論文

Improvement of the Round Complexity of Perfectly Concealing Bit Commitment Schemes

Takeshi Koshiba* Yoshiharu Seri

We improve the upper bound on the round complexity for perfectly concealing bit commitment schemes based on the general computational assumption. The best known scheme is the one-way permutation based scheme due to Naor, Ostrovsky, Venkatesan and Yung and its round complexity is $O(n)$. We consider a naive parallel version of their scheme of the multiplicity $\log n$ and obtain an $O(n/\log n)$-round scheme. Our improvement answers a question, raised by them, whether their $O(n)$-round scheme is essential with respect to the round complexity. Though such a parallelization raises an analytic difficulty, we introduce a new analysis technique and then overcome the difficulty. Our technique copes with *expected almost* pairwise independent random variables instead of the pairwise independence, which is a key property in their analysis. While the expected almost pairwise independence plays an important role in our security proof, it also provides alternative security proof for the original scheme.

Keywords: bit commitment, computational binding, one-way permutation, perfect concealing, round complexity, zero-knowledge argument

1 Introduction

A notion of one-way functions is one of the most fundamental notions in cryptology. Constructions of cryptographic protocols from cryptographic primitives such as one-way functions help us to develop the foundations of cryptography. Especially, reducing complexity assumptions or requirements for cryptographic primitives leads us to essentially understand the nature of cryptography.

A construction of pseudorandom generators from any one-way functions [6] is one of the most important results in the foundations of cryptography, because pseudorandom generators are still primitive for other cryptographic protocols. Digital signature schemes are also constructible from one-way functions [9, 10]. Besides one-way functions, bit commitment schemes are building blocks for cryptographic protocols and (non-uniform) computationally concealing statistically binding schemes are built in zero-knowledge proof systems, introduced in [5], for any NP language [3]. Furthermore, Naor, in [7], showed that computationally concealing statistically binding bit commitment schemes are constructible from pseudorandom generators (i.e., from one-way functions). Another type of bit commitment scheme, say statistically concealing computationally binding scheme, can be utilized in zero-knowledge arguments [1].

A bit commitment scheme is a two-party protocol and there are messages to be exchanged between the two parties. Since a bit commitment scheme is a cryptographic primitive, it is desirable to be efficient in several points (e.g., the total size of messages exchanged during the protocol, or the round of communications in the protocol). In this paper, we focus on the round complexity of bit commitment schemes based on general computational assumptions. Naor's computationally concealing statistically binding scheme [7] is of constant round. (Precisely speaking, his scheme is of one round in the commit phase and of one round in the reveal phase.) On the other hand, the $O(n)$-round one-way permutation based scheme by Naor, Ostrovsky, Venkatesan and Yung [8] is the most round-efficient of all known statistical concealing computationally binding protocols based on general computational assumption. (If we allow parties the quantum computational powers, the quantum one-way permutation based scheme by Dumais, Mayers and Salvail [2] is of constant round.)

In this paper, we propose a perfectly concealing computationally binding bit commitment scheme of better round complexity. We consider a naive parallel version of Naor-Ostrovsky-Venkatesan-Yung scheme [8] of the multiplicity $\log n$ and obtain an $O(n/\log n)$-round scheme. Though such a parallelization raises an analytic difficulty, we introduce a new analysis technique and then overcome the

*Division of Mathematics, Electronics and Informatics, Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama 338-8570, Japan.

（原稿受付日：平成18年 4月 7日）

difficulty. Our technique copes with *expected almost* pairwise independent random variables instead of the pairwise independence, which is a key property in their analysis. While the expected almost pairwise independence plays an important role in our security proof, it also provides alternative security proof for the original scheme. Our improvement answers a question, raised in [8], whether their $O(n)$-round scheme is essential with respect to the round complexity.

2 Preliminaries

Notations and Conventions

We introduce some useful notations and conventions. For any pair $u, v \in \{0,1\}^n$ of strings, let $\langle u, v \rangle = \sum u_i v_i \bmod 2$, where $u = u_1 \cdots u_n$ and $v = v_1 \cdots v_n$. Since n-bit strings can be regarded as n-dimensional vectors over GF(2), $\langle u, v \rangle$ corresponds to the inner product of n-dimensional vectors u and v. For an m-tuple $\hat{h} = (h_1, h_2, \ldots, h_m)$ of n-bit strings and an n-bit string y, $\langle\!\langle \hat{h}; y \rangle\!\rangle$ denotes the m-bit string $s = s_1 s_2 \cdots s_m$ where the i-th bit $s_i = \langle h_i, y \rangle$ for all $1 \le i \le m$. Note that an equation $\langle\!\langle \hat{h}; y \rangle\!\rangle = s$ can be regarded as m simultaneous equations with n unknown variables. By $a \in_R A$ we denote the element a is randomly chosen from the set A.

Bit Commitment Scheme

A bit commitment scheme is a two-party protocol. The protocol consists of two phases: commit phase and reveal phase. In the commit phase, the sender, say Sam, has a bit b in his private space and he wants to commit b to the receiver, say Rachel. They exchange messages and at the end of the commit phase Rachel gets some information that represents b. In the reveal phase, Sam confides b to Rachel by exchanging messages. At the end of the reveal phase, Rachel judges whether the information gotten in the reveal phase really represents b or not.

In this paper, we consider bit commitment schemes satisfying the perfectly concealing property and the computationally binding property. A formal definition follows.

Definition 2.1 A *perfectly concealing computationally binding bit commitment scheme* must satisfy the following four conditions.

(Correctness) If both parties are honest, then for any bit $b \in \{0,1\}$ the sender has, the receiver accepts with certainty.

(Efficiency) Both parties must obey some probabilistic polynomial-time algorithms.

(Perfect Concealing) Even if a receiver $\mathcal{R}$ is dishonest and computationally unbounded, the distributions of the conversation between $\mathcal{R}$ and an honest sender in case $b = 0$ and $b = 1$ are identical.

