IEICE TRANS. FUNDAMENTALS, VOL.E90–A, NO.7 JULY 2007

PAPER

A Straight-Line Extractable Non-malleable Commitment Scheme

Seiko ARITA[†a)], ***Member***

SUMMARY Non-malleability is an important security property of commitment schemes. The property means security against the man-in-the-middle attack, and it is defined and proved in the simulation paradigm using the corresponding simulator. Many known non-malleable commitment schemes have the common drawback that their corresponding simulators do not work in a straight-line manner, requires rewinding of the adversary. Due to this fact, such schemes are proved non-malleable only in the stand-alone cases. In the multiple-instances setting, i.e., when the scheme is performed concurrently with many instances of itself, such schemes cannot be proved non-malleable. The paper shows an efficient commitment scheme proven to be non-malleable even in the multiple-instances setting, based on the KEA1 and DDH assumptions. Our scheme has a simulator that works in a straight-line manner by using the KEA1-extractor instead of the rewinding strategy.

key words: *commitment scheme, non-malleability, the KEA1 assumption, extractability*

1. Introduction

1.1 Commitment Schemes and Its Non-malleability

A commitment scheme, which is one of the most fundamental cryptographic protocols, is a two-party two-phase protocol:

1. Commit phase:
   A sender, through some interactions with a receiver if necessary, makes a commitment $c = com(m; r)$ to a message m with some randomness r and sends it to the receiver.
2. Open phase:
   The sender sends a decommit information m, r of c to the receiver. The receiver determines the validity of m, r by checking $c = com(m; r)$. If valid, it accepts m (else rejects).

(In the paper, we focus on a string commitment rather than a bit commitment.) Fundamental requirements for a commitment scheme are hiding and binding properties. It is *hiding* when commitments $c = com(m; r)$ are indistinguishable among different m's, and it is *binding* when it is infeasible to generate a commitment which can be correctly opened by distinct messages at once.

Dolev et al. [7] defines *non-malleability* (NM) of a commitment scheme, which means security against the man-in-the-middle (MIM) attack. Suppose an adversary $\mathcal{A}$ is in the MIM-setting, i.e., $\mathcal{A}$ is in the middle of honest left and right parties. A commitment scheme is called non-malleable when an adversary $\mathcal{A}$, given a commitment c to a message m from the left party, cannot generate a commitment c^* to another distinct message m^*, which is in some polynomially-computable relation R with m, for the right party. Here, $\mathcal{A}$ is supposed to get an open message m, r of c from the left party when it attempts to open c^* to m^* to the right party. (Strictly, this is non-malleability *with respect to opening*. When $\mathcal{A}$ is supposed only to commit, i.e., not given the decommitment m, r and not required to open c^*, it is called non-malleability *with respect to commitment*. In this paper, we focus on non-malleability with respect to opening.)

To prove non-malleability, we need a simulator Sim of $\mathcal{A}$'s behavior. Sim, alone without any help from the left party (especially without any knowledge of c, m, r), must generate a commitment c^* to m^* which has a relation R with m with the same probability as in the case of $\mathcal{A}$ in the MIM-setting.

1.2 A Generic Method for Non-malleable Commitment Schemes

Key properties for establishing non-malleability of commitment schemes are equivocality and extractability. A commitment scheme is called *equivocal* when one can, using some trapdoor information, generate a commitment that can be opened by any message later. A commitment scheme is called *extractable* when one can, also using some trapdoor information, extract the message under the commitment without any decommit information.

Generically, a non-malleable commitment scheme is constructed through enhancing some primitive commitment scheme to obtain equivocality and extractability at once. If such enhancement succeeds, its non-malleability can be proved as follows. Suppose an adversary $\mathcal{A}$ in the MIM-setting is given. The simulator needs to simulate the left view of $\mathcal{A}$ without the knowledge of the real message m. In order to do that, Sim as a left party commits to a dummy message m_0 for $\mathcal{A}$ through an equivocal commitment c. By equivocality, c can be opened by the real message m later instead of m_0, so the view of $\mathcal{A}$ is indistinguishable from its real view with a true commitment to m. Next, from the commitment c^* generated by the simulated $\mathcal{A}$, Sim (using

Manuscript received December 27, 2006.
Manuscript revised February 21, 2007.
Final manuscript received April 6, 2007.

[†]The author is with Institute of Information Security, Yokohama-shi, 221-0835 Japan.
a) E-mail: arita@iisec.ac.jp

DOI: 10.1093/ietfec/e90–a.7.1384

some trapdoor) extracts the committed message m^* by the extractability. Thus, Sim can find m^* without knowing the true m and its commitment c at all with the same probability that $\mathcal{A}$ commits to such m^* in the real MIM attack. This means non-malleability of the scheme.

1.3 Known Non-malleable Commitment Schemes

There are some known non-malleable commitment schemes. Crescenzo et al. [2] constructed a non-interactive and non-malleable commitment scheme. Unfortunately, the resulting commitments are large (i.e., $O(|m|k)$-bit with the message length $|m|$ and the security parameter k). Crescenzo et al. [3] and Fischlin and Fischlin [8], respectively, enhanced DL (discrete-logarithm) based Pedersen's commitment scheme [11] into ones with equivocality and extractability, and give efficient non-malleable commitment schemes.

One common drawback among those efficient non-malleable schemes is the fact that their corresponding simulators do not work in a straight-line manner, requires rewinding of the adversary. Due to this fact, such schemes are proved non-malleable only in the stand-alone cases. In the multiple-instances setting, i.e., when the scheme is performed concurrently with many instances of itself, such schemes cannot be proved non-malleable. In such a setting, one rewinding of $\mathcal{A}$ recursively invokes another rewinding of $\mathcal{A}$, which recursively invokes another rewinding..., eventually results in super-polynomial time simulation.

Damgård and Nielsen [6] show a universally composable (in particular, non-malleable in the multiple-instances setting) commitment scheme, which is as efficient as non-malleable schemes by [3], [8]. The key point of the scheme is the use of the *trapdoor* discrete logarithm problem. The trapdoor of the DLP (corresponding to the scheme's Common Reference String (CRS)) enables extractability without rewinding and leads to a straight-line simulator and UC-secureness of their scheme. Unfortunately, the trapdoor DLP needs a non-standard assumption, called the p-subgroup assumption. A more serious problem with the scheme is that the scheme requires a very strong trust in the third party who provides the CRS of the scheme. More precisely, the CRS of the scheme consists of the following pieces of information:

$$N, EK_1, EK_2, \ldots, EK_n.$$

Here, N is the system-wide modulus of the form $N = P^2Q$ with large primes P and Q, and EK_i is the (public) key to be used for making commitments for party P_i. That is, all parties using this protocol work in the same modulus N with tailored commitment keys for each of them. (So, the length of CRS is proportional to the number of parities involved.) Now suppose that the third party gets corrupted and the primes P and Q are known to the adversary. Then, the adversary can extract all of the messages under all of the commitments among all of the parities using the P and Q, just as he/she can decrypt ciphertexts using the (master) secret key. This means that the scheme requires a very strong trust in the third party providing the CRS, and once the third party gets corrupted all of the security of the scheme collapses catastrophically.

1.4 Our Result

We show another efficient DL-based commitment scheme. Our scheme is the first straight-line extractable commitment scheme based on the KEA1 assumption. Although our scheme may not be UC-secure, it is proven to be non-malleable even in the multiple-instances setting, using the straight-line extractor. Our simulator works in a straight-line manner by using the KEA1-extractor.

The KEA1 assumption is non-standard like the p-subgroup assumption used in [6]. Moreover, the KEA1 can be said "more non-standard," since it is of non-black-box type, i.e., the assumption depends on the code of the adversary. However, we believe especially when all (comparable) schemes we have are proved secure only under non-standard assumptions, it is desirable to have several schemes proved under different non-standard assumptions, since each of non-standard assumptions can collapse accidentally due to its non-standard property. So, it should be meaningful to have another new commitment scheme proved under the KEA1 assumption. More constructively, the use of KEA1 assumption brings us the following merit.

As an advantage compared to the scheme of [6], our scheme can avoid the above-mentioned catastrophic collapse of security by corrupting the third party providing the CRS. The advantage is the effect of the KEA1 assumption that enables extractability not only without rewinding but also without having any system-wide master trapdoor. In fact, in our scheme, even if trapdoors of CRS become known to the adversary, the adversary cannot extract messages under commitments. It is because the adversary cannot obtain non-black-box access to honest parties as the simulator does against adversaries in the proof of security. Note that to use KEA1-extractor one needs non-black-box access to the target.

