Provably Secure Key-Updating Schemes in Identity-Based Systems

S. Shinozaki † *T. Itoh* ‡ *A. Fujioka* † *S. Tsujii* †

† Department of Electrical and Electronic Engineering
Faculty of Engineering
Tokyo Institute of Technology
O-okayama, Meguro-ku, Tokyo 152, Japan

‡ Department of Information Processing
The Graduate School at Nagatsuta
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 227, Japan
e-mail: titoh@cc.titech.ac.jp

Abstract:

In this paper, we present *Key-Updating Schemes* in identity-based (identification or signature) systems, and consider the security of the schemes. We propose two kinds of key-updating schemes, i.e., one is sequential type and the other is parallel type, and show that both schemes are equivalent to each other in a polynomial time sense, i.e., there exists a deterministic polynomial time algorithm that transforms the sequential key-updating scheme to the parallel one, and vice versa. We also show that even if any polynomially many entities conspire to find a secret-key of any other entities, both key-updating schemes are provably secure against polynomially many times key-updating if decrypting RSA is hard.

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 16-30, 1991.

1 Introduction

In identity-based systems, each entity i has his(her) own identity number ID_i, and a trusted center needs to generate a pair of a public information P (known to all entities) and a secret information S (known to only the trusted center), and a pair of public-key PK_i and secret-key SK_i for entity i. Let a probabilistic polynomial time algorithm CKG be a *center-key generator* that, on input 1^k, outputs a pair of the public information P ($|P| = O(k^c)$ for some constant $c > 0$) and the secret information S ($|S| = O(k^d)$ for some constant $d > 0$), i.e., $CKG(1^k) = \langle P, S\rangle$, and let a probabilistic polynomial time algorithm EKG be a *entity-key generator* that, on input 1^k, P, S, and ID_i, outputs a pair of public-key PK_i and secret-key SK_i for entity i, i.e., $EKG(\langle 1^k, P, S, ID_i\rangle) = \langle PK_i, SK_i\rangle$. Note that k is the security parameter.

When a foolish entity j carelessly loses his secret-key SK_j or reveals it and asks the trusted center again to generate a new pair of public-key PK_j' and secret-key SK_j' for him, what should the trusted center do? If the system is provably secure (see, e.g., [FS], [FFS], [GQ], [OO].), i.e., there exist no efficient algorithms for entity j to derive the secret information S from P, ID_j, and a single pair of $\langle PK_j, SK_j\rangle$, then (presumably) the simplest and secure way to update the secret-key SK_j to SK_j' is to make the trusted center run CKG on input 1^k in order to regenerate a new pair of public information P' and secret information S' and to make the trusted center regenerate a new pair of public-key PK_j' and secret-key SK_j' for the entity j by running EKG on input 1^k, P', S', and ID_j (or SK_j). This scheme, however, imposes cumbersome procedures on the trusted center and all entities, because the trusted center must regenerate not only a new pair of public-key PK_j' and secret-key SK_j' for the foolish entity j but a new pair of public-key PK_i' and secret-key SK_i' for every entity i ($\neq j$).

Another way to update the secret-key SK_j to SK_j' is to make the trusted center run only EKG on input 1^k, P, S, and ID_j (or SK_j) and to regenerate a new pair of public-key PK_j' and secret-key SK_j' only for the foolish entity j, while those for the other entities i ($\neq j$) are unchanged. This scheme is much simpler than before, but unfortunately there might be a possibility that the entity j can derive the secret information S efficiently from P, ID_j, PK_j, SK_j, PK_j', and SK_j'.

Thus this provokes us to construct efficient and *provably* secure key-updating schemes in identity-based systems in the above sense. To do this, we take the extended Fiat-Shamir scheme [GQ], [OO] as an identity-based system, and apply two kinds of key-updating schemes, one is sequential and the other is parallel, to the extended Fiat-Shamir scheme. (The details will be discussed in Section 2.) We also show that our key-updating schemes are *provably* secure against polynomially many times key-updating even if any polynomially many entities conspire to find a secret-key of any other entities.

The organization of this paper is as follows: Section 2 presents a brief description of *key-generation* and *key-distribution* in the extended Fiat-Shamir scheme [GQ], [OO], and proposes two kinds of *key-updating* schemes, sequential one and parallel one; Section 3 shows that both schemes are equivalent to each other in a polynomial time sense, i.e., there exists a polynomial time algorithm that transforms the sequential key-updating scheme to the parallel one, and vise versa; Section 4 gives a main result that both key-updating schemes are provably secure against polynomially many times key-updating, i.e., any polynomially many conspiring entities can not find a secret-key of any other entities under the assumption that decrypting RSA is hard; and Section 5 finally gives conclusion and remarks, and refers to extensions of our results to more general settings and the security of the schemes against conspiracy of entities.

