Everlasting Multi-Party Computation

Dominique Unruh

University of Tartu

Abstract. A protocol has everlasting security if it is secure against adversaries that are computationally unlimited *after* the protocol execution. This models the fact that we cannot predict which cryptographic schemes will be broken, say, several decades after the protocol execution. In classical cryptography, everlasting security is difficult to achieve: even using trusted setup like common reference strings or signature cards, many tasks such as secure communication and oblivious transfer cannot be achieved with everlasting security. An analogous result in the quantum setting excludes protocols based on common reference strings, but not protocols using a signature card. We define a variant of the Universal Composability framework, everlasting quantum-UC, and show that in this model, we can implement secure communication and general multi-party computation using signature cards as trusted setup.

1 Introduction

Everlasting security. Computers and algorithms improve over time and so does the ability of an adversary to break cryptographic complexity assumptions and protocols. It may be feasible to make a good estimate as to which computational problems are hard *today*, and which encryption schemes unbroken. But it is very difficult to make more than an educated guess as to which cryptographic schemes will be secure, say, ten years from now. Key length recommendations (e.g., [1,2,3]) can only be made based on the assumption that progress continues at a similar rate as today; unexpected algorithmic progress and future technologies like quantum computers can render even the most paranoid choices for the key length obsolete.

This situation is very problematic if we wish to run cryptographic protocols on highly sensitive data such as medical or financial data or government secrets. Such data often has to stay confidential for many decades. But an adversary might intercept messages from a protocol that is secure today, store them, and some decades later, when the underlying cryptosystems have been broken, decrypt them. For highly sensitive data, this would not be an acceptable risk.

One way out is to use protocols with unconditional (information-theoretical) security that are not based on any computational hardness assumptions. For many tasks, however, unconditionally secure protocols simply do not exist (in particular if we cannot assume an majority of honest participants). A compromise is the concept of *everlasting security*. In a nutshell, a protocol is everlastingly

secure if it cannot be broken by an adversary that becomes computationally unlimited *after* the protocol execution. This guarantees that all assumptions need only to hold *during* the protocol execution, sensitive data is not threatened by possible future attacks on today's schemes. We only need to reliably judge the *current* state of the art, not future technologies.

Unfortunately, also for everlasting security, we have strong impossibility results. It is straightforward to see that everlastingly secure public key encryption is not possible, symmetric encryption needs keys as long as the transmitted messages, and most secure multi-party computations (MPC) are impossible (e.g., oblivious transfer, see Section 3).

Quantum cryptography. Since the inception of quantum key distribution (QKD) by Bennett and Brassard [4], it has been known that quantum cryptography can achieve tasks that are impossible in a classical setting: a shared key can be agreed upon between two parties such that even a computationally unlimited eavesdropper does not learn that key. Classically, this is easily seen to be impossible. Crépeau and Kilian [5] showed how, given only a commitment scheme, we can securely realize an oblivious transfer (OT), which in turn, using ideas from Kilian [6] can be used to implement arbitrary unconditionally secure MPC. Classically, given only a commitment, it is impossible to construct arbitrary unconditionally secure MPC (or even everlastingly secure ones, see Section 3). Initial enthusiasm was, however, dampened by strong impossibility results. Mayers [7] showed that it is impossible to construct an unconditionally secure commitment from scratch. Similar impossibilities hold for OT and many other function evaluations (Lo [8]). So the goal to get unconditionally secure MPC is not achievable, even with quantum cryptography.

Also, the usefulness of QKD has been challenged (e.g., by Bernstein [9], who also raises other concerns than the following). To run a QKD protocol, an authenticated channel is needed. But how to implement such a channel? If we use a public key infrastructure for signing messages, we lose unconditional security and thus the main advantage of QKD. If we use shared key authentication, a key needs to be exchanged beforehand. (And, if we exchange an authentication key in a personal meeting, why not just exchange enough key material for one-time pad encryption – storage is cheap.)

Everlasting quantum security. A simple change of focus resolves the problems described in the previous paragraph. Instead of seeing the goal of quantum cryptography in achieving unconditional security, we can see it as achieving *everlasting security*. For example, if we run a QKD protocol and authenticate all messages using signatures and a public key infrastructure, then we do not get an unconditionally secure protocol, but we do get everlasting security: only the signatures are vulnerable to unlimited adversaries, but breaking the security of the signatures after the protocol execution does not help the adversary to recover the key. (Experience and the discussion on composition below show that one has to be careful: we need to check that signatures and QKD indeed play together well and compose securely. We answer this positively in Section 4: we achieve everlastingly secure universally composable security.)

What about secure MPC? Recall that for constructing unconditionally secure MPC in the quantum setting, the only missing ingredient was a commitment. Once we have a commitment, unconditionally secure MPC protocols exist [10]. Unconditionally secure commitments do not exist, but everlastingly secure ones do! Consider a statistically hiding commitment. That is, the binding property may be subject to computational assumptions, but the hiding property holds with respect to unlimited adversaries. Such a scheme is in fact everlastingly secure. Being able to break the binding property of a commitment after the protocol end is of no use – the recipient of the commitment is not listening any more. And the hiding property, i.e., the secrecy of the committed data, holds forever. So a statistically hiding commitment is in fact everlastingly secure. It seems that we have all ingredients for everlastingly secure quantum MPC. The next paragraph, however, shows that the situation is considerably more subtle.

We stress that the neither the concept of everlasting security nor the idea of combining it with quantum cryptography is original to this paper. For example, [11] already suggested to combine QKD with computational authenticated, albeit without proof or analysis of composition problems.

Everlasting security and composition – a cautionary tale. As discussed above, statistically hiding commitments are in fact everlastingly secure, and there are quantum protocols that construct unconditionally secure OT (among other things). Thus, composing a statistically hiding commitment with such a protocol will give us an everlastingly secure OT in the bare model (i.e., not using any trusted setup). But it turns out that this reasoning is wrong! Lo's impossibility of OT [8] can be easily modified to show that unconditional OT is impossible, even if we consider only passive (semi-honest) adversaries. But everlasting security implies unconditional security against passive adversaries: A passive adversary is one that during the protocol follows the protocol (and thus in particular is computationally bounded) but after the protocol may perform unlimited computations. Thus Lo's impossibility excludes the existence of everlastingly secure OTs.

What happened? The problem is that although statistically hiding commitments are everlastingly secure on their own, they lose their security when composed. Composition problems are common in cryptography, but we find this case particularly instructive: The commitment does not lose its security only when composed with some contrived protocol, but instead in a natural construction. And not only does a particular construction break down, we are faced with a general impossibility. And the resulting protocol is insecure in a strong sense: an unlimited adversary can guess either Alice's or Bob's input. (As opposed to a situation where the "break" consists solely of the non-existence of a required simulator.)