Our commitment scheme satisfies the following theorem:

Theorem 1. *Under the KEA1 and DDH assumptions, the commitment scheme is non-malleable in the strong CRS model. Moreover, the simulator works straight-line in the strict-polynomial time.*

Since our simulator works straight-line without rewinding, the following corollary is immediate from the theorem:

Corollary 1. *Under the KEA1 and DDH assumptions, the commitment scheme is non-malleable in the strong CRS model in the multiple-instance setting.*

2. Definitions

First, we recall the definition of non-malleability of a com-

mitment scheme in the "strong" CRS model, and the definition of the KEA1 assumption.

2.1 Non-malleable Commitment in the Strong CRS Model

Definition 1 (non-malleable commitment in the strong CRS model)**.** *Let Com be a commitment scheme. Let $\mathcal{A}$ be any probabilistic polynomial-time adversary and R be any non-trivial polynomial-time computable relation over message space $\mathcal{M}$. Here, non-triviality of R means that R doesn't contain any reflexive pair (x, x). Define two experiments $\mathbf{Exp}^{\mathsf{real}}_{Com}(\mathcal{A}, R)$ and $\mathbf{Exp}^{\mathsf{sim}}_{Com}(\mathcal{A}, R)$ as follows (k is a security parameter, $\mathcal{L}$ and $\mathcal{R}$ denotes an honest sender and receiver, respectively):*

$\mathbf{Exp}^{\mathsf{real}}_{Com}(\mathcal{A}, R)$:
$\sigma, \sigma^* \leftarrow \{0,1\}^k$; $m \leftarrow \mathcal{M}$;
$\mathcal{L}$ commits to m for $\mathcal{A}$ by Com with c under CRS σ;
$\mathcal{A}$ commits for $\mathcal{R}$ by Com with c^* under CRS σ^*;
$\mathcal{L}$ sends decommit m, r of c to $\mathcal{A}$ under σ;
$\mathcal{A}$ sends decommit m^*, r^* of c^* to $\mathcal{R}$ under σ^*;
Output $R(m, m^*)$

$\mathbf{Exp}^{\mathsf{sim}}_{Com}(\mathcal{A}, R)$:
$\sigma^* \leftarrow \{0,1\}^k$; $m \leftarrow \mathcal{M}$;
$\mathcal{A}$ commits for $\mathcal{R}$ by Com with c^* under CRS σ^*;
$\mathcal{A}$ sends decommit m^*, r^* of c^* to $\mathcal{R}$ under σ^*;
Output $R(m, m^*)$

A commitment scheme Com is said to be non-malleable *(with respect to open) if for any $\mathcal{A}$ there exists some probabilistic polynomial-time algorithm Sim such that for any R we have*

$$\Pr[\mathbf{Exp}^{\mathsf{real}}_{Com}(\mathcal{A}, R) = 1] - \Pr[\mathbf{Exp}^{\mathsf{sim}}_{Com}(Sim, R) = 1] < \epsilon(\cdot)$$

with some negligible function ϵ.

The above definition of the non-malleability of commitment schemes is the standard one (used in, e.g., [3], [8]) with the exception that we are using a "strong" CRS model, that is, CRS's are randomly and independently chosen for the left and right sessions. The strong CRS model is common in UC-setting, as used in the scheme of [6]. On the while, note that schemes of [3], [8] use a "weak" CRS model, where a single CRS is shared among the two sessions. As seen later, our commitment scheme is non-malleable only in the strong CRS model, not in the weak CRS model. We believe there are scenarios where the strong CRS model is meaningful, for example, the case where some portions of CRS are prepared for every receivers, as in [6].

A commitment scheme is called *non-malleable in multiple-instance setting* when it is non-malleable even if many instances of the commitment scheme are performed concurrently in the presence of the MIM adversary $\mathcal{A}$. The adversary $\mathcal{A}$ receives polynomially-many commitments $c_1,..,c_n$ from left parties and manages to make relating commitments $c_1^*,..,c_n^*$ for right parties. The formal definition is a straightforward extension of Definition 1 and is omitted.

2.2 The KEA1 Assumption

The KEA1 assumption [1], [9] for group $G = \langle g \rangle$ means that it is possible only when one knows b to generate a pair (g^b, g^{ab}) for a randomly selected g^a.

Definition 2 (The KEA1 Assumption [1])**.** *Let G be a probabilistic polynomial-time algorithm (p.p.a.) which on the input of a security parameter k, outputs a prime number q of k bits and a generator g of a group of order q. For any string w and any p.p.a.'s G, H, H^*, an experiment $\mathbf{Exp}^{w}_{G,H,H^*}$ is defined as follows.*

$\mathbf{Exp}^{w}_{G,H,H^*}$:
$(q, g) \leftarrow G(1^k)$; $a \overset{\$}{\leftarrow} \mathbb{Z}_q$; $A = g^a$;
$(B, W) \leftarrow H(q, g, A, w)$;
$b \leftarrow H^*(q, g, A, w)$;
If $W = B^a$, $B \neq g^b$ then return 1; Else return 0.

G is called to satisfy the KEA1 assumption *if for any w and any adversary H there exists an extractor H^* with the negligible* $\mathbf{Adv}^{w}_{G,H,H^*}(k) = \mathbf{Pr}[\mathbf{Exp}^{w}_{G,H,H^*}(k) = 1]$.

3. Our Commitment Scheme

We describe our commitment scheme and show its hiding and binding properties. The proposed scheme uses a technique similar to a "twin encryption technique" [4], [12] and uses $\{q, g_0, h_0, g_1, h_1, \sigma\}$ as CRS. The q in the CRS is a prime order of group G with a generator g. We assume the KEA1 assumption for G. The rest of CRS are generated as follows:

$$g_0 \overset{\$}{\leftarrow} \langle g \rangle;\ e_1 \overset{\$}{\leftarrow} \mathbb{Z}_q,\ g_1 = g_0^{e_1};$$
$$d_0 \overset{\$}{\leftarrow} \mathbb{Z}_q,\ h_0 = g_0^{d_0}; d_1 \overset{\$}{\leftarrow} \mathbb{Z}_q,\ h_1 = g_1^{d_1}; \sigma \overset{\$}{\leftarrow} \{0,1\}^l.$$

σ is CRS for a non-interactive zero-knowledge argument system $\Pi = (l, P, V, S = (S_1, S_2))$ for the NP-language $\{(g, h, j, k) \mid \exists b \in \mathbb{Z}_q, j = g^b, k = h^b\}$ of DH tuples on G.

Commitment: $\mathcal{S}$ commits to message $m (\in \mathbb{Z}_q)$ for $\mathcal{R}$ as follows:

1. $\mathcal{R}$ randomly chooses $b_0, b_1 \in \mathbb{Z}_q^*$, and computes $j_0 = g_0^{b_0},\ k_0 = h_0^{b_0},\ j_1 = g_1^{b_1},\ k_1 = h_1^{b_1}$. Then, $\mathcal{R}$ computes ZK proof $\pi_i \leftarrow P((g_i, h_i, j_i, k_i), b_i, \sigma)$ for $i = 0, 1$. $\mathcal{R}$ sends $j_0, k_0, j_1, k_1, \pi_0, \pi_1$ to $\mathcal{S}$.
2. $\mathcal{S}$ verifies $V(\pi_i, (g_i, h_i, j_i, k_i), \sigma) = 1$ for $i = 0, 1$. If it is, $\mathcal{S}$ computes:

   $$r \overset{\$}{\leftarrow} \mathbb{Z}_q,\ a_1, a_2 \overset{\$}{\leftarrow} \mathbb{Z}_q^*;$$
   $$g_1' = g_1^{a_1},\ h_1' = h_1^{a_2},\ j_1' = j_1^{a_1},\ k_1' = k_1^{a_2};$$
   $$M_0 = g_0^m h_0^r,\ M_1 = {g_1'}^m {h_1'}^r,\ L_0 = j_0^m k_0^r,\ L_1 = {j_1'}^m {k_1'}^r$$

   and sends $g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1$ to $\mathcal{R}$.

3. $\mathcal{R}$ checks $j'_1 = {g'_1}^{b_1}$ and $k'_1 = {h'_1}^{b_1}$. If it is not, $\mathcal{R}$ aborts.

Decommitment: $\mathcal{S}$ opens the commitment to $\mathcal{R}$ canonically:

1. $\mathcal{S}$ sends m, r to $\mathcal{R}$.
2. $\mathcal{R}$ verifies all of the equations $M_0 = g_0^m h_0^r$, $M_1 = {g'_1}^m {h'_1}^r$, $L_0 = j_0^m k_0^r$ and $L_1 = {j'_1}^m {k'_1}^r$ hold. If it does, $\mathcal{R}$ outputs m. Otherwise it aborts.

