Group to Group Commitments Do Not Shrink

Masayuki Abe, Kristiyan Haralambiev, and Miyako Ohkubo

NTT Information Sharing Platform Laboratories
abe.masayuki@lab.ntt.co.jp
New York University
kkh@cs.nyu.edu
Security Architecture Laboratory, NSRI, NICT
m.ohkubo@nict.go.jp

Abstract. We investigate commitment schemes whose messages, keys, commitments, and decommitments are elements of bilinear groups, and whose openings are verified by pairing product equations. Such commitments facilitate efficient zero-knowledge proofs of knowledge of a correct opening. We show two lower bounds on such schemes: a commitment cannot be shorter than the message and verifying the opening in a symmetric bilinear group setting requires evaluating at least two independent pairing product equations. We also present optimal constructions that match the lower bounds in symmetric and asymmetric bilinear group settings.

Keywords: Structure-Preserving Commitments, Homomorphic Trapdoor Commitments.

1 Introduction

Efficient cryptographic protocols are often hand-crafted and their underlying idea is hardly visible. On the other hand, modular design offers conceptual simplicity in exchange of losing efficiency. Structure-preserving cryptography [1] is a concept that facilitates modular yet reasonably efficient construction of cryptographic protocols. It provides inter-operable cryptographic building blocks whose input/output data consist only of group elements and their computations preserve the group structure. Combined with the Groth-Sahai (GS) proof system [18], such structure-preserving schemes allow proofs of knowledge about privacy-sensitive data present in their inputs and outputs. Commitments [9,1], various signatures [1,10,2], and adaptive chosen-ciphertext secure public-key encryption [8] have been presented in the context of structure-preserving cryptography. They yield a number of applications including various privacy-protecting signatures [1], efficient zero-knowledge arguments [17], and efficient leakage-resilient signatures [13].

We revisit structure preserving commitment schemes. Their keys, messages, commitments, and decommitments are elements of bilinear groups, and the opening is verified by evaluating pairing product equations. Using a bilinear map $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, messages from the base group are either committed to target

D. Pointcheval and T. Johansson (Eds.): EUROCRYPT 2012, LNCS 7237, pp. 301–317, 2012.

group elements and the commitments are shrinking, or committed to group elements from the same group but commitments are larger than the messages. In other words, there are two types of commitment functions: either "$\mathbb{G} \to \mathbb{G}_T$ and shrinking" or "$\mathbb{G} \to \mathbb{G}$ and expanding". The former type, [1,16], takes multiple elements in the base group $\mathbb{G}$ as input and shrinks them into a constant number of elements in the target group $\mathbb{G}_T$ by exploiting the one-way nature of the mapping from $\mathbb{G}$ to $\mathbb{G}_T$. Involving elements in $\mathbb{G}_T$ in a commitment is acceptable as long as witness-indistinguishability is sufficient for the accompanying GS proofs, but it is problematic if zero-knowledge is necessary. The latter type, [9,3], which we call *strictly* structure-preserving schemes, takes messages in $\mathbb{G}$ and also yields commitments in $\mathbb{G}$. Unfortunately, due to the absence of a one-way structure in the mapping from $\mathbb{G}$ to $\mathbb{G}$, their construction is more involved. Moreover, they are expanding: commitments are 2-3 times larger than messages in the known constructions. Nothing is known about the lower bound, and constructing more efficient commitment schemes of the latter type has been an open problem.

Our Results. This paper presents two lower bounds on strictly structure-preserving commitment schemes. First, we show that for a message of size k the commitment must be at least size k; thus, negatively answering to the above-stated open problem. This lower bound highlights the gap from the known upper bound of $2k$ in [3]. The lower bound is obtained by assuming that key generation and commitment functions are algebraic. By algebraic algorithms we mean any computation conditioned so that, when outputting a group element, the algorithm "knows" its representation with respect to given bases. The class covers a wide range of algorithms including all constructions in the standard model to the best our knowledge. See Section 2.5 for more detailed discussion.

Next, we show that strictly structure-preserving commitment schemes for symmetric bilinear groups require at least two pairing product equations in the verification. The number of equations, as well as the size of commitments, is an important factor in determining efficiency since the size of a zero-knowledge proof of a correct opening grows linearly with the number of verification equations. A scheme described in [3] achieves this bound but verifies k elements from a commitment in one equation and other k elements in the other equation, which requires $2k$ elements for a commitment. Thus it does not match to the first lower bound. Because the lower bounds of a commitment size and the number of equations are independent, we see that a scheme that achieves both bounds is missing.

We close the gap by presenting two optimal constructions (except for small additive constants). The first construction works over asymmetric bilinear groups, yields commitments of size $k + 1$, and verifies with one equation. The second construction works over symmetric bilinear groups, yields commitments of $k+2$, and verifies with two equations. Both constructions implement trapdoor and homomorphic properties. The schemes are computationally binding based on simple standard computational assumptions. Finally, we assess their efficiency in combination with GS zero-knowledge proofs of correct opening.

2 Preliminaries

2.1 Bilinear Groups

Let $\mathcal{G}$ be a bilinear group generator that takes security parameter 1^λ and outputs a description of bilinear groups $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$ where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order p, e is an efficient and non-degenerating bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, and G and $\tilde{G}$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. By Λ^*, we denote Λ without the generators G and $\tilde{G}$, i.e., $\Lambda^* = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$.

By Λ_{sym} we denote a special case of Λ where $\mathbb{G}_1 = \mathbb{G}_2$ (and $G = \tilde{G}$), which is also referred to as a symmetric setting. Λ_{sxdh} denotes a case where the decision Diffie-Hellman (DDH) assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$. This means that no efficient mapping is available for either direction. Λ_{sxdh} is usually referred to as the symmetric external DDH (SXDH) setting [22,6,15,23]. For practical differences between Λ_{sym} and Λ_{sxdh}, please refer to [14].

2.2 Notations

By $\mathbb{G}$, we denote a base group, $\mathbb{G}_1$ or $\mathbb{G}_2$, when the difference is not important. By $\mathbb{G}^*$ we denote $\mathbb{G} \setminus \{1_\mathbb{G}\}$. We use upper case letters to group elements and corresponding lower case letters to represent the discrete-log of the group element with respect to a fixed (but not necessarily explicit) base. For a set or a vector of group elements, $\boldsymbol{X} \in \mathbb{G}^n$, the size of $\boldsymbol{X}$ refers to n and is denoted as $|\boldsymbol{X}|$. We consider $\boldsymbol{X}$ as a row vector. For a vector or an ordered set $\boldsymbol{X}$, the i-th element is denoted as $\boldsymbol{X}[i]$ or X_i.

We use multiplicative notations for group operations and additive notation for vector operations. The transpose of $\boldsymbol{X}$ is denoted as $\boldsymbol{X}^t$. A concatenation of vectors $\boldsymbol{X} \in \mathbb{G}^n$ and $\boldsymbol{Y} \in \mathbb{G}^k$ is denoted as $\boldsymbol{X}||\boldsymbol{Y} \stackrel{\text{def}}{=} (X_1, \ldots, X_n, Y_1, \ldots, Y_k)$. For $\boldsymbol{X} \in \mathbb{G}^n$ and $\boldsymbol{a} \in \mathbb{Z}_p^n$, we define $\boldsymbol{a}\,\boldsymbol{X}^t \stackrel{\text{def}}{=} \prod_{i=1}^n X_i^{a_i}$. For a matrix $A \in \mathbb{Z}_p^k \times \mathbb{Z}_p^n$ and $\boldsymbol{X} \in \mathbb{G}^n$, $A\,\boldsymbol{X}^t \stackrel{\text{def}}{=} (\prod_{i=1}^n X_i^{a_{1,i}}, \cdots, \prod_{i=1}^n X_i^{a_{k,i}})^t$, where $a_{i,j}$ is entry (i,j) of A. For $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{G}^n$, $\boldsymbol{X} + \boldsymbol{Y} \stackrel{\text{def}}{=} (X_1 \cdot Y_1, \ldots, X_n \cdot Y_n)$. For $\boldsymbol{X} \in \mathbb{G}_1^n$ and $\boldsymbol{Y} \in \mathbb{G}_2^n$, $\boldsymbol{X} \cdot \boldsymbol{Y}^t$ is defined as $\prod_{i=1}^n e(X_i, Y_i)$. By $\mathbf{0} \in \mathbb{G}^n$ we denote additive unity vector $\mathbf{0} = \{1_\mathbb{G}, \ldots, 1_\mathbb{G}\}$.

