On Possibility of Universally Composable Commitments Based on Noisy Channels

Rafael Dowsley[1], **Jörn Müller-Quade**[2], **Anderson C. A. Nascimento**[1]

[1]Department of Electrical Engineering, University of Brasilia.
Campus Universitario Darcy Ribeiro, Brasilia, CEP: 70910-900, Brazil.

[2]Universitaet Karlsruhe, Institut fuer Algorithmen und Kognitive Systeme.
Am Fasanengarten 5, 76128 Karlsruhe, Germany

rafaeldowsley@redes.unb.br, muellerq@ira.uka.de, andclay@ene.unb.br

Abstract. *Universally Composable (UC) Commitment is a strong notion that guarantees security even when the commitment protocol is composed with arbitrary protocols running many of their copies in parallel. It is impossible to implement a protocol that realizes UC Commitment without set-up assumptions. However, it has been implemented using such assumptions as common reference string, certified public keys and random oracles. In this paper we prove that the existence of a binary symmetric channel between the parties makes possible the accomplishment of UC Commitment.*

1. Introduction

Commitment is one of the most fundamental cryptographic protocols. It is used as a sub-protocol in applications such as secure multi-party computation [GMW87], contract signing [EGL85] and zero-knowledge proofs [GMW91, Gol01, BCC88]. A commitment protocol involves two players: the committer and the receiver. The idea behind the notion of commitment is simple: the committer provides the receiver with a digital equivalent of a "sealed envelope". This envelope should contain a value x in the commitment phase of the protocol. Before the committer helps the receiver in opening the envelope, the receiver should learn nothing about the value x. Additionally, the committer should not be able to change x after the commitment phase. When the committer helps the receiver in opening the envelope in the decommitment phase, the receiver learns the value x.

A very large number of commitment protocols are known based on various assumptions in the standalone setting, but this notion does not guarantee security when multiple copies of the protocol run at the same time, or when the commitment protocols are used within other protocols. Universally Composable (UC) Commitment [Can05, CF01] is a notion of security that holds even when the commitment scheme is concurrently composed with an arbitrary set of protocols.

This notion of security is so strong that is impossible to obtain UC commitment protocol if no set-up assumption is provided [CF01]. UC commitment protocols were constructed in the common reference string (CRS) model [CF01, CLOS02, DN02, DG03]. In the CRS model there exists an honestly generated random string at the system initialization; the simulator can generate its own string (as long as it looks indistinguishable from the honestly generated one). Barak et. al. [BCNP04] show how to make the above schemes work in the key set-up model in the presence of a static adversary. In this model, parties have certified public keys. Dodis et. al. [DPW05] extend these results to adaptive corruptions. A UC commitment protocol was constructed in the random oracle model by Hofheinz and Müller-Quade [HM04]. Prabhakaran and Sahai [PS04] introduced a model in which all the parties, the adversary and the simulator are given oracle access to super-polynomial angels. In this model one can securely implement any multiparty functionality without setup assumptions.

The current work introduces a new set-up assumption. We prove that there exists a UC Commitment protocol secure against computationally unbounded adaptive adversaries based on the existence of noisy channels. Specifically, we prove that a binary symmetric channel is a valid set-up assumption to construct a UC commitment protocol.

The potential of the noisy channel for cryptography purposes was first used by Wyner [Wyn75] for exchanging a secret key in the presence of an eavesdropper. Later Maurer [Mau93], Ahlswede and Csiszár [AC93] extended the results. Crépeau and Kilian [CK88] implemented the first bit commitment based on noisy channel that is information information theoretically secure. Their idea was later improved and extended in [Cre97, DKS99, INW03].

2. Preliminaries

We present some useful notions and techniques that we will use in this paper.

2.1. Coding Theory

A binary error-correcting code $\mathcal{C}$ is $[n, k, d]$ linear if it has codeword length n, dimension k and minimal distance d and it is a linear subspace of cardinality 2^k of $\{0, 1\}^n$ such that for any two distinct words $c_1, c_2 \in \mathcal{C}$ $d_H(c_1, c_2) \geq d$ holds, where d_H is the Hamming distance between the two word (i.e., the number of bits that they differ). This type of code is specified by a generating matrix G of dimension $k \times n$ or by a parity check matrix H of dimension $n \times (n - k)$. The following theorem appears in [MS77, Ch. 17, Prob. 30] and shows that random linear codes achieve the Gilbert-Varshamov bound:

Theorem 1 *There exists a constant $\rho > 1$ such that a random binary matrix G of size $Rn \times n$ defines a binary linear code with minimal distance at least γn except with probability not greater than $\rho^{(R-C_\gamma)n}, C_\gamma = 1 - H(\gamma)$, where $H(\gamma)$ is the binary entropy function and $R < C_\gamma$.*

We also need a theorem from [GI03]:

Theorem 2 *There is a probabilistic polynomial time procedure to construct codes whose rate vs. distance trade-off meets the Gilbert-Varshamov bound with high probability for all rates less than 10^{-4}. Furthermore, these codes can be decoded in polynomial time up to half the relative distance, and in fact this latter decoding property can be "certified", i.e., one can verify in deterministic polynomial time that such decoding will indeed be possible for the constructed code.*

2.2. Statistical Indistinguishability

A function f mapping non-negative integers to non-negative reals is called *negligible* if for all positive numbers c, there exists an integer n_0 such that for all $n > n_0$, we have $f(n) < 1/n^c$.