It is easily seen that one can commit to a k-bit message with $O(k)$ bits in $O(k^3)$ computations with the scheme, using, e.g., an efficient NIZK scheme $\mathsf{compile}(\mathcal{P}_{eqdlog})$ of [5] as Π.

Lemma 1. *Under the DDH assumption, the proposed scheme is computationally hiding.*

Proof. (Sketch) Among all the messages from $\mathcal{S}$ to $\mathcal{R}$, those depending on m are $M_0 = g_0^m h_0^r, M_1 = {g'_1}^m {h'_1}^r, L_0 = j_0^m k_0^r$ and $L_1 = {j'_1}^m {k'_1}^r$. By the soundness of ZK argument system Π, we see that there are $b_0, b_1 \in \mathbb{Z}_q$ such that $L_0 = M_0^{b_0}, L_1 = M_1^{b_1}$. So, it is sufficient to show M_0, M_1 hides m computationally.

Let $s \xleftarrow{\$} \mathbb{Z}_q$ and $M'_1 = g_1^{a_1 m} h_1^{a_2 s}$. Obviously (M_0, M'_1) hides m perfectly. So, the claim follows if (M_0, M_1) and (M_0, M'_1) are computationally indistinguishable for any p.p.a. A. But, if some A distinguishes (M_0, M_1) and (M_0, M'_1), there must be a following distinguisher D against the DDH assumption:

> Distinguisher D on inputs $(h_0, h_0^r, h_1, h(= h_1^r \text{ or } h_1^s))$:
> $m \leftarrow \mathcal{M}$; $a_1, a_2 \xleftarrow{\$} \mathbb{Z}_q^*$; $g_0, g_1 \xleftarrow{\$} \langle g \rangle$;
> Sets g_0, h_0, g_1, h_1 as CRS;
> $g'_1 = g_1^{a_1}$, $h'_1 = h_1^{a_2}$;
> return $A(g'_1, h'_1, g_0^m h_0^r, {g'_1}^m h^{a_2})$.

□

As in the case of Pedersen's commitment scheme [11], we see the scheme is computationally binding under the DLA.

4. Equivocality and Extractability of the Scheme

We show the equivocality and extractability of our scheme. The former is used to simulate the left party against the adversary in MIM-setting, while the latter is used to extract the committed message from the adversary's commitment for the right party in the proof of non-malleability of the scheme. We use the following simple fact: If and only if we have $\log_{g_0} h_0 \neq \log_{g_1} h_1$, two equations $M_0 = g_0^m h_0^r$, $M_1 = g_1^m h_1^r$ among m, r determines m, r.

4.1 Equivocality

First, we show equivocality of the scheme, that is, there is a simulator Sim that can generate, using some trapdoor information on the CRS, commitments for adversary $\mathcal{A}$, which can be opened by any messages later.

Sim generates an equivocal commitment for adversary $\mathcal{A}$ as follows. Suppose $\mathcal{A}$, playing the role of receiver, sends the first message $j_0, k_0, j_1, k_1, \pi_0, \pi_1$ to Sim. After verifying the validity of proofs π_0 and π_1, Sim computes:

$$m_0 \leftarrow M,\ r_0 \xleftarrow{\$} \mathbb{Z}_q,\ a_1 \xleftarrow{\$} \mathbb{Z}_q^*,\ a_2 = d_0 d_1^{-1} a_1;$$
$$g'_1 = g_1^{a_1},\ h'_1 = h_1^{a_2},\ j'_1 = j_1^{a_1},\ k'_1 = k_1^{a_2};$$
$$M_0 = g_0^m h_0^r,\ M_1 = {g'_1}^m {h'_1}^r,\ L_0 = j_0^m k_0^r,\ L_1 = {j'_1}^m {k'_1}^r.$$

Then, Sim sends $g'_1, h'_1, j'_1, k'_1, M_0, M_1, L_0, L_1$ to $\mathcal{A}$.

Note the only difference between the simulated commitment and the honest one is in the generation of a_2: the simulated a_2 satisfies a relation $a_2 = d_0 d_1^{-1} a_1$ with a_1, but the real a_2 is independently random. We show in the proof of Theorem 1 $\mathcal{A}$ cannot distinguish between simulated commitments and real commitments under the DDH assumption.

It is easily seen that the simulated commitment is opened by any message m. In fact, since $a_2 = d_0 d_1^{-1} a_1$, we have $d_0 = \log_{g_0}(h_0) = \log_{g'_1}(h'_1)$. Moreover, by the soundness of the proofs π_0, π_1, we have $\log_{j_0}(k_0) = \log_{g_0}(h_0) = d_0$ and $\log_{j'_1}(k'_1) = \log_{g'_1}(h'_1) = d_0$. So, M_0, M_1, L_0, L_1 can be opened by any message m with $r = r_0 + (m_0 - m)/d_0$.

4.2 Extractability

Second, we show the extractability of the scheme, that is, there is a simulator Sim which can extract a message m from an adversary's commitment using some trapdoor information on the CRS.

Let $g_0, h_0, g_1, h_1, \sigma$ denote the CRS of the scheme. We assume a simulator Sim knows discrete logarithms e_1, d_0, d_1 among them: $g_1 = g_0^{e_1}, h_0 = g_0^{d_0}, h_1 = g_1^{d_1}$. Suppose Sim, simulating a receiver, honestly sends the first message to an adversary $\mathcal{A}$ who plays a sender's role, i.e., Sim randomly chooses $b_0, b_1 \in \mathbb{Z}_q^*$, and computes

$$j_0 = g_0^{b_0},\ k_0 = h_0^{b_0},\ j_1 = g_1^{b_1},\ k_1 = h_1^{b_1};$$
$$\pi_i \leftarrow P(\mathrm{DH}(g_i, h_i, j_i, k_i), b_i, \sigma) \text{ for } i = 0, 1;$$

and sends $j_0, k_0, j_1, k_1, \pi_0, \pi_1$ to $\mathcal{A}$.

Then, $\mathcal{A}$ sends commitment $g'_1, h'_1, j'_1, k'_1, M_0, M_1, L_0, L_1$ to Sim. On the condition that the commitment should be opened correctly, we have equalities

$$j'_1 = {g'_1}^{b_1},\ k'_1 = {h'_1}^{b_1} \tag{1}$$

and there must be m, r such that $M_0 = g_0^m h_0^r$, $M_1 = {g'_1}^m {h'_1}^r$, $L_0 = j_0^m k_0^r$, $L_1 = {j'_1}^m {k'_1}^r$.

In the above, Sim sends a randomly selected j_1 to $\mathcal{A}$ and $\mathcal{A}$ returns to Sim (g'_1, j'_1) which constitute a DH-tuple (g_1, j_1, g'_1, j'_1) by Eq. (1). That is, $\mathcal{A}$ is seen as playing the role of the KEA1-adversary against Sim. So, using the corresponding KEA1-extractor, Sim can extract a_1 satisfying $g'_1 = g_1^{a_1}$. Similarly, Sim obtains a_2 satisfying $h'_1 = h_1^{a_2}$.

Then, Sim can compute $\alpha = e_1 a_1,\ \beta = e_1 d_0^{-1} d_1 a_2$ and get

$$g_0^m = (M_0^{\beta} M_1^{-1})^{\frac{1}{\beta-\alpha}}. \quad (2)$$

Here, note that $\alpha = \beta$ happens with only a negligible probability, because if $\alpha = \beta$ then $g_0, h_0, g_1, {h'_1}^{1/a_1}$ constitute a DH-tuple and $\mathcal{A}$ should violate CDH assumption. (If the same CRS was used among the left and right sessions in the MIM-setting, $\mathcal{A}$ could reuse the h'_1, which was generated by Sim in the left session, also in the right session, and then the fact $\alpha = \beta$ (in the right session) doesn't imply the violation of CDH assumption. This is why we need the strong CRS model.)

Since $\log_{j_0} j_1 = \log_{g_0} g_1 = e_1$, $\log_{j_0} k_0 = \log_{g_0} h_0 = d_0$ and $\log_{j_1} k_1 = \log_{g_1} h_1 = d_1$, using the same α, β, Sim gets also

$$j_0^m = (L_0^{\beta} L_1^{-1})^{\frac{1}{\beta-\alpha}}. \quad (3)$$

Thus, Sim can compute g_0^m, j_0^m from $\mathcal{A}$'s commitment $g'_1, h'_1, j'_1, k'_1, M_0, M_1, L_0, L_1$ and the trapdoor e_1, d_0, d_1 on the CRS. Here, $\mathcal{A}$ is seen as playing the role of the KEA1-adversary again: $\mathcal{A}$ gets a random j_0 and returns g_0^m, j_0^m. Using the corresponding KEA1-extractor, Sim can extract the message m.

