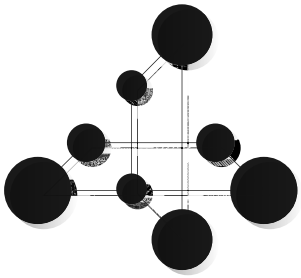
Linköping Studies in Science and Technology. Dissertations.
No. 1517

Authentication in Quantum Key Distribution: Security Proof and Universal Hash Functions

Aysajan Abidin
ئەيساجان ئابىدىن



INFORMATION CODING
Linköping University

Division of Information Coding
Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2013

Linköping Studies in Science and Technology. Dissertations.
No. 1517

**Authentication in Quantum Key Distribution:
Security Proof and Universal Hash Functions**

Aysajan Abidin
ئەيساجان ئابىدىن

aysajan@isy.liu.se
www.icg.isy.liu.se/en/
Division of Information Coding
Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

ISBN 978-91-7519-625-1 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2013

To my Mother, Guzelnur, and Ehsan

ئانامغا، گۈزەلنۇرغا ۋە ئېھسانغا بېغىشلاپ

Abstract

Quantum Key Distribution (QKD) is a secret key agreement technique that consists of two parts: quantum transmission and measurement on a quantum channel, and classical post-processing on a public communication channel. It enjoys provable unconditional security provided that the public communication channel is *immutable*. Otherwise, QKD is vulnerable to a man-in-the-middle attack. Immutable public communication channels, however, do not exist in practice. So we need to use authentication that implements the properties of an immutable channel as well as possible. One scheme that serves this purpose well is the Wegman-Carter authentication (WCA), which is built upon Almost Strongly Universal$_2$ (ASU$_2$) hashing. This scheme uses a new key in each authentication attempt to select a hash function from an ASU$_2$ family, which is then used to generate the authentication tag for a message.

The main focus of this dissertation is on authentication in the context of QKD. We study ASU$_2$ hash functions, security of QKD that employs a computationally secure authentication, and also security of authentication with a partially known key. Specifically, we study the following.

First, Universal hash functions and their constructions are reviewed, and as well as a new construction of ASU$_2$ hash functions is presented. Second, security of QKD that employs a specific computationally secure authentication is studied. We present detailed attacks on various practical implementations of QKD that employs this authentication. We also provide countermeasures and prove necessary and sufficient conditions for upgrading the security of the authentication to the level of unconditional security. Third, Universal hash function based multiple authentication is studied. This uses a fixed ASU$_2$ hash function followed by one-time pad encryption, to keep the hash function secret. We show that the one-time pad is necessary in every round for the authentication to be unconditionally secure. Lastly, we study security of the WCA scheme, in the case of a partially known authentication key. Here we prove tight information-theoretic security bounds and also analyse security using witness indistinguishability as used in the Universal Composability framework.

Populärvetenskaplig sammanfattning

Risken för illegal avlyssning av information, till exempel vid penningtransaktioner, tvingar fram allt mer avancerade tekniker för kryptering. När man skickar krypterade meddelanden via datornätverk är ett svårlöst problem hur kryptonyckeln ska överföras. Ett sätt är att skicka den med kurir (vanlig post eller, som i agentfilmer, en person med attachéväska fastlåst vid handleden). En kurir måste förstås vara pålitlig, annars finns risken att nyckeln omärkligt kopieras på vägen. En annan teknik är så kallad öppen-nyckel-överföring som används för Internetbank och säkerhetsfunktioner i webbläsare (https). Öppen-nyckel-överföring anses säker, eftersom det krävs stora beräkningar för att knäcka de långa strängar av databitar (omkring 3 000) som nyckeln består av.

Det finns en ny teknik för att överföra nyckeln som kallas kvantkryptografi där säkerheten garanteras av kvantmekaniska naturlagar. Än så länge är det dock mycket få som använder den. Det behövs en speciell hårdvara med, till exempel, en typ av laser som sänder ut enstaka polariserade ljuspartiklar (fotoner) via optisk fiber eller genom luften. Några företag och banker i österrike provar systemet och försök pågår med satellitt-överföring. Säkerheten garanteras eftersom kvantmekaniska objekt har den mystiska egenheten att de inte tål att mätas eller manipuleras utan att förändras. Om någon försöker kopiera en kvantmekaniskt kodad nyckel på vägen, så kommer det att märkas i form av brus. En avlyssnare kan ställa till problem, men inte få ut någon användbar information utan att det märks.

Denna avhandling handlar om den del av ett kvantkryptosystem som ska se till att man överför nyckeln till rätt person, autentiseringsdelen. Två frågor behandlas: dels hur man gör systemet effektivare, så att man får ut mer kryptonyckel ur det, och dels hur man hanterar små informationsläckor som inte kan undvikas i systemet. Det senare behövs eftersom man inte kan undvika brus helt och hållet, då kvantmekaniska objekt är mycket känsliga. Man måste därför ta hänsyn till bruset, och räkna med ett (mycket) litet informationsläckage. Avhandlingen behandlar flera system tänkta att öka nyckelproduktionen, ger svar på frågor om hur de olika varianter som finns av kvantkryptografi kan hantera informationsläckage, och även råd om säker användning av systemen.

Acknowledgments

"I like prefaces. I read them. Sometimes I do not read any further."
– Malcolm Lowry

This dissertation accounts for two years of full time research. Part of the works in this dissertation was done when I was a PhD student at Mathematics Department, where I received my Licentiate degree. Then I joined the Information Coding division at Electrical Engineering department, where the main results in this dissertation are obtained. This work would not have been possible without the help of many people around me. Here I would like to grab this opportunity to thank them from the bottom of my heart.

First of all, I would like to thank my supervisor, Associate Professor Jan-Åke Larsson, for introducing me to the field of authentication and Quantum Cryptography. I especially thank him for taking me as his PhD student even though I knew nothing about cryptography when I started. He was always patient, supportive, and encouraging. His door was always wide open for both scientific and non-scientific discussions.

Special thanks go to my co-supervisor, Professor emeritus Viiveke Fåk, and Professor Robert Forchheimer for their help and counsel. I would also like to extend my appreciation to Associate Professor Bengt-Ove Turesson at the Mathematics Department for his help during my early years as a PhD student.

I certainly do not forget to thank my colleagues and friends at the math department and in the division of Information Coding at Electrical Engineering department, who created such a nice and friendly working atmosphere. It is my pleasure to get to know you all.

I am deeply thankful to my mother, sisters, and brother for their unconditional love. For me, you are the source of endless encouragement and support. I cannot thank you all enough, but I can surely say this: Without you and your love, I would never have been able to come this far to realise what used to be one of my late father's dreams.

Last but certainly not least, I am very grateful to my lovely wife, Güzelnur, and awesome son, Ehsan, for all the love, joy and happiness they give me every day. I love you both very much.

مەن بۇ پۈرسەتتىن پايدىلىنىپ مېھرىبان ئانامنىڭ مېنى تەربىيلەپ قاتارغا قوشۇش جەريانىدا ماڭا سىڭدۈرگەن چەكسىز ئەجرىگە ۋە ماڭا بەرگەن ئالەمچە مېھرىمۇھەببىتىگە چىن كۆڭلۈمدىن چوڭقۇر تەشەككۈر ئېيتىمەن. ئۇلۇغ ئاللاھدىن تېنىڭىزگە سالامەتلىك ۋە ئۇزۇن ئۆمۈر تىلەيمەن.

Linköping, April 10, 2013
Aysajan Abidin

ئەيساجان ئابدىن

Contents

1

Introduction and Outline

When two parties, which have not had previous contact and are separated far from each other in space, want to communicate digital messages with each other secretly, it is impossible for them to achieve this without sharing a string of secret bits. They need secret keys to be able to encrypt their messages to each other. They can either use a courier to send the secret key, or meet in person to exchange keys so that they can send secret messages to each other later on. Both of these are time-consuming and expensive.

In cryptography, it is a tradition to refer to the two parties as Alice and Bob. Also, Eve is the traditional name used to refer to the eavesdropper or the adversary who wants to eavesdrop on Alice and Bob's communication. The same tradition is followed in this dissertation. So from now on, the names, Alice, Bob, and Eve, will appear frequently in the rest of this dissertation.

Alice and Bob need a less time-consuming and less expensive way of sharing a secret, which they can use to exchange secret messages with each other. Public Key Cryptography (PKC) offers one solution to this problem. PKC schemes are based on computationally hard[1] problems in number theory such as prime factoring (as in RSA), solving discrete logarithm problems (equivalently known as Diffie-Hellman problem) and so on. The security of these systems, however, is solely built on the (unproven) assumptions that the above mentioned problems are computationally hard to solve using classical computers.

There are, however, quantum algorithms such as Shor's algorithm [1], which can be applied to solve the factoring problems and discrete logarithm problems efficiently (with polynomial effort) on a quantum computer. This implies that quantum computers, if ever built, can be used to break RSA or Diffie-Hellman cryptosystems. Therefore, there is a need for secret key agreement schemes that can resist attacks enabled by sudden advances in computing technologies, both classical and quantum, and that are also preferably prov-

[1] Here computationally hard means that the best algorithm for a problem depends exponentially, in time, on the input size.

ably secure.

Quantum Key Distribution (QKD) is a provably unconditionally secure secret key agreement technique based on the laws of quantum mechanics. It consists of two parts: quantum transmission and measurement on a quantum channel, and post-processing on a public communication channel. Since its introduction by Bennett and Brassard [2] in 1984 (BB84) and independently by Ekert [3] in 1991, QKD has been widely studied, and enormous theoretical and technological advances have been made, which led to commercial QKD products manufactured by, for example, IDQuantique, based in Geneva. The most attractive property of QKD is its provable security without any assumptions on the adversary's computational power and storage capability.

In QKD, Alice and Bob first exchange quantum signals over the quantum channel to generate a raw key. Then, they agree on a shared secret key from the raw key by performing a joint post-processing of the raw key by communicating on the public channel. Chapter 2 presents a more detailed introduction to QKD.

Like any key exchange protocol, however, the security of QKD requires that the public channel is *immutable*; see, for example, [4]. If the public channel is not immutable, QKD can trivially be broken by a man-in-the-middle (MITM) attack. Therefore, an immutable public communication channel is a must. More specifically, we must guarantee that the adversary can not insert or modify the (classical) messages exchanged over the public channel between Alice and Bob during the post-processing phase of the QKD protocol. Immutable channels, however, do not exist in practice. We can only use authentication schemes that emulate the properties of an immutable channel as close as possible. QKD requires such schemes to be information-theoretically secure (ITS), since QKD itself is intended to be ITS. One such scheme is the well-known Wegman-Carter authentication (WCA). It is based on Almost Strongly Universal$_2$ (ASU$_2$) hashing and is the standard in QKD. We will give more details on WCA in Chapter 2.

In a QKD system, to be able to achieve authentication Alice and Bob preshare a secret key that is long enough for authentication purposes in the initial QKD round. This means that Alice and Bob have to meet in person before they start using QKD to exchange this shared secret. In the subsequent QKD rounds, Alice and Bob use a portion of the key generated by QKD in the present (or the previous) round. This has two consequences. One is that reserving a portion of the generated key for authentication means that the key-consumption rate of authentication directly influences the key output rate of QKD. And the other is that Alice and Bob no longer use perfectly secret keys for authentication. This is because, even though QKD is ITS, keys generated by QKD are *not* perfectly secret. In fact, QKD generated keys are ε_{Q}-perfect [5, 6], meaning that the distribution of the key, from Eve's point of view, has at most ε_{Q} trace distance to the uniform. Therefore, it is quite relevant and important to study security of authentication with a partially known (or imperfect) key and of QKD itself that employs low key-consuming authentication schemes.

The main contribution of this dissertation relates, directly or indirectly, to ASU$_2$ hash functions, security of authentication with a partially known key, and security of QKD that employs a non-ITS authentication.

Outline

The structure of the thesis is as follows. In Part I, we present some necessary background. In particular, we briefly explain how QKD works, what is authentication and why it is important in QKD, which type of (classical) authentication is used in QKD, and present the definitions of Universal hash functions and their use in the construction of unconditionally secure message authentication codes.

Part II presents the main contributions of this dissertation in separate chapters. In particular, we first review Universal hash functions and present a new construction of ASU_2 hash functions, with fixed security parameter and relatively short key length. These two properties make the new construction suitable for authentication purposes in QKD. Here, we also review other known constructions of ASU_2 hash functions and compare them with each other. For details, see Chapter 3.

Second, security of QKD when it employs a specific computationally secure two-step authentication protocol is studied. Here, we present an attack strategy and detailed attacks on a practical implementation of QKD that employs the two-step authentication to show how an adversary can break the QKD system. We also present countermeasures that increase the adversary's demand for computational power, and prove necessary and sufficient conditions for upgrading the security of the two-step authentication to the level of unconditional security. We present these in Chapter 4.

Third, Universal hash function based multiple authentication is also studied. Here, we show that the one-time pad is necessary in every round for the authentication to be unconditionally secure. We devote Chapter 5 to this.

Fourth, we study security of the WCA scheme, in the case of a partially known authentication key. The partial knowledge of the attacker is measured as the trace distance between the authentication key distribution and the uniform distribution; this is the usual measure in QKD. We first analyse information-theoretic security and prove tight security bounds, then analyse security using witness indistinguishability. This is covered in Chapter 6.

Finally, Chapter 7 ends Part II with a conclusion of the whole thesis and with a short discussion of some possible future extensions.

Part III consists of five papers, details of which are given below.

Included papers

Whenever needed, the following papers will be referred to by their corresponding Roman numerals throughout the thesis. The contribution statements refer to the contributions the thesis author have made to the papers.

Paper I: New universal hash functions

This paper is published in WEWoRC 2011, Lecture Notes in Computer Science, Vol. 7242, pp. 99-108, Springer Verlag, 2012.

Authors: Aysajan Abidin and Jan-Åke Larsson

Abstract: Universal hash functions are important building blocks for unconditionally secure message authentication codes. In this paper, we present a new construction of a class of ε-Almost Strongly Universal$_2$ hash functions with much smaller description (or key) length than the Wegman-Carter construction. Unlike some other constructions, our new construction has a very short key length and a security parameter ε that is independent of the message length, which makes it suitable for authentication in practical applications such as Quantum Cryptography.

Contributions: The thesis author constructed the hash function family reported in this paper, and calculated its key consumption, and also wrote most of the paper. The comparisons in terms of performance and key consumption is joint work between the authors of this paper.

Paper II: Vulnerability of "A novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography"

This is published in International Journal of Quantum Information, Vol. 7(5), pages 1047-1052, 2009.

Authors: Aysajan Abidin and Jan-Åke Larsson

Abstract: In this paper we review and comment on “A novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography”, [M. Peev *et al*., *Int. J. Quant. Inform.*, **3**, 225, (2005)]. In particular, we point out that the proposed primitive is not secure when used in a generic protocol, and needs additional authenticating properties of the surrounding quantum-cryptographic protocol.

Contributions: This paper is a result of a discussion between the authors of the paper on what attacks are possible in quantum key distribution. After this, the task of the thesis author was to find countermeasures, and also to write most of the paper.

Paper III: Attacks on quantum key distribution protocols that employ non-ITS authentication

This paper is submitted and available in arXiv as arXiv:1209.0365.

Authors: Christoph Pacher, Aysajan Abidin, Thomas Lorünser, Momtchil Peev, Rupert Ursin, Anton Zeilinger, and Jan-Åke Larsson

Abstract: We demonstrate how adversaries with unbounded computing resources can break QKD protocols which employ a particular message authentication code suggested previously. This authentication code, featuring low key consumption, is not information-theoretically secure (ITS) since for each message the eavesdropper has intercepted she is able to send a different message from a set of messages that she can calculate by finding collisions of a cryptographic hash function. However, when this authentication code was introduced it was shown to prevent straightforward man-in-the-middle (MITM) attacks against QKD protocols.

In this paper, we prove that the set of messages that collide with any given message under this authentication code contains with high probability a message that has small Hamming distance to any other given message. Based on this fact we present extended

MITM attacks against different versions of BB84 QKD protocols using the addressed authentication code; for three protocols we describe every single action taken by the adversary. For all protocols the adversary can obtain complete knowledge of the key, and for most protocols her success probability in doing so approaches unity.

Since the attacks work against all authentication methods which allow to calculate colliding messages, the underlying building blocks of the presented attacks expose the potential pitfalls arising as a consequence of non-ITS authentication in QKD-postprocessing. We propose countermeasures, increasing the eavesdroppers demand for computational power, and also prove necessary and sufficient conditions for upgrading the discussed authentication code to the ITS level.

Contributions: The attacks in this paper utilize the idea in Paper II. Here, the thesis author expanded and formalized the countermeasures and proved security when using them. He also wrote most of the text concerning countermeasures.