One may be tempted to suggest that the failure is not related to the everlasting security, but to the non-composability of the commitments. Damgård and Nielsen [12] present commitment schemes that are universally composable (we elaborate on this notion below, it is a security notion that essentially guarantees "worry-free" composition), that only need a predistributed common reference

strings (CRS), and that are statistically hiding.[1] Yet, when using these commitments to get everlastingly secure OT, we run into the same problem again: We would get an everlastingly secure OT using a CRS, but a generalization of Lo's impossibility shows that no everlastingly secure OT protocols exist even given a CRS (see Section 3).[2] (See also page 12 for another view on the problem in the quantum case.)

Quantum everlasting universal composability. The preceding paragraph shows that, in the setting of everlasting security, it is vital to find definitions that guarantee composability. One salient approach is the Universal Composability (UC) framework by Canetti [14]. In the UC framework, we compare a protocol π against a so-called ideal functionality $\mathcal{F}$ which describes what π should ideally do. (E.g., $\mathcal{F}$ could be a commitment functionality that registers the value Alice commits to, but forwards it to Bob only when Alice requests an open.) We say π UC-emulates $\mathcal{F}$ if for any adversary Adv (that attacks π) there is a simulator Sim (that "attacks" $\mathcal{F}$) we have that no machine $\mathcal{Z}$ (the environment) can distinguish π running with Adv (real model) from $\mathcal{F}$ running with Sim (ideal model). The intuition behind this is that Adv can perform only attacks that can be mimicked by Sim. Since $\mathcal{F}$ is secure by definition, Adv can perform no "harmful" attacks. A salient property of the UC framework is that UC secure protocols can be composed in arbitrary ways (universal composition). By tweaking the details of the definition, we get various variants of UC: If $\mathcal{Z}$, Sim, Adv are polynomial-time, we have computational UC. If they are unlimited, statistical UC (modeling unconditional security). Unlimited quantum machines lead to the definition of statistical quantum-UC [10].

Müller-Quade and Unruh [13] showed that the UC framework can also be adapted to the setting of everlasting security: We quantify over $\mathcal{Z}$, Sim, Adv that are polynomial-time, but we say that $\mathcal{Z}$ distinguishes the real and ideal model if the distribution of $\mathcal{Z}$'s output is not *statistically* indistinguishable. That is, a protocol is considered insecure if one can distinguish real and ideal model when being polynomial-time during the protocol, but unlimited afterwards (statistical indistinguishability means that no *unlimited* machine can distinguish).

The ideas from [13] can be easily adapted to the quantum case. In Section 2, we introduce everlasting quantum UC (eqUC). Here $\mathcal{Z}$, Sim, Adv are quantum-polynomial-time machines (representing the fact that adversaries are limited during the protocol run), but we require that the quantum state output by $\mathcal{Z}$ in the real and ideal model is trace-indistinguishable (two quantum states are trace-indistinguishable if no unlimited quantum machine can distinguish them). The eqUC security notion inherits all composability properties from the UC notion. Also, protocols that are secure with respect to statistical classical or statistical quantum UC are also eqUC-secure. In particular, known quantum protocols for

[1] The schemes given in [12] were only shown secure classically. But we think it likely that similar protocols can be constructed in the quantum setting, too.

[2] That Damgård and Nielsen's commitment does not compose well in an everlasting security setting was already observed in [13]. Their example, however, only shows insecurity when composing with contrived protocols.

constructing MPC from commitments [10] are also eqUC secure. Thus, if we find an eqUC-secure commitment protocol, we immediately get eqUC-secure MPC protocols by composition.

Everlasting quantum-UC commitments. The problem of everlasting UC commitments in the classical setting was already studied in [13]. Their protocol uses a signature card as trusted setup.[3] Here a signature card is a trusted device (modeled as a functionality) such that the owner of the card can sign messages, everyone can access the public key, and no-one (not even the owner) can get the secret key.[4] Their protocol is, however, only known to be secure in the classical setting. In fact, when we try to prove the protocol secure in a quantum setting, we stumble upon an interesting difficulty in the interplay of zero-knowledge proofs of knowledge and signature schemes.

A core step in the protocol is that Alice performs a proof of knowledge P showing that she knows a certain signature σ. In the security proof, we then show that Alice must have obtained σ from the signature card: Assume Alice successfully performs P without requesting σ first. Since P is a proof of knowledge, there is an extractor E (using Alice and indirectly the signing oracle as a black box) that returns a valid witness, i.e., the signature σ. Since E returns the signature without requesting it from the signing oracle, we have a contradiction to the unforgeability of the signature scheme.

It seems that the same reasoning applies against quantum adversaries if we use quantum proofs of knowledge instead. Unfortunately, this is not the case. In a quantum proof of knowledge (as defined by Unruh [17]), an extractor with black box access to the prover executes both the prover (modeled as a unitary operation) as well as its inverse (i.e., the inverse of that unitary). This is the quantum analogue of classical rewinding. So the extractor E will invoke not only the signing oracle, but also its inverse! But unforgeability will not guarantee that there are no forgeries when the adversary accesses the inverse of the signing oracle. Hence the security proof fails.

To avoid this problem, we need a new protocol which does not require rewinding in the same places of the security proof where we use the unforgeability of the signature scheme. We present such a protocol; it is considerably more involved than the one from [13]. We believe that our approach is of independent interest because it shows one way around the limitations of quantum proofs of knowledge.

Bounded quantum storage model. We quickly compare the concept of everlasting security in this paper with the bounded quantum storage model (BQSM; [18]). The BQSM achieves very similar goals. Security in the BQSM guarantees that the protocol cannot be broken by an adversary that has limited quantum memory during the protocol execution and unlimited quantum memory after the execution. The BQSM is thus analogous to everlasting security as discussed

[3] It is impossible to construct UC commitments without using some trusted setup such as a CRS [15]. [13] shows that for everlasting UC, even a CRS is not sufficient.

[4] The last property is mandated, e.g., by the German signature card law [16].

here, except that it considers quantum memory where we consider computational power. The advantage of the BQSM over our model is that when using a BQSM protocol, we only need to make assumptions about the power of the adversary (its quantum memory). In contrast, in our model we need to assume that the computational power is limited *and* that certain mathematical problems are hard. In our view, the main disadvantage of the BQSM is that it might be useful only for a limited time: currently, we may assume a small limit on the adversary's quantum memory. Should quantum technology advance, though, quantum memory might become cheap, and at that point BQSM protocols must not be used any more. In contrast, with everlasting security as in this paper, if an assumption we use in a protocol is broken, it is likely that there still are other assumptions that can be used – we can then fix the protocol by switching the underlying problem. Also, BQSM protocol tend to have a high communication complexity, and composition is more involved (in particular when we wish for universal composability [19]). Then again, our approach requires trusted setup (signature cards). An interesting goal would be protocols that are simultaneously secure in our model and the BQSM.