(Computational Binding) The probability that any probabilistic polynomial time sender $\mathcal{S}$ can reveal two different values of b is negligible, where the probability is over the internal coin tosses of both parties.

Remark. Polynomials in the above definitions are with respect to the security parameter.

One-Way Permutation

Let f be a function from $\{0,1\}^*$ to $\{0,1\}^*$. If f is 1-to-1 and length-preserving then f is said to be a *permutation*. Furthermore, if f is polynomial-time computable and hard to invert then f is said to be *one-way*. A formal definition of one-way permutation follows.

Definition 2.2 Let f be a permutation. If f satisfies the following two conditions then f is said to be *one-way*:

- f is polynomial-time computable;
- for every probabilistic polynomial-time algorithm $\mathcal{A}$, for every polynomial p and for sufficiently large n,

$$\Pr[\mathcal{A}(f(U_n)) = U_n] \le \frac{1}{p(n)},$$

  where the probability above is over internal coin tosses in $\mathcal{A}$ and the uniform distribution U_n on $\{0,1\}^n$.

3 New Scheme

In this section, we show our perfectly concealing bit commitment scheme based on one-way permutation. As mentioned, our scheme is a parallel version of the Naor-Ostrovsky-Venkatesan-Yung scheme [8].

Let n be the *security parameter* and $m = \lfloor \log n \rfloor$ be the *multiplicity parameter*. Let f be a one-way permutation. We assume that m divides $n-1$ and let $r = (n-1)/m$. (This assumption is not really essential, and is only made for convenience.) Let $H^{[i]} = \{0^{i-1}1w \mid w \in \{0,1\}^{n-i}\}$ for any i such that $1 \le i \le n$ and $\mathcal{H}^{(j)} = H^{[(j-1)m+1]} \times H^{[(j-1)m+2]} \times \cdots \times H^{[jm]}$ for any j such that $1 \le j \le r$. Let $H_\ell^{[i]} = \{0^{i-1}1\ell w \mid w \in \{0,1\}^{n-i-|\ell|}\}$ for any $\ell \in \{0,1\}^*$ and for any i. Let $\mathcal{H}_{\hat{\ell}}^{(j)} = H_{\ell_1}^{[(j-1)m+1]} \times H_{\ell_2}^{[(j-1)m+2]} \times \cdots \times H_{\ell_m}^{[jm]}$ for any $\hat{\ell} = (\ell_1, \ell_2, ..., \ell_m) \in (\{0,1\}^*)^m$ and for any j.

Now we are ready to describe our scheme.

[Commit Phase]

step 1. A sender Sam chooses $x \in \{0,1\}^n$ randomly and computes $y = f(x)$. Let b be a bit to be committed to a receiver Rachel.

step 2. For k from 1 to r,

- Rachel chooses an m-tuple of n-bit strings
$$\hat{h}^{(k)} = (h_{(k-1)m+1}, h_{(k-1)m+2}, \cdots, h_{km}) \in_R \mathcal{H}^{(k)}$$
randomly and sends $\hat{h}^{(k)}$ to Sam.
- Sam computes
$$\hat{c}^{(k)} = (c_{(k-1)m+1}, c_{(k-1)m+2}, \cdots, c_{km}) = \langle\!\langle \hat{h}^{(k)}; y \rangle\!\rangle$$
and sends $\hat{c}^{(k)}$ to Rachel.

step 3. Sam solves the linear equation system
$$c_i = \langle h_i, z \rangle \quad \text{for all } i \text{ such that } 1 \le i \le n-1$$
and obtains two solutions z_0 and z_1, where z_0 is lexicographically smaller than z_1. (Recall that z can be identified with n variables over GF(2) and the above equations are defined over GF(2).) Since either z_0 or z_1 is equal to y, let d be a bit such that $z_d = y$. Sam sends $e = b \oplus d$ to Rachel.

step 4. Rachel also solves the same linear equation system and obtains the same two solutions z_0 and z_1.

[Reveal Phase]

step 5. Sam sends b and x to Rachel.

step 6. Rachel computes $y' = f(x)$ and verifies that $c_i = \langle h_i, y' \rangle$ for all i such that $1 \le i \le n-1$. If $y' = y_d$ where $d = b \oplus e$, then Rachel accepts.

Theorem 3.1 *The above $O(n/\log n)$-round bit commitment protocol satisfies the perfectly concealing and the computationally binding conditions.*

Besides the concealing and binding conditions, the correctness and the efficiency of our protocol follow from the construction. We will give a proof for the perfectly concealing condition in Section 5.1 and one for the computationally binding condition in Section 5.2. As a typical consequence, Theorem 3.1 improves the round complexity of the perfect zero-knowledge arguments for any language in NP assuming the existence of one-way permutations.

4 Expected Almost Pairwise Independence

Before analyzing our protocol, we mention a new technique for the analysis. The *pairwise independence* of random variables is a commonly used tool to analyze probabilistic behavior of algorithms and protocols. The *almost pairwise independence* is a relaxed notion of the pairwise independence is also used in cryptography, especially in universal hash functions. In this section, we introduce yet another relaxed notion of the pairwise independence.

Let $X_1, X_2, \ldots, X_n$ be binary random variables such that $\Pr[X_i = 1] = p$ for all i. Then, $\mathrm{E}[X_i] = p$ and $\mathrm{Var}[X_i] = p - p^2$. If
$$\Pr[X_i = b \wedge X_j = b'] \le \Pr[X_i = b]\Pr[X_j = b'] + \epsilon$$
for any $b, b' \in \{0,1\}$ and for any pair (i,j) such that $i \neq j$ then $X_1, X_2, \ldots, X_n$ are said to be ϵ-*almost pairwise independent*. It is easy to see that the almost pairwise independence is defined for the "worst" pair. We consider how the value $\Pr[X_i = b \wedge X_j = b']$ "on average" is apart from the value $\Pr[X_i = b] \cdot \Pr[X_j = b']$. If
$$\binom{n}{2}^{-1} \sum_{i \neq j} \Pr[X_i = b \wedge X_j = b'] \le \Pr[X = b]\Pr[X = b'] + \epsilon$$
for any $b, b' \in \{0,1\}$ then $X_1, X_2, \ldots, X_n$ are said to be ϵ-*expected almost pairwise independent*, where X is a binary random variable such that $\Pr[X = 1] = p$.

