How to achieve bidirectional zero-knowledge authentication?

No Author Given

No Institute Given

Abstract. Due to the completeness, reliability and zero-knowledge nature, the zero-knowledge proof is widely used to design various protocols, including zero-knowledge authentication protocols. However, the existing zero-knowledge proof scheme cannot realize bidirectional authentication. In this paper, we design a series of bidirectional zero-knowledge protocols based on two new flavors of operations applicable to multiplicative cyclic group. The two notions are formally defined in this paper. We also provide some formal definitions and properties for the two notions. According to our definitions, any bounded polynomial function defined on multiplicative cyclic group has duality and mirror. Based on the two operations, we introduce and formally define dual commitment scheme and mirror commitment scheme. Besides, we provide two efficient constructions for dual commitment and mirror commitment respectively based on CDH assumption and RSA assumption, and named $\mathbf{DC_{CDH}}$, $\mathbf{DC_{RSA}}$, $\mathbf{MC_{CDH}}$ and $\mathbf{MC_{RSA}}$ respectively. We also provide the extended version supporting multiple messages in the appendix. Then, we design some efficient non-interactive as well as interactive zero-knowledge authentication protocols based on these commitments. The protocols allow two participants to achieve mutual zero-knowledge authentication only a communication initialization is needed. Different from other commitment schemes, our schemes can't be used to construct other schemes for cryptography, such as, verifiable secret sharing, zero-knowledge sets, credentials and content extraction signatures, but also can provide technical support for privacy protection of users in distributed scenarios. [1]

Keywords: duality· mirror· dual commitment· mirror commitment· zero-knowledge authentication· non-interactive protocol

A zero-knowledge proof, proposed by Goldwasser, Micali and Rackoff in 1985 [8], has become a fundamental protocol in cryptography. Due to the completeness, reliability and zero-knowledge nature, the zero-knowledge proof is favored by experts and scholars. Then it is widely used for the construction of public key encryption [9], signature [10], identity authentication [11,17–19], secret sharing [23] and other classical cryptography fields as well as blockchain [12, 14–16, 24], privacy computing [13], cloud computing [20], MPC [21] and other popular technology. However, the efficiency, scalability and other problems make zero-knowledge

[1] An extended version of this paper appears in Cryptology ePrint Archive

proof unable to run on resource-constrained equipment. A large number of scholars have carried out in-depth research on this issue and proposed a variety of new zero-knowledge proof implementation schemes [26–32]. These schemes can all make a prover be able to convince a verifier of the validity of some NP statement disclosing more than the fact that the prover knows a witness that satisfies the statement efficiently. Some of schemes even has good performance. But, under the condition of only one initialization, all schemes can only verify the verifier's statement to the prover, and cannot realize role exchange. In this paper, we designed a series of new zero-knowledge authentication protocols based on our newly defined cryptographic primitive: dual commitment and mirror commitment. These protocols can achieve mutual zero-knowledge authentication with only a communication initialization needed. Besides, our schemes also can be widely used for the construction of other schemes, such as verifiable secret sharing, zero-knowledge sets, credentials and content extraction signatures and so on. Our main contributions are as follows.

- We first provide two new notions applicable to multiplicative cyclic group, named duality and mirror.
- We first propose two new cryptographic commitment schemes based on duality and mirror, which we call dual commitment scheme and mirror commitment scheme. Besides, we also provide two efficient constructions for dual commitment and mirror commitment respectively based on CDH assumption and RSA assumption, and named $\mathbf{DC_{CDH}}$, $\mathbf{DC_{RSA}}$, $\mathbf{MC_{CDH}}$ and $\mathbf{MC_{RSA}}$ respectively. Moreover, we give the extended version of these constructions, which supports multiple messages.
- We first design two efficient non-interactive zero-knowledge authentication protocols for these commitments. The protocols allow two participants to submit commitments to each other so that they can achieve mutual zero-knowledge authentication only a communication is needed.

1 Preliminaries

1.1 Notation

We denote by $poly(\lambda)$ any polynomial function that is bounded by a polynomial in λ, where $\lambda \in \mathbb{N}$ is the security parameter. We denote any function that is *negligible* in the security parameter with $negl(\lambda)$ if it vanishes faster than the inverse of any polynomial. We say that an algorithm is *ppt* if and only if it is modeled as a probabilistic turing machine that runs in time polynomial in λ. Given a set S, we denote by $x \leftarrow S$ that x is uniformly sampled from S.

1.2 Commitments

Commitment turned out to be an extremely important primitive in cryptography and has been used as a building block to realize highly non-trivial protocols and primitives. Informally, a commitment scheme is a two-phase protocol between

a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. In committing phase, the prover $\mathcal{P}$ commits to a statement m with a string c using some appropriate algorithm. In the decommitting stage, the prover reveals the opening information op and the message m to the verifier, who can check whether c was indeed a valid commitment on m. A commitment scheme is said to be non-interactive if each phase requires only one message from $\mathcal{P}$ to $\mathcal{V}$. All algorithms have access to a public random string r generated by a trusted setup party.

In their most basic form commitment schemes are expected to meet hiding and binding. A commitment scheme is hiding means with this that it should not reveal information about the committed message to a computationally bounded attacker.

Definition 1 (Hiding). A commitment scheme with commitment algorithm $Commit$ is hiding if there exists a negligible function $negl(\lambda)$ such that for any *ppt* attacker $\mathcal{A}$, for a randomly sampled $r \leftarrow Setup(1^\lambda)$, and for all pairs of messages (m_0, m_1), we have that

$$Pr[\mathcal{A}(r, c) = b | b \leftarrow 0, 1; c \leftarrow Commit(r, m_b)] \leq \frac{1}{2} + negl(\lambda).$$

Definition 2 (Binding). A verification algorithm Verify is binding if there exists a negligible function $negl(\lambda)$ such that for any *ppt* attacker $\mathcal{A}$ and for a randomly sampled $r \leftarrow Setup(1^\lambda)$, we have that

$$Pr[Verify(r, c, op, m) = 1 \wedge Verify(r, c, op', m') = 1 \wedge m \neq m' | \\ (c, op, m, op', m') \leftarrow \mathcal{A}(r)] \leq negl(\lambda).$$

1.3 Computational Assumptions

Here we formally describe the computational hardness assumptions that we need for the security of our construction.