Paper IV: On security of universal hash function based multiple authentication

This article is published in ICICS 2012, Lecture Notes in Computer Science, Vol. 7618, pp. 303-310, Springer Verlag, 2012.

Authors: Aysajan Abidin

Abstract: Universal hash function based multiple authentication was originally proposed by Wegman and Carter in 1981. In this authentication, a series of messages are authenticated by first hashing each message by a fixed (almost) strongly universal$_2$ hash function and then encrypting the hash value with a preshared one-time pad. This authentication is unconditionally secure. In this paper, we show that the unconditional security cannot be guaranteed if the hash function output for the first message is not encrypted, as remarked in [19]. This means that it is not only sufficient, but also necessary, to encrypt the hash of every message to be authenticated in order to have unconditional security. The security loss is demonstrated by a simple existential forgery attack. The impact of the attack is also discussed at the end.

Contributions: This paper is written by the thesis author alone. He found the attack in the paper and wrote the paper.

Paper V: Direct proof of security of Wegman-Carter authentication with partially known key

This paper is submitted and available in arXiv as arXiv:1303.0210.

Authors: Aysajan Abidin and Jan-Åke Larsson

Abstract: ITS authentication is needed in QKD. In this paper, we study security of an ITS authentication scheme proposed by Wegman&Carter, in the case of partially known authentication key. This scheme uses a new authentication key in each authentication attempt, to select a hash function from an Almost Strongly Universal$_2$ hash function family. The partial knowledge of the attacker is measured as the trace distance between the authentication key distribution and the uniform distribution; this is the usual measure in

QKD. We provide direct proofs of security of the scheme, when using partially known key, first in the information-theoretic setting and then in terms of witness indistinguishability as used in the Universal Composability (UC) framework. We find that if the authentication procedure has a failure probability ε and the authentication key has an ε' trace distance to the uniform, then under ITS, the adversary's success probability conditioned on an authentic message-tag pair is only bounded by $\varepsilon + |\mathcal{T}|\varepsilon'$, where $|\mathcal{T}|$ is the size of the set of tags. Furthermore, the trace distance between the authentication key distribution and the uniform increases to $|\mathcal{T}|\varepsilon'$ after having seen an authentic message-tag pair. Despite this, we are able to prove directly that the authenticated channel is indistinguishable from an (ideal) authentic channel (the desired functionality), except with probability less than $\varepsilon+\varepsilon'$. This proves that the scheme is $(\varepsilon+\varepsilon')$-UC-secure, without using the composability theorem.

Contributions: In this paper, the technical results are largely the thesis author's, and for some details aided by the supervisor. One of the major tasks was to understand the composability framework, as used in the last part of the paper, and apply it to the authentication protocol that was studied in this paper. The thesis author also wrote most of the paper.

Not included papers

The following papers are not included in the thesis.

> Aysajan Abidin and Jan-Åke Larsson, "*Special properties of strongly universal$_2$ hash functions important in quantum cryptography,*" AIP Conference Proceedings, Vol. 1101, pages 289-293, 2009.
>
> Aysajan Abidin, Christoph Pacher, Thomas Lorünser, Jan-Åke Larsson, and Momtchil Peev, "*Quantum cryptography and authentication with low key-consumption,*" Proc. of SPIE Vol. 8189, 818916-, 2011.
>
> Aysajan Abidin and Jan-Åke Larsson, "*Security of authentication with a fixed key in quantum key distribution,*" *arXiv:1109.5168*, 2011.

Part I

Background

2

Background

QKD is an elegant use of quantum mechanics in secure key distribution, and is one application of quantum physics at the individual quanta level [7]. Keys generated from QKD are secret, or more accurately ε-perfect (cf. Definition 2.8), provided that the communication between Alice and Bob is authentic [4–6]. This chapter briefly explains how QKD works, why do we need authentication in QKD, and what type of (classical) authentication is used in QKD. Also, some definitions that are used throughout this thesis are given.

2.1 Quantum Key Distribution

We focus on the BB84 protocol [2] which at a high level consists of two main steps: quantum transmission on the quantum channel and classical post-processing on the public communication channel. The first step generates two weakly correlated and partially secret raw keys, one for Alice and one for Bob. The second step transforms the raw keys into an identical, highly secret key pair, by classical post-processing. This second step, namely, the classical post-processing, further consists of sifting, error estimation and reconciliation, privacy amplification, and authentication.

There are other QKD protocols as well, such as Ekert's entanglement based E91 protocol [3] and SARG04 protocol by Scarani *et al.* [8]. They also consist of those two steps at a high level, but differ, among others, in the way the quantum particles or photons are prepared and transmitted. An explanation of any one of those QKD schemes will suffice to understand the basic idea behind QKD. So we choose the BB84 protocol, which is the first proposed QKD scheme.

The following is a brief explanation of the five steps of the BB84 protocol; see [9–13] for more detailed explanations. More details will be given later.

(1) Raw key generation: Alice sends a series of single photons each modulated in a random basis, either in rectilinear basis of vertical and horizontal, or diagonal basis

of $45°$ and $135°$, with a random value 0 or 1 to Bob. For example, in the rectilinear basis 0 is encoded as a horizontal state and 1 as a vertical state, and in the diagonal basis 0 is encoded as a $45°$ state and 1 as a $135°$ state. Bob chooses his measurement basis randomly and independently from Alice and reads the values. Then he sends Alice an authenticated time stamp to end the quantum transmission. Now they have two random bit sequences called raw keys, of which on average 75% is the same.

(2) Sifting: After the quantum transmission is over, Bob publicly announces his measurement basis, but *not* his measurement results, to Alice, and Alice responds to him with a message saying which bases do not match each other. Then they discard all cases where Bob chose a different basis: This is called sifting. They now have two almost identical smaller keys, of which Eve perhaps has some knowledge. Note that sifting can be done by Alice publicly announcing her preparation basis to Bob and then Bob responding to her with a message saying which bases do not match each other. This does not have any effect on the sifted key.

(3) Error estimation and reconciliation: To reconcile the two almost identical sifted keys, Alice sends error-correction information (random maps and the output values) to Bob, and error-corrects the sifted key that she shares with Bob. Bob responds by a message that signals which subsets matched and which subsets were successfully error-corrected, and also indicates the error rate of the sifted key; in simple schemes this can be used as error estimate.

(4) Privacy amplification: It is possible that some information is leaked to Eve during error correction. Therefore, to further increase the secrecy of the error corrected keys, Alice and Bob perform privacy amplification. This is done by Alice choosing a random map, and sending that over the classical channel, whereafter Alice and Bob apply this map to their respective reconciled keys. It is important to note in here that Eve's information on the key after privacy amplification is *not* reduced all the way to zero, but it is very small.

(5) Authentication: It is crucial for security to authenticate some (or all) of the classical messages communicated during the public discussion. We will explain why this is the case, how authentication is achieved and what type of authentication must be used in the following sections.

These are the five major steps of a typical BB84 type of QKD protocol. As we can see, except for the raw key generation, all the other steps are performed on the public communication channel, see Figure 2.1. This tells us how important the public channel is. As we shall see later, it is vital for the security of QKD to guarantee the authenticity of the messages exchanged on the public channel.

Before we move on to the next section, let us look at the above mentioned steps of a typical BB84 protocol in a bit more detail. In Chapter 4, we will demonstrate an attack on this protocol when it employs a specific low key-consuming non-ITS authentication scheme. In the description below, g_k denotes a keyed hash function identified by the key k and $g_k(m)$ the computation of the authentication tag for a message m.

(1) Raw key generation:

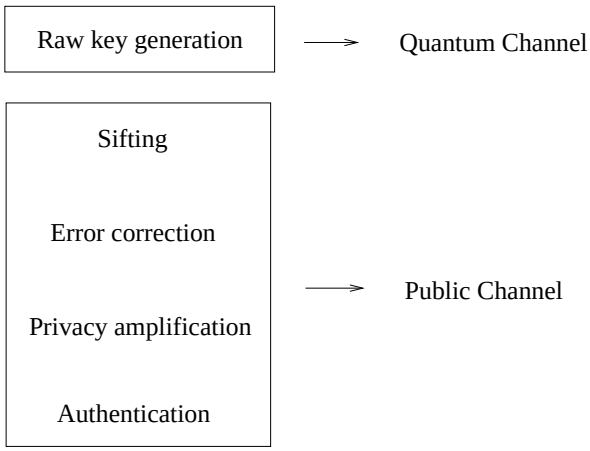


Figure 2.1: *QKD as a whole.*

(1a) Alice first prepares two random bitstrings of length N: $d^{\mathbf{A}}$ (data bits), and $b^{\mathbf{A}}$ (encoding bases, rectilinear or diagonal). For each pair $\left(d_i^{\mathbf{A}}, b_i^{\mathbf{A}}\right)$, she generates the corresponding quantum state $\rho_i^{\mathbf{A}}$. Here and below in the description, $i = 1, 2, \cdots, N$. She then sends the quantum state $\rho^{\mathbf{A}} = \bigotimes_{i=1}^{N} \rho_i^{\mathbf{A}}$ ("string" of all $\rho_i^{\mathbf{A}}$'s) to Bob.

(1b) Bob selects a string of random measurement bases of length N, $b^{\mathbf{B}}$. He then measures $\rho^{\mathbf{A}}$ in these bases $b^{\mathbf{B}}$ and obtains measurement outcomes $d^{\mathbf{B}} \in \{0, 1, \perp\}^N$. Here $d_i^{\mathbf{B}} = \perp$ denotes that Bob's detectors did not produce a measurement result (e.g. due to absorption in the channel, imperfection of the detectors, and etc.). After the measurement, he sends an acknowledgment message with tag, i.e. $\left(\mathrm{m}_{\mathrm{ack}}, g_{k_1}(\mathrm{m}_{\mathrm{ack}})\right)$, to Alice to end the quantum transmission phase.

(2) Sifting:

(2a) After receiving the acknowledgement message from Bob, Alice sends $\left(b^{\mathbf{A}}, g_{k_2}(b^{\mathbf{A}})\right)$ to Bob.

(2b) Bob calculates a bitstring $b^{\mathbf{A=B}}$, such that $b_i^{\mathbf{A=B}} = 1$ if he and Alice prepared and measured $\rho_i^{\mathbf{A}}$ in the same basis, and $b_i^{\mathbf{A=B}} = 0$ otherwise or if he did not measure anything. He removes from $d^{\mathbf{B}}$ all bits $d_i^{\mathbf{B}}$ where $b_i^{\mathbf{A=B}} = 0$ and obtains his sifted key $s^{\mathbf{B}}$. He then sends $\left(b^{\mathbf{A=B}}, g_{k_3}(b^{\mathbf{A=B}})\right)$ to Alice.

(2c) Upon receiving $b^{\mathbf{A=B}}$ from Bob, Alice removes from $d^{\mathbf{A}}$ all bits $d_i^{\mathbf{A}}$ where $b_i^{\mathbf{A=B}} = 0$ and obtains her sifted key $s^{\mathbf{A}}$.

(3) Error estimation and reconciliation:

(3a) Alice estimates the error rate of the quantum channel, selects a forward error correction algorithm ECA and calculates a (binary) bitstring of parity information, $p(s^{\mathbf{A}})$. She then determines a confirmation function CO, calculates $CO(s^{\mathbf{A}})$, and sends $\big(T := (\text{ECA}, p(s^{\mathbf{A}}), CO, CO(s^{\mathbf{A}})), g_{k_4}(T)\big)$ to Bob.

(3b) Bob uses $p(s^{\mathbf{A}})$ to correct $s^{\mathbf{B}}$ and obtains $\hat{s}^{\mathbf{B}}$. He then uses CO to calculate $CO(\hat{s}^{\mathbf{B}})$. He further checks whether $CO(\hat{s}^{\mathbf{B}}) = CO(s^{\mathbf{A}})$. If they are identical, he calculates the error rate ε and sends $\big(\varepsilon, g_{k_5}(\varepsilon)\big)$ to Alice; if not, he sends $\big(\textit{fail}, g_{k_5}(\textit{fail})\big)$ to Alice and aborts the protocol.

(4) Privacy amplification:

(4a) If Alice receives ε, then she determines a privacy function P, sends $\big(P, g_{k_6}(P)\big)$ to Bob, and calculates her final key $K^{\mathbf{A}} = P(s^{\mathbf{A}})$. If she receives *fail* instead, she aborts the protocol.

(4b) If Bob has not aborted in step (3b) and has received P from Alice, then he calculates his final key $K^{\mathbf{B}} = P(\hat{s}^{\mathbf{B}})$. The final keys $K^{\mathbf{A}}$ and $K^{\mathbf{B}}$ are identical with overwhelming probability.

Note that in practice when a message is received, the receiver checks whether it is authentic. If the message is authentic, then he/she continues the next protocol step; otherwise, aborts the protocol. In the above description, we implicitly assumed that all received messages pass the authenticity test and thus skipped it in the description.

Next, we will discuss why authentication is important in QKD.

2.2 Authentication

Authentication is one of the important main topics within cryptography. There are different types of authentication such as, user authentication, message authentication and so on. In this thesis, we exclusively deal with message authentication (MA). The goal of MA is to provide Alice and Bob with a means to make sure that received messages originated from the other participant. In particular, MA allows Alice and Bob to send each other messages in such a way that any modification of them can be detected with very high probability. In the following subsections, we first look at why we need authentication in QKD, then discuss what type of authentication is used in QKD.

2.2.1 The necessity of authentication in QKD

QKD without immutable public channel, or in practice QKD without authentication, is like any other key exchange or distribution schemes susceptible to a man-in-the-middle (MITM) attack. In particular, in an MITM attack on a QKD protocol, Eve first cuts the quantum and the public channels and connects the loose ends to her QKD devices identical to Alice and Bob's QKD devices. Then she impersonates Bob to Alice and Alice to Bob during the quantum transmission process and the subsequent public discussions (or the post-processing), see Figure 2.2. For the attack to be successful Eve needs, among

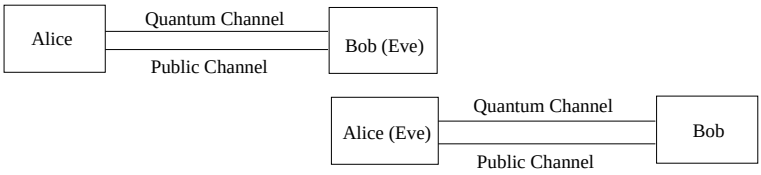


Figure 2.2: *Man-in-the-middle (MITM) attack on QKD.*

other things, to substitute the classical message from one legitimate user (Alice) to the other (Bob) without being noticed. Eve can do this without being noticed if the messages exchanged over the public communication channel are not authenticated. To prohibit such an attack on QKD, proper message authentication is needed. Therefore, message authentication is a must for QKD to be secure.

In QKD, one can use either immediate authentication or delayed authentication. In the former, messages are authenticated immediately after they are received, as in the protocol description in the previous section, while in the latter, all messages exchanged during one whole session are authenticated together at the end of the session. As to which messages in the post-processing phase to authenticate, see for instance [13].

2.2.2 The choice of authentication in QKD

There are various types of message authentication (MA), or message authentication codes (MACs). According to their security, we can divide MACs into two categories: one is information-theoretically secure (ITS) MACs and one is computationally secure MACs. The former is also called unconditionally secure MACs and is built upon a special family of keyed hash functions called Almost Strongly Universal$_2$ (ASU$_2$) hashing; see the next section for details. The latter is based on cryptographic hash functions such as **SHA1**, **SHA2**, **SHA3**, **RIPEMD**, and so on. The security of the latter MACs is based on the computational complexity of finding collisions for the cryptographic hash function involved. Since QKD is intended to be provably unconditionally secure, ITS MACs are naturally the standard choice of authentication in QKD. In Chapter 4, we will see how an MITM attack can break QKD protocols that employ a non-ITS authentication. So here, we specially focus on MACs that are ITS.

We next present definitions of Universal hash functions that are the building blocks of an ITS MAC scheme known as the Wegman-Carter authentication (WCA).

2.3 Universal hash functions

Since its first introduction by Wegman and Carter [14] in 1979, Universal hash functions have been extensively studied over the years [15–25]. They have diverse applications from cryptography to computer science to coding theory. In cryptography, they can be used for, among others, constructing unconditionally secure MACs (the WCA). They can also be used for error-correction and privacy amplifications in QKD [10–12, 17, 26].

Definitions

In what follows, $\mathcal{M}$ and $\mathcal{T}$ denote finite sets of messages and tags, respectively, and the size $|\mathcal{M}|$ of $\mathcal{M}$ is greater than or equal to the size $|\mathcal{T}|$ of $\mathcal{T}$. The set of hash functions from $\mathcal{M}$ to $\mathcal{T}$ is denoted as $\mathcal{H}$.

