# 10 Quantum encryption

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

Of course not. Not only are there many further topics of interest in quantum cryptography, which we will explore in the following chapters, but the question of encryption itself still has many facets to reveal. Indeed, there are many questions that we can ask. 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's 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 examine those questions. In addition we will uncover some new possibilities for quantum encryption, such as encryption with certifiable deletion, that make it an attractive option, adding on special quantum features even for users who only care about encrypting classical messages.

## 10.1 Notions of quantum encryption

We start by investigating the need for a large classical key. 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 10.1.1** *A private-key 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*

- Perfectly correct *if for any $n$-qubit state $\rho$ 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 $\sigma_0$ such that for any $n$-qubit state $\rho$,*

$$\frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathrm{Enc}_k(\rho) \;=\; \sigma_0 \;. \tag{10.1}$$

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. 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 $\sigma_0$ is some fixed state that may depend on the scheme but not on

$\rho$. 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, with $\sigma_0 = \frac{1}{2^n}\mathbb{I}$.

We remark that in the definition we used the terminology "private-key" quantum encryption. 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. In Section 10.1.3 we will see another type of encryption scheme such that the sender and receiver don't need to use the same key, and the key for encryption is "public". Such schemes will be called "public-key" encryption schemes.

**Exercise 10.1.1** Now that we know about side information, we would be justified in arguing that the following security definition is stronger: a quantum encryption scheme is called *super-perfectly secure* if there exists a state $\sigma_0$ such that for all quantum states $\rho_{AE}$ where $A$ is $n$ qubits and $E$ is arbitrary,

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

Indeed, this would mean that even if the adversary has some prior correlation with the message, represented here by the system $E$, then they have no information at all about the encrypted message. Show carefully that any perfectly secure scheme is automatically super-perfectly secure.

Let's now show that the definition of perfect security in Definition 10.1.1 indeed requires long keys.

**Theorem 10.1.1** *Let* $(\mathrm{Enc},\mathrm{Dec})$ *be a perfectly correct and secure private-key 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 CPTP maps.) 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 $\rho$. From this it is possible to show (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 $(A,B)\mapsto\frac{1}{2^n}\,\mathrm{Tr}(A^\dagger B)$

we can compute

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

where the second line uses that $\|X^{k_1'}Z^{k_2'}\|_F^2 = 2^n$ and the last line uses that the rows of $A$ have norm at most 1 since $A$ is unitary. So

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

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

So perfectly secret quantum encryption schemes require long keys. To encrypt using shorter keys we will have to let go of the notion of perfect security. There are two main ways in which 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 further to only require that encryptions of distinct messages look similar to "reasonably powerful" adversaries. We discuss this in Section 10.1.2. Finally, we will conclude by giving a quick overview of the notion of *public-key* encryption.

## 10.1.1 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 $\rho$,

$$\Big\|\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\mathrm{Enc}_k(\rho) - \sigma_0\Big\|_1 \,\leq\, \varepsilon\,, \tag{10.2}$$

where $\varepsilon$ is some very small quantity. By the interpretation of the trace distance, this would immediately imply that no adversary, given either $\mathrm{Enc}_k(\rho_0)$ or $\mathrm{Enc}_k(\rho_1)$, would be able to distinguish these two states with an advantage larger than $\varepsilon$. If $\varepsilon$ 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 (10.2) instead of (10.1) is called an *$\varepsilon$-approximate encryption scheme*.

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 (10.2) holds, where $\mathrm{Enc}_k$ 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 $\varepsilon$ that is almost exponentially small in $n$, we can get a savings of a factor 2 in the key length, bringing it down almost to the same length as the *classical* one-time pad. 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.

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

**Exercise 10.1.2** Suppose given a message $|m\rangle$ that is chosen uniformly at random in $\{0,1\}^m$, 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 cyphertext is completely unknown to Eve, i.e.

$$\frac{1}{2^{2n}} \sum_{(k_1,k_2)} \big(\mathrm{Enc}_{(k_1,k_2)} \otimes \mathbb{I}_E\big)\big(|\Phi\rangle\langle\Phi|\big) \;=\; \rho_A \otimes \rho_E \;, \tag{10.3}$$

for some density matrices $\rho_A$ and $\rho_E$ that you will determine.