Definition 1 *Two sequences $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ of random variables are called* statistically indistinguishable *if*

$$\frac{1}{2} \cdot \sum_{s\in S_n} |Pr[X_n = s] - Pr[Y_n = s]|$$

is negligible, where S_n is the union of the supports of X_n and Y_n.

3. UC Commitment Definitions

3.1. The General Framework

This section summarizes the parts of the Universally Composable framework [Can05] that are relevant to this work. The security of a protocol to carry out a task is established in three phases. First, we should formalize the process of executing a protocol in the presence of an adversary and an environment. Next, we should formalize an ideal protocol for carrying out the task using a "trusted party". In the ideal protocol the trusted party captures the requirements of the desired task and the parties cannot communicate among themselves. Finally, we prove that the real protocol emulates the ideal protocol.

The environment in the UC framework represents all activity external to the running protocol, so it provides inputs to the parties running the protocol and receives the outputs that the parties generate during the execution of the protocol. The environment tries to distinguish between attacks on real executions of the protocol and simulated attacks against the ideal functionality. If no environment can distinguish the two situations, the real protocol emulates the ideal functionality.

The computational model. The computational model extends the interactive Turing machine (ITM) model [GMR89, Gol01]. The programs run by parties are represented as Turing machines with shared tapes. Specifically, the input and output tapes model inputs and outputs that are received from and given to other programs running on the same machine, and the communication tapes model messages sent to and received from the network.

An ITM instance (ITI) is an instance of a program running on specific data. A system (I, C) of ITMs consists of an initial ITM I and a control function C. An instance of I is invoked when the execution of the system starts. Each ITI can invoke other ITIs and write messages on some tapes of the others ITIs, the control function C determines which tapes of which ITIs can be written by the ITI. In addition, each ITI has a unique ID that specifies two fields: the session ID (SID) and the party ID (PID). At any time only one ITI is active, it can execute its code and also write only once to the tape of other ITI. The output of an execution of a system is the output of I, and the execution of the system ends when I halts. Adversarial entities are also modeled as ITMs.

An ITM M is locally probabilistic polynomial time (PPT) if, at any point during the execution of any ITI μ with code M, the overall running time so far is bounded by a polynomial in the security parameter and the overall length of input, and in addition the number of bits written on the input tapes of other ITIs, plus the number of other ITIs invoked by μ, is less than the length of μ's input so far.

The adversarial model. The parties have unique identities and are locally PPT. The network is asynchronous without guaranteed delivery of messages. The communication is public, but authenticated (i.e., we assume that an authentication functionality is available to all players). The adversary is adaptive in corrupting parties, and is active in its control over corrupted parties. Any number of parties can be corrupted. Finally, the adversary, the environment and the simulator are allowed unbounded complexity.

Protocol execution in the real-life model. We sketch the process of executing a given protocol π (run by parties $P_1, \ldots, P_n$) with some adversary $\mathcal{A}$ and an environment machine $\mathcal{Z}$ with input z. The model of executing π is the extended system of ITMs ($\mathcal{Z}$,$C^{\pi,\mathcal{A}}_{EXEC}$), where $\mathcal{Z}$ is the initial environment and $C^{\pi,\mathcal{A}}_{EXEC}$ is the control function.

The first ITI to be invoked by $\mathcal{Z}$ is set by the control function to be $\mathcal{A}$. $\mathcal{Z}$ can also invoke an unlimited number of ITIs, give inputs to them, and receive outputs from them, but $\mathcal{Z}$ can only invoke ITIs with the same SID and the code of these ITIs is set by $C^{\pi,\mathcal{A}}_{EXEC}$ to be the code of π. $\mathcal{Z}$ can communicate only with the above ITIs.

Parties and sub-parties of π can invoke ITIs and pass inputs and outputs to other ITIs of the same instance of π. Parties of π can also pass outputs to the environment. They can also write messages on the incoming communication tape of the adversary. These messages may specify the identity of the final destination of the message. $\mathcal{A}$ can send messages to any ITI ($\mathcal{A}$ delivers the message).

$\mathcal{A}$ cannot invoke ITIs as subroutines (it can invoke new ITIs by delivering messages to them), but it can corrupt parties or sub-parties of π. After receiving a special message (corrupt id) from the environment, the adversary corrupts a party or sub-party by delivering the message (corrupt). By the definition of the process of corrupting, the environment always knows which parties are corrupted.

