Chapter 7

Qubit codes and cryptography

7.1 Qubit codes

According to the previous lecture, for a spin-$\frac{1}{2}$, any selection of the orientation $\vec{a}$ defines a particular code

$$b(|\uparrow_a\rangle) = \mathtt{0}\,, \qquad b(|\downarrow_a\rangle) = \mathtt{1}\,. \tag{7.1}$$

In contrast to the classical bit, the qubit allows for infinitely many different codes, each code being labeled by a spatial unit vector $\vec{a}$.

In quantum computation one usually works in a particular basis $\{|\mathtt{0}\rangle, |\mathtt{1}\rangle\}$, called the *computation basis*. A common assignment is given by $|\mathtt{0}\rangle = |\uparrow_z\rangle$, $|\mathtt{1}\rangle = |\downarrow_z\rangle$.

In quantum communication, however, the participating parties, that is sender, receiver, and eavesdropper, may use different sets of basis vectors. Using spherical coordinates to parametrize the unit vector $\vec{a}$ with respect to a fixed cartesian coor-

dinate system,

$$\vec{a} = (\cos\varphi \sin\vartheta, \sin\varphi \sin\vartheta, \cos\vartheta) \tag{7.2}$$

the expansion of the $\hat{\sigma}_a$-eigenvectors in terms of the $\hat{\sigma}_z$-eigenvectors, for example, reads

$$|\uparrow_a\rangle = \cos\tfrac{\vartheta}{2} e^{-i\frac{\varphi}{2}} |\uparrow_z\rangle , + \sin\tfrac{\vartheta}{2} e^{i\frac{\varphi}{2}} |\downarrow_z\rangle , \tag{7.3}$$

$$|\downarrow_a\rangle = -\sin\tfrac{\vartheta}{2} e^{-i\frac{\varphi}{2}} |\uparrow_z\rangle , + \cos\tfrac{\vartheta}{2} e^{i\frac{\varphi}{2}} |\downarrow_z\rangle , \tag{7.4}$$

with the inverse of this transformation given by

$$|\uparrow_z\rangle = \cos\tfrac{\vartheta}{2} e^{i\frac{\varphi}{2}} |\uparrow_a\rangle - \sin\tfrac{\vartheta}{2} e^{i\frac{\varphi}{2}} |\downarrow_a\rangle , \tag{7.5}$$

$$|\downarrow_z\rangle = \sin\tfrac{\vartheta}{2} e^{-i\frac{\varphi}{2}} |\uparrow_a\rangle , + \cos\tfrac{\vartheta}{2} e^{-i\frac{\varphi}{2}} |\downarrow_a\rangle . \tag{7.6}$$

Here the phase convention differs from the parametrization in (5.2). As this concerns only the *global phase*, but not the *relative phase*, this difference is not observable and hence irrelevant.

7.2 Quantum Cryptography

Given a *message*, say `10110`, and a *key*, say `11010`, one may *encrypt* the message by adding bits modulo 2. The resulting *cipher*, here `01100`, can not be *decrypted*, unless one knows the key. Thus, if both sender and receiver share a key which nobody else knows, they can communicate over a *public channel* without revealing any of their secrets to a potential eavesdropper.[1]

The critical point of this particular scheme, which is called *one-time pad* cryptography, is the distribution of the key. Key distribution was identified *the* major problem

[1] Uncoditional sucurity requires to use the key only once (for one message).

of cryptography when – at the dawn of the information age – privacy on public channels became a subject of major concern: for Alice and Bob at different locations, in order to establish the key, they must communicate plain text (the key) over a channel which could – or could not – be tabbed by an eavesdropper (public channel). Of course the mere possibility of tabbing is not the problem. The problem is that – according to the laws of classical physics – Alice and Bob have no possibility to find out whether there was an eavesdropper in the line or not.

In contemporary cryptography, like the RSA scheme, the problem of key distribution is avoided by making the key public. Cryptography in these schemes, which is called *public key cryptography*, relies on certain mathematical trap-door functions, like factoring of large numbers, which is easy in one direction (multiplication), but difficult in the other (factoring).

RSA scheme Alice thinks of a composite number $n = pq$ with p, q prime numbers. She finds d coprime to $(p-1)(q-1)$, and computes e from $ed = 1 \bmod (p-1)(q-1)$. Alice publishes e and n, but she keeps d (and of course p and q) private. If Bob wants to send her a message M (an integer number), he encrypts M using e and n as a key, $C = M^e \bmod n$. Upon receipt of the cryptogram C, Alice decrypts computing $D = C^d \bmod n$. By construction $D = M$.