5. Non-malleability of the Commitment Scheme

Now we show the main theorem:

Theorem 1. *Under the KEA1 and DDH assumptions, the commitment scheme is non-malleable in the strong CRS model. Moreover, the simulator works straight-line in the strict-polynomial time.*

Since our simulator works straight-line without rewinding, the following corollary is immediate from the theorem:

Corollary 1. *Under the KEA1 and DDH assumptions, the commitment scheme is non-malleable in the strong CRS model in the multiple-instance setting.*

Before proceeding to the formal proof of Theorem 1, we point out some key-points that make the proof work. Suppose an adversary $\mathcal{A}$ in the MIM setting is given. Simulator Sim generates independent CRS's for the left and right sessions with trapdoor information as specified in the scheme. $\mathcal{A}$ is invoked given the CRS. In order to prove non-malleability, Sim has to "simulate from the left and extract from the right" against $\mathcal{A}$.

Simulate from the left: To simulate the left view of $\mathcal{A}$ without the knowledge of the real message m, Sim as a left party commits to a dummy message m_0 for $\mathcal{A}$ through an equivocal commitment c using the trapdoor information as shown in Sect. 4.1. By equivocality, the commitment c cannot be distinguished from real commitments and it can be opened by the real message m instead of m_0, later. This indicates that Sim can correctly commit to m without the knowledge of m.

When doing an equivocal commitment, Sim fakes a_1, a_2 as shown in Sect. 4.1. We will show an equivocal commitment with the faked a_1, a_2 is indistinguishable from a real one under the DDH assumption.

Extract from the right: From the commitment c^* generated by $\mathcal{A}$, Sim, as the simulated right party, can extract the committed message m^* by using the trapdoor information and the suitable KEA1-extractors as descried in Sect. 4.2. Here, note that when $\mathcal{A}$ generated c^*, $\mathcal{A}$ only knew an equivocal commitment c. So, m^* must be independent of m. This means that Sim can find m^* without using m at all with the same probability that $\mathcal{A}$ outputs such m^* in the real MIM setting. This means non-malleability of the commitment scheme.

Proof of Theorem 1

Suppose an adversary $\mathcal{A}$ against the scheme in the MIM setting and a nontrivial computable relation R on message space $\mathcal{M}$ is given (here, non-triviality of R means that R doesn't contain any reflexive pair (x, x)). In the following, we define six experiments $\mathbf{Exp}_0, \mathbf{Exp}_1, \cdots, \mathbf{Exp}_5$ below (for the full description of those experiments, see Appendix). $\mathbf{Exp}_0$ is identical to the real experiment $\mathbf{Exp}_{Com}^{\mathsf{real}}$ specified with our commitment scheme and $\mathbf{Exp}_5$ is seen as the ideal experiment $\mathbf{Exp}_{Com}^{\mathsf{sim}}$ specified with our commitment scheme and a suitable simulator derived from the adversary in $\mathbf{Exp}_0$ (see Definition 1). In order to prove the non-malleability of the scheme, we need to show outputs of $\mathbf{Exp}_0$ and $\mathbf{Exp}_5$ are indistinguishable. We do that by showing (outputs of) $\mathbf{Exp}_i$ and $\mathbf{Exp}_{i+1}$ are indistinguishable for $i = 0$ to 4 step by step.

As stated, $\mathbf{Exp}_0$ is $\mathbf{Exp}_{Com}^{\mathsf{real}}$ specified with our commitment scheme. The only difference between $\mathbf{Exp}_1$ and $\mathbf{Exp}_0$ is in the generation of a_2: a_2 in $\mathbf{Exp}_1$ is equal to $d_0 d_1^{-1} a_1$ (instead of a random element). To prove $\mathbf{Exp}_1$ is indistinguishable from $\mathbf{Exp}_0$, we first define a game $\mathbf{Game}_{LR}(D_{LR})$ for a probabilistic polynomial-time algorithm D_{LR} as follows.

$\mathbf{Game}_{LR}(D_{LR})$:
$g_0 \xleftarrow{\$} \langle g \rangle$; $e_1, d_0, d_1 \xleftarrow{\$} \mathbb{Z}_q$, $g_1 = g_0^{e_1}, h_0 = g_0^{d_0}, h_1 = g_1^{d_1}$; $c_{LR} \xleftarrow{\$} \{0, 1\}$;
Invoke $D_{LR}((g_0, h_0), (g_1, h_1))$;
  If D_{LR} makes a query j_1, k_1, then
    If $k_1 \neq j_1^{d_1}$, then abort;
    $a_1 \xleftarrow{\$} \mathbb{Z}_q^*$;
    If $c_{LR} = 0$, then $a_2 \xleftarrow{\$} \mathbb{Z}_q^*$ else $a_2 = d_0 d_1^{-1} a_1$;
    $g'_1 = g_1^{a_1}, h'_1 = h_1^{a_2}, j'_1 = {j_1}^{a_1}, k'_1 = {k_1}^{a_2}$;
    Return g'_1, h'_1, j'_1, k'_1 to D_{LR};
D_{LR} outputs $\hat{c}$ and halt;
Output $\hat{c}$;

In the above, D_{LR} is supposed to make a query once at most. The advantage of D_{LR} is defined by $\mathbf{Adv}_{D_{LR}} = 2\Pr[\hat{c} = c_{LR}] - 1$. We call $G = \langle g \rangle$ *LR-secure* if the advantage $\mathbf{Adv}_{D_{LR}}$ is negligible for any probabilistic polynomial-time algorithm D_{LR}. As to *LR*-secureness, we show two lemmas:

Lemma 2. *If G is LR-secure, then* $\mathbf{Exp}_1$ *is indistinguishable from* $\mathbf{Exp}_0$.

Proof. Suppose, on the contrary, there is a distinguisher $D_{0,1}$ between $\mathbf{Exp}_0$ and $\mathbf{Exp}_1$ with a non-negligible advantage. We construct an adversary D_{LR} with a non-negligible advantage against G in $\mathbf{Game}_{LR}$ by using $D_{0,1}$.

According to the definition of $\mathbf{Game}_{LR}$, we prepare with g_0, h_0, g_1, h_1 and $c \xleftarrow{\$} \{0,1\}$. Given g_0, h_0 and g_1, h_1, the D_{LR} proceeds as follows. First, D_{LR} chooses a message m and uniformly selects σ from $\{0,1\}^l$. Moreover, D_{LR} uniformly select u_0 from G, t_1, s_0, s_1 from $\mathbb{Z}_q$, and η from $\{0,1\}^l$, and computes $u_1 = u_0^{t_1}, v_0 = u_0^{s_0}, v_1 = u_1^{s_1}$. Then, D_{LR} sets $\{g_0, h_0, g_1, h_1, \sigma\}$ as the left CRS and $\{u_0, v_0, u_1, v_1, \eta\}$ as the right CRS, and invokes the adversary $\mathcal{A}$. D_{LR}, simulating the right party, generates the first message $j_0, k_0, j_1, k_1, \pi_0, \pi_1$ honestly and sends it to $\mathcal{A}$. $\mathcal{A}$ is supposed to send $j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$ to the left party simulated by D_{LR}. D_{LR}, after verifying proofs π_0 and π_1 just like the honest left party, makes a query j_1^*, k_1^* to the oracle in $\mathbf{Game}_{LR}$, which replies with g_1', h_1', j_1', k_1' (note by the soundness of the proof π_1, we have $k_1^* = j_1^{*d_1}$). From now on, using this g_1', h_1', j_1', k_1' as a real g_1', h_1', j_1', k_1', D_{LR} proceeds just as in $\mathbf{Exp}_0$ (or $\mathbf{Exp}_1$) until $\mathcal{A}$ outputs some decommit message m^*, r^*. Finally, D_{LR} outputs $D_{0,1}(R(m, m^*))$.

It is obvious that the distribution of $R(m, m^*)$ in the above is identical to the one of $\mathbf{Exp}_0$ if $c = 0$ and to $\mathbf{Exp}_1$ if $c = 1$. So, the advantage of $D_{0,1}$ is transferred to the one of D_{LR}, which makes a contradiction to the assumption of *LR*-secureness of G. □

Lemma 3. *G is LR-secure under the DDH and KEA1 assumptions on G.*

Proof. Suppose, on the contrary, there is an adversary D_{LR} with a non-negligible advantage against $\mathbf{Game}_{LR}$ in G. We construct a DDH distinguisher D_{ddh} with a non-negligible advantage on G by using D_{LR} and a suitable KEA1-extractor.