In the classical setting, the bounded storage model can also be used [20] but has very high communication complexity (quadratic in the memory bound). [21] shows that if we combine bounded storage with temporary computational assumptions, then in the random oracle model we achieve lower communication complexity (but they also show impossibilities when not using the random oracle model). In contrast, our work uses quantum communication and temporary computational assumptions, but no bounded storage.

Further related work. [22] also considers the problem of using an unconditionally hiding computationally binding commitment to construct a quantum OT (as opposed to using directly a functionality). They show that with such a commitment, OT can be realized (no impossibility results are given). However, their OT protocol only computationally hides the sender's inputs (although one may be tempted to assume otherwise as the commitments that are used are unconditionally hiding). In fact, our impossibility results imply that their OT cannot be everlastingly secure.

Organization & contribution. In Section 2 we present the everlasting quantum UC model and the corresponding composition theorem. In Section 3 we show the impossibility of everlastingly secure OT in the classical and the quantum setting using various functionalities. In Section 4 we show that using signature cards or a public key infrastructure, an everlastingly quantum-UC-secure secure channel can be implemented. In Section 5 we implement arbitrary everlastingly quantum-UC-secure multi-party computation using signature cards. Many details and proofs are omitted for space reasons, these are given in the full version [23].

2 Everlasting quantum UC

We now give a terse overview of the definition of everlasting quantum UC (eqUC). Our definition is based on the modeling of UC in the quantum case from [10]. For a full definition, see [23]. The only difference between the definition from [10] and ours is that we allow the environment to output a quantum state and that we require that state to be trace-indistinguishable between real and ideal model. See also [13] for additional discussion on how to model everlasting security in the UC framework.

The basic concept is that of a network. A *network* $\mathbf{N}$ is a set of quantum machines. Each machine maintains a quantum state and can send and receive messages from other machines in the network. A message can be a quantum state. In a network, there is a distinguished machine $\mathcal{Z}$. This machine is initially activated with some input z. When a machine is activated by an incoming message, it can apply an arbitrary quantum operation to the message and its state, producing a new state and an outgoing message. Then the recipient of that message is activated. If a machine sends no outgoing message, $\mathcal{Z}$ is activated. At any point, $\mathcal{Z}$ may terminate with some output quantum state. We denote by $\mathrm{QExec}_{\mathbf{N}}(\eta, z)$ the state output by $\mathcal{Z}$ after an execution of the network $\mathbf{N}$ when $\mathcal{Z}$ gets initial input z and the security parameter is η. We call two networks $\mathbf{N}, \mathbf{N}'$ *trace-indistinguishable* $\mathrm{TD}(\mathrm{QExec}_{\mathbf{N}}(\eta, z), \mathrm{QExec}_{\mathbf{N}'}(\eta, z)) \leq \mu(k)$ is negligible for all $z \in \{0,1\}^*$ and $k \in \mathbb{N}$ where $\mathrm{TD}(\rho, \rho')$ denotes the so-called trace-distance between two quantum states.

A protocol π is a network without $\mathcal{Z}$ or adversary. We cannot execute π itself, but given machines Adv and $\mathcal{Z}$, we can run $\pi \cup \{\mathrm{Adv}, \mathcal{Z}\}$. Given a set C of party identities, let π^C denote the result of replacing, for each $id \in C$, the party with id id by the *corruption party* P^C_{id}. This corruption party just forwards all its communication to the adversary and is controlled by it.

We can now specify everlasting quantum-UC-security. The fact that in this definition, we require the networks to be trace-indistinguishable (i.e., even an unlimited machine cannot distinguish the output states of $\mathcal{Z}$ in real and ideal model), models the fact that in everlasting security, we allow unlimited computations *after* the protocol execution. During the protocol execution, environment, adversary, and simulator are quantum-polynomial-time.

Definition 1 (Everlasting quantum-UC-security). *Let protocols π and ρ be given. We say* π everlastingly quantum-UC-emulates (short eqUC-emulates) *ρ iff for every set C of party ids and for every quantum-polynomial-time adversary* Adv *there is a quantum-polynomial-time simulator* Sim *such that for every quantum-polynomial-time environment $\mathcal{Z}$, the networks $\pi^C \cup \{\mathrm{Adv}, \mathcal{Z}\}$ and $\rho^C \cup \{\mathrm{Sim}, \mathcal{Z}\}$ are trace-indistinguishable.*

We can now define security by comparing a protocol π with some ideal functionality $\mathcal{F}$. If we say that π eqUC-emulates a functionality $\mathcal{F}$, we mean that π eqUC-emulates $\rho_{\mathcal{F}}$ where the ideal protocol $\rho_{\mathcal{F}}$ is the protocol consisting of the functionality $\mathcal{F}$ plus the so-called *dummy-parties*. For each party in π, there is

a dummy-party $\tilde{P}$ that just forwards messages between the environment $\mathcal{Z}$ and the functionality $\mathcal{F}$. The reason for introducing dummy-parties is that dummy-parties can be corrupted. By corrupting Alice in the ideal protocol, the simulator controls the dummy-party and thus effectively Alice's inputs to $\mathcal{F}$ and also gets the outputs from $\mathcal{F}$ to Alice.

If, e.g., we wish to express the fact that π is a eqUC-secure commitment, we say that π eqUC-emulates $\mathcal{F}_{\mathrm{COM}}$ where $\mathcal{F}_{\mathrm{COM}}$ is the commitment functionality defined below. We specify two functionalities that will be used in this paper.

Definition 2 (Commitment). *Let A and B be two parties. The functionality $\mathcal{F}_{\mathrm{COM}}^{A\to B,\ell}$ behaves as follows: Upon (the first) input $(\texttt{commit}, x)$ with $x \in \{0,1\}^{\ell(k)}$ from A, send $\texttt{committed}$ to B. Upon input $\texttt{open}$ from A send $(\texttt{open}, x)$ to B. All communication/input/output is classical.*

We call A the sender and B the recipient.