Definition 3 (Discrete Logarithm Assumption). DLA Let $\mathcal{G}$ be a multiplicative cyclic group of order p proportional to the security parameter λ and let g be a generator of $\mathcal{G}$. We say that the discrete logarithm problem is hard if, for a random integer $x \in \mathbb{Z}_p$ and for all *ppt* attackers $\mathcal{A}$, there exists a negligible function $negl(\lambda)$ such that

$$Pr[\mathcal{A}(\mathcal{G}, g, g^x) = x] \leq negl(\lambda).$$

Definition 4 (Computational Diffie-Hellman Assumption, CDH). Let $\mathcal{G}$ be a multiplicative cyclic group of order p proportional to the security parameter λ and let g be a generator of $\mathcal{G}$. We say that the computational Diffie-Hellman problem is hard if, for two random integers $x, y \in \mathbb{Z}_p$ and for all *ppt* attackers $\mathcal{A}$, there exists a negligible function $negl(\lambda)$ such that

$$Pr[\mathcal{A}(\mathcal{G}, g, g^x, g^y) = g^{xy}] \leq negl(\lambda).$$

Definition 5 (RSA Assumption, RSA). Let $\lambda \in \mathbb{N}$ be the security parameter, N is a random RSA modulus of length, z be a random element in $\mathbb{Z}_N$

and e be an $(\ell+1)$-bit prime (for a parameter ℓ). Then we say that the RSA assumption holds if for any *ppt* attackers $\mathcal{A}$, the probability

$$Pr[\mathcal{A}(N, y, y^e) = z] \leq negl(\lambda).$$

Definition 6 (Square Computational Diffie-Hellman Assumption, CDH) Let $\mathcal{G}$ be a multiplicative cyclic group of order p proportional to the security parameter λ and let g be a generator of $\mathcal{G}$ and $a \xleftarrow{\$} \mathbb{Z}_p$. We say that the Square Computational Diffie-Hellman Assumption holds in $\mathbb{G}$ if for every *ppt* attackers $\mathcal{A}$, the probability

$$Pr[A(g, g^a) = g^{a^2}] \leq negl(\lambda)$$

In [2,3] is shown that the Square-CDH assumption is equivalent to the classical Computational Diffie-Hellman (CDH) assumption.

1.4 Duality and Mirror Function on multiplicative cyclic group

Here we extend the notion of dual and mirror in logical algebra and provide a formal definition of duality and mirror applicable to multiplicative cyclic group.
Definition 6 (Dual on multiplicative cyclic group). Let $\mathcal{F}$ be a polynomial function defined on multiplicative cyclic group $\mathbb{G}$, where g is the generator of $\mathbb{G}$. Another polynomial function $\mathcal{F}^*$ defined on $\mathbb{G}$ is said to be the duality of function $\mathcal{F}$ if it may be obtained from $\mathcal{F}$ by replacing the corresponding operation symbols with the following replacement rules and has the same operation order as $\mathcal{F}$, recording as $\mathcal{F} \triangleright \mathcal{F}^*$.

- Replace $+, \times$ with $\times, +$.
- Replace $-, /$ with $/, -$, where $/$ represents the inverse operation defined on multiplicative cyclic group.
- Replace $1, 0$ with $0, 1$.

To facilitate readers to better understand the definition, we give three extended definitions and three examples to explain these definitions.
Definition 7 (unidirectional Dual) If $\mathcal{F}^*$ is the duality of $\mathcal{F}$ while $\mathcal{A}^*$ is not the duality of $\mathcal{F}$. We say $\mathcal{F}$ and $\mathcal{F}^*$ are unidirectional dual, recording as $\mathcal{F} \triangleright \mathcal{F}^*$.
Definition 8 (Bidirectional Dual) If $\mathcal{F}$ is the duality of $\mathcal{F}^*$ while $\mathcal{F}^*$ is also the duality of $\mathcal{F}$. We say $\mathcal{F}$ and $\mathcal{F}^*$ are bidirectional dual, recording as $\mathcal{F} \triangleleft\triangleright \mathcal{F}^*$.
Definition 9 (Self Dual) If $\mathcal{F}$ is the duality of $\mathcal{F}$. We say $\mathcal{F}$ is self-dual, recording as $\overset{\triangledown}{\mathcal{F}}$.
Example 1 $\mathcal{F}^* = x * g - y * h$ is the duality of $\mathcal{F} = g^x / h^y$, where g,h,x,$y \in \mathbb{G}$. However, $\mathcal{F}^*$ is not the duality of $\mathcal{F}$. Then, $\mathcal{F} \triangleright \mathcal{F}^*$.
Example 2 $\mathcal{F}^* = (x+g) * (y-h) \triangleleft\triangleright \mathcal{F} = xg + y/h$ are Bidirectional dual.
Example 3 $\mathcal{F}^* = z$ is self dual, where $z \in \mathbb{G}$.
Definition 10 (Mirror on multiplicative cyclic group). Let $\mathcal{F} = \sum_{i=0}^{n} a_i x_i^{b_i}$ be a polynomial function defined on multiplicative cyclic group $\mathbb{G}$, where g is the generator of $\mathbb{G}$ and $\forall i \in \mathcal{Z}_p, a_i, x_i, b_i \in \mathbb{G}$. Another polynomial function $\mathcal{F}^\star$

defined on $\mathbb{G}$ is said to be the mirror of function $\mathcal{F}$ if it equals to $\sum_{i=0}^{n} a_{n-i} x_i^{b_{n-i}}$ and has the same operation order as $\mathcal{F}$, recording as $\mathcal{F} \Leftrightarrow \mathcal{F}^{\star}$. To facilitate readers to better understand the definition, we give a extended definition, a example and a theorem based on definition 10.

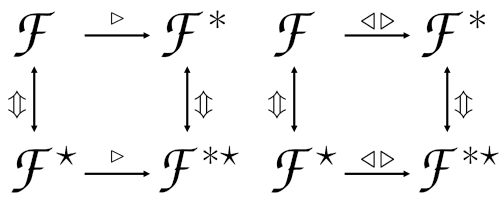
Proposition 1. If $\mathcal{F}^{\star}$ is the mirror of $\mathcal{F}$, then $\mathcal{F}$ must also be the mirror of $\mathcal{F}^{\star}$. It can be easily proved according to definition 10.

Definition 11 (Self Mirror) If $\mathcal{F}$ is the mirror of $\mathcal{F}$. We say $\mathcal{F}$ is self-mirror, recording as $\overset{\star}{\mathcal{F}}$.

Example 4 If $\mathcal{F} = \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} a_i x_i^{b_i} + a_i x_{n-i}^{b_i}$, then, $\mathcal{F}$ must be self-mirror.

Proposition 2. If $\mathcal{F}$ is a $poly(\lambda)$ defined on multiplicative cyclic group $\mathbb{G}$, where g is the generator of $\mathbb{G}$, $\mathcal{F}^*$ is the duality of $\mathcal{F}$ and $(F^*)^{\star}$ is the mirror of $\mathcal{F}^*$, $\mathcal{F}^{\star}$ is the mirror of $\mathcal{F}$ and $(\mathcal{F}^{\star})^*$ is the duality of $\mathcal{F}^{\star}$ then $(\mathcal{F}^{\star})^* = (\mathcal{F}^*)^{\star}$, recording as $\mathcal{F}^{**}$. We show the diagram for $\mathcal{F}$, $\mathcal{F}^*$, $\mathcal{F}^{\star}$ and $\mathcal{F}^{**}$ in Fig.1.

Example 5 If $\mathcal{F} = \sum_{i=0}^{n} a_i x_i^{b_i}$, then, we can get that $\mathcal{F}^{\star} = \sum_{i=0}^{n} a_{n-i} x_i^{b_{n-i}}$, $\mathcal{F}^* = \sum_{i=0}^{n} (a_i + b_i x_i)$. Then we can compute that $(\mathcal{F}^{\star})^* = \sum_{i=0}^{n} (a_{n-i} + b_{n-i} x_i)$ and $(\mathcal{F}^*)^{\star} = \sum_{i=0}^{n} (a_{n-i} + b_{n-i} x_i)$. Obviously, $(\mathcal{F}^*)^{\star} = (\mathcal{F}^{\star})^* = \mathcal{F}^{**}$.