For $a_{ij} \in \mathbb{Z}_p, T \in \mathbb{G}_T$, $X_i \in \mathbb{G}_1$, and $Y_j \in \mathbb{G}_2$, an equation of the form

$$\prod_i \prod_j e(X_i, Y_j)^{a_{ij}} = T$$

is called a pairing product equation (PPE). With our notation, any pairing product equation for variables $\boldsymbol{X} \in \mathbb{G}_1^k$ and $\boldsymbol{Y} \in \mathbb{G}_2^n$ can be represented as $\boldsymbol{X}\,A\,\boldsymbol{Y}^t = T$ where A is a $k \times n$ matrix over $\mathbb{Z}_p$ and T is a constant in $\mathbb{G}_T$. For convenience, we may abuse these notations for vectors that consist of elements from both $\mathbb{G}_1$ and $\mathbb{G}_2$ assuming that relevant entries of a multiplied scaler matrix are zero so that the computation is well defined in either $\mathbb{G}_1$ or $\mathbb{G}_2$.

For a sequence of events, $E_1, \ldots, E_n$ and a statement S, $\Pr[E_1, \ldots, E_n : S]$ denotes the probability that S is satisfied when events $E_1, \ldots, E_n$ occur. The probability is taken over the random coins used in the events.

2.3 Commitment Schemes

We focus on non-interactive commitment schemes and follow a standard syntactical definition with the following setup.

Definition 1 (Commitment Scheme). *A commitment scheme C is a quadruple of efficient algorithms $\mathsf{C} = (\mathsf{Setup}, \mathsf{Key}, \mathsf{Com}, \mathsf{Vrf})$ in which;*

- $gk \leftarrow \mathsf{Setup}(1^\lambda)$ *is a common parameter generator that takes security parameter λ and outputs a set of common parameters, gk.*
- $ck \leftarrow \mathsf{Key}(gk)$ *is a key generator that takes gk as input and outputs commitment-key ck. It may take extra parameters as input if needed. It is assumed that ck determines the message space $\mathcal{M}_{ck}$. A messages is valid if it is in $\mathcal{M}_{ck}$.*
- $(com, open) \leftarrow \mathsf{Com}(ck, msg)$ *is a commitment function that takes ck and message, msg, and outputs commitment, com, and opening information, $open$.*
- $1/0 \leftarrow \mathsf{Vrf}(ck, com, msg, open)$ *is a verification function that takes ck, com, msg, and $open$ as input, and outputs 1 or 0 representing acceptance or rejection, respectively.*

It is required that $\Pr[gk \leftarrow \mathsf{Setup}(1^\lambda),\ ck \leftarrow \mathsf{Key}(gk),\ msg \leftarrow \mathcal{M}_{ck},\ (com, open) \leftarrow \mathsf{Com}(ck, msg) : 1 \leftarrow \mathsf{Vrf}(ck, com, msg, open)] = 1$.

Definition 2 (Binding and Hiding Properties). *A commitment scheme is binding if, for any polynomial-time adversary $\mathcal{A}$,* $\Pr[gk \leftarrow \mathsf{Setup}(1^\lambda), ck \leftarrow \mathsf{Key}(gk), (com, msg, open, msg', open') \leftarrow \mathcal{A}(ck) : 1 \leftarrow \mathsf{Vrf}(ck, com, msg, open) \wedge 1 \leftarrow \mathsf{Vrf}(ck, com, msg', open')]$ *is negligible. It is hiding if, for any polynomial-time adversary $\mathcal{A}$, advantage* $\Pr[1 \leftarrow \mathrm{Hide}^{\mathsf{TC}}_{\mathcal{A}}(1)] - \Pr[1 \leftarrow \mathrm{Hide}^{\mathsf{TC}}_{\mathcal{A}}(0)]$ *is negligible in λ where $b' \leftarrow \mathrm{Hide}^{\mathsf{TC}}_{\mathcal{A}}(b)$ is the process that $gk \leftarrow \mathsf{Setup}(1^\lambda)$, $ck \leftarrow \mathsf{Key}(gk)$, $(msg_0, msg_1, \omega) \leftarrow \mathcal{A}(ck)$, $(com, -) \leftarrow \mathsf{Com}(ck, msg_b)$, $b' \leftarrow \mathcal{A}(\omega, com)$.*

Definition 3 (Trapdoor Commitment Scheme). *A commitment scheme is called a trapdoor commitment scheme if Key additionally outputs a trapdoor-key tk, and there is an efficient algorithm Equiv called equivocation algorithm that takes $(ck, tk, com, msg, open, msg')$ as input and outputs $open'$ such that, for legitimately generated ck, tk, and any valid messages msg and msg', it holds that $(com, open) \leftarrow \mathsf{Com}(ck, msg)$, $open' \leftarrow \mathsf{Equiv}(ck, tk, com, msg, open, msg')$, $1 \leftarrow \mathsf{Vrf}(ck, com, msg', open')$, and two distributions $(ck, com, msg, open)$ and $(ck, com, msg', open')$ over all choices of msg and msg' are indistinguishable.*

Definition 3 is usually referred to as chameleon hash [20], and, in fact, is a stronger requirement than the common definition of a trapdoor commitment scheme (e.g., see [16]), which allows a different algorithm (taking tk as an input) to compute equivocalable commitments.

Definition 4 (Homomorphic Commitment Scheme). *A commitment scheme is homomorphic if, for any legitimately generated ck, three binary operations, $\cdot$, $\odot$, $\otimes$, are defined, and for any valid messages, msg and msg', it holds that $(com, open) \leftarrow \mathsf{Com}(ck, msg)$, $(com, open) \leftarrow \mathsf{Com}(ck, msg)$, $1 \leftarrow \mathsf{Vrf}(ck, com \cdot com', msg \odot msg', open \otimes open')$ with probability 1.*

2.4 Strictly Structure-Preserving Commitments

Definition 5 (Strictly Structure-Preserving Commitments). *A commitment scheme* C *is strictly structure-preserving with respect to a bilinear group generator* $\mathcal{G}$ *if*

- $\mathsf{Setup}(1^\lambda)$ *outputs* gk *that consists of* $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$ *generated by* $\mathcal{G}(1^\lambda)$*,*
- Key *outputs* ck *that consists of* Λ^* *and group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*
- *the messages consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*
- Com *outputs* com *and* $open$ *that consist of elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, and*
- Vrf *evaluates membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and evaluating pairing product equations over* Λ^**.*

Function Setup may also determine non-group elements, such as constants in $\mathbb{Z}_p$, which are given implicitly to other functions as system parameters. Note that the size of a message, denoted by k, may be limited by the size of ck. Also note that, in a previous work [1], com is allowed to include elements in $\mathbb{G}_T$ while it is limited to $\mathbb{G}$ in the above *strict* case. This results in limiting the pairing product equations in Vrf to have $T = 1_{\mathbb{G}_T}$ since none of ck, com, msg, $open$ could include elements from $\mathbb{G}_T$. Our lower bounds, however, hold even if ck and $open$ include $T \neq 1$ used for verification.

2.5 Algebraic Algorithms

Roughly, an algorithm $\mathcal{A}$ is algebraic over Λ if, whenever $\mathcal{A}$ is given elements $(X_1, \ldots, X_n)$ of a group and outputs an element Y in the same group, $\mathcal{A}$ should "know" a representation $(r_1, \ldots, r_n)$ of Y that fulfils $Y = \prod X_i^{r_i}$. We require the property only with respect to the base groups. A formal definition follows.