2 Key-Updating Schemes

2.1 Extended Fiat-Shamir Scheme

This subsection presents a brief description of key-generation and key-distribution in the extended Fiat-Shamir scheme [GQ], [OO]. The extended Fiat-Shamir scheme is an extension of the Fiat-Shamir scheme [FS], [FFS], and is shown, under the assumption that factoring is hard, to be zero-knowledge in the sequential execution (of the protocol) and to be non-transferable in the parallel execution (of the protocol). This scheme is an identity-based system, and thus the trusted center needs to generate a pair of public information P (known to all entities) and secret information S (known to only trusted center) and to distribute a pair of public-key PK_i and secret-key SK_i for each entity i with his identity number ID_i, in the following way:

The trusted center has two probabilistic polynomial time algorithms, i.e., *center-key generator* CKG and *entity-key generator* EKG; On input 1^k, the center-key generator CKG outputs a pair of public information n $(= p \cdot q)$ and secret information $\langle p, q \rangle$, where $p, q \in \mathcal{OP}$ and $|p| = |q| = k$, and on input 1^k, n, $\langle p, q \rangle$, and ID_i, the entity-key generator EKG outputs a pair of public-key e_i and secret-key S_i for entity i such that $S_i^{e_i} \equiv ID_i \pmod{n}$. Note that $\mathcal{OP}$ denotes a set of odd primes and $|a|$ denotes the length of binary encoding of a. For details of identification and signature protocols in the extended Fiat-Shamir scheme, see [GQ], [OO].

2.2 Key-Updating Schemes

In this subsection, we propose two kinds of key-updating schemes, sequential one [FT] and parallel one, in the extended Fiat-Shamir scheme. Consider the case where some entity i asks the trusted center to issue a new pair of public-key e_i' and secret-key S_i' for the entity i in some reason, e.g., losing or revealing his original secret-key S_i.

Informally, our key-updating schemes are as follows: (1) Sequential Key-Updating Scheme (**SKU**) is a key-updating scheme in which the trusted center runs the entity-key generator EKG on input 1^k, n, $\langle p,q\rangle$, e_i, and S_i (instead of ID_i), and generates a new pair of $\langle e_i', S_i'\rangle$ such that $S_i'^{e_i'} \equiv S_i \pmod n$ and $e_i \neq e_i'$, and (2) Parallel Key-Updating Scheme (**PKU**) is a key-updating scheme in which the trusted center runs the entity-key generator EKG on input 1^k, n, $\langle p,q\rangle$, e_i, and ID_i, and generates a new pair of $\langle e_i', S_i'\rangle$ such that $S_i'^{e_i'} \equiv ID_i \pmod n$ and $e_i \neq e_i'$. Note that for entity i, a pair of public-key and secret-key will be $\langle e_i e_i', S_i'\rangle$ in **SKU**, while will be $\langle e_i', S_i'\rangle$ in **PKU**.

This formulation, however, does not necessarily match our desire, because a malicious entity j might ask the trusted center to issue new pairs of $\langle e_j', S_j'\rangle$ many times for compromising the secret information $\langle p,q\rangle$. Then we formally define our key-updating schemes in more general settings.

Let $U(|n|)$ be any fixed polynomial in $|n|$, and let $\mathcal{OP}(\ell)$ denote a set of odd primes less than ℓ. Here we assume that each entity i is allowed to ask the trusted center to issue new pairs of $\langle e_i', S_i'\rangle$ at most $U(|n|)$ times.

Sequential Key-Updating Scheme (SKU):

Initial Key-Setting Stage: For each entity i (with $ID_i \in \mathcal{Z}_n^*$), the trusted center distributes a pair of his public-key $e_i^{(0)}$ and his secret-key $S_i^{(0)}$ such that $ID_i \equiv \left\{S_i^{(0)}\right\}^{e_i^{(0)}}$ $\pmod n$, where $e_i^{(0)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$, $S_i^{(0)} \not\equiv ID_i \pmod n$, and $ID_i^2 \not\equiv 1 \pmod n$.

Key-Updating Stage: For entity i in the r_i-*th* ($1 \leq r_i \leq U(|n|)$) key-updating, the trusted center distributes a new pair of $\langle e_i^{(r_i)}, S_i^{(r_i)}\rangle$ such that $S_i^{(r_i-1)} \equiv \left\{S_i^{(r_i)}\right\}^{e_i^{(r_i)}}$ $\pmod n$, where $e_i^{(r_i)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$, $e_i^{(j)} \neq e_i^{(r_i)}$ $(0 \leq j < r_i)$, $S_i^{(r_i)} \not\equiv ID_i \pmod n$, and $S_i^{(j)} \not\equiv S_i^{(r_i)} \pmod n$ $(0 \leq j < r_i)$.