Let $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z, \overrightarrow{r})$ denote the output of environment $\mathcal{Z}$ when interacting with adversary $\mathcal{A}$ and parties running protocol π on security parameter k, input z and random input $\overrightarrow{r} = r_{\mathcal{Z}}, r_{\mathcal{A}}, r_1 \ldots r_n$ as described above (z and $r_{\mathcal{Z}}$ for $\mathcal{Z}$, $r_{\mathcal{A}}$ for $\mathcal{A}$; r_i for party P_i). Let $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z)$ denote the random variable describing $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z, \overrightarrow{r})$ when $\overrightarrow{r}$ is uniformly chosen. Let $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ denote the ensemble $\{\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z)\}_{k\in\mathbb{N},z\in\{0,1\}^*}$.

Ideal protocols. An ideal functionality $\mathcal{F}$ represents the desired properties of a given task. Conceptually, $\mathcal{F}$ is treated as a local subroutine by the several parties that use it, and so the communication between the parties and $\mathcal{F}$ is supposedly secure (i.e., messages are sent by input and output tapes). Therefore, $\mathcal{F}$ is an ITM with input tape that many ITIs can write on it and $\mathcal{F}$ can write on the subroutine output tapes of multiple ITIs. The PID of $\mathcal{F}$ is set to $\perp$, and it expects that all inputs come from ITIs with SID equal to its. Finally, $\mathcal{F}$ can communicate with the adversary by using its communication tape and it is responsible for determining the effects of corrupting. In subsections 3.3 and 3.2 we define the ideal functionalities for commitment and binary symmetric channel, respectively, that we use in this work.

The ideal protocol for an ideal functionality $\mathcal{F}$ ($\mathtt{IDEAL}_{\mathcal{F}}$) involves an ideal protocol adversary $\mathcal{S}$, an environment $\mathcal{Z}$ on input z and a set of dummy parties that interacts as defined below. Whenever a dummy party is activated with input x, it writes x onto the input tape of $\mathcal{F}_{(sid,\perp)}$. Whenever the dummy party is activated with value x on its subroutine output tape, it writes x on the subroutine output tape of $\mathcal{Z}$. The ideal protocol adversary $\mathcal{S}$ has no access to the contents of messages sent between dummy parties and $\mathcal{F}$, and it should send corruption messages directly to $\mathcal{F}$ that is responsible for determining the effects of corrupting any dummy party. The ideal functionality receives messages from the dummy parties by reading its input tape and sends messages to them by writing to their subroutine output tape. In the ideal protocol there is no communication among the parties using the adversary to deliver the message.

Let $\texttt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z, \overrightarrow{r})$ denote the output of environment $\mathcal{Z}$ after interacting with adversary $\mathcal{S}$ and ideal functionality $\mathcal{F}$ in the ideal protocol, on security parameter k, input z, and random input $\overrightarrow{r} = r_{\mathcal{Z}}, r_{\mathcal{S}}, r_{\mathcal{F}}$ as described above (z and $r_{\mathcal{Z}}$ for $\mathcal{Z}$, $r_{\mathcal{S}}$ for $\mathcal{S}$, $r_{\mathcal{F}}$ for $\mathcal{F}$). Let $\texttt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z)$ denote the random variable describing $\texttt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z, \overrightarrow{r})$ when $\overrightarrow{r}$ is uniformly chosen. Let $\texttt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ denote the ensemble $\{\texttt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z)\}_{k\in\mathbb{N}, z\in\{0,1\}^*}$.

Realizing an ideal functionality. We say that a protocol π statistically UC-realizes an ideal functionality $\mathcal{F}$ if for any real-life adversary $\mathcal{A}$ there exists an ideal-protocol adversary $\mathcal{S}$ such that no environment $\mathcal{Z}$, on any input z, can tell with non-negligible probability whether it is interacting with $\mathcal{A}$ and parties running π in the real-life process, or it is interacting with $\mathcal{S}$ and $\mathcal{F}$ in the ideal protocol. This means that, from the point of view of the environment, running protocol π is statistically indistinguishable of interacting with an ideal protocol for $\mathcal{F}$.

Definition 2 *Let $n \in \mathbb{N}$. Let $\mathcal{F}$ be an ideal functionality and let π be an n-party protocol. We say that π statistically UC-realizes $\mathcal{F}$ if for any adversary $\mathcal{A}$ there exists an ideal-protocol adversary $\mathcal{S}$ such that for any environment $\mathcal{Z}$ we have that $\texttt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ and $\texttt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ are statistically indistinguishable.*

Hybrid protocols. In hybrid protocols in addition to sending messages to other parties using the adversary to deliver them in the usual way, the parties can also use instances of ideal functionalities. This is done by calling the corresponding instances of the ideal protocol for these functionalities invoking dummy parties for $\mathcal{F}$, which in turn invoke an instance of $\mathcal{F}$ (i.e. in an $\mathcal{F}$-hybrid protocol the parties can include subroutine calls to $\texttt{IDEAL}_{\mathcal{F}}$). Each copy of $\mathcal{F}$ is identified via a session identifier (SID) chosen by the parties of the $\mathcal{F}$-hybrid protocol. The communication between the dummy parties and $\mathcal{F}$ mimics the ideal protocol.