We prepare with an input g_0, h_0, g_1', h_1' for D_{ddh} as usual:

$g_0 \xleftarrow{\$} \langle g \rangle$;
$d_0, e_1' \xleftarrow{\$} \mathbb{Z}_q,\ h_0 = g_0^{d_0}, g_1' = g_0^{e_1'}$;
$c_{DDH} \xleftarrow{\$} \{0,1\}$;
If $c_{DDH} = 0$ then $d_1' \xleftarrow{\$} \mathbb{Z}_q$ else $d_1' = d_0$;
$h_1' = g_1'^{d_1'}$;

Given g_0, h_0, g_1', h_1', the D_{ddh} proceeds as follows. First, D_{ddh} generates two random elements g_1, h_1 on G by $e_1, d_1 \xleftarrow{\$} \mathbb{Z}_q$, $g_1 = g_0^{e_1}, h_1 = g_1^{d_1}$. Next, D_{ddh} invokes D_{LR} with inputs of g_0, h_0, g_1, h_1 and with a random tape R. When, D_{LR} makes a query j_1, k_1 (remember such query is once at most), D_{ddh} verifies $k_1 = j_1^{d_1}$ (if not it aborts), and calls the KEA1-extractor H^* (described below) with inputs of g_1, h_1 and an auxiliary input g_0, h_0, R to get an output b. Then, D_{ddh} computes $j_1' = g_1'^{\,b}$ and $k_1' = h_1'^{\,b}$, and replies D_{LR} with g_1', h_1', j_1', k_1'. Finally, D_{ddh} outputs an output $\hat{c}$ of D_{LR}.

In the above, H^* is the KEA1-extractor corresponding to

> $H(g_1, h_1; g_0, h_0; R)$:
>
> Invoke D_{LR} with the input of g_1, h_1, g_0, h_0
> and the random tape of R;
> When D_{LR} makes a query j_1, k_1, output j_1, k_1;

We can suppose outputs of H satisfy $k_1 = j_1^d$ conditioned on the success of D_{LR}. So, by the KEA1 assumption, we have

$$j_1 = g_1^b,\ k_1 = h_1^b \tag{4}$$

for $b = H^*(g_1, h_1; g_0, h_0, R)$ with negligible exceptions.

Let

$$a_1 = e_1'/e_1,\ a_2 = a_1 d_1'/d_1.$$

It is directly verified that

$$g_1' = g_1^{a_1},\ h_1' = h_1^{a_2}.$$

Then, under Eq. (4), we have

$$j_1' = g_1'^{\,b} = g_1^{a_1 b} = j_1^{a_1}$$
$$k_1' = h_1'^{\,b} = h_1^{a_2 b} = k_1^{a_2}.$$

Moreover,

$$c_{DDH} = 1 \Leftrightarrow d_1' = d_0 \Leftrightarrow a_2 = a_1 d_0/d_1.$$

Thus, we see that in the case of Eq. (4) holds, the view of D_{LR} simulated by D_{ddh} on a DDH-tuple is identical to the view of D_{LR} in $\mathbf{Game}_{LR}$ with $c_{LR} = 0$ and the view simulated on a random tuple is identical to the view with $c_{LR} = 1$. So, the advantage of D_{LR} is transferred to the one of D_{ddh} with a negligible loss, and a contradiction to the DDH assumption. □

Lemmas 2 and 3 show that $\mathbf{Exp}_1$ is indistinguishable from $\mathbf{Exp}_0$ under the DDH and KEA1 assumptions on G.

We proceed to $\mathbf{Exp}_2$. The only difference between $\mathbf{Exp}_2$ and $\mathbf{Exp}_1$ is that the simulated left party in $\mathbf{Exp}_2$ commits to a dummy message m_0 instead of m. However, since a_1 and a_2 are faked both in $\mathbf{Exp_1}$ and $\mathbf{Exp_2}$, the commitment M_0, M_1, L_0, L_1 is equivocal in both experiments. Hence $\mathbf{Exp}_2$ is indistinguishable from $\mathbf{Exp}_1$.

We proceed to $\mathbf{Exp}_3$. The only difference between $\mathbf{Exp}_3$ and $\mathbf{Exp}_2$ is that we use, in the right session of $\mathbf{Exp}_3$, the simulator (S_1, S_2) of the non-interactive zero-knowledge argument system $\Pi = (l, P, V, S = (S_1, S_2))$ instead of the real prover P (and don't use the witness b_i). By the zero-knowledge property of Π, $\mathbf{Exp}_3$ is indistinguishable from $\mathbf{Exp}_2$.

We proceed to $\mathbf{Exp}_4$. The only difference between $\mathbf{Exp}_4$ and $\mathbf{Exp}_3$ is that we use the KEA1-extractor H^* to extract the message n^* under the adversary's commitment and output $R(m, n^*)$ instead of $R(m, m^*)$ in $\mathbf{Exp}_4$. The KEA1-extractor H^* corresponds to the following KEA1-adversary H. Given u_0, $j_0(= u_0^{b_0})$ and auxiliary input w (independent of b_0), H outputs $u_0^{n^*}$, $j_0^{n^*}$ by simulating the MIM-setting against $\mathcal{A}$:

$H(u_0, j_0; w = (R_A, m, g_0, e_1, d_0, d_1, \sigma,$
$t_1, s_0, s_1, z, b_1, z_0, z_1, m_0, r_0, a_1))$:

Set $(g_0, h_0 = g_0^{d_0}, g_1 = g_0^{e_1}, h_1 = g_1^{d_1}, \sigma)$ as the left CRS and $(u_0, v_0 = u_0^{s_0}, u_1 = u_0^{t_1}, v_1 = u_1^{s_1}, \eta)$ as the right CRS with $(\eta, \xi) = S_1(1^k; z)$;
Invoke $\mathcal{A}$ with random tape R_A;
Compute simulated proofs π_0, π_1 as in $\mathbf{Exp}_4$ and send $j_0, k_0 = j_0^{s_0}, j_1 = u_1^{b_1}, k_1 = v_1^{b_1}, \pi_0, \pi_1$ to $\mathcal{A}$, which in turn sends $j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$ to $\mathcal{L}$;
Generate $g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1$ as in $\mathbf{Exp}_4$ and send it to $\mathcal{A}$, which in turn sends u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* to $\mathcal{R}$;
$a_1^* := H_1^*(u_1, j_1; w_1)$; $a_2^* := H_2^*(v_1, k_1; w_2)$;
(Here, $w_1 = w_2 = (R_A, m, g_0, e_1, d_0, d_1, \sigma, u_0, v_0, s_1, z,$ $j_0, k_0, z_0, z_1, m_0, r_0, a_1))$ is independent of b_1.)
$\alpha := t_1 a_1^*$; $\beta := t_1 s_0^{-1} s_1 a_2^*$; If $\alpha = \beta$ then abort;
Output $((M_0^{*\beta} M_1^{*-1})^{1/(\beta-\alpha)}, (L_0^{*\beta} L_1^{*-1})^{1/(\beta-\alpha)})$;

In the above, the KEA1-extractor H_1^* in H corresponds to KEA1-adversary H_1, which given u_1, $j_1(= u_1^{b_1})$ and w_1 (independent of b_1), outputs u_1^*, j_1^* as follows:

$H_1(u_1, j_1; w_1 = (R_A, m, g_0, e_1, d_0, d_1, \sigma,$
$u_0, v_0, s_1, z, j_0, k_0, z_0, z_1, m_0, r_0, a_1))$:

Set $(g_0, h_0 = g_0^{d_0}, g_1 = g_0^{e_1}, h_1 = g_1^{d_1}, \sigma)$ as the left CRS and $(u_0, v_0, u_1, v_1 = u_1^{s_1}, \eta)$ as the right CRS with $(\eta, \xi) = S_1(1^k; z)$;
Invoke $\mathcal{A}$ with random tape R_A;
Compute simulated proofs π_0, π_1 as in $\mathbf{Exp}_4$ and send $j_0, k_0, j_1, k_1 = j_1^{s_1}, \pi_0, \pi_1$ to $\mathcal{A}$, which in turn sends $j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$ to $\mathcal{L}$;
Generate $g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1$ as in $\mathbf{Exp}_4$ and send it to $\mathcal{A}$, which in turn sends u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* to $\mathcal{R}$;
Output (u_1^*, j_1^*);

The KEA1-extractor H_2^* in H corresponds to KEA1-adversary H_2, which given $v_1, k_1(= v_1^{b_1})$ and $w_2(= w_1)$, outputs v_1^*, k_1^* in the similar way as H_1. We omit the details of H_2. The description of $\mathbf{Exp}_4$ is completed.