(a) the dual is unidirectional dual

(b) the dual is Bidirectional dual

Fig. 1. Relation for $\mathcal{F}$, $\mathcal{F}^*$, $\mathcal{F}^{\star}$ and $\mathcal{F}^{**}$

2 Dual Commitment

In this section, on the basis of the definition of duality in section 2.4, we provide a formal definition of a dual commitment scheme, followed by two constructions. In the first construction, the commitment, designed based on CDH Assumption, is a unidirectional dual commitment. While in the second construction, the commitment, designed based on RSA Assumption, is a bidirectional dual commitment. We also prove the security properties and discuss some useful features of our constructions.

2.1 Definition

A dual commitment consists of seven *ppt* algorithms: **Setup**, **Commit**, **Open**, $\mathbf{Verify}^{\mathbf{part}}$, $\mathbf{Verify}^{\mathbf{full}}$, $\mathbf{Update}^{\mathbf{message}}$ and $\mathbf{Update}^{\mathbf{proof}}$.

- $(c, pp) \leftarrow \mathbf{Setup}(1^\lambda)$ Given the security parameter λ, the setup algorithm $Setup$ outputs a public random string c and some public parameters pp (which implicitly define the message space $\mathcal{M}_{pp}$, randomizer space R_{pp} and commitment space C_{pp}.)
- $(c, c^*, aux) \leftarrow \mathbf{Commit}(r, m, pp)$ Given the public random string r, a message m and public parameters pp, the commitment algorithm $Commit$ outputs a commitment c, a dual commitment c^* and corresponding auxiliary information aux.
- $op \leftarrow \mathbf{Open}(m, aux, pp)$ This algorithm is run by the committer to produce a proof op that m is the committed message and pp is the public parameters. In particular, notice that in the case when some updates have occurred the auxiliary information aux can include the update information produced by these updates.
- $b \leftarrow \mathbf{Verify}^{\mathbf{part}}(c, m, c|c^*, pp, op)$ Given the public random string c, a message m, a commitment c and opening information op, the partial verification algorithm $Verify^{part}$ outputs 1 if op is a valid opening for commitment c or dual commitment c^* on message m.
- $(b, t) \leftarrow \mathbf{Verify}^{\mathbf{full}}(c, m, c, c^*, pp, op)$ Given the public random string c, a message m, a commitment c, a commitment c^*, opening information op, the full verification algorithm $Verify^{full}$ outputs b=1 if op is a valid opening for commitment c and dual commitment c^* on message m. $Verify^{full}$ outputs t=2 if b=1 and $c \triangleleft\triangleright c^*$ are Bidirectional dual, outputs t=1 if b=1 and $c \triangleright c^*$, outputs t=- 1 if other conditions occur.
- $(c', c^{*'}, U) \leftarrow \mathbf{Update}^{\mathbf{message}}(c, c^*, m, m')$ This algorithm is run by the committer to update the dual commitment by changing the message m to m'. The algorithm takes as input the old message m, the new message m', the commitment c and the dual commitment c^* of message m. It outputs a new commitment c' and a new dual commitment $c^{*'}$ together with an update information U.
- $(op') \leftarrow \mathbf{Update}^{\mathbf{proof}}(c, c^*, U, op)$ This algorithm can be run by any user who holds a proof op for message m, and it allows the user to compute an updated proof op' (and the updated commitment c' and $c^{*'}$) such that op' will be valid. Basically, the value U contains the updated information.

For correctness, we require that $\forall\lambda \in \mathbb{N}$, for all honestly generated parameters pp, an honest committer should be able to correctly generate a commitment, a dual commitment and a proof op for all message $m \in \mathcal{M}$. Then, a honest verifier can correctly verify the correctness of a proof, a commitment and a dual commitment and the relevance of the commitment and the dual commitment for all message $m \in \mathcal{M}$.

For security, we require that a malicious committer should not be able to convincingly present two different messages m and m' with respect to c and c^*. we formally define the security and correctness of a dual commitment scheme.

Definition 12. We say (**Setup**, **Commit**, **Open**, $\mathbf{Verify}^{\mathbf{part}}$, $\mathbf{Verify}^{\mathbf{full}}$, $\mathbf{Update}^{\mathbf{message}}$ and $\mathbf{Update}^{\mathbf{proof}}$) is a secure dual commitment scheme if it satisfies the following properties.

Correctness. Let $(r, pp) \leftarrow \mathbf{Setup}(1^\lambda)$ and $(c, c^*, aux) \leftarrow \mathbf{Commit}(r, m, pp)$. For a commitment c and a dual commitment c^* output by $\mathbf{Commit}(r, m, pp)$, and all $m \in \mathcal{M}$, the output of $\mathbf{Open}(m, aux, pp)$ can be successfully verified by $\mathbf{Verify}^{\mathbf{part}}(r, m, c|c^*, pp, op)$ and $\mathbf{Verify}^{\mathbf{full}}(r, m, c, c^*, pp, op)$.
Binding. For all adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, where $\mathcal{A}_0$ is *ppt* (and $\mathcal{A}_1$ is not computationally bounded), and for a randomly sampled $(r, pp) \leftarrow \mathbf{Setup}(1^\lambda)$, we have that:

$$\begin{aligned} Pr[Verify^{part}(r, m, c|c^*, pp, op) = 1 \wedge Verify^{part} \\ (r, m', c|c^*, pp, op') = 1 \wedge m \neq m'|(c|c^*, op, m) \leftarrow \\ \mathcal{A}_0(r); (m', op') \leftarrow \mathcal{A}_1(r, state)] \leq negl(\lambda). \end{aligned}$$

Besides,

$$\begin{aligned} Pr[Verify^{full}(r, m, c, c^*, pp, op) = 1 \wedge Verify^{full} \\ (r, m', c, c^*, pp, op') = 1 \wedge m \neq m'|(c, c^*, op, m) \leftarrow \\ \mathcal{A}_0(r); (m', op') \leftarrow \mathcal{A}_1(r, state)] \leq negl(\lambda). \end{aligned}$$

Hiding.for any *ppt* attacker $\mathcal{A}$, for a randomly sampled $(r, pp) \leftarrow \mathbf{Setup}(1^\lambda)$, and for all pairs of messages (m_0, m_1), we have that

$$Pr[\mathcal{A}(r, c, c^*) = b|b \leftarrow 0, 1; (c, c^*, aux) \leftarrow Commit(r, m, pp)] \leq \frac{1}{2} + negl(\lambda).$$

2.2 A unidirectional Dual Commitment based on CDH: $\mathbf{DC_{CDH}}$

Here we propose an implementation of concise unidirectional dual commitment $\mathbf{DC_{CDH}}$ for single message based on the CDH assumption of multiplicative cyclic group of order p proportional to the security parameter λ, where g is the generator . Precisely , the security of the scheme reduces to the Square Computational Diffie-Hellman assumption (see Definition 6 in Section 2.1), which has been shown equivalent of the standard CDH assumption [2, 3](see Definition 4 in Section 2.1).
$\mathbf{Setup}(1^\lambda)$ Let $\mathbb{G}$ be a multiplicative cyclic group of order p proportional to the security parameter λ and let g be a generator of $\mathbb{G}$. Randomly choose $z_c, z_1, z_2 \leftarrow \mathbb{Z}_p$. Set $r = g^{z_c}$, $h_1 = g^{z_1}$, $h_2 = g^{z_2}$. Set $pp = (g, h_1, h_2)$. The message space is $\mathcal{M} = \mathbb{Z}_p$.
$\mathbf{Commit}(r, m, pp)$ Compute

$$c = h_1^m h_2^r,\ c^* = m * h_1 + r * h_2$$

and output $C = (c, c^*, aux)$ and the auxiliary information $aux = none$.
$\mathbf{Open}(m, r, pp)$ Compute

$$op_c = h_1^m,\ op_{c^*} = m * h_1$$

and output $op = (op_c, op_{c^*})$.

