String commitment scheme with low output locality

Hideaki Miyaji
Graduate School of Engineering,
Osaka University
hideaki@cy2sec.comm.eng.osaka-u.ac.jp

Akinori Kawachi
Dept. Info. Eng.,
Mie University
QIQB, OTRI,
Osaka University
kawachi@cs.info.mie-u.ac.jp

Atsuko Miyaji
Graduate School of Engineering,
Osaka University
miyaji@comm.eng.osaka-u.ac.jp

***Abstract*—Commitment schemes are important tools for various protocols. However, no scheme with output locality have been proposed yet. Low output locality is a property of functions that every output bit to depend on a small number of input bits. In this paper, we construct a commitment scheme having low output locality from a modified lattice-based hash function for the first time. We also prove that our scheme satisfies the binding property by using the collision resistance of the lattice-based hash functions and the hiding property by using a modified version of the leftover hash lemma, respectively.**

***Index Terms*—hash functions, commitment schemes, output locality**

I. Introduction

The complexity of cryptographic primitives is a fundamental research problem for constructions of highly efficient and secure protocols [3], [9]. Applebaum et al. showed groundbreaking results for low-complexity cryptographic constructions of fundamental primitives [2]. Their technique provided a general framework for conversion to low-complexity cryptographic functions, including one-way and pseudorandom functions of low output locality.

The output locality is a natural complexity measure of computational efficiency for Boolean functions. A Boolean function has an output locality k, if each output bit depends on at most k input bits. One can easily see that functions of low-output locality are implementable by low-depth circuits, implying high parallelizability. In an extreme case, if a function has constant output locality, it can be decomposed into much smaller functions that can be computed by constant-depth circuits in parallel. The low-depth cryptographic functions play crucial roles in some situations. For example, the bootstrapping method requires a low-depth decryption function for lattice-based fully homomorphic public-key encryption [6].

From the results of Applebaum et al., approaches to low-complexity cryptographic constructions have been developed [2]. They most recently provided constructions for collision-resistant hash functions of low-output locality from computationally hard problems of lattices and multivariate polynomials [2].

On the other hand, a commitment scheme is a fundamental protocol and a key necessity for achieving basic cryptographic tasks, such as zero-knowledge identification and more [5]. The scheme is performed between two parties (i.e., sender and receiver) with commitment and decommitment phases. In the commitment phase, the sender converts his message to a commitment string and sends it to the receiver. In the decommitment phase, the sender sends his message and a decommitment string that allows the receiver to verify if the commitment string was indeed generated from his message. The security of commitment schemes is formalized by two properties (i.e., hiding and binding). The hiding property guarantees that no receiver can obtain any partial information of the message prior to the decommitment phase. The binding property guarantees that no sender can choose one of more than two candidate messages by switching decommitment strings during the decommitment phase.

In this paper, we examine the possibility of low-complexity constructions for higher-level cryptographic protocols. We propose the commitment scheme with low output locality for the first time. As far as the authors' knowledge, no commitment scheme has low output locality so far. For example, the well-known standard commitment schemes of Pedersen [10] and of Halevi and Micali [7] do not have low output locality. In particular, we construct the commitment scheme of non-trivial (expected) output locality from a variant of lattice problems, called the binary shortest-vector problem (bSVP).

To construct the commitment scheme with low output locality, we focus on two primitives. One is a commitment scheme from a well-known lattice problem of the short integer solution (SIS) problem [8]. Their scheme made use of a lattice-based collision resistant hash function of the "matrix-vector multiplication" form, $y = M \cdot x$ for a matrix M and vector x.

In their scheme, the binding property is shown from the collision resistant property of M. The hiding property is derived from the so-called leftover hash lemma [11]. They proved that every two commitment strings generated from distinct messages were statistically indistinguishable with high probability with respect to random choices of M. Another primitive is a collision-resistant hash function of low output locality based on the matrix-vector multiplication form with a randomized encoding function [2]. They first constructed a function, $f(x) = M \cdot ex(x)$, where ex is an expanding function that dilutes Hamming weight on input x to achieve non-trivial output locality and collision-resistant properties from the intractability of bSVP. Their important idea is to apply a randomized encoding function $\hat{f}(x)$ to the function $f(x)$ to achieve much lower output locality. Here, we say that a function $\hat{f}(x)$ is a randomized encoding function of f if $\hat{f}(x)$ satisfies δ-correctness and (t, ϵ)-privacy in [1].