The critical point of public key cryptography is the factoring. You, and I, and many more, may *find* factoring difficult, but there is no mathematical law – or natural law for that matter – which *proves* factoring to be difficult. It may well be that there is somebody out there, and he better does not reveal that he is out there, who knows how to compute the mathematical trap-door function "factoring" as easy as we know how to multiply.

In Quantum cryptography, instead of using a mathematical trap-door function and

a public key, one uses a physical law that prevents eavesdropping in distributing a private key over a public channel.

7.3 Quantum public key distribution (BB84-protocol)

[C. Bennett and G. Brassard, *Proceedings of the IEEE International Conference on Computer Systems and Signal Processing*, New York 1984, p. 175–179.]

Alice and Bob wish to share a secret, a random key, say. The key is a sufficiently long sequence of random bits. They go through the following 4-step protocol

1. Public gauge Alice and Bob erect a coordinate system. In particular they agree on what is meant by directions $\vec{e}_z$ and $\vec{e}_x$. They also set up their binary code

$$b(|\uparrow_z\rangle) = b(|\uparrow_x\rangle) = \mathtt{0}\,, \tag{7.7}$$
$$b(|\downarrow_z\rangle) = b(|\downarrow_x\rangle) = \mathtt{1}\,. \tag{7.8}$$

Note that both, the coordinate system and the code may well be public knowledge.

2. Distribution Alice generates of a random sequence of binary digits. She also generates of a random sequence of directions $\vec{e}_x$, $\vec{e}_z$. Alice prepares a sequence of qubits in states chosen according to the code from the set $\{|\uparrow_z\rangle, |\downarrow_z\rangle, |\uparrow_x\rangle, |\downarrow_x\rangle\}$. She sends the qubits to Bob. Bob, who does not know which particular direction Alice has chosen, measures the states of the qubits by choosing the alignment of his Stern-Gerlach device randomly but consciously either $\vec{e}_x$ or $\vec{e}_z$. He decodes his measurement results using the code (7.8).

3. **Public discussion** Bob broadcasts the alignments he has chosen for each qubit, without revealing the outcome of his measurements, however. Alice in turn broadcasts, for each qubit, whether Bob's alignment agrees with the alignment she used in the preparation. From their private records they discard all data where the alignments don't match. In a perfect world the remaining records of binary digits would agree one by one. However, the world is bad, that is an eavesdropper may have been in the line. In order to detect the eavesdropper, Alice and Bob go through a process which is called

4. **Authentication** Alice and Bob publish part of their remaining records. If they agree it means that with some high level of confidence, no eavesdropper has tried to intercept the distribution, and thus the rest of the records can be considered a secret only Alice and Bob share.

The merit of this protocol is that – in contrast to the classical physics situation – it allows to detect the eavesdropper. Eve, like Bob, does not know to a given qubit the orientation of Alice's SG device. With probability 1/2 she measures with the wrong alignment. If Bob measures with the correct alignment, the probability that Eve's qubit will end up in the wrong channel is 1/2. The probability to detect the eavesdropping on a single authenticate is therefore $\frac{1}{4}$. The probability *not* to detect an eavesdropper in N-bit authentication is therefore $\left(\frac{3}{4}\right)^N$, that is for sufficiently large N the eavesdropper will be detected with certainty.

Note that the secret Alice and Bob share in the end is neither know to Alice nor to Bob in advance. It "emerges" in course of the communication. Once the communication has passed the test for authentication, that is security has been established, the secret can be used to exchange other secrets.

No.	Alice			Bob		Res.	——Auth.——		Key
1	1	$\hat{\sigma}_x$	$\vert\downarrow_x\rangle$	$\hat{\sigma}_z$	1 (or 0)	-	-	-	-
2	0	$\hat{\sigma}_x$	$\vert\uparrow_x\rangle$	$\hat{\sigma}_x$	0	Ok	-	-	0
3	0	$\hat{\sigma}_z$	$\vert\uparrow_z\rangle$	$\hat{\sigma}_x$	1 (or 0)	-	-	-	-
4	0	$\hat{\sigma}_z$	$\vert\uparrow_z\rangle$	$\hat{\sigma}_z$	0	Ok	0	Ok	-

Figure 7.1: BB84 protocol example

7.4 B92 protocol

[C. H. Bennett, "Quantum Cryptography using any two nonorthogonal states", *Phys. Rev. Lett.* **68**(21), 3121–3124 (1992)]

In the wake of the BB84 protocol many more protocols for the quantum public key distribution have been proposed. Most important are the E91 protocol, a scheme proposed by Ekert in 1991 which exploits the peculiarities of quantum entanglement, and – in response to that scheme – a scheme proposed by Bennett in 1992, which is more in the spirit of BB84 but is based on fewer states.

[To be completed]