ADAPTIVE AND COMPOSABLE NON-INTERACTIVE STRING-COMMITMENT PROTOCOLS

Huafei Zhu[1], Tadashi Araragi[2], Takashi Nishide[3] and Kouichi Sakurai[3]
[1]*Institute for Infocomm Research, A-STAR, Singapore*
[2]*NTT Communication Science Laboratories, Kyoto, Japan*
[3]*Department of Computer Science and Communication Engineering, Kyushu University, Fukuoka, Japan*

Keywords: Non-interactive, String-commitment protocol, Universally composable security.

Abstract: Designing non-committing encryptions tolerating adaptive adversaries is a challenging task. In this paper, a simple implementation of non-committing encryptions is presented and analyzed in the strongest security model. We show that the proposed non-committing encryption scheme is provably secure against adaptive adversaries in the universally composable framework assuming that the decisional Diffie-Hellman problem is hard.

1 INTRODUCTION

Informally, a commitment scheme is a two-party protocol that has two phases: a committing phase, where a receiver of the commitment obtains some information which amounts to a commitment to an unknown value (sealed by a committer), and a reveal phase, where the receiver obtains an opening of the commitment to some value (revealed by the committer). Commitment is an essential building block in many cryptographic protocols, such as zero-knowledge protocols (e.g., (Brassard et al., 1988; Goldreich et al., 1987; Damgård, 1989)), general functional evaluation protocols (e.g., (Goldreich et al., 1987; Galil et al., 1987)), contract-signing and electronic commerce, and more (see (Goldreich, 2001; Goldreich, 2004) for further reference) and has been studied extensively in the past two decades (e.g., (Blum, 1981; Naor, 1991; Canetti and Fischlin, 2001; Naor et al., 9912; Barak et al., 2004; Canetti et al., 2007)).

Universally composable (UC) commitments guarantee that a commitment protocol behaves like an ideal commitment service, even when concurrently composed with an arbitrary set of protocols. To prove security of a commitment scheme realizes the UC-security in the presence of an adaptive adversary, one must construct an ideal-world adversary such that the adversary's view of a real-life execution of a commitment protocol can be simulated given just the data the adversary is entitled to. That is, to prove the UC-security, a commitment scheme running between a committer P_i and a receiver P_j in an environment $\mathcal{Z}$ must be equivocable and extractable. To simulate the case where the honest committer P_i sends a commitment c to the receiver P_j in the real-world, an ideal-world adversary $\mathcal{S}$ must interpret this fake commitment c as a genuine commitment of a message m (the value m is revealed by the ideal commitment functionality during the reveal phase). As such, the commitment scheme must be equivocable. If the real-world adversary $\mathcal{A}$ sends a commitment c to P_j on behalf of the corrupted committer P_i, the ideal-world adversary $\mathcal{S}$ must extract the implicit message m which is the explicit input to the commitment functionality. As such, the commitment scheme must be extractable. It follows that a commitment scheme that realizes UC-security in the presence of adaptive adversaries must be equivocable and extractable.

The universally composable security (UC-security) is so strong a notion that a commitment scheme cannot be implemented in the plain model (Canetti and Fischlin, 2001). Thus, all known commitment schemes are worked in the so called common reference string model. A commitment scheme is called common-reference-string reusable (reusable, in short) if a common reference string is reused for multiple commitments.

Zhu H., Araragi T., Nishide T. and Sakurai K. (2010).
ADAPTIVE AND COMPOSABLE NON-INTERACTIVE STRING-COMMITMENT PROTOCOLS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 354-361
DOI: 10.5220/0002980503540361

1.1 The State-of-the-art

The state-of-the-art non-interactive commitment schemes in the universally composable framework are mainly constructed from the following two categories: non-interactive, universally compsosably secure bit-commitment schemes and interactive universally composable string-commitment schemes.

1.1.1 Universally Composably Non-interactive Bit-commitment Schemes

Canetti and Fischlin (Canetti and Fischlin, 2001) have proposed two basic approaches for constructions of non-interactive and universally composable bit-commitment schemes in the common reference string model. The first construction of commitment protocol is based on any trapdoor permutation in the one-time common reference string model. The second construction is based on the existence of claw-free pairs of trapdoor permutations in the reusable common reference string model, where the honest players are assumed that they faithfully erase some parts of their internal randomness (i.e., their commitment scheme works in the internal randomness erasure model). Canetti and Fischlin then proposed an improved bit-commitment scheme based on the Diffie-Hellman assumption in the (randomness) non-erasure model.