$\mathbf{Verify}^{\mathbf{part}}(r, m, c|c^*, pp, op_c|op_{c^*})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } b = 1 \text{ and } c = op_c * h_2^r \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } c^* = op_{c^*} + r * h_2 \\ 0 & \text{otherwise} \end{cases}$$

and output $b_1 \vee b_2$.
$\mathbf{Verify}^{\mathbf{full}}(r, m, c, c^*, pp, op_c, op_{c^*})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } b = 1 \text{ and } c = op_c * h_2^r \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } c^* = op_{c^*} + r * h_2 \\ 0 & \text{otherwise} \end{cases}$$

$$b = b_1 \wedge b_2, \quad t = \begin{cases} 1, & \text{if } b = 1 \text{ and } c \triangleleft\triangleright c^* \\ 0, & \text{if } b = 1 \text{ and } c \triangleright c^* \\ -1 & \text{otherwise} \end{cases}$$

and output (b,t).
$\mathbf{Update}^{\mathbf{message}}(c, c^*, m, m')$ Compute the updated commitment $c' = c * h_1^{m'-m}$ and dual commitment $c^{*'} = c^* + h_2(m' - m)$. Finally output $C' = (c', c^{*'})$ and $U = (m, m')$.
$\mathbf{Update}^{\mathbf{proof}}(c, c^*, U, op)$ A client who owns a proof op, that is valid to c and c^* for the message m, can produce a new proof $op' = (op_c * h_1^{m'-m}, op_{c^*} * h_1(m' - m)$.

The correctness of the scheme can be easily verified by inspection. We prove its security via the following theorem.

Theorem 1. If the CDH assumption holds, then the scheme defined above is a concise dual commitment. We prove the theorem by showing that the scheme satisfies the binding property. The proof of the theorem is showed in the extended version of this paper.

2.3 A Double Dual Commitment based on RSA: $\mathbf{DC_{RSA}}$

Here we propose a realization of double dual commitment $\mathbf{DC_{RSA}}$ for a single message from the RSA assumption (whose definition is given in section 2.1). Appendix A shows the double dual commitment scheme supporting multiple messages.

$\mathbf{Setup}(1^\lambda, \ell)$ Randomly choose two $\ell/2$-bit primes p_1, p_2, set $N = p_1 p_2$, and then choose $2(\ell + 1)$-bit primes e_1, e_2, a, r that do not divide $\varphi(N)$. Compute,

$$S_1 = a^{e_2}, S_2 = a^{e_1}$$

The public parameters pp are $(N, a, r, S_1, S_2, e_1, e_2)$. The message space is $M = \{0, 1\}^\ell$.

$\mathbf{Commit}(r, m, pp)$ Compute

$$c = S_1^m S_2^r = a^{e_2 m + e_1 r}, \; c^* = a^{(e_2+m)(e_1+r)}$$

and output $C = (c, c^*, aux)$ and the auxiliary information $aux = none$.

Open(m, r, pp) Compute

$$op_c = S_1^{\frac{m}{e_2}},\ op_{c^*} = S_1^{\frac{r}{e_1}} S_2^{\frac{m}{e_2}}$$

and output $op = (op_c, op_{c^*})$. Notice that knowledge of pp allows to compute op_c efficiently without the factorization of N.
Verify$^{\mathbf{part}}(r, m, c|c^*, pp, op_c|op_{c^*})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } S_1 S_2 op_{c^*}^{e_1} a^{m*r} \ mod\ N = op_{c^*} \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } S_2^r op_c^{e_2} \ mod\ N = op_c \\ 0 & \text{otherwise} \end{cases}$$

and output $b = b_1 \vee b_2$.
Verify$^{\mathbf{full}}(r, m, c, c^*, pp, op_c, op_{c^*})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } S_1 S_2 op_{c^*}^{e_1} a^{m*r} \ mod\ N = c^* \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } (S_2^r op_c^{e_2}) \ mod\ N = c \\ 0 & \text{otherwise} \end{cases}$$

$$b = b_1 \wedge b_2, \quad t = \begin{cases} 1, & \text{if } b = 1 \text{ and } c \lhd \rhd c^* \\ 0, & \text{if } b = 1 \text{ and } c \rhd c^* \\ -1 & \text{otherwise} \end{cases}$$

and output (b,t).
Update$^{\mathbf{message}}(c, c^*, m, m')$ Compute the updated commitment $c' = c * S_1^{m'-m}$ and dual commitment $c^{*'} = c^* * a^{(e_1+r)(m'-m)}$. Finally output $C' = (c', c^{*'})$ and $U = (m, m')$.
Update$^{\mathbf{proof}}(c, c^*, U, op)$ A client who owns a proof op, that is valid to c and c^* for the message m, can produce a new proof $op' = (op_c * S_1^{\frac{m-m'}{e_2}}, op_{c^*} * S_2^{\frac{m-m'}{e_1}})$.

In order for the verification process to be correct, notice that one should also check that the S_1, S_2 are correctly generated with respect to a and the exponents e_1, e_2. The correctness of the scheme can be easily verified by inspection. We prove its security via the following theorem.
Theorem 2. If the RSA assumption holds, then the scheme defined above is a concise dual commitment. We prove the theorem by showing that the scheme satisfies the binding property. The proof of the theorem is showed in the extended version of this paper.

In this section, On the basis of the definition of mirror in section 2.4, we provide a formal definition of a mirror commitment scheme, followed by two constructions. In the first construction, the commitment was designed based on CDH Assumption. While in the second construction, the commitment was designed based on RSA Assumption. We also prove the security properties and discuss some useful features of our constructions.

2.4 Definition

A mirror commitment consists of seven *ppt* algorithms: **Setup**, **Commit**, **Open**, **Verify**$^{\mathbf{part}}$, **Verify**$^{\mathbf{full}}$, **Update**$^{\mathbf{message}}$ and **Update**$^{\mathbf{proof}}$.

