Composable Everlasting Commitments

Sushanta Paudyal

May 27, 2014

1 Introduction

This report is a prelude to a paper on Everlasting Multi-Party Computation (MPC) by Unruh [1]. We introduce the reader to three concepts that are crucial to grasp before reading said paper.

The first concept is *everlasting security*. Everlasting security is a notion whereby we consider a cryptographic protocol to be secure if it cannot be broken by an adversary that is computationally limited during protocol execution and becomes computationally unlimited afterward. Since everlastingly secure protocols make no assumptions on the computational power of an adversary after execution, designers of such protocols need only concern themselves with the current state of computational power that maybe available to an adversary, and do not need to worry about how powerful the adversary might become afterward, any number of years from now. This is useful for situations where one might need to protect sensitive data for a very long time.

Next, we introduce the reader to a model called the Universal Composability (UC) framework; we briefly discuss how this framework is adapted to the setting of everlasting security. An everlastingly secure protocol within the UC framework is called *everlasting UC secure*.

Finally, we end the report with a discussion on a specific *commitment protocol* (defined later) and see how such a protocol can be made secure within the Universal Composability (UC) framework. We very briefly comment on how an everlasting UC secure commitment protocol gives us everlasting multi-party computation.

2 Everlasting [Quantum] Security

A protocol (π) is said to have everlasting security if an adversary cannot break π after π has been executed. Further, the adversary is computationally

limited during protocol execution and unlimited after execution.

Example 1. Everlastingly secure protocols

Consider a Key Distribution protocol where:
- messages are authenticated with signatures, and
- a Public Key Infrastructure (PKI) is used.
We schematically describe two variants of the said protocol below.

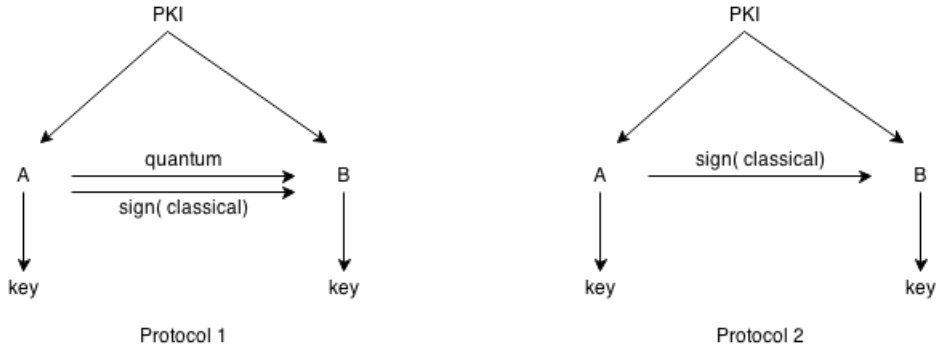


Figure 1: Parties A and B exchange signed messages in both protocols, they exchange 'quantum' messages as well in Protocol 1.

The only difference between protocols 1 and 2 is that messages exchanged in protocol 1 have an additional quantum part; we refer the reader to [1] for specifics about protocol 1. Let us see if either of the protocols are everlastingly secure.

Consider protocol 2. An adversary (let us call her Emmi) is computationally limited during protocol execution, so she collects all signed messages coming from Avia (A) and stores them. Now, after the protocol execution, Emmi becomes computationally unlimited. So, she can break the signatures on the collected messages; this would give her the encrypted messages, which she can then decrypt (again, since she is computationally unlimited), thus obtaining every message exchanged during protocol execution. This would lead Emmi to eventually calculate the shared key, thus breaking the protocol. Thus, protocol 2 is not everlastingly secure.

Now consider protocol 1. Emmi is again computationally limited during protocol execution. She can collect exchanged messages, but will modify the quantum part of a message if she does. If Avia signs and sends a checksum with her message, Brock (B) will know that the message has been tampered with by Emmi, and he can discard that message. Thus, we have altogether stopped Emmi from copying the quantum part of the message

during protocol execution so that when she becomes computationally unlimited afterward, she will not be able to attack this protocol in the way she attacked protocol 2 before. Protocol 1 is thus everlastingly secure. ■

3 Impossibilities when realizing Everlasting Security

In the preceding section, we showed how everlasting security was not realizable in a specific key distribution protocol without the use of quantum cryptography. In this section, we take a step forward and show how it is impossible to realize everlasting security in a more general case. To that end, this section will center around sketching a proof of the following theorem.

Theorem 1. There is no everlastingly secure Oblivious Transfer protocol which only uses arbitrarily many instances of passively realizable functionalities.

Before we sketch a proof, we introduce the following.

- Oblivious Transfer (OT) protocol: is a protocol in which a sender transfers one of two pieces of information to a receiver, but remains oblivious as to what piece has been picked by the receiver; the receiver picks only one piece of information. OT is a building block for multiparty computation.
- Passively realizable functionalities: are functionalities that can be realized by protocols with respect to unlimited passive adversaries. A passive adversary is one who only 'listens' to the protocol.