Definition 6 (Algebraic Algorithm). *Let* $\mathcal{A}$ *be a probabilistic polynomial time algorithm that takes a bilinear group description* Λ*, a string* $aux \in \{0,1\}^*$*, and base group elements* $\boldsymbol{X} \in \mathbb{G}^k$ *for some* k *as input; and outputs a group element in* $\mathbb{G}$ *and a string* $ext \in \{0,1\}^*$*. Algorithm* $\mathcal{A}$ *is called algebraic with respect to* $\mathcal{G}$ *if there exists a probabilistic polynomial-time algorithm,* Ext*, receiving the same input as* $\mathcal{A}$ *including the same random coins such that for any* $\Lambda \leftarrow \mathcal{G}(1^\lambda)$*, all polynomial size* $\boldsymbol{X} \neq (1, \ldots, 1)$*, and* aux*, the following probability, taken over coin* r*, is negligible in* λ*.*

$$\Pr\left[\begin{array}{l}(Y_1, \ldots, Y_n, ext) \leftarrow \mathcal{A}(\Lambda, \boldsymbol{X}, aux\,;\, r),\\ (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n,\ ext) \leftarrow \mathsf{Ext}(\Lambda, \boldsymbol{X}, aux\,;\, r)\end{array} \;:\; \exists i \in \{1, \ldots, n\}\ s.t.\ Y_i \neq \prod_{j=1}^{k} X_j^{y_{i,j}}\right]$$

The notion is often used for restricting a class of reduction algorithms for showing impossibility of security proofs for practical cryptographic schemes by black-box reduction, e.g., [7,11]. The notion in this case implies the limitation of current reduction techniques and considered as "not overly restrictive" as it covers all known efficient reductions.

The notion is also used for characterising constructions of cryptographic schemes. In [2], the signing function is assumed computable only with generic operations, which implies that it is algebraic. A closely related concept is known as the knowledge of exponent assumption [12,19,5]. It is applied to adversary $\mathcal{A}$ and considered as a "very strong assumption" since it is hardly falsifiable. It is also generally undesirable to put a limitation on the ability of a malicious party.

Similar to [2], but with slightly more generality, we put a restriction on the key generation and commitment algorithms so that they are algebraic. Though this narrows the coverage of our result, it still covers quite a wide range of approaches. It also suggests a direction to find a new construction that includes non-algebraic operations yet the relation can be efficiently verified by generic operations through pairing product equations.

2.6 Assumptions

Assumption 7 (Double Pairing Assumption (DBP)). Given Λ and $(G_z, G_r) \leftarrow \mathbb{G}_1^{*2}$, it is hard to find $(Z, R) \in \mathbb{G}_2^* \times \mathbb{G}_2^*$ that satisfies

$$1 = e(G_z, Z)\, e(G_r, R). \tag{1}$$

Assumption 8 (Simultaneous Double Pairing Assumption (SDP)). Given Λ and $(G_z, G_r, F_z, F_s) \leftarrow \mathbb{G}_1^{*4}$, it is hard to find $(Z, R, S) \in \mathbb{G}_2^{*3}$ that satisfies

$$1 = e(G_z, Z)\, e(G_r, R) \qquad \text{and} \qquad 1 = e(F_z, Z)\, e(F_s, S). \tag{2}$$

DBP is implied by DDH in $\mathbb{G}_1$. It does not hold for Λ_{sym}. SDP is implied by DLIN [9] for Λ_{sym}. When Λ_{sxdh} is considered, we can assume the dual version of these assumptions that swap $\mathbb{G}_1$ and $\mathbb{G}_2$.

3 Lower Bounds

We show two lower bounds for strictly structure-preserving commitment scheme C over $\mathcal{G}$. Let $\Lambda \leftarrow \mathcal{G}(1^\lambda)$, $ck := (\Lambda^*, \boldsymbol{V})$, $msg := \boldsymbol{M}$, $com := \boldsymbol{C}$, $open := \boldsymbol{D}$, where $\boldsymbol{V}$, $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{D}$ are vectors of elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ in Λ. Let ℓ_v, ℓ_m, and ℓ_c denote the size of $\boldsymbol{V}$, $\boldsymbol{M}$, and $\boldsymbol{C}$, respectively.

3.1 Commitment Size

Theorem 9. *If the discrete-logarithm problem in the base groups of Λ is hard, Key and Com are algebraic, and $\ell_c < \ell_m$, then C is not binding.*

Proof. Algorithm Com takes $(\Lambda^*, \boldsymbol{V}, \boldsymbol{M})$ as input and outputs $(\boldsymbol{C}, \boldsymbol{D})$ under the constraint that $\ell_c < \ell_m$. Since Com is algebraic, there exists an associated algorithm $\mathsf{Ext}_{\mathsf{Com}}$ that takes the same input as Com does and outputs matrices B_1, B_2, B_3, B_4 over $\mathbb{Z}_p$ for which

$$(\boldsymbol{C})^t = B_1\, (\boldsymbol{M})^t + B_2\, (\boldsymbol{V})^t \quad \text{and} \quad (\boldsymbol{D})^t = B_3\, (\boldsymbol{M})^t + B_4\, (\boldsymbol{V})^t \tag{3}$$

hold. Note that B_1 is an $\ell_c \times \ell_m$ rectangular matrix. We first consider the symmetric bilinear setting where $\mathbb{G}_1 = \mathbb{G}_2$ and represent the group by $\mathbb{G}$. We later argue that the same argument holds for asymmetric setting with trivial modifications.

We construct an adversary $\mathcal{A}$ that breaks the binding property of C. First $\mathcal{A}$ selects arbitrary $\boldsymbol{M}$ and computes $(\boldsymbol{C}, \boldsymbol{D}) \leftarrow \mathsf{Com}(\Lambda^*, \boldsymbol{V}, \boldsymbol{M})$. It then runs $\mathsf{Ext}_{\mathsf{Com}}(\Lambda^*, \boldsymbol{V}, \boldsymbol{M})$ and obtains $B_1, \ldots, B_4$. If an i-th column of B_1 is zero, then $\boldsymbol{M'}$ is formed by replacing M_i in $\boldsymbol{M}$ with a fresh arbitrary M'_i. If none of the columns of B_1 are zero, $\mathcal{A}$ finds a non-zero vector $\boldsymbol{R}$ that satisfies $B_1\,(\boldsymbol{R})^t = \boldsymbol{0}$. Then it computes $\boldsymbol{M'} = \boldsymbol{M} + \boldsymbol{R}$. In either case, $\mathcal{A}$ then computes $(\boldsymbol{D'})^t := B_3\,(\boldsymbol{M'})^t + B_4\,(\boldsymbol{V})^t$, and outputs $(\boldsymbol{C}, \boldsymbol{M}, \boldsymbol{D}, \boldsymbol{M'}, \boldsymbol{D'})$. This completes the description of $\mathcal{A}$.

We first show that the above $\boldsymbol{R}$ can be efficiently found. By applying standard Gaussian elimination to B_1, one can efficiently find S_1 that is the largest regular sub-matrix of B_1. Let I and J be the set of indexes of rows and columns of B_1, respectively, that form S_1. By $\bar{I}$ and $\bar{J}$, we denote the rest of the indexes in B_1. Note that $|I| = |J|$ and $|J| + |\bar{J}| = \ell_m$. Consider matrix S_2 of size $|I| \times |\bar{J}|$ formed by selecting entries $B_1[i][j]$, $i \in I$, and $j \in \bar{J}$. Such S_2 can be formed since $\bar{J}$ is not empty due to $\ell_c < \ell_m$. Select arbitrary non-zero vector $\boldsymbol{R}_2$ of size $|\bar{J}|$ and compute $(\boldsymbol{R}_1)^t = -S_1^{-1}\,S_2\,(\boldsymbol{R}_2)^t$. Then $\boldsymbol{R}_1$ is a vector of size $|J|$. Then compose $\boldsymbol{R}$ from $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ in such a way that $\boldsymbol{R}[J[i]] := \boldsymbol{R}_1[i]$ and $\boldsymbol{R}[\bar{J}[i]] := \boldsymbol{R}_2[i]$. Since $\boldsymbol{R}_2$ is not zero, the resulting $\boldsymbol{R}$ is not zero as well. Let S be a matrix consisting of rows of B_1 that belong to I. It then holds that $S \cdot (\boldsymbol{R})^t = S_1\,(\boldsymbol{R}_1)^t + S_2\,(\boldsymbol{R}_2)^t = \boldsymbol{0}$. Since other rows of B_1 are linearly dependent on S, we have $B_1\,(\boldsymbol{R})^t = \boldsymbol{0}$ as expected.

