Abstract

In this overview we provide a condensed and simplified summary of a paper [Unr15] targeting a reader with a limited background in a field of quantum cryptography. General notion of commitment schemes is explained as well as an inapplicability of its classical definition in a quantum setting. Later we provide a new definition of quantum commitments (presented in the paper) called “collapse-binding” outlining its main properties and practical usefulness.

1 Introduction

1.1 What is a commitment scheme?

Commitment scheme is an important cryptographic primitive that has numerous applications in a variety of communication protocols such as coin-flipping, secure computation, zero-knowledge proofs and lately proof-of-stake systems [DGKR17].

A typical commitment is a two-party protocol that consist of two phases: the commit phase and the reveal (open) phase. The purpose of a commitment is to allow the sender to transfer a certain value related to a message \( m \) (commit phase) in such a way that receiver learns nothing about the \( m \) (hiding property) while making it impossible for the sender to change his mind about \( m \) later (binding property). Later the sender reveals \( m \) and proves that it was indeed the message he committed to earlier (reveal phase).

The value sender transmits during the commit phase is called a commitment while during the reveal phase sender usually sends not only \( m \) itself but also a specific value called open. The following diagram summarizes the way the protocol operates on a high-level:

---

*This overview was composed as part of the requirements for a course Research Seminar in Cryptography (MTAT.07.022) held in University of Tartu during Autumn 2017.*
Commitment Scheme (informal)

<table>
<thead>
<tr>
<th>Sender (Prover)</th>
<th>Receiver (Verifier)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Commit Phase</td>
</tr>
<tr>
<td>open ← {0, 1}^*</td>
<td></td>
</tr>
<tr>
<td>(c, open) := Commit(m)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c</td>
</tr>
<tr>
<td>Reveal Phase</td>
<td></td>
</tr>
<tr>
<td>(m, open)</td>
<td></td>
</tr>
<tr>
<td>ok := Verify(c, m, open)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: General commitment scheme

All valid commitment schemes have the following properties:

**Definition 1.1 (Perfect completeness – informal).** A honest recipient always accepts (verifies successfully) a triple (c, m, open) from a honest prover.

**Definition 1.2 (Classical-style binding – informal).** No quantum-polynomial-time algorithm A can output, except with negligible probability, a commitment c (i.e., the message sent during the commit phase) as well as two openings u, u’ that open c to two different messages m, m’.

**Definition 1.3 (Hiding – informal).** No quantum-polynomial-time algorithm A can distinguish, except with negligible probability, commitments c, c’ that were produced for two corresponding messages m, m’ given only commitments.

1.2 Quantum setting definitions

A number of relevant definitions from the field of quantum cryptography are presented here to allow for better understanding of the following material. The below definitions was taken from the lecture notes for the course Quantum Cryptography given by Dominique Unruh in University of Tartu¹.

**Definition 1.4 (Quantum-polynomial-time algorithm – informal).** An algorithm that is executed on a quantum computer and has a running time that is upper-bounded by a polynomial expression in the size of the input is quantum-polynomial-time algorithm.

**Definition 1.5 (Quantum State – informal).** An n-dimensional quantum state is presented by a vector |\phi\rangle \in \mathbb{C}^n with ||\phi|| = 1 (here \mathbb{C}^n is a Hilbert space).

¹The course materials can be found at this web page
Definition 1.6 (Measurement). A (projective) measurement on a Hilbert space $\mathcal{H}$ is specified by a family $\{P_i\}_{i \in I}$ of orthogonal projections on $\mathcal{H}$ labelled with the possible measurement outcomes $iI$. The projections have to be pairwise orthogonal, i.e., $\forall i \neq j \ P_i P_j = 0$. And the projections have sum up to 1, i.e., $\sum_i P_i = 1_\mathcal{H}$ where $1_\mathcal{H}$ is the identity on $\mathcal{H}$.

When measuring a state $|\psi\rangle \in \mathcal{H}$, the outcome $i$ occurs with probability $\|P_i |\phi\rangle\|^2$.

If the outcome $i$ occurs, the state after the measurement (post-measurement state) is $\frac{P_i |\phi\rangle}{\|P_i |\psi\rangle\|}$.

1.3 Quantum setting peculiarities

While there exist a well-established security definition of a binding property for commitment schemes in a classical setting (Definition 1.2) it is proved to be inadequate in a quantum setting [ARU14]. It’s possible to come up with a commitment scheme $C$ that is classically-binding but there exists a quantum-polynomial-time adversary $A$ that first produces a commitment $c$, then expects a message $m$ as input, and then opens $c$ to $m$:

Quantum Attack
1: \quad c \leftarrow C \ # commit to a random commitment
2: \quad m \ # get message as input
3: \quad open := A'(m,c) \ # compute a valid opening

Figure 2: Quantum attack on a classical binding definition

The sub-routine $A'$ of the adversary requires a measurement of its quantum state. After producing a valid opening $open$ the state is destroyed thus it is not possible for the adversary to provide valid openings for two messages simultaneously. Hence the commitment scheme $C$ is classically-binding.