Canetti, Lindell, Ostrovsky and Sahai (Canetti et al., 2002) have presented a new universally composable non-interactive bit-commitment protocol that is secure against adaptive adversary based on the existence of enhanced trapdoor permutations in the common reference string model. Their scheme realizes the UC-security in the the multi-session ideal commitment functionality, an extension of the single-session ideal commitment functionality presented in (Canetti and Fischlin, 2001). The Canetti and Fischlin commitment schemes (Canetti and Fischlin, 2001) and the Canetti, Lindell, Ostrovsky and Sahai commitment schemes (Canetti et al., 2002) use $\Omega(\lambda)$ bits to commit a bit, where λ is a security parameter. These pioneer works are important from point view of the theoretical research.

1.1.2 Universally Composably Interactive String-commitment Schemes

Damgård and Nielsen (Damgård and Nielsen, 2002) have presented practical interactive string-commitment protocols in the common reference string model. The Damgård and Nielsen interactive string-commitment protocol realizes the UC-security in the presence of adaptive adversaries but the size of the common reference string grows linearly with the number of participants. Damgård and Groth (Damgård and Groth, 2003) then proposed an improved commitment scheme with constant common reference string size which is independent with the number of the parties in the commitment protocol.

Camenisch and Shoup (Camenisch and Shoup, 2003) have constructed alternative interactive universally composably secure string-commitment protocols in the context of verifiably committed encryptions. Their construction is based on the zero-knowledge proof of an encryption indeed decrypts to a valid opening of a commitment. This construction realizes universally composable security assuming the Diffie-Hellman assumption is hard in the common reference model.

1.1.3 Universally Composably Non-interactive String-commitment Schemes

Very recently, Nishimaki, Fujisaki and Tanaka (Nishimaki et al., 2009) have proposed an interesting universally composable non-interactive string-commitment scheme based on all-but-one trapdoor functions introduced by Peikert and Waters in STOC 2008 (Peikert and Waters, 2008). The Nishimaki-Fujisaki-Tanaka's non-interactive string commitment is one time (a common reference string is refreshed whenever a new session starts). The idea of their implementation is sketched below.

Let Σ =(SKGen, Sign, Veri) be a signature scheme that is secure against adaptive chosen-message attack in the sense of Goldwasser, Micali and Rivest (Goldwasser et al., 1988). Let Δ = (EGen, Enc, Dec) be Damgård-Jurik's length-flexible public-key encryption scheme (Damgård and Jurik, 2001). To commit a message $m \in \mathcal{M}$, a common-reference-string generation algorithm (CRS) invokes the key generation algorithm SKGen of the underlying signature scheme to produce a pair of verification key and signing key (vk^*, sk^*). CRS then invokes the encryption algorithm Enc to produce a ciphertxt $\mathrm{Enc}(vk^*)$ of the public verification key. The common reference string σ is $\mathrm{Enc}(vk^*)$ together with a description of a pair-wise independent hash function $\mathcal{H}$. Given σ and m, a committer S invokes SKGen to generate a new pair of verification and signing key (vk, sk), and then generates a randomized ciphertext C of the message $(vk^* - vk)m$. That is, the committer S invokes the encryption algorithm Enc which takes $(vk^* - vk)m$ as input to produce a ciphertext C (=$\mathrm{Enc}((vk^* - vk)m, r_m)$) with randomness r_m. To simulate the view of the honest committer S, the lossy branch vk^* will be set to vk. As such, *the common-reference-string in the Nishimaki-Fujisaki-Tanaka's commitment scheme is one-time.*

1.2 This Work

This paper studies non-interactive (no interactive communication between a committer and a receiver), reusable (common-reference-string reused for multi-commitments) string-commitment schemes in the universally composable framework in the presence of adaptive adversaries. To the best of our knowledge, no construction of universally composable, non-interactive string-commitment in the presence of the adaptive adversary is known. This leaves an interesting research problem: how to construct adaptive (*here "adaptive" means that any adversary in our model is adaptive*) and composable (*here "composable" means that the protocol is universally composable in the Canetti's framework*) string-commitment protocols (*here "string-commitment" means that the length of a committed message is* $\{0,1\}^l$, $l > 1$) in the common reference string reusable model (*here "reusable" means that the common reference string can be used for multi-session and hence it is not a one-time common reference string model*) without erasure (*here "non-erasure" means that a party is not assumed to erase its internal state during the protocol execution*)?