We next show that $\mathcal{A}$ outputs a valid answer. First, $1 \leftarrow \mathsf{Vrf}(\Lambda, \boldsymbol{V}, \boldsymbol{C}, \boldsymbol{M}, \boldsymbol{D})$ holds due to the correctness of C. Recall that Vrf consists of evaluating PPEs. Every PPE in Vrf can be represented by

$$\mathsf{PPE}_i : \quad (\boldsymbol{V}||\boldsymbol{C}||\boldsymbol{M}||\boldsymbol{D})\,A_i\,(\boldsymbol{V}||\boldsymbol{C}||\boldsymbol{M}||\boldsymbol{D})^t = 1 \tag{4}$$

with some constant matrix A_i over $\mathbb{Z}_p$. Suppose that $\mathsf{Ext}_{\mathsf{Com}}$ is successful and (3) indeed holds. Then (4) can be rewritten by

$$(\boldsymbol{V}||\boldsymbol{M})\,E_i\,(\boldsymbol{V}||\boldsymbol{M})^t = 1 \tag{5}$$

with matrix E_i in which

$$E_i = F\,A_i\,F^t \quad \text{where} \quad F = \begin{pmatrix} \boldsymbol{1}_{\ell_v} & B_2^t & \boldsymbol{0}_{\ell_v} & B_4^t \\ \boldsymbol{0}_{\ell_m} & B_1^t & \boldsymbol{1}_{\ell_m} & B_3^t \end{pmatrix} \tag{6}$$

where $\boldsymbol{1}_n$ and $\boldsymbol{0}_n$ denote $n \times n$ identity and zero matrices over $\mathbb{Z}_p$, respectively. Note that E_i depends on $\boldsymbol{M}$ (through the computation of B_1 to B_4); hence, (5) holds for that $\boldsymbol{M}$. Nevertheless, we claim that any $\boldsymbol{M'}$ that is even unrelated to E_i fulfils (4) as long as (5) is fulfilled and $\boldsymbol{C}$ and $\boldsymbol{D}$ are computed as in (3).

Claim. For valid $\boldsymbol{M'}(\neq \boldsymbol{M})$ that fulfils

$$(\boldsymbol{V}||\boldsymbol{M'})\,E_i\,(\boldsymbol{V}||\boldsymbol{M'})^t = 1, \tag{7}$$

for all i, relation

$$(\boldsymbol{V}||\boldsymbol{C}'||\boldsymbol{M}'||\boldsymbol{D}')\, A_i\, (\boldsymbol{V}||\boldsymbol{C}'||\boldsymbol{M}'||\boldsymbol{D}')^t = 1 \tag{8}$$

holds for all i with respect to

$$(\boldsymbol{C}')^t := B_1\, (\boldsymbol{M}')^t + B_2\, (\boldsymbol{V})^t \quad \text{and} \quad (\boldsymbol{D}')^t := B_3\, (\boldsymbol{M}')^t + B_4\, (\boldsymbol{V})^t. \tag{9}$$

Proof is trivial by converting (7) into (8) by using (6) and (9). As a consequence, such $(\boldsymbol{C}', \boldsymbol{M}', \boldsymbol{D}')$ fulfils $1 \leftarrow \mathsf{Vrf}(\Lambda^*, \boldsymbol{V}, \boldsymbol{C}', \boldsymbol{M}', \boldsymbol{D}')$. We next make a strong claim that any $\boldsymbol{M}'$ satisfies (7).

Claim. If the discrete-logarithm problem in $\mathbb{G}$ is hard, the relation (7) holds for any $\boldsymbol{M}' \in \mathbb{G}^{\ell_m}$.

Intuition is that Com and $\mathsf{Ext}_{\mathsf{Com}}$ do not know the discrete-log of $\boldsymbol{M}$ in computing B_1 to B_4. Thus the only way to fulfil (5) is to set B_1 to B_4 so that (5) is trivial for $\boldsymbol{M}$. It then holds for any $\boldsymbol{M}'$ as in (7). To formally reduce to the hardness of the discrete-logarithm problem, we also require $\mathsf{Ext}_{\mathsf{Key}}$ to be algebraic so that $\boldsymbol{v}$ is available to our reduction algorithm.

Proof. Consider the relation in the exponents of (7) where $\boldsymbol{V}$ is a constant and $\boldsymbol{M}'$ is a variable. The relation is in a quadratic form, say $Q_i(\boldsymbol{m}') = 0$, whose coefficients can be computed efficiently from E_i. To prove the statement, it suffices to show that Q_i is a constant polynomial for all i.

Suppose, on the contrary, that there exists i where Q_i is a non-trivial polynomial with probability ϵ_Q that is not negligible. The probability is taken over the choice of $\boldsymbol{V}$, $\boldsymbol{M}$. (Recall that E_i depends on $\boldsymbol{V}$ and $\boldsymbol{M}$. It also depends on the randomness of the extractor of Com, but the theorem statement is conditional on the success of the extractor.) We construct algorithm $\mathcal{D}$ that solves the discrete logarithm problem by using Key, Com, and their extractors $\mathsf{Ext}_{\mathsf{Key}}$ and $\mathsf{Ext}_{\mathsf{Com}}$ as follows. Let (Λ, Y) be an instance of the discrete-logarithm problem where Λ includes base G. The goal is to compute $x := \log_G Y$. Given (Λ, Y), algorithm $\mathcal{D}$ first generates commitment key $(ck, tk) \leftarrow \mathsf{Key}(\Lambda, k)$ where $ck = (\Lambda^*, \boldsymbol{V})$. By invoking $\mathsf{Ext}_{\mathsf{Key}}$, algorithm $\mathcal{D}$ obtains discrete-log $\boldsymbol{v}$ of $\boldsymbol{V}$ with respect to G. ($\mathcal{D}$ halts if negligible extraction error occurs.) It then forms $\boldsymbol{M}$ by setting $M_j := Y^{\gamma_j}$ with random γ_j, and runs $(\boldsymbol{C}, \boldsymbol{D}) \leftarrow \mathsf{Com}(\Lambda^*, \boldsymbol{V}, \boldsymbol{M})$. By running $\mathsf{Ext}_{\mathsf{Com}}$, algorithm $\mathcal{D}$ obtains B_1, B_2, B_3 and B_4. It then computes E_i and further obtains quadratic polynomial Q_i that is non-trivial by hypothesis. By using the relation that $m_j = \gamma_j \cdot x$, $\mathcal{D}$ converts Q_i into quadratic polynomial Q_i' in x, which is also non-trivial except for negligible probability. (The probability is over the choice of every γ_i. Rigorously, the bound is given by Schwartz's lemma [21] since Q_i is a low-degree polynomial in γ_j.) Finally, $\mathcal{D}$ solves $Q_i'(x) = 0$ and outputs x. The running time of $\mathcal{D}$ is polynomial since Key, Com, and their extractors run in polynomial-time and other computations are obviously executable in polynomial-time. The success probability of $\mathcal{D}$ is almost the same as ϵ_Q except for the negligible errors. This contradicts the hardness of the discrete-logarithm problem in $\mathbb{G}$. ∎

Now recall that $\boldsymbol{M}'$ is set to $\boldsymbol{M} + \boldsymbol{R}$ and that $B_1\,\boldsymbol{R} = \boldsymbol{0}$. Thus,

$$(\boldsymbol{C}')^t = B_1\,(\boldsymbol{M}')^t + B_2\,(\boldsymbol{V})^t = B_1\,(\boldsymbol{M})^t + B_2\,(\boldsymbol{V})^t = (\boldsymbol{C})^{\boldsymbol{t}}. \tag{10}$$

Due to the above claims, $1 \leftarrow \mathsf{Vrf}(\Lambda, \boldsymbol{V}, \boldsymbol{C}, \boldsymbol{M}', \boldsymbol{D}')$ holds. Furthermore, $\boldsymbol{M} \neq \boldsymbol{M}'$ since $\boldsymbol{R} \neq \boldsymbol{0}$. Thus, $(\boldsymbol{C}, \boldsymbol{M}, \boldsymbol{D}, \boldsymbol{M}', \boldsymbol{D}')$ is a valid solution that breaks the binding property of C. This completes the proof in the symmetric group setting.