Definition 2.1 (Universal$_2$ hash functions). Let $\mathcal{M}$ and $\mathcal{T}$ be finite sets. A class $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$ is Universal$_2$ (U$_2$) if there exist at most $|\mathcal{H}|/|\mathcal{T}|$ hash functions $h \in \mathcal{H}$ such that $h(m_1) = h(m_2)$ for any two distinct $m_1, m_2 \in \mathcal{M}$.

If there are at most $\varepsilon|\mathcal{H}|$ hash functions instead, for $\varepsilon > 1/|\mathcal{T}|$, the class $\mathcal{H}$ is ε-Almost Universal$_2$ (ε-AU$_2$).

Definition 2.2 (XOR Universal$_2$ hash functions). Let $\mathcal{M}$ and $\mathcal{T}$ be as before. A class $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$ is XOR Universal$_2$ (XU$_2$) if there exists at most $|\mathcal{H}|/|\mathcal{T}|$ hash functions $h \in \mathcal{H}$ such that $h(m_1) = h(m_2) \oplus t$ for any two distinct $m_1, m_2 \in \mathcal{M}$ and any $t \in \mathcal{T}$.

If there are at most $\varepsilon|\mathcal{H}|$ hash functions instead, for $\varepsilon > 1/|\mathcal{T}|$, the class $\mathcal{H}$ is ε-Almost XOR Universal$_2$ (ε-AXU$_2$).

Definition 2.3 (Strongly Universal$_2$ hash functions). Let $\mathcal{M}$ and $\mathcal{T}$ be as before. A class $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$ is Strongly Universal$_2$ (SU$_2$) if the following two conditions are satisfied:

(a) The number of hash functions in $\mathcal{H}$ that takes an arbitrary $m_1 \in \mathcal{M}$ to an arbitrary $t_1 \in \mathcal{T}$ is exactly $|\mathcal{H}|/|\mathcal{T}|$.

(b) The fraction of those functions that also takes an arbitrary $m_2 \neq m_1$ in $\mathcal{M}$ to an arbitrary $t_2 \in \mathcal{T}$ (possibly equal to t_1) is $1/|\mathcal{T}|$.

If the fraction in (b) instead is at most $\varepsilon(> 1/|\mathcal{T}|)$, the class $\mathcal{H}$ is ε-Almost Strongly Universal$_2$ (ε-ASU$_2$).

Note that since $\varepsilon \geq 1/|\mathcal{T}|$ [15], SU$_2$ hash functions are the optimal case, corresponding to $1/|\mathcal{T}|$-ASU$_2$ hash functions.

Definition 2.4 (XOR-linear). A hash function h from $\mathcal{M} \to \mathcal{T}$ is called XOR-linear if, for any two m, $m' \in \mathcal{M}$, $h(m \oplus m') = h(m) \oplus h(m')$. Similarly, a family $\mathcal{H}$ is called XOR-linear if any hash function $h \in \mathcal{H}$ is XOR-linear.

More definitions

In addition to the definitions of Universal hash functions, we need or mention the following definitions in this dissertation. The first is the Hamming distance, which will be used in Chapter 4, between two bitstrings.

Definition 2.5 (Hamming distance). Let x and y be two binary strings of equal length. Then the Hamming distance between x and y is defined as the number of positions at which they differ.

The second is the trace distance (to the uniform) that we use as the measure of Eve's partial knowledge of a key.

Definition 2.6 (The trace distance). This is also known as the variational distance or the statistical distance between two probability distributions P_X and P'_X, and is

$$\delta(P_X, P'_X) = \frac{1}{2} \sum_{x \in \mathcal{X}} |P_X(x) - P'_X(x)|. \tag{2.1}$$

The third is the definition of a class of hash functions known as Variationally Universal$_2$ [43].

Definition 2.7 (ε-Varitationlly Universal$_2$ hash functions). Let $\mathcal{M}$ and $\mathcal{T}$ be as before. A class $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$ is ε-Variationally Universal$_2$ if the following two conditions are satisfied:

(a) The number of hash functions in $\mathcal{H}$ that takes an arbitrary $m_1 \in \mathcal{M}$ to an arbitrary $t_1 \in \mathcal{T}$ is exactly $|\mathcal{H}|/|\mathcal{T}|$.

(b) For any two distinct $m_1, m_2 \in \mathcal{M}$ and $t_1 \in \mathcal{T}$, and for a randomly chosen $h \in \mathcal{H}$, the trace distance between the uniform distribution on $\mathcal{T}$ and the conditional distribution of $h(m_2)$ given that $h(m_1) = t_1$ is bounded by ε.

Such hash functions can be constructed by composing ε-AU$_2$ hash functions with SU$_2$ hash functions [43].

Lastly, when we say a key is perfect or ε-perfect, the following notion is used.

Definition 2.8 (Perfectness). A key k is called *perfect* if it is uniformly distributed from the adversary's point of view; a key k is called *ε-perfect*, if its distribution has an ε trace distance to the uniform.

2.4 Unconditionally secure MAC – the Wegman-Carter authentication

Unconditionally secure authentication theory was first developed by Simmons in [27] and later by Wegman and Carter in [14, 28]. Wegman and Carter proposed using the classes of ε-ASU$_2$ hash functions for unconditionally secure MAC constructions. Hence, the name "Wegman-Carter authentication".

When talking about security of an authentication scheme, there are two probabilities to bound: the probability of success in an *impersonation* attack, and the probability of success in a *substitution* attack. In an impersonation attack, the adversary pretends to be a legitimate user and tries to generate the correct tag for a (forged) message with no additional information, as would be given by a valid message-tag pair. In a substitution attack, the adversary intercepts a valid message-tag pair and tries to replace it with a new message-tag pair. This latter attack is more powerful than the former [23].

It is fairly straightforward to see that ε-ASU$_2$ hash functions can be used to construct unconditionally secure authentication schemes in a natural way. Let Alice and Bob share a secret key k to identify a hash function f_k in a family $\mathcal{H}$ of ε-ASU$_2$ hash functions from $\mathcal{M}$ to $\mathcal{T}$. Alice sends her message m along with $t = f_k(m)$ to Bob. Upon receiving m

and t, Bob verifies the authenticity of m by comparing $f_k(m)$ with t. If $f_k(m)$ and t are identical, then Bob accepts m as authentic; otherwise, m will be rejected.

Now, if Eve tries to impersonate Alice and sends m' without knowing the key k, or f_k, the best she can do is to guess the correct tag for m'. The probability of success in this case is $1/|\mathcal{T}|$. Even if Eve waits until seeing a valid message-tag pair (m, t) from Alice, the probability of guessing the correct tag t' for m' is at most ε; cf. Definition 2.3. In other words, even seeing a valid message-tag pair does not increase Eve's success probability above ε. Therefore, by using a family of ε-ASU$_2$ hash functions with suitably chosen ε, one can achieve unconditionally secure message authentication.

In this scheme, however, a key cannot be used more than once, because a repeated use of the same key may give Eve enough information to forge a valid message-tag pair; Definition 2.3 does not say anything about set sizes for three message-tag pairs. Therefore, in the mode of operation considered here, WCA, a new secret key is used for each authentication. The key length for typical known families of ε-ASU$_2$ hash functions is logarithmic in the message length $\log |\mathcal{M}|$ [15–25]. Here and throughout this dissertation, $\log$ denotes the binary logarithm. Hence, the key-consumption rate of WCA is logarithmic in the message length.

To be able to reuse the same hash function many times, Wegman and Carter also proposed an authentication using encrypted tags in [28]. In particular, a message m is first hashed by a secret but fixed hash function f to $f(m)$, then $f(m)$ is encrypted with a one-time pad key to generate the tag t. The key length in this case asymptotically approaches the tag length. This authentication scheme is known as Universal hash function based multiple authentication (WCA+OTP). We will study security of these two schemes in Chapter 5 and 6.

2.5 How to achieve authentication in QKD

Let us now explain how authentication is achieved in QKD. First, Alice and Bob preshare a secret key k long enough to authenticate messages exchanged during the initial QKD round. Then, after the quantum transmission (or raw key generation) phase is completed, Alice sends her message m_A along with its authentication tag $t_\mathrm{A} = f_k(m_\mathrm{A})$, where f_k is an ε-ASU$_2$ hash function identified by k, to Bob. The message here contains, for example, the settings used for encoding/decoding on the quantum channel. Upon receiving the message-tag pair $(m_\mathrm{A}, t_\mathrm{A})$, Bob verifies the authenticity of m_A by comparing t_A with a tag that he generated for the received message using f_k. If they are identical, then Bob can be certain, with high probability, that the message did originate from Alice; otherwise, he rejects the message.

If all goes well and a key is generated successfully in the initial QKD round, then Alice and Bob can reserve a portion of this newly generated key for authentication purposes in the next round. Therefore, in general, a portion of the key generated by QKD in the present round is used to authenticate messages in the subsequent round. For this reason QKD is more accurately called Quantum Key Growing (QKG). Nevertheless, we will use the more accustomed term QKD in this thesis.

There are two consequences of reserving a portion of the QKD-generated key for authentication purposes in the subsequent QKD rounds. First, the key-consumption rate

of authentication has a direct impact on the key output rate of QKD. As mentioned in the preceding section, the key length for typical known families of ε-ASU$_2$ hash functions, and thus the key-consumption rate of the WCA, is logarithmic in the message length $\log|\mathcal{M}|$. We will review these families in the next chapter. Second, authentication uses a partially known (or imperfect) key. This is because QKD-generated keys are ε_{Q}-perfect [5, 6], cf. Definition 2.8. In Chapter 6, we will study security of the WCA scheme with a partially known key in detail.

Part II

Contributions

3

Universal Hash Function Constructions

This chapter begins Part II of the dissertation, in which we present the main contributions of the dissertation in details. As we have seen in the previous chapter, ε-ASU$_2$ hash functions form the basis for ITS MACs. Since their introduction, there has been numerous constructions of ε-ASU$_2$ hash functions suitable for authentication [15–18, 20–22, 24, 25, 28–34]. This chapter provides an overview of some known constructions of ε-ASU$_2$ hash functions including a new one from Paper III. Also, a comparative overview is given in terms of their key length and security parameter.

How to construct?

First let us look at ways of constructing ε-ASU$_2$ hash functions. There are several ways to construct classes of ε-ASU$_2$ hash functions, and the following theorem due to Stinson [16] is one of them.

Theorem 3.1 (Composition)
Let $\mathcal{F}$ be a set of ε_1-AU$_2$ hash functions from $\mathcal{M} \to \mathcal{Z}$, and let $\mathcal{G}$ be a set of ε_2-ASU$_2$ hash functions from $\mathcal{Z} \to \mathcal{T}$. Then, $\mathcal{H} = \mathcal{G} \circ \mathcal{F}$ is an ε-ASU$_2$ hash function family from $\mathcal{M} \to \mathcal{T}$ with $\varepsilon = \varepsilon_1 + \varepsilon_2$.

In the above theorem, if $\mathcal{G}$ is an ε_2-AU$_2$ family of hash functions instead, then so is $\mathcal{H}$ with $\varepsilon = \varepsilon_1 + \varepsilon_2$.

Another way is to use ideas from [20, 21], in particular, compose an ε-AXU$_2$ family with an OTP and that results in an ε-ASU$_2$ family. The resulting family in this case has a security parameter ε that depends on the message length. One can also use direct methods to construct ASU$_2$ hash functions. But, usually recursive techniques based on the composition theorem 3.1 above gives us hash functions with small description length [16]. Of course, direct methods are the building blocks of recursive methods.

Lower bounds

There are lower bounds on the description length (or key length) for ε-ASU$_2$ hash functions derived by Stinson [16], Kabatiankii *et al.* [35], Gemmel and Naor [36], and Nguyen and Roscoe [37]. Nguyen and Roscoe [37] provided new combinatorial bounds that are tighter than the other bounds for the key length. They also identified a value for ε that represents a threshold in the behavior of the various bounds and classified different lower bounds in relation to the threshold value of ε. In particular, Nguyen and Roscoe gave

$$|\mathcal{F}| > |\mathcal{T}|\lceil \log|\mathcal{M}|/\log|\mathcal{Z}| - 1\rceil. \tag{3.1}$$

In [37] there are two lower bounds, one for the case when the message length is a multiple of the tag length and the other for when it is not, but both can be written in this way.

Let us also recall the lower bound by Stinson in [16].

Theorem 3.2 (Lower bound for ε-ASU$_2$ hash function families [16])
If there exists an ε-ASU$_2$ family $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$, then

$$|\mathcal{H}| \geq \frac{|\mathcal{M}|(|\mathcal{T}|-1)^2}{|\mathcal{T}|\varepsilon(|\mathcal{M}|-1)+|\mathcal{T}|-|\mathcal{M}|}+1. \tag{3.2}$$

The proof can be found in [16]. In the SU$_2$ case, this simplifies to

$$|\mathcal{H}| \geq |\mathcal{M}|(|\mathcal{T}|-1)+1. \tag{3.3}$$

Otherwise, if $|\mathcal{M}| \gg |\mathcal{T}|$ the bound simplifies to (in terms of key length)

$$\log|\mathcal{H}| \geq 2\log(|\mathcal{T}|-1) - \log(\varepsilon|\mathcal{T}|-1). \tag{3.4}$$

If in addition $\varepsilon = c/|\mathcal{T}|$ for some constant c and $|\mathcal{T}|$ is large, the right-hand side is close to $2\log|\mathcal{T}|$. If one allows ε to increase when $|\mathcal{M}|$ increases, the bounds decrease which makes it easier to approach $2\log|\mathcal{T}|$, as we shall see below.

3.1 Wegman-Carter construction

We start by Wegman-Carter construction from [28]. Wegman and Carter gave the first construction of ASU$_2$ hash functions in [28] that make use of the following Strongly Universal$_2$ (SU$_2$) hash functions as building blocks.

The SU$_2$ family $\mathcal{H}_1$

In [14], Carter and Wegman gave constructions of several SU$_2$ hash function families. One of them is denoted $\mathcal{H}_1$, which is defined as follows. Let $\mathcal{M}$ and $\mathcal{T}$ be finite sets of size 2^i and 2^j, respectively, with $j \leq i$. Let p be the smallest prime number greater than 2^i. For each $q \in \mathbf{Z}_p \setminus \{0\}$ and $r \in \mathbf{Z}_p$, define a hash function $f_{(q,r)} : \mathcal{M} \to \mathcal{T}$ by the following rule

$$f_{(q,r)}(m) \equiv ((mq+r) \mod p) \mod |\mathcal{T}|. \tag{3.5}$$

Then, $\mathcal{H}_1 = \{f_{(q,r)} : q \in \mathbf{Z}_p \setminus \{0\} \text{ and } r \in \mathbf{Z}_p\}$ is an SU$_2$ family. In fact, it is close to being an SU$_2$ hash function family, in the sense that for a randomly chosen $m \in \mathcal{M}$

there are slightly more hash functions in $\mathcal{H}_1$ that map m to small tag values than to large tag values. The required key length to identify a hash function in this family $\mathcal{H}_1$ is $\log(p(p-1))$.

The properties of this hash function family and another SU_2 hash function family denoted $\mathcal{H}_3$ by Carter and Wegman [14] are studied in [38, 39].

The construction

Let $\mathcal{M}$ be the set of all messages of length $\log|\mathcal{M}|$, and $\mathcal{T}$ be the set of all tags of length $\log|\mathcal{T}|$. Let $L = \log|\mathcal{T}| + \log\log\log|\mathcal{M}|$. Let $\mathcal{H}$ be a set of SU_2 hash function family from the set of strings of length $2L$ to the set of strings of length L. Now let $\mathcal{H}'$ be the set of hash functions from $\mathcal{M}$ to $\mathcal{T}$ constructed as follows. A message $m \in \mathcal{M}$ is first broken into substrings of length $2L$. If needed, the last substring is padded with zeros. Thus, the message is broken into $\lceil\log|\mathcal{M}|/2L\rceil$ substrings. Then, a hash function $h_1 \in \mathcal{H}$ is applied to all the substrings and the resulting outcomes are concatenated. The length of the concatenated strings is now roughly half the length of the original message. This process is repeated using $h_2, h_3, \cdots \in \mathcal{H}$ until only one substring of length L remains. The least significant $\log|\mathcal{T}|$ bits of this last substring is taken as a tag for the message. The sequence of these hash functions $(h_1, h_2, \cdots)$ form a hash function $h' \in \mathcal{H}'$. The length of these sequences of hash functions is $\log\log|\mathcal{M}| - \log\log|\mathcal{T}|$. The key needed to identify h' is the concatenation of the keys needed to identify $h_1, h_2, \cdots$. If the hash function family $\mathcal{H}_1$, which will be introduced in the next section, is used for $\mathcal{H}$, then the key length for $\mathcal{H}'$ will be $4L\log\log|\mathcal{M}|$. This family of hash functions $\mathcal{H}'$ is $2/|\mathcal{T}|$-ASU_2, see [28] for details.