From the definitions, ϵ-almost pairwise independence implies ϵ-expected almost pairwise independence. On the other hand, the converse does not hold in general. For some applications the expected almost pairwise independence may be sufficient. Actually in this paper, the expected almost pairwise independence plays a key role for the security analysis of our new scheme.

5 Security Analysis

In this section, we give a security proof for our scheme. First, we will show that our scheme satisfies the perfectly concealing condition. Next, we will show that our scheme satisfies the computationally binding condition. Both proof structures are similar to the proofs in [8] except the utilization of our new technique discussed in the previous section. The technique of the expected almost pairwise independence essentially contributes to the reduction of the round complexity.

5.1 Perfect Concealing

Lemma 5.1 (Perfect Concealing) *For any cheating receiver $\mathcal{R}$, the distribution of the conversation between the honest sender and $\mathcal{R}$ in the commit phase is independent of the value of the bit b.*

Proof. Let $\hat{h}^{(k)} = (h_{(k-1)m+1}, h_{(k-1)m+2}, \cdots, h_{km}) \in \mathcal{H}^{(k)}$ be a (malicious) choice in the k-th response of the cheating receiver $\mathcal{R}$. $\mathcal{R}$'s choice may depend on $h_1, h_2, \ldots, h_{(k-1)m}$ and $c_1, c_2, \ldots, c_{(k-1)m}$. However, the conditional distribution of y given $h_1, h_2, \ldots, h_{(k-1)m}$ and $c_1, c_2, \ldots, c_{(k-1)m}$ is still uniform because of their shapes. Moreover, since $h_1, h_2, \ldots, h_{km}$ are independent, we have for any $\hat{b} \in \{0,1\}^m$

$$\Pr[\langle\!\langle \hat{h}^{(k)}; y \rangle\!\rangle = \hat{b}] = \frac{1}{2^m}.$$

Furthermore, since the value of d is uniformly distributed over $\{0,1\}$, then conversations $(h_1, \ldots, h_{n-1}, c_1, \ldots, c_{n-1}, e)$ between the honest sender and $\mathcal{R}$ in the commit phase is independent of the value of b. □

5.2 Computational Binding

Lemma 5.2 (Computational Binding) *Suppose that $\mathcal{S}$ is a probabilistic polynomial-time cheating sender that follows the protocol in the commit phase. Also suppose that $\mathcal{S}$ can reveal to an honest receiver two different values of b with non-negligible probability $\epsilon = \epsilon(n)$, where the probability is over the internal coin tosses of $\mathcal{S}$ and the honest receiver. Then, there exists a probabilistic polynomial-time inverter for f that violates the one-wayness of f.*

We devote the rest of this subsection to the proof of Lemma 5.2. To prove Lemma 5.2, we construct a probabilistic polynomial-time inverter $\mathcal{A}$ for f by mimicking a honest receiver $\mathcal{R}$ which interacts with the cheating sender $\mathcal{S}$.

First of all, without loss of generality, we can assume that $\mathcal{S}$ is deterministic. The following standard argument justifies this assumption. Suppose that $\mathcal{S}$ succeeds (i.e., reveals two different values of b) with non-negligible probability ϵ and the probability is over the internal coin tosses of $\mathcal{S}$ and $\mathcal{R}$ (i.e., messages (or, queries) from $\mathcal{R}$). By a counting argument, the fraction of internal coin tosses with which $\mathcal{S}$ succeeds on at least $\epsilon/2$ of $\mathcal{R}$'s queries is $\epsilon/2$ at least. If we prepare sufficiently many (i.e., $2n/\epsilon$) random assignments for internal coin tosses for $\mathcal{A}$, then there exists one of the $2n/\epsilon$ assignments with which $\mathcal{S}$ succeeds on at least $\epsilon/2$ of $\mathcal{R}$'s queries with overwhelming probability.

Next, it is convenient to represent the cheating strategy of $\mathcal{S}$ in the commit phase, depending on queries from $\mathcal{R}$, as a rooted tree T of depth r. A node U_k at the k-th level is defined by queries $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(k-1)}$, where each $\hat{h}^{(j)}$ is in $\mathcal{H}^{(j)}$ for $1 \leq j \leq k-1$. Every node at the k-th level has $|\mathcal{H}^{(k)}|$ outgoing edges. Each of the outgoing edges corresponds to some queries (i.e., an m-tuple of n-bit strings) $\hat{h}^{(k)}$ in $\mathcal{H}^{(k)}$ and leads to a different node at the $(k+1)$th level. The cheating strategy of $\mathcal{S}$ specifies a labeling of the edges of T with an m-bit string w. For a node U_k defined by queries $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(k-1)}$, the label w on an edge $\hat{h}^{(k)}$ is the response $\hat{c}^{(k)}$ of $\mathcal{S}$ to the queries $\hat{h}^{(k)}$ in the k-th round, on condition that the previous queries were $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(k-1)}$. We denote it by $L_{\mathcal{S}}(U_k, \hat{h}^{(k)})$. Since $\mathcal{S}$ can be regarded as a deterministic polynomial-time algorithm, the inverter algorithm $\mathcal{A}$ can completely control the behavior of $\mathcal{S}$. It implies that $\mathcal{A}$ can compute the labeling $L_{\mathcal{S}}$. (Note that $\mathcal{A}$ does not have the tree in the memory.) For a leaf node U_{r+1} defined by $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(r)}$, let $U_1, U_2, \ldots, U_r$ be nodes on the path from the root node to the leaf node U_{r+1} and let $\{y_0(U_{r+1}), y_1(U_{r+1})\}$ be the set of images consistent with the labeling of $\mathcal{S}$. Namely, for all $1 \leq j \leq r$ and for all $b \in \{0,1\}$,

$$L_{\mathcal{S}}(U_j, \hat{h}^{(j)}) = \langle\!\langle \hat{h}^{(j)}; y_b \rangle\!\rangle.$$

The leaf node U_{r+1}, defined by $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(r)}$, is said to be *good* if $\mathcal{S}$ can reveal two different values of b when $\mathcal{R}$'s queries are $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(r)}$. This implies that $\mathcal{S}$ can invert both $y_0(U_{r+1})$ and $y_1(U_{r+1})$, which we will see later.