Universal Composition. Let π be a protocol that makes subroutine calls to $\mathcal{F}$, and let ρ be a protocol that statistically UC-emulates $\mathcal{F}$. The composed protocol $\pi^{\rho/\mathcal{F}}$ is constructed by modifying the code of each ITM in π so that messages sent to each dummy party of $\mathcal{F}$ with identity (sid, pid) in protocol π are replaced with messages sent to a copy of ρ with the same identity (sid, pid) in the protocol $\pi^{\rho/\mathcal{F}}$. Each output value generated by a copy of ρ with identity (sid, pid) is treated as a message received from the corresponding dummy party of $\mathcal{F}$ with identity (sid, pid) in protocol π.

A protocol ρ is subroutine respecting if the only input/output interface between each instance of ρ and other protocol instances is done by the actual parties of ρ (i.e., the sub-parties of ρ exchange input/output only with parties or sub-parties of this instance.).

The composition theorem basically says that if ρ statistically UC-emulates protocol $\mathcal{F}$ then an execution of the composed protocol $\pi^{\rho/\mathcal{F}}$ "emulates" an execution of protocol π.

Theorem 3 *Let ρ be a protocol that statistically UC-realizes ideal functionality $\mathcal{F}$ and ρ is subroutine respecting. Then protocol $\pi^{\rho/\mathcal{F}}$ statistically UC-emulates protocol π.*

A specific corollary of the composition theorem states that if π statistically UC-realizes some functionality $\mathcal{G}$ in the $\mathcal{F}$-hybrid model, and ρ statistically UC-realizes $\mathcal{F}$ in the real-life model, then $\pi^{\rho/\mathcal{F}}$ statistically UC- realizes $\mathcal{G}$ in the real-life model.

Corollary 4 *Let $\mathcal{F}$, $\mathcal{G}$ be ideal functionalities. Let π be a subroutine respecting protocol that statistically UC-realizes $\mathcal{G}$ in the $\mathcal{F}$-hybrid model and let ρ be a protocol that statistically UC-realizes $\mathcal{F}$. Then protocol $\pi^{\rho/\mathcal{F}}$ statistically UC-realizes $\mathcal{G}$.*

3.2. The Binary Symmetric Channel Model

The Binary Symmetric Channel (BSC) model is the hybrid model in which the participants have ideal access to a Binary Symmetric Channel with error probability ϵ. Below we describe the functionality $\mathcal{F}_{BSC,\epsilon}$, where P is the sender and R the receiver.

1. Upon receiving an input (Send, sid, b) from P, verify that $sid = (P, R, sid')$ for some R and that $b \in \{0, 1\}$, else ignore the input. Next, choose a random bit r such that Pr$[r = 1] = \epsilon$ and output (Sent, sid, b') to R with $b' = b \oplus r$.

In the ideal process for the functionality $\mathcal{F}_{COM}$ (described in section 3.3) the BSC is not used, so the ideal protocol adversary (simulator) that simulates a real-life adversary can play the role of $\mathcal{F}_{BSC,\epsilon}$ for the simulated adversary.

3.3. The Commitment Functionality

We present the ideal bit commitment functionality as described in [Can05] (a modified version of the first formalized functionality in [CF01]). The functionality is similar to the idea of a "sealed envelope" containing a value x. Before the committer helps the receiver in opening the envelope, the receiver learns nothing about the value x. But the sender cannot change the value after the commitment phase. When the sender helps the receiver in opening the envelope in the decommitment phase, the receiver learns the value x. Below we describe the functionality $\mathcal{F}_{COM}$.

1. Upon receiving an input (Commit, sid, x) from P, verify that $sid = (P, R, sid')$ for some R, else ignore the input. Next, record x and generate a public delayed output (Receipt, sid) to R. Once x is recorded, ignore any subsequent Commit inputs.
2. Upon receiving an input (Open, sid) from P, proceed as follows: If there is a recorded value x then generate a public delayed output (Open, sid, x) to R. Otherwise, do nothing.

3. Upon receiving a message (Corrupt-committer, sid) from the adversary, send x to the adversary. Furthermore, if the adversary now provides a value x', and the Receipt output was not yet written on R's tape, then change the recorded value to x'.

The commitment phase is modeled in item 1 of the functionality in which the $\mathcal{F}_{COM}$ receives the value committed to, records the value and send a public delayed output to the receiver to notify that a commitment was received (i.e. the message is first sent to the adversary, and later sent to the receiver when the confirmation from the adversary is received). The sid must contain the identities of the committer and receiver.

The opening phase takes place when the committer sends a message to $\mathcal{F}_{COM}$ to open the commitment as indicated in item 2 of $\mathcal{F}_{COM}$. If the committer has already recorded a value then a public delayed output with the value is generated to the receiver.