3.2 Stinson

Stinson [16] gave several constructions of AU_2 and ASU_2 hash functions. One of them is a construction of ε-ASU_2 hash functions with $\varepsilon = (\log\log|\mathcal{M}| - \log\log|\mathcal{T}| + 1)/\log|\mathcal{T}|$ and a key length roughly $(\log\log|\mathcal{M}| - \log\log|\mathcal{T}| + 2)\log|\mathcal{T}|$, a reduction by a factor four of the key length needed in the Wegman-Carter construction. This construction is a combination of both direct and recursive methods. First, the following combination of direct and recursive constructions are used to construct an AU_2 hash functions. All lemmas and theorems that follow in this section are due to Stinson [16], and the proofs can be found in that paper.

Lemma 3.1
Let p be a prime. Let $|\mathcal{M}| = p^2$ and $|\mathcal{T}| = p$. For each $k \in GF(p)$, define a function $h_k : GF(p) \times GF(p) \to GF(p)$ as

$$h_k(x, y) = kx + y,$$

for $x, y \in GF(p)$. Then

$$\mathcal{H} := \{h_k : k \in GF(p)\}$$

is a Universal$_2$ (U_2) hash function family.

Theorem 3.3 (Cartesian Product)
If there exists an ε-AU$_2$ family $\mathcal{H}$ of hash functions from $\mathcal{M} \to \mathcal{T}$, then, for all integer $i \geq 1$, there exists an ε-AU$_2$ family $\mathcal{G}$ of hash functions from $\mathcal{M}^i \to \mathcal{T}^i$, where $\mathcal{M}^i = \underbrace{\mathcal{M} \times \cdots \times \mathcal{M}}_{i}$ and $\mathcal{T}^i = \underbrace{\mathcal{T} \times \cdots \times \mathcal{T}}_{i}$, with $|\mathcal{G}| = |\mathcal{H}|$.

Now combining the above two and using the composition theorem 3.1 would give the following.

Theorem 3.4
Let p be a prime power and let $i \geq 1$ be an integer. And let $|\mathcal{M}| = p^{2^i}$ and $|\mathcal{T}| = p$. Then there exists an $(\varepsilon = i/p)$-AU$_2$ family $\mathcal{H}$ of hash functions from $\mathcal{M} \to \mathcal{T}$ with $|\mathcal{H}| = p^i$.

The following direct construction of SU$_2$ hash functions is also needed in Stinson's construction

Theorem 3.5
Let p be a prime power and let s, t be integers with $s \geq t$. Also let $|\mathcal{M}| = p^s$ and $|\mathcal{T}| = p^t$. Then there exists an SU$_2$ family $\mathcal{H}$ of hash functions from $\mathcal{M} \to \mathcal{T}$ with $|\mathcal{H}| = p^{s+t}$.

Now the construction is as follows and it is the result of combining theorem 3.1, 3.4, and 3.5.

Theorem 3.6
Let p be a prime power and let $i \geq 1$ be an integer. Let s, t be integers with $s \geq t$. Also let $|\mathcal{M}| = p^{2^i s}$ and $|\mathcal{T}| = p^t$. Then there exists an ε-ASU$_2$ family $\mathcal{H}$ of hash functions from $\mathcal{M} \to \mathcal{T}$ with $\varepsilon = i/p^s + 1/p^t$ and $|\mathcal{H}| = p^{(i+1)s+t}$.

If we choose $p = 2$, $s = t = \log|\mathcal{T}|$, and $i = \lceil \log\log|\mathcal{M}| - \log\log|\mathcal{T}| \rceil$, then by using the above theorem we obtain an ε-ASU$_2$ family of hash functions from $\mathcal{M} \to \mathcal{T}$, with $\varepsilon = (\log\log|\mathcal{M}| - \log\log|\mathcal{T}| + 1)/|\mathcal{T}|$ and the description length approximately $(\log\log|\mathcal{M}| - \log\log|\mathcal{T}| + 2)\log|\mathcal{T}|$.

3.3 den Boer

In [24], den Boer constructed a scheme with $\varepsilon = \log|\mathcal{M}|/(|\mathcal{T}|\log|\mathcal{T}|)$. This construction is as follows. First, an element $m \in \mathcal{M}$ is split into $m_1, m_2, \cdots, m_n$, each of which is an element in $GF(2^{\log|\mathcal{T}|})$, so $\log|\mathcal{M}| = n\log|\mathcal{T}|$. Define

$$h_{k_1,k_2}(m) = k_1 + \sum_{i=1}^{n} m_i k_2^i, \tag{3.6}$$

where $k_1, k_2 \in GF(2^{\log|\mathcal{T}|})$ and the addition and multiplication are done in $GF(2^{\log|\mathcal{T}|})$. Then, the set

$$\mathcal{H} := \{h_{k_1,k_2} : k_1, k_2 \in GF(2^{\log|\mathcal{T}|})\}$$

is ε-ASU$_2$, with $\varepsilon = |\mathcal{M}|/(|\mathcal{T}|\log|\mathcal{T}|)$; see [24] for details and proof.

This is also called polynomial evaluation hashing over finite fields and was independently discovered by Johansson *et al* [23] and Taylor [40].

3.4 Bierbrauer *et al.*

The following construction is due to Bierbrauer *et al.* [22]. It also uses the composition theorem 3.1 to compose an AU_2 hash function with an SU_2 hash function and achieve an $(\varepsilon = 2/|\mathcal{T}|)$-$ASU_2$ hash function from $\mathcal{M} \to \mathcal{T}$. In detail, it is as follows. Let r, s be integers, and let $q = 2^r$ and $Q = 2^{r+s}$. Let $|\mathcal{M}| = Q^{1+2^s} = 2^{(r+s)(1+2^s)}$ and $|\mathcal{T}| = q = 2^r$. First, an $(\varepsilon_1 = 1/q)$-AU_2 family of hash functions from $\mathcal{M} \to GF(Q)$ is constructed by employing a Reed-Solomon-code over $GF(Q)$ of length Q and of minimum distance $1 + 2^s$. Then, the hash functions in this family are composed with the SU_2 hash functions from $GF(Q) \to \mathcal{T}$ (so $\varepsilon_2 = 1/q$). Thus, the result is an $(\varepsilon = 2/q)$-ASU_2 family $\mathcal{H}$ of hash functions from $\mathcal{M} \to \mathcal{T}$, with $|\mathcal{H}| = Q^2 q = 2^{3r+2s}$.

3.5 Krawczyk

Krawczyk proposed the following LFSR-based hashing [20].

LFSR-based hashing

The basic idea is to use an LFSR with a short key, a secret initial string and a secret feedback polynomial, to generate a longer key that selects a hash function from an ε-AU_2 hash function family. This can be viewed as selecting a certain subset of the linear maps from binary vectors m in $\mathcal{M}$ to binary vectors t in $\mathcal{T}$.

The full set of linear maps from $\mathcal{M}$ to $\mathcal{T}$ is an SU_2 hash function family [14], there denoted $\mathcal{H}_3$. In matrix language, $\mathcal{H}_3$ consists of Boolean matrices of size $\log|\mathcal{T}| \times \log|\mathcal{M}|$. A Boolean matrix is a matrix whose entries are binary. The description length of the hash functions in $\mathcal{H}_3$ is $(\log|\mathcal{M}|)(\log|\mathcal{T}|)$, which makes it impractical. However, if the matrices are restricted to be Toeplitz matrices (constant on diagonals), then the corresponding set of hash functions is still $Universal_2$ [41]. The description length of the hash functions is now reduced to $\log|\mathcal{M}| + \log|\mathcal{T}| - 1$, since a Toeplitz matrix can be uniquely identified by the first column and the first row of the matrix.

With a further restriction on the Toeplitz matrix, it is possible to obtain an ε_1-AXU_2 hash function family with a much smaller description length. In particular, if the consecutive columns of the Toeplitz matrix are restricted to be the consecutive states of an LFSR of length $\log|\mathcal{T}|$, then the hash functions constructed from these matrices form an ε_1-AXU_2 hash function family with $\varepsilon_1 = (2\log|\mathcal{M}|)/|\mathcal{T}|$. The description length of the hash functions in this family is $2\log|\mathcal{T}| + 1$, which is the sum of the length of the initial state and the feedback polynomial; see [20] for details.

Krawczyk's construction continues with a composition with an OTP. More precisely, it is a composition of an ε_1-AXU_2 hash function with an OTP, which has length $\log|\mathcal{T}|$. Therefore, the construction by Krawczyk has $\varepsilon = (1 + 2\log|\mathcal{M}|)/|\mathcal{T}|$ and the key length $3\log|\mathcal{T}| + 1$, which is the sum of the key length of the LFSR-based hash function and the length of the OTP. We note that the feedback polynomials used in the LFSR-based hashing are irreducible, so that the actual key length is slightly less than $3\log|\mathcal{T}| + 1$; see [20, 21] for details on usage and key length.

3.6 Other constructions

There are also other constructions such as Bucket hashing by Rogaway [32], MMH (Multilinear Modular Hashing) by Halevi and Krawczyk [30], and UMAC by Black *et al.* [31]. All three are designed to achieve best software performance and thus very fast. But they have some undesirable properties that make them unsuitable for authentication in QKD. For example, Bucket hashing requires a very long key and has a long output, and is only ε-AU$_2$, so it has to be composed with an (A)SU$_2$ hash function to become ε-ASU$_2$. Another paper [42] by Johansson proposes a bucket hashing scheme with a small key, but this does not have fixed ε and still has a relatively long output. MMH [30] and UMAC [31] are the fastest ones but they require very long keys.

3.7 A new construction

In Paper I, we proposed a construction that uses the composition theorem 3.1 to compose the LFSR-based hashing with an SU$_2$ hashing. The result is an ε-ASU$_2$ hash function family with $\varepsilon = 2/|\mathcal{T}|$ and a relatively small description length.

LFSR-based hashing followed by an SU$_2$ hash function

The goal is to construct an ε-ASU$_2$ hash function family from $\mathcal{M}$ to $\mathcal{T}$ with $\varepsilon = 2/|\mathcal{T}|$ and a small key length. To this end, the LFSR-based hashing [20] and the composition theorem 3.1 are used. Recall that the composition theorem states that if $h = g \circ f$ is the composition of an ε_1-AU$_2$ hash function f from $\mathcal{M} \to \mathcal{Z}$ with an SU$_2$ hash function g from $\mathcal{Z} \to \mathcal{T}$, then h is an ε-ASU$_2$ hash function from $\mathcal{M} \to \mathcal{T}$, with $\varepsilon = \varepsilon_1 + 1/|\mathcal{T}|$. So now, let f be an LFSR-based hash function from $\mathcal{M} \to \mathcal{Z}$. Then f is an ε_1-AU$_2$ hash function with $\varepsilon_1 = (2 \log |\mathcal{M}|)/|\mathcal{Z}|$. Composing f with an SU$_2$ hash function h from $\mathcal{Z} \to \mathcal{T}$ results in an ε-ASU$_2$ hash function from $\mathcal{M} \to \mathcal{T}$ with

$$\varepsilon = \frac{2 \log |\mathcal{M}|}{|\mathcal{Z}|} + \frac{1}{|\mathcal{T}|}. \tag{3.7}$$

Since the goal is to make $\varepsilon = 2/|\mathcal{T}|$, the size $|\mathcal{Z}|$ of the intermediate set $\mathcal{Z}$ should be

$$|\mathcal{Z}| = 2|\mathcal{T}| \log |\mathcal{M}|. \tag{3.8}$$

Formally, this gives the following construction. Let $\mathcal{F}$ be a set of LFSR-based hash functions from $\mathcal{M} \to \mathcal{Z}$, where $\mathcal{Z}$ is an intermediate set of bitstrings of length $\log |\mathcal{T}| + \log \log |\mathcal{M}| + 1$. Then, $\mathcal{F}$ is an ε_1-AU$_2$ with $\varepsilon = (2 \log |\mathcal{M}|)/|\mathcal{Z}| = 1/|\mathcal{T}|$. Let $\mathcal{G}$ be a set of SU$_2$ hash functions from $\mathcal{Z} \to \mathcal{T}$, and also let $\mathcal{H} = \mathcal{G} \circ \mathcal{F}$. Then, by the composition theorem 3.1, it follows that $\mathcal{H}$ is a set of ε-ASU$_2$ hash functions from $\mathcal{M} \to \mathcal{T}$ with $\varepsilon = 2/|\mathcal{T}|$.

We know from Section 3.5 that the family $\mathcal{F}$ of LFSR-based hash functions from $\mathcal{M} \to \mathcal{Z}$ has description length

$$l_{\mathcal{F}} = 2 \log |\mathcal{Z}| + 1 = 2 \log |\mathcal{T}| + 2 \log \log |\mathcal{M}| + 3. \tag{3.9}$$

For the SU_2 family $\mathcal{G}$ of hash functions, the shortest possible description length is roughly equal to $\log|\mathcal{Z}|+\log|\mathcal{T}|$ (cf. (3.3)). And there are indeed constructions whose key length equals exactly $\log|\mathcal{Z}|+\log|\mathcal{T}|$; an example is the construction in Lemma 10 of Bierbrauer *et al.* [22]. Thus, choosing a set of SU_2 hash functions from $\mathcal{Z}\to\mathcal{T}$ with key length $\log|\mathcal{Z}|+\log|\mathcal{T}|$ as a candidate for $\mathcal{G}$ would make the key length $l_{\mathcal{H}}$ for $\mathcal{H}$ equal

$$l_{\mathcal{H}} = l_{\mathcal{F}} + l_{\mathcal{G}} = 4\log|\mathcal{T}| + 3\log\log|\mathcal{M}| + 4. \tag{3.10}$$

We remark that this construction is also Variationally Universal$_2$, cf. Definition 2.7, because it is a composition of ε-AU_2 hash functions with SU_2 hash functions [43].

An alternative to key reduction

This construction uses SU_2 hash functions from $\mathcal{Z}\to\mathcal{T}$, but according to the composition theorem 3.1 ASU_2 hash functions suffice to make the composed function an ASU_2. Therefore, instead of using an SU_2 hash function one can use an ASU_2 hash function from $\mathcal{Z}\to\mathcal{T}$ to further reduce the key length. For instance, if we use the polynomial evaluation hashing, the den Boer's scheme, in the second phase and choose $\log|\mathcal{Z}| = \log\log|\mathcal{M}| + \log|\mathcal{T}|$, then the composition of the LFSR-based hashing from $\mathcal{M}\to\mathcal{Z}$ with the polynomial evaluation hashing from $\mathcal{Z}\to\mathcal{T}$ gives us an ε-ASU_2 hash function, with key length $2\log\log|\mathcal{M}| + 4\log|\mathcal{T}| + 1$ and

$$\varepsilon = \frac{2}{|\mathcal{T}|} + \frac{\log\log|\mathcal{M}| + \log|\mathcal{T}|}{|\mathcal{T}|\log|\mathcal{T}|}. \tag{3.11}$$

This is now larger than $2/|\mathcal{T}|$, but it is bounded above by $4/|\mathcal{T}|$ whenever $\log|\mathcal{M}| \le |\mathcal{T}|$.

3.8 Comparative analysis

Let us now compare the above constructions in terms of their key length, security parameter, and performance. Table 3.1 lists the relevant data in terms of the key length and the security parameter ε.