- $(r, pp) \leftarrow \mathbf{Setup}(1^\lambda)$ Given the security parameter λ, the setup algorithm $Setup$ outputs a public random string r and some public parameters pp (which implicitly define the message space $\mathcal{M}_{pp}$, randomizer space R_{pp} and commitment space C_{pp}.)
- $(c, c^\star, aux) \leftarrow \mathbf{Commit}(r, m, pp)$ Given the public random string r, a message m and public parameters pp, the commitment algorithm $Commit$ outputs a commitment c, a dual commitment $c^\star$ and corresponding auxiliary information aux.
- $op \leftarrow \mathbf{Open}(m, aux, pp)$ This algorithm is run by the committer to produce a proof op that m is the committed message and pp is the public parameters. In particular, notice that in the case when some updates have occurred the auxiliary information aux can include the update information produced by these updates.
- $b \leftarrow \mathbf{Verify}^{\mathbf{part}}(r, m, c|c^\star, pp, op)$ Given the public random string r, a message m, a commitment c and opening information op, the partial verification algorithm $Verify^{part}$ outputs 1 if op is a valid opening for commitment c or dual commitment $c^\star$ on message m.
- $(b, t) \leftarrow \mathbf{Verify}^{\mathbf{full}}(r, m, c, c^\star, pp, op)$ Given the public random string r, a message m, a commitment c, a commitment $c^\star$, opening information op, the full verification algorithm $Verify^{full}$ outputs b=1 if op is a valid opening for commitment c and dual commitment $c^\star$ on message m. $Verify^{full}$ outputs t=1 if b=1 and $c \Leftrightarrow c^\star$, and outputs t=0 if other conditions occur.
- $(c', c^{\star'}, U) \leftarrow \mathbf{Update}^{\mathbf{message}}(c, c^\star, m, m')$ This algorithm is run by the committer to update the dual commitment by changing the message m to m'. The algorithm takes as input the old message m, the new message m', the commitment c and the dual commitment $c^\star$ of message m. It outputs a new commitment c' and a new dual commitment $c^{\star'}$ together with an updated information U.
- $(op') \leftarrow \mathbf{Update}^{\mathbf{proof}}(c, c^\star, U, op)$ This algorithm can be run by any user who holds a proof op for message m, and it allows the user to compute an updated proof op' (and the updated commitment c' and $c^{\star'}$) such that op' will be valid. Basically, the value U contains the update information.

For correctness, we require that $\forall \lambda \in \mathbb{N}$, for all honestly generated parameters pp, an honest committer should be able to correctly generate a commitment, a mirror commitment and a proof op for all message $m \in \mathcal{M}$. Then, an honest verifier can correctly verify the correctness of a proof, a commitment and a mirror commitment and the relevance of the commitment and the mirror commitment for all messages $m \in \mathcal{M}$.

For security, we require that a malicious committer should not be able to convincingly present two different messages m and m' with respect to c and $c^\star$. we formally define the security and correctness of a mirror commitment scheme.

Definition 13. We say (**Setup**, **Commit**, **Open**, $\mathbf{Verify}^{\mathbf{part}}$, $\mathbf{Verify}^{\mathbf{full}}$, $\mathbf{Update}^{\mathbf{message}}$ and $\mathbf{Update}^{\mathbf{proof}}$) is a secure dual commitment scheme if it satisfies the following properties.

Correctness. Let $(r, pp) \leftarrow \mathbf{Setup}(1^\lambda)$ and $(c, c^\star, aux) \leftarrow \mathbf{Commit}(r, m, pp)$. For a commitment c and a mirror commitment c^* output by $\mathbf{Commit}(r, m, pp)$,

and all $m \in \mathcal{M}$, the output of $\mathbf{Open}(m, aux, pp)$ can be successfully verified by $\mathbf{Verify^{part}}(r, m, c|c^\star, pp, op)$ and $\mathbf{Verify^{full}}(r, m, c, c^\star, pp, op)$.
Binding. For all adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, where $\mathcal{A}_0$ is *ppt* (and $\mathcal{A}_1$ is not computationally bounded), and for a randomly sampled $(r, pp) \leftarrow \mathbf{Setup}(1^\lambda)$, we have that:

$$\begin{aligned} Pr[Verify^{part}(r, m, c|c^\star, pp, op) = 1 \wedge Verify^{part} \\ (r, m', c|c^\star, pp, op') = 1 \wedge m \neq m'|(c|c^\star, op, m) \leftarrow \\ \mathcal{A}_0(r); (m', op') \leftarrow \mathcal{A}_1(r, state)] \leq negl(\lambda). \end{aligned}$$

Besides,

$$\begin{aligned} Pr[Verify^{full}(r, m, c, c^\star, pp, op) = 1 \wedge Verify^{full} \\ (r, m', c, c^\star, pp, op') = 1 \wedge m \neq m'|(c, c^\star, op, m) \leftarrow \\ \mathcal{A}_0(r); (m', op') \leftarrow \mathcal{A}_1(r, state)] \leq negl(\lambda). \end{aligned}$$

Hiding. For all *ppt* adversaries $\mathcal{A}$, for a randomly sampled $(r, pp) \leftarrow \mathbf{Setup}(1^\lambda)$, and for all pairs of messages (m_0, m_1), we have that

$$Pr[\mathcal{A}(r, c, c^\star) = b|b \leftarrow 0, 1; (c, c^\star, aux) \leftarrow Commit(r, m, pp)] \leq \frac{1}{2} + negl(\lambda).$$

2.5 A Mirror Commitment based on CDH: $\mathbf{MC_{CDH}}$

Here we propose an implementation of concise mirror commitment $\mathbf{MC_{CDH}}$ for a single message based on the CDH assumption in multiplicative cyclic group G of order p proportional to the security parameter λ, where g is the generator. Precisely, the security of the scheme reduces to the Square Computational Diffie-Hellman assumption(see Definition 6 in Section 2.1), which has been shown equivalent to the standard CDH assumption [2, 3](see Definition 4 in Section 2.1). Appendix B shows the mirror commitment scheme supporting multiple messages.
$\mathbf{Setup}(1^\lambda)$ Let $\mathbb{G}$ be a multiplicative cyclic group of order p proportional to the security parameter λ and let g be a generator of $\mathbb{G}$. Randomly choose $z_c, z_1, z_2 \leftarrow \mathbb{Z}_p$. Set $r = g^{z_c}$, $h_1 = g^{z_1}$, $h_2 = g^{z_2}$. Set $pp = (g, h_1, h_2)$. The message space is $\mathcal{M} = \mathbb{Z}_p$.
$\mathbf{Commit}(r, m, pp)$ Compute

$$c = h_1^m h_2^r,\ c^\star = h_1^r h_2^m$$

and output $C = (c, c^\star, aux)$ and the auxiliary information $aux = none$.
$\mathbf{Open}(m, r, pp)$ Compute

$$op_c = h_1^m,\ op_{c^\star} = h_2^m$$

and output $op = (op_c, op_{c^*})$.

$\mathbf{Verify}^{\mathbf{part}}(r, m, c|c^\star, pp, op_c|op_{c^\star})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } c = op_c * h_2^r \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } c = op_{c^\star} * h_1^r \\ 0 & \text{otherwise} \end{cases}$$

and output $b = b_1 \vee b_2$.
$\mathbf{Verify}^{\mathbf{full}}(r, m, c, c^\star, pp, op_c, op_{c^\star})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } c = op_c * h_2^r \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } c = op_{c^\star} * h_1^r \\ 0 & \text{otherwise} \end{cases}$$

$$b = b_1 \wedge b_2, \quad t = \begin{cases} 1, & \text{if } b = 1 \text{ and } c \Leftrightarrow c^\star \\ 0 & \text{otherwise} \end{cases}$$

and output (b,t).
$\mathbf{Update}^{\mathbf{message}}(c, c^\star, m, m')$ Compute the updated commitment $c' = c * h_1^{m'-m}$ and dual commitment $c^{\star'} = c^\star * h_2^{m'-m}$. Finally output $C' = (c', c^{\star'})$ and $U = (m, m')$.
$\mathbf{Update}^{\mathbf{proof}}(c, c^\star, U, op)$ A client who owns a proof op, that is valid to c and $c^\star$ for the message m, can produce a new proof $op' = (op_c * h_1^{m'-m}, op_{c^\star} * h_2^{m'-m})$.