**Exercise 10.1.3** Imagine now that Alice decides to encrypt her message using an *approximate* encryption scheme, such that the total number of keys is $K \le \frac{1}{2}2^{2n}$. 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 (10.3) is at least some constant.

The exercise shows 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 cyphertext after encryption. This is in contrast to perfect encryption, which perfectly destroys all correlations.

A savings 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.

### 10.1.2 Computational Security

To introduce the notion of computational security we reformulate the security definition for approximate encryption from the previous section as a game between a *challenger* and an *adversary*. In the game, the challenger always plays honestly, while the adversary tries to win at best it can by optimizing its strategy.

1 The challenger generates parameters for the encryption scheme, i.e. it selects a key $k \in \mathcal{K}$ uniformly at random.
2 The adversary prepares an arbitrary quantum state $\rho_{ME}$, where $M$ is a register the size of a plain-text message, and $E$ is a quantum register that the adversary keeps to itself. The adversary sends register $M$ to the challenger.
3 The challenger selects a uniformly random $c \in \{0,1\}$. If $c = 0$ then the challenger encrypts $M$:

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

and sends register $C$, now containing the ciphertext, to the adversary. If $c = 1$ then the challenger 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 the adversary. In this case,

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

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

To understand the definition let's first think through it in the case where we force the system $E$ to be trivial. Let $\rho_M$ be the state prepared by the adversary 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 $\frac{1}{|\mathcal{K}|}\sum_k \mathrm{Enc}_k(|0\rangle\langle 0|)$. By definition of the trace distance the adversary's maximum success probability is

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

We see that if $\mathrm{Enc}$ is $\varepsilon$-approximate secure in the sense of (10.2) 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 (10.2) holds with right-hand side $2\varepsilon$.

The same reasoning applies when $E$ is included, and corresponds to a strengthening of (10.2) 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.

Note that the choice of encrypting $|0\rangle\langle 0|_M$ in case $c = 1$ in the definition of the security game is pretty 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 it more power and hence obtain a potentially stronger notion of security (for example, when there are more than two possible plaintexts).

The security game introduced above thus provides a different way to think about the security definition for quantum encryption. This new perspective allows us to ask the following question: what if we only care about certain classes of adversaries? What if, for example, we restrict the adversary to have quantum memory, or computational time, bounded by some number, which naturally should be much bigger than the space or time it takes to honestly encrypt and decrypt, but perhaps is not infinite either (because who has infinite time?). Perhaps surprisingly, this question leads to very large improvements in the performance of encryption schemes as well as other cryptographic primitives. Investigating these questions in detail would lead us far beyond the scope of this book, so we restrict ourselves to an informal discussion and refer the reader to the references in the chapter notes for 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 in order for the adversary to win in the security game, the adversary must be able to solve an instance of the computational problem that is related to the private key (and ideally, such that a uniform key leads to a uniform instance of the computational problem). 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 proved by *reduction*: given any adversary that succeeds in the security game with too large probability, one shows how this adversary can be used to break the computational problem. While in principle this may be possible, we don't believe it to be possible in any reasonable amount of time, and so the scheme is just as secure as the underlying problem is truly hard.[1]

The savings obtained by restricting to computational security can be very large, indeed exponential. Indeed there exist encryption schemes that are computationally secure and can encrypt messages of any length $n$ using a key that scales only with the desired "security", i.e. it is possible to use a key of length $\lambda$ to encrypt $n$-bit messages with security $\varepsilon = 2^{-\Omega(\lambda)}$. 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.

[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!

### 10.1.3 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.

A striking observation that revolutionized cryptography in the 1990s is that encryption and decryption need not be treated symmetrically. From the security standpoint they don't need to be symmetric: while it is crucial that only the authorized party is able to decrypt a ciphertext, 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 considered 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, which are typically more efficient and can be more secure, and public-key schemes, which require higher encryption and decryption times but are much more efficient in settings where there are many users, can co-exist.

## 10.2 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 for a future financial transaction that she wishes to have executed. Alice could establish a private 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

[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.

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

able to send the ciphertext to the bank, but 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 ever having learned any information at all about Alice's aborted transaction?