Now we show $\mathbf{Exp}_4$ is indistinguishable from $\mathbf{Exp}_3$. First, note that the view of $\mathcal{A}$ in H, H_1, H_2 is identical to the view of $\mathcal{A}$ in $\mathbf{Exp}_3$ (or $\mathbf{Exp}_4$), because we distribute the randomness used in $\mathbf{Exp}_3$ (or $\mathbf{Exp}_4$) to H, H_1, H_2 through auxiliary inputs. So, it is sufficient to show n^* derived by H^* is equal to m^* decommitted by $\mathcal{A}$ with an overwhelming probability conditioned on $\mathcal{A}$'s success. This is nothing but the extractability.

More precisely, by the KEA1 assumption on (H_1, H_1^*), we see that the output a_1^* of H_1^* satisfies

$$u_1^* = u_1^{a_1^*},\ j_1^* = j_1^{a_1^*} \tag{5}$$

with only negligible exceptions. Similarly, a_2^* satisfies

$$v_1^* = v_1^{a_2^*},\ k_1^* = k_1^{a_2^*} \tag{6}$$

with only negligible exceptions. Since m^*, r^* in $\mathbf{Exp}_3$ is a valid decommitment for $M_0^*, M_1^*, L_0^*, L_1^*$ (conditioned on $\mathcal{A}$'s success), we have $M_0^* = u_0^{m^*} v_0^{r^*}$, $M_1^* = u_1^{*m^*} v_1^{*r^*}$, $L_0^* = j_0^{m^*} k_0^{r^*}$, $L_1^* = j_1^{*m^*} k_1^{*r^*}$. By Eqs. (2) and (3) in Sect. 4.2 together with Eqs. (5) and (6), we see that, if α, β (defined in the description of H) are not equal, $u_0^{m^*} = (M_0^{*\beta} M_1^{*-1})^{1/(\beta-\alpha)}$, $j_0^{m^*} = (L_0^{*\beta} L_1^{*-1})^{1/(\beta-\alpha)}$ and the KEA1 assumption on (H, H^*) means that $n^* = m^*$ with only negligible exceptions. The remaining case $\alpha = \beta$ happens with only a negligible probability by the following Lemma 4:

Lemma 4. *The case of $\alpha = \beta$ happens with only a negligible probability under Discrete Logarithmic Assumption.*

Proof. Assume, on the contrary, the case $\alpha = \beta$ happens with a non-negligible probability. We construct discrete-logarithm extractor E as follows:

$E(u_0, v_0(= u_0^{s_0}))$:

Generate randomness $(R_A, m, g_0, e_1, d_0, d_1, \sigma,$ $t_1, s_1, z, b_0, b_1, z_0, z_1, m_0, r_0, a_1))$ as in $\mathbf{Exp}_4$ (or $\mathbf{Exp}_3$);
Set $(g_0, h_0 = g_0^{d_0}, g_1 = g_0^{e_1}, h_1 = g_1^{d_1}, \sigma)$ as the left CRS and $(u_0, v_0, u_1 = u_0^{t_1}, v_1 = u_1^{s_1}, \eta)$ as the right CRS with $(\eta, \xi) = S_1(1^k; z)$;
Invoke $\mathcal{A}$ with random tape R_A;
Compute $j_0, k_0, j_1, k_1, \pi_0, \pi_1$ as in $\mathbf{Exp}_4$ and send it to $\mathcal{A}$, which in turn sends $j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$ to $\mathcal{L}$;
Generate $g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1$ as in $\mathbf{Exp}_4$ and send it to $\mathcal{A}$, which in turn sends u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* to $\mathcal{R}$;
$a_1^* := H_1^*(u_1, j_1; w_1)$; $a_2^* := H_2^*(v_1, k_1; w_2)$;
(Here, $w_1 = w_2 = (R_A, m, g_0, e_1, d_0, d_1, \sigma, u_0, v_0, s_1,$ $z, j_0, k_0, z_0, z_1, m_0, r_0, a_1))$ is independent of b_1.)
Output $s_1 a_2^* / a_1^*$;

Recall $\alpha = t_1 a_1^*$ and $\beta = t_1 s_0^{-1} s_1 a_2^*$ by definition. So, $\alpha = \beta$ means $a_1^* = s_0^{-1} s_1 a_2^*$, and this means that $s_0 = s_1 a_2^* / a_1^*$. □

Thus, we have shown that $\mathbf{Exp}_4$ is indistinguishable from $\mathbf{Exp}_3$.

We arrive at the final experiment $\mathbf{Exp}_5$. The only difference between $\mathbf{Exp}_5$ and $\mathbf{Exp}_4$ is that the simulated left party sends (m_0, r_0) as decommit information instead of (m, r) in $\mathbf{Exp}_5$. Since n^* is determined before decommitment by the left party, it is obvious that $\mathbf{Exp}_5$ is identically distributed to $\mathbf{Exp}_4$. Moreover, note that the simulated left party $\mathcal{L}$ in

$\mathbf{Exp}_5$ doesn't use the message m. So, $\mathcal{L}$ and $\mathcal{A}$ in $\mathbf{Exp}_5$ constitute the desired simulator Sim in the ideal experiment $\mathbf{Exp}^{\mathsf{sim}}_{Com}$. The proof is completed.

6. Conclusion

We showed another efficient DL-based commitment scheme, which is proven to be non-malleable even in the multiple-instances setting, based on the KEA1 and DDH assumptions. Our scheme has a simulator that works in a straight-line manner by using the KEA1-extractor instead of the rewinding strategy. The KEA1 assumption enables our scheme's extractability not only without rewinding but also without having any system-wide master trapdoor. So, our scheme can avoid the catastrophic collapse of security possible by corrupting the third party providing the CRS.

References

[1] M. Bellare and A. Palacio, "The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols," Proc. Crypto 2004, pp.273–289, LNCS 3152, 2004.
[2] G.D. Crescenzo, Y. Ishai, and R. Ostrovsky, "Non-interactive and non-malleable commitment," STOC 1998, pp.141–150, 1998.
[3] G.D. Crescenzo, J. Katz, R. Ostrovsky, and A. Smith, "Efficient and non-interactive non-malleable commitments," Proc. EuroCrypt 2001, pp.40–59, LNCS 2045, 2001.
[4] I. Damgård, "Towards practical public key systems secure against chosen ciphertext attacks," CRYPTO'91, pp.445–456, 1991.
[5] I. Damgård, N. Fazio, and A. Nicolosi, "Non-interactive zero-knowledge from homomorphic encryption," Proc. TCC 2006, pp.41–59, LNCS 3876, 2006.
[6] I. Damgård and J.B. Nielsen, "Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor," Proc. Crypto 2002, pp.581–596, LNCS 2442, 2002.
[7] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," SIAM J. Computing, vol.30, pp.391–437, 2000.
[8] M. Fischlin and R. Fischlin, "Efficient non-malleable commitment schemes," Proc. Crypto 2000, pp.413–431, LNCS 1880, 2000.
[9] S. Hada and T. Tanaka, "On the existence of 3-round zero-knowledge protocols," Proc. Crypto'98, pp.408–423, LNCS 1462, 1998.
[10] J. Katz and Y. Lindell, "Handling expected polynomial-time strategies in simulation-based security proofs," Proc. TCC 2005, pp.128–149, LNCS 3378, 2005.
[11] T.P. Pedersen, "Non-interactive and information-theoretical secure verifiable secret sharing," Proc. Crypto'91, pp.129–140, LNCS 576, 1991.
[12] C. Rackoff and D.R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," CRYPTO'91, pp.433–444, 1991.