Table 3.1: *The key length and ε for the new and the existing constructions. The key length for Bierbrauer et al is approximate because of the need to invert se^s in the construction. This involves the Lambert W function (see, e.g., [44]), whose asymptotics for large s gives the expression below.*

Construction	ε	Key length
Wegman-Carter [28]	$2/\lvert\mathcal{T}\rvert$	$4(\log\lvert\mathcal{T}\rvert + \log\log\log\lvert\mathcal{M}\rvert)\log\log\lvert\mathcal{M}\rvert$
Stinson [16]	$(\log\log\lvert\mathcal{M}\rvert - \log\log\lvert\mathcal{T}\rvert + 1)/\lvert\mathcal{T}\rvert$	$(\log\log\lvert\mathcal{M}\rvert - \log\log\lvert\mathcal{T}\rvert + 2)\log\lvert\mathcal{T}\rvert$
den Boer [24]	$(\log\lvert\mathcal{M}\rvert/\log\lvert\mathcal{T}\rvert)/\lvert\mathcal{T}\rvert$	$2\log\lvert\mathcal{T}\rvert$
Bierbrauer *et al.* [22]	$2/\lvert\mathcal{T}\rvert$	$\approx 3\log\lvert\mathcal{T}\rvert + 2\log\log\lvert\mathcal{M}\rvert$
Krawczyk [20]	$(1+2\log\lvert\mathcal{M}\rvert)/\lvert\mathcal{T}\rvert$	$3\log\lvert\mathcal{T}\rvert + 1$
New construction	$2/\lvert\mathcal{T}\rvert$	$4\log\lvert\mathcal{T}\rvert + 3\log\log\lvert\mathcal{M}\rvert + 4$

As can be seen from the table, the new construction like the constructions by Wegman-Carter [28] and Bierbrauer *et al.* [22] has a fixed $\varepsilon = 2/|\mathcal{T}|$, while the others have ε

dependent logarithmically or linearly on the message length $\log|\mathcal{M}|$. In terms of the key length, the new construction clearly consumes much less key than the constructions by Wegman-Carter [28] and Stinson [16], but not as little as the constructions by den Boer, Krawczyk, and Bierbrauer. The construction by den Boer, or the polynomial evaluation based hashing, has the lowest key length $2\log|\mathcal{T}|$ at the cost of an increase in ε.

Another way to compare the schemes is to fix the security parameter ε, and from that and the message length $\log|\mathcal{M}|$ determine the tag length $\log|\mathcal{T}|$ and key length. This is done in Table 3.2, but with the exception of the constructions by Wegman-Carter and Stinson. As we can see from the table, the tag length does not depend on $\log|\mathcal{M}|$ for Bierbrauer *et al.* [22] and the new scheme, while it increases when the message size increases for den Boer [24] and Krawczyk [20]. In terms of key length dependence on $\log|\mathcal{M}|$, the constructions by den Boer [24] and Bierbrauer *et al.* are somewhat better than Krawczyk [20] and the new scheme.

Table 3.2: *The key length and tag length, given ε and $|\mathcal{M}|$. Here, the entries for den Boer are approximate; an approximation of the inverse to $|\mathcal{T}|\log|\mathcal{T}|$ involves the asymptotics of the Lambert W function.*

Construction	$\log\|\mathcal{T}\|$	Key length
den Boer [24]	$\approx -\log\varepsilon + \log\log|\mathcal{M}|$	$\approx -2\log\varepsilon + 2\log\log|\mathcal{M}|$
Bierbrauer *et al.* [22]	$-\log\varepsilon + 1$	$\approx -3\log\varepsilon + 2\log\log|\mathcal{M}|$
Krawczyk [20]	$-\log\varepsilon + \log(1 + 2\log|\mathcal{M}|)$	$-3\log\varepsilon + 3\log(1 + 2\log|\mathcal{M}|) + 1$
New construction	$-\log\varepsilon + 1$	$-4\log\varepsilon + 3\log\log|\mathcal{M}| + 8$

With the alternative approach to key reduction, the key length for the new construction can be reduced to $-4\log\varepsilon + 2\log\log|\mathcal{M}| + 9$ from $-4\log\varepsilon + 3\log\log|\mathcal{M}| + 8$, but with a slight increase in the tag length, for a given $\log|\mathcal{M}|$ and ε. From the Table 3.2, we can also see that, for practical values of parameters ε and $\log|\mathcal{M}|$, e.g., 2^{-64} and 2^{20} respectively, the dominating term in the key length is $-\log\varepsilon$. So, the difference in key length between the constructions, that have a simple relation between the required tag length and the given parameters ε and $\log|\mathcal{M}|$, namely, Bierbrauer *el al.* and the new one, is *not* so big.

3.9 Summary

In this chapter we have given an overview of various known constructions of ASU_2 hash functions, as well as a new one, all suitable for constructing unconditionally secure authentication codes. A comparative analysis of them in terms of their key length and security parameter is also given. Some of the constructions have short key length but their security parameter depends on the message length, while some have a fixed security parameter but their key length is slightly longer.

4

Security of QKD with non-ITS Authentication

We mentioned in Chapter 2 that information-theoretically secure (ITS) authentication is needed in QKD. But what does happen if one instead uses a non-ITS authentication in QKD? This is the main theme of this chapter. We do not, however, cover the general case of non-ITS authentication, which can be difficult to classify. Rather, we consider an implementation of QKD with a particular instantiation of non-ITS authentication. In particular, we investigate security of a QKD protocol with a non-ITS two-step authentication originally proposed in [45] due to its low key-consumption. For complete exposition of the results we refer to Paper II [46] and III [47].

4.1 The authentication scheme

In [45], the authors propose an authentication primitive which aims at decreasing the key consumption of authentication in QKD to improve the key output rate of QKD. The authentication scheme works as follows. Let $\mathcal{M}$ be the set of all binary strings of length $\log|\mathcal{M}|$ (or the set of all messages of length $\log|\mathcal{M}|$), and let $\mathcal{T}$ be the set of all binary strings of length $\log|\mathcal{T}|$ with $\log|\mathcal{T}| < \log|\mathcal{M}|$ (or the set of all tags of length $\log|\mathcal{T}|$). A message m_{A} is first mapped from $\mathcal{M}$ to $\mathcal{Z}$, where $\mathcal{Z}$ is the set of all binary strings of length $\log|\mathcal{Z}|$ with $\log|\mathcal{T}| < \log|\mathcal{Z}| < \log|\mathcal{M}|$, by a publicly known hash function f so that $z_{\mathrm{A}} = f(m_{\mathrm{A}})$. Then, z_{A} is mapped by a secret $h_k \in \mathcal{H}_{\mathcal{Z}}$ to a tag $t_{\mathrm{A}} = h_k(z_{\mathrm{A}})$, where $\mathcal{H}_{\mathcal{Z}} : \mathcal{Z} \mapsto \mathcal{T}$ is an SU_2 family of hash functions and the subscript k is the secret key needed to identify a hash function. The message-tag pair $(m_{\mathrm{A}}, t_{\mathrm{A}})$ is sent over the public channel. To authenticate the message $m_{\mathrm{A}} \in \mathcal{M}$, the legitimate receiver computes $g_k(m_{\mathrm{A}}) = h_k(f(m_{\mathrm{A}}))$ and compares it to t_{A}. If they are identical, then the message will be accepted as authentic; otherwise, it will be rejected. Since $\log|\mathcal{Z}|$ is fixed independently of $\log|\mathcal{M}|$, the key length required for authentication is constant, regardless of the length of the message to be authenticated.

QKD with this authentication protocol was proved to be resistant against the following

straightforward man-in-the-middle (MITM) attack [45, 47].

Definition 4.1 (straightforward man-in-the-middle (MITM) attack). In the *straightforward man-in-the-middle attack* the eavesdropper (Eve) builds or buys a pair of QKD devices identical to those of the legitimate parties (Alice and Bob) and cuts "in the middle" the quantum and classical communication channels connecting Alice and Bob. She now connects each of her devices to the "loose ends" of the quantum and classical channels and launches two *independent* QKD sessions, one with Alice and the other with Bob. Eve effectively pretends to be Bob to Alice and Alice to Bob. Eventually she shares a (different) key with each of the legitimate parties which allows her to communicate with them independently. If Alice sends an encrypted message to Bob, Eve can intercept the message and decrypt it, encrypt it with the key she shares with Bob, and send it to Bob.

In particular, in [45] it was assumed that when Eve mounts this straightforward MITM attack on QKD, she runs two independent QKD sessions; one with Alice and the other with Bob. Also, Eve has random but fixed messages. Random because she measures the particles that she received from Alice in randomly chosen measurement bases and sends particles prepared in randomly chosen bases to Bob. Fixed because the messages she can send to either Alice or Bob are determined by her actions and she cannot change them. Under these assumptions, it was proved in [45] that Eve's probability of success in breaking the authentication is upper bounded by $\varepsilon_1 + \varepsilon_2$. Here, ε_1 is the probability that Alice's message m_{A} and Eve's message $m_{\mathrm{E}}(\neq m_{\mathrm{A}})$ collide under the publicly known hash function f, and $\varepsilon_2 = 1/|\mathcal{T}|$ is the probability of guessing the correct tag when an SU_2 hash function family is used; the proof can be found in the original paper [45].

However, as we shall see below, the two QKD sessions, namely, the Alice-Eve and Eve-Bob QKD sessions need not be independent. Moreover, Eve does not have to follow any rules; in other words, she can cheat.

4.2 The problem

The main problem with this two-step authentication scheme is that whenever Eve intercepts a message-tag pair, say $(m_{\mathrm{A}}, t_{\mathrm{A}})$, then she can check whether her message m_{E} collides with m_{A} under f, that is, whether $f(m_{\mathrm{E}}) = f(m_{\mathrm{A}})$. If they do, then Eve can substitute m_{A} with m_{E} and just send $(m_{\mathrm{E}}, t_{\mathrm{A}})$, since $t_{\mathrm{E}} = t_{\mathrm{A}}$. If $f(m_{\mathrm{E}}) \neq f(m_{\mathrm{A}})$, Eve can choose a different message m'_{E}, which collides with m_{A} under f, and substitute m_{A} with m'_{E}. We must mention here that in QKD no restriction is put on the adversary's computational power and storage capability. Because the intended security of QKD is unconditional security in the presence of an adversary with unlimited computer power and memory.

There is, however, one additional requirement on m'_{E}; it has to have a small Hamming distance (cf. Definition 2.5) to m_{E}. This is because, in the context of QKD, the messages exchanged between the protocol participants during a QKD session contain the settings used for encoding/decoding on the quantum channel, error-correction information and description of a random map depending on in which phase it is sent. Eve's original message m_{E} also contains such information and any changes in m_{E} may result in some error. But by choosing an m'_{E} with a small Hamming distance to m_{E} – in the case of

$f(m_{\mathrm{E}}) \neq f(m_{\mathrm{A}})$, so that $f(m'_{\mathrm{E}}) = f(m_{\mathrm{A}})$ – Eve can have a control over the amount of error introduced so that it will be within the acceptable threshold and be taken care of during, e.g., the reconciliation step.

In Paper II, we use the preceding idea and observe that an eavesdropper can apply more advanced strategies than a straightforward MITM attack and can in fact break some simplified QKD protocols. Security of these protocols, however, can easily be strengthened by appropriate modifications of the post-processing steps, as was argued in [48]. It remained as an open question whether our proposed attack techniques in Paper II are still applicable in practical realisations of QKD. Preliminary results that confirm the applicability of our attack strategy were reported in [49], with some discussions on how to mitigate the problem. In Paper III, we present a series of sophisticated attacks, based on the attack strategy in Paper II, that can break practical implementations of QKD protocols that employ the two-step authentication. The attack details on one of the QKD protocols is presented in the next section. We also propose possible countermeasures and prove necessary and sufficient conditions for unconditional security in Paper III.

4.3 An attack on QKD with this authentication

We now go through the details of an attack on a BB84 protocol that we described in Chapter 2, when the two-step authentication is used.

Protocol description

First, let us recall the BB84 protocol from Chapter 2.

(1) Raw key generation:

(1a) Alice first prepares two random bitstrings of length N: $d^{\mathbf{A}}$ (data bits), and $b^{\mathbf{A}}$ (encoding bases, rectilinear or diagonal). For each pair $\left(d_i^{\mathbf{A}}, b_i^{\mathbf{A}}\right)$, she generates the corresponding quantum state $\rho_i^{\mathbf{A}}$. Here and below in the description, $i = 1, 2, \cdots, N$. She then sends the quantum state $\rho^{\mathbf{A}} = \bigotimes_{i=1}^{N} \rho_i^{\mathbf{A}}$ ("string" of all $\rho_i^{\mathbf{A}}$'s) to Bob.

(1b) Bob selects a string of random measurement bases of length N, $b^{\mathbf{B}}$. He then measures $\rho^{\mathbf{A}}$ in these bases $b^{\mathbf{B}}$ and obtains measurement outcomes $d^{\mathbf{B}} \in \{0, 1, \perp\}^N$. Here $d_i^{\mathbf{B}} = \perp$ denotes that Bob's detectors did not produce a measurement result (e.g. due to absorption in the channel, imperfection of the detectors, and etc.). After the measurement, he sends an acknowledgment message with tag, i.e. $\left(\mathrm{m}_{\mathrm{ack}}, g_{k_1}(\mathrm{m}_{\mathrm{ack}})\right)$, to Alice to end the quantum transmission phase.

(2) Sifting:

(2a) After receiving the acknowledgement message from Bob, Alice sends $\left(b^{\mathbf{A}}, g_{k_2}(b^{\mathbf{A}})\right)$ to Bob.

(2b) Bob calculates a bitstring $b^{\mathbf{A}=\mathbf{B}}$, such that $b_i^{\mathbf{A}=\mathbf{B}} = 1$ if he and Alice prepared and measured $\rho_i^{\mathbf{A}}$ in the same basis, and $b_i^{\mathbf{A}=\mathbf{B}} = 0$ otherwise or if he did not measure anything. He removes from $d^{\mathbf{B}}$ all bits $d_i^{\mathbf{B}}$ where $b_i^{\mathbf{A}=\mathbf{B}} = 0$ and obtains his sifted key $s^{\mathbf{B}}$. He then sends $\left(b^{\mathbf{A}=\mathbf{B}}, g_{k_3}(b^{\mathbf{A}=\mathbf{B}})\right)$ to Alice.

(2c) Upon receiving $b^{\mathbf{A}=\mathbf{B}}$ from Bob, Alice removes from $d^{\mathbf{A}}$ all bits $d_i^{\mathbf{A}}$ where $b_i^{\mathbf{A}=\mathbf{B}} = 0$ and obtains her sifted key $s^{\mathbf{A}}$.

(3) Error estimation and reconciliation:

(3a) Alice estimates the error rate of the quantum channel, selects a forward error correction algorithm ECA and calculates a (binary) bitstring of parity information, $p(s^{\mathbf{A}})$. She then determines a confirmation function CO, calculates $CO(s^{\mathbf{A}})$, and sends $\left(T := (\text{ECA}, p(s^{\mathbf{A}}), CO, CO(s^{\mathbf{A}})), g_{k_4}(T)\right)$ to Bob.

(3b) Bob uses $p(s^{\mathbf{A}})$ to correct $s^{\mathbf{B}}$ and obtains $\hat{s}^{\mathbf{B}}$. He then uses CO to calculate $CO(\hat{s}^{\mathbf{B}})$. He further checks whether $CO(\hat{s}^{\mathbf{B}}) = CO(s^{\mathbf{A}})$. If they are identical, he calculates the error rate ε and sends $\left(\varepsilon, g_{k_5}(\varepsilon)\right)$ to Alice; if not, he sends $\left(\textit{fail}, g_{k_5}(\textit{fail})\right)$ to Alice and aborts the protocol.

(4) Privacy amplification:

(4a) If Alice receives ε, then she determines a privacy function P, sends $\left(P, g_{k_6}(P)\right)$ to Bob, and calculates her final key $K^{\mathbf{A}} = P(s^{\mathbf{A}})$. If she receives *fail* instead, she aborts the protocol.

(4b) If Bob has not aborted in step (3b) and has received P from Alice, then he calculates his final key $K^{\mathbf{B}} = P(\hat{s}^{\mathbf{B}})$. The final keys $K^{\mathbf{A}}$ and $K^{\mathbf{B}}$ are identical with overwhelming probability.

Attack on the protocol

First, Eve impersonates Bob and connects with Alice and also impersonates Alice and connects with Bob. She is assumed to be able to prepare, store, and measure quantum states; and she is equipped with a computational power that enables her to find collisions for f.

(1) Raw key generation:

(1a) Alice performs step (1a) of the protocol.

(1a') Eve intercepts the quantum state $\rho^{\mathbf{A}}$ from Alice and saves it in her quantum memory. She then prepares a random state $\rho^{\mathbf{E}}$ and sends it to Bob (she does exactly as Alice did in step (1a) of the protocol).

(1b) Bob performs step (1b) of the protocol using $\rho^{\mathbf{E}}$ instead of $\rho^{\mathbf{A}}$, denoted: $\rho^{\mathbf{A}} \rightarrow \rho^{\mathbf{E}}$.

(2) Sifting:

(2a) Alice performs step (2a) of the protocol.

(2a') Eve intercepts Alice's message $b^{\mathbf{A}}$ with corresponding tag $g_{k_2}(b^{\mathbf{A}})$, measures her quantum memory in bases $b^{\mathbf{A}}$ and obtains $d^{\mathbf{A}}$. She checks whether $f(b^{\mathbf{E}}) = f(b^{\mathbf{A}})$. If not, then she determines $\tilde{b^{\mathbf{E}}}$ such that it collides with $b^{\mathbf{A}}$ under f and has a small Hamming distance to $b^{\mathbf{E}}$. She sends $\left(\tilde{b^{\mathbf{E}}}, g_{k_2}(\tilde{b^{\mathbf{E}}}) = g_{k_2}(b^{\mathbf{A}})\right)$ to Bob.