Classically this is clearly impossible. 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. However, if the ciphertext is quantum then by the no-cloning principle it is no longer clear that this can always be done. Nevertheless the definition of quantum encryption is not sufficient to cover this use case, as for example the quantum one-time pad, when used on a classical message, results in a perfectly cloneable ciphertext. Let's then introduce a security definition that covers our new use case for quantum encryption, and then give a scheme that realizes the definition.

### 10.2.1 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 $\mathrm{Del}$ that given a ciphertext "deletes" it and obtains a "proof of deletion" $\pi$, and a "verification of deletion" procedure $\mathrm{VerDel}$ 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 parameter $\lambda$. This quantifies the security of the scheme with respect to deletion, and we'll explain it later.)

**Definition 10.2.1** *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 $(k, dk) \leftarrow \mathrm{Gen}()$ that generates a secret key $k$.*
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.*
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 $\pi$. Note that $\mathrm{Del}$ 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 $\pi$ 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 $m$, if $\mathrm{Enc}_k(m) = (c, dk)$ then $\mathrm{Dec}_k(c) = m$. 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 $\mathrm{VerDel}_k(dk, \pi_k) = 1$.

For security, we first require perfect security for the encryption scheme, i.e. condition (**??**).[4] 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

[4] 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.

we introduce a security game. This game is of the same type as the one in Section 10.1.2, and once again is played between a honest *challenger* and a possibly malicious *adversary*. The idea is that a scheme will be called a *certified deletion* (encryption scheme) with security $\lambda$ if and only if no adversary can win in the game with probability much larger than $2^{-\lambda}$.

1 The challenger selects a key $k \leftarrow \mathrm{Gen}()$.
2 The adversary prepares an arbitrary quantum state $\rho_{ME}$, where $M$ is a classical register the size of a plaintext message and $E$ is a quantum register that the adversary keeps to itself. The adversary sends register $M$ to the challenger.
3 The challenger selects a $c \in \{0,1\}$ uniformly at random. If $c = 0$ then the challenger encrypts $M$:

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

and sends register $C$, now containing the (possibly quantum) ciphertext, to the adversary. The challenger keeps register $D$, which contains the (classical) deletion key $dk$. If $c = 1$ then the challenger 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 the adversary, keeping register $D$ as before.
4 The adversary sends a proof of deletion $\pi \in \{0,1\}^\lambda$ to the challenger.
5 The challenger sends the secret key $k$ to the adversary.
6 The adversary produces a guess $d \in \{0,1\}$.
7 The challenger declares that the adversary 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 10.1.2, there are two key differences. First, the adversary is asked for some additional information: at step 4, it has to return a "deletion proof" $\pi$. What we imagine here is that the challenger has asked for their ciphertext to be deleted, and the adversary is supposed to comply by sending the proof $\pi$, which is checked in the last step. Now, the validity of this proof is *supposed* to guarantee that the adversary has deleted the ciphertext. How do we check this? Here comes the second difference: at the next step, the challenger *reveals* the secret key $k$ to the challenger! We say that the adversary wins the game if, first of all its "deletion certificate" is accepted, and second it is able to discover which plaintext was encoded by the adversary. 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 $\pi$ 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 the adversary 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 10.2.1** 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.

### 10.2.2 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 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 to implement 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. By itself this solution 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 $\mathcal{I} \in \{1,\ldots,\lambda\}$ 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 $c' = m \oplus u \oplus_{i\in\mathcal{I}} x_i$ 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 $k = (u,\mathcal{I})$, $\mathrm{Dec}_k$ measures $|\phi\rangle$ in the standard basis to obtain a string $y$ and returns $m = c' \oplus u \oplus_{i\in\mathcal{I}} y_i$.

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 think through a given 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 here. For any message $m \in \{0,1\}$, according to item 2. the associated ciphertext takes the form $c = (c', |\phi\rangle)$ where $c' = m \oplus u \oplus_{i\in\mathcal{I}} x_i$ 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

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

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 $m' = m \oplus_{i\in\mathcal{I}} x_i$, 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 $\sigma_0$ represents a uniform mixture over all possible $|\phi\rangle$ (which you can check equals $2^{-\lambda}\mathbb{I}$, where the identity is over $\lambda$ 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.