The basic strategy of the inverter $\mathcal{A}$ is to try to find a good leaf by random sampling. In order to compute, given $y \in \{0,1\}^n$, the preimage of y, $\mathcal{A}$ must take a path to a good leaf U_{r+1} such that $y \in \{y_0(U_{r+1}), y_1(U_{r+1})\}$. To take the path, $\mathcal{A}$ must choose queries $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(r)}$ such that

$$L_{\mathcal{S}}(U_j, \hat{h}^{(j)}) = \langle\!\langle \hat{h}^{(j)}, y \rangle\!\rangle$$

for all $1 \leq j \leq r$. If $\mathcal{S}$ is honest, a simple analysis of the random sampling is sufficient to prove the security. However, since $\mathcal{S}$ does not have to honestly respond to the queries from $\mathcal{R}$, that makes the security analysis harder.

Inverting Algorithm

Now, let us describe the inverter algorithm $\mathcal{A}$. $\mathcal{A}$ is given a random string $y \in \{0,1\}^n$ and tries to compute the preimage of y. Fix $t = r - 8(\log(n/\epsilon) + m + 1)/m = r - 8\log(2n^2/\epsilon)/\log n$.

step 1. $\mathcal{A}$ makes $\mathcal{S}$ perform step 1 in the commit phase. (Though $\mathcal{S}$ chooses a random element x' and computes $y' = f(x')$, these values are not important.)

step 2. For k from 1 to t

- **2.1.** $\mathcal{A}$ chooses $\hat{h}^{(k)} \in \mathcal{H}^{(k)}$ randomly.
- **2.2.** $\mathcal{A}$ sends $\hat{h}^{(k)}$ to $\mathcal{S}$ and obtains $\hat{c}^{(k)}$ from $\mathcal{S}$.
- **2.3.** Unless $\hat{c}^{(k)} = \langle\!\langle \hat{h}^{(k)}; y \rangle\!\rangle$ then $\mathcal{A}$ rewinds $\mathcal{S}$ to the state before its reply and restarts from step 2.1.

step 3. If $\mathcal{A}$ reaches the $(t+1)$-th level, it chooses the remaining $n-tm-1$ queries $h_{tm+1}, h_{tm+2}, \ldots, h_{n-1}$ uniformly at random.

step 4. $\mathcal{A}$ checks whether the path to the leaf is labeled consistently with $\langle h_{tm+1}, y\rangle, \langle h_{tm+2}, y\rangle, \ldots, \langle h_{n-1}, y\rangle$. If this is the case and the leaf is good, then $\mathcal{A}$ makes $\mathcal{S}$ reveal two different values of b. At the same time, $\mathcal{A}$ obtains both x' and the preimage x of y from $\mathcal{S}$ and outputs x. Otherwise $\mathcal{A}$ aborts.

In what follows, we analyze the success probability and the time complexity of $\mathcal{A}$. We introduce several notations only for the analysis.

Notation

Let U_k be a node at the k-th level of the tree T defined by $\hat{h}^{(1)}, \hat{h}^{(2)}, \ldots, \hat{h}^{(k-1)}$. Let $\hat{c}^{(1)}, \hat{c}^{(2)}, \ldots, \hat{c}^{(k-1)}$ be labels assigned to the path from the root to U_k. We say that $y \in \{0,1\}$ is an *image* in U_k if $\langle\langle \hat{h}^{(j)}; y\rangle\rangle = \hat{c}^{(j)}$ for any $1 \leq j \leq k-1$. $\mathcal{I}(U_k)$ denotes the set of images in U_k. Note that $|\mathcal{I}(U_k)| = 2^{n-m(k-1)}$ for any k. We say that $\hat{h}^{(k)} \in \mathcal{H}^{(k)}$ is a *query m-tuple* of U_k. Note that there are $2^{m(n-k)-m(m-1)/2}$ query m-tuples from any node U_k at the k-the level. Let $A(U_k, y) = |\{\hat{h}^{(k)} \in \mathcal{H}^{(k)} : L_{\mathcal{S}}(U_k, \hat{h}^{(k)}) = \langle\langle h^{(k)}; y\rangle\rangle\}|$. We say that an image y is *balanced in U_k* at the k-th level if

$$\frac{1}{2^m}\left(1-\frac{1}{n}\right) \leq \frac{A(U_k, y)}{2^{m(n-k)-\frac{1}{2}m(m-1)}} \leq \frac{1}{2^m}\left(1+\frac{1}{n}\right).$$

We say that an image y is *fully balanced in U_k* of the kth level if it is balanced in all the ancestor nodes of U_k. Let $\mathcal{F}(U_k)$ be the set of images in $\mathcal{I}(U_k)$ that are fully balanced in U_k. For a set $\mathcal{H}$ of query m-tuples from a node U and $y \in \mathcal{I}(U)$, the *discrepancy* of y at $\mathcal{H}$ is defined as the difference between the expected number of agreeing query m-tuples and the actual number of query m-tuples in $\mathcal{H}$ that agree with y. Formally, it is defined by

$$\left|\left|\{\hat{h} \in \mathcal{H} : L_{\mathcal{S}}(U, y) = \langle\langle \hat{h}; y\rangle\rangle\}\right| - \frac{1}{2^m}|\mathcal{H}|\right|.$$