(2b) Bob performs step (2b) of the protocol but with $b^{\mathbf{A}} \to \tilde{b^{\mathbf{E}}}$, $b^{\mathbf{A=B}} \to b^{\mathbf{E=B}}$ and obtains $s^{\mathbf{B}}$. Then he sends $b^{\mathbf{E=B}}$ with $g_{k_3}(b^{\mathbf{E=B}})$ to Eve.

(2b') Eve removes from $d^{\mathbf{E}}$ all bits $d_i^{\mathbf{E}}$ where $b_i^{\mathbf{E=B}} = 0$ and obtains $s^{\mathbf{E}} \approx s^{\mathbf{B}}$.

(2b") Eve determines a message $\tilde{b^{\mathbf{A=E}}}$ such that in the next step (2c) Alice will obtain a sifted key that has a small Hamming distance to $s^{\mathbf{E}}$ and that $\tilde{b^{\mathbf{A=E}}}$ and $b^{\mathbf{E=B}}$ collide under f. She then sends $\left(\tilde{b^{\mathbf{A=E}}}, g_{k_3}(\tilde{b^{\mathbf{A=E}}}) = g_{k_3}(b^{\mathbf{E=B}})\right)$ to Alice.

(2c) Alice performs step (2c) of the protocol ($b^{\mathbf{A=B}} \to \tilde{b^{\mathbf{A=E}}}$) and obtains $s^{\mathbf{A}} \approx s^{\mathbf{E}}$.

Eve now has $s^{\mathbf{E}} \approx s^{\mathbf{A}} \approx s^{\mathbf{B}}$ and she is almost done. In the following steps of the protocol Eve only listens to Alice's parity, confirmation and privacy amplification information $p(s^{\mathbf{A}})$, $CO(s^{\mathbf{A}})$ and P , respectively.

(3) Error estimation and reconciliation:

(3a) Alice performs step (3a) of the protocol.

(3a') Eve intercepts T and corrects her $s^{\mathbf{E}}$ and forwards T to Bob intact.

(3b) Bob performs step (3b) of the protocol.

(3b') Eve does nothing but waits for Alice to send the privacy amplification function.

(4) Privacy amplification:

(4a) Alice performs protocol step (4a) and obtains $K^{\mathbf{A}}$.

(4a') Eve intercepts P and privacy amplifies her $s^{\mathbf{E}}$ to obtain her share of the key $K^{\mathbf{E}}$.

(4b) Bob follows protocol step (4b) and obtains $K^{\mathbf{B}}$.

The final keys $K^{\mathbf{A}}$, $K^{\mathbf{E}}$, and $K^{\mathbf{B}}$ are identical.

Next, we consider some countermeasures that result in a range of complications to Eve: (i) increasing Eve's computational load substantially; (ii) forcing her to do considerable online computation rather than offline; and (iii) depriving her of any attack potential by finally re-establishing ITS for the modified construction.

4.4 Countermeasures

We propose a countermeasure that mitigates or, at a cost, prohibits the attack demonstrated in the previous section and in Paper II and III. As we shall see below, there are a number of possibilities ranging from increasing Eve's need for large computational power, all the way to information-theoretic security. As can be expected, the cost of this security improvement comes in the form of an increased secret key consumption.

As discussed earlier, the main enabler of the attacks is that whenever Eve intercepts Alice's message, she can immediately check if her message m_{E} coincides with Alice's under the publicly known hash function f. If not, Eve is free to choose another message m'_{E} that does coincide with Alice's under f, although in some situations there is a small price to pay as described above. To prohibit this we should make it difficult or impossible for Eve to check for this coincidence. The essence of our proposed countermeasure is to use an extra bitsequence to make the output of the public hash function difficult to predict, or even secret, to Eve. This is done in the following way: prepend an extra bitsequence S to the message and authenticate the result. Instead of using the tag $t = g_K(m) = h_K(f(m))$, use the tag $t = g_K(S||m) = h_K(f(S||m))$. If, for example, S is random and secret to Eve, then the output $f(S||m)$ will also be secret to Eve, and she will not be able to search for coincidences in the above manner.

We emphasize that S should be prepended to the message before applying f. The bitsequence S should *not* be concatenated with $f(m)$. The reason for this is fairly obvious. If S is concatenated with $f(m)$ so that $t = h_K(S||f(m))$ or $t = h_K(f(m)||S)$, then Eve can still apply her original attack strategy. All Eve needs in this case is still to find a message that collides with Alice's message under f. We should also stress that for certain classes of hash functions, prepending S to the message has advantages over appending to m (so that $t = h_K(f(m||S))$). When using iterative hash functions like **SHA-1** to calculate $f(m||S)$, Eve can ignore S and search instead for a message m' such that $f(m') = f(m)$. This is known as a partial-message collision attack, see Chapter 5 in Ref. [50]. If f is computed iteratively, $f(m') = f(m)$ will automatically give $f(m'||S) = f(m||S)$ (with appropriate block lengths). This is prohibited by prepending S to the message instead, before computing $f(S||m)$.

Of course, a random secret S would consume secret key, and this may not be desirable. Selecting S can be done in a few ways, and these are the alternatives (including a random secret S): a salt, a nonce, a fixed secret key, and a (one-time) secret key. This countermeasure is simple to implement, and the last alternative seems preferable, if only the key consumption is low. Choosing S to be of the same size as the tag gives a high computational load on Eve, and is efficient in terms of key consumption. It is, however, difficult to estimate the probability of success for Eve, if she has large computational power.

Let us now examine what conditions need to be fulfilled to make the two-step authentication ITS. If the last alternative above is used, it is clear that we want a low probability of collision for a random value of S. This is obtained if any two distinct messages collide under f only for a small number of values of S. More formally, let $\mathcal{S}$ be the set of values of S. Then, if for any two distinct $m_1, m_2 \in \mathcal{M}$, $|\{S \in \mathcal{S} : f(S||m_1) = f(S||m_2)\}| \leq \varepsilon'|\mathcal{S}|$, then we naturally have a low collision probability. A close look at the above condition would tell us that it is precisely the condition for a family of hash functions indexed

by S to be ε'-AU$_2$. The following theorem, whose proof can be found in Paper III, states that this condition is necessary and sufficient for unconditional security.

Theorem 4.1
Let $\mathcal{M}$, $\mathcal{Z}$ and $\mathcal{T}$ be finite sets. Let $\mathcal{F}$ be a family of hash functions from $\mathcal{M}$ to $\mathcal{Z}$, $\mathcal{H}$ a family of SU$_2$ hash functions from $\mathcal{Z}$ to $\mathcal{T}$, and $\mathcal{G} := \mathcal{H} \circ \mathcal{F}$, where $\circ$ stands for element-wise composition. Then $\mathcal{G}$ is ε-ASU$_2$ if and only if $\mathcal{F}$ is ε'-AU$_2$, where $\varepsilon = \varepsilon'(1 - 1/|\mathcal{T}|) + 1/|\mathcal{T}|$.

Thus, to make the two-step authentication ITS, we should construct our fixed public hash function f with the help of an AU$_2$ hash function family $\mathcal{F}$ as follows:

$$f(S||m) = f_S(m),\ f_S \in \mathcal{F}. \tag{4.1}$$

In words, f separates S from the concatenation $S||m$ and uses it as an index to select from the hash function family $\mathcal{F}$ an individual member f_S which is applied to the original message m. Theorem 4.1 makes it possible to relate the message length $\log|\mathcal{M}|$, the security parameter ε', and the key consumption of the system. The exact relations between them can be computed by employing the constructions from the previous chapter.

4.5 Summary

The main conclusion from what we have seen is that *QKD that employs the two-step authentication is not secure*. The MITM attacks we discussed here and in Paper II and III make use of the fact that even if Eve's message does not collide with Alice's under the public hash function f, she is free to choose another message that has a small Hamming distance to her original message.

We stress that the discussed attack pattern is *not* restricted to one single instance, the specific authentication mechanism that we study in this chapter. We conjecture that whenever collisions can be found and the protocol does not use additional secret key [46, 48] (e.g. for encryption of messages) the adversary can compromise the security of the key generated by QKD, following an approach similar to that which is presented here and in Paper III.

The countermeasures that we propose use more secret key, specifically to prevent finding collisions. Prepending a secret key material to the message, before applying the public hash function, will increase the computational resources needed for a successful attack substantially, at a low cost in terms of key material. To achieve unconditional security, however, one should replace the public hash function with an Almost Universal$_2$ hash function family. This last requirement is both necessary and sufficient for unconditional security.

5

Security of Universal Hash Function Based Multiple Authentication

In Chapter 2, we have briefly mentioned the Universal hash function based multiple authentication, originally proposed by Wegman and Carter [28], and we used WCA+OTP to denote this scheme. This scheme uses a fixed ε-ASU$_2$ hash function f_{k_1} followed by a one-time pad (OTP) k_2. In particular, let $\mathcal{H}$ be a family of ε-ASU$_2$ hash functions from the message space $\mathcal{M}$ to the tag space $\mathcal{T}$. Let $\mathcal{K}_1$ be a set of keys of length $\log|\mathcal{H}|$. Let $f_{k_1} \in \mathcal{H}$ be a fixed hash function identified by a randomly chosen secret key $k_1 \in \mathcal{K}_1$. Then in the WCA+OTP scheme, for a series m_i, $i = 1, 2, \cdots$ of messages, the corresponding authentication tags are computed as

$$t_i = f_{k_1}(m_i) \oplus k_2^{(i)}, \quad i = 1, 2, \cdots, \tag{5.1}$$

where $k_2^{(i)}$, for $i = 1, 2, \cdots$, are preshared OTPs. In this chapter we present the heart of Paper IV [51], in which we show that the OTP is necessary in all authentication attempts for this scheme to be unconditionally secure, as opposed to an earlier suggestion in [19] that the OTP $k_2^{(1)}$ in the first round can be skipped.

5.1 A proposed use

It has been stated in [19] as a remark that in this scheme (5.1) the OTP in the initial round can be omitted, since in the authors' own words "it is not necessary". That is, for the first message m_1, $f_{k_1}(m_1)$ can be sent as is. So with this small revision the above scheme becomes as follows: The authentication tag for a series m_i, $i = 1, 2, \cdots$ of messages is now computed as

$$t = \begin{cases} f_{k_1}(m_1), & i = 1, \\ f_{k_1}(m_i) \oplus k_2^{(i-1)}, & i = 2, 3, \cdots. \end{cases} \tag{5.2}$$

Below we will see that this new scheme is not secure in general and there may exist a very simple MAC forgery attack in this case. In the remainder of this chapter when we refer to

the fixed hash function f_{k_1} identified by k_1, we omit the subscript k_1 and simply write f.

5.2 Necessity of the one-time pad

Let us first note that for (5.1) to be ITS, f (or $\mathcal{H}$) at least needs to be AXU_2 [21]. So, for (5.2) to be secure, the subset $\mathcal{H}_{m_1 \mapsto t_1}$ of $\mathcal{H}$ that Eve identifies after seeing the first message-tag pair (m_1, t_1) should be AXU_2. We will now see shortly that this requirement does not necessarily be satisfied even when $\mathcal{H}$ is SU_2, the strongest family of all $\mathrm{Universal}_2$ hash function families.

As described in (5.2), the first message m_1 is sent along with the authentication tag $t_1 = f(m_1)$ from Alice to Bob. Eve intercepts the message-tag pair (m_1, t_1) and identifies the set $\mathcal{H}_{m_1 \mapsto t_1} := \{f \in \mathcal{H} : f(m_1) = t_1\}$. Note that $|\mathcal{H}_{m_1 \mapsto t_1}| = |\mathcal{H}|/|\mathcal{T}|$ according to the definition of ASU_2 hash functions. So, at the end of the first round, from Eve's point of view, the (fixed) secret hash function f is taken from $\mathcal{H}_{m_1 \mapsto t_1}$ instead of $\mathcal{H}$. For the scheme in described by (5.2) to be secure, we need to have, for any two distinct $m, m' \in \mathcal{M}$ and any $t \in \mathcal{T}$,

$$|\{f \in \mathcal{H}_{m_1 \mapsto t_1} : f(m) \oplus f(m') = t\}| \leq \varepsilon |\mathcal{H}_{m_1 \mapsto t_1}|, \tag{5.3}$$

since this would mean that $\mathcal{H}_{m_1 \mapsto t_1}$ is ε-AXU_2. Here, ε is Eve's success probability when attacking the system. The definitions of (A)SU_2 hash functions, however, does not guarantee that (5.3) holds. In fact, $|\{f \in \mathcal{H}_{m_1 \mapsto t_1} : f(m) \oplus f(m') = t\}|$, for some distinct $m, m' \in \mathcal{M}$ and $t \in \mathcal{T}$, could be as large as $|\mathcal{H}_{m_1 \mapsto t_1}|$. If this is the case, then there is a very simple existential forgery attack that Eve can use to attack the authentication. In particular, in the second round, Eve intercepts Alice's message-tag pair (m_2, t_2), where $t_2 = f(m_2) \oplus k_2^{(1)}$, that is sent to Bob. Eve then searches for m_{E} such that $f(m_2) \oplus f(m_{\mathrm{E}}) = t$ is fixed by all $f \in \mathcal{H}_{m_1 \mapsto t_1}$, and sends $(m_{\mathrm{E}}, t \oplus t_2)$ to Bob. Since $f(m_{\mathrm{E}}) \oplus k_2^{(1)} = f(m_{\mathrm{E}}) \oplus f(m_2) \oplus t_2 = t \oplus t_2$, Bob will accept $(m_{\mathrm{E}}, t \oplus t_2)$ as a valid pair.

If the underlying hash function family is XOR-linear (cf. Definition 2.4), then the attack is straightforward. Indeed, there are (A)SU_2 families of hash functions that are XOR-linear, e.g., the SU_2 family $\mathcal{H}_3$ in [28]. In this case, Eve simply observes the first message-tag pair (m_1, t_1) with $t_1 = f(m_1)$ from Alice to Bob, and saves a copy of (m_1, t_1) in her memory. Then in the second round, she intercepts (m_2, t_2) with $t_2 = f(m_2) \oplus k_2^{(1)}$, and replaces it with $(m_{\mathrm{E}}, t_1 \oplus t_2)$ where $m_{\mathrm{E}} = m_1 \oplus m_2$. Eve now knows that m_{E} will be accepted as authentic, because the hash function f is XOR-linear and then

$$f(m_{\mathrm{E}}) \oplus k_2^{(1)} = f(m_1 \oplus m_2) \oplus k_2^{(1)} = f(m_1) \oplus f(m_2) \oplus k_2^{(1)} = t_1 \oplus t_2. \tag{5.4}$$

In the subsequent rounds, Eve uses the same strategy to forge the MAC for a new message chosen similarly to m_{E} above. In general, at the i-th round, Eve replaces the message-tag pair (m_i, t_i) that she intercepted with $(m_1 \oplus m_i, t_1 \oplus t_i)$.

So in general, the following theorem holds.

Theorem 5.1
Consider the authentication scheme described in (5.2)*. Assume that the adversary has*

access to (m_1, t_1) *and* (m_2, t_2)*. Also assume that the OTP is uniformly distributed as a random variable* $K_2^{(1)}$*, from the adversary's point of view. Then, the probability of a successful substitution in the second round is bounded by*

$$\Pr\left\{ f(m_E) \oplus K_2^{(1)} = t_E \,\middle|\, f(m_1) = t, f(m_2) \oplus K_2^{(1)} = t_2 \right\} \leq 1. \tag{5.5}$$

5.3 Summary

In this short chapter, we have reviewed the WCA+OTP scheme and showed that, for this scheme to be ITS, the OTP is necessary in all authentication attempts. We have also showed that for a certain class of (A)SU_2 hash functions there exists a very simple and straightforward attack on the scheme, in the case when the OTP is not used in the first authentication round.

6

Security of the WCA with a Partially Known Key

This chapter presents the last contribution of the dissertation. It was mentioned in the introduction that in QKD after the preshared key is used up, Alice and Bob use a portion of the key generated by QKD for authentication in the later rounds. It was also mentioned that although QKD is provably ITS, keys generated by QKD are *not* perfect, cf. Definition 2.8. In fact, most often crypto keys are not perfect. There is some information, whether it be in the form of guessing probability or trace distance, that the adversary knows about the key. QKD-generated keys are no exception; they are actually ε_{Q}-perfect [5,6]. So this means that except for the initial few rounds in which the preshared key is used, authentication in QKD uses a partially known key. Hence, security of the WCA scheme with a partially known key is the theme of this chapter.