Remark 2.1: In the r_i-*th* key-updating of **SKU**, a pair of the public-key and the secret-key will be $\langle e_i^{(0)} e_i^{(1)} \cdots e_i^{(r_i)}, S_i^{(r_i)}\rangle$. The condition $S_i^{(j)} \not\equiv ID_i \pmod n$ $(0 \leq j \leq r_i)$ shows that the trusted center avoids distributing trivial secret-key $S_i^{(j)}$, and the condition $S_i^{(j)} \not\equiv S_i^{(r_i)} \pmod n$ $(0 \leq j < r_i)$ implies that the trusted center does not distribute the same secret-key $S_i^{(r_i)}$ again, because old secret-keys might be known to someone else. The trusted center does not care about collisions of secret-keys among entities.

Parallel Key-Updating Scheme (PKU):

Initial Key-Setting Stage: For each entity i (with $ID_i \in \mathcal{Z}_n^*$), the trusted center distributes a pair of his public-key $f_i^{(0)}$ and his secret-key $T_i^{(0)}$ such that $ID_i \equiv \left\{T_i^{(0)}\right\}^{f_i^{(0)}}$ (mod n), where $f_i^{(0)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$, $T_i^{(0)} \not\equiv ID_i \pmod n$, and $ID_i^2 \not\equiv 1 \pmod n$.

Key-Updating Stage: For entity i in the r_i-*th* ($1 \le r_i \le U(|n|)$) key-updating, the trusted center distributes a new pair of $\langle f_i^{(r_i)}, T_i^{(r_i)}\rangle$ such that $ID_i \equiv \left\{T_i^{(r_i)}\right\}^{f_i^{(r_i)}} \pmod n$, where $f_i^{(r_i)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$, $f_i^{(j)} \ne f_i^{(r_i)}$ ($0 \le j < r_i$), $T_i^{(r_i)} \not\equiv ID_i \pmod n$, $T_i^{(j)} \not\equiv T_i^{(r_i)}$ (mod n) ($0 \le j < r_i$), and $ID_i^{f_i^{(j)} \cdots f_i^{(r_i)}} \not\equiv ID_i \pmod n$ ($0 \le j < r_i$).

Remark 2.2: In the r_i-*th* key-updating of **PKU**, a pair of the public-key and the secret-key will be $\langle f_i^{(r_i)}, T_i^{(r_i)}\rangle$. The meaning of conditions $T_i^{(j)} \not\equiv ID_i \pmod n$ ($0 \le j \le r_i$) and $T_i^{(j)} \not\equiv T_i^{(r_i)} \pmod n$ ($0 \le j < r_i$) is similar to the one in the Remark 2.1. The trusted center does not care about collisions of secret-keys among entities.

3 Transforms Between SKU and PKU

This section shows that key-updating schemes **SKU** and **PKU** are equivalent to each other in a polynomial time sense, i.e., there exists a deterministic polynomial time algorithm that transforms **SKU** to **PKU**, and vice versa.

Let $\mathcal{SC}_k$ denote a set of strong composites with the security parameter k, i.e.,

$$\mathcal{SC}_k = \{n \mid n = p \cdot q,\ p \ne q,\ |p| = |q| = k,\\ p = 2p' + 1,\ q = 2q' + 1,\ p, q, p', q' \in \mathcal{OP}\}.$$

To prove that for $n \in \mathcal{SC}_k$, key-updating schemes **SKU** and **PKU** are deterministic polynomial time transformable to each other, we need to show the following lemmas:

Lemma 3.1: *Let $n \in \mathcal{SC}_k$. Then for any odd e less than $\lfloor\sqrt{n}/4\rfloor$, $e < \min\{p', q'\}$ and $\gcd(e, \lambda(n)) = 1$, where $\lambda(n)$ is the Carmichael function* [Kr] *of n.*

Proof: From the definition of $\mathcal{SC}_k$, it follows that

$$\lambda(n) = \mathrm{lcm}(p-1, q-1) = \mathrm{lcm}(2p', 2q') = 2p'q'.$$

Note that $n \in \mathcal{SC}_k$, i.e., $|p| = |q| = k$, then $2 \cdot \min\{p, q\} > \max\{p, q\}$. Hence,

$$\begin{aligned} \lfloor\sqrt{n}/4\rfloor &\le \lfloor\max\{p,q\}/4\rfloor \\ &\le \lfloor\min\{p,q\}/2\rfloor \\ &= \lfloor(2\cdot\min\{p',q'\}+1)/2\rfloor \\ &= \lfloor\min\{p',q'\}+1/2\rfloor = \min\{p',q'\}, \end{aligned}$$

and thus $e < \min\{p', q'\}$. It immediately follows, from the fact that $\lambda(n) = 2p'q'$, that $\gcd(e, \lambda(n)) = \gcd(e, 2p'q') = 1$, because $e < \min\{p', q'\}$, e is odd, and $p', q' \in \mathcal{OP}$. □