Analysis

Claim 1 *Let U_k be a node at the k-th level. For any $\hat{\ell} = (\ell_1, \ell_2, \ldots, \ell_m) \in \{0,1\}^{(t-k+1)m} \times \{0,1\}^{(t-k+1)m-1} \times \cdots \times \{0,1\}^{(t-k)m+1}$ and for any $\hat{h}^{(k)} = (h_{(k-1)m+1}, h_{(k-1)m+2}, \ldots, h_{km}) \in \mathcal{H}_{\hat{\ell}}^{(k)}$, let $a_{\hat{h}^{(k)}}$ be a binary random variable over $z \in_R \mathcal{I}(U_k)$ such that $a_{\hat{h}^{(k)}} = 1$ if $L_{\mathcal{S}}(U_k, z) = \langle\langle \hat{h}^{(k)}; z\rangle\rangle$ and 0 otherwise. Then*

$$\Pr\left[\left|\sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}} - \frac{1}{2^m}\left|\mathcal{H}_{\hat{\ell}}^{(k)}\right|\right| \geq 2^{(m-\frac{1}{8})(n-tm-1)}\right] \leq m \cdot 2^{-\frac{3}{4}(n-tm-1)}.$$

From the technical point of view, the following proof is a main contribution in this paper. Though the statement itself is as simple as Claim 1 in [8], the proof is not simpler. As mentioned, we devise a new technique to show the above claim. The expected almost pairwise independence plays a key role in the following proof. (Precisely speaking, in the proof, we show a relaxed property of the expected almost pairwise independence. Namely, we take care of the probability $\Pr[a_{\hat{h}^{(k)}} = b \wedge a_{\hat{h'}^{(k)}} = b']$ only for the case $b = b' = 1$.)

Proof. For any $\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}$, we have

$$\mathrm{E}[a_{\hat{h}^{(k)}}] = \frac{1}{2^m} \quad \text{and} \quad \mathrm{Var}[a_{\hat{h}^{(k)}}] = \frac{2^m-1}{2^{2m}}. \tag{1}$$

Now, we are interested in the upper bound on

$$\begin{aligned}\mathrm{Var}\left[\sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}}\right] &= \sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} \mathrm{Var}[a_{\hat{h}^{(k)}}] \\ &\quad + \sum_{\hat{h}^{(k)}, \hat{h'}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} \mathrm{Cov}[a_{\hat{h}^{(k)}}, a_{\hat{h'}^{(k)}}].\end{aligned} \tag{2}$$

A crucial observation is that, for almost all pairs of query m-tuples, their covariances are zero. Since

$$\mathrm{Cov}[a_{\hat{h}^{(k)}}, a_{\hat{h'}^{(k)}}] \leq \frac{2^m-1}{2^{2m}}, \tag{3}$$

we will give an upper bound on the variance in eq.(2) by estimating the number of pairs having non-zero covariance. Note that the covariance of a pair $(a_{\hat{h}^{(k)}}, a_{\hat{h'}^{(k)}})$ is non-zero if and only if the random variables $a_{\hat{h}^{(k)}}$ and $a_{\hat{h'}^{(k)}}$ are independent. Thus, we consider the correlation between them more precisely. Let

$$\begin{aligned}\hat{h}^{(k)} &= (h_{(k-1)m+1}, h_{(k-1)m+2}, \ldots, h_{km}) \quad \text{and} \\ \hat{h'}^{(k)} &= (h'_{(k-1)m+1}, h'_{(k-1)m+2}, \ldots, h'_{km}).\end{aligned}$$

If $h_{(k-1)m+1}, h_{(k-1)m+2}, \ldots, h_{km}, h'_{(k-1)m+1}, h'_{(k-1)m+2}, \ldots, h'_{km}$ are linearly independent as vectors, then the random variables $a_{\hat{h}^{(k)}}$ and $a_{\hat{h'}^{(k)}}$ are independent. We know that vectors in $\hat{h}^{(k)}$ are linearly independent and so are vectors in $\hat{h'}^{(k)}$. Another observation is that the following statements are equivalent.

- $h_{(k-1)m+1}, h_{(k-1)m+2}, \ldots, h_{km}, h'_{(k-1)m+1}, h'_{(k-1)m+2}, \ldots, h'_{km}$ are linearly independent.
- $h_{(k-1)m+1}, h_{(k-1)m+2}, \ldots, h_{km}, v_1, v_2, \ldots, v_m$ are linearly independent, where $v_i = h_{(k-1)m+i} \oplus h'_{(k-1)m+i} \in \{0^{tm+1}w : w \in \{0,1\}^{n-tm-1}\}$ for all $1 \leq i \leq m$.

So, we can bound from below the number V of assignments for which the $2m$ vectors are linearly independent as follows:

$$V \geq (2^{n-tm-1})^m \cdot \prod_{i=1}^{m} (2^{n-tm-1} - 1 - 2^{i-1}). \qquad (4)$$

The factor $(2^{n-tm-1})^m$ in eq.(4) comes from that every assignment for $h_{(k-1)m+1}, h_{(k-1)m+2}, \ldots, h_{km}$ makes them linearly independent. The term "-1" in eq.(4) means the exclusion of the case $v_i = 0^n$ and the term "-2^{i-1}" means the exclusion of all the vectors in $\mathit{span}(v_1, v_2, \ldots, v_{i-1})$. Thus, the number W of assignments for which the $2m$ vectors are not linearly independent is upper-bounded as follows:

$$\begin{aligned} W &\leq (2^{n-tm-1})^{2m} - (2^{n-tm-1})^m \cdot \prod_{i=1}^{m} (2^{n-tm-1} - 1 - 2^{i-1}) \\ &\leq (1 + 2^{m-1})(2^{n-tm-1})^{2m-1}. \qquad (5) \end{aligned}$$