A straightforward approach to construct a commitment scheme with low output locality is to combine these two primitives: the commitment scheme based on SIS, and the collision-resistant hash function with a randomized encoding function. However, it is not easy to give a proof on its hiding property of the commitment scheme based on the collision-resistant hash function with a randomized encoding function. To overcome this difficulty, we construct the collision-resistant hash function $\mathsf{H_{LO}}$ without using a randomized encoding function. We evaluate the output locality of $\mathsf{H_{LO}}$ explicitly. Then we construct the string commitment scheme $(\mathsf{S_{END}}, \mathsf{R_{ECEIVE}})$ based on $\mathsf{H_{LO}}$. We evaluate output locality of our commitment scheme in Theorem 3. We also prove that our commitment scheme satisfies the computationally-binding property in Theorem 4 and the statistically-hiding property in Theorems 5 and 6. When we give a proof on the statistically-hiding property, we prove a variant of the leftover hash lemma applicable to random inputs with diluted Hamming weight and biased random matrices.

This paper is organized as follows. Section II summarizes the commitment scheme, the hash function, and output locality. Section III explains building blocks of our construction. In Section IV, we present our commitment scheme. Finally, we conclude our results in Section V.

II. Preliminaries

In this section, we summarize the commitment scheme, the hash function, and output locality.

A. Commitment Scheme

First, we summarize the notation used in this paper.

- S: sender
- R: receiver
- 1^k: security parameter
- a: message string
- r: random string
- c: commitment string
- d : decommitment string
- $d' \neq d$: decommitment string, where S wants to cheat
- $\perp$: rejection symbol that R outputs for invalid inputs
- $\varepsilon(k)$: negligible function
- $\mathrm{Hw}(x)$: Hamming weight of x
- Comm: a commitment scheme
- $\mathsf{S_{END}}$: an algorithm that makes a commitment string from security parameter 1^k
- $\mathsf{R_{ECEIVE}}$: an algorithm that verifies the correctness of the commitment string from $(1^k, c, d)$
- n: a positive integer
- m: a positive integer less than n
- $H_2(p) := -p\log(p) - (1-p)\log(1-p)$ denotes the binary entropy function where $p \in [0, 1]$

Next, we provide a definition of the commitment scheme, which follows [4].

Definition 1 (Commitment Scheme)**.** *A commitment scheme,* $\mathsf{Comm}(\mathsf{S_{END}}, \mathsf{R_{ECEIVE}})$*, is a two-phase protocol between the probabilistic polynomial-time party,* S*, and the unbounded computational party,* R*, respectively the sender and receiver.*

During the first phase (commitment), S *commits string* a *to a pair of keys,* (c, d)*, by executing* $(c, d) \longleftarrow \mathsf{S_{END}}(1^k, \mathsf{a})$ *and by sending c (commitment string) to* R*. Given c, the unbounded computational party receiver,* R*, cannot guess the string with a probability significantly better than* ε*. This is the hiding property.*

During the second phase (decommitment), S *reveals the string,* a*, and sends the key, d, (decommitment string) and* a *to* R*. Next,* R *checks by executing* $\mathsf{R_{ECEIVE}}(1^k, c, d)$ *whether* *the decommitment string is valid. If not,* $\mathsf{R}_{\mathsf{ECEIVE}}(1^k, c, d)$ *outputs a special string,* $\perp$*, meaning that* R *rejects the decommitment of* S*. Otherwise,* $\mathsf{R}_{\mathsf{ECEIVE}}(1^k, c, d)$ *can efficiently compute the string,* a*, revealed by* S*, and sees that* a *was indeed chosen by* S *during the first phase. This is the binding property.*

In the following, we give security notions of the commitment scheme $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$ in Definitions 2 and 4.

Definition 2 (Computationally binding property [7])**.** *We say that* $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$ *is computationally binding if it is computationally infeasible to generate a commitment string, c, and two decommitment strings,* $d, d'(d \neq d')$*, such that* R *would compute one message,* a*, from* (c, d) *and a different message* a' *from* (c, d')*. In detail, for every probabilistic polynomial-time adversary, given* ${\mathsf{S}_{\mathsf{END}}}'(1^k)$*, the following occurs.*

$$\Pr\left[\begin{array}{ll} & \mathsf{R}_{\mathsf{ECEIVE}}(1^k, c, d) \neq\perp, \\ (c, d, d') \leftarrow {\mathsf{S}_{\mathsf{END}}}'(1^k): & \mathsf{R}_{\mathsf{ECEIVE}}(1^k, c, d') \neq\perp, \\ & \mathsf{R}_{\mathsf{ECEIVE}}(1^k, c, d) \\ & \neq \mathsf{R}_{\mathsf{ECEIVE}}(1^k, c, d') \end{array}\right]$$

$< \varepsilon(k)$,
where $\varepsilon(k)$ *is a negligible function in k. We then say that the commitment scheme,* $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$*, is computationally binding.*

Before we define statistically hiding, we review a definition of statistical distances.