1.2.1 The Technique

Our non-interactive string-commitment protocol is based on Paillier's homomorphic encryption scheme. Recall that the difficulty to realize the uc-security of a commitment protocol is to provide an efficient method to reach the equivocability and extractability once a common reference string is given.

1. To realize the extractability, we allow a simulator to run a key generation algorithm of the Paillier's homomorphic encryption scheme. We allow the simulator to randomly select two ciphertexts K_1 and K_2. The common reference string is defined by (K_1, K_2). Since the simulator knows the trapdoor of the underlying public-key encryption scheme, it follows that the simulator is able to extract the all encrypted messages (including the randomness used to generate the common reference string and extractable keys sketched below).
2. To realize the equivocability, we will construct a random key K (=$K_1{}^{r_1}K_2{}^{r_2}$) from the common reference string (K_1, K_2). The random key K is a base to commit a message m in the form $K^m r_m^N \bmod N^2$. The committer P_i then invokes 3-move Σ-protocol and proves the knowledge of (r_1, r_2) to a receiver P_j. Let **PoK** be a transcript of the zero-knowledge derived from the Σ-protocol. The commitment of a string m is denoted by $(K, C, \mathbf{PoK})$, where $C = K^m r_m^N \bmod N^2$.

   Let $\psi(k,r) = (1+N)^k r^N \bmod N^2$ be an equivocable key (intuitively, a key is equivocable if it is of from $\psi(0,r)$, i.e., $k = 0$, the randomness r of the equivocable key K is called trapdoor string; a key K is called extractable if it is of form $\psi(k,r)$ ($k \neq 0$)). Let xKey be a set of all extractable keys and eKey be a set of all equivocable keys. The key point to reach the equivocability is that we allow a simulator to select the randomness (r_1, r_2) so that K can be either an extractable key or an equivocable key. In case that K is an extractable key, the simulator is able to extract the implicit input message of a corrupted party. In case that K is an equivocable key, the simulator is able to modify the internal state when an honest party gets corrupted.

We stress that the standard rewinding technique for extracting the knowledge of a zero-knowledge proof is not allowed in the universally composable framework of Canetti (Canetti, 2001). This means that we cannot get the implicit input message m by rewinding a knowledge prover. Fortunately, in our construction, a simulator knows the secret key of the Paillier's encryption scheme and the randomness (r_1, r_2) used to generate K (the base to commit a message m) that are sufficient for the simulator to extract the message m.

We also stress that a straight-forward application of a 3-move interactive Σ-protocol results in an interactive string-commitment protocol. A well-known technique for making interactive Σ-protocols non-interactive is the Fiat-Shamir heuristic, where a random challenge string e is computed by the prover as a hash of the statement proved and the first message K. Unfortunately, if the Fiat-Shamir heuristic is applied then the resulting string-commitment protocol works in the random oracle only. To avoid using of the random oracle model, we will apply the Damgård, Fazio and Nicolosi's method (Damgård et al., 2006) for compiling a class of Σ-protocols into non-interactive zero-knowledge arguments $\widetilde{\Sigma}$, where a verifier is assumed to hold a pair of registered public/secret keys. As a result, our non-interactive string-commitment scheme works in the registered public key model (we refer to the reader (Damgård et al., 2006) for more details).

1.2.2 The Result

We claim that the adaptive and composable non-interactive string-commitment scheme presented and analyzed in this paper reaches the UC-security in the presence of adaptive adversaries in the

common reference string model assuming that the underlying Paillier's public-key encryption scheme is semantically secure, and the underlying Damgård-Fazio-Nicolosi's non-interactive protocol is zero-knowledge in the registered public-key model. If the underlying Paillier's public-key encryption scheme is replaced by Damgård-Jurik's length-flexible public key encryption scheme (Damgård and Jurik, 2001), then the non-interactive string-commitment is length-flexible as well.