The correctness of the scheme can be easily verified by inspection. We prove its security via the following theorem.

Theorem 3. If the CDH assumption holds, then the scheme defined above is a concise dual commitment. We prove the theorem by showing that the scheme satisfies the binding property. The proof of the theorem is showed in the extended version of this paper.

2.6 A Mirror Commitment based on RSA: $\mathbf{MC_{RSA}}$

Here we propose an implication of mirror commitment $\mathbf{MC_{RSA}}$ for a single message from the RSA assumption (whose definition is given in section 2.1). Appendix C shows the dual commitment scheme supporting multiple messages.
$\mathbf{Setup}(1^\lambda, \ell)$ Randomly choose two $\ell/2$-bit primes p_1, p_2, set $N = p_1 p_2$, and then choose $2(\ell+1)$-bit primes e_1, e_2, a, r that do not divide $\varphi(N)$. Compute,

$$S_1 = a^{e_2}, \ S_2 = a^{e_1}$$

The public parameters pp are $(N, a, r, S_1, S_2, e_1, e_2)$. The message space is $M = \{0,1\}^\ell$.
$\mathbf{Commit}(r, m, pp)$ Compute

$$c = S_1^m S_2^r = a^{e_2 m + e_1 r}, \ c^\star = S_1^r S_2^m = a^{e_1 m + e_2 r}$$

and output $C = (c, c^\star, aux)$ and the auxiliary information $aux = none$.
$\mathbf{Open}(m, r, pp)$ Compute

$$op_c = S_1^{\frac{m}{e_2}} \ mod \ N, \ op_{c^\star} = S_2^{\frac{m}{e_1}} \ mod \ N$$

and output $op = (op_c, op_{c^\star})$. Notice that knowledge of pp allows to compute op_c efficiently without the factorization of N.

$\mathbf{Verify^{part}}(r, m, c|c^\star, pp, op_c|op_{c^\star})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } S_2^r op_c^{e_2} \ mod \ N = c \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } S_1^r op_{c^\star}^{e_1} \ mod \ N = c^\star \\ 0, & \text{otherwise} \end{cases}$$

and output $b = b_1 \vee b_2$.

$\mathbf{Verify^{full}}(r, m, c, c^*, pp, op_c, op_{c^*})$ Compute

$$b_1 = \begin{cases} 1, & \text{if } S_2^r op_c^{e_2} \ mod \ N = c \\ 0 & \text{otherwise} \end{cases}, \quad b_2 = \begin{cases} 1, & \text{if } S_1^r op_{c^\star}^{e_1} \ mod \ N = c^\star \\ 0, & \text{otherwise} \end{cases}$$

$$b = b_1 \wedge b_2, \quad t = \begin{cases} 1, & \text{if } b = 1 \text{ and } c \Leftrightarrow c^* \\ 0, & \text{otherwise} \end{cases}$$

and output (b,t).

$\mathbf{Update^{message}}(c, c^\star, m, m')$ Compute the updated commitment $c' = c * S_1^{m'-m}$ and dual commitment $c^{\star'} = c^\star * S_2^{m'-m}$. Finally output $C' = (c', c^{\star'})$ and $U = (m, m')$.

$\mathbf{Update^{proof}}(c, c^\star, U, op)$ A client who owns a proof op, that is valid to c and $c^\star$ for the message m, can produce a new proof $op' = (op_\star * S_1^{\frac{m-m'}{e_2}}, op_{c^\star} * S_2^{\frac{m-m'}{e_1}})$.

In order for the verification process to be correct, notice that one should also check that the S_1, S_2 are correctly generated with respect to a and the exponents e_1, e_2. The correctness of the scheme can be easily verified by inspection. We prove its security via the following theorem.

Theorem 4. If the RSA assumption holds, then the scheme defined above is a concise mirror commitment. We prove the theorem by showing that the scheme satisfies the binding property. The proof of the theorem is showed in the extended version of this paper.

3 Features on Mirror Commitment and Dual Commitment

We next discuss some important features of Mirror Commitment and Dual Commitment.

Theorem 5. Homomorphism. $\mathbf{DC_{CDH}}$ and $\mathbf{MC_{CDH}}$ are both (additive) homomorphic in nature. The proof of the theorem is showed in the extended version of this paper.

Theorem 6. Standard Security Properties $\mathbf{DC_{RSA}}$ and $\mathbf{MC_{RSA}}$ are computationally hiding under the RSA assumption. $\mathbf{DC_{CDH}}$ and $\mathbf{MC_{CDH}}$ are statistically binding.The proof of the theorem is showed in the extended version of this paper.

Theorem 7. Trapdoor Commitment. $\mathbf{DC_{CDH}}, \mathbf{MC_{CDH}}, \mathbf{DC_{RSA}}, \mathbf{MC_{RSA}}$ are also trapdoor commitment schemes, where $r = g^{z_c}$ is the trapdoor. For DC_{CDH}, given crs, a simulator can create witnesses for arbitrary values with respect to $C = h_1^m h_2^{crs}$ for an unknown crs. To "prove" m (where m is message supposedly committed to by C), output op. It can easily be checked that $\mathbf{Verify^{part}}(C, m, op, pp) = 1$ and $\mathbf{Verify^{full}}(C, m, op, pp) = 1$. The same also applies to MC_{CDH}, DC_{RSA}, MC_{RSA}.

4 Bidirectional Zero-knowledge Authentication Protocols

In this section, we describe applications of our commitment schemes to construct bidirectional non-interactive zero-knowledge authentication protocols. A Prover($\mathcal{P}$) can convince a Verifier ($\mathcal{V}$) that it is legal user by proving c^* is the duality of c or $c^\star$ is the mirror of c without revealing any privacy information and they can still continue to authenticate without another initialization even after changing roles. Here, we give the instance of constructing non-interactive and interactive bidirectional zero-knowledge authentication protocols through $\mathbf{DC_{RSA}}$ and $\mathbf{MC_{RSA}}$. However, $\mathbf{DC_{CDH}}$ is unidirectional dual commitment so that it can't be used to build bidirectional authentication protocol, but, it can be used to build unidirectional authentication protocol. The rest 2 instances of constructing zero-knowledge authentication protocol through $\mathbf{DC_{CDH}}$, $\mathbf{MC_{CDH}}$ are similar to $\mathbf{DC_{RSA}}$ and $\mathbf{MC_{RSA}}$. They are showed in the extended version of this paper.

zero-knowledge authentication for $\mathbf{DC_{RSA}}$

Let p_1, p_2 are two $\ell/2$-bit primes , set $N = p_1 p_2$. Let e_1, e_2, a, r are three $2(\ell+1)$-bit primes that do not divide $\varphi(N)$. Let $S_1 = a^{e_2}, S_2 = a^{e_1}, m \in \{0,1\}^\ell$. The protocol presented in algorithm 1 as a sigma protocol for the relation $\mathbb{R}_1$.