By putting equations (1), (2), (3) and (5) together, we have

$$\begin{aligned} \mathrm{Var}\left[\sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}} \right] &\leq \frac{2^m - 1}{2^{2m}} \Big((2^{n-tm-1})^m \\ &\quad + (1 + 2^{m-1})(2^{n-tm-1})^{2m-1} \Big) \\ &\leq m \cdot (2^{n-tm-1})^{2m-1}. \end{aligned}$$

By Chebyschev's inequality,

$$\begin{aligned} \Pr\Bigg[\Bigg| \sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}} - \mathrm{E}\left[\sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}} \right] \Bigg| \\ \geq \lambda \sqrt{\mathrm{Var}\left[\sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}} \right]} \Bigg] \leq \frac{1}{\lambda^2}. \end{aligned}$$

Taking $\lambda = m^{-\frac{1}{2}} \cdot 2^{\frac{3}{8}(n-tm-1)}$ we get the assertion of the claim. □

Though the following claim corresponds to Claim 2 in [8], we can simplify the proof by eliminating the discussion on the pairwise independence.

Claim 2 *For any node U_{t+1} at the $(t+1)$-th level and for random $z \in_R \mathcal{I}(U_{t+1})$, we have $\Pr[z \in \mathcal{F}(U_{t+1})] \geq 1 - \gamma$ where $\gamma = n2^{-\frac{5}{8}(n-tm-1)}$.*

Proof. Let $U_1, U_2, \ldots, U_t$ be the nodes on the path to U_{t+1}. We show that for any U_i along the path almost $z \in \mathcal{I}(U_{t+1})$ are balanced. By Claim 1, we have for any $\hat{\ell} = (\ell_1, \ell_2, \ldots, \ell_m) \in \{0,1\}^{(t-k+1)m} \times \{0,1\}^{(t-k+1)m-1} \times \cdots \times \{0,1\}^{(t-k)m+1}$

$$\begin{aligned} \Pr\left[\left| \sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}} - \frac{1}{2^m} \left| \mathcal{H}_{\hat{\ell}}^{(k)} \right| \right| \geq 2^{(m-\frac{1}{8})(n-tm-1)} \right] \\ \leq m \cdot 2^{-\frac{3}{4}(n-tm-1)}. \end{aligned}$$

Let $b_{\hat{\ell}} = 1$ if

$$\left| \sum_{\hat{h}^{(k)} \in \mathcal{H}_{\hat{\ell}}^{(k)}} a_{\hat{h}^{(k)}} - \frac{1}{2^m} \left| \mathcal{H}_{\hat{\ell}}^{(k)} \right| \right| \geq 2^{(m-\frac{1}{8})(n-tm-1)}$$

and $b_{\hat{\ell}} = 0$ otherwise. By Markov's inequality, we have

$$\Pr\left[\sum_{\hat{\ell}} b_{\hat{\ell}} > \frac{2^{(t-k+1)m^2 - m(m-1)/2}}{2^{\frac{1}{8}(n-tm-1)}} \right] \leq m \cdot 2^{-\frac{5}{8}(n-tm-1)},$$

where the number of $\hat{\ell}$'s is $2^{(t-k+1)m^2 - m(m-1)/2}$. That is, the probability that, for more than a fraction $2^{-\frac{1}{8}(n-tm-1)}$ of the $\hat{\ell}$'s, the set $\mathcal{H}_{\hat{\ell}}^{(k)}$ has a discrepancy larger than $2^{(m-\frac{1}{8})(n-tm-1)}$ is at most $m2^{-\frac{5}{8}(n-tm-1)}$. Thus, with probability at least $1 - m \cdot 2^{-\frac{5}{8}(n-tm-1)}$ the total discrepancy at node U_k is at most

$$\begin{aligned} & 2^{-\frac{1}{8}(n-tm-1)} \cdot 2^{(t-k+1)m^2 - m(m-1)/2} \cdot 2^{m(n-tm-1)} \\ & + (1 - 2^{-\frac{1}{8}(n-tm-1)}) 2^{(t-k+1)m^2 - m(m-1)/2} 2^{(m-\frac{1}{8})(n-tm-1)} \\ & \quad \leq 2 \cdot 2^{(t-k+1)m^2 - m(m-1)/2} \cdot 2^{(m-\frac{1}{8})(n-tm-1)} \\ & \quad \leq 2^{\sum_{i=0}^{m-1}(n-(k-1)m-1-i)} \cdot 2^{-\frac{1}{8}(n-tm-1)+1}, \end{aligned}$$

where $|\mathcal{H}^{(k)}| = 2^{\sum_{i=0}^{m-1}(n-(k-1)m-1-i)}$, the first summand is an upper bound on the contribution of the $\mathcal{H}_{\hat{\ell}}^{(k)}$'s where $b_{\hat{\ell}} = 1$ and the second the contribution of the $\mathcal{H}_{\hat{\ell}}^{(k)}$'s where $b_{\hat{\ell}} = 0$. Hence for $z \in_R \mathcal{I}(U_{t+1})$ with probability at least $1 - m \cdot 2^{-\frac{5}{8}(n-tm-1)}$ we have

$$2^{-m} - 2^{-\frac{1}{8}(n-tm-1)+1} \leq \frac{A(U_k, z)}{|\mathcal{H}^{(k)}|} \leq 2^{-m} + 2^{-\frac{1}{8}(n-tm-1)+1}.$$

Since $t = r - 8(\log(n/\epsilon) + m + 1)/m$,

$$\begin{aligned} \frac{1}{2^m}\left(1 - \frac{1}{n}\right) &\leq 2^{-m} - 2^{-\frac{1}{8}(n-tm-1)+1} \leq \frac{A(U_k, z)}{|\mathcal{H}^{(k)}|} \\ &\leq 2^{-m} + 2^{-\frac{1}{8}(n-tm-1)+1} \leq \frac{1}{2^m}\left(1 + \frac{1}{n}\right). \end{aligned}$$

The probability that z is balanced in all the levels is therefore at least $1 - rm2^{-\frac{5}{8}(n-tm-1)} > 1 - n2^{-\frac{5}{8}(n-tm-1)}$. □

Claim 3 *For any node U_{t+1} at the $(t+1)$-th level and for any $z \in \mathcal{F}(U_{t+1})$,*

$$\begin{aligned} \frac{1}{2^n e^{1/m}} \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}} &\leq \Pr[\mathcal{A} \textit{ reaches } U_{t+1} \textit{ and } y = z] \\ &\leq \frac{e^{1/m}}{2^n} \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}}, \end{aligned}$$

where the probability is over the random choice of y and the internal coin tosses of $\mathcal{A}$.