Definition 3 (Statistical distance)**.** *Given two probability distributions,* ϕ_1 *and* ϕ_2*, on a finite set, S, we define the statistical distance between them as*
$d(\phi_1, \phi_2) := \frac{1}{2}\sum_{x \in S} |\phi_1(x) - \phi_2(x)|$.

Definition 4 (Statistically hiding property [7])**.** *Let* $\mathsf{a} \in \{0,1\}^*$ *be a message string, and let* $C_k(\mathsf{a})$ *denote the distribution over the commitment string for* a *in a commitment scheme,* $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$*. Thus,* $C_k(\mathsf{a})$ *is the distribution of the first coordinates of the pair that is obtained by running algorithm* $\mathsf{S}_{\mathsf{END}}(1^k, \mathsf{a})$*. The commitment scheme,* $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$*, is statistically hiding if* R *cannot distinguish two commitment strings less than* ε*, as follows;*

$$\forall \mathsf{a}_1, \mathsf{a}_2 \in \{0,1\}^*, d\left(C_k(\mathsf{a}_1), C_k(\mathsf{a}_2)\right) < \varepsilon(k),$$

where $\varepsilon(k)$ *is a negligible function in k.*

In this paper, we construct the string commitment scheme, $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$, which satisfies computationally-binding and statistically-hiding properties.

B. Hash Functions

In this section, we summarize the definition of hash functions. A hash function converts input bits of arbitrary length into compressed output bits of shorter length. When a hash function satisfies Definition 5, it has collision resistance.

Definition 5 (Collision Resistance)**.** *We have arbitrary probabilistic polynomial algorithm, Adv, given a description of hash function and length parameter as inputs. If the probability of Adv to output* $x, x' \in \{0,1\}^k$ *satisfying* $x \neq x'$ *and* $f(x) = f(x')$ *is negligible, the function is a collision-resistant hash function.*

$$\Pr[Adv(f, 1^k) \to x, x'(x \neq x')\ s.t.\ f(x) = f(x')] < \varepsilon(k),$$

where $\varepsilon(k)$ *is a negligible function in k.*

We construct a string commitment scheme from lattice-based hash functions. We use a slightly modified version of the hash functions of Applebaum et al. [2]. The hash function has output locality, and it is based on binary shortest vector problem (bSVP). We review bSVP in Definition 9. First, we provide the definitions of output locality, expansion, relative hamming weight, and hamming weight.

Definition 6 (Output Locality)**.** *We say function h has output locality d if each of output bits depends on at most d input bits.*

Definition 7 (Expansion)**.** *We say* n/k *is an expansion of a function* $f : \{0,1\}^k \to \{0,1\}^n (k < n)$*.*

Definition 8 (Hamming Weight, Relative Hamming Weight)**.** *Let* $x = (x_1, ..., x_n) \in \mathbb{F}_2^n$ *be a vector. The number of "1"s in x is called the Hamming weight, which we denote by* $\mathrm{Hw}(x)$*. The ratio of "1"s in x is called the relative Hamming weight, which we denote by* $\Delta(x)$*.*

Definition 9 (bSVP)**.** *For a weight parameter,* $\delta(n), \delta : \mathbb{N} \to (0, 1/2)$*, and an efficient sampler,* $M(1^n)$*, which samples* $m \times n$ *binary matrices, the (M, δ)–bSVP assumption asserts that, for every efficient algorithm, Adv, the probability,*

$$Pr_{M \overset{R}{\leftarrow} M(1^n)}[Adv(M) = x | x \neq 0, Mx = 0 \ and\ \Delta(x) \leq \delta]$$

$< \varepsilon(n)$

where $\Delta(x)$ is the relative Hamming weight of x and $\varepsilon(n)$ is a negligible function in n.

In [2], they supposed that the matrix sampler $M(1^n)$ generates a uniformly random binary $m \times n$ matrix. In this paper, we consider a slightly biased version of matrix samplers. Our matrix sampler $M_\gamma(1^n)$ generates $m \times n$ binary matrix of which each element is taken to 1 (and to 0) with probability γ (and $1-\gamma$, respectively) independently of other elements for some constant $\gamma \in (0, 1/2)$. Thus, every row of the matrix is expected to have γn "1"s rather than $n/2$ in the case of [2].

III. Building blocks

In this section, we review the hash-function construction that we use in this paper. The collision resistance of the hash function is based on the bSVP assumption. Applebaum et al. constructed an expand function to use the bSVP assumption, which dilutes relative Hamming weight of input bits. (In the statement of the lemma, we fix output locality to 1, but they showed a general case of output locality d.)