In the asymmetric setting where $\boldsymbol{M}$ and other vectors consist of elements from both $\mathbb{G}_1$ and $\mathbb{G}_2$, essentially the same argument holds since elements in the gruops do not mix each other. In the following, we only describe the points where the argument has to be adjusted.

- Every vector is split into $\mathbb{G}_1$ vector and $\mathbb{G}_2$ vector, e.g., $\boldsymbol{M} = (\boldsymbol{M}_1, \boldsymbol{M}_2) \in \mathbb{G}_1^{\ell_{m1}} \times \mathbb{G}_2^{\ell_{m2}}$ for $\ell_{m1} + \ell_{m2} = \ell_m$.
- By running $\mathsf{Ext}_{\mathsf{Com}}$, we obtain B_j in the form of

$$B_j = \begin{pmatrix} B_{j1} & \boldsymbol{0} \\ \boldsymbol{0} & B_{j2} \end{pmatrix} \tag{11}$$

  so that linear computation such as (3) is well defined.
- Without loss of generality, we assume that $|\boldsymbol{C}_1| < |\boldsymbol{M}_1|$. (Otherwise, $|\boldsymbol{C}_2| < |\boldsymbol{M}_2|$ holds.) Then, we can obtain non-zero vector $\boldsymbol{R}_1$ from B_{11} in the same way as we obtain $\boldsymbol{R}$ from B_1 in the symmetric case. By setting $\boldsymbol{R} = (\boldsymbol{R}_1, \boldsymbol{0})$, we have $B_1\boldsymbol{R} = \boldsymbol{0}$ as desired.
- Pairing product equations (4), (5), (7) and (8) are modified so that their left and right vectors consist only of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, for computational consistency. Also, matrix E_i in (6) is modified to $E_i = F_1\,A_i\,(F_2)^t$ where F_i is formed by using B_{1i}, B_{2i}, B_{3i}, and B_{4i} in the same manner as in F in (6).
- In the second claim, we require hardness of the discrete-logarithm problem in both $\mathbb{G}_1$ and $\mathbb{G}_2$. Depending on which of $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ polynomial Q_i is non-trivial, we solve the discrete-logarithm problem in $\mathbb{G}_1$ or $\mathbb{G}_2$, respectively.

∎

3.2 Number of Verification Equations

Theorem 10. *If $\Lambda = \Lambda_{\mathsf{sym}}$, $\ell_m \geq 2$, and Vrf evaluates only one PPE, then C is not binding.*

Proof. By focusing on M_1 and M_2 in $\boldsymbol{M}$, the PPE in the verification can be written as

$$e(M_1, M_1)^{a_1}\; e(M_1, K_1)^{b_1}\; e(M_2, M_2)^{a_2}\; e(M_2, K_2)^{b_2}\; e(M_1, M_2)^{c}\; P = 1 \tag{12}$$

where $a_1, b_1, a_2, b_2, c \in \mathbb{Z}_p$ are constants determined by the common parameter, and K_1 and K_2 are linear combinations of elements in $\boldsymbol{V}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{M} \setminus \{M_1, M_2\}$, and P is a product of pairings that does not involve M_1 and M_2. Let f be the polynomial that represents the relation in the exponent of the leftmost five pairings of (12). Namely,

$$f := a_1\, m_1^2 + b_1\, k_1\, m_1 + c\, m_1 m_2 + b_2\, k_2\, m_2 + a_2\, m_2^2, \tag{13}$$

where m_1, m_2, k_1, and k_2 are the discrete-logs of M_1, M_2, K_1, and K_2 with respect to the generator, say G, in Λ.

Given a commitment-key $(\Lambda^*, \boldsymbol{V})$, we set $\boldsymbol{M} = \mathbf{1}$ and honestly compute $\boldsymbol{C}$ and $\boldsymbol{D}$ by running Com. These $\boldsymbol{C}$ and $\boldsymbol{D}$ define K_1, K_2, and P in (12). Let $f(m_1, m_2)$ be f, as defined in (13), with k_1 and k_2 determined by these K_1 and K_2. We have $f(0,0) = 0$ and look for another pair $(m_1', m_2') \neq (0,0)$ that fulfils $f(m_1', m_2') = 0$. Such (m_1', m_2') yield $(M_1', M_2') = (G^{m_1'}, G^{m_2}) \neq (1,1)$.

Next, we show how to obtain such (M_1', M_2'):

- If $(a_1, a_2, c) = (0,0,0)$, we have $f(m_1, m_2) = b_1\, k_1\, m_1 + b_2\, k_2\, m_2$. We then proceed with the following sub-cases.
  - If $b_1\, k_1 \neq 0$ and $b_2\, k_2 \neq 0$, then $m_1' := k_2$ and $m_2' := (-b_1/b_2) \cdot k_1$ results in $(m_1', m_2') \neq (0,0)$ and $f(m_1', m_2') = 0$. Thus, setting $M_1' := K_2$ and $M_2' := K_1^{-b_1/b_2}$ works.
  - If $b_i k_i = 0$ for $i = 1$ or $i = 2$, or both, $f(m_1, m_2)$ is independent of m_i. Therefore, any non-zero m_i' suffices. Simply select arbitrary non-zero m_i' and compute $M_i' = G^{m_i'}$.
- If $(a_1, a_2, c) \neq (0,0,0)$, we do as follows.
  - If $b_1\, k_1 = 0$ and $b_2\, k_2 = 0$, we have $f(m_1, m_2) = a_1\, m_1^2 + c\, m_1\, m_2 + a_2\, m_2^2$. By selecting non-zero m_1' and solving m_2' for $f = 0$ (if $f(m_1, m_2) = 0$ is independent of m_2, arbitrary m_2' suffices), we have $(M_1', M_2') = (G^{m_1}, G^{m_2}) \neq (1,1)$.
  - If $b_1\, k_1 \neq 0$ or $b_2\, k_2 \neq 0$, we consider setting $m_2 = \delta\, m_1$ for some δ. With this relation, (13) is written as

$$f(m_1, m_2) = m_1 \left\{ (a_1 + a_2\, \delta^2 + c\, \delta)\, m_1 + (b_1\, k_1 + b_2\, k_2\, \delta) \right\}. \tag{14}$$

    We need (14) to have a non-zero solution for m_1. Therefore, we set δ so that $a_1 + a_2\, \delta^2 + c\, \delta \neq 0$ and $b_1\, k_1 + b_2\, k_2\, \delta \neq 0$ hold. (There are at most two δ for which these inequalities do not hold. For an arbitrary δ, the first inequality can be tested directly, whereas the second is through the relation $K_1^{b_1} K_2^{b_2 \delta} \neq 1$. Thus, by trying at most three non-zero different δ, we have an appropriate δ.) Then

$$m_1' = -\frac{b_1\, k_1 + b_2\, k_2 \delta}{a_1 + a_2\, \delta^2 + c\, \delta} \quad \text{and} \quad m_2' = \delta\, m_1'$$

    fulfil $(m_1', m_2') \neq (0,0)$ and $f(m_1', m_2') = 0$. This corresponds to setting

$$M_1' := (K_1^{b_1} K_2^{b_2\, \delta})^{\frac{1}{a_1 + a_2\, \delta^2 + c\, \delta}} \quad \text{and} \quad M_2' := (M_1')^{\delta}.$$

By replacing M_1 and M_2 in $\boldsymbol{M}$ with M_1' and M_2' computed as described above, we obtain $\boldsymbol{M}' \neq \boldsymbol{M}$, which is consistent with $\boldsymbol{C}$ and $\boldsymbol{D}$; Hence, the binding property breaks.