Definition 3 (Signature card). *Let $\mathfrak{S} = (\mathrm{KG}, \mathrm{Sign}, \mathrm{Verify})$ be a signature scheme. Let A be a party. Then the functionality $\mathcal{F}_{\mathrm{SC}}^{\mathfrak{S},A}$ (*signature card *for scheme $\mathfrak{S}$ with owner A) behaves as follows: Upon the first activation, $\mathcal{F}_{\mathrm{SC}}^{\mathfrak{S},A}$ chooses a verification/signing key pair (pk, sk) using the key generation algorithm $\mathrm{KG}(1^\lambda)$. Upon a message $(\texttt{getpk})$ from a party P or the adversary, it sends pk to P or the adversary, respectively. Upon a message $(\texttt{sign}, m)$ from A $\mathcal{F}_{\mathrm{SC}}^{\mathfrak{S},A}$ computes $\sigma \leftarrow \mathrm{Sign}(sk, m)$ and sends (pk, σ) to A.*

All communication/input/output is classical.

One of the salient features of the UC model is the universal composition theorem. It says that if π eqUC-emulates $\mathcal{F}$, then we can replace $\mathcal{F}$ by π in any context. (Thus allowing for modular protocol design.) The proof of the following theorem follows the lines of that for quantum UC [10].

Theorem 1 (Universal composition theorem). *Let $\mathcal{F}$ and $\mathcal{G}$ be quantum-polynomial-time functionalities. Let π and $\sigma^{\mathcal{F}}$ be quantum-polynomial-time protocols. Here the notation $\sigma^{\mathcal{F}}$ means that σ invokes (possibly many) instances of $\mathcal{F}$. Assume π eqUC-emulates $\mathcal{F}$. Assume further that $\sigma^{\mathcal{F}}$ eqUC-emulates $\mathcal{G}$. Then σ^{π} eqUC-emulates $\mathcal{G}$. (Here σ^{π} is the result of replacing $\mathcal{F}$ by the protocol π in $\sigma^{\mathcal{F}}$.)*

3 Impossibilities

In Section 5, we show that by using signature cards and a quantum channel, we can construct general everlastingly secure MPC protocols. The question arises whether both signature cards and quantum channels are needed. We answer this question positively by showing that (a) in the classical setting, most typical trusted setup (including signature cards) is not sufficient to implement everlasting OT and that (b) in the quantum setting, typical trusted setup such as a CRS is not sufficient to implement everlasting OT. The impossibilities even apply if we do not try to achieve UC security but only to implement a stand-alone OT.

For space reasons, we only give a short overview here. For precise statements and proofs see [23].

Classical impossibilities. The basic observation underlying our impossibility result is that a protocol that is everlastingly secure is also secure against unlimited passive adversaries. This is due to the fact that a passive adversary follows the protocol during the protocol execution (and is thus polynomial-time) and only after the protocol execution performs an unlimited computation. Thus if an unlimited passive adversary could break the protocol, the protocol would not be everlastingly secure either.

We call a functionality $\mathcal{F}$ passively-realizable if there is a protocol that realizes $\mathcal{F}$ with respect to unlimited passive adversaries. We show that the following functionalities are passively-realizable: the coin-toss $\mathcal{F}_{\mathrm{CT}}$, the common reference string $\mathcal{F}_{\mathrm{CRS}}$, the public key infrastructure $\mathcal{F}_{\mathrm{PKI}}$, the commitment $\mathcal{F}_{\mathrm{COM}}$, and the signature card $\mathcal{F}_{\mathrm{SC}}$.

Assume now an everlastingly secure OT protocol π that uses a passively-realizable functionality $\mathcal{F}$. Then π is also secure against passive unlimited adversaries. Let ρ be the protocol that realizes $\mathcal{F}$ (passively). Then π', resulting from replacing $\mathcal{F}$ by ρ, will still be an OT secure against passive unlimited adversaries. (Here, of course, we have to be careful with our definition of passively realizing a functionality – the notion needs to compose such that π' is still secure.) But π' does not use any functionality, and we know that no OT protocol in the bare model can be secure against unlimited passive adversaries.

Concluding, we get:

Theorem 2 (Simplified). *There is no everlastingly secure OT protocol which only uses arbitrarily many instances of $\mathcal{F}_{\mathrm{CT}}$ (coin-toss), $\mathcal{F}_{\mathrm{CRS}}$ (common reference string), $\mathcal{F}_{\mathrm{COM}}$ (commitment), $\mathcal{F}_{\mathrm{PKI}}$ (public key infrastructure), and $\mathcal{F}_{\mathrm{SC}}$ (signature cards).*

Quantum impossibilities. The impossibility in the quantum case follows similar lines. However, the classical notion of passive adversaries does not make sense in the quantum case. (A passive adversary copies all data, this is not possible in the quantum case.) To solve this issue, we consider only protocols that perform no measurements (unitary protocols). Any protocol can be transformed into such a protocol at the expense of additional quantum memory. We call a functionality $\mathcal{F}$ quantum-passively-realizable if there is a unitary protocol π that realizes $\mathcal{F}$ with respect to passive unlimited adversaries (that follow the protocol exactly and do not even copy information). Notice that the requirement that π has to be unitary has the effect that the protocol cannot just throw away information. Thus an adversary that is passive will still have some information left over after the protocol execution. The following functionalities turn out to be quantum-passively-realizable: coin toss $\mathcal{F}_{\mathrm{CT}}$, predistributed EPR pairs $\mathcal{F}_{\mathrm{EPR}}$, public key infrastructure $\mathcal{F}_{\mathrm{PKI}}$ (assuming the secret key is uniquely determined by the public key). However, signature cards and commitments are not! (The reason being that signature cards and commitments do not allow to commit/sign superposi-

tions of messages and thus enforce measurements. This cannot be realized with a unitary protocol.)

Then we can proceed as in the classical case: Assume an everlasting quantum OT protocol π using a quantum-passively-realizable functionality $\mathcal{F}$. This protocol is also secure against unlimited passive adversaries (in the above sense). By replacing $\mathcal{F}$ by the protocol ρ that realizes $\mathcal{F}$, we get a quantum OT protocol π' not using any functionality that is secure against unlimited passive adversaries. But Lo [8] shows that such protocols do not exist. Thus we get:

Theorem 3 (Simplified). *There is no quantum-polynomial-time everlastingly secure OT protocol which only uses arbitrarily many instances of $\mathcal{F}_{\mathrm{CT}}$ (coin-toss), $\mathcal{F}_{\mathrm{CRS}}$ (common reference string), $\mathcal{F}_{\mathrm{EPR}}$ (predistributed EPR pair), $\mathcal{F}_{\mathrm{PKI}}$ (public key infrastructure; assuming that the secret key is uniquely determined by the public key).*

4 Everlasting quantum key distribution

The first application of quantum everlasting security we present in this paper is a new view on quantum key distribution (QKD). Instead of thinking of QKD as a method for getting unconditionally secure message transmission (but then being stuck with the problem of how to realize authenticated channels), we can combine QKD with a computationally secure authenticated channel to get everlastingly secure message transmission. This was already suggested in [11, Section 3.1], but no formal statement or proof was given. We only give a short overview here, for details see [23]. The first step is to implement an authenticated channel from, say, a signature card. (All results in this section also hold with a normal public key infrastructure instead of a signature card.)