Since the proposed non-interactive string-commitment scheme is reusable and length-flexible and universally composable against, it follows that our result extends the recent work of Zhu (Zhu, 2009) *which is provably secure against non-adaptive in the universally composable framework.* As a result, we provide a solution to the open problem posed in (Zhu, 2009).

Road Map. The rest of the paper is organized as follows. In Section 2, security definition of commit schemes is sketched; Our adaptive and composable non-interactive string-commitment scheme is presented and analyzed in Section 3. We conclude our work in Section 4.

2 PRELIMINARIES

2.1 The Universally Composable Framework

We work in the standard universally composable framework of Canetti (Canetti, 2001), where all participants are modeled as probabilistic polynomial time (PPT) Turing machines. Security of protocols is defined by comparing the protocol execution to an ideal process for carrying out the desired task. Namely, the process of executing a protocol in the presence of an adversary and in a given computational environment is first formalized. Next an ideal processing for carrying out the task at hand is formalized. In the ideal processing the parties do not communicate with each other; instead they have access to an ideal functionality which is essentially an incorruptible trust party that is programmed to capture the desired requirements from the task at hand. A protocol is said to securely realize a task if the processing of running the protocol emulates the ideal process of that task. We assume the reader is familiar with the standard notion of UC security. The detailed descriptions of the executions, and definitions of $\text{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ and $\text{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$ are omitted and refer to the reader (Canetti, 2001) for more details.

2.2 The Common Reference String Model

The functionality of common reference string model assumes that all participants have access to a common string that is drawn from some specified distribution $\mathcal{D}$. The common reference string is chosen ahead of the time and is made available before any interaction starts. The common reference string functionality defined below is due to Canetti and Fischlin (Canetti and Fischlin, 2001).

Functionality $\mathcal{F}_{\mathsf{crs}}^{\mathcal{D}}$

$\mathcal{F}_{\mathsf{crs}}^{\mathcal{D}}$ proceeds as follows, when parameterized by a distribution $\mathcal{D}$.

- when receiving a message (sid, P_i, P_j) from P_i, let $\mathsf{crs} \leftarrow \mathcal{D}(1^n)$ and send (sid, crs) to P_i, and send (crs, sid, P_i, P_j) to the adversary, where sid is a session identity. Next when receiving (sid, P_i, P_j) from P_j (and only from P_j), send (sid, crs) to p_j and to the adversary, and halt.

2.3 The Commitment Functionality

To capture the notion of reusability, one must define the functionality of multi commitment, decommitment processes. The commitment functionality defined below is due to Canetti, Lindell, Ostrovsky and Sahai (Canetti et al., 2002).

Functionality $\mathcal{F}_{\mathsf{mcom}}$

$\mathcal{F}_{\mathsf{mcom}}$ proceeds as follows, running with parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$

- **Commit Phase.** Upon receiving a value (**commit**, sid, cid, P_i, P_j, $m \in \mathcal{M}$), record the tuple (sid, cid, P_i, P_j, m) and send the message (**receipt**, sid, cid, P_i, P_j) to P_j and $\mathcal{S}$. Ignore any future **commit** messages with the same cid from P_i to P_j.
- **Open Phase.** Upon receiving a value (**open**, sid, cid) from P_i: If a tuple (sid, cid, P_i, P_j, m) was previously recorded, then send the message (**open**, sid, cid, P_i, P_j, m) to P_j and $\mathcal{S}$ and halt; otherwise ignore.

Definition. Let $\mathcal{F}_{\text{mcom}}$ be a multi commitment functionality. A protocol π is said to universally composable realize $\mathcal{F}_{\text{mcom}}$ if for any adversary $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that for all environments $\mathcal{Z}$, the ensemble $\text{IDEAL}_{\mathcal{F}_{\text{mcom}},\mathcal{S},\mathcal{Z}}$ is computationally indistinguishable with the ensemble $\text{REAL}_{\pi,\mathcal{A},\mathcal{Z}}$.

3 NON-INTERACTIVE STRING-COMMITMENT SCHEMES

We will make use of Paillier's probabilistic public key system (Paillier, 1999) to construct non-interactive, universally composable and reusable commitment schemes in this paper.

