Chapter 1

Physical Security in Quantum Key Distribution

Contents

1.1 Cryptography in the Age of Quantum Computers

For a decade or two now, Quantum Computing has been creating a buzz that nobody in Computer Science and adjacent fields could evade. Originating in the 1980ies as a highly academic fusion applying concepts from Computer Science in Quantum Physics, its concepts have long found their way into popular science articles. Quantum Computing encompasses a model of computation that is fundamentally different from the *classical*[1] digital circuits that underly all of modern computing. While at first this might seem like a step backwards into the era of early 1900s analog computing, the capabilites of a future quantum computer promise to far outpace those of contemporary classical computers. Key to this improved processing capability is a property called *Quantum Parallelism*. What this refers to is the fact that a quantum computer's internal state can simultaneously represent a multitude of states of a classical, digital computer, and the quantum computer can operate on all those states at once using a single quantum operation.

Applying Quantum Parallelism to practical problems is far more complicated than, e.g., translating a digital circuit solving some equation to a quantum circuit, but for certain problems we already know *quantum algorithms* that for large inputs solve these problems much faster than any classical computer ever could. Two of these algorithms, one by Shor and one by Grover are what caused most of the buzz around the field of quantum computing, because they spell trouble for a large part of modern cryptography.

Besides the computational speed-up promised by Quantum Parallelism, there is one more interesting aspect of Quantum Computing where it radically deviates from classical computing. The reason modern cryptography exists is that when we transmit (or store!) classical information through some channel (or storage!) that we do not control, there is nothing we can do to prevent an attacker from reading this information. Even with cryptography we cannot prevent this, but cryptography gives us tools to very effectively make whatever information the attacker is able to read useless to them.

A basic principle of Quantum Physics is the *No-Cloning Theorem*, which states that it is impossible to create an identical, independent copy of an arbitrary, unknown quantum state. An implication of this theorem is that when we encode classical information into quantum states in just the right way, we can make it so that an attacker atttempting to eavesdrop on our quantum information can only actually read this information by destroying it in the process. This property can be exploited to replace a number of classical asymmetric primitives in interactive settings, the most popular application of which is replacing an asymmetric Diffie-Hellman key exchange with a quantum process called Quantum Key Distribution that yields much of the same properties.

In the past decades, the field of cryptography has been fundamentally shaped by the development of Quantum Computing and Quantum Key Distri-

[1] In Quantum Computing, the term *classical* is used as the complement of *quantum*, and refers to the digital computers we know and (sometimes) love. This terminology stems from the distinction between classical and quantum physics.

bution. However, the popular conception that all of today's cryptography will be broken and that we have to start from scratch is not accurate. Quantum Computing poses an unique threat to modern cryptography, and Quantum Key Distribution is a promising new tool, but the practical implications of both are much more subtle than how they may be portrayed. In the remainder of this chapter, we will look into the practical implications of these quantum technologies, and we will come to two major conclusions: First, that while the underlying cryptographic primitives will change, apart from some minor engineering issues cryptography as a whole will remain largely the same. Second, that while Quantum Key Distribution is hailed as a revolution for network security, its practical advantages will remain far short of how it is usually conceptualized, and hardware security will assume a pivotal role in the practical security of Quantum Key Distribution systems that is a stark departure from its relative irrelevance in today's applied cryptography.

Building on these conclusions, we will end this chapter with a study of a use case that illustrates a practical design for a secure network employing Quantum Key Distribution. Relying on both established classical and quantum primitives with known security properties we will elaborate how one can construct a large-scale network from those primitives that provides practical security to its users that goes beyond the (surprisingly limited) extents of quantum security proofs.

1.1.1 Computational Assumptions and Information-Theoretic Security

In the past paragraphs we have briefly mentioned that Quantum Computing provides a significant speed-up that can be applied to solve many cryptographic problems fast enough for it to become a problem, but we have not elaborated on what that means in practice. In this section, we will attempt to provide concrete numbers to quantify the threat that both Shor's and Grover's algorithm pose to modern cryptography.

Shor's algorithm allows for the factorization of large numbers in polynomial time on a quantum computer, a problem whose hardness (or the hardness of variants of which) is the foundation for the vast majority of today's asymmetric cryptography.