4 Optimal Constructions

4.1 In Asymmetric Setting

Let $\mathcal{G}$ be a generator of asymmetric bilinear groups. Scheme 1 in Fig. 1 is for messages $\boldsymbol{M} = (M_1, \ldots, M_k) \in \mathbb{G}_2^k$ for some fixed constant k specified at the time of commitment-key generation. The default generators G and $\tilde{G}$ in Λ can be used as G_0 and H, respectively. One can switch $\mathbb{G}_1$ and $\mathbb{G}_2$ in the description to obtain a dual scheme that accepts messages in $\mathbb{G}_1$. It also implies a scheme for messages from both $\mathbb{G}_1$ and $\mathbb{G}_2$. We show that the scheme is correct, perfectly hiding, and computationally binding as well as trapdoor and homomorphic.

[Scheme 1]

$\mathsf{Setup}(1^\lambda)$: Run $\mathcal{G}(1^\lambda)$ and obtain $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$. Output Λ.
$\mathsf{Key}(\Lambda, k)$: Select G_0 and H uniformly from $\mathbb{G}_1^*$ and $\mathbb{G}_2^*$, respectively. For $i = 1, \ldots, k$, compute $G_i := G_0^{\gamma_i}$ for random $\gamma_i \in \mathbb{Z}_p^*$. Output commitment-key $ck = (\Lambda^*, H, G_0, \ldots, G_k)$ and trapdoor $tk = (\gamma_1, \ldots, \gamma_k)$.
$\mathsf{Com}(ck, \boldsymbol{M})$: Randomly select $\tau_0, \ldots, \tau_k \in \mathbb{Z}_p$ and compute

$$C_i := M_i \cdot H^{\tau_i} \text{ (for } i = 1, \ldots, k), \quad C_{k+1} := \prod_{j=0}^{k} G_j^{\tau_j}, \text{ and} \tag{15}$$

$$D := H^{\tau_0}. \tag{16}$$

Then output $\boldsymbol{C} := (C_1, \ldots, C_{k+1})$ and D.
$\mathsf{Vrf}(ck, \boldsymbol{C}, \boldsymbol{M}, D)$: Output 1 if

$$e(C_{k+1}, H) = e(G_0, D) \prod_{i=1}^{k} e(G_i, C_i/M_i) \tag{17}$$

holds. Output 0, otherwise.
$\mathsf{Equiv}(ck, tk, \boldsymbol{C}, \boldsymbol{M}, D, \boldsymbol{M}')$: Take $(\gamma_1, \ldots, \gamma_k)$ from tk. Then output D' such that

$$D' := D \cdot \prod_{i=1}^{k} (M_i'/M_i)^{\gamma_i}. \tag{18}$$

Fig. 1. Homomorphic trapdoor commitment scheme in asymmetric bilinear group setting

Theorem 11. *Scheme 1 is correct.*

Proof. For any $\boldsymbol{C}$ and D correctly computed for ck and $\boldsymbol{M}$ as in (15), the right-hand of verification equation (17) is

$$e(G_0, D) \prod_{i=1}^{k} e(G_i, C_i/M_i) = e(G_0, H^{\tau_0}) \prod_{i=1}^{k} e(G_i, H^{\tau_i}) = e(C_{k+1}, H). \tag{19}$$

Thus $(ck, \boldsymbol{C}, \boldsymbol{M}, D)$ passes the verification with probability 1.

Theorem 12. *Scheme 1 is perfectly hiding and computationally binding if the DBP assumption holds for Λ.*

Proof. It is perfectly hiding because, for every commitment $\boldsymbol{C} = (C_1, \ldots, C_{k+1}) \in \mathbb{G}_1 \times \mathbb{G}_2^k$ and every message $\boldsymbol{M} = (M_1, \ldots, M_k) \in \mathbb{G}_2^k$, there exists a unique $(\tau_0, \ldots, \tau_k) \in \mathbb{Z}_p^{k+1}$ that is consistent with relations (15), (16) and (17).

The binding property is proven by constructing an algorithm $\mathcal{B}$ that breaks DBP using an adversary $\mathcal{A}$ that successfully computes two openings for a commitment. Given an instance (Λ, G_z, G_r) of DBP, algorithm $\mathcal{B}$ works as follows.

- Randomly select ρ_0 from $\mathbb{Z}_p^*$ and compute $G_0 := G_r^{\rho_0}$.
- For $i = 1, \ldots, k$, randomly select $\zeta_i \in \mathbb{Z}_p^*$ and $\rho_i \in \mathbb{Z}_p$ and compute $G_i := G_z^{\zeta_i} G_r^{\rho_i}$. If $G_i = 1$ for any i, $\mathcal{B}$ aborts; since the probability for this is negligible, this does not affect the overall success of $\mathcal{B}$.
- Run $\mathcal{A}$ with input $ck = (\Lambda^*, H, G_0, \ldots, G_k)$.
- Given commitment $\boldsymbol{C}$ and two openings $(\boldsymbol{M}, \boldsymbol{D})$ and $(\boldsymbol{M}', \boldsymbol{D}')$ from $\mathcal{A}$, compute

$$Z^\star = \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\zeta_i} \qquad \text{and} \qquad R^\star = \left(\frac{D}{D'}\right)^{\rho_0} \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\rho_i}. \tag{20}$$

- Output $(Z^\star, R^\star)$.

Since both $(\boldsymbol{M}, \boldsymbol{D})$ and $(\boldsymbol{M}', \boldsymbol{D}')$ fulfil (17) for the same commitment $\boldsymbol{C}$, dividing the two verification equations yields

$$1 = e\left(G_0, \frac{D}{D'}\right) \prod_{i=1}^{k} e\left(G_i, \frac{M_i'}{M_i}\right) = e\left(G_r^{\rho_0}, \frac{D}{D'}\right) \prod_{i=1}^{k} e\left(G_z^{\zeta_i} G_r^{\rho_i}, \frac{M_i'}{M_i}\right) \tag{21}$$

$$= e\left(G_z, \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\zeta_i}\right) e\left(G_r, \left(\frac{D}{D'}\right)^{\rho_0} \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\rho_i}\right) \tag{22}$$

$$= e(G_z, Z^\star)\, e(G_r, R^\star). \tag{23}$$

Since $\boldsymbol{M} \neq \boldsymbol{M}'$, there exists i such that $M_i'/M_i \neq 1$. Also, ζ_i is independent from the view of the adversary, i.e., for every choice of ζ_i, there exist a corresponding ρ_i that gives the same G_i. Accordingly, $Z^\star = \prod_i (M_i'/M_i)^{\zeta_i} \neq 1$ holds with overwhelming probability, and $(Z^\star, R^\star)$ is a valid answer to the instance of DBP. Therefore, $\mathcal{B}$ breaks DBP with the same probability that $\mathcal{A}$ breaks the binding property of Scheme 1 (minus a negligible difference). ∎

Theorem 13. *Scheme 1 is trapdoor and homomorphic.*

Proof. For the trapdoor property, observe that, for any trapdoor tk generated by Key, and for any valid $\boldsymbol{M}$ and $(\boldsymbol{C}, D)$ generated by Com, and D' generated by Equiv for any valid $\boldsymbol{M}'$, it holds that

$$e(G_0, D') \prod_{i=1}^{k} e(G_i, C_i/M'_i) = e(G_0, D \cdot \prod_{i=1}^{k} (M'_i/M_i)^{\gamma_i}) \prod_{i=1}^{k} e(G_i, C_i/M'_i) \quad (24)$$

$$= e(G_0, D) \prod_{i=1}^{k} e(G_0^{\gamma_i}, M'_i/M_i) \prod_{i=1}^{k} e(G_i, C_i/M'_i) \quad (25)$$

$$= e(G_0, D) \prod_{i=1}^{k} e(G_i, C_i/M_i) \quad (26)$$

$$= e(C_{k+1}, H). \quad (27)$$

Thus $(\boldsymbol{M}', D')$ is a correct opening of $\boldsymbol{C}$ computed from $\boldsymbol{M}$. Also observe that $(ck, \boldsymbol{M}, \boldsymbol{C})$ uniquely determines D and so is $(ck, \boldsymbol{M}', \boldsymbol{C})$ and D'. Therefore, distributions $(ck, \boldsymbol{M}, \boldsymbol{C}, D)$ and $(ck, \boldsymbol{M}', \boldsymbol{C}, D')$ over all choices of $\boldsymbol{M}$ and $\boldsymbol{M}'$ are identical.