Lemma 1 (Expand Function with Low Output Locality [2])**.** *Fix any $\delta \in (0, 1/2)$. Suppose $\beta \le \delta/2$ and $n/k = \lceil 1/H_2(\beta) \rceil$ for $\beta \in (0, 1/4)$ and natural numbers n, k. There exists an efficiently computable function $ex : \{0,1\}^k \to \{0,1\}^n$ such that (1) ex is injective, (2) $\Delta(ex(x)) \le \beta$ for every x, and (3) ex has output locality* 1.

proof: In the statement of this lemma, we fixed output locality to 1, but we show a general case of output locality d. Suppose that $n/k = c/d$ for constants c and d.

We construct a function, $ex : \{0,1\}^k \to \{0,1\}^n$, consisting of two steps. First, we divide input $\{0,1\}^k$ into blocks of d bits and extend a d-bit block to a c-bit block by a function, $ex0 : \{0,1\}^d \to \{0,1\}^c$, which is provided in Algorithm 1. Note that $c = \lceil 1/H_2(\beta) \rceil d$. Then, the number of strings of relative Hamming weight at most β is less than 2^d. Hence, the outputs of $ex0$ have relative Hamming weight at most β and $ex0$ is injective. In addition, $ex0$ obviously has output locality d.

We combine $ex0$ to construct a function ex, which we describe in Algorithm 2. From the construction of $ex0$, the properties of (1), (2) and (3) (of output locality d) are immediately satisfied. We will show the images of expand function in Fig. 1.

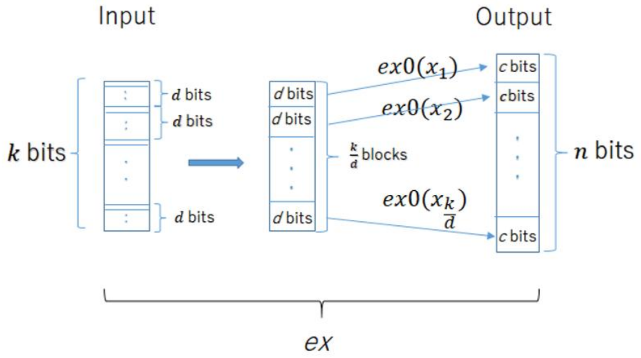

Fig. 1. ex function

Algorithm 1 $ex0$ function

```
Input: $x \in \{0,1\}^d$
Output: $ex0(x_i) \in \{0,1\}^c$
Identify $x \in \{0,1\}^d$ as a binary representation of natural numbers in $\{0, \dots, 2^d - 1\}$ naturally.
Set $y \in \{0,1\}^c$ to the $(x+1)$-th string of relative Hamming weight at most $\beta$.
return $y \in \{0,1\}^c$.
```

Algorithm 2 ex function

```
Input: $x \in \{0,1\}^k$
Output: $ex(x) \in \{0,1\}^n$
Partition $k$-bit input into $k/d$ input blocks of $d$ bits each.
Apply $ex0$ to each input block and generate $k/d$ output blocks of $c$ bits.
return $ex0(x_1) \circ ex0(x_2) \circ .. \circ ex0(x_{k/d}) = ex(x)$.
```

■

IV. Proposed Commitment Scheme

A. Proposed Hash Function and its Low-Output Locality

Our proposed commitment scheme uses a hash function with low output locality. The hash function comprises a matrix and an expand function, ex, as shown in Algorithm 3. Let $\mathsf{H}_{\mathsf{LO}} : \{0,1\}^k \to \{0,1\}^m$.

Algorithm 3 Hash function: $\mathsf{H_{LO}}$

```
Input: $x \in \{0,1\}^k$ , $M \in M_{m,n}(\mathbb{F}_2)$.
Output: $y \in \{0,1\}^m$
ex : $x \in \{0,1\}^k \rightarrow \{0,1\}^n$
Compute $y = M \cdot ex(x)$.
return $y$
```

Next, we discuss the output locality of $\mathsf{H_{LO}}$. By using the expand function, ex, the output locality of ex can be suppressed by the constant, 1, in Lemma 1. However, $\mathsf{H_{LO}}$ needs to multiply the output of ex by a matrix, M, where output of ex is treated as a vector. This is why we cannot suppress the output locality as a constant.

Theorem 1 (Output Locality). *Let $c = n/k = \lceil 1/H_2(\beta) \rceil$, let γ be an expected relative Hamming weight in every row of the matrix M. Then, the hash function $\mathsf{H_{LO}}$ has expected $\gamma \cdot c \cdot k$ output locality.*

proof: We first review the definition of output locality. A function h has output locality d if each output depends on at most d inputs. Thus, we compare between the number of input bits influencing every output bit in $\mathsf{H_{LO}}$ and the number of input bits k. If $\mathsf{H_{LO}}$ has output locality, the number of input bits influencing each output bit in $\mathsf{H_{LO}}$ should be less than k.