While Shor's algorithm attacks the foundations of most modern asymmetric cryptography, Grover's algorithm can be applied to hash functionss and symmetric cryptography. Fundamentally, Grover's algorithm is a search algorithm that allows a quantum computer to find one target entry out of an *unstructured* list of N source entries in $\mathcal{O}\left(\sqrt{N}\right)$ time instead of the $\mathcal{O}(N)$ time that a classical computer would require for an exhaustive search. Applied to cryptography, we model the key space of a symmetric cipher as the unstructured list that is input to the algorithm, and set it to search for the key that results in the successful decryption of a given ciphtertext.

An important nuance applying these algorithms to cryptography is that while both provide significant speed-ups over classical computers, the speed-up of Shor's algorithm is exponential and effectively breaks most modern

asymmetric cryptography as it erases the asymmetric nature of the underlying mathematical problem's computational complexity. That is, for an asymmetric cryptosystem susceptible to Shor's algorithm, there is no set of parameters that is large enough to be safe.

In contrast to this, while Grover's algorithm radically speeds up the breaking of a symmetric cryptosystem, this speed-up is only quadratic. In practice this means that it halves the security level of a given symmetric cipher. While this is bad news for applications that parameterize these symmetric primitives to a security level at the lower end of what is considered secure today, the advantage provided by Grover's algorithm can easily be compensated by doubling key size. Longer key sizes require more storage or bandwidth for the additional bits and result in slightly slower operation of the cipher, but this additional cost is easily manageable even without any improvement in today's hardware.

1.2 The Practical Security Implications of Quantum Computing

Given that as of yet, noone has claimed to have a quantum computer powerful enough to pose a threat to current cryptographic protocols, one may ask the fair question why the possible future development of such a machine would be consequential for today's cryptographic practice. The answer to this question lies in *Store-Now-Decrypt-Later* attacks. In such attacks, the attacker records all data transmitted between a cryptographic protocol's parties. The security of any key exchange protocol rests on a computational hardness assumption about some particular problem. When this assumption falls, for example because of a powerful quantum computer becoming available, the attacker can then retroactively break the security of those stored protocol instances and decrypt all traffic.

Modern cryptographic protocols such as TLS or the Signal messenger's key ratchet are designed with facilities to provide some degree of protection against key compromise called *(Perfect) Forward Secrecy*. Forward Secrecy means that a compromise of keys at one protocol step will not break the secrecy of past protocol steps. Forward Secrecy is achieved by repeatedly mixing fresh key material called *Ephemeral Keys* into the protocol's secret state. For a post-quantum attacker, this implies that to decrypt a run of a forward-secret cryptographic protocol, the quantum algorithm breaking the protocol's computational assumption must be run a number of times, but this results only in a linear increase of both protocol and attack complexity, which turns out to no advantage for the defender.

Store-Now-Decrypt-Later attacks are considered a serious threat today based on the stark discrepancy between the capacity of today's inexpensive storage media, and the comparatively tiny bandwidth of cryptographic protocols in applications such as *End-To-End-Encrypted* text messaging. A single hard drive can conceivably store years of a person's encrypted digital communications.

There has been ongoing work on quantum secure cryptographic algorithms, and standardization of several such algorithms is progressing. However, in the time frame of cryptosystems, these algorithms are still rather young and the recent discovery of a catastrophic key recovery attack against the Supersingular Isogeny Diffie-Hellman protocol (SIDH)[**hazay_efficient_2023**] illustrates the risk in the use of immature cryptographic primitives. Thus, recommendations on the concrete steps that should be taken today to mitigate Store-Now-Decrypt-Later attacks vary. For instance, Google's under its threat model as laid out in **schmieg_blog_2024** recommends a list of quantum secure counterparts to classically secure cryptographic algorithms, but recognizes the relative immaturity of these quantum secure algorithms and consequently recommends *Hybrid Deployment*, where a young, quantum secure algorithm is paired with a mature classically secure algorithm such that *both* algorithms would have to be broken to compromise the composite protocol's security. Given that quantum secure public key cryptography tends to have both a much larger key and/or ciphertext size and worse performance compared to state-of-the-art Elliptic Curve-based key exchange or signature algorithms, pairing it with a classically secure alternative incurs only a negligible overhead in key storage, network communication and computation costs.

1.3 The Physics of Quantum Computing

1.4 Quantum Key Distribution

As we discussed in Section 1.1.1, quantum computers promise novel attacks on many contemporary cryptographic systems. At the same time, quantum technology also promises new cryptographic primitives that support security guarantees beyond what can be realized with the best classical computers. The core of this nascent field of Quantum Cryptography is a set of methods that are collectively called Quantum Key Distribution.