Proof sketch. We begin by writing the premise of our proof sketch.

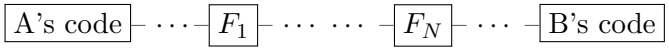
P1 Everlastingly secure OT protocol implies that the protocol is secure against unlimited passive adversaries; this comes from the definition of everlasting security. Everlastingly secure protocols are, per definition, secure against unlimited adversaries (passive or not) after protocol execution. During protocol execution, an unlimited passive adversary would only follow the protocol, without interacting; this would still keep the protocol everlastingly secure.

P2 π is an everlastingly secure OT protocol.

P3 π only uses arbitrarily many instances of passively realizable functionalities.

We work toward a contradiction.
From **P3**, we represent π as a set of functionalities, along with whatever the parties (call them A and B) execute. Consider the following illustration.

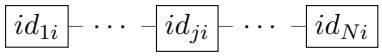
π :



where F_i: a passively realizable functionality.

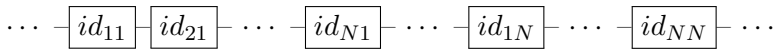
From the definition of passively realizable functionalities, we have a protocol, ρ_i, that realizes F_i against unlimited passive adversaries. We represent ρ_i as a set of *machines*, written as 'id' with a subscript. Consider the illustration below.

ρ_i:



We now construct π' by 'replacing' each passively realizable functionality (F_i) by ρ_i. We show this pictorially.

π' :



π' is still an OT protocol that is secure against unlimited passive adversaries. But notice that π' uses no passively realizable functionalities.
It is known that no OT protocol in the 'bare' model can be secure against unlimited passive adversaries; this is another way of phrasing that no OT protocol that uses no functionalities can be secure against unlimited passive adversaries.

Thus, π' cannot be secure against unlimited passive adversaries. This contradicts **P1** and concludes the proof sketch. □

The sharp reader will have noticed that in the preceding proof sketch, we replaced a functionality by a protocol that implemented the functionality. Such replacements/ compositions may not always result in a secure protocol. We now describe a framework called Universal Composability and see how protocols compose in this framework.

4 Everlasting Quantum Universal Composability

The framework of Universal Composability (UC) is a general-purpose model designed for analyzing cryptographic protocols [2]. In this framework, security is defined in the sense of protocol emulation; we say protocol π 'UC-emulates' an ideal functionality F if no machine (or environment) can distinguish π running with an adversary from F running with a simulator that mimics the adversary. This would mean that any attack against the protocol could be simulated as an attack against the functionality; since the functionality is ideal, it follows that the protocol is secure.

Example 2. Security of a composed protocol

Consider a protocol π that UC-emulates an ideal functionality F. Also consider another protocol σ^F that UC-emulates another ideal functionality G; by σ^F, we denote a protocol where σ invokes polynomially many instances of F. We symbolically represent our scenario in the following way:

$$\pi \geq F$$
$$\sigma^F \geq G.$$

It would be of interest to know whether $\sigma^\pi \geq G$. If so, it would mean that σ invokes instances of a protocol that emulates F, and the composed protocol UC-emulates G. This is what we naturally expect and is what we did, without justification, in the proof sketch of section 3.

To show this, we first quote the Universal Composition Theorem; we do not elaborate further on this theorem.

Theorem 2. [Universal Composition Theorem] Let π, ρ and σ be quantum-polynomial-time protocols. Assume that π everlasting quantum UC emulates ρ. Then σ^π everlasting quantum UC emulates σ^ρ. [1]

In our example,

1. $\pi \geq F$ (given)
2. $\sigma^\pi \geq \sigma^F$ (from Theorem 2, adapted to the non-everlasting, non-quantum setting.)
3. $\sigma^F \geq G$ (given)
4. $\sigma^\pi \geq G$ (from 2 and 3, we have taken transitivity for granted). ■

The UC framework is adapted to the setting of everlasting security in the following way:

If a protocol is everlastingly quantum-UC-secure,

then a quantum machine cannot distinguish between the protocol and an ideal functionality when being polynomial-time during protocol execution, but unlimited afterwards.

We can tweak the requirements on the quantum machine during and after protocol execution to obtain other flavors of quantum UC security; we avoid wordy discussions and instead refer the reader to figure 2.