We first consider information-theoretic security, then consider witness indistinguishability as used in the Universally Composable (UC) framework. The results will be stated without proof, which can be found in Paper V.

6.1 Information-theoretic security

Our analysis is focused on the case when the WCA scheme is used with a key that has, from Eve's point of view, ε'-trace distance (cf. Definition 2.6) to the uniform (as a random variable). So, before we look at information-theoretic security, let us take a detour and recall some properties of a probability distribution that has a nonzero trace distance to the uniform.

Probabilities of sets with non-uniform underlying distribution

What we recall below are some simple results of probabilities of subsets of key values, or hash functions, when the key is ε-perfect. In general we denote the probability of a subset

of values $\mathcal{X}' \subseteq \mathcal{X}$ by

$$P_X(\mathcal{X}') = \sum_{x \in \mathcal{X}'} P_X(x).$$

First we note a simple property of the probability of a subset of $\mathcal{X}$, when the distribution has a nonzero trace distance to the uniform distribution.

Lemma 6.1
If the trace distance between P_X and the uniform distribution is ε, then for any subset $\mathcal{X}' \subseteq \mathcal{X}$,

$$\left| P_X(\mathcal{X}') - \frac{|\mathcal{X}'|}{|\mathcal{X}|} \right| \leq \varepsilon. \tag{6.1}$$

Also, there are subsets that reach the bound.

From this lemma follows a bound for the conditional probability of an even smaller subset of $\mathcal{X}$, when the distribution has a nonzero trace distance to the uniform distribution. We will use this later when discussing security with preexisting partial knowledge and additional gained knowledge in the message exchange.

Theorem 6.1
If the trace distance between P_X and the uniform distribution is ε, then for any subsets $\mathcal{X}'' \subseteq \mathcal{X}' \subseteq \mathcal{X}$,

$$\left| P_X(\mathcal{X}'' \,|\, \mathcal{X}') - \frac{|\mathcal{X}''|}{|\mathcal{X}'|} \right| \leq \frac{|\mathcal{X}|}{|\mathcal{X}'|}\varepsilon. \tag{6.2}$$

Also, there are subsets which reach the bound.

Using the above theorem, we can derive a bound for the trace distance of the conditional distribution of x on a subset $\mathcal{X}' \subseteq \mathcal{X}$. This will be useful when discussing trace distance in relation to security later.

Theorem 6.2
If the trace distance between P_X and the uniform distribution is ε, then given a subset $\mathcal{X}' \subseteq \mathcal{X}$, the conditional distribution of x on $\mathcal{X}'$ has trace distance to the uniform (on $\mathcal{X}'$) that is bounded by

$$\tfrac{1}{2} \sum_{x \in \mathcal{X}'} \left| P_X(x \,|\, \mathcal{X}') - \frac{1}{|\mathcal{X}'|} \right| \leq \frac{|\mathcal{X}|}{|\mathcal{X}'|}\varepsilon. \tag{6.3}$$

For certain subsets $\mathcal{X}'$, the bound is reached.

Security in the information-theoretic setting

Now we analyse security of the WCA scheme in information-theoretic setting, in the scenario where the key is ε'-perfect. Recall that the WCA scheme uses ε-ASU$_2$ hashing, and is ε-secure, meaning that the probability of success in a substitution attack is bounded above by ε, if the authentication key is uniformly distributed (perfect). We will now analyse what happens when this is not the case, when the trace distance to the uniform is nonzero. This means that the authentication key is a random variable K to Eve, and we use ε' to denote its trace distance to the uniform.

We start by giving an example of how large Eve's probability for a successful substitution attack can become, even when using a SU_2 family. Since we are talking about a substitution attack, we need to calculate the probability conditioned on Eve having seen a message-tag pair (m,t) from Alice. Let $\mathcal{K}_+$ and $\mathcal{K}_-$ be the set of keys whose probability of being the correct key is higher and lower than $1/|\mathcal{K}|$, respectively. Then, one possible distribution is

$$P_K(k) = \begin{cases} \frac{1}{|\mathcal{K}|} + \varepsilon', & \text{if } k \in \mathcal{K}_+ = \{k_+\} \\ \frac{1}{|\mathcal{K}|} - \varepsilon' \frac{1}{|\mathcal{K}_-|}, & \text{if } k \in \mathcal{K}_- \\ \frac{1}{|\mathcal{K}|}, & \text{otherwise.} \end{cases} \tag{6.4}$$

This has trace distance ε' to the uniform. If $\varepsilon' > 1/|\mathcal{K}|$, the set $\mathcal{K}_-$ must contain more than one value. (Compare with the distribution used in [52] where $P_K(k) = 0$ if $k \in \mathcal{K}_-$; $P_K(k) = 1/(|\mathcal{K}| - |\mathcal{K}_-|)$ if $k \in \mathcal{K}_+ = \mathcal{K} \setminus \mathcal{K}_-$; and $\varepsilon' = |\mathcal{K}_-|/|\mathcal{K}|$.) It is easy to see that Eve's probability for success, without more information on K, is maximal if she chooses $t_\mathrm{E} = f_{k_+}(m_\mathrm{E})$ and m_E is such that $t_\mathrm{E} \neq f_{k_-}(m_\mathrm{E})$ for all $k_- \in \mathcal{K}_-$. Since the hash function family is SU_2, $|\{k : f_k(m_\mathrm{E}) = t_\mathrm{E}\}| = |\mathcal{K}|/|\mathcal{T}|$, and this set contains k_+ but excludes $\mathcal{K}_-$ so that

$$\Pr\left\{f_K(m_\mathrm{E}) = t_\mathrm{E}\right\} = \frac{1}{|\mathcal{K}|} + \varepsilon' + \Big(\frac{|\mathcal{K}|}{|\mathcal{T}|} - 1\Big)\frac{1}{|\mathcal{K}|} = \frac{|\mathcal{K}|}{|\mathcal{T}|}\frac{1}{|\mathcal{K}|} + \varepsilon' = \frac{1}{|\mathcal{T}|} + \varepsilon'. \tag{6.5}$$

It is also easy to see that Eve's probability for success increases if she sees a valid message-tag pair $(m, t = f_K(m))$. Eve's gain will now depend on m, and her gain is maximal if both $f_{k_+}(m) = t$ and $f_{k_-}(m) = t$ for all $k_- \in \mathcal{K}_-$, so that

$$\Pr\left\{f_K(m) = t\right\} = \frac{|\mathcal{K}|}{|\mathcal{T}|}\frac{1}{|\mathcal{K}|} + \varepsilon' - |\mathcal{K}_-|\varepsilon'\frac{1}{|\mathcal{K}_-|} = \frac{1}{|\mathcal{T}|}. \tag{6.6}$$

If ε' is small, there will exist such messages m. Since the hash function family is SU_2, $|\{k : f_k(m_\mathrm{E}) = t_\mathrm{E} \wedge f_k(m) = t\}| = |\mathcal{K}|/|\mathcal{T}|^2$, and again this set contains k_+ but excludes $\mathcal{K}_-$. Therefore

$$\begin{aligned} \Pr\left\{f_K(m_\mathrm{E}) = t_\mathrm{E} \,\middle|\, f_K(m) = t\right\} &= \frac{\Pr\left\{f_K(m_\mathrm{E}) = t_\mathrm{E} \wedge f_K(m) = t\right\}}{\Pr\left\{f_K(m) = t\right\}} \\ &= \frac{\frac{|\mathcal{K}|}{|\mathcal{T}|^2}\frac{1}{|\mathcal{K}|} + \varepsilon'}{\frac{1}{|\mathcal{T}|}} = \frac{\frac{1}{|\mathcal{T}|^2} + \varepsilon'}{\frac{1}{|\mathcal{T}|}} = \frac{1}{|\mathcal{T}|} + |\mathcal{T}|\varepsilon'. \end{aligned} \tag{6.7}$$

Note that this is an equation, not an inequality. Before seeing (m,t) Eve's probability of a successful message insertion attack equals $1/|\mathcal{T}| + \varepsilon'$. After seeing (m,t), Eve's probability of a successful substitution attack *equals* $1/|\mathcal{T}| + |\mathcal{T}|\varepsilon'$.

This might be taken as cause for alarm, but one should note that this is message-dependent: not all message-tag pairs (m,t) will cause such an increase. It was pointed out already in [52] that the message and used key value may be such that Eve may have this unexpectedly high probability of success. On the other hand, in some situations (here, when $f_{k_+}(m) \neq t$), Eve will instead find out that her most likely key value was, in fact, not used, and that she must remove it from the set of possible key values. In this

case, the information she had becomes unusable; she will have lost information. But, importantly, Eve can find out if there was a gain or not, before performing an active (guessing) attack, by using her distribution of K and the received message-tag pair from Alice. Eve then only performs an active attack if her success probability has increased (sufficiently, see [52]). From Alice's point of view, the probability of having her message-tag pair *and* a successful attack from Eve is $1/|\mathcal{T}| + \varepsilon'$, but this probability is *per round*, not per guess (by Eve). Eve does not need to reveal herself by guessing frequently; she can wait for the beneficial case where her success probability is high [52].

Therefore, there is a clear need for an upper bound for the success probability in this situation. For general ε-ASU_2-based authentication, the following theorem holds.

Theorem 6.3 (Bound for guessing probability with partially known key)
Consider the WCA scheme based on ε-ASU$_2$ hashing. If the authentication key is ε'-perfect (as random variable K to the adversary), the probability of a successful message insertion is bounded by

$$\Pr\left\{f_K(m_E) = t_E\right\} \leq \frac{1}{|\mathcal{T}|} + \varepsilon'. \tag{6.8}$$

If in addition the adversary has access to a valid message-tag pair (m, t), the probability of a successful substitution is bounded by

$$\Pr\left\{f_K(m_E) = t_E \,\middle|\, f_K(m) = t\right\} \leq \varepsilon + |\mathcal{T}|\varepsilon'. \tag{6.9}$$

This theorem tells us that the previous example really is a worst-case scenario, so that the upper bound for Eve's success probability after seeing a message-tag pair is $\varepsilon + |\mathcal{T}|\varepsilon'$. Conversely, the example shows that the bound is sharp: there are situations where the bound is reached, so the bound cannot be lowered if one wants information-theoretic security.

In the UC framework (to be discussed below), the relevant figure of merit is the trace distance to the uniform distribution, and not the guessing probability as given above. And also the trace distance increases by the same amount, in the beneficial case for Eve. The key is still random to Eve, but the distribution conditioned on her new knowledge that $h_K(m) = t$ has a larger trace distance to the uniform. A uniform distribution conditioned on $h_K(m) = t$ would be constant at $|\mathcal{T}|/|\mathcal{K}|$ (the set of still possible keys has the size $|\mathcal{K}|/|\mathcal{T}|$), but in our example, if both $f_{k_+}(m) = t$ and $f_{k_-}(m) = t$ for all $k_- \in \mathcal{K}_-$,

$$\begin{aligned} P_K(k_+ \,|\, h_K(m) = t) &= \frac{\Pr\{K = k_+ \wedge h_K(m) = t\}}{P\{h_K(m) = t\}} = \frac{P_K(k_+)}{\Pr\{h_K(m) = t\}} \\ &= \frac{\frac{1}{|\mathcal{K}|} + \varepsilon'}{\frac{1}{|\mathcal{T}|}} = \frac{|\mathcal{T}|}{|\mathcal{K}|} + |\mathcal{T}|\varepsilon'. \end{aligned} \tag{6.10}$$

This forces the conditional distribution of the key to have a high trace distance to the uniform. As before, the example gives the worst-case scenario, and an upper bound for this trace distance is given by the following theorem.

Theorem 6.4 (Bound for trace distance with partially known key)
Consider the WCA scheme based on ε-ASU$_2$ hashing. If the authentication key is ε'-perfect (as random variable K to the adversary), and the adversary has access to a valid

message-tag pair (m,t)*, then the trace distance from the conditional probability to the uniform is bounded by*

$$\tfrac{1}{2}\sum_{k:f_k(m)=t}\left|P_K(k\mid f_K(m)=t)-\frac{1}{|\{k:f_k(m)=t\}|}\right|\leq|\mathcal{T}|\varepsilon'. \tag{6.11}$$

Again, the bound is sharp because of the example: there are situations where the bound is reached, so the bound cannot be lowered if one wants information-theoretic security. Note that, again, that this depends on (m,t), and a similar argument as that used above applies to Eve's success rate. The upper bound is only reached in beneficial situations (for Eve).

The example shows that the bounds cannot be lowered, but are only reached for certain (m,t). This means that the notion of ITS used here is ill suited for the situation. It works well for perfect keys, because there, the probability of a successful attack is equally bounded, with a low bound. It is clear that the situation is the same whether one looks at guessing probability or trace distance; there is a substantial, but non-constant increase. This is the reason to turn to the notion of indistinguishability, which is better suited for this situation.

6.2 Indistinguishability from ideal authentication

The notion of witness indistinguishability was first introduced in [53]. Here, we use the indistinguishability notion to prove that, despite the substantially high bound for ITS, the WCA scheme with an ε'-perfect key is indistinguishable from the ideal authentication, except with probability $\varepsilon+\varepsilon'$. As a natural consequence, UC security of the WCA scheme with an ε'-perfect key directly follows from our proof of indistinguishability.

The ideal functionality of authentication, an *authentic* channel $\mathcal{F}$, connects Alice and Bob in such a way that Bob can be certain that any message output from the channel was sent by Alice. If the message was modified on the channel, the symbol $\perp$ is delivered, see Fig. 6.1. In other words, messages received from $\mathcal{F}$ are either authentic or blocked, and so *cannot* be successfully modified or substituted. Note that there is no confidentiality requirement, so the message can be read by anyone. The real implementation of authentication in the WCA scheme has three components, as depicted in Fig. 6.1: a tag generation algorithm TAG, a verification algorithm VRFY, and a key source. Both TAG and VRFY use the same key. From an input message m, Alice uses TAG to compute a message-tag pair (m,t) where $t=f_k(m)$ and f_k is a hash function from an ε-ASU$_2$ family identified by k. Bob uses VRFY to verify a received message-tag pair (m',t'), and VRFY outputs m' if $f_k(m')=t'$ (for example if $m'=m$ and $t'=t$), otherwise $\perp$.

The distinguisher (in UC terminology, the *environment*) $\mathcal{Z}$ should not be able to distinguish the two systems, except with low probability. It can attempt to distinguish the two by controlling the input to the system (the message m), and the output from the channel (m',t'). The systems should be indistinguishable even under the presence of an *adversary* $m^{\mathbf{A}}$, and it is sufficient to consider the system under an adversary completely controlled by the environment [54], a *dummy adversary* that only forwards the desired channel output from the environment. As is, the systems are trivially distinguishable because of the lack of a tag in the ideal system. We therefore add a *simulator* $\mathcal{S}$ to the ideal functionality,

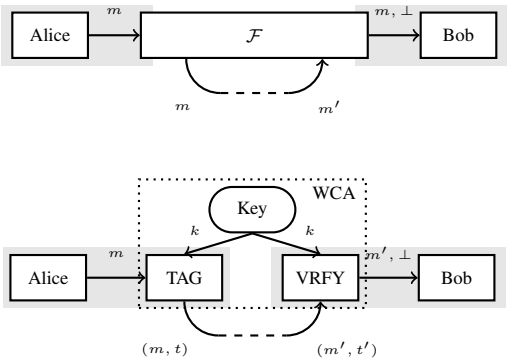


Figure 6.1: *On the top is the ideal functionality: Alice gives her message m to the ideal functionality $\mathcal{F}$, which delivers it to Bob if it has not been modified on the channel ($m' = m$), otherwise the symbol $\perp$ is delivered. On the bottom is the real implementation in WCA: Alice uses the tag generation algorithm TAG to generate a tag t and sends (m, t). At the receiving end, Bob uses the verification algorithm VRFY to check if the received (m', t') is a valid pair. If not, the symbol $\perp$ is delivered.*

that adds a tag t that is generated from m using the appropriate key and hash function to make it indistinguishable from the real case, and strips off any received tag t' after the channel. The name simulator also alludes to simulating the adversary, and is especially simple when simulating the dummy adversary.

We now want to ensure that the environment $\mathcal{Z}$ cannot distinguish between the two cases *(a)* it is interacting with $m^{\mathbf{A}}$ and participants running the WCA scheme or *(b)* it is interacting with $\mathcal{S}$ and participants running $\mathcal{F}$, except with low probability (see Fig. 6.2). Perhaps we should point out that the description here differs slightly from that of [55]. The WCA scheme is resolved in somewhat finer detail and is separated from the participants, and the ideal functionality is that of an authentic channel rather than an immutable but blockable channel. This is done solely for the purpose of clear comparison of the real and the ideal cases, and does not affect the results of the security evaluation. Now, having set the stage, we can state our main theorem.