Item 3 of the functionality models the response when the adversary corrupts some party. The $\mathcal{F}_{COM}$ sends the recorded value to the adversary and lets him modify the value if the Receipt message was not yet written to the receiver's tape.

4. UC Commitment Based on Noisy Channel

We now state that there exists a hybrid protocol using the ideal functionalities for binary symmetric channel ($\mathcal{F}_{BSC,\epsilon}$) and authenticated communication ($\mathcal{F}_{AUTH}$ [Can05]) that statistically UC-realizes the ideal functionality $\mathcal{F}_{COM}$:

Theorem 5 *It is possible to statistically UC-realize $\mathcal{F}_{COM}$ with an ($\mathcal{F}_{AUTH}$,$\mathcal{F}_{BSC,\epsilon}$)-hybrid protocol.*

Specifically we prove the following lemma:

Lemma 6 *The bit commitment protocol based on noisy channels proposed in [Cre97] statistically UC-realizes $\mathcal{F}_{COM}$ in the binary symmetric channel model.*

4.1. The Protocol

This section describes the Commitment Protocol based on Noisy Channels that was proposed by Crépeau [Cre97] adapted to the UC framework.

Let $0 < \epsilon < 1/2$ be the error probability of the binary symmetric channel, and γ and δ be positive numbers.

Commitment Phase

1. When R starts its execution, it chooses at random and sends to P a binary linear $[n, k, d]$-code $\mathcal{C}$ with parameters $k/n = 1 - H(\gamma)$.
2. P:
    - picks random n-bit string m and sends it to R,
    - picks random codeword $c \in \mathcal{C}$ such that $c \odot m = b$ (where $\odot$ denotes the scalar product),
    - send the message (Send, sid, c_i) to $\mathcal{F}_{BSC,\epsilon}$ for each $1 \leq i \leq n$. R receives $c' = c'_1 c'_2 \ldots c'_n$
3. Upon receiving m and c', R outputs (Receipt, sid)

After commitment phase the two parties keep their outputs secret. If the sender decides to unveil the bit he starts decommitment phase below.

Decommitment Phase

1. P sends b and c to R
2. if $(c \in \mathcal{C}) \wedge (b = c \odot m) \wedge (d_H(c, c') < \epsilon n + \lambda n)$ then R accepts and outputs (Open, sid, x), else he rejects.

The parties use the public authenticated network to send the noiseless messages.

We briefly argue about the protocol stand-alone security (see [Cre97] for details). According to Theorem 1, the code chosen by Alice has minimal distance at least γn but for a negligible probability. Any string z (not necessarily a codeword) is at a distance smaller than $(\gamma/2)n$ from at most one codeword, lets call it c. If Alice unveils a codeword different from c, the expected hamming distance between z and this other codeword will be at least $(\gamma/2 + \epsilon)n$ but for a negligible probability. Therefore, by choosing an appropriate λ, the cheating probability for Alice is negligible in n.

From the strong law of large numbers, the protocol always work for honest Alice and Bob but for a negligible probability.

Finally, from the left-over hash lemma [ILL89], one sees that Bob's views in the case Alice commits to a zero or a one are indistinguishable.

4.2. UC Security of the Protocol

We construct the ideal-protocol adversary $\mathcal{S}$ as follows. $\mathcal{S}$ runs a simulated copy of $\mathcal{A}$ in a black-box way, plays the role of the ideal functionality $\mathcal{F}_{BSC,\epsilon}$ and simulates a copy of the hybrid interaction of π for the simulated adversary $\mathcal{A}$. In addition, $\mathcal{S}$ forwards all inputs from $\mathcal{Z}$ to $\mathcal{A}'s$ input and all outputs from $\mathcal{A}$ to $\mathcal{Z}$. $\mathcal{S}$ should be able to extract the committed value from the messages that it receives from $\mathcal{A}$ if the sender is corrupted and also should be able to send a commitment in the hybrid interaction and later open it to both values. Below we describe the procedures of the simulator in each occasion:

1. In the beginning of the simulation with $\mathcal{A}$ and $\mathcal{Z}$, $\mathcal{S}$ chooses a random binary linear $[n, k, d]$-code $\mathcal{C}$ with parameters $k = (1 - H(\gamma) + \delta)n$ and $d \geq \gamma\epsilon n$ and sends it in the simulated hybrid execution as the error correcting code that the receiver selected. We can invoke theorem 1.
2. If the environment $\mathcal{Z}$ writes a message (Commit, sid, b) on the input tape of an uncorrupted party P, P copies the message to the functionality $\mathcal{F}_{COM}$ and $\mathcal{S}$ is informed about the commitment. Then, $\mathcal{S}$ sends a random n-bit string m to $\mathcal{A}$. If the receiver is corrupted, $\mathcal{S}$ chooses a random codeword and simulates the received c' and sends it to $\mathcal{A}$ (playing the role of $\mathcal{F}_{BSC,\epsilon}$). If the receiver is honest and $\mathcal{A}$ delivers m to the simulated receiver, $\mathcal{S}$ allows $\mathcal{F}_{COM}$ to output (Receipt, sid) to the receiver in the ideal protocol.
3. If $\mathcal{Z}$ writes a message (Open, sid) on the input tape of some uncorrupted party P, P copies the message to the functionality $\mathcal{F}_{COM}$. If P has previously committed to a value b, $\mathcal{S}$ will receive the bit b. If the receiver is corrupted, $\mathcal{S}$ that also knows c' and m tries to find $c \in \mathcal{C}$ such that the receiver will accept c in the test performed on step 2 of Decommitment phase. If the receiver is uncorrupted, $\mathcal{S}$ tries to find c such that $b = c \odot m$. If $\mathcal{S}$ finds $c \in \mathcal{C}$ according to the above criteria, it sends c