Lemma 3.2: *Let $n \in \mathcal{SC}_k$, and $x \in Z_n^*$ such that $x^2 \not\equiv 1 \pmod{n}$. For any distinct odd numbers a_1 and a_2, $x^{a_2-a_1} \not\equiv 1 \pmod{n}$, where $a_1, a_2 < \lfloor\sqrt{n}/4\rfloor$.*

Proof: By Contradiction. Without loss of generality, we assume that $a_1 < a_2$. Assume that $x^{a_2-a_1} \equiv 1 \pmod{n}$. This implies that the order of x modulo n divides both $\lambda(n)$ and $a_2 - a_1$. Since $n \in \mathcal{SC}_k$, $\lambda(n) = 2p'q'$ and $0 < a_2 - a_1 < \min\{p', q'\}$ (see Lemma 3.1.), and thus the order of x modulo n is equal to either 1 or 2. This, however, contradicts the assumption that $x^2 \not\equiv 1 \pmod{n}$. Hence $x^{a_2-a_1} \not\equiv 1 \pmod{n}$. □

Lemma 3.3: *Let r be any positive integer and let $n \in \mathcal{SC}_k$. Let $ID \in Z_n^*$, $S^{(i)} \in Z_n^*$ $(0 \le i \le r)$, and $e^{(i)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ $(0 \le i \le r)$ satisfy the relation that $ID \equiv \left\{S^{(0)}\right\}^{e^{(0)}} \pmod{n}$, $S^{(i-1)} \equiv \left\{S^{(i)}\right\}^{e^{(i)}} \pmod{n}$ $(1 \le i \le r)$, where $ID^2 \not\equiv 1 \pmod{n}$ and $e^{(i)} \ne e^{(j)}$ $(0 \le i < j \le r)$. Then $ID \not\equiv S^{(i)} \pmod{n}$ $(0 \le i \le r)$ and $S^{(i)} \not\equiv S^{(j)} \pmod{n}$ $(0 \le i < j \le r)$ iff $ID^{e^{(i)} \cdots e^{(j)}} \not\equiv ID \pmod{n}$ $(0 \le i < j \le r)$.*

Proof: Since $n \in \mathcal{SC}_k$ and ID $(\in Z_n^*)$ satisfies $ID^2 \not\equiv 1 \pmod{n}$, it follows, from Lemma 3.2, that for $e^{(i)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ $(0 \le i \le r)$, $ID^{e^{(i)}-1} \not\equiv 1 \pmod{n}$ $(0 \le i \le r)$, and hence $ID^{e^{(i)}} \not\equiv ID \pmod{n}$ $(0 \le i \le r)$. Note that $x^e \equiv y^e \pmod{n}$ iff $x \equiv y \pmod{n}$ for e such that $\gcd(e, \lambda(n)) = 1$. (see Lemma 3.1.) Then for all i $(0 \le i \le r)$,

$$\begin{aligned} ID \equiv S^{(i)} \pmod{n} &\iff ID^{e^{(0)} \cdots e^{(i)}} \equiv \left\{S^{(i)}\right\}^{e^{(0)} \cdots e^{(i)}} \pmod{n} \\ &\iff ID^{e^{(0)} \cdots e^{(i)}} \equiv ID \pmod{n}. \end{aligned}$$

On the other hand, for any i, j $(0 \le i < j \le r)$, we have

$$\begin{aligned} S^{(i)} \equiv S^{(j)} \pmod{n} &\iff \left\{S^{(i)}\right\}^{e^{(0)} \cdots e^{(i)} e^{(i+1)} \cdots e^{(j)}} \equiv \left\{S^{(j)}\right\}^{e^{(0)} \cdots e^{(j)}} \pmod{n} \\ &\iff ID^{e^{(i+1)} \cdots e^{(j)}} \equiv ID \pmod{n}. \end{aligned}$$

Hence $ID \not\equiv S^{(i)} \pmod{n}$ $(0 \le i \le r)$ and $S^{(i)} \not\equiv S^{(j)}$ $(0 \le i < j \le r)$ iff $ID^{e^{(i)} \cdots e^{(j)}} \not\equiv ID \pmod{n}$ $(0 \le i < j \le r)$. □

Lemma 3.4: *Let r be any positive integer and let $n \in \mathcal{SC}_k$. Let $ID \in Z_n^*$, $T^{(i)} \in Z_n^*$ $(0 \le i \le r)$, and $f^{(i)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ $(0 \le i \le r)$ satisfy the relation that $ID \equiv \left\{T^{(i)}\right\}^{f^{(i)}} \pmod{n}$ $(0 \le i \le r)$, where $ID^2 \not\equiv 1 \pmod{n}$, and $f^{(i)} \ne f^{(j)}$ $(0 \le i < j \le r)$. Then $ID \not\equiv T^{(i)} \pmod{n}$ $(0 \le i \le r)$ and $T^{(i)} \not\equiv T^{(j)} \pmod{n}$ $(0 \le i < j \le r)$.*