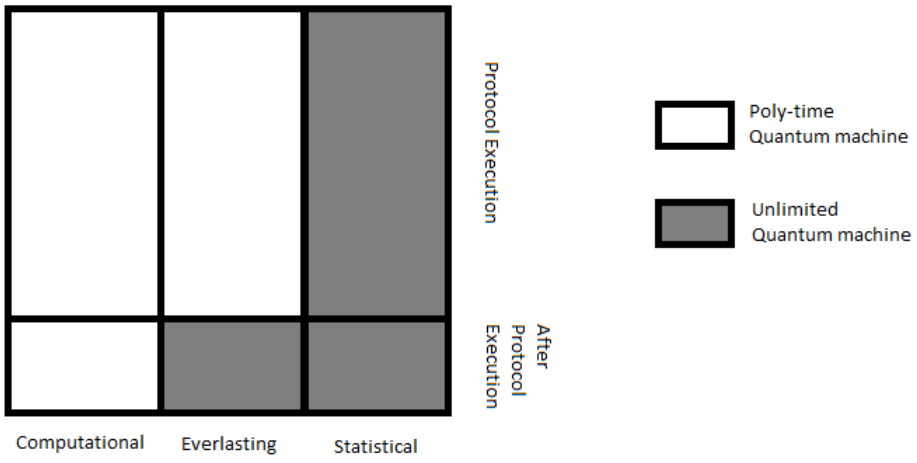


Figure 2: Three variants of quantum UC secure protocols.

In the final section of this report, we will see how a 'commitment protocol' can be made everlasting UC secure. We call such commitments everlasting UC commitments.

5 Everlasting UC Commitments

A commitment is a scheme that allows a party to commit to a chosen value while keeping it hidden to others, with the ability to open the committed value later [3]. We consider a commitment protocol that uses a signature card; a signature card is a trusted device such that the owner of the card can sign messages, everyone can access the public key of the card, and no one (including the owner) can get the secret key. Commitments are of interest

because previous works have shown that it is possible to get everlasting MPC from everlastingly secure commitment protocols.
For a commitment protocol to be everlasting UC secure, we require the following:

1. The protocol needs to be statistically hiding: this means that an adversary (potentially unlimited) who obtains a commitment of a message m, call this $COM(m)$, cannot bruteforce her way into obtaining m.
2. The protocol needs to be computationally binding: this means that a commiter cannot change his mind after committing to a message.
3. The protocol is extractable: this means that a simulator that controls the signature card can 'extract' the message from the commitment.
4. The protocol needs to be equivocal: the simulator in control of the signature card can change the value in the commitment, hence compromising the binding property.

We illustrate our commitment protocol below.

Commit to m: A $\rightarrow$ B: $c := COM(m)$

Proof: I know signature σ on (m, u) s.t. u opens c as m
or I know the secret key of F_{SC}

Open: A $\rightarrow$ B: m

Proof: I know u that opens c as m
or I know the secret key of F_{SC}

In the remainder of this section, we will see how these requirements for everlasting UC security are fulfilled by our commitment protocol.

Consider extractability. We want a simulator to be able to find out the committed message whilst preventing an adversary from bruteforcing her way into finding m from $COM(m)$. To fix this dilemma, we require Avia to sign (m, u) using the signature card; here 'm' is the message and 'u' would be the opening information for $COM(m)$. Let σ be the signature obtained on (m, u). Avia can successfully commit to a message only if she can make a zero knowledge proof (a proof that reveals nothing except the validity of the proven fact) that she knows the signature σ. This now allows the simulator to extract the message: since the simulator controls the signature card, it will have received various (m, u) pairs from the commiter; one of these pairs was signed to give σ. Thus, the simulator can search for such a pair and

open $COM(m)$ using u from that (m, u) pair.

Now consider equivocality. We want to let the simulator change the value of the message in the commitment, but we want to stop the commiter from doing so. Avia can successfully send an 'open' for the commitment if she can make a zero knowledge proof that she knows a u which can open $COM(m)$ as m. To allow the simulator to 'cheat,' we allow the simulator to produce a fake proof; to do so, we add a clause to the zero knowledge proof where the proof also succeeds if the prover can prove that s/he knows the secret key of the signature card. Since only the simulator knows the secret key, it will always succeed, whatever value it committed to, thus allowing equivocality.

And finally, the first two requirements on hiding and binding are also fulfilled by the commitment protocol under examination. The reasoning is hidden in the previous two paragraphs.

This concludes our discussion on commitments. Crépeau and Kilian [4] and Unruh [5] have shown that it is possible to construct an everlastingly secure Oblivious Transfer (OT) protocol from an everlastingly secure commitment protocol. And, Kilian [6] and Unruh [1] have shown that from an everlastingly secure OT protocol, we get **everlasting multi-party computation.**

References

[1] Dominique Unruh, Everlasting Multi-Party Computation. In *Crypto 2013*, volume 8043 of LNCS, pages 380-397. Springer, 2013.

[2] Universal composability. n.d. In *Wikipedia*. Retrieved May 27, 2014, from `http://en.wikipedia.org/wiki/Universal_composability`

[3] Commitment scheme. n.d. In *Wikipedia*. Retrieved May 27, 2014, from `http://en.wikipedia.org/wiki/Commitment_scheme`

[4] Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *FOCS 1988*, pages 42–52. IEEE, 1988.

[5] Dominique Unruh. Universally composable quantum multi-party computation. In *Eurocrypt 2010*, LNCS, pages 486–505. Springer, 2010. Preprint on arXiv:0910.2912 [quant-ph].

[6] Joe Kilian. Founding cryptography on oblivious transfer. In *STOC 1988*, pages 20–31. ACM, 1988.