Algorithm 1 Interactive protocol for $\mathbf{DC_{RSA}}$
$\mathcal{R}_1 = c, c^* \in \mathbb{G}; m_1, m_2 \in \mathbb{Z}_l : c = a^{e_2 m + e_1 r}, c^* = a^{(e_2+m)(e_1+r)}$

$\mathbb{P}$: $r_0, r_1 \xleftarrow{\$} \mathbb{Z}_l$ *and* *computes* :
$R_0 = a^{r_0(e_1+e_2)}, R_1 = a^{r_1(e_1-e_2)}$
$\mathbb{P} \to \mathbb{V}$: R_0, R_1
$\mathbb{V} \leftarrow \mathbb{P}$: $t \xleftarrow{\$} \mathbb{Z}_l$
$\mathbb{P}$: computes:
$y_0 = a^{r_0 + t(m+r)}, y_1 = a^{r_1 + t(r-m)}$
$\mathbb{P} \to \mathbb{V}$: y_0, y_1
$\mathbb{V}$: returns *Accept* if and only if the following hold:
$y_0^{(e_1+e_2)} * y_1^{(e_1-e_2)} / c^{2t} \stackrel{?}{=} R_0 * R_1$ (1)

This protocol may be made non-interactive via the Fiat-Shamir [33] technique, where the verifier challenge is replaced by a suitable transcript hash. This technique further allows for binding an arbitrary proof context into the transcript. Algorithm 2 shows an example non-interactive protocol.

Algorithm 2 Non-interactive protocol for $\mathbf{DC_{RSA}}$
$\mathcal{R}_1 = c, c^* \in \mathbb{G}; m_1, m_2 \in \mathbb{Z}_l : c = a^{e_2 m + e_1 r}, c^* = a^{(e_2+m)(e_1+r)}$

$\mathbb{P}$: $r_0, r_1 \xleftarrow{\$} \mathbb{Z}_l$ *and* *computes* :
$R_0 = a^{r_0(e_1+e_2)}, R_1 = a^{r_1(e_1-e_2)}$
$t = H_s(R_0 * R_1, c, c^*)$ *where* $H_s(*)$ *is a hash function*
$y_0 = a^{r_0+t(m+r)}, y_1 = a^{r_1+t(r-m)}$
$\mathbb{P} \to \mathbb{V}$: t, y_0, y_1
$\mathbb{V}$: returns *Accept* if and only if the following hold:
$H_s(y_0^{(e_1+e_2)} * y_1^{(e_1-e_2)}/c^{2t}, c, c^*) \stackrel{?}{=} t$

Theorem 8 Both non-interactive and interactive zero-knowledge authentication protocols for $\mathbf{DC_{RSA}}$ can be used to realize bidirectional authentication with only an initialization. The proof of the theorem is showed in the extended version of this paper.

zero-knowledge authentication for $\mathbf{MC_{RSA}}$

Let p_1, p_2 are two $\ell/2$-bit primes , set $N = p_1 p_2$. Let e_1, e_2, a, r are three $2(\ell+1)$-bit primes that do not divide $\varphi(N)$. Let $S_1 = a^{e_2}, S_2 = a^{e_1}, m \in \{0,1\}^\ell$. The protocol presented in algorithm 3 as a sigma protocol for the relation $\mathbb{R}_2$.

Algorithm 3 Interactive protocol for $\mathbf{MC_{RSA}}$
$\mathcal{R}_1 = c, c^\star \in \mathbb{G}; m_1, m_2 \in \mathbb{Z}_l : c = a^{e_2 m + e_1 r}, c^\star = a^{e_2 r e_1 m}$

$\mathbb{P}$: $r_0, r_1 \xleftarrow{\$} \mathbb{Z}_l$ and computes:
$R_0 = a^{r_0(e_1+e_2)}, R_1 = a^{r_1(e_1-e_2)}$
$\mathbb{P} \to \mathbb{V}$: R_0, R_1
$\mathbb{V} \leftarrow \mathbb{P}$: $t \xleftarrow{\$} \mathbb{Z}_l$
$\mathbb{P}$: computes:
$y_0 = a^{r_0+t(m+r)}, y_1 = a^{r_1+t(m-r)}$
$\mathbb{P} \to \mathbb{V}$: y_0, y_1
$\mathbb{V}$: returns *Accept* if and only if the following hold:
$y_0^{(e_1+e_2)}/(c * c^\star)^t \stackrel{?}{=} R_0$ (4), $y_1^{(e_1-e_2)}/(c/c^\star)^t \stackrel{?}{=} R_1$ (5)

This protocol may be made non-interactive via the Fiat-Shamir [33] technique, where the verifier challenge is replaced by a suitable transcript hash. This technique further allows for binding an arbitrary proof context into the transcript. Algorithm 4 shows an example non-interactive protocol.

Theorem 9 Both non-interactive and interactive zero-knowledge authentication protocols for $\mathbf{MC_{RSA}}$ can be used to realize bidirectional authentication with only an initialization. The proof of the theorem is showed in the extended version of this paper.

Algorithm 4 Non-interactive protocol for $\mathbf{MC_{RSA}}$
$\mathcal{R}_1 = c, c^\star \in \mathbb{G}; m_1, m_2 \in \mathbb{Z}_l : c = a^{e_2 m + e_1 r}, c^\star = a^{(e_2 r)(e_1 m)}$

$\mathbb{P}$: $r_0, r_1 \xleftarrow{\$} \mathbb{Z}_l$ and computes:
$R_0 = a^{r_0(e_1+e_2)}, R_1 = a^{r_1(e_1-e_2)}$
$t = H_s(R_0, R_1, c, c^\star)$ *where* $H_s(*)$ *is a hash function*
$y_0 = a^{r_0+t(m+r)}, y_1 = a^{r_1+t(m-r)}$
$\mathbb{P} \to \mathbb{V}$: t, y_0, y_1
$\mathbb{V}$: returns *Accept* if and only if the following hold:
$H_s(y_0^{(e_1+e_2)}/(c * c^\star)^t, y_1^{(e_1-e_2)}/(c/c^\star)^t, c, c^*) \stackrel{?}{=} t$

5 Open Problems

Finally, we list a few open problems related to the commitment and mirror commitment schemes. 1. Is it possible to construct efficient polynomial commitment schemes under weaker assumptions? 2. What other protocols do dual commitment and mirror commitment improves? (For example, can commitment and mirror commitment reduce communication of asynchronous VSS protocols or verifiable shuffles? See the protocol of Groth and Ishai [5]) 3. We have mainly focused on communication costs, but our construction asks for nontrivial computation. Is it possible to reduce computation costs as well? 4. Whether the dual commitment and mirror commitment can be used to construct the oblivious transfer protocols?