Proof: Since $n \in \mathcal{SC}_k$ and $ID(\in Z_n^*)$ satisfies $ID^2 \not\equiv 1 \pmod{n}$, it follows, from Lemma 3.2, that for $f^{(i)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ $(0 \le i \le r)$, $ID^{f^{(j)}-f^{(i)}} \not\equiv 1 \pmod{n}$ $(0 \le i < j \le r)$ and $ID^{f^{(i)}-1} \not\equiv 1 \pmod{n}$ $(0 \le i \le r)$. Then for all i $(0 \le i \le r)$, we have

$$\begin{aligned} ID \equiv T^{(i)} \pmod{n} &\iff ID^{f^{(i)}} \equiv \left\{T^{(i)}\right\}^{f^{(i)}} \pmod{n} \\ &\iff ID^{f^{(i)}} \equiv ID \pmod{n} \\ &\iff ID^{f^{(i)}-1} \equiv 1 \pmod{n}. \end{aligned}$$

On the other hand, for any i, j $(0 \le i < j \le r)$, we also have

$$\begin{aligned} T^{(i)} \equiv T^{(j)} \pmod{n} &\iff \left\{\left\{T^{(i)}\right\}^{f^{(i)}}\right\}^{f^{(j)}} \equiv \left\{\left\{T^{(j)}\right\}^{f^{(j)}}\right\}^{f^{(i)}} \pmod{n} \\ &\iff ID^{f^{(j)}} \equiv ID^{f^{(i)}} \pmod{n} \\ &\iff ID^{f^{(j)}-f^{(i)}} \equiv 1 \pmod{n}, \end{aligned}$$

hence $ID \not\equiv T^{(i)} \pmod{n}$ $(0 \le i \le r)$ and $T^{(i)} \not\equiv T^{(j)} \pmod{n}$ $(0 \le i < j \le r)$. □

Let $U(|n|)$ be any fixed polynomial in $|n|$ and let r be any positive integer not greater than $U(|n|)$. Here we define $\mathcal{C}_{\mathrm{SKU}}$ to be a set of tuples $\langle n, ID, \boldsymbol{S}^{(r)}, \boldsymbol{e}^{(r)}\rangle$ that satisfy

$$\begin{aligned} ID &\equiv \left\{S^{(0)}\right\}^{e^{(0)}} \pmod{n}; \\ S^{(i-1)} &\equiv \left\{S^{(i)}\right\}^{e^{(i)}} \pmod{n}\ (1 \le i \le r); \\ ID^2 &\not\equiv 1 \pmod{n}; \\ e^{(i)} &\in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)\ (0 \le i \le r); \\ e^{(i)} &\ne e^{(j)}\ (0 \le i < j \le r); \\ ID^{e^{(i)} \cdots e^{(j)}} &\not\equiv ID \pmod{n}\ (0 \le i < j \le r), \end{aligned}$$

in the r-th key-updating of **SKU** (see Lemma 3.3.), where $n \in \mathcal{SC}_k$, $ID \in Z_n^*$, $\boldsymbol{S}^{(r)} = (S^{(0)}, S^{(1)}, \ldots, S^{(r)})$, and $\boldsymbol{e}^{(r)} = (e^{(0)}, e^{(1)}, \ldots, e^{(r)})$. In a way similar to the above, we define $\mathcal{C}_{\mathrm{PKU}}$ to be a set of tuples $\langle n, ID, \boldsymbol{T}^{(r)}, \boldsymbol{f}^{(r)}\rangle$ that satisfy

$$\begin{aligned} ID &\equiv \left\{T^{(i)}\right\}^{f^{(i)}} \pmod{n}\ (0 \le i \le r); \\ ID^2 &\not\equiv 1 \pmod{n}; \\ f^{(i)} &\in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)\ (0 \le i \le r); \\ f^{(i)} &\ne f^{(j)}\ (0 \le i < j \le r); \\ ID^{f^{(i)} \cdots f^{(j)}} &\not\equiv ID \pmod{n}\ (0 \le i < j \le r), \end{aligned}$$

in the r-*th* key-updating of **PKU** (see Lemma 3.4.), where $n \in \mathcal{SC}_k$, $ID \in Z_n^*$, $\boldsymbol{T}^{(r)} = (T^{(0)}, T^{(1)}, \ldots, T^{(r)})$, and $\boldsymbol{f}^{(r)} = (f^{(0)}, f^{(1)}, \ldots, f^{(r)})$.