To check the homomorphic property, let $(ck, \boldsymbol{C}, \boldsymbol{M}, D)$ and $(ck, \boldsymbol{C}', \boldsymbol{M}', D')$ satisfy verification equation (17). Also, let $\boldsymbol{M}^\star := \boldsymbol{M} + \boldsymbol{M}'$, $\boldsymbol{C}^\star := \boldsymbol{C} + \boldsymbol{C}'$, and $D^\star := D \cdot D'$. Then it holds that

$$e(G_0, D^\star) \prod_{i=1}^{k} e(G_i, C_i^\star/M_i^\star) \quad (28)$$

$$= e(G_0, D)\, e(G_0, D') \prod_{i=1}^{k} e(G_i, C_i/M_i) \prod_{i=1}^{k} e(G_i, C'_i/M'_i) \quad (29)$$

$$= e(C_{k+1}, H)\, e(C'_{k+1}, H) \quad (30)$$

$$= e(C^\star_{k+1}, H). \quad (31)$$

4.2 In Symmetric Setting

Let $\mathcal{G}$ be a generator of symmetric bilinear groups. Scheme 2 in Fig. 2 is for messages $\boldsymbol{M} = (M_1, \ldots, M_k) \in \mathbb{G}_1^k$ for some fixed constant k specified at the time of commitment-key generation. The default generator G in Λ can be used as H in the key generation.

Theorem 14. *Scheme 2 is correct.*

Proof. For correctly generated/computed $(ck, \boldsymbol{C}, \boldsymbol{M}, \boldsymbol{D})$, the following holds:

$$e(G_0, D_1) \prod_{i=1}^{k} e(G_i, C_i/M_i) = e(G_0, H^{\tau_0}) \prod_{i=1}^{k} e(G_i, H^{\tau_i}) = e(C_{k+1}, H) \quad (37)$$

$$e(F_0, D_2) \prod_{i=1}^{k} e(F_i, C_i/M_i) = e(F_0, H^{\mu_0}) \prod_{i=1}^{k} e(F_i, H^{\tau_i}) = e(C_{k+2}, H). \quad (38)$$

Thus it passes the verification with probability 1.

[Scheme 2]

$\mathsf{Setup}(1^\lambda)$: Run $\mathcal{G}(1^\lambda)$ and obtain $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_T, e, G)$. Output Λ.

$\mathsf{Key}(\Lambda, k)$: Select H, G_0 and F_0 from $\mathbb{G}_1$ uniformly. For $i = 1, \ldots, k$, compute $G_i := G_0^{\gamma_i}$ and $F_i := F_0^{\delta_i}$ for random $\gamma_i, \delta_i \in \mathbb{Z}_p^*$. Output $ck := (\Lambda^*, H, (G_i, F_i)_{i=0}^k)$ and $tk := ((\gamma_i, \delta_i)_{i=1}^k)$.

$\mathsf{Com}(ck, \boldsymbol{M})$: Choose $\mu_0, \tau_0, \ldots, \tau_k \in \mathbb{Z}_p$ randomly and compute (for $i = 1, \ldots k$)

$$C_i := M_i \cdot H^{\tau_i}, \quad C_{k+1} := G_0^{\tau_0} \prod_{j=1}^{k} G_j^{\tau_j}, \quad C_{k+2} := F_0^{\mu_0} \prod_{j=1}^{k} F_j^{\tau_j}, \tag{32}$$

$$D_1 := H^{\tau_0}, \text{ and } \quad D_2 := H^{\mu_0}. \tag{33}$$

Output $\boldsymbol{C} := (C_1, \ldots, C_{k+2})$ and $\boldsymbol{D} = (D_1, D_2)$.

$\mathsf{Vrf}(ck, \boldsymbol{C}, \boldsymbol{M}, \boldsymbol{D})$: Output 1 if the following equations hold. Output 0, otherwise.

$$e(C_{k+1}, H) = e(G_0, D_1) \prod_{i=1}^{k} e(G_i, C_i/M_i) \tag{34}$$

$$e(C_{k+2}, H) = e(F_0, D_2) \prod_{i=1}^{k} e(F_i, C_i/M_i) \tag{35}$$

$\mathsf{Equiv}(ck, tk, \boldsymbol{C}, \boldsymbol{M}, \boldsymbol{D}, \boldsymbol{M}')$: Parse tk as $((\gamma_i, \delta_i)_{i=1}^k)$. Output $\boldsymbol{D}' = (D_1', D_2')$ such that

$$D_1' := D_1 \cdot \prod_{i=1}^{k} (M_i'/M_i)^{\gamma_i}, \quad \text{and} \quad D_2' := D_2 \cdot \prod_{i=1}^{k} (M_i'/M_i)^{\delta_i}. \tag{36}$$

Fig. 2. Homomorphic trapdoor commitment scheme in symmetric bilinear group setting

Theorem 15. *Scheme 2 is perfectly hiding and computationally binding if the SDP assumption holds for Λ.*

Proof. It is perfectly hiding due to the uniform choice of $(\mu_0, \tau_0, \tau_1, \ldots, \tau_k)$ when committing, and due to the fact that for every commitment $\boldsymbol{C} = (C_1, \ldots, C_{k+2}) \in {\mathbb{G}_1}^{k+2}$ and for every message $\boldsymbol{M} = (M_1, \ldots, M_k) \in {\mathbb{G}_1}^k$ there exists a unique pair (D_1, D_2) that satisfies equations (34)-(35).

The binding property is shown by constructing an algorithm $\mathcal{B}$ that breaks SDP using an adversary $\mathcal{A}$ that successfully computes two openings for a commitment. Given an instance $(\Lambda, G_z, G_r, F_z, F_s)$ of SDP, algorithm $\mathcal{B}$ works as follows.

- Pick random ρ_0 and ω_0 from $\mathbb{Z}_p^*$ and compute $G_0 := G_r^{\rho_0}$, and $F_0 := F_s^{\omega_0}$.
- For $i = 1, \ldots, k$, pick random $\zeta_i \in \mathbb{Z}_p^*$ and $\rho_i, \omega_i \in \mathbb{Z}_p$ and compute $G_i := G_z^{\zeta_i} G_r^{\rho_i}$, and $F_i := F_z^{\zeta_i} F_s^{\omega_i}$. If $G_i = 1$ or $F_1 = 1$ for any i, $\mathcal{B}$ aborts; since the probability for this is negligible, we can ignore such cases.

– Run $\mathcal{A}$ with input $ck = (\Lambda^*, H, G_0, F_0, \ldots, G_k, F_k)$.
– Given commitment $\boldsymbol{C}$ and two openings $(\boldsymbol{M}, \boldsymbol{D})$ and $(\boldsymbol{M}', \boldsymbol{D}')$ from $\mathcal{A}$, compute

$$Z^\star = \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\zeta_i}, R^\star = \left(\frac{D_1}{D_2'}\right)^{\rho_0} \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\rho_i}, S^\star = \left(\frac{D_2}{D_2'}\right)^{\omega_0} \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\omega_i}.$$

– Output $(Z^\star, R^\star, S^\star)$.