References

1. Torben P. Pedersen. 1991. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'91). Springer-Verlag, Berlin, Heidelberg, 129–140.
2. Feng Bao, Robert Deng, and HuaFei Zhu. Variations of diffie-hellman problem. In Sihan Qing, Dieter Gollmann, and Jianying Zhou, editors, Information and Communications Security, volume 2836 of Lecture Notes in Computer Science, pages 301–312. Springer Berlin / Heidelberg, 2003.
3. Ueli M. Maurer and Stefan Wolf. Diffie-Hellman oracles. In Neal Koblitz, editor, CRYPTO'96, volume 1109 of LNCS, pages 268–282. Springer, August 1996.
4. Adi Shamir. On the generation of cryptographically strong pseudorandom sequences. ACM Trans. Comput. Syst., 1(1):38–44, 1983.
5. J. Groth and Y. Ishai. Sub-linear zero-knowledge argument for correctness of a shuffle. In Proceedings of EUROCRYPT'08, volume 4965 of LNCS, pages 379–396. Springer, 2008.
6. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO'84. Springer
7. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: CRYPTO'91. Springer
8. Goldwasser, S., S. Micali, and C. Rackoff, The Knowledge Complexity of Interactive Proof Systems, in 17th Annual Symposium on Theory Of Computing (STOC), 1985.

9. SAHAI A. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security, In: Pro- ceedings of 40th Annual Symposium on Foundations of Computer Science (FOCS 1999). IEEE, 1999: 543–553.
10. Arun, A., Bonneau, J., Clark, J. (2022). Short-lived Zero-Knowledge Proofs and Signatures. In: Agrawal, S., Lin, D. (eds) Advances in Cryptology – ASIACRYPT 2022. ASIACRYPT 2022. Lecture Notes in Computer Science, vol 13793. Springer, Cham.
11. Du, R., Li, X., Liu, Y. (2022). A Cross-domain Authentication Scheme Based on Zero-Knowledge Proof. In: Lai, Y., Wang, T., Jiang, M., Xu, G., Liang, W., Castiglione, A. (eds) Algorithms and Architectures for Parallel Processing. ICA3PP 2021. Lecture Notes in Computer Science(), vol 13156. Springer, Cham.
12. MA S L, DENG Y, HE D B, et al. An efficient NIZK scheme for privacy-preserving transactions over account- model blockchain. IEEE Transactions on Dependable Secure Computing, 2021, 18(2): 641–65
13. CHASE M, GANESH C, MOHASSEL P. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials. In: Advances in Cryptology-CRYPTO 2016, Part III. Springer Berlin Heidelberg, 2016: 49
14. BEN-SASSON E, CHIESA A, GARMAN C, et al. Zerocash: Decentralized anonymous payments from bitcoin. In: Proceedings of 2014 IEEE Symposium on Security and Privacy (S&P 2014). IEEE, 2014: 459–4
15. DANEZIS G, FOURNET C, KOHLWEISS M, et al. Pinocchio coin: Building zerocoin from a succinct pairing- based proof system. In: Proceedings of the First ACM Workshop on Language Support for Privacy-enhancing Technologies (PETShop 2013). ACM, 2013: 27
16. GABIZON A, WILLIAMSON Z J, CIOBOTARU O. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. IACR Cryptology ePrint Archive, 20
17. ZHANG Z F, YANG K, HU X X, et al. Practical anonymous password authentication and TLS with anonymous client authentication[C]. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communica- tions Security (CCS 2016). ACM, 2016: 11791
18. LI Z P, WANG D, MORAIS E. Quantum-safe round-optimal password authentication for mobile devices[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(3): 1885-1899.
19. WANG Q X, WANG D, CHENG C, et al. Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices[J]. IEEE Transactions on Dependable and Secure Computing, 2021: 1–1.
20. Kumar, S.B., Mandal, R.K., Mukherjee, K., Dwivedi, R.K. (2022). Security of Cloud Computing Using Quantum Zero-Knowledge Proof System. In: Dahal, K., Giri, D., Neogy, S., Dutta, S., Kumar, S. (eds) Internet of Things and Its Applications. Lecture Notes in Electrical Engineering, vol 825. Springer, Singapore.
21. Feneuil, T., Maire, J., Rivain, M., Vergnaud, D. (2022). Zero-Knowledge Protocols for the Subset Sum Problem from MPC-in-the-Head with Rejection. In: Agrawal, S., Lin, D. (eds) Advances in Cryptology – ASIACRYPT 2022. ASIACRYPT 2022. Lecture Notes in Computer Science, vol 13792. Springer, Cham.
22. Hazay, C., Venkitasubramaniam, M. & Weiss, M. ZK-PCPs from Leakage-Resilient Secret Sharing. J Cryptol 35, 23 (2022).
23. Huang, J., Huang, T., Zhang, J. (2023). zkChain: An Efficient Blockchain Privacy Protection Scheme Based on zero-knowledge-SNARKs. In: Xu, Y., Yan, H., Teng,

H., Cai, J., Li, J. (eds) Machine Learning for Cyber Security. ML4CS 2022. Lecture Notes in Computer Science, vol 13656. Springer, Cham.
24. XIE T C, ZHANG J H, ZHANG Y P, et al. Libra: Succinct zero-knowledge proofs with optimal prover computation. In: Advances in Cryptology-CRYPTO 2019, Part III. Springer Cham, 2019: 733–764.
25. SETTY S. Spartan: Efficient and general-purpose zkSNARKs without trusted setup. In: Advances in Cryptology-CRYPTO 2020, Part III. Springer Cham, 2020: 704–737.
26. ZHANG J H, XIE T C, ZHANG Y P, et al. Transparent polynomial delegation and its applications to zero knowledge proof[C]. In: Proceedings of 2020 IEEE Symposium on Security and Privacy (S&P 2020). IEEE, 2020: 859–876.
27. ZHANG J H, LIU T Y, WANG W, et al. Doubly efficient interactive proofs for general arithmetic circuits with linear prover time[C]. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS 2021). ACM, 2021: 159–177
28. BOOTLE J, CERULLI A, CHAIDOS P, et al. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Advances in Cryptology-EUROCRYPT 2016, Part II. Springer Berlin Heidelberg, 2016: 327–35
29. WAHBY R S, TZIALLA I, SHELAT A, et al. Doubly-efficient zkSNARKs without trusted setup. In: Proceedings of 2018 IEEE Symposium on Security and Privacy (S&P 2018). IEEE, 2018: 926–943.
30. GIACOMELLI I, MADSEN J, ORLANDI C. ZKBoo: Faster zero-knowledge for Boolean circuits. In: Pro- ceedings of the 25th USENIX Conference on Security Symposium (SEC'16). USENIX, 2016: 1069-1083.
31. GVILI Y, SCHEFFLER S, VARIA M. BooLigero: Improved sublinear zero knowledge proofs for Boolean cir- cuits. In: Financial Cryptography and Data Security-FC 2021, Part I. Springer Berlin Heidelberg, 2021: 476–496.
32. Cui, H., Zhang, K. (2021). A Simple Post-Quantum Non-interactive Zero-Knowledge Proof from Garbled Circuits. In: Yu, Y., Yung, M. (eds) Information Security and Cryptology. Inscrypt 2021. Lecture Notes in Computer Science(), vol 13007. Springer, Cham.
33. Fiat A, Shamir A.How to prove yourself:Practical solutions to identifcation and signature problems(CI/Advances in Cryptology-CRYPTO"86.Berlin:Springer,1987:186-194