$\mathsf{H_{LO}}$ is expressed by $M \cdot ex(x)$. The matrix in $\mathsf{H_{LO}}$ is taken to 1 (and to 0) with probability γ (and $1-\gamma$, respectively) independently of other elements for some constant $\gamma \in (0, 1/2)$. An expected "1"s in every row in the matrix is γn. The output locality in $ex(x)$ is 1 from Lemma1, so an expected output locality in $\mathsf{H_{LO}}$ is γn. From $n = ck$, $\gamma n \cdot 1 = \gamma \cdot ck$. Therefore, $\mathsf{H_{LO}}$ has expected $\gamma \cdot c \cdot k$ output locality. ▮

From Theorem 1, the expected output locality γck is strictly less than k by setting $\gamma < \lceil 1/H_2(\beta) \rceil^{-1}$.

B. Collision resistance of $\mathsf{H_{LO}}$

In this section, we show that $\mathsf{H_{LO}}$ satisfies collision resistance under the bSVP assumption.

Theorem 2. $\mathsf{H_{LO}}$ *has collision resistance under the (M_γ, δ)–bSVP assumption.*

proof:

We assume there exists an adversary, Adv, which defeats collision resistance. Then, we can write the following for some non-negligible function $\varepsilon'(n)$.

$\exists Adv \; s.t. \; \Pr_M[Adv(M) \rightarrow (x_1, x_2) s.t. \; x_1 \neq x_2 \wedge M \cdot ex(x_1) = M \cdot ex(x_2)] > \varepsilon'(n).$

Adv can obtain (x_1, x_2) from the given M as an input. Next, we construct another adversary, Adv', which breaks the bSVP assumption.

Algorithm 4 Adv'

```
Input: $M$
Output: $ex(x_1), ex(x_2)$
Execute $Adv(M) \rightarrow (x_1, x_2)$.
Compute $ex(x_1), ex(x_2)$ from $x_1, x_2$.
return $ex(x_1), ex(x_2)$
```

From Algorithm 4, the adversary Adv' executes

$$Adv'(M) \rightarrow (ex(x_1), ex(x_2)).$$

$(ex(x_1) \neq ex(x_2)$ from injectivity of $ex.)$

We next compute $ex(x_1) - ex(x_2)$ and obtain $M(ex(x_1) - ex(x_2)) = M \cdot ex(x_1) - M \cdot ex(x_2)$. The relative hamming weight of each $ex(x_1)$ and $ex(x_2)$ are at most β. Thus, the relative hamming weight of $ex(x_1) - ex(x_2)$ is at most 2β. From the construction of ex in Lemma 1 we have $\beta \leq \delta/2$. Hence, the relative hamming weight of $ex(x_1) - ex(x_2)$ is less than δ. We can derive an equation below for some non-negligible function $\varepsilon''(n)$ and $x' = ex(x_1) - ex(x_2)$.

$$\Pr_{\mathrm{M} \overset{R}{\leftarrow} M(1^n)}[\text{Adv'(M)} = x' | x' \neq 0, Mx' = 0 \text{ and } \Delta(x') \leq \delta] > \varepsilon''(n)$$

.

Algorithm 4 shows that Adv' can defeat the bSVP assumption in Definition 9. Therefore, $\mathsf{H_{LO}}$ has collision resistance via the contraposition. ▮

Now, we propose a string commitment scheme, $\mathsf{Comm}(\mathsf{S_{END}}, \mathsf{R_{ECEIVE}})$ for a message string $\mathsf{a} \in \{0,1\}^{k/2}$, based on $\mathsf{H_{LO}}$ in section IV-C.

C. Protocol of Proposed Commitment Scheme

In this section, we show our proposed commitment scheme based on hash function $\mathsf{H_{LO}}$. The commitment scheme comprises initialization and a commitment phase and a decommitment phase. We explain each phase.

Initialization

Prior to the commitment phase, both S and R share the following information.

- ex:$\{0,1\}^k \rightarrow \{0,1\}^n$
- matrix $M \in M_{m,n}(\mathbb{F}_2)$
- 1^k : security parameter

Commitment Phase by S

1) Choose a random number $\mathsf{r} \in \{0,1\}^{k/2}$ as the key of hash functions.
2) Choose a message string $\mathsf{a} \in \{0,1\}^{k/2}$, and concatenate a and r as $c = \mathsf{a}||\mathsf{r}$.
3) Compute $ex(c) \in \{0,1\}^n$
4) Compute $M \cdot ex(c)$.
5) Sends $com(\mathsf{a};\mathsf{r}) = M \cdot ex(c)$ to R as a commitment string.