We use $A_{\mathrm{SKU}\to\mathrm{PKU}}$ to denote any algorithm that, on input $\langle n, ID, \boldsymbol{S}^{(r)}, \boldsymbol{e}^{(r)} \rangle \in \mathcal{C}_{\mathrm{SKU}}$, outputs $\langle n, ID, \boldsymbol{T}^{(r)}, \boldsymbol{f}^{(r)} \rangle \in \mathcal{C}_{\mathrm{PKU}}$, and $A_{\mathrm{PKU}\to\mathrm{SKU}}$ to denote any algorithm that, on input $\langle n, ID, \boldsymbol{T}^{(r)}, \boldsymbol{f}^{(r)} \rangle \in \mathcal{C}_{\mathrm{PKU}}$, outputs $\langle n, ID, \boldsymbol{S}^{(r)}, \boldsymbol{e}^{(r)} \rangle \in \mathcal{C}_{\mathrm{SKU}}$. Then we have the following theorems on deterministic polynomial time transformability between **SKU** and **PKU**.

Theorem 3.5: *There exists a deterministic polynomial time algorithm $A_{\mathrm{SKU}\to\mathrm{PKU}}$.*

Sketch of Proof: On input $\langle n, ID, \boldsymbol{S}^{(r)}, \boldsymbol{e}^{(r)} \rangle \in \mathcal{C}_{\mathrm{SKU}}$, the algorithm $A_{\mathrm{SKU}\to\mathrm{PKU}}$ sets $T^{(0)} := S^{(0)}$, $f^{(i)} := e^{(i)}$ $(0 \le i \le r)$, and computes $T^{(i)} \equiv \left\{S^{(i)}\right\}^{e^{(0)}\cdots e^{(i-1)}} \pmod n$ $(1 \le i \le r)$. Then it outputs $\langle n, ID, \boldsymbol{T}^{(r)}, \boldsymbol{f}^{(r)} \rangle$, where $\boldsymbol{T}^{(r)} = (T^{(0)}, T^{(1)}, \ldots, T^{(r)})$, $\boldsymbol{f}^{(r)} = (f^{(0)}, f^{(1)}, \ldots, f^{(r)})$. It is easy to see that the algorithm $A_{\mathrm{SKU}\to\mathrm{PKU}}$ runs in deterministic polynomial time and $\langle n, ID, \boldsymbol{T}^{(r)}, \boldsymbol{f}^{(r)} \rangle \in \mathcal{C}_{\mathrm{PKU}}$. □

Theorem 3.6: *There exists a deterministic polynomial time algorithm $A_{\mathrm{PKU}\to\mathrm{SKU}}$.*

Sketch of Proof: Let n be an odd composite. We assume here that x_1 is the a_1-*th* root of y modulo n, and x_2 is the a_2-*th* root of y modulo n, where $\gcd(a_1, a_2) = 1$ and $y \in Z_n^*$. Then we can compute x, the a_1a_2-*th* root of y modulo n, by algorithm E (see below.) in deterministic polynomial time without knowing prime factors of n. On input n, y, x_1, a_1, x_2, and a_2, the algorithm E computes two integers s and t such that $ta_1 + sa_2 = 1$ by Euclidean algorithm, and outputs $x \equiv x_1^s \cdot x_2^t \pmod n$. It is easy to see that the algorithm E runs in deterministic polynomial time and x is the a_1a_2-*th* root of y modulo n. The algorithm $A_{\mathrm{PKU}\to\mathrm{SKU}}$ runs in the following way:

On input $\langle n, ID, \boldsymbol{T}^{(r)}, \boldsymbol{f}^{(r)} \rangle \in \mathcal{C}_{\mathrm{PKU}}$, the algorithm $A_{\mathrm{PKU}\to\mathrm{SKU}}$ sets $T^{(0)} := S^{(0)}$, $f^{(0)} := e^{(0)}$, and computes x_i $(1 \le i \le r)$ by running the deterministic polynomial time algorithm E on input $\langle n, ID, S^{(i-1)}, e^{(0)}e^{(1)}\cdots e^{(i-1)}, T^{(i)}, f^{(i)} \rangle$. Then the algorithm $A_{\mathrm{PKU}\to\mathrm{SKU}}$ substitutes x_i to $S^{(i)}$ and $f^{(i)}$ to $e^{(i)}$ $(1 \le i \le r)$, and outputs $\langle n, ID, \boldsymbol{S}^{(r)}, \boldsymbol{e}^{(r)} \rangle$, where $\boldsymbol{S}^{(r)} = (S^{(0)}, S^{(1)}, \ldots, S^{(r)})$ and $\boldsymbol{e}^{(r)} = (e^{(0)}, e^{(1)}, \ldots, e^{(r)})$.