3.1 xKeys and eKeys

Borrowing the notations and notions from Damgård and Jurik (Damgård and Jurik, 2001), we define extractable keys (xKeys) and equivocable keys (eKeys) in the context of the Paillier's encryption scheme. Let $\psi(k,r)= (1+N)^k r^N \bmod N^2$. A key is called equivocable if it is of from $\psi(0,r)$. The randomness r of the equivocable key K is called trapdoor string. A key K is called extractable if it is of form $\psi(k,r)$ ($k \neq 0$). Let xKey be a set of all extractable keys and eKey be a set of all equivocable keys. If the decisional composite residuosity assumption (DCRA) introduced in (Paillier, 1999) holds, then elements of form $\psi(0,r)$ cannot be distinguished from the element of the form $\psi(k,r)$, where r is uniformly from Z_N^* and k is any fixed element in Z_N.

3.2 The Damgård-Fazio-Nicolosi's Non-interactive Zero-knowledge Protocol

A Σ-protocol for a relation R is an interactive proof system Σ for $L_R := \{x \mid \exists w : (x,w) \in R\}$ with the conversation of form (a,e,z), where (a,z) is computed by a prover and e is selected by a verifier.

Damgård, Fazio and Nicolosi (Damgård et al., 2006) provide a method for compiling a class of 3-move Σ-protocols into non-interactive zero-knowledge arguments $\widetilde{\Sigma}$. Their method is based on homomorphic encryptions (say, Paillier's encryption scheme) and does not use random oracles. The Damgård-Fazio-Nicolosi's non-interactive zero-knowledge protocol requires that a private/public key pair is set up for the verifier (i.e., it works in the registered public-key model). Below, we sketch the Damgård-Fazio-Nicolosi's compiler:

1. Given an instance (x,w) to prove, a prover P gets a verifier's registered public key pk_V derived from the Paillier's encryption scheme, together with a ciphertext c broadcast by the verifier, where c is an encryption of a random string e (the randomness e is selected and encrypted by the verifier under the public-key pk_V, i.e., $c = \mathcal{E}_{pk_V}(e, r_e)$);
2. the prover P generates the first message a using the randomness r and then computes a randomized ciphertext $Z = \mathcal{E}_{pk_V}(r)c^w$. Finally, the prover sends $(x,(a,Z))$ to the verifier.
3. Upon receiving $(x,(a,Z))$, the verifier decrypts Z to get z ($z = r + ew$ by the correctness) and checks that whether $(x,(a,e,z))$ is valid transcript. If the transcript $(x,(a,e,z))$ is valid then accepts; otherwise, rejects the received transcript.

(due to (Damgård et al., 2006)) Damgård, Fazio and Nicolosi have shown that the non-interactive zero-knowledge protocol $\widetilde{\Sigma}$ is complete and sound in the registered public-key model.

3.3 The Description

The non-interactive string commitment protocol π presented in this section is based on the Paillier's encryption scheme. We stress that if the underlying Paillier's public-key encryption scheme is replaced by Damgård-Jurik's length-flexible public key encryption scheme (Damgård and Jurik, 2001), then the described non-interactive string-commitment is length-flexible. Below, we describe our string-commitment protocol in the context of the Pailler's encryption scheme (the description of string-commitment protocol based on the Damgård-Jurik's is straight-forward and thus omitted).

1. **Common-reference-string Generation Phase.** On input a security parameter 1^k, $((p,q),N) \leftarrow$ **Gen**(1^k). Let $K_1 \leftarrow (1+N)^{k_1} {r_{k_1}}^N \bmod N^2$ and $K_2 \leftarrow (1+N)^{k_2} {r_{k_2}}^N \bmod N^2$, where $k_1 \neq 0$ and $k_2 \neq 0$, i.e., both K_1 and K_2 are xKeys. The common reference string $\sigma = (N, K_1, K_2)$. The trapdoor string τ is (p,q).