Lemma 1 (Authenticated channels from signature cards). *Let $\mathfrak{S}$ be a quantum existentially unforgeable signature-scheme. Then there is a polynomial-time classical protocol π using one instance of $\mathcal{F}_{\mathrm{SC}}^{\mathfrak{S},A}$ such that π eqUC-emulates $\mathcal{F}_{\mathrm{auth}}^{A\to B}$. Here $\mathcal{F}_{\mathrm{auth}}^{A\to B}$ denotes an authenticated channel from A to B.*

The proof presents no surprises. Using $\mathcal{F}_{\mathrm{auth}}^{A\to B}$ and $\mathcal{F}_{\mathrm{auth}}^{B\to A}$, we can implement (with statistical quantum-UC-security) a bidirectional secure channel between Alice and Bob using existing protocols from the literature [24,25,26].

Corollary 1 (Secure channels from signature cards). *Let $\mathfrak{S}$ be a quantum existentially unforgeable signature-scheme. Then there is a polynomial-time protocol π using one instance of $\mathcal{F}_{\mathrm{SC}}^{A,\mathfrak{S}}$ and $\mathcal{F}_{\mathrm{SC}}^{B,\mathfrak{S}}$ each such that π eqUC-emulates $\mathcal{F}_{\mathrm{secchan}}$. (Here $\mathcal{F}_{\mathrm{secchan}}$ denotes a bidirectional secure channel between A and B.)*

5 Everlasting quantum multi-party computation

Classical everlasting UC commitments. In the classical setting, Müller-Quade and Unruh [13] presented a protocol that everlastingly *classical*-UC-emulates (called "long-term UC-emulates" there, ecUC-emulates in the following)

the commitment functionality $\mathcal{F}_{\text{COM}}$ and that uses a signature card $\mathcal{F}_{\text{SC}}$. There protocol cannot be proven secure in the quantum setting (at least we do not know how), but it is instructive to understand their protocol before we present ours.[5]

In order for a commitment protocol to be everlastingly UC secure, we need to achieve the following: Obviously, it needs to be statistically hiding and computationally binding. Furthermore we need that the protocol is extractable: a simulator who controls the signature card can find out what value Alice committed to. And the protocol needs to be equivocal: a simulator who controls the signature card can cheat the binding property and open to a different value. The simulators need to behave in a way that is statistically indistinguishable from the honest behavior of the parties.

The difficulty lies in the extractability. If the committed value can be extracted by the simulator from the interaction, then it must be somehow contained in that interaction, and an unlimited entity can extract it. But that would contradict the statistical hiding property. The approach is to use the signature card $\mathcal{F}_{\text{SC}}^{A}$. When Alice wishes to commit to a value m, we force her to obtain a signature on m. Since the simulator controls $\mathcal{F}_{\text{SC}}$, and since Alice can only sign using $\mathcal{F}_{\text{SC}}$ (even the owner of the signature card does not know the secret key), the simulator will learn m. How do we force Alice to sign m? First, Alice commits to m using a commitment COM. Then Alice obtains a signature σ on (m, u) from $\mathcal{F}_{\text{SC}}$ where u is the opening information for $\text{COM}(m)$. And then Alice proves that she knows a signature σ on (m, u) for some u that opens $\text{COM}(m)$ as m. (Here COM is statistically hiding, and the proof is a statistically witness-indistinguishable argument of knowledge.) **Commit phase:**

$$A \xrightarrow{\quad c := \text{COM}(m) \quad} B$$
$$\xrightarrow{\quad \text{Proof: I know signature } \sigma \text{ on } (m,u) \text{ s.t. } u \text{ opens } c \text{ as } m \text{ or I know the secret key of } \mathcal{F}_{\text{SC}} \quad}$$

We now have extractability: Alice can only succeed in the proof if she gets a signature on (m, u). But then all the simulator has to do is to check which query (m, u) to $\mathcal{F}_{\text{SC}}$ opens the commitment c, and then he knows m. (We explain the "or I know the secret key"-part in a moment.) In the open phase, we cannot just send u, then we would not have equivocality. Instead, Alice proves that she *could* open c as m. **Open phase:**

$$A \xrightarrow{\quad m \quad} B$$
$$\xrightarrow{\quad \text{Proof: I know } u \text{ that opens } c \text{ as } m \text{ or I know the secret key of } \mathcal{F}_{\text{SC}} \quad}$$

Now, if the simulator wishes to equivocate, he simply commits to 0, and later he produces a fake proof that he can open c as m. To produce this fake proof, we

[5] [13] actually first construct a ecUC zero-knowledge proof and use that one to construct an ecUC commitment. For clarity, we present and discuss a direct construction instead. An analogous discussion applies to their original zero-knowledge protocol.

have added the "or I know the secret key sk"-part. Since the simulator knows sk (he controls $\mathcal{F}_{\text{SC}}$), he can always perform the proof using sk as witness. (While Alice, not knowing sk, is forced to prove the part of the statement before the "or".)

Another (quantum) view on the problem. It has been pointed out (by an anonymous reviewer) that in the quantum case, the problem is actually the following: Using a standard unconditionally hiding commitment scheme fails to achieve everlasting security when using it to construct an OT. But this is not due to composability issues, but to the fact that commitment schemes do not force the committer to commit to a classical value, allowing commitments to superpositions instead. In contrast, an ideal commitment functionality would not allow the commit to occur in superposition. This also matches what we do in our quantum-secure protocol below: The signature card forces the committed message to be classical.

We believe this view to be correct, too. Indeed, our protocol would not work if the signature card would allow the adversary to sign superpositions of messages. Yet, this view only partially explains the situation: Even in the purely classical case described above, standard commitments are not sufficient. But in the classical case, the possibility of committing to superpositions obviously cannot be the reason for the problem, indicating that composition is at least part of the problem. In fact, we believe that non-composition and the possibility to commit to superpositions might actually be two sides of the same coin. For example, composition usually requires extractability, i.e., the fact that the adversary can only commit to values he knows. But if the adversary can commit to superpositions, he cannot know what he commits to. It would be interesting (but beyond the scope of this work) to explore this connection further.