and b to the adversary $\mathcal{A}$ in the role of the uncorrupted sender; otherwise it stops. If the receiver is honest and $\mathcal{A}$ delivers b and c to the simulated receiver, $\mathcal{S}$ allows $\mathcal{F}_{COM}$ to output (Open, sid, b) to the receiver in the ideal protocol.

4. If $\mathcal{A}$ lets some corrupted party P commit to a bit b, $\mathcal{S}$ views c. If the receiver is also corrupted, $\mathcal{S}$ just simulates the received c'. If the receiver is uncorrupted, $\mathcal{S}$ views also m. If c is a codeword, then $\mathcal{S}$ can compute the value of b and send the message (Commit, sid, b) to $\mathcal{F}_{COM}$. In the case c is not a codeword, $\mathcal{S}$ tries to find $\hat{c}$ such that $d_H(\hat{c}, c) < \gamma n/2$. Note that there exist no more than one codeword $\hat{c}$ satisfying $d_H(\hat{c}, c) < \gamma n/2$. If $\mathcal{S}$ finds it, then $\mathcal{S}$ computes $b = \hat{c} \odot m$ and sends the message (Commit, sid, b) to $\mathcal{F}_{COM}$; otherwise $\mathcal{S}$ sends a random bit b.
5. If $\mathcal{A}$ tells some corrupted party P to open a valid commitment with bit b', codeword c^* and the receiver is not corrupted, then $\mathcal{S}$ simulates the received c' (according to c that it knows) and checks if an honest receiver would accept b' and c^* as valid opening information for commit. If an honest user would reject it, then $\mathcal{S}$ stops; otherwise $\mathcal{S}$ sends (Open, sid) to $\mathcal{F}_{COM}$.
6. If $\mathcal{A}$ corrupts the sender, then $\mathcal{S}$ corrupts the sender in the ideal protocol and learns b. In the case that the (Receipt, sid) output was written on the receiver tape before the corruption, the adversary knows m, $\mathcal{C}$ and possibly c' (only if the receiver is already corrupted). $\mathcal{S}$ tries to find $c \in \mathcal{C}$ such that $c \odot m = b$ and such that if the receiver is corrupted c will be accepted in the receiver's test. If it finds such c, it sends c and b to $\mathcal{A}$; otherwise it stops. If the (Receipt, sid) output was not written on the receiver tape before the corruption, then $\mathcal{A}$ has not yet delivered m and c has not yet been sent to $\mathcal{F}_{BSC,\epsilon}$. $\mathcal{A}$ can thus change b, m and c. $\mathcal{S}$ follows the procedures of item 4.
7. If $\mathcal{A}$ corrupts the receiver, then $\mathcal{S}$ corrupts the receiver in the ideal protocol. If the receiver is corrupted before $\mathcal{A}$ delivers $\mathcal{C}$ to the simulated sender, $\mathcal{A}$ could change $\mathcal{C}$. If the receiver is corrupted after the commitment and before the opening, $\mathcal{S}$ plays the role of $\mathcal{F}_{BSC,\epsilon}$ and thus it can send a valid c' to $\mathcal{A}$ (i.e., random n-bit string if the sender is not corrupted, otherwise it simulates $BS_\epsilon(c)$). If the receiver is corrupted after the opening phase, $\mathcal{S}$ also learns b and can thus send also c and b to $\mathcal{A}$.

We analyze below the probabilities of the events that can result in different views $\texttt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ and $\texttt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$.