Since both $(\boldsymbol{M}, D_1)$ and $(\boldsymbol{M}', D_1')$ fulfils (34) with $\boldsymbol{C}$, dividing the two equations yields

$$\begin{aligned} 1 &= e\left(G_0, \frac{D_1}{D_1'}\right) \prod_{i=1}^{k} e\left(G_i, \frac{M_i'}{M_i}\right) = e\left(G_r^{\rho_0}, \frac{D_1}{D_1'}\right) \prod_{i=1}^{k} e\left(G_z^{\zeta_i} G_r^{\rho_i}, \frac{M_i'}{M_i}\right) \\ &= e\left(G_z, \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\zeta_i}\right) e\left(G_r, \left(\frac{D_1}{D_1'}\right)^{\rho_0} \prod_{i=1}^{k} \left(\frac{M_i'}{M_i}\right)^{\rho_i}\right) \\ &= e(G_z, Z^\star)\, e(G_r, R^\star). \end{aligned}$$

Similarly, from $(\boldsymbol{M}, D_2)$ and $(\boldsymbol{M}', D_2')$ fulfilling (35) with $\boldsymbol{C}$, we have

$$1 = e\left(F_0, \frac{D_2}{D_2'}\right) \prod_{i=1}^{k} e\left(F_i, \frac{M_i'}{M_i}\right) = e(F_z, Z^\star)\, e(F_s, S^\star).$$

Since $\boldsymbol{M} \neq \boldsymbol{M}'$, there exists i such that $M_i'/M_i \neq 1$. Observe that ζ_i is independent from the view of the adversary, i.e., for every choice of ζ_i, there exist corresponding ρ_i and ω_i that give the same G_i and F_i, respectively. Thus, $Z^\star = \prod_i (M_i'/M_i)^{\zeta_i} \neq 1$ holds with overwhelming probability, and $(Z^\star, R^\star, S^\star)$ is a valid answer to the instance of SDP. Accordingly, $\mathcal{B}$ breaks SDP if $\mathcal{A}$ can break the binding property with a non-negligible probability. ∎

Theorem 16. *Scheme 2 is trapdoor and homomorphic.*

The proof is analogous to that that of Theorem 13; thus, it is omitted.

4.3 Efficiency

Table 1 compares storage and computation costs to commit to a message consisting of k group elements. Schemes for symmetric setting are above the line and those for asymmetric setting are below the line. In [3], another scheme in an asymmetric setting is discussed without details. The scheme yields a commitment of at least $2k$, which is not optimal.

We also assess the efficiency in combination of GS proofs. A typical proof statement would be "I can open the commitment." It uses $(\boldsymbol{M}, \boldsymbol{D})$ as witness and $(\boldsymbol{V}, \boldsymbol{C})$ as constants in the theorem statement represented by PPEs in the

Table 1. Efficiency comparison. The size indicates the number of elements in a commitment-key $\boldsymbol{V}$, a commitment $\boldsymbol{C}$, and a decommitment $\boldsymbol{D}$ for a message $\boldsymbol{M}$ consisting of k group elements. For Scheme 1, (x, y) means x elements in $\mathbb{G}_1$ (or $\mathbb{G}_2$) and y elements in $\mathbb{G}_2$ (or $\mathbb{G}_1$, resp.). #(pairings) and #(PPE) indicate the number of pairings and pairing product equations in the verification predicate, respectively.

Scheme	Setting	$\lvert\boldsymbol{V}\rvert$	$\lvert\boldsymbol{M}\rvert$	$\lvert\boldsymbol{C}\rvert$	$\lvert\boldsymbol{D}\rvert$	#(pairings)	#(PPE)	assumption
CLY09 [9]	Λ_{sym}	5	k	$3k$	$3k$	$9k$	$3k$	DLIN
AHO10 [3]	Λ_{sym}	$2k+2$	k	$2k+2$	2	$2k+2$	2	SDP
Scheme 2	Λ_{sym}	$2k+2$	k	$k+2$	2	$2k+4$	2	SDP
Scheme 1	Λ_{sxdh}	$(k,\ 0)$	$(0,\ k)$	$(1,\ k)$	$(0,\ 1)$	$k+2$	1	DBP

verification predicate. Table 2 shows the size of the witness, theorem, and proof in the example. We also show the total size for a theorem and a proof in bits with a reasonable parameter setting (which is considered as comparable security to an RSA modulus of 2000 bits) where elements in $\mathbb{G}$ are 380 bits in the symmetric setting, and elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ are 224 bits and 448 bits, respectively, assuming the use of point compression [4]. Scheme 1 is optimized by considering the dual scheme taking messages from $\mathbb{G}_1$.

Table 2. Storage costs for proving correct opening in zero-knowledge by GS proofs. Figures for |proof| include commitments of the witness and a proof. Size in bits indicates |theorem| + |proof| in bits.

Scheme	Setting	|witness|	|theorem|	|proof|	Size in Bits $k=1$	5	10
CLY09 [9]	Λ_{sym}	$4k$	$3k+5$	$39k$	17860	81700	161500
AHO10 [3]	Λ_{sym}	$k+2$	$4k+4$	$15k+24$	17860	46740	82840
Scheme 2	Λ_{sym}	$k+2$	$3k+4$	$12k+21$	15200	38000	66500
Scheme 1	Λ_{sxdh}	$(0,\ k+1)$	$(k+1,\ k)$	$(0,\ 6k+8)$	4256	12320	22400

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)
2. Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 649–666. Springer, Heidelberg (2011)
3. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on group elements for modular protocol designs. IACR ePrint Archive, Report 2010/133 (2010)
4. Barreto, P.S.L.M., Naehrig, M.: Pairing-Friendly Elliptic Curves of Prime Order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006)

5. Bellare, M., Palacio, A.: The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 273–289. Springer, Heidelberg (2004)
6. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
7. Boneh, D., Venkatesan, R.: Breaking RSA May Not Be Equivalent to Factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 59–71. Springer, Heidelberg (1998)
8. Camenisch, J., Haralambiev, K., Kohlweiss, M., Lapon, J., Naessens, V.: Structure Preserving CCA Secure Encryption and Applications. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 89–106. Springer, Heidelberg (2011)
9. Cathalo, J., Libert, B., Yung, M.: Group Encryption: Non-interactive Realization in the Standard Model. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 179–196. Springer, Heidelberg (2009)
10. Chase, M., Kohlweiss, M.: A domain transformation for structure-preserving signatures on group elements. IACR ePrint Archive, Report 2011/342 (2011)
11. Coron, J.-S.: Optimal Security Proofs for PSS and Other Signature Schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002)
12. Damgsard, I.: Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)
13. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient Public-Key Cryptography in the Presence of Key Leakage. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 613–631. Springer, Heidelberg (2010)
14. Galbraith, S., Paterson, K., Smart, N.: Pairings for cryptographers. IACR ePrint archive, Report 2006/165 (2006)
15. Galbraith, S.D., Rotger, V.: Easy decision-Diffie-Hellman groups. LMS Journal of Computation and Mathematics 7 (2004)
16. Groth, J.: Homomorphic trapdoor commitments to group elements. IACR ePrint Archive, Report 2009/007 (January 2009)
17. Groth, J.: Efficient Zero-Knowledge Arguments from Two-Tiered Homomorphic Commitments. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 431–448. Springer, Heidelberg (2011)
18. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008); Full version available: IACR ePrint Archive 2007/155
19. Hada, S., Tanaka, T.: On the Existence of 3-Round Zero-Knowledge Protocols. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 354–369. Springer, Heidelberg (1998); Full version available from IACR e-print archive 1999/009
20. Krawczyk, H., Rabin, T.: Chameleon hashing and signatures. IACR ePrint archive, Report 1998/010 (1998)
21. Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. Journal of the ACM 27(4) (1980)
22. Scott, M.: Authenticated id-based key exchange and remote log-in with simple token and pin number. IACR ePrint Archive, Report 2002/164 (2002)
23. Verheul, E.R.: Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. J. Cryptology 17(4), 277–296 (2004)