## 10.2.3 Proof of certified deletion property

We first consider the case where $\lambda = 1$. Let's rewrite the security game from Section 10.2.1, when specialized to our scheme 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 The challenger selects $u \in \{0, 1\}$ and $\theta \in \{0, 1\}$ uniformly at random.
2 The challenger selects $c \in \{0, 1\}$ uniformly at random. They select $x \in \{0, 1\}$ uniformly at random and define $|\phi\rangle = |x\rangle_\theta$. If $\theta = 0$ the challenger sets $c' = c \oplus u \oplus x$. If $\theta = 1$ the challenger sets $c' = c \oplus u$. The challenger returns $c = (c', |\phi\rangle)$ to the adversary.
3 The adversary sends a deletion proof $\pi \in \{0, 1\}$ to the challenger.
4 The challenger sends $k = (u, \theta)$ to the adversary.
5 The adversary produces a guess $d \in \{0, 1\}$.
6 The challenger declares that the adversary has won if and only if $d = c$ *and* (if $\theta = 1$ then $\pi = x$).

In this description we did not let the adversary choose the plaintext $m$ and we also ignored the possibility for them 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 the adversary's power. More formally, if any adversary can succeed in the earlier security game with probability $\varepsilon$ then they can also succeed in this new security game with probability $\varepsilon$. 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 the adversary. First of all, we will only check the condition that $d = c$ in case when $\theta = 0$. Note that when $\theta = 0$, then the adversary receives $c' = c \oplus u \oplus x$, and they also receive $u$ at step 4. So the probability that they guess $c$ correctly is exactly the same as the probability that they guess $x$ (since they can convert from one to the other using $x = c \oplus (c' \oplus u)$, where they always have both $c'$ and $u$). So, in this step we replace item 6 by

6 The challenger declares that the adversary 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 the adversary. 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 The challenger selects $x \in \{0, 1\}$ and $\theta \in \{0, 1\}$ uniformly at random. They set $|\phi\rangle = |x\rangle_\theta$ and send $|\phi\rangle$ to the adversary.
2 The adversary sends a deletion proof $\pi \in \{0, 1\}$ to the challenger.
3 The challenger sends $\theta$ to the adversary.
4 The adversary produces a guess $d \in \{0, 1\}$.
5 The challenger declares that the adversary 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. In order 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 The adversary is split in two parts, $B$ and $E$. The adversary prepares an arbitrary state $\rho_{ABE}$ such that $A$ is a single qubit and sends $A$ to the challenger.
2 The challenger selects a $\theta \in \{0, 1\}$ uniformly at random. They measure $A$ in the basis $\theta$ to obtain an $x \in \{0, 1\}$.

3 $B$ sends $\pi \in \{0,1\}$ to the challenger.
4 The challenger sends $\theta$ to $E$, who responds with a $d \in \{0,1\}$.
5 The challenger declares that the adversary 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 the adversary could first prepare a state of the form $|\mathrm{EPR}\rangle_{AE} \otimes |0\rangle\langle 0|_B$ and send $A$ to the challenger; then, they would compute $\pi$ from $E$ and copy it to register $B$, and leave the post-measurement state in $E$ until they receive $\theta$. 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 to 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 relation in (**??**). This relation can be stated as follows. For any state $\rho_{ABE}$ where $A$ is a single qubit, it holds that

$$H_{\max}(X_A|B) + H_{\min}(Z_A|E) \geq 1 \ , \tag{10.4}$$

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 (10.4) 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 is not very small! In general we would like the adversary 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 $\lambda$ that can be larger than 1. From the point of view of the security game, considering higher parameters $\lambda$ is equivalent to performing a repetition of the case $\lambda = 1$ in parallel, multiple times. To analyze the game gor general $\lambda$ 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 $p_s^\lambda$ on the success probability in the $\lambda$-repeated game. This bound goes exponentially fast to zero with $\lambda$, and so by choosing $\lambda$ 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!

## 10.3 Chapter notes

The discussion in this chapter only scratches the surface of computational security, which is not a focus of this book.

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 [**?**].

A comparison of many a priori different definitions of computational security for quantum encryption can be found in [**?**].

The notion of encryption with certified deletion is studied in [**?**], where the protocol given here is adapted from.