Now we prove that in items 3 and 6 of the simulator, $\mathcal{S}$ can find a codeword c so that the receiver accepts it in the test performed on step 2 of Decommitment phase with overwhelming probability. The density of codewords in the random binary linear code is

$$\begin{aligned} \frac{2^{nR}}{2^n} &= 2^{n(1-H(\gamma)+\delta-1)} \\ &= 2^{n(\delta-H(\gamma))} \end{aligned}$$

Knowing that there are no less than $2^{n(H(\epsilon)-\delta')}$ jointly typical words with c' but for a negligible probability (this follows from the Asymptotic Equipartition Property [CT91]), the expected number of jointly typical codewords for the received c' in the random binary linear code is

$$2^{n(\delta-H(\gamma))}2^{n(H(\epsilon)-\delta')} = 2^{n(\delta-\delta'-H(\gamma)+H(\epsilon))}$$

Setting $\delta > \delta'$ and $\gamma < \epsilon$ and applying the Chernoff bound [Che52], the number of jointly typical codewords is exponential in n with overwhelming probability. Thus, $\mathcal{S}$ only stops in these items with negligible probability. In item 3 of the simulator, $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ and $\mathtt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ differ also if an honest committer in the hybrid interaction is unable to open a valid commitment, but this probability is exponentially small in the security parameter [Cre97].

The output generated in item 5 of the simulator will differ from $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ only if a dishonest committer in the hybrid interaction succeeds to open a bit other than the bit he committed to in the commitment phase, but this probability is exponentially small in the security parameter [Cre97]. The codeword used in item 4 of the simulator to compute the bit b that $\mathcal{S}$ send to $\mathcal{F}_{COM}$ is the only codeword (if it exists) such that $d_H(\hat{c}, c) < \gamma n/2$ and so the only that will be accepted with non-negligible probability in the test of the decommitment phase. If $\mathcal{S}$ doesn't find such codeword, it can send a random bit to $\mathcal{F}_{COM}$ because any codeword used in the decommitment phase will be accepted only with negligible probability.

A dishonest receiver that knows $BS_\epsilon(c)$ and m in the hybrid interaction has, before the decommitment, negligible information about b [Cre97]. A bad code generated by a corrupted receiver can easily be detected by the sender just observing the rank of the matrix, because k is the parameter that assures the sender's security.

As all events that can result in different views have negligible probabilities, $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ and $\mathtt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ are statistically indistinguishable. This completes the security proof of the protocol, and so the lemma and the theorem are valid.

4.3. Efficient Simulator

We deal with computationally unbounded adversaries, so efficient simulators are not a big issue. However, we note here that there are ways to turn our simulator into efficient ones by changing slightly the protocol. In the simulator above there are two inefficient procedures in the execution: the searching in items 3 and 6 of the simulator for c that an honest receiver will accept in the test performed on step 2 of Decommitment phase and the searching in item 4 of the simulator for $\hat{c}$ such that $d_H(\hat{c}, c) < \gamma n/2$ (that is executed if the c sent by the corrupted sender is not a codeword). We can make the execution of simulator polynomial in the complexity of the adversary using a code generated according to theorem 2 and list decoding to find the codewords that are closer to any word. That works if $H(\epsilon) > 1 - \delta - 10^{-4}$. The proofs that the resulting protocol is stand-alone secure are the same as in [Cre97], since the code specified by theorem 2 meets the Gilbert-Varshamov bound. Details are left to an extended version of this work.

5. Conclusion

We have proved that is possible to statistically UC-realizes $\mathcal{F}_{COM}$ with an ($\mathcal{F}_{AUTH}$,$\mathcal{F}_{BSC,\epsilon}$)-hybrid protocol, and so the noisy channel is a valid set-up assumption to UC-realize the ideal commitment functionality.

References

[AC93] R. Ahlswede, I. Csiszár, Common Randomness in Information Theory and Cryptography – Part I: Secret Sharing, IEEE Trans. Inf. Theory, vol. 39, no. 4, pp. 1121–1132, 1993.

[BCNP04] B. Barak, R. Canetti, J. B. Nielsen, R. Pass, Universally Composable Protocols with Relaxed Set-Up Assumptions, 36th FOCS, pp.186-195, 2004.

[BBCM95] C. H. Bernett, G. Brassard, C. Crépeau and U. M. Maurer, Generalized Privacy Amplification, IEEE Transaction on Information Theory, Volume 41, Number 6, November 1995, pp. 1915-1923.

[BBCS92] C. H. Bernett, G. Brassard, C. Crépeau and M. H. Skubiszewska, Practical Quantum Oblivious Transfer, In Advances in Cryptology: Proceedings of Crypto '91, Vol. 576, page 351-366, 1992.

[BBR88] C. H. Bennett , G. Brassard , J. Robert, Privacy Amplification by Public Discussion, SIAM Journal on Computing, v.17, n.2, p.210-229, April 1988.

[BCC88] G. Brassard, D. Chaum and C. Crépeau, Minimum Disclosure Proofs of Knowledge, JCSS, Vol. 37, No. 2, pages 156-189, 1988.

[BCJL93] G. Brassard, C. Crépeau, R. Jotza and D. Langlois, A Quantum Bit Commitment Scheme Provably Unbreakable by Both Parties. Proceeding of the 34th IEEE Symposium on Foundations of Computer Science, 1993, pp. 362-371.

[Can05] R. Canetti, Universally Composable Security: A New Paradigm for Cryptographic Protocols. 2005, Available at http://eprint.iacr.org/2000/067, Extended Abstract appeared in proceedings of the 42nd Symposium on Foundations of Computer Science (FOCS), 2001.