As the adversary can provide a valid opening for any $m$ of his choosing it certainly makes the classical definition useless in a quantum setting.

1.4 Contribution

The main contribution of [Unr15] is a new definition for the computational-binding property for commitment schemes, called “collapse-binding”. While there are other definitions exist (see Section 2.2) the proposed definition has several useful properties (see Section 2.3) that make it the what computationally-binding commitments are in a classical setting.

Additionally the straitened notion of collision-resistant hash functions was given called “collapsing hash functions” which allows to turn several standard construction of commitment schemes to be computationally binding.

\footnote{This fact makes a quantum attack in Section 1.3 possible as the pre-measurement state is destroyed.}
2 Definitions and properties

2.1 Commitment scheme

A commitment scheme \( (\text{com}, \text{verify}) \) consists of a quantum-polynomial-time algorithm \(^3\) \text{com} and a deterministic quantum-polynomial-time algorithm \( \text{verify} \). \((c, u) \leftarrow \text{com}(1^n, m) \) returns a commitment \( c \) and the opening information \( u \) for the message \( m \) and security parameter \( \eta \). \( c \) alone is supposed not to reveal any information about \( m \) (hiding). To open (reveal), we send \((m, u)\) to the recipient who checks whether \( \text{verify}(1^n, c, m, u) = 1 \). Both \text{com} and \text{verify} have classical input and output. \text{com} has a well-defined message space \( \text{MSP}_\eta \) that also depends on the security parameter \( \eta \) (e.g., \( \{0, 1\}^n \)). Furthermore, for technical reasons, we assume that it is possible to find triples \((c, m, u)\) with \( \text{verify}(1^n, c, m, u) = 1 \) with probability 1 in quantum-polynomial-time in \( \eta \).

The basic properties of a commitment scheme are the following:

**Definition 2.1 (Perfect completeness).** \( (\text{com},\text{verify}) \) has perfect completeness iff \( \forall m \in \text{MSP}_\eta, \Pr[\text{verify}(1^n, c, m, u) = 1 : (c, u) \leftarrow \text{com}(1^n, m)] = 1 \).

**Definition 2.2 (Computational hiding).** \( (\text{com},\text{verify}) \) is computationally hiding iff for any quantum-polynomial-time \( A \) and any polynomial \( \zeta \), there is a negligible \( \mu \) such that for any \( \eta \), any \( m_0, m_1 \in \text{MSP}_\eta \) with \( |m_0|,|m_1| \leq \zeta(\eta) \), and any \( |\psi\rangle,|P_0 - P_1| \leq \mu(\eta) \) where \( P_i := \Pr[b = 1 : (c, u) \leftarrow \text{com}(1^n, m_i), b \leftarrow A(1^n, |\psi\rangle, c)] \).

**Definition 2.3 (Statistical hiding).** Like computational hiding, except that we quantify over all \( A \) (not just quantum-polynomial-time \( A \)).

**Definition 2.4 (Classical-style computational binding).** \( (\text{com},\text{verify}) \) is classical-style (computationally) binding iff for any quantum-polynomial-time algorithm \( A \), the following is negligible in \( \eta \) :

\[
\Pr[\text{verify}(1^n, c, m, u) = 1 \land \text{verify}(1^n, c, m', u') = 1 \land m \neq m' : (c, m, u, m', u') \leftarrow A(1^n)]
\]

**Definition 2.5 (Parallel composition (with respect to a property \( \Theta \)).** Let \( (\text{com},\text{verify}) \) be a commitment scheme that has a property \( \Theta \). Let \( n = n(\eta) \) be polynomially-bounded and quantum-polynomial-time computable integer. Let \( (\text{com}^n,\text{verify}^n) \) be the \( n \)-fold parallel composition of \( (\text{com},\text{verify}) \). That is, its message space is \( \text{MSP}^n \). And \( \text{com}^n(m_1, \ldots, m_n) \) computes \((c_i, u_i) \leftarrow \text{com}(m_i) \) for \( i = 1, \ldots, n \), and returns \((c, u) \) with \( c := (c_1, \ldots, c_n) \) and \( u := (u_1, \ldots, u_n) \). And \( \text{verify}^n((c_1, \ldots, c_n), (m_1, \ldots, m_n), (u_1, \ldots, u_n)) = 1 \) iff \( \forall i \text{ verify}(c_i, m_i, u_i) = 1 \).