It is not difficult to see that the algorithm $A_{\mathrm{PKU}\to\mathrm{SKU}}$ runs in deterministic polynomial time and $\langle n, ID, \boldsymbol{S}^{(r)}, \boldsymbol{e}^{(r)} \rangle \in \mathcal{C}_{\mathrm{SKU}}$. □

4 SKU and PKU are Provably Secure

This section shows that key-updating schemes **SKU** and **PKU** are provably secure against polynomially many times key-updating under the assumption that decrypting RSA is hard for $n \in \mathcal{SC}_k$, i.e., even if any polynomially many entities conspire, they can not find a secret-key of any other entity in polynomially many times key-updating.

To show this, we provide several lemmas in the following:

Lemma 4.1: *Let $n \in \mathcal{SC}_k$ and let $U(|n|)$ be any fixed polynomial in $|n|$. Let r be any positive integer not greater than $U(|n|)$ and let $e^{(i)} < \lfloor\sqrt{n}/4\rfloor$ $(0 \le i \le r-1)$ be distinct r odd primes. Then the probability P that for any $d < \lfloor\sqrt{n}/4\rfloor$, $d \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ and $d \ne e^{(i)}$ $(0 \le i \le r-1)$ is greater than $C/|n|$ for some $C > 0$ and sufficiently large n.*

Proof: Let $\pi(x)$ denote the number of primes not greater than x $(x \ge 2)$. From prime number theorem [HW], it follows that

$$\pi(x) > C_0 \frac{x}{\log_2 x},$$

for some constant C_0. Then the probability P is

$$\begin{aligned} P &> \frac{C_0 \dfrac{\lfloor\sqrt{n}/4\rfloor - 1}{\log_2(\lfloor\sqrt{n}/4\rfloor - 1)} - (r+1)}{\lfloor\sqrt{n}/4\rfloor - 1} > \frac{C_1}{\log_2\lfloor\sqrt{n}/4\rfloor} - \frac{r+1}{\lfloor\sqrt{n}/4\rfloor - 1} \\ &> \frac{C_1}{\log_2\lfloor\sqrt{n}/4\rfloor} - \frac{2U(|n|)}{\lfloor\sqrt{n}/4\rfloor - 1} > \frac{C_2}{\log_2\lfloor\sqrt{n}/4\rfloor} \\ &> \frac{C_3}{\log_2(\sqrt{n}/4)} > \frac{C_4}{\lfloor\log_2 n\rfloor + 1}, \end{aligned}$$

thus $P > C/|n|$ for some constant C and sufficiently large n. □

Lemma 4.2: *Let $n \in \mathcal{SC}_k$ and let $U(|n|)$ be any fixed polynomial in $|n|$. Let r be any positive integer not greater than $U(|n|)$ and let $\mathbf{e}^{(r-1)} = (e^{(0)}, e^{(1)}, \ldots, e^{(r-1)})$, where $e^{(i)} < \lfloor\sqrt{n}/4\rfloor$ $(0 \le i \le r-1)$ are distinct r odd primes. Define $\hat{\mathbf{e}}^{(r)} = (\hat{e}^{(0)}, \hat{e}^{(1)}, \ldots, \hat{e}^{(r)})$ to be $\hat{\mathbf{e}}^{(r)} = (e^{(0)}, e^{(1)}, \ldots, e^{(r-1)}, d)$ for any $d \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ such that $d \ne e^{(i)}$ $(0 \le i \le r-1)$. Then for any $g \in Z_n^*$ such that $g^2 \not\equiv 1 \pmod n$, $g^{\hat{e}^{(i)} \cdots \hat{e}^{(j)}} \not\equiv g \pmod n$ $(0 \le i < j \le r)$ iff $g^{e^{(s)} \cdots e^{(t)}} \not\equiv g \pmod n$ $(0 \le s < t \le r-1)$ and $\left\{\prod_{\ell=i}^{r-1} e^{(\ell)}\right\} \cdot d \not\equiv 1 \pmod L$ $(0 \le i \le r-1)$, where L is the order of g modulo n.*

Proof: Let L denote the order of g modulo n. Then it is clear that

$$g^{e^{(s)} \cdots e^{(t)}} \not\equiv g \pmod n \iff \prod_{\ell=s}^{t} e^{(\ell)} \not\equiv 1 \pmod L,$$

for all s, t $(0 \le s < t \le r-1)$, and

$$g^{\hat{e}^{(i)} \cdots \hat{e}^{(j)}} \not\equiv g \pmod n \iff \prod_{\ell=i}^{j} \hat{e}^{(\ell)} \not\equiv 1 \pmod L,$$

for all i, j $(0 \leq i < j \leq r)$. Thus it suffices to show that

$$\prod_{\ell=i}^{j} \hat{e}^{(\ell)} \not\equiv 1 \pmod{L} \ (0 \leq i < j \leq r)$$
$$\Longleftrightarrow \prod_{\ell=s}^{t} e^{(\ell)} \not\equiv 1 \pmod{L} \ (0 \leq s < t \leq r-1)$$
$$\bigwedge \left\{ \prod_{\ell=i}^{r-1} e^{(\ell)} \right\} \cdot d \not\equiv 1 \pmod{L} \ (0 \leq i \leq r-1).$$