2. **The Committing Phase.** On input a message $m \in Z_N$, the committer P_i performs the following computations
   - P_i randomly selects $r_1, r_2 \in Z_N$ and computes $K = K_1^{r_1} K_2^{r_2} \bmod N^2$;
   - P_i then invokes the Damgård-Fazio-Nicolosi's non-interactive zero-knowledge argument $\widetilde{\Sigma}$ and proves the knowledge $r_1 \in Z_N$ and $r_2 \in Z_n$ such that $K = K_1^{r_1} K_2^{r_2}$ to P_j. Let **PoK** be a transcript of zero-knowledge argument derived from the Damgård-Fazio-Nicolosi's protocol $\widetilde{\Sigma}$;
   - P_i then computes $K^m r_m^N \bmod N^2$. Let $C = K^m r_m^N \bmod N^2$.
   - Finally P_i sends $(K,$ **PoK**$, C)$ to the receiver P_j.

3. **The Opening Phase.** Upon receiving $(K$, **PoK**, $C)$ and (m, r_m), the receiver P_j first checks the validity of the received transcript **PoK**. If it is invalid, then outputs $\perp$; otherwise, P_j checks that $C \stackrel{?}{=} K^m r_m^N \bmod N^2$. If the check is invalid, P_j outputs $\perp$, otherwise, it outputs "accept".

This ends the description of the non-interactive string-commitment scheme

3.4 The Proof of Security

Theorem. The non-interactive string-commitment protocol π reaches the UC-security in the presence of adaptive adversaries in the reusable common-reference-string model assuming that the underlying Paillier's public-key encryption scheme is semantically secure, and the underlying Damgård-Fazio-Nicolosi's non-interactive protocol is zero-knowledge in the registered public-key model.

Proof. We describe the ideal model adversary $\mathcal{S}$ which comprises the following 6 simulation steps (S. 1 - S. 6):

- S. 1): At the outset of the simulator $\mathcal{S}$ prepares a common reference string σ by invoking the key generation algorithm $\mathcal{K}$ of the underlying Paillier's encryption scheme and outputs (pk', sk'). Given pk', $\mathcal{S}$ randomly selects $K_1' \in C$ and $K_2' \in C$. Let $\sigma' = (pk', K_1', K_2')$. The trapdoor string τ' is sk'. The simulator keeps τ' secret and broadcasts σ' to all participants.
- S. 2): If at the some point in the execution the environment $\mathcal{Z}$ writes a message (**commit**, *sid*, *cid*, P_i, P_j, m) on the tape of the uncorrupted party P_i, then the ideal world simulator $\mathcal{S}$ who cannot read the actual message m, generates a simulated view of the real world committer P_i via the following computations:
  - On input σ', $\mathcal{S}$ extracts $(k_1', r_{k_1'})$ and $(k_2', r_{k_2'})$ from the common reference string σ' with the help of the auxiliary string sk'; Note that K_1' and K_2' are chosen uniformly at random. As a result, K_1' and K_2' are xKeys with overwhelming probability.
  - $\mathcal{S}$ randomly chooses $r'_1 \in Z_N$ and computes $r'_2 \in Z_N$ from the equation $k'_1 r'_1 + k'_2 r'_2 = 0 \bmod N$; Let $K' = K_1'^{r'_1} K_2'^{r'_2} \bmod N^2$.
  - $\mathcal{S}$ then invokes the Damgård-Fazio-Nicolosi's non-interactive zero-knowledge protocol $\widetilde{\Sigma}$ and proves to P_j the knowledge (r_1', r_2') such that $K' = K_1'^{r_1'} K_2'^{r_2'} \bmod N^2$. Let **PoK**$'$ be a transcript of generated by Damgård-Fazio-Nicolosi's non-interactive zero-knowledge protocol $\widetilde{\Sigma}$ for proving the knowledge (r_1', r_2') such that $K' = K_1'^{r_1'} K_2'^{r_2'} \bmod N^2$;
  - $\mathcal{S}$ randomly selects m' and $r_{m'}$ and sets $C' = K'^{m'} \mathcal{E}(0, r_{m'})$.

