so we can simply remove it. Slightly reorganizing the description of the steps we arrive at the following game.

1 Charlie selects $x \in \{0,1\}$ and $\theta \in \{0,1\}$ uniformly at random. He creates $|\phi\rangle = |x\rangle_\theta$ and sends $|\phi\rangle$ to Eve.
2 Eve sends a deletion proof $\pi \in \{0,1\}$ to Charlie.
3 Charlie sends θ to Eve.
4 Eve produces a guess $d \in \{0,1\}$.
5 Charlie declares that Eve has won if and only (if $\theta = 0$ then $d = x$) *and* (if $\theta = 1$ then $\pi = x$).

With these simplifications in place, our goal is to show that no adversary can succeed in the game with probability that is too close to 1: the smaller a bound we can show the better. To do this we apply a similar proof strategy to our analysis of the BB'84 protocol in Chapter **??**. Specifically, we start by considering a purified version of the game, as follows.

1 Eve is split in two parts, B and E. Eve prepares an arbitrary state ρ_{CBE} where C is a single qubit and sends C to Charlie.

2 Charlie selects a $\theta \in \{0,1\}$ uniformly at random. He measures C in the basis θ to obtain an $x \in \{0,1\}$.
3 B sends $\pi \in \{0,1\}$ to Charlie.
4 Charlie sends θ to E, who responds with a $d \in \{0,1\}$.
5 Charlie declares that Eve has won if and only (if $\theta = 0$ then $d = x$) *and* (if $\theta = 1$ then $\pi = x$).

Once again, this new, "purified" game gives more power to the adversary. To see why, observe that Eve could first prepare a state of the form $|\mathrm{EPR}\rangle_{CE} \otimes |0\rangle_B$ and send C to Charlie; then, she could compute π from E and copy it to register B, and leave the post-measurement state in E until she receives θ. It's not hard to see that any adversary using a strategy of that form succeeds in the purified game with the same probability as in the non-purified game. As usual, this is because measuring an EPR pair in any basis has the effect of collapsing both halves of the EPR pair to the same post-measurement state.

To conclude the analysis of the purified game we make use of an entropic uncertainty relation that is a generalization of the first inequality in (**??**). This relation can be stated as follows. For any state ρ_{ABE} where A is a single qubit, it holds that

$$H_{\max}(X_A|B) + H_{\min}(Z_A|E) \geq 1 \,, \tag{11.4}$$

where X_A is a random variable that denotes the outcome of a measurement of A in the Hadamard basis, and Z_A the outcome of a measurement of A in the standard basis. The second entropy, $H_{\min}(Z_A|E)$, we are already familiar with, and this is equal to $-\log(P_{guess}(Z_A|E))$, where $P_{guess}(Z_A|E)$ is exactly the maximum probability with which the adversary can succeed in the "(if $\theta = 0$ then $d = x$)" part of the security game. The quantity $H_{\max}(X_A|B)$ is the *max-entropy*. This is defined a little bit differently from the min-entropy. For our purposes we only need to consider the case where X_A and B are both a single classical bit, since we may as well consider B to contain the proof $\pi \in \{0,1\}$. If we let $p(x,b)$ denote the joint distribution of two bits x and b, then

$$H_{\max}(X|B) \;=\; \log\Big(\sum_b \Pr(B=b)\Big(\sum_x \sqrt{\Pr(X=x|B=b)}\Big)^2\Big) \,.$$

While this expression may seem a little more complicated than we'd like, from a qualitative point of view we can observe that $H_{\max}$ is only close to 1 if the expression inside the $\log$ is close to 2, which requires $\sum_x \sqrt{\Pr(X=x|B=b)}$ to be close to $\sqrt{2}$ for both values of b. This, in turn, requires $\Pr(X=x|B=b)$ to be close to $\frac{1}{2}$ for both values of x. In other words, for $H_{\max}$ to be close to 1, x must be different from b with probability close to $1/2$.

To summarize our findings, qualitatively the entropic equation (11.4) implies the existence of a trade-off between the probability that $d = x$ (when $\theta = 0$) and that $\pi = x$ (when $\theta = 1$) in the purified version of the security game. This is because at least one of the two entropies must be larger than $1/2$, and the qualitative reasoning above suggests that this implies an upper bound on the adversary's probability of winning in the corresponding part of the security game. Concretely, this trade-off implies that there is a constant $0 \leq p_s < 1$ such that no adversary can win in the game with probability larger than p_s. (It is possible to obtain precise estimates on p_s by carefully working through the definitions of the entropies and their relation to the guessing probabilities, but we satisfy ourselves with the qualitative statement.)

The constant p_s we have obtained bounds the success probability of an adversary in the certified deletion security game. However, this constant might not be very small! Instead we would like Eve to have a probability of cheating, i.e. providing a valid proof of deletion *and* being able to recover the plaintext, that is very small. This is why in the definition of the scheme we introduced a parameter λ that can be bigger than 1. From the point of view of the security game, considering higher parameters λ is equivalent to performing a repetition of the case $\lambda = 1$ in parallel, multiple times. To analyze the game for general λ we can proceed in two different ways. First of all, similar to our work in Chapter **??** we can consider the case of an adversary that behaves in an i.i.d. manner. In this case we directly obtain an upper

bound of the form p_s^λ on the success probability in the λ-repeated game. This bound goes exponentially fast to zero with λ, and so by choosing λ sufficiently large we can make the success probability as small as we want. However, in general the adversary may not behave in an independent manner and can apply a global strategy in the security game. The analysis of such strategies is challenging technically, and lies beyond the scope of this book. Suffice it to say that, even in this more general setting, an exponentially decaying bound on the success probability can also be shown. This proves that the scheme introduced in the previous section is a good certified deletion encryption scheme. One more success for quantum information!

Quiz 11.2.2 *Let's check the uncertainty relation* (11.4) *on a couple examples. First, suppose that* ρ_{ABE} *consists of an EPR pair between* A *and* E*, and* B *is in state* $|0\rangle_B$*. In this case, what is the value of the pair* $(H_{\max}(X_A|B), H_{\min}(Z_A|E))$*?*

a) $(0,0)$
b) $(0,1)$
c) $(1,1)$
d) $(1,0)$

Quiz 11.2.3 *Same question if* AB *is in an EPR pair, and* E *is in state* $|0\rangle_E$*.*

a) $(0,0)$
b) $(0,1)$
c) $(1,1)$
d) $(1,0)$

11.3 Chapter notes

For the general argument showing that perfect n-qubit quantum encryption schemes require keys of length $2n$, see [**?**]. The problem of approximate encryption is considered in [**?**], where a randomized construction is given. For a deterministic construction along the lines mentioned in this chapter, see [**?**]. The discussion in this chapter only scratches the surface of computational security, which is not a focus of this book. A comparison of many a priori different definitions of computational security for quantum encryption can be found in [**?**]. For much much more on computational security in the classical setting, we refer to the classic introductory book by Katz and Lindell [**?**].

The notion of encryption with certified deletion is studied in [**?**], which is where the protocol given here is adapted from. The uncertainty relation (11.4) is shown in [**?**].

11.4 Quiz Solutions

Quiz 11.1.1 b)
Quiz 11.2.1 a)
Quiz 11.2.3 c)