Informally speaking, a Quantum Key Distribution system is a system that distributes a secret key between two[2] parties such that after a successful execution of the protocol, each of the two parties holds a copy of a randomly generated secret key, and the probability that an attacker was able to extract some portion of the key during the protocol's execution can be bounded to some negligible ϵ by each of the parties.

Quantum Key Distribution provides a similar service as cryptographic key exchange protocols such as the classic Diffie-Hellman key exchange provide. The core difference between QKD and cryptographic key exchange protocols is that QKD provides information-theoretic security based on the No-Cloning Theorem, where cryptographic protocols provide only computational security

[2]Although the key distribution problem can conceptually be framed for any number $n \geq 2$ of parties, practical treatment is almost always limited to the two-party case. In case of QKD, problem instances for $n > 2$ parties can trivially be reduced to $(n^2 - n)/2$ invocations of the two-party protocol, combined with any information-theoretically secure secret sharing scheme.

based on the computational hardness assumption underlying some public-key cryptosystem.

QKD is attractive in that it gives practically useful security guarantees without relying on any computational hardness assumptions. This way, QKD would remain secure even in a scenario where a hybrid deployment of a classically secure but mature algorithm paired with a quantum secure but young algorithm as discussed in Section poses too much of a risk—a scenario where both large quantum computers arrive and a flaw in the quantum secure algorithm is found. Note that here, because we assume we have large quantum computers, the possibility of a flaw in the quantum secure algorithm extends beyond mathematical flaws leading to practical attacks with classical computers, and includes novel quantum algorithms.

1.4.1 The Technical Implementation of QKD

On the technical level, QKD must be distinguished from general Quantum Computing. While QKD systems employ the No-Cloning Theorem and sometimes quantum entanglement in their operation, the scope of their quantum operations is very limited. QKD systems always operate on photons, while general quantum computers use a variety of physical implementations for their qubits that include photons and squeezed light, but extend over atom nuclei, trapped ions, various aspects of currents in superconducters into phonons[**berrios_high_2012**].

The central challenge in general quantum computers is extending the lifetime of the quantum state encoding a qubit. Quantum states are extremely sensitive to disturbances, and despite the best efforts to shield their quantum states against external influence, their lifetime is still inconveniently short compared to the timescales required for quantum computation, resulting in significant amounts of noise in the output of quantum algorithms run on contemporary quantum computers. Quantum Key Distribution systems use photons and only perform a handful of operations on each photonic state between generation and measurement, with the vast majority of the state's lifetime spent in transit between the two endpoints of the QKD protocol.

1.5 Quantum Networking

1.6 Securing QKD Networks with Inertial HSMs

As we discussed above, when it comes down to practical, end-to-end security properties, Quantum Key Distribution removes trust in the hardness of particular mathematical problems (good!), but increases trust in the physical integrity of the transceivers of the QKD link (bad!). In scenarios where the communicating parties are all located within physical proximity, in QKD meaning within at most a few hundred kilometers from each other depending on secret key rate requirements, this added trust is of no consequence because the communcating parties' hardware must be trusted in either QKD-assisted

or purely classical setups. However, this trust requirement becomes a burden as soon as at least one party is too far away (or higher secret key rates are required), as now physically trusted relays become necessary.

Extrapolating to practical deployments, we can make two predictions. First, as QKD only solves key distribution, but the actual data transfer still happens through normal off-the-shelf telecommunications components in QKD networks, there is no reason for a practical QKD setup to *not* also use classical cryptography as an additional layer for defense in depth, meaning the QKD setup will at worst degrade to the same security a purely classical system would provide, never less.

The second prediction we can make is that any practical QKD network will have to use trusted relays to bridge large distances. While in certain specialized applications such as the proposed financial QKD network in Swizerland smaller, isolated networks are conceivable, in every telecommunication system from the telegraph through the telephone system and up to the internet it has been shown conclusively that there is a real demand for a unified, global interconnected network.

In this section, we will outline a solution that provides practical, end-to-end security in large-scale QKD networks by delegating the hardware trust issue of QKD relays to Inertial Hardware Security Modules. The primary design challenges we will address are the systems' overall envelope design, optical passthroughs, and matching the cryptographic assumptions behind the IHSM's heartbeat and alarm subsystem to those of the QKD application.

1.7 Outlook