Difficulties in the quantum case. Now assume we wish to prove the above protocol secure in the quantum case. Then instead of an argument of knowledge, we need to use a quantum argument of knowledge. But then we run into problems when showing extractability. To show extractability, we need to show that Alice cannot perform the first proof without first sending (m, u) to $\mathcal{F}_{\text{SC}}$. To do so, consider an execution where Alice performs the proof without sending (m, u) to $\mathcal{F}_{\text{SC}}$. We can then consider Alice as a prover $A^{\mathcal{O}}$ with access to a signing oracle $\mathcal{O}$. Applying the extractor E from the argument of knowledge to Alice, we get that $E^{A^{\mathcal{O}}}$ outputs a witness to the statement that is proven. I.e., either a signature on (m, u) or the secret key sk of $\mathcal{O}$. Since $E^{A^{\mathcal{O}}}$ has only black-box access to $\mathcal{O}$, and since $A^{\mathcal{O}}$ and thus also $E^{A^{\mathcal{O}}}$ never signs (m, u), both possibilities contradict the existential unforgeability of the signature scheme. This reasoning works in the classical case. In the quantum case (following [17]), however, the extractor $E^{A^{\mathcal{O}}}$, while rewinding, does the following: It applies both U and U^{-1} where U is the unitary transformation describing the operation of $A^{\mathcal{O}}$. Thus, indirectly $E^{A^{\mathcal{O}}}$ invokes not only $\mathcal{O}$, but also its inverse. Existential unforgeability makes no statement in this case. It could well be that given access to the inverse of $\mathcal{O}$, we can efficiently construct forgeries or even extract the secret key.

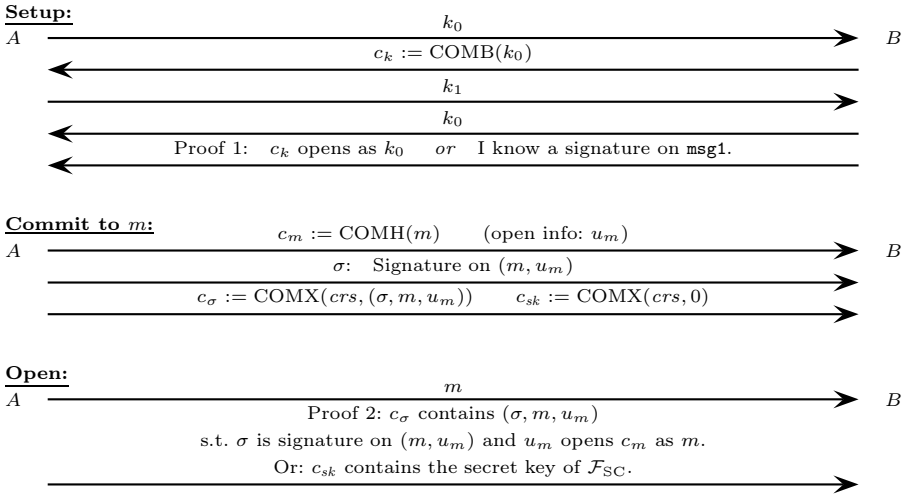

Fig. 1. The commitment protocol based on signature cards – overview. Proof 1 is a witness indistinguishable argument of knowledge, proof 2 is a statistically witness indistinguishable argument. COMH is a statistically hiding quantum-computationally binding commitment. COMB is a quantum-computationally hiding perfectly binding commitment. COMX is a dual-mode commitment.

Note: At a first glance, it might seem that invoking the inverse of $\mathcal{O}$ is not a problem due to the following reasoning. An oracle $\mathcal{O}$ implementing a function $f(x)$ is usually modeled as a unitary mapping $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$. That unitary is self-inverse, so applying $\mathcal{O}^{-1}$ is equivalent to applying $\mathcal{O}$.

However, if the signing oracle $\mathcal{O}$ is modeled in this way, then it can be queried on superposition. Instead, $\mathcal{O}$ should measure the message to be signed first. This could be realised by copying the message (using CNOTs) into fresh ancillae bits. But then $\mathcal{O}$ is not self-inverse any more. Furthermore, to formulate the existential unforgeability, $\mathcal{O}$ additionally needs to keep track of all messages that were signed (otherwise it is not possible to define a "fresh" forgery). Applying the inverse of $\mathcal{O}$ will remove messages from this list, making the notion of a fresh message meaningless.

Our approach. To solve this problem, we need to construct a new protocol in whose security proof we do not need to rewind the signing oracle. A protocol overview is given in Figure 1. We now explain the intuition behind the protocol. As explained above, the main challenge is the extractability of the protocol: Alice commits to m using a commitment scheme COMB, the unveil information is u_m. We need to make sure that Alice is forced to sign (m, u_m) in order to complete the protocol. We cannot just perform a proof of knowledge that Alice knows such a signature σ on (m, u_m) – it might be that Alice proves that she knows a signatures without actually knowing it. To force Alice to actually know the signature, we use the following approach: During the commit phase, Alice commits to (σ, m, u_m) using a commitment scheme COMX. ($c_\sigma := \mathrm{COMX}((\sigma, m, u_m))$.)

And additionally, we let Alice prove ("proof 2" in Figure 1) that the resulting commitment c_σ indeed contains a valid signature σ on (m, u_m). However, we seem to have the same problem as before: How do we guarantee that Alice knows the content of the resulting commitment c_σ? We cannot use rewinding for the same reason as before. Instead, we use a so-called dual-mode commitment for c_σ. A dual-mode commitment COMX depends on a public parameter *crs*: If *crs* is honestly chosen, then COMX is statistically hiding (we need this as otherwise the overall protocol would not be statistically hiding and thus not everlastingly secure). But *crs* can also be chosen in a special way together with a trapdoor *td* such that using *td*, we can efficiently compute (σ, m, u_m) given $c_\sigma = \mathrm{COMX}(crs, (\sigma, m, u_m))$.

Then we can prove extractability of the eqUC commitment protocol roughly as follows:

1. For extracting, the simulator looks at the list of signing queries to $\mathcal{F}_{\mathrm{SC}}$ and finds a suitable pair (m, u_m). We need to show that if Alice opens successfully, there must have been such a signing query for (m, u_m) during the commit phase.
2. To show that, consider a game consisting of an execution with corrupted Alice and that simulator. We change the game such that instead of picking *crs* honestly, we pick it together with a trapdoor *td*. (We discuss below how to do that.)
   Note: the new game will only be computationally indistinguishable from the preceding one. But this does not contradict everlasting security: we are in a side-arm of the proof in order to bound the probability of a certain event ("Alice opens without signing (m, u_m)"). The extracting simulator will still be statistically indistinguishable from an honest recipient of the commitment since the extracting simulator just passively looks at the signing queries.
3. We use the soundness of "proof 2" to show that c_σ contains with overwhelming probability a valid signature σ on (m, u_m). (In the full proof, we need to additionally exclude that Alice proves the alternative option that c_{sk} contains the secret key.)
   Note: we do not claim at this point that Alice knows σ, we only show that whatever is extracted from c_σ using *td* is a valid signature on (m, u_m). In particular, we do not use the unforgeability of the signature scheme in this step.
4. Now we use the unforgeability: We have derived that extracting c_σ using *td* produces a signature on (m, u_m). If this would be the case without having sent (m, u_m) to $\mathcal{F}_{\mathrm{SC}}$, we would have produced a forgery, contradicting unforgeability.
5. So Alice always signs (m, u_m), hence the simulator from Step 1 succeeds with overwhelming probability in extracting.