  The simulator $\mathcal{S}$ then tells the real world adversary $\mathcal{A}$ that P_i has sent $(K'$, **PoK**$'$, $C')$ to P_j.
- S. 3): If at the some point in the execution $\mathcal{Z}$ instructs an corrupted party P_i to open the commitment (**open**, *sid*, *cid*, P_i, P_j, m), $\mathcal{S}$ learns m^* via the functionality $\mathcal{F}_{\text{mcom}}$ and then modifies the internal state of $(K'$, **PoK**$'$, $C')$ such that $(K'$, **PoK**$'$, $C')$ looks like a genuine commitment of the string $m^* \in Z_N$ from the point view of the environment $\mathcal{Z}$. That is,
  - (equivocation) Since $K' = K_1'^{r_1'} K_2'^{r_2'} \bmod N^2$ is an eKey (recall that the simulator randomly selects $r_1' \in Z_N$ and then computes r_2' from the equation $k_1' r_1' + k_2' r_2' = 0 \bmod N$), the simulator $\mathcal{S}$ must provide (m^*, r_{m^*}) such that $C = K'^{m'} \mathcal{E}_{pk'}(0, r_{m'}) = K'^{m^*} \mathcal{E}_{pk'}(0, r_{m^*})$. This is an easy task since $\mathcal{S}$ knows the trapdoor string sk'.
- S. 4): If the simulated adversary $\mathcal{A}$ lets the corrupted party P_i send (**commit**, *sid*, *cid*, P_i, P_j, $(K'$, **PoK**$'$, $C'))$ to an honest party P_j. Given K' and **PoK**$'$, the simulator $\mathcal{S}$ checks the validity of **PoK**$'$. If the check is valid, $\mathcal{S}$ performs the following computations
  - (extraction) $\mathcal{S}$ first extracts k' from K' with the help of the secret key sk'; $\mathcal{S}$ then extracts $k' \times m' \bmod N$ from C' with the help of the secret key sk'. Finally, $\mathcal{S}$ sends the extracted message m' to the functionality $\mathcal{F}_{\text{mcom}}$.
- S. 5): If $\mathcal{A}$ tells the corrupted party P_i to open a valid commitment C' correctly with message m^*, then $\mathcal{S}$ compares m^* with the previously extracted message m' and stops if they differ; otherwise, $\mathcal{S}$ sends (**open**, *sid*, *cid*, P_i, P_j) in name of the party to the functionality $\mathcal{F}_{\text{mcom}}$. If P_i is supposed to decommit incorrectly, then $\mathcal{S}$ also sends an incorrect opening to the functionality.
- S. 6): Whenever the simulated $\mathcal{A}$ demands to corrupt a party, $\mathcal{S}$ corrupts this party in the ideal model and learns all internal information of the party. $\mathcal{S}$ first adapts possible decommitment information about the previously given but not yet unopened commitment of this party, like in the case if an honest party decommitting. After this, $\mathcal{S}$ gives all this adjusted information to $\mathcal{A}$.

This ends the description of the simulator.

We first show that the distribution of public-key pk generated by the protocol π is identical to the public-key pk' generated by the simulator. The random variables (K_1, K_2) in π are xKeys. The random variables (K_1', K_2') generated by the simulator are random ciphertetxts. It follows that the distribution of the common reference string $\sigma = (pk, K_1, K_2)$ generated in generated in the protocol π is computationally indistinguishable from the distribution of the common reference string $\sigma' = (pk', K_1', K_2')$ generated by the simulator.

We then show that the distribution of the view in the protocol π is computationally indistinguishable from that of the simulation assuming that the Paillier's encryption scheme is semantically secure and the Damgård-Fazio-Nicolosi's non-interactive protocol is zero-knowledge. Let $(K,$ **PoK**, $C)$ be random variables generated in π and $(K',$ **PoK**$',$ $C')$ be random variables generated by the simulator. Note that K is an xKey in π (with overwhelming probability) while K' is an eKey in the simulation (with overwhelming probability). Also notice that C is an xKey in π (with overwhelming probability) while C' is an eKey in the simulation (with overwhelming probability). Since the Paillier's encryption scheme is semantically secure, it follows that the random variables (K, C) and (K', C') are computationally indistinguishable.

Since the Damgård-Fazio-Nicolosi's non-interactive protocol is zero-knowledge, it follows that the distribution of the random variable **PoK** and **PoK**$'$ are identical. As a result, the random variables $(K,$ **PoK**, $C)$ and $(K',$ **PoK**$',$ $C')$ are computationally indistinguishable assuming that the Paillier's encryption scheme is semantically secure and the Damgård-Fazio-Nicolosi's non-interactive protocol is zero-knowledge.