Decommitment Phase from S to R

S executes:

1) S sends $(\mathsf{a},\mathsf{r}) \in \{0,1\}^{k/2} \times \{0,1\}^{k/2}$ to R as a decommitment string d

R executes

1) Compute $c = \mathsf{a}||\mathsf{r}$ from d.
2) Compute $ex(c)$.
3) Compute $M \cdot ex(c)$ and if $com(\mathsf{a};\mathsf{r}) = M \cdot ex(c)$. If so, $\mathsf{R_{ECEIVE}}$ outputs a. Otherwise, $\mathsf{R_{ECEIVE}}$ outputs $\perp$.

Output locality of our commitment scheme follows easily from Theorem 1.

Theorem 3 (Output Locality). *Let $c = n/k = \lceil 1/H_2(\beta) \rceil$, let γ be an expected relative Hamming weight in every row of the matrix M. Then, our commitment scheme $(\mathsf{S_{END}}, \mathsf{R_{ECEIVE}})$ has expected $\gamma \cdot c \cdot k$ output locality.*

Next, we prove the security of the commitment scheme we constructed.

D. Computationally Binding Property

We show computationally binding property from the contraposition.

Theorem 4. *Our commitment scheme $\mathsf{Comm}(\mathsf{S_{END}}, \mathsf{R_{ECEIVE}})$ satisfies the computationally binding property under the (M_γ, δ)–bSVP assumption.*

proof:

We assume there exists an adversary, $Adv_{\mathsf{S_{END}}}$, that defeats the computationally binding property of the commitment scheme, $\mathsf{Comm}(\mathsf{S_{END}}, \mathsf{R_{ECEIVE}})$, based on $\mathsf{H_{LO}}$. We derive the equation below with the non-negligible function, $\varepsilon(k)$

$$\Pr\left[\begin{array}{cc} & \mathsf{R_{ECEIVE}}(1^k, c, d) \neq \perp, \\ (c,d,d') \leftarrow Adv_{\mathsf{S_{END}}}(1^k, M) : & \mathsf{R_{ECEIVE}}(1^k, c, d') \neq \perp, \\ & \mathsf{R_{ECEIVE}}(1^k, c, d) \\ & \neq \mathsf{R_{ECEIVE}}(1^k, c, d') \end{array}\right]$$

$$> \varepsilon(k).$$

$Adv_{\mathsf{S_{END}}}$ executes

$$Adv_{\mathsf{S_{END}}}(1^k, M) \rightarrow (c, d, d').$$

Next, we construct another adversary Adv' that solves the collision resistance of $\mathsf{H_{LO}}$. We describe how another adversary Adv' works after Adv gains c, d, d' in Algorithm 5.

Algorithm 5 Adv' for collision resistance

```
Input: $(c, d, d')$
Output: $x_1, x_2$
Compute $d = (\mathsf{a}, \mathsf{r}), d' = (\mathsf{a}', \mathsf{r}')$ from $(c, d, d')$.
Compute $x_1, x_2$ from message string $\mathsf{a}$ and random number $\mathsf{r}$. $x_1 = \mathsf{a} \circ \mathsf{r}$, $x_2 = \mathsf{a}' \circ \mathsf{r}'$
return Output $x_1, x_2$
```

Adv' executes $Adv'(c, d, d') \rightarrow (x_1, x_2)$. Adv' knows the value of matrix M. Thus, Adv' can compute $M \cdot x_1$ and $M \cdot x_2$. Therefore, we can derive the equation below with a non-negligible function, $\varepsilon'(k)$

$\Pr[Adv'(M \cdot x) \rightarrow (x_1, x_2) \ s.t. \ M \cdot x_1 = M \cdot x_2 \wedge (x_1 \neq x_2)] \geq \varepsilon'(k)$.

This shows there exists an Adv' that defeats the collision resistance of hash functions. However, we prove that $\mathsf{H_{LO}}$ has collision resistance by Theorem 2 under the (M_γ, δ)–bSVP assumption. Therefore, the commitment scheme has the computationally binding property under the (M_γ, δ)–bSVP assumption from contraposition. ∎

E. Proof of the Statistically Hiding Property

In this section, we prove that the commitment scheme, $\mathsf{Comm}(\mathsf{S_{END}}, \mathsf{R_{ECEIVE}})$, satisfies the statistically hiding property. We first consider a modified version of the leftover hash lemma, proved by Regev [11]. The modified version of the leftover hash lemma indicates that the statistical distance between the uniform distribution and the distribution of the sum of randomly selected subsets are small.