One thing is missing in this description: How to pick *crs* in a way that we can choose it together with a trapdoor in Step 2? For this, we have the setup phase in Figure 1. Here *crs* is chosen using a coin toss that is designed such that Bob, if he knows a signature on a special message `msg1`, can cheat and choose *crs*

arbitrarily. In Step 2, this allows us to pick *crs* together with a trapdoor by requesting a signature `msg1` from $\mathcal{F}_{\mathrm{SC}}$. (Here `msg1` is an arbitrary fixed bitstring, but syntactically different from all other messages occurring in the protocol.)

Notice that "proof 1" in the coin toss protocol needs to be "of knowledge" (more precisely, a witness-indistinguishable argument of knowledge). However, we do not run into problems with the combination of rewinding and unforgeability this time, because during the execution of "proof 1", the signature card is not accessed by the honest verifier Alice. (And thus the signing oracle is not accessed by the extractor at all.)

Thus, the protocol from Figure 1 is extractable.

Finally, we need to see how to achieve equivocality. Fortunately, this is easy: The equivocating simulator commits to the secret key sk of $\mathcal{F}_{\mathrm{SC}}$ in the commitment c_{sk} (he knows it since he controls $\mathcal{F}_{\mathrm{SC}}$) and commits to 0 in c_σ. Then, in the open phase, to open as an arbitrary m, the simulator just performs "proof 2" using the fact that c_{sk} indeed contains sk. Thus the protocol is equivocal, too. (No fake CRS is needed in this case.)

5.1 The full protocol description

We fix the following notation for interactive commitment schemes: If COM is a commitment scheme, we denote by $(c, u) \leftarrow \mathrm{COM}_{C,R}(1^\eta, m)$ an execution of the commit phase with sender C and recipient R where C commits to the message m. After the protocol execution, both C and R know the value c (e.g., c could be the protocol transcript), intuitively c represents the commitment itself. Furthermore, C gets the value u, the opening information. We assume that the opening phase consists of C sending (m, u), and R verifying the open phase via a deterministic function $\mathrm{COMVerify}(c, m, u)$. For commitments that take a public parameter *crs*, we add this parameter as an additional argument to $\mathrm{COM}_{C,R}$ and COMVerify.

We now give a definition of dual-mode commitments. The definition is close to that of dual-mode commitments in [27]. The main difference is that we additionally require that the honestly chosen CRS is uniformly chosen from a set *CRS*. As discussed in [27], dual-mode commitments (also according to our definition) can be constructed from Regev's cryptosystem [28].

Definition 4. *A dual-mode commitment* COM *is an interactive commitment with a public common reference string crs and which has the following properties:*

- *The common reference string crs is chosen from a set CRS such that one can efficiently sample elements of CRS that are statistically indistinguishable from uniform, and such that CRS is endowed with an arbitrary group operation $*$ (e.g., CRS could be $\{0,1\}^n$ or $\mathbb{Z}_n$ for some n). The operation $*$ is efficiently computable, and inverses with respect to $*$ are efficiently computable.*
- Statistical hiding: *For crs chosen uniformly from CRS,* COM *is statistically hiding.*

– Fake-CRS: *There is an algorithm* $(crs, td) \leftarrow \mathrm{COMFakeCRS}(1^\eta)$ *such that* crs *is quantum-computationally indistinguishable from being uniformly distributed on* CRS.
– Extractability: *There is an efficient algorithm* COMExtract *such that for any quantum-polynomial-time* A*, we have that the following probability is negligible:*

$$\begin{aligned}&\Pr[\exists u, m.\ (m \neq m' \wedge \mathrm{COMVerify}(crs, c, m, u) = 1) : \\ &\quad (crs, td) \leftarrow \mathrm{COMFakeCRS}(1^\eta), \\ &\quad c \leftarrow \mathrm{COM}_{A,R}(crs), m' \leftarrow \mathrm{COMExtract}(td, c)]\end{aligned}$$

Here $c \leftarrow \mathrm{COM}_{A,R}(crs)$ *stands, in abuse of notation, for a commit phase between the adversary* A *and an honest recipient* R*. The value* c *is the value* R *gets at the end of the commit phase.*

Furthermore, we will need a signature scheme $\mathfrak{S}$ that has some (very natural) additional properties besides quantum existential unforgeability. First, we will need deterministic verification. This just means that the verification algorithm is not randomized. Second, we will need that $\mathfrak{S}$ has a matchingKeys-predicate. This means that there is a predicate matchingKeys that can be decided in deterministic polynomial time, and such that for pk, sk chosen according to the key generation algorithm, we have $\mathrm{matchingKeys}(pk, sk) = 1$ with overwhelming probability. And given pk as chosen by the key generation, a quantum polynomial-time algorithm outputs sk with $\mathrm{matchingKeys}(pk, sk) = 1$ only with negligible probability. (Intuitively, this just means that there is a well-defined concept of whether a given secret key matches a given public key.)

Theorem 4 (Commitments from signature cards). *Let* A *and* B *be parties. Let* ℓ *be an integer. Assume the existence of: quantum-computationally witness-indistinguishable quantum arguments of knowledge, statistically witness-indistinguishable arguments,*[6] *statistically hiding quantum-computationally binding commitments, quantum-computationally hiding perfectly binding commitments, dual-mode commitments. Assume that* $\mathfrak{S}$ *is a quantum existentially unforgeable signature scheme with deterministic verification and with* matchingKeys*-predicate.*

Then there is a protocol π *using secure channels and one instance of* $\mathcal{F}_{\mathrm{SC}}^{A,\mathfrak{S}}$ *such that* π *eqUC-emulates* $(\mathcal{F}_{\mathrm{COM}}^{A\to B,\ell})^*$*. (Here* $(\mathcal{F}_{\mathrm{COM}}^{A\to B,\ell})^*$ *is the functionality consisting of many instances of* $\mathcal{F}_{\mathrm{COM}}^{A\to B,\ell}$*. I.e., we can perform many commitments using a single signature card.)*

The protocol π is shown in Figure 1. A more precise description and a security proof are given in [23].