Proof. The first inequality comes from the following.

$$
\begin{aligned}
&\Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y = z] \\
&= \frac{1}{2^n} \cdot \prod_{i=1}^{t} \frac{1}{A(U_i, z)} \\
&\geq \tfrac{1}{2^n\left(1+\frac{1}{n}\right)^{n/m}} \cdot \textstyle\prod_{i=1}^{tm} \frac{1}{2^{n-i-1}} \\
&\geq \frac{1}{2^n e^{1/m}} \cdot \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}}.
\end{aligned}
$$

The second inequality comes from the following.

$$
\begin{aligned}
&\Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y = z] \\
&= \frac{1}{2^n} \cdot \prod_{i=1}^{t} \frac{1}{A(U_i, z)} \\
&\leq \frac{1}{2^n\left(1-\frac{1}{n}\right)^{n/m}} \cdot \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}} \\
&\leq \frac{e^{1/m}}{2^n} \cdot \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}}.
\end{aligned}
$$

□

Recall that the goodness has been defined only for leaf nodes. Let us extend the notion of goodness for leaves to for internal nodes. We say that an internal node U is *good* if at least $\epsilon/4$ of the leaf nodes at the subtree rooted at U are good. From the assumption, the fraction of good leaves is at least $\epsilon/2$ and thus the fraction of good nodes among those of any fixed level is at least $\epsilon/4$, since all of them have the same number of leaves.

Claim 4 *The probability that $\mathcal{A}$ reaches some good node U_{t+1} at the $(t+1)$-th level and $y \in \mathcal{F}(U_{t+1})$ is at least $\epsilon(1-\gamma)/4e^{1/m}$ where the probability is over the random choice of y and the internal coin tosses of $\mathcal{A}$.*

Proof. Let U_{t+1} be a good node at the $(t+1)$-th level. Then

$$
\begin{aligned}
&\Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y \in \mathcal{F}(U_{t+1})] \\
&= \sum_{y \in \mathcal{F}(U_{t+1})} \Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y = z] \\
&\geq \sum_{y \in \mathcal{F}(U_{t+1})} \frac{1}{2^n e^{1/m}} \cdot \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}} \\
&\geq \frac{2^{n-tm}(1-\gamma)}{2^n e^{1/m}} \cdot \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}} \\
&= \frac{1-\gamma}{e} \cdot \prod_{i=1}^{tm} \frac{1}{2^{n-i}}.
\end{aligned}
$$

The first inequality follows from Claim 3 and the second from Claim 2. Since there are $\prod_{i=1}^{tm} 2^{n-i}$ nodes at the $(t+1)$-th level and at least $\frac{\epsilon}{4} \cdot \prod_{i=1}^{tm} 2^{n-i}$ nodes are good, the probability that the image chosen is fully balanced in a good node at the $(t+1)$-th level is at least $\epsilon(1-\gamma)/4e^{1/m}$. □

Claim 5 *In any good node U_{t+1} at the $(t+1)$-th level, the fraction of the good leaves at the subtree rooted in U_{t+1} that have at least one image in $\mathcal{F}(U_{t+1})$ is at least $\epsilon/8$.*

Proof. Any pair of images $y_1 \neq y_2$ in $\mathcal{I}(U_{t+1})$ can be together in at most $1/2^{n-tm-1}$ of the leaves of the subtree rooted at U_{t+1}. By Claim 2, there exists at most $\gamma 2^{n-tm}$ images in $\mathcal{I}(U_{t+1})$ that are not fully balanced in U_{t+1}. Therefore the fraction of the leaves of the subtree rooted in U_{t+1} where both of their images are from $\mathcal{I}(U_{t+1}) \setminus \mathcal{F}(U_{t+1})$ is bounded by

$$
\binom{\gamma 2^{n-tm}}{2} \cdot \frac{1}{2^{n-tm-1}}.
$$

Since

$$
\begin{aligned}
\binom{\gamma 2^{n-tm}}{2} \cdot \frac{1}{2^{n-tm-1}} &\leq 2\gamma^2 2^{n-tm-1} \\
&= n^2 2^{-\frac{1}{4}(n-tm-1)+1} \\
&= n^2 2^{-2(\log\frac{n}{\epsilon}+m+1)+1} \\
&\leq \frac{\epsilon^2}{2^{2m+1}} \\
&= \frac{\epsilon^2}{2n^2},
\end{aligned}
$$

we have that at least $\epsilon/4 - \epsilon^2/2n^2 \geq \epsilon/8$ of the leaves are both good and have at least one image in $\mathcal{F}(U_{t+1})$. □

Claim 6 *For any good node U_{t+1} at the $(t+1)$-th level and any $z \in \mathcal{F}(U_{t+1})$, on condition $\mathcal{A}$ reaches U_{t+1} and $y \in \mathcal{F}(U_{t+1})$, the probability that $y = z$ is at least $1/e^{2/m}2^{n-tm}$ where the probability is over the random choice of y and the internal coin tosses of $\mathcal{A}$.*

Proof. For fixed U_{t+1} and $z \in \mathcal{F}(U_{t+1})$, we would like to bound from below the value

$$
Q = \frac{\Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y = z]}{\Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y \in \mathcal{F}(U_{t+1})]}.
$$