Lemma 2 (A version of the leftover hash lemma [11]). *Let G be a finite Abelian group and let l be a positive integer. For any l elements, $g_1, ..., g_l \in G$, consider the statistical distance between the uniform distribution on G and the distribution given by the sum of a random subset of $g_1, ..., g_l$. Then, the expectation of this statistical distance over a uniform choice of $g_1, ..., g_l \in G$ is at most $\sqrt{|G|/2^l}$.*

In particular, the probability that this statistical distance is more than $\sqrt[4]{|G|/2^l}$ is at most $\sqrt[4]{|G|/2^l}$.

The hiding property of Kawachi et al.'s lattice-based commitment scheme [8] was proved from this lemma directly by identifying G as a set of possible columns in M. However, we cannot apply this lemma directly to our case since $ex(x)$ is not uniformly at random over $\{0,1\}^\ell$ and M is biased. Thus, we have to modify this lemma.

Theorem 5. *Let $k = 5m$, $5m < \ell \leq 10m$ and $G = \{0,1\}^m$. We choose ℓ elements, $g_1, \ldots, g_\ell \in \{0,1\}^m$ so that $g_{i,j} = 1$ with probability $\gamma \in (0, 1/2)$ independently for every i, j. For sufficiently large m, the expected statistical distance between $\sum_i g_i ex(x)_i$ and the uniform distribution over $\{0,1\}^m$ with respect to the random choice of $g_1, \ldots, g_\ell \in \{0,1\}^m$ is at most $\sqrt{m}(5/4)^{-5m} + 2^{-m}$. In particular, the probability that the statistical distance is more than $\sqrt{\sqrt{m}(5/4)^{-5m} + 2^{-m}}$ is at most $O(\sqrt{\sqrt{m}(5/4)^{-5m} + 2^{-m}})$.*

proof: Let $g = (g_1, \ldots, g_\ell)$. For $h \in G$, we define

$$P_g(h) = \Pr_{x \in \{0,1\}^k}\left[\sum_{i=1}^{\ell} ex(x)_i g_i = h\right].$$

The expectation of the statistical distance with uniform distribution is

$$\begin{aligned}
&\operatorname*{Exp}_g\left[\frac{1}{2}\sum_{h \in \{0,1\}^m} |P_g(h) - 2^{-m}|\right] \\
&\leq \operatorname*{Exp}_g\left[\frac{2^{m/2}}{2}\left(\sum_{h \in \{0,1\}^m}\left(P_g(h) - 2^{-m}\right)^2\right)^{1/2}\right] \\
&= \frac{2^{m/2}}{2}\operatorname*{Exp}_g\left[\left(\sum_{h \in \{0,1\}^m} P_g(h)^2 - 2^{-m}\right)^{1/2}\right] \\
&\leq \frac{2^{m/2}}{2}\left(\operatorname*{Exp}_g\left[\sum_h P_g(h)^2\right] - 2^{-m}\right)^{1/2}.
\end{aligned}$$

We want to evaluate $\operatorname*{Exp}_g\left[\sum_h P_g(h)^2\right]$. So, we first evaluate $\sum_h P_g(h)^2$.

$$\begin{aligned}
&\sum_h P_g(h)^2 = \Pr_{x,x' \in \{0,1\}^k}\left[\sum_{i=1}^{\ell} ex(x)_i g_i = \sum_{i=1}^{\ell} ex(x')_i g_i\right] \\
&= \Pr_{x,x' \in \{0,1\}^k}\left[\sum_{i=1}^{\ell} ex(x)_i g_i = \sum_{i=1}^{\ell} ex(x')_i g_i \wedge ex(x) = ex(x')\right] \\
&+ \Pr_{x,x' \in \{0,1\}^k}\left[\sum_{i=1}^{\ell} ex(x)_i g_i = \sum_{i=1}^{\ell} ex(x')_i g_i \wedge ex(x) \neq ex(x')\right] \\
&\leq \frac{1}{2^k} \\
&+ \Pr_{x,x' \in \{0,1\}^k}\left[\sum_{i=1}^{\ell} ex(x)_i g_i = \sum_{i=1}^{\ell} ex(x')_i g_i \middle| ex(x) \neq ex(x')\right].
\end{aligned}$$

We can then evaluate the expectation

$$\Pr_{x,x' \in \{0,1\}^k}\left[\sum_{i=1}^{\ell} ex(x)_i g_i = \sum_{i=1}^{\ell} ex(x')_i g_i \middle| ex(x) \neq ex(x')\right]$$

as below.