[6] Quantum-computational witness-indistinguishability is defined analogously to the computational witness-indistinguishability (as in, e.g., [29]). Quantum arguments and quantum arguments of knowledge are defined like quantum proofs [30] and quantum proofs of knowledge [17], except that we consider only quantum-polynomial-time provers instead of unlimited provers.

Corollary 2 (Everlasting two-party computation). *Let A and B be parties. Let $\mathcal{G}$ be a well-formed*[7] *classical probabilistic-polynomial-time functionality involving A and B. Under the conditions from Theorem 4, there is a protocol $\pi_{\mathcal{G}}$ using one instance of $\mathcal{F}_{\mathrm{SC}}^{A,\mathfrak{S}}$ such that $\pi_{\mathcal{G}}$ eqUC-emulates $\mathcal{G}^*$.*

This corollary follows from combining Theorem 4 with known statistically secure protocols from [31,32,10]. Analogously, we get everlasting multi-party computation at the price of using more instances of $\mathcal{F}_{\mathrm{SC}}$.

Acknowledgments. This work was funded by institutional research grant IUT2-1 from the Estonian Research Council, and by European Regional Development Fund and the Estonian ICT program 2011-2015 (3.2.1202.12-0001), by the European Social Fund's Doctoral Studies and Internationalisation Programme DoRa, by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS.

References

1. ECRYPT II: Yearly report on algorithms and keysizes. D.SPA.17 Rev. 1.0, ICT-2007-216676 (June 2011)
2. NIST: Recommendation for key management. Special Publication 800-57 Part 1 Rev. 3 (May 2011)
3. Bundesnetzagentur, BSI: Algorithms for qualified electronic signatures (May 2011)
4. Bennett, C.H., Brassard, G.: Quantum cryptography: Public-key distribution and coin tossing. In: IEEE International Conference on Computers, Systems and Signal Processing 1984, IEEE (1984) 175–179
5. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions (extended abstract). In: FOCS 1988, IEEE (1988) 42–52
6. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC 1988, ACM (1988) 20–31
7. Mayers, D.: Unconditionally secure quantum bit commitment is impossible. Physical Review Letters **78**(17) (1997) 3414–3417
8. Lo, H.K.: Insecurity of quantum secure computations. Phys. Rev. A **56** (Aug 1997) 1154–1162 Eprint on arXiv:quant-ph/9611031v2.
9. Bernstein, D.: Cost-benefit analysis of quantum cryptography. Classical and Quantum Information Assurance Foundations and Practice, Dagstuhl Seminar 09311 (2009) Abstract at `http://drops.dagstuhl.de/opus/volltexte/2010/2365`, slides at `http://cr.yp.to/talks/2009.07.28/slides.pdf`.
10. Unruh, D.: Universally composable quantum multi-party computation. In: Eurocrypt 2010. LNCS, Springer (2010) 486–505
11. Alleaume, R., et al.: Secoqc white paper on quantum key distribution and cryptography. arXiv:quant-ph/0701168v1 (2007)

[7] Well-formedness describes certain technical restrictions stemming from the proof by Ishai et al. [31]: Whenever the functionality gets an input, the adversary is informed about the length of that input. Whenever the functionality makes an output, the adversary is informed about the length of that output and may decide when this output is to be scheduled.

12. Damgård, I., Nielsen, J.B.: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: CRYPTO '02. Volume 2442 of LNCS., Springer (2002) 581–596
13. Müller-Quade, J., Unruh, D.: Long-term security and universal composability. Journal of Cryptology **23**(4) (October 2010) 594–671
14. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS 2001, IEEE (2001) 136–145
15. Canetti, R., Fischlin, M.: Universally composable commitments. In: Crypto 2001. Volume 2139 of LNCS., Springer (2001) 19–40
16. Gesetz über Rahmenbedingungen für elektronische Signaturen. Bundesgesetzblatt I 2001, 876 (May 2001) Online available at `http://bundesrecht.juris.de/sigg_2001/index.html`.
17. Unruh, D.: Quantum proofs of knowledge. In: Eurocrypt 2012. Volume 7237 of LNCS., Springer (April 2012) 135–152 Preprint on IACR ePrint 2010/212.
18. Damgård, I., Fehr, S., Salvail, L., Schaffner, C.: Cryptography in the bounded quantum-storage model. In: FOCS 2005. (2005) 449–458
19. Unruh, D.: Concurrent composition in the bounded quantum storage model. In: Eurocrypt 2011. Volume 6632 of LNCS., Springer (May 2011) 467–486
20. Maurer, U.: Conditionally-perfect secrecy and a provably-secure randomized cipher. Journal of Cryptology **5**(1) (1992) 53–66
21. Harnik, D., Naor, M.: On everlasting security in the hybrid bounded storage model. In: ICALP 2006. Volume 4052 of LNCS., Springer (2006) 192–203
22. Crépeau, C., Dumais, P., Mayers, D., Salvail, L.: Computational collapse of quantum state with application to oblivious transfer. In: TCC 2004. Volume 2951 of LNCS., Springer (2004) 374–393
23. Unruh, D.: Everlasting multi-party computation. IACR ePrint 2012/177 (2013) Full version of this paper.
24. Renner, R., König, R.: Universally composable privacy amplification against quantum adversaries. In: TCC 2005. Volume 3378 of LNCS., Springer (2005) 407–425
25. Ben-Or, M., Horodecki, M., Leung, D.W., Mayers, D., Oppenheim, J.: The universal composable security of quantum key distribution. In: TCC 2005. Volume 3378 of LNCS., Springer (2005) 386–406
26. Raub, D., Müller-Quade, J., Steinwandt, R.: On the security and composability of the one time pad. In: Theory and Practice of Computer Science, Proceedings of SOFSEM 2005. Number 3381 in LNCS, Springer (2005) 288–297
27. Damgård, I., Fehr, S., Lunemann, C., Salvail, L., Schaffner, C.: Improving the security of quantum protocols. In: Crypto 2009. Volume 5677 of LNCS., Springer (2009) 408–427
28. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (2009)
29. Goldreich, O.: Foundations of Cryptography – Volume 1 (Basic Tools). Cambridge University Press (August 2001)
30. Watrous, J.: Zero-knowledge against quantum attacks. SIAM J. Comput. **39**(1) (2009) 25–58
31. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Crypto '08. Volume 5157 of LNCS. (2008) 572–591
32. Wullschleger, J.: Oblivious-Transfer Amplification. PhD thesis, ETH Zurich (March 2007) arXiv:cs/0608076v3 [cs.CR].