Finally, we know that r_m' is computed from the equation $r_k'^{m'} r_{m'} = r_k^m r_m \bmod N$. The distribution of the random variable $(m, r_m))$ in π is identical to the distribution of the random variables generated by the simulator. As such, the distribution of the view $((K,$ **PoK**, $C)$ and $(m, r_m)))$ generated by the protocol π is computationally indistinguishable to the view $((K',$ **PoK**$',$ $C')$ and $(m', r_{m'})))$ generated by the simulator. As a result, we know that $\text{IDEAL}_{\mathcal{F}_{\text{mcom}}, \mathcal{S}, \mathcal{Z}} = \text{REAL}_{\pi, \mathcal{A}, \mathcal{Z}}$.

4 CONCLUSIONS

In this paper an adaptive and composable non-interactive string-commitment protocol has presented and analyzed. We have shown that the proposed commitment protocol realizes the universally composable security in the presence of the adaptive adversaries in the reusable common reference string model assuming that the underlying Paillier's public-key encryption scheme is semantically secure, and the underlying Damgård-Fazio-Nicolosi's non-interactive protocol is zero-knowledge in the registered public-key model.

REFERENCES

Barak, B., Canetti, R., Nielsen, J., and Pass, R. (2004). Universally composable protocols with relaxed set-up assumptions. In *FOCS*. IEEE.

Blum, M. (1981). Coin flipping by telephone. In *CRYPTO*. Springer.

Brassard, G., Chaum, D., and Crépeau, C. (1988). Minimum disclosure proofs of knowledge. In *J. Comput. Syst. Sci*. Elsevier.

Camenisch, J. and Shoup, V. (2003). Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO*. Springer.

Canetti, R. (2001). Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*. IEEE.

Canetti, R., Dodis, Y., Pass, R., and Walfish, S. (2007). Universally composable security with global setup. In *TCC*. Springer.

Canetti, R. and Fischlin, M. (2001). Universally composable commitments. In *CRYPTO*. Springer.

Canetti, R., Lindell, Y., Ostrovsky, R., and Sahai, A. (2002). Minimum disclosure proofs of knowledge. In *STOC*. IEEE.

Damgård, I. (1989). On the existence of bit commitment schemes and zero-knowledge proofs. In *CRYPTO*. Springer.

Damgård, I., Fazio, N., and Nicolosi, A. (2006). Non-interactive zero-knowledge from homomorphic encryption. In *TCC*. Springer.

Damgård, I. and Groth, J. (2003). Non-interactive and reusable non-malleable commitment schemes. In *STOC*. IEEE.

Damgård, I. and Jurik, M. (2001). Non-interactive zero-knowledge from homomorphic encryption. In *PKC*. Springer.

Damgård, I. and Nielsen, J. (2002). Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*. Springer.

Galil, Z., Haber, S., and Yung, M. (1987). Cryptographic computation: Secure faut-tolerant protocols and the public-key model. In *CRYPTO*. Springer.

Goldreich, O. (2001). *Foundations of Cryptography, Volume 1*. Cambridge University Press, London, 1st edition.

Goldreich, O. (2004). *Foundations of Cryptography, Volume 2*. Cambridge University Press, London, 1st edition.

Goldreich, O., Micali, S., and Wigderson, A. (1987). How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*. IEEE.

Goldwasser, S., Micali, S., and Rivest, R. (1988). A digital signature scheme secure against adaptive chosen-message attacks. In *SIAM J. Comput.* ACM.

Naor, M. (1991). Bit commitment using pseudorandomness. In *J. Cryptology*. Springer.

Naor, M., Ostrovsky, R., Venkatesan, R., and Yung, M. (19912). Perfect zero-knowledge arguments for np can be based on general complexity assumptions. In *CRYPTO*. Springer.

Nishimaki, R., Tanaka, K., and Fujisaki, E. (2009). Efficient non-interactive universally composable string-commitment schemes. In *ProvSec*. Springer.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*. Springer.

Peikert, C. and Waters, B. (2008). Lossy trapdoor functions and their applications. In *STOC*. IEEE.

Zhu, H. (2009). New constructions for reusable, non-erasure and universally composable commitments. In *ISPEC*. Springer.