**2.2 Other binding definitions**

There exist other binding definitions that circumvent the limitations of classical-binding property from Section 1.2 though they introduce certain limitations that motivate a need for a new definition. The following table summaries their properties and limitations:

---

\(^3\)Algorithms are assumed to be quantum for a greater generality. In a practical setting they should be classical instead.
<table>
<thead>
<tr>
<th>Definition</th>
<th>Limitations</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-binding [BCJL93]</td>
<td>- Specific to bit commitments (no straightforward generalization to string commitments case)</td>
<td>Not used in any research as a subprotocol (to our knowledge)</td>
</tr>
<tr>
<td>[May97] [DMS00]</td>
<td>- Unclear behaviour under composition (Definition 2.5)</td>
<td></td>
</tr>
<tr>
<td>[CDMS04] [CSST11]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDMS-binding [CDMS04]</td>
<td>- Parametrized by a specific family of functions that should be chosen for a particular use case</td>
<td>Used in OT protocol from [CDMS04] but hardly applicable in many other contexts</td>
</tr>
<tr>
<td></td>
<td>- Not known whether the definition is composable (according to Definition 2.5)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- No provable construction of CDMS-binding commitments is known</td>
<td></td>
</tr>
<tr>
<td>Perfectly-binding</td>
<td>- Cannot be statistically hiding [May97]</td>
<td>There exist some constructions [BLX11]</td>
</tr>
<tr>
<td></td>
<td>- Cannot be short (length of a commitment must be as long as the length of a committed message)</td>
<td></td>
</tr>
<tr>
<td>UC commitments [Unr10]</td>
<td>- Additional setup is required (e.g. common reference strings)</td>
<td>Construction is given in the original paper [Unr10]</td>
</tr>
<tr>
<td></td>
<td>- Constructions tend to be more complex, less efficient and use stronger computational assumptions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Cannot be short (if using CRS)</td>
<td></td>
</tr>
<tr>
<td>Q-binding [DFS04]</td>
<td>- Limited composability (follows directly from the definition)</td>
<td>The only known construction uses trapdoor thus needs stronger assumptions</td>
</tr>
<tr>
<td></td>
<td>- Unclear how definition can be used in rewinding-based proofs</td>
<td></td>
</tr>
<tr>
<td>DFRSS-binding [DFR+07]</td>
<td>- Definition was intended for protocols in the bounded quantum storage model (adversary possesses limited amount of memory). It’s unclear how the definition would behave in a standard model (without memory limitation)</td>
<td>Construction is given in the original paper</td>
</tr>
<tr>
<td></td>
<td>- Cannot be statistically hiding</td>
<td></td>
</tr>
</tbody>
</table>

| Table 1: Existing binding definitions and their limitations |

2.3 Collapse-binding commitments

**Definition 2.6 (Collapse-binding).** For algorithms $A, B$, consider the following games:

$$
\text{Game}_1: \quad (S, M, U, c) \leftarrow A(1^n), \quad ok \leftarrow V_c(M, U), \quad m \leftarrow M_{ok}(M), \quad b \leftarrow B(1^n, S, M, U)
$$

$$
\text{Game}_2: \quad (S, M, U, c) \leftarrow A(1^n), \quad ok \leftarrow V_c(M, U), \quad b \leftarrow B(1^n, S, M, U)
$$

Here $S, M, U$ are quantum registers. $V_c$ is a measurement whether $M, U$ contains a valid opening, formally $V_c$ is defined through the projector $\sum_{m,u,\text{verify}(1^n,c,m,u)=1} |m\rangle \langle m| \otimes |u\rangle \langle u|$. 

\( M_{ok} \) is a measurement of \( M \) in the computational basis if \( ok=1 \), and does nothing if \( ok=0 \) (i.e., it sets \( m := \perp \) and does not touch the register \( M \)).

A commitment scheme is collapse-binding iff for any quantum-polynomial-time algorithms \( A, B \) the difference \( |\Pr[b = 1 : \text{Game}_1] - \Pr[b = 1 : \text{Game}_2]| \) is negligible.