Appendix: Experiments in the Proof of Theorem 1

$\mathbf{Exp}_0$:

$m \leftarrow \mathcal{M}$, $g_0 \overset{\$}{\leftarrow} G$, $e_1, d_0, d_1 \overset{\$}{\leftarrow} \mathbb{Z}_q$;
$g_1 = g_0^{e_1}$, $h_0 = g_0^{d_0}$, $h_1 = g_1^{d_1}$, $\sigma \overset{\$}{\leftarrow} \{0,1\}^l$;
$u_0 \overset{\$}{\leftarrow} G$, $t_1, s_0, s_1 \overset{\$}{\leftarrow} \mathbb{Z}_q$, $u_1 = u_0^{t_1}$, $v_0 = u_0^{s_0}$, $v_1 = u_1^{s_1}$;
$\eta \overset{\$}{\leftarrow} \{0,1\}^l$;
Invoke $\mathcal{A}$ with the left CRS $(g_0, h_0, g_1, h_1, \sigma)$ and the right CRS $(u_0, v_0, u_1, v_1, \eta)$;

$\mathcal{L}$	$\mathcal{A}$	$\mathcal{R}$
		$b_0, b_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;
		$j_0 = u_0^{b_0}, k_0 = v_0^{b_0}$;
		$j_1 = u_1^{b_1}, k_1 = v_1^{b_1}$;
		$\pi_0 \leftarrow P((u_0, v_0, j_0, k_0), b_0, \eta)$;
		$\pi_1 \leftarrow P((u_1, v_1, j_1, k_1), b_1, \eta)$;
		$\Leftarrow j_0, k_0, j_1, k_1, \pi_0, \pi_1$;
	$\Leftarrow j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$;	
Verify π_0^* and π_1^*; If not valid, abort;		
$r \overset{\$}{\leftarrow} \mathbb{Z}_q$, $a_1, a_2 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;		
$g_1' = g_1^{a_1}$, $h_1' = h_1^{a_2}$;		
$j_1' = j_1^{*a_1}$, $k_1' = k_1^{*a_2}$;		
$M_0 = g_0^m h_0^r$;		
$M_1 = g_1'^m h_1'^r$;		
$L_0 = j_0^{*m} k_0^{*r}$;		
$L_1 = j_1'^m k_1'^r$;		
$g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1 \Rightarrow$;		
	$u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* \Rightarrow$;	
		Verify $j_1^* = u_1^{*b_1}$ and $k_1^* = v_1^{*b_1}$; If not valid, abort;
$m, r \Rightarrow$;		
	$m^*, r^* \Rightarrow$;	
		Verify all of the following: $M_0^* = u_0^{m^*} v_0^{r^*}$, $M_1^* = u_1^{*m^*} v_1^{*r^*}$, $L_0^* = j_0^{m^*} k^{r^*}{}_0$, $L_1^* = j_1^{*m^*} k_1^{*r^*}$; If not valid, abort;

Output $R(m, m^*)$;

$\mathbf{Exp}_1$:

$m \leftarrow \mathcal{M}, g_0 \overset{\$}{\leftarrow} G,\ e_1, d_0, d_1 \overset{\$}{\leftarrow} \mathbb{Z}_q$;
$g_1 = g_0^{e_1}, h_0 = g_0^{d_0}, h_1 = g_1^{d_1}, \sigma \overset{\$}{\leftarrow} \{0,1\}^l$;
$u_0 \overset{\$}{\leftarrow} G,\ t_1, s_0, s_1 \overset{\$}{\leftarrow} \mathbb{Z}_q, u_1 = u_0^{t_1}, v_0 = u_0^{s_0}, v_1 = u_1^{s_1}$;
$\eta \overset{\$}{\leftarrow} \{0,1\}^l$;
Invoke $\mathcal{A}$ with the left CRS $(g_0, h_0, g_1, h_1, \sigma)$ and the right CRS $(u_0, v_0, u_1, v_1, \eta)$;

$\mathcal{L}$	$\mathcal{A}$	$\mathcal{R}$
		$b_0, b_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;
		$j_0 = u_0^{b_0}, k_0 = v_0^{b_0}$;
		$j_1 = u_1^{b_1}, k_1 = v_1^{b_1}$;
		$\pi_0 \leftarrow P((u_0, v_0, j_0, k_0), b_0, \eta)$;
		$\pi_1 \leftarrow P((u_1, v_1, j_1, k_1), b_1, \eta)$;
		$\Leftarrow j_0, k_0, j_1, k_1, \pi_0, \pi_1$;
	$\Leftarrow j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$;	
Verify π_0^* and π_1^*; If not valid, abort;		
$r \overset{\$}{\leftarrow} \mathbb{Z}_q,\ a_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;		
$\boxed{a_2 = d_0 d_1^{-1} a_1;}$		
$g_1' = g_1^{a_1}, h_1' = h_1^{a_2}$;		
$j_1' = j_1^{*a_1}, k_1' = k_1^{*a_2}$;		
$M_0 = g_0^m h_0^r$;		
$M_1 = g_1'^m h_1'^r$;		
$L_0 = j_0^{*m} k_0^{*r}$;		
$L_1 = j_1'^m k_1'^r$;		
$g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1 \Rightarrow$;		
	$u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* \Rightarrow$;	
		Verify $j_1^* = u_1^{*b_1}$ and $k_1^* = v_1^{*b_1}$; If not valid, abort;
$m, r \Rightarrow$;		
	$m^*, r^* \Rightarrow$;	
		Verify all of the following: $M_0^* = u_0^{m^*} v_0^{r^*}$, $M_1^* = u_1^{*m^*} v_1^{*r^*}$, $L_0^* = j_0^{m^*} k_0^{r^*}$, $L_1^* = j_1^{*m^*} k_1^{*r^*}$; If not valid, abort;
Output $R(m, m^*)$;		

$\mathbf{Exp}_2$:

$m \leftarrow \mathcal{M}, g_0 \overset{\$}{\leftarrow} G,\ e_1, d_0, d_1 \overset{\$}{\leftarrow} \mathbb{Z}_q$;
$g_1 = g_0^{e_1}, h_0 = g_0^{d_0}, h_1 = g_1^{d_1}, \sigma \overset{\$}{\leftarrow} \{0,1\}^l$;
$u_0 \overset{\$}{\leftarrow} G,\ t_1, s_0, s_1 \overset{\$}{\leftarrow} \mathbb{Z}_q, u_1 = u_0^{t_1}, v_0 = u_0^{s_0}, v_1 = u_1^{s_1}$;
$\eta \overset{\$}{\leftarrow} \{0,1\}^l$;
Invoke $\mathcal{A}$ with the left CRS $(g_0, h_0, g_1, h_1, \sigma)$ and the right CRS $(u_0, v_0, u_1, v_1, \eta)$;

$\mathcal{L}$	$\mathcal{A}$	$\mathcal{R}$
		$b_0, b_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;
		$j_0 = u_0^{b_0}, k_0 = v_0^{b_0}$;
		$j_1 = u_1^{b_1}, k_1 = v_1^{b_1}$;
		$\pi_0 \leftarrow P((u_0, v_0, j_0, k_0), b_0, \eta)$;
		$\pi_1 \leftarrow P((u_1, v_1, j_1, k_1), b_1, \eta)$;
		$\Leftarrow j_0, k_0, j_1, k_1, \pi_0, \pi_1$;
	$\Leftarrow j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$;	
Verify π_0^* and π_1^*; If not valid, abort;		
$\boxed{m_0 \leftarrow \mathcal{M}, r_0 \overset{\$}{\leftarrow} \mathbb{Z}_q;}$		
$a_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*,\ a_2 = d_0 d_1^{-1} a_1$;		
$g_1' = g_1^{a_1}, h_1' = h_1^{a_2}$;		
$j_1' = j_1^{*a_1}, k_1' = k_1^{*a_2}$;		
$M_0 = g_0^{m_0} h_0^{r_0}$;		
$M_1 = g_1'^{m_0} h_1'^{r_0}$;		
$L_0 = j_0^{*m_0} k_0^{*r_0}$;		
$L_1 = j_1'^{m_0} k_1'^{r_0}$;		
$g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1 \Rightarrow$;		
	$u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* \Rightarrow$;	
		Verify $j_1^* = u_1^{*b_1}$ and $k_1^* = v_1^{*b_1}$; If not valid, abort;
$\boxed{r = r_0 + (m - m_0)/d_0;}$		
$m, r \Rightarrow$;		
	$m^*, r^* \Rightarrow$;	
		Verify validity of (m^*, r^*); If not valid, abort;
Output $R(m, m^*)$;		

$\mathbf{Exp}_3$:

$m \leftarrow \mathcal{M}, g_0 \overset{\$}{\leftarrow} G,\ e_1, d_0, d_1 \overset{\$}{\leftarrow} \mathbb{Z}_q$;
$g_1 = g_0^{e_1}, h_0 = g_0^{d_0}, h_1 = g_1^{d_1}, \sigma \overset{\$}{\leftarrow} \{0,1\}^l$;
$u_0 \overset{\$}{\leftarrow} G,\ t_1, s_0, s_1 \overset{\$}{\leftarrow} \mathbb{Z}_q, u_1 = u_0^{t_1}, v_0 = u_0^{s_0}, v_1 = u_1^{s_1}$;
$(\eta, \xi) = S_1(1^k; z)$;
Invoke $\mathcal{A}$ with the left CRS $(g_0, h_0, g_1, h_1, \sigma)$ and the right CRS $(u_0, v_0, u_1, v_1, \eta)$;