We know from the first inequality of Claim 3 that

$$
\begin{aligned}
&\Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y \in \mathcal{F}(U_{t+1})] \\
&= \sum_{y' \in \mathcal{F}(U_{t+1})} \Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y = y'] \\
&\leq |\mathcal{F}(U_{t+1})| \cdot \frac{e^{1/m}}{2^n} \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}} \\
&\leq |\mathcal{I}(U_{t+1})| \cdot \frac{e^{1/m}}{2^n} \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}} \\
&\leq \frac{e^{1/m} \cdot 2^{n-tm}}{2^n} \cdot \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}}.
\end{aligned}
$$

On the other hand, from the second inequality of Claim 3, for any $z \in \mathcal{F}(U_{t+1})$, we have that

$$
\Pr[\mathcal{A} \text{ reaches } U_{t+1} \text{ and } y = z] \geq \frac{1}{e^{1/m}2^n} \prod_{i=1}^{tm} \frac{1}{2^{n-i-1}}.
$$

Therefore $Q \geq 1/e^{2/m}2^{n-tm}$. □

Claim 7 *The probability that $\mathcal{A}$ is successful is at least $\epsilon^{10}/65e^{3/\log n}(2n^2)^8$ where the probability is over the random choice of y and the internal coin tosses of $\mathcal{A}$.*

Proof. We say that $\mathcal{A}$ succeeds if

- $\mathcal{A}$ reaches a good node U_{t+1} at the $(t+1)$-th level and $y \in \mathcal{F}(U_j)$;
- A random choice of $h_{tm+1}, h_{tm+2}, \ldots, h_{n-1}$ defines a path to a good leaf that has at least one image, say z, in $\mathcal{F}(U_{t+1})$;
- $y = z$.

By Claims 4, 5 and 6, the probability that $\mathcal{A}$ succeeds is at least

$$\begin{aligned}
&\frac{\epsilon(1-\gamma)}{4e^{1/m}} \cdot \frac{\epsilon}{8} \cdot \frac{1}{e^{2/m}2^{n-tm}} \\
&= \epsilon^2 \cdot \frac{1-\gamma}{32 \cdot e^{3/m} \cdot 2^{n-tm}} \\
&> \frac{\epsilon^{10}}{65e^{3/m}(n2^{m+1})^8} \\
&= \frac{\epsilon^{10}}{65e^{3/\log n}(2n^2)^8}.
\end{aligned}$$

□

Thus, we can say that $\mathcal{A}$ inverts the one-way permutation f if an input y is fully balanced at the $(t+1)$-th level. On the other hand, the above analysis does not guarantee that $\mathcal{A}$ reaches the $(t+1)$-th level. So, some care must be taken. If y is fully balanced at the $(t+1)$-th level, then y is balanced in U_k for all $1 \leq k \leq t$ and therefore $A(U_k, y)/|\mathcal{H}^{(k)}| > 1/2^{m+1} = 1/2n$. It implies that $4n$ trials for step 2.1 in the inverting algorithm $\mathcal{A}$ are sufficient. The probability that $\mathcal{A}$ does not proceed to step 3 after $4n$ trials is exponentially small. If the rare case occurs then $\mathcal{A}$ may abort. Totally, $4nt \leq 4n^2/\log n$ trials are enough for $\mathcal{A}$ to reach the $(t+1)$-th level. Therefore, $\mathcal{A}$ runs in polynomial time and its success probability is at least $\epsilon^{10}/65e^{3/\log n}(2n^2)^8 - e^{-n}$.

6 Concluding Remarks

We have considered a naive parallel version of Naor-Ostrovsky-Venkatesan-Yung scheme [8] of the multiplicity $\log n$ and obtain an $O(n/\log n)$-round scheme. By introducing a technique called expected almost pairwise independence, we have shown that their protocol can be improved. Trivially, we can set the multiplicity parameter $m = c\log n$ and obtain the similar results. Moreover, we can set $m = 1$ and this means that our proof is alternative proof for the original scheme without the pairwise independence technique. On the other hand, an extension of our approach to the case $m = \omega(\log n)$ is not easy, since both the success probability and the efficiency of the inverting algorithm would violate the allowance.

We have utilized the expected almost pairwise independence for the security proof of our scheme. Since it is a natural generalization of the pairwise independence, we believe that the technique has other cryptographic applications.

References

[1] G. Brassard, D. Chaum and C. Crépeau: Minimum disclosure proofs of knowledge, *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[2] P. Dumais, D. Mayers and L. Salvail: Perfectly concealing quantum bit commitment from any quantum one-way permutation, In *Advances in Cryptology – EUROCRYPT 2000*, LNCS 1807, pp.300–315, 2000.

[3] O. Goldreich, S. Micali and A. Wigderson: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems, *J. Assoc. Comput. Mach.*, 38(3): 691–729, 1991.

[4] O. Goldreich: A uniform-complexity treatment of encryption and zero-knowledge, *J. Cryptol.*, 6(1):21–53, 1993.

[5] S. Goldwasser, S. Micali and C. Rackoff: The knowledge complexity of interactive proof systems, *SIAM J. Comput.*, 18(1):186–208, 1989.

[6] J. Håstad, R. Impagliazzo, L.A. Levin and M. Luby: A pseudorandom generator from any one-way function, *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[7] M. Naor: Bit commitment using pseudorandomness, *J. Cryptol.*, 4(2):151–158, 1991.

[8] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung: Perfect zero-knowledge arguments for NP using any one-way permutation, *J. Cryptol.*, 11(2):87–108, 1998.

[9] M. Naor and M. Yung: Universal one-way hash functions and their cryptographic applications, In *Proc. 21st ACM Symposium on Theory of Computing*, pp.33–43, 1989.

[10] J. Rompel: One-way functions are necessary and sufficient for secure signatures, In *Proc. 22nd ACM Symposium on Theory of Computing*, pp.387–394, 1990.