[CF01] R. Canetti and M. Fischlin, Universally Composable Commitments. In Advances in Cryptology - Crypto 2001, Volume 2139, pages 19-40, 2001.

[CLOS02] R. Canetti, Y. Lindell, R. Ostrovsky, A. Sahai, Universally Composable Two Party and Multi-party Secure Computation, 34th STOC, pp. 494-503, 2002.

[CW79] J. L. Carter and M. N. Wegman, Universal Classes of Hash Functions, Journal of Computer and System Sciences, Vol. 18, pp. 143-154, 1979.

[CK88] C. Crépeau, J. Kilian, "Achieving Oblivious Transfer Using Weakened Security Assumptions", Proc. 29^{th} FOCS, pp. 42–52, 1988, IEEE.

[Cre97] C. Crépeau. Efficient Cryptographic Protocols Based on Noisy Channels. Advances in Cryptology: Proceedings of Eurocrypt '97 , pages 306-317, 1997.

[Che52] H. Chernoff, A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on The Sum of Observations, Ann. Math. Statistics, vol. 23, pp. 493–507, 1952.

[CT91] T. Cover, J. Thomas, Elements of Information Theory. Wiley. 1991.

[DFMS04] I. Damgård, S. Fehr, K. Morozov, L. Salvail: Unfair Noisy Channels and Oblivious Transfer. TCC 2004: 355-373

[DG03] I. Damgård, J. Groth, Non Interactive and Reusable Non-malleable Commitment Schemes, 34th STOC, pp. 426-437, 2003.

[DKS99] I. Damgård, J. Kilian, L. Salvail, "On the (Im)possibility of Basing Oblivious Transfer and Bit Commitment on Weakened Security Assumptions", Advances in Cryptology: EUROCRYPT 1999, pp. 56–73, Springer 1999.

[DN02] I. Damgård and J. B. Nielsen, Perfect Hiding and Perfect Binding Universally Composable Commitment Schemes with Constant Expansion Factor, CRYPTO 2002, pp.581-596, 2002.

[DPW05] Y. Dodis, R. Pass and S. Walfish. Fully Simulatable Multiparty Computation. Manuscript, 2005.

[EGL85] S. Even, O. Goldreich, A. Lempel. A Randomized Protocol for Signing Contracts. Communications of the ACM 28(6), 637-647, 1985.

[Gol01] O. Goldreich, Foundations of Cryptography : Volume 1 - Basic Tools, Cambridge University Press, 2001.

[GMW87] O. Goldreich, S. Micali and A. Wigderson, How to Play Any Mental Game or a Completeness Theorem for Protocols with Honest Majority, STOC '87.

[GMW91] O. Goldreich, S. Micali and A. Wigderson, Proofs that Yield Nothing but their Validity or All Languages in NP have Zero-Knowledge Proof System, J. ACM 38(3): 691-729. 1991.

[GMR89] S. Goldwasser, S. Micali and C. Rackoff, The Knowledge Complexity of Interactive Proof Systems, SIAM Journal on Comput., Vol. 18, No. 1, 1989, pp. 186-208.

[GI03] V. Guruswami, P. Indyk, Efficiently Decodable Low-Rate Codes Meeting Gilbert Varshamov Bound, invited to the 41st Annual Allerton Conference on Communication, Control and Computing, 2003.

[HM04] D. Hofheinz and J. Müller-Quade, Universally Composable Commitments Using Random Oracles, Theory of Cryptography Conference (TCC), LNCS 2951 pp, 58-74, 2004.

[Hol06] T. Holenstein, Strengthening Key Agreement Using Hard-core Sets, PhD thesis, Swiss Federal Institute of Technology (ETH) Zurich.

[HR06] T. Holenstein and R. Renner, On the Randomness of Independent Experiments, http://arxiv.org/abs/cs.IT/0608007.

[INW03] H. Imai, A. C. A. Nascimento, A. Winter, Commitment Capacity of Discrete Memoryless Channels, IMA Int. Conf. 2003: 35-51

[ILL89] R. Impagliazzo , L. A. Levin , M. Luby, Pseudo-random Generation From One-way Functions, Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing, p.12-24, May 14-17, 1989.

[MS77] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes, North-Holland, 1977.

[Mau93] U. Maurer, Protocols for Secret Key Agreement by Public Discussion Based on Common Information, CRYPTO 1992, pp. 461–470, Springer 1993., Secret Key Agreement by Public Discussion,IEEE Trans. Inf. Theory, vol. 39, no. 3, pp. 733–742, 1993.

[PS04] M. Prabhakaran and A. Sahai, New Notions of Security: Achieving Universal Composability without Trusted Setup, In Proc. of STOC, 2004.

[Wyn75] A. D. Wyner, The Wire Tap Channel, Bell System Tech. Journal, vol. 54, pp. 1355–1387, 1975.