$\mathcal{L}$	$\mathcal{A}$	$\mathcal{R}$
		$b_0, b_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;
		$j_0 = u_0^{b_0}, k_0 = v_0^{b_0}$;
		$j_1 = u_1^{b_1}, k_1 = v_1^{b_1}$;
		$z_0, z_1 \overset{\$}{\leftarrow} \{0,1\}^*$;
		$\pi_0 \leftarrow$
		$S_2((u_0, v_0, j_0, k_0), \xi, \eta; z_0)$;
		$\pi_1 \leftarrow$
		$S_2((u_1, v_1, j_1, k_1), \xi, \eta; z_1)$;
		$\Leftarrow j_0, k_0, j_1, k_1,$
π_0, π_1;		
	$\Leftarrow j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$;	
Verify π_0^* and π_1^*;		
If not valid, abort;		
$m_0 \leftarrow \mathcal{M},\ r_0 \overset{\$}{\leftarrow} \mathbb{Z}_q$;		
$a_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*,\ a_2 = d_0 d_1^{-1} a_1$;		
$g_1' = g_1^{a_1}, h_1' = h_1^{a_2}$;		
$j_1' = j_1^{*a_1}, k_1' = k_1^{*a_2}$;		
$M_0 = g_0^{m_0} h_0^{r_0}$;		
$M_1 = g_1'^{m_0} h_1'^{r_0}$;		
$L_0 = j_0^{*m_0} k_0^{*r_0}$;		
$L_1 = j_1'^{m_0} k_1'^{r_0}$;		
$g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1 \Rightarrow$;		
	$u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* \Rightarrow$;	
		Verify $j_1^* = u_1^{*b_1}$ and $k_1^* = v_1^{*b_1}$;
		If not valid, abort;
$r = r_0 + (m - m_0)/d_0$;		
$m, r \Rightarrow$;		
	$m^*, r^* \Rightarrow$;	
		Verify validity of (m^*, r^*);
		If not valid, abort;
Output $R(m, m^*)$;		

$\mathbf{Exp}_4$:

$m \leftarrow \mathcal{M}, g_0 \overset{\$}{\leftarrow} G,\ e_1, d_0, d_1 \overset{\$}{\leftarrow} \mathbb{Z}_q$;
$g_1 = g_0^{e_1}, h_0 = g_0^{d_0}, h_1 = g_1^{d_1}, \sigma \overset{\$}{\leftarrow} \{0,1\}^l$;
$u_0 \overset{\$}{\leftarrow} G,\ t_1, s_0, s_1 \overset{\$}{\leftarrow} \mathbb{Z}_q, u_1 = u_0^{t_1}, v_0 = u_0^{s_0}, v_1 = u_1^{s_1}$;
$(\eta, \xi) = S_1(1^k; z)$;
Invoke $\mathcal{A}$ with the left CRS $(g_0, h_0, g_1, h_1, \sigma)$ and the right CRS $(u_0, v_0, u_1, v_1, \eta)$ and random tape R_A;

$\mathcal{L}$	$\mathcal{A}$	$\mathcal{R}$
		$b_0, b_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;
		$j_0 = u_0^{b_0}, k_0 = v_0^{b_0}$;
		$j_1 = u_1^{b_1}, k_1 = v_1^{b_1}$;
		$z_0, z_1 \overset{\$}{\leftarrow} \{0,1\}^*$;
		$\pi_0 \leftarrow$
		$S_2((u_0, v_0, j_0, k_0), \xi, \eta; z_0)$;
		$\pi_1 \leftarrow$
		$S_2((u_1, v_1, j_1, k_1), \xi, \eta; z_1)$;
		$\Leftarrow j_0, k_0, j_1, k_1,$
π_0, π_1;		
	$\Leftarrow j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$;	
Verify π_0^* and π_1^*;		
$m_0 \leftarrow \mathcal{M},\ r_0 \overset{\$}{\leftarrow} \mathbb{Z}_q$;		
$a_1 \overset{\$}{\leftarrow} \mathbb{Z}_q^*,\ a_2 = d_0 d_1^{-1} a_1$;		
$g_1' = g_1^{a_1}, h_1' = h_1^{a_2}$;		
$j_1' = j_1^{*a_1}, k_1' = k_1^{*a_2}$;		
$M_0 = g_0^{m_0} h_0^{r_0}$;		
$M_1 = g_1'^{m_0} h_1'^{r_0}$;		
$L_0 = j_0^{*m_0} k_0^{*r_0}$;		
$L_1 = j_1'^{m_0} k_1'^{r_0}$;		
$g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1 \Rightarrow$;		
	$u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* \Rightarrow$;	
		Verify $j_1^* = u_1^{*b_1}$ and $k_1^* = v_1^{*b_1}$;
		If not valid, abort;
		$n^* = H^*(u_0, j_0, w)$;
$r = r_0 + (m - m_0)/d_0$;		
$m, r \Rightarrow$;		
	$m^*, r^* \Rightarrow$;	
		Verify validity of (m^*, r^*);
		If not valid, abort;
Output $R(m, n^*)$;		

$\mathbf{Exp}_5$:

$m \leftarrow \mathcal{M}, g_0 \stackrel{\$}{\leftarrow} G,\ e_1, d_0, d_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$;
$g_1 = g_0^{e_1}, h_0 = g_0^{d_0}, h_1 = g_1^{d_1}, \sigma \stackrel{\$}{\leftarrow} \{0,1\}^l$;
$u_0 \stackrel{\$}{\leftarrow} G,\ t_1, s_0, s_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q, u_1 = u_0^{t_1}, v_0 = u_0^{s_0}, v_1 = u_1^{s_1}$;
$(\eta, \xi) = S_1(1^k; z)$;
Invoke $\mathcal{A}$ with the left CRS $(g_0, h_0, g_1, h_1, \sigma)$ and the right CRS $(u_0, v_0, u_1, v_1, \eta)$ and random tape R_A;

$\mathcal{L}$	$\mathcal{A}$	$\mathcal{R}$
		$b_0, b_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$;
		$j_0 = u_0^{b_0}, k_0 = v_0^{b_0}$;
		$j_1 = u_1^{b_1}, k_1 = v_1^{b_1}$;
		$z_0, z_1 \stackrel{\$}{\leftarrow} \{0,1\}^*$;
		$\pi_0 \leftarrow S_2((u_0, v_0, j_0, k_0), \xi, \eta; z_0)$;
		$\pi_1 \leftarrow S_2((u_1, v_1, j_1, k_1), \xi, \eta; z_1)$;
		$\Leftarrow j_0, k_0, j_1, k_1, \pi_0, \pi_1$;
	$\Leftarrow j_0^*, k_0^*, j_1^*, k_1^*, \pi_0^*, \pi_1^*$;	
Verify π_0^* and π_1^*;		
$m_0 \leftarrow \mathcal{M},\ r_0 \stackrel{\$}{\leftarrow} \mathbb{Z}_q$;		
$a_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*,\ a_2 = d_0 d_1^{-1} a_1$;		
$g_1' = g_1^{a_1}, h_1' = h_1^{a_2}$;		
$j_1' = j_1^{*a_1}, k_1' = k_1^{*a_2}$;		
$M_0 = g_0^{m_0} h_0^{r_0}$;		
$M_1 = g_1'^{m_0} h_1'^{r_0}$;		
$L_0 = j_0^{*m_0} k_0^{*r_0}$;		
$L_1 = j_1'^{m_0} k_1'^{r_0}$;		
$g_1', h_1', j_1', k_1', M_0, M_1, L_0, L_1 \Rightarrow$;		
	$u_1^*, v_1^*, j_1^*, k_1^*, M_0^*, M_1^*, L_0^*, L_1^* \Rightarrow$;	
		Verify $j_1^* = u_1^{*b_1}$ and $k_1^* = v_1^{*b_1}$; If not valid, abort; $n^* = H^*(u_0, j_0, w)$;
$\boxed{m_0, r_0 \Rightarrow}$;		
	$m^*, r^* \Rightarrow$;	
		Verify validity of (m^*, r^*); If not valid, abort;
Output $R(m, n^*)$;		

Seiko Arita has been interested in prime numbers, algebraic curves and cryptographic protocols. He is with Institute of Information Security, Kanagawa, Japan. He is a member of JMS.