Theorem 6.5 (Indistinguishability)
No distinguisher $\mathcal{Z}$ can distinguish between the two cases

(a) *it is interacting with $m^{\mathbf{A}}$ and participants running the WCA scheme based on ε-ASU_2 hashing using ε'-perfect authentication key, or*

(b) *it is interacting with $\mathcal{S}$ and participants running $\mathcal{F}$*

except with probability $\varepsilon + \varepsilon'$.

The proof is in Paper V and uses a direct approach, instead of the universally composability theorem.

This theorem tells us that the WCA scheme with an ε'-perfect key behaves like the ideal scheme, except with probability $\varepsilon + \varepsilon'$. This may seem to contradict our previous

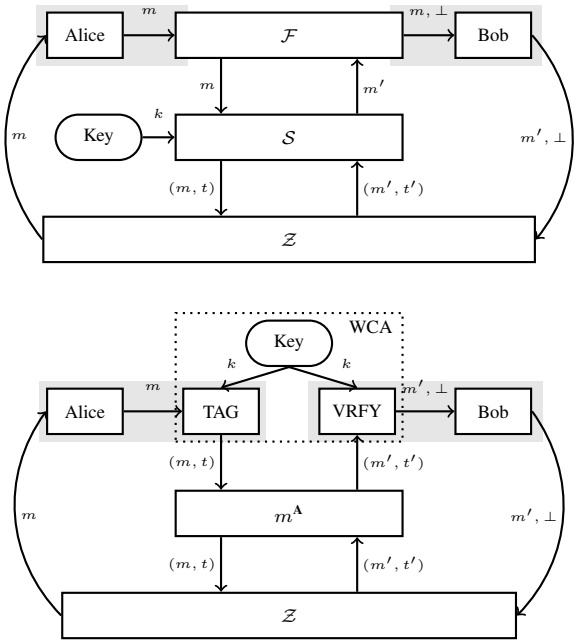


Figure 6.2: *On the top is the ideal case: the ideal functionality $\mathcal{F}$ and simulator $\mathcal{S}$ complete with key input. On the bottom is the real case: the WCA scheme and an adversary $m^{\mathbf{A}}$. The environment $\mathcal{Z}$ wants to distinguish between the two given all the input and output from the system.*

result in the information-theoretic setting, but it does not. The information-theoretic result should be understood as pointing out that the attacker will have high success probability in some rounds, after having seen a valid message-tag pair. The current result shows that this happens seldom enough to retain the expected security. The important lesson is that the attacker can refrain from performing an active attack, if the success probability is low after having seen a valid message-tag pair. This is because she can calculate her success probability from available knowledge on the key and the additional information obtained from a valid message-tag pair. In essence she does not need to reveal herself at each attempt to break the system, but needs only take this risk when the success probability is high. The security parameters should not be read as "the probability that an attacker is revealed, in each attack" but rather "the probability that the system is broken, in each round." It is important to keep this in mind when using this type of authentication, and of course, the size of the security parameters ε and ε' should be chosen accordingly.

Now, the UC security of the WCA scheme with a partially known key follows immediately from the preceding theorem.

Corollary 6.1 (UC security)
Consider the WCA scheme based on ε-ASU$_2$ hashing. Assume that the authentication key k is ε'-perfect. Then the WCA scheme is $\varepsilon + \varepsilon'$-UC-secure.

6.3 Summary

We have presented a security analysis of WCA scheme, in the case of a partially known key whose distribution has ε' trace distance to the uniform distribution. We first analysed information-theoretic security and gave tight upper bounds for the adversary's success probability of breaking the scheme with impersonation and substitution attacks in the information-theoretic setting, with success probability upper bounded by $1/|\mathcal{T}| + \varepsilon'$ and $\varepsilon + |\mathcal{T}|\varepsilon'$, respectively. The latter is substantially higher than expected, but we give an example that reaches the bound, meaning that the bound is sharp. Also in terms of trace distance, a similar increase can be noted. The best possible upper bound to the trace distance after having seen a valid message-tag pair is $|\mathcal{T}|\varepsilon'$; the same example tells us that this bound is sharp. Since the information-theoretic bounds we obtained are substantially higher than what one would expect (e.g. from the UC framework), we then analysed whether the scheme is secure in terms of witness indistinguishability. Despite the high success probability bound and increase in trace distance, the WCA scheme with an ε'-perfect key is indeed indistinguishable from the ideal functionality, except with probability less than $\varepsilon + \varepsilon'$. As a natural consequence, UC security of the scheme with a partially known key follows from our indistinguishability result.

7

Concluding Remarks and Outlook

We end this part of the dissertation with a summary of what we have done so far and with a short discussion of possible future extensions of our work. Without doubt, authentication is indispensable for security of QKD. Proper security analysis of authentication in the context of a partially known key and, more importantly, of QKD itself that does not use information-theoretically secure authentication, is one of the crucial aspects of building a secure QKD system. In this thesis we attempted to do just that. In particular, we studied (a) security of an information-theoretically secure authentication scheme, namely the Wegman-Carter authentication, with a partially known key, (b) security of QKD that employs a specific computationally secure authentication, and (c) Almost Strongly Universal$_2$ (ASU$_2$) hash functions that are the building blocks of unconditionally secure authentication.

For possible extensions of this research, the following can be interesting candidates. One, we have seen various constructions of ASU$_2$ hash functions in Chapter 3. So, finding an efficient construction of ASU$_n$ hash functions, for $n > 2$, with a short key length, would be an interesting line of research. This would enable us to use the same key $n - 1$ times for authentication without jeopardising unconditional security. One can also examine the constructions presented in Chapter 3 to see if one can obtain constructions with optimal security parameter and key length.

Two, it was conjectured in Chapter 4 that whenever collisions in authentication can be found, they could be used to mount an attack on QKD following the same strategy and similar interleaving approaches from Paper II and III. Therefore, a more general and interesting research question would be to prove this conjecture.

Three, we have studied security of the Wegman-Carter authentication in the context of a partially known key in Chapter 6. So, a natural extension of this would be to apply similar security analysis to prove security of the Universal hash function based multiple authentication in the case of a partially known key.

To sum up, the Wegman-Carter authentication based on ASU$_2$ hashing is very well-suited for authentication in QKD. The key consumption rate of this type of authentication

is not high, because as we have seen in Chapter 3 there are efficient (in terms of key length) constructions of ASU_2 hash functions. Our analysis of the security of QKD in the case of a particular computationally secure authentication in Chapter 4 also suggests that unconditionally secure authentication is a must. The Wegman-Carter authentication scheme itself is unconditionally secure, when it uses a completely secret key. And finally, as shown in Chapter 6, for suitably chosen security parameters, it remains secure even when it uses QKD-generated keys.

Bibliography

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (S. Goldwasser, ed.), vol. 35 of *SFCS '94*, pp. 124–134, IEEE Computer Society, 1994.

[2] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Comput. Syst. Signal Process.*, (Bangalore, India), pp. 175–179, 1984.

[3] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Phys. Rev. Lett.*, vol. 67, pp. 661–663, Aug 1991.

[4] P. W. Shor and J. Preskill, "Simple proof of security of the bb84 quantum key distribution protocol," *Phys. Rev. Lett.*, vol. 85, pp. 441–444, 2000.

[5] R. Renner, *Security of quantum key distribution.* PhD thesis, ETH, Zürich, Switzerland, 2005.

[6] J. Mueller-Quade and R. Renner, "Composability in quantum cryptography," *New J. Phys.*, vol. 11, pp. 085006, 18 p, 2009.

[7] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Rev. Mod. Phys.*, vol. 74, pp. 145–195, Mar 2002.

[8] V. Scarani, A. Acín, G. Ribordy, and N. Gisin, "Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations," *Phys. Rev. Lett.*, vol. 92, p. 057901, 2004.

[9] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. A. Smolin, "Experimental quantum cryptography," *J. Cryptology*, vol. 5, no. 1, pp. 3–28, 1992.

[10] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Eurocrypt '93*, LNCS, pp. 410–423, Springer-Verlag, Berlin, 1994.

[11] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM J. Comput.*, vol. 17, no. 2, pp. 210–229, 1988.

[12] C. Bennett, G. Brassard, C. Crepeau, and U. Maurer, "Generalized privacy amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915 –1923, 1995.

[13] G. Gilbert and H. M., "Practical quantum cryptography: A comprehensive analysis," *arXiv:quant-ph/0009027v5*, 2003.

[14] L. Carter and M. N. Wegman, "Universal classes of hash functions," *J. Comput. Syst. Sci.*, vol. 18, pp. 143–154, 1979.

[15] D. R. Stinson, *Cryptography: Theory and Practice*. Discrete Mathematics and Its Applications, Chapman and Hall/CRC, 3rd ed., 2006.

[16] D. R. Stinson, "Universal hash families and the leftover hash lemma, and applications to cryptography and computing," *J. Combin. Math. Combin. Comput.*, vol. 42, pp. 3–31, 2002.

[17] D. R. Stinson, "On the connections between universal hashing, combinatorial designs and error-correcting codes," *Congressus Numerantium*, vol. 114, pp. 7–27, 1996.

[18] D. R. Stinson, "Combinatorial techniques for universal hashing," *J. Comput. Syst. Sci.*, vol. 48, pp. 337–346, 1994.

[19] M. Atici and D. R. Stinson, "Universal hashing and multiple authentication," in *CRYPTO '96* (N. Koblitz, ed.), vol. 1109 of *LNCS*, pp. 16–30, 1996.

[20] H. Krawczyk, "Lfsr-based hashing and authentication," in *CRYPTO '94* (Y. Desmedt, ed.), vol. 839 of *LNCS*, pp. 129–139, 1994.

[21] H. Krawczyk, "New hash functions for message authentication," in *EUROCRYPT '95* (L. C. Guillou and J.-J. Quisquater, eds.), vol. 921 of *LNCS*, pp. 301–310, 1995.

[22] J. Bierbrauer, T. Johansson, G. Kabatianskii, and B. Smeets, "On families of hash functions via geometric codes and concatenation," in *CRYPTO '93* (D. Stinson, ed.), vol. 773 of *LNCS*, pp. 331–342, 1994.

[23] T. Johansson, G. Kabatianskii, and B. Smeets, "On the relations between a-codes and codes correcting independent errors," in *EUROCRYPT '93* (D. Stinson, ed.), vol. 765 of *T. Helleseth*, pp. 1–11, 1994.

[24] B. den Boer, "A simple and key-economical unconditional authentication scheme," *J. Comp. Sec.*, vol. 2, pp. 65–72, 1993.

[25] A. Abidin and J.-Å. Larsson, "New universal hash functions," in *WEWoRC 2011* (S. Lucks and F. Armknecht, eds.), vol. 7242 of *LNCS*, pp. 99–108, Springer-Verlag, 2012.

[26] T. Johansson, *Contributions to unconditionally secure authentication.* PhD thesis, Lund University, Sweden, 1994.

[27] G. Simmons, "A survey of information authentication," *Proceedings of the IEEE*, vol. 76, pp. 603 –620, may 1988.

[28] M. N. Wegman and L. Carter, "New hash functions and their use in authentication and set equality," *J. Comput. Syst. Sci.*, vol. 22, pp. 265–279, 1981.

[29] J. Black, *Message authentication codes.* PhD thesis, University of California Davis, USA, 2000.

[30] S. Halevi and H. Krawczyk, "MMH: Software message authentication in the Gbit/second rates," in *Fast Software Encryption* (E. Biham, ed.), vol. 1267 of *LNCS*, pp. 172–189, 1997.

[31] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "UMAC: Fast and secure message authentication," in *CRYPTO '99* (M. Wiener, ed.), vol. 1666 of *LNCS*, pp. 216–233, 1999.

[32] P. Rogaway, "Bucket hashing and its application to fast message authentication," in *CRYPTO '95* (D. Coppersmith, ed.), vol. 963 of *LNCS*, pp. 29–42, 1995.

[33] D. J. Bernstein, "The poly1305-aes message-authentication code," in *Fast Software Encryption* (H. Gilbert and H. Handschuh, eds.), vol. 3557 of *LNCS*, pp. 32–49, Springer-Verlag, 2005.

[34] V. Shoup, "On fast and provably secure message authentication based on universal hashing," in *CRYPTO '96* (N. Koblitz, ed.), vol. 1109 of *LNCS*, pp. 313–328, Springer Berlin / Heidelberg, 1996.

[35] G. Kabatianskii, B. J. M. Smeets, and T. Johansson, "On the cardinality of systematic authentication codes via error-correcting codes," *IEEE Trans. Inf. Theory*, vol. 42, pp. 566–578, 1996.

[36] P. Gemmell and M. Naor, "Codes for interactive authentication," in *CRYPTO '93* (D. R. Stinson, ed.), vol. 773 of *LNCS*, pp. 355–367, 1994.

[37] L. H. Nguyen and A. W. Roscoe, "A new bound for t-wise almost universal hash functions." IACR Cryptology ePrint Archive, Report 2009/153, 2009.

[38] A. Abidin, "Weaknesses of authentication in quantum cryptography and strongly universal hash functions," 2010.

[39] A. Abidin and J.-Å. Larsson, "Special properties of strongly universal$_2$ hash functions important in quantum cryptography," in *AIP Conference Proceedings, Foundations of Probability and Physics – 5*, vol. 1101, pp. 289–293, American Institute of Physics, 2009.

[40] R. Taylor, "An integrity check value algorithm for stream ciphers," in *Advances in Cryptology - CRYPTO '93* (D. Stinson, ed.), vol. 773 of *Lecture Notes in Computer Science*, pp. 40–48, Springer-Verlag, 1994.

[41] Y. Mansour, N. Nisan, and P. Tiwari, "The computational complexity of universal hashing," in *Proceedings of the 22nd annual ACM symposium on theory of computing*, STOC '90, pp. 235–243, ACM, 1990.

[42] T. Johansson, "Bucket hashing with a small key size," in *EUROCRYPT '97* (W. Fumy, ed.), vol. 1233 of *LNCS*, pp. 149–162, 1997.

[43] T. Krovets and P. Rogaway, "Variationally universal hashing," *Information Processing Letters*, pp. 36–39, 2006.

[44] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," *Adv. Comput. Math.*, vol. 5, pp. 329–359, 1996.

[45] M. Peev, M. Nölle, O. Maurhardt, T. Lorünser, M. Suda, A. Poppe, R. Ursin, A. Fedrizzi, and A. Zeilinger, "A novel protocol-authentication algorithm ruling out a man-in-the middle attack in quantum cryptography," *International Journal of Quantum Information*, vol. 3, pp. 225–231, Mar 2005.

[46] A. Abidin and J.-Å. Larsson, "Vulnerability of "A novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography"," *International Journal of Quantum Information*, vol. 7, pp. 1047–1052, Aug 2009.

[47] C. Pacher, A. Abidin, T. Lorünser, M. Peev, R. Ursin, A. Zeilinger, and J.-Å. Larsson, "Attacks on quantum key distribution protocols that employ non-its authentication," *New J. Phys.*, 2012.

[48] M. Peev, C. Pacher, T. Lorünser, M. Nölle, A. Poppe, O. Maurhart, M. Suda, A. Fedrizzi, R. Ursin, and A. Zeilinger, "Response to "vulnerability of 'a novel protocol-authentication algorithm ruling out a man-in-the-middle attack in quantum cryptography' "," *International Journal of Quantum Information*, vol. 7, pp. 1401–1407, OCT 2009.

[49] A. Abidin, C. Pacher, T. Lorünser, M. Peev, and J.-Å. Larsson, "Quantum cryptography and authentication with low key-consumption," in *Proc. of SPIE*, vol. 8189, pp. 818916–, 2011.

[50] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering*. Wily Publishing, Inc., 2010.

[51] A. Abidin, "On security of universal hash function based multiple authentication," in *ICICS 2012* (T. Chim and T. Yuen, eds.), vol. 7618 of *LNCS*, pp. 303–310, Springer-Verlag, 2012.

[52] J. Cederlöf and J.-Å. Larsson, "Security aspects of the authentication used in quantum cryptography," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1735–1741, 2008.

[53] U. Feige and A. Shamir, "Witness indistinguishability and witness hiding protocols," in *Proceedings of the 22nd ACM Symposium on Theory of Computing*, pp. 416–426, 1990.

[54] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Annual Symposium on Foundations of Computer Science - Proceedings*, pp. 136–145, 2001.

[55] C. Portmann, "Key recycling in authentication," *arXiv:1202.1229v1*, 2012.