$$\begin{aligned}
&\operatorname*{Exp}_{x \neq x'}\left[\Pr_g\left[\sum_{i=1}^{\ell} ex(x)_i \cdot g_i = \sum_{i=1}^{\ell} ex(x')_i \cdot g_i\right]\right] \\
&= \sum_{x \neq x'} \frac{1}{2^\ell(2^\ell - 1)} \cdot \Pr\left[\sum_{i=1}^{\ell}(ex(x)_i - ex(x')_i)g_i = 0\right] \\
&= \sum_{d=1}^{\ell} \sum_{\Delta(x,x')=d} \frac{1}{2^\ell(2^\ell - 1)} \cdot \Pr\left[\sum_{i=1}^{\ell} g_i = 0\right] \\
&= \sum_{d=1}^{m} \sum_{wt(x)=d} \frac{1}{(2^\ell - 1)} \cdot \Pr\left[\sum_{i=1}^{\ell} g_i = 0\right] \\
&= \Pr\left[\sum_{i=1}^{\ell} g_i = 0\right].
\end{aligned}$$

Since $g_{i,j}$ is independently set to 0 with probability $1-\gamma$ for every i, j, we have

$$\Pr\left[\sum_{i=1}^{\ell} g_{i,j} = 0\right] = \frac{1}{2} + \frac{\{(1-2\gamma)\}^\ell}{2}.$$

Therefore, it holds that

$$\Pr\left[\sum_{i=1}^{\ell} g_i = 0\right] = \left(\frac{1}{2} + \frac{\{(1-2\gamma)\}^\ell}{2}\right)^m.$$

Hence, the expected statistical distance is

$$\operatorname*{Exp}_g\left[\frac{1}{2}\sum_{h \in \{0,1\}^m} |P_g(h) - 2^{-m}|\right]$$

$$\leq 2^{m/2-1}\left\{1/2^k + \left(\frac{1}{2} + \frac{\{(1-2\gamma)\}^\ell}{2}\right)^m - 1/2^m\right\}^{1/2}.$$

Since $k = 5m$ and $5m < \ell \leq 10m$, this expectation is bounded by $\sqrt{m}(5/4)^{-5m} + 2^{-m}$ above for sufficiently large m. In particular, From Markov's inequality, the probability that this statistical distance is more than $\sqrt{\sqrt{m}(5/4)^{-5m} + 2^{-m}}$ is at most $\sqrt{\sqrt{m}(5/4)^{-5m} + 2^{-m}}$.
▮

From Theorem 5 and the hybrid argument, we can immediately prove the statistically hiding property of our commitment scheme as follows.

Theorem 6. *Let $Com_M(\mathsf{a})$ be the probability distribution of a commitment string in our commitment scheme* $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$ *for a matrix M and message string* a. *Then, the statistical distance between $Com_M(\mathsf{a})$ and $Com_M(\mathsf{a})$is negligible with probability exponentially close to* 1 *with respect to a random choice of M.*

proof:

Using the triangle inequality, we obtain

$$d(Com_M(\mathsf{a}), Com_M(\mathsf{a}'))$$
$$\leq d(U, Com_M(\mathsf{a})) + d(U, Com_M(\mathsf{a}'))$$

for every message, a and a', where U is the uniform distribution. From Theorem 5, $d(U, Com_M(\mathsf{a}))$ and $d(U, Com_M(\mathsf{a}))$ is negligible with probability exponentially close to 1, and thus, so is $d(Com_M(\mathsf{a}), Com_M(\mathsf{a}'))$.
▮

From Theorem 6, we can see that the statistically hiding property holds except exponentially small probability with respect to the random choice of M.

V. Conclusion

We have proposed the commitment scheme, $\mathsf{Comm}(\mathsf{S}_{\mathsf{END}}, \mathsf{R}_{\mathsf{ECEIVE}})$, that satisfies:

- the expected output locality which is strictly less than input length;
- the computationally binding property; and
- the statistically hiding property.

Importantly, this is the first commitment scheme that satisfies non-trivially low output locality.

ACKNOWLEDGEMENT

This work is partially supported by Microsoft Research Asia, CREST(JPMJCR1404) at Japan Science and Technology Agency, Project for Establishing a Nationwide Practical Education Network for IT Human Resources Development, Education Network for Practical Information Technologies, and Innovation Platform for Society 5.0 at MEXT, JSPS Grant-in-Aid for Scientific Research (A) No. 16H01705, (B) No. 17H01695, JSPS Grant-in-Aid for Young Scientists (B) No. 17K12640.

References

[1] Benny Applebaum. Garbled circuits as randomized encodings of functions: a primer. *IACR Cryptology ePrint Archive*, 2017:385, 2017.

[2] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-complexity cryptographic hash functions. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 7:1–7:31, 2017.

[3] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in nc^0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[4] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam D. Smith. Efficient and non-interactive non-malleable commitment. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, pages 40–59, 2001.

[5] Ivan Damgard. Commitment schemes and zero-knowledge protocols. In *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, pages 63–86, 1998.

[6] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

[7] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, pages 201–215, 1996.

[8] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, pages 372–389, 2008.

[9] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

[10] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, pages 129–140, 1991.

[11] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.