The following are the properties of collapse-binding definition:

1. Holds for a partial measurement
2. Allows for a parallel composition (satisfies Definition 2.5 where \( \Theta \) is a collapse-binding property)

### 2.4 Collapsing hash functions

In practical constructions of commitment schemes collision-resistant hash functions are usually used as a commit function:

\[
    c := H(m||u), \quad u \leftarrow U
\]

**Definition 2.7 (Cannonical commitment scheme).** Given a hash function \( H \) and a parameter \( l_u = l_u(\eta) \), the canonical commitment scheme for \( H \) is:

- **Message space** \( MSP_\eta := \{0,1\}^* \).
- **commit\_can** \((m)\): Pick \( u \leftarrow \{0,1\}^{l_u} \). Compute: \( c := H(m||u) \). Return \((c,u)\).
- **verify\_can** \((c,m,u)\): Return 1 iff \( H(m||u) = c \).

Though it is not enough for a commitment scheme to be collapse-binding [ARU14]. This notion is summarized in the following theorem:

**Theorem 1 (Attack on collision-resistance\(^4\)).** There is an oracle \( \theta \) relative to which there exists a collision-resistant hash function \( H \) such that the canonical commitment scheme and both Halevi-Micali commitment schemes [HM96] using \( H \) admit the following attack:

There is a quantum-polynomial-time adversary \( A^\theta \) that outputs a commitment \( c \), then expects a bit \( b \), and then outputs with overwhelming probability a pair \((m,u)\) such that \( \text{verify}(c,m,u) = 1 \) and the first bit of \( m \) is \( b \).

As collision-resistance is not a sufficiently strong property in a quantum setting in [Unr15] the strengthened definition is proposed:

**Definition 2.8 (Collapsing hash function).** For a function \( H \) and algorithms \( A, B \), consider the following games:

\[
    \text{Game}_1: (S,M,c) \leftarrow A(1^\eta), \quad m \leftarrow M_{\text{comp}}(M), \quad b \leftarrow B(1^\eta, S,M)
\]

\[
    \text{Game}_2: (S,M,c) \leftarrow A(1^\eta), \quad b \leftarrow B(1^\eta, S,M)
\]

Here \( S,M \) are quantum registers. \( M_{\text{comp}}(M) \) is a measurement of \( M \) in the computational basis.

We call an adversary \((A,B)\) valid if \( \Pr[H(m) = c] = 1 \) when we run \((S,M,c) \leftarrow A(1^\eta)\) and measure \( M \) in the computational basis as \( m \).

A function \( H \) is collapsing iff for any quantum-polynomial-time valid adversary \((A,B)\), the difference \( \text{adv} := |\Pr[b = 1 : \text{Game}_1] - \Pr[b = 1 : \text{Game}_2]| \) is negligible. (We call \( \text{adv} \) the advantage.)

---

\(^4\) \( H \) is collision-resistant iff for any quantum-polynomial-time \( A \), \( \Pr[x \neq x' \land H(x) = H(x') : (x, x' \leftarrow A(1^\eta))] \) is negligible.
The following are the properties of collapsing hash functions:

1. Definition holds for a partial measurement
2. An injective hash function $H$ is collapsing with advantage 0
3. A collapsing hash function is collision resistant
4. If $f$ and $g$ are collapsing, so is $g \circ f$
5. Random oracles are collapsing

2.5 Commitments from collapsing hash functions

It was shown in an original paper that given a collapsing hash function it’s fairly trivial to construct a collapse-binding commitment. Collapsing hash functions is proved to be a drop-in replacement for collision resistant hash function. The following definitions sum up this statements:

**Corollary 2.1 (Collapse-binding commitment from a collapsing function).** Let $f$ be a collapsing hash function. Let $(\text{com}, \text{verify})$ be a collapse binding commitment scheme. Let $\text{com}_f(1^n, f(m))$ and $\text{verify}_f(1^n, c, m, u) = \text{verify}(1^n, c, f(m), u)$. Then $(\text{com}_f, \text{verify}_f)$ is a collapse-binding commitment scheme.

**Lemma 2.1 (Cannonical commitment from collapsing hash functions).** If $H$ is collapsing, then the canonical commitment scheme $(\text{com}_{can}, \text{verify}_{can})$ is collapse binding.

In order to use collapse-binding commitments in practice there is a need to have a practical collapsing hash function. Later in the paper it’s shown that SHA-3 hash function is collapsing which makes it a considerable candidate for practical implementations.

3 Conclusion

Collapse-binding commitments make discovered quantum attacks impossible under a new definition and provide clear guarantees for a parallel composition without restricting a scheme from being statistically hiding (in contrast to perfectly-binding commitments).

A stronger security notion of collapsing hash functions allow to create practical collapse-binding commitments expanding the message space while being a drop-in replacement for collision-resistant hash functions. SHA-3 is proved to be collapsing which means that more secure commitment schemes can be be already constructed and adopted.

References


---

5 This is a shortened version of Lemma 31 from the original paper. Collapse-binding of Halevi-Micali commitment [HM96] was omitted to save space.