When $j < r$, $\prod_{\ell=i}^{j} \hat{e}^{(\ell)} \not\equiv 1 \pmod{L}$ $(0 \leq i < j \leq r-1)$ iff $\prod_{\ell=s}^{t} e^{(\ell)} \not\equiv 1 \pmod{L}$ $(0 \leq s < t \leq r-1)$, and when $j = r$, $\prod_{\ell=i}^{j} \hat{e}^{(\ell)} \not\equiv 1 \pmod{L}$ $(0 \leq i \leq r-1)$ iff $\left\{\prod_{\ell=i}^{r-1} e^{(\ell)}\right\} \cdot d \not\equiv 1 \pmod{L}$ $(0 \leq i \leq r-1)$. Thus it is immediate to see that $g^{\hat{e}^{(i)} \cdots \hat{e}^{(j)}} \not\equiv g \pmod{n}$ $(0 \leq i < j \leq r)$ iff $g^{e^{(s)} \cdots e^{(t)}} \not\equiv g \pmod{n}$ $(0 \leq s < t \leq r-1)$ and $\left\{\prod_{\ell=i}^{r-1} e^{(\ell)}\right\} \cdot d \not\equiv 1 \pmod{L}$ $(0 \leq i \leq r-1)$. □

For $n \in \mathcal{SC}_k$ and $e^{(i)} \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ $(0 \leq i \leq r-1)$ such that $e^{(i)} \neq e^{(j)}$ $(0 \leq i < j \leq r-1)$, we define $\mathcal{D}_{r+1}$ to be a set of $(r+1)$ distinct $d_j \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ $(1 \leq j \leq r+1)$ such that $d_j \neq e^{(i)}$ $(1 \leq j \leq r+1, 0 \leq i \leq r-1)$.

Lemma 4.3: *Let $n \in \mathcal{SC}_k$ and let $U(|n|)$ be any fixed polynomial in $|n|$. Let r be any positive integer not greater than $U(|n|)$ and let $e^{(r-1)} = (e^{(0)}, e^{(1)}, \ldots, e^{(r-1)})$, where $e^{(i)} < \lfloor\sqrt{n}/4\rfloor$ $(0 \leq i \leq r-1)$ are distinct r odd primes. Then for any $g \in Z_n^*$ such that $g^2 \not\equiv 1 \pmod{n}$, there exists at least one $d \in \mathcal{D}_{r+1}$ such that $\left\{\prod_{\ell=i}^{r-1} e^{(\ell)}\right\} \cdot d \not\equiv 1 \pmod{L}$ for any i $(0 \leq i \leq r-1)$, where L is the order of g modulo n.*

Proof: Let L denote the order of g modulo n. Since $n \in \mathcal{SC}_k$ and g $(\in Z_n^*)$ satisfies $g^2 \not\equiv 1 \pmod{n}$, $L \geq \min\{p', q'\}$. From Lemma 3.1, it follows that $\lfloor\sqrt{n}/4\rfloor \leq \min\{p', q'\}$, and thus for any $d < \lfloor\sqrt{n}/4\rfloor$, $d < L$. For some i $(0 \leq i \leq r-1)$, there exist at most r distinct $d_j < \lfloor\sqrt{n}/4\rfloor$ that satisfy $\left\{\prod_{\ell=i}^{r-1} e^{(\ell)}\right\} \cdot d_j \equiv 1 \pmod{L}$, hence at least one $d \in \mathcal{D}_{r+1}$ must satisfy $\left\{\prod_{\ell=i}^{r-1} e^{(\ell)}\right\} \cdot d \not\equiv 1 \pmod{L}$ for any i $(0 \leq i \leq r-1)$. □

For simplicity, we assume that every entity i is numbered as $1, 2, \ldots$. Let $E(|n|)$ and $U(|n|)$ be any fixed polynomials in $|n|$. When m $(< E(|n|))$ entities, each of which is in the r_i-th $(1 \leq r_i \leq U(|n|))$ key-updating, conspire to find a secret-key of any other entity u $(> m)$, they can use the following information in **SKU**.

$$n \in \mathcal{SC}_k;\ e_u^{(r_u)} = e'_0 e'_1 \cdots e'_{r_u},\ e'_i \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)\ (0 \leq i \leq r_u);$$
$$ID_u \in Z_n^*\ (ID_u^2 \not\equiv 1 \pmod{n});\ (n, ID_i, S_i^{(r_i)}, e_i^{(r_i)}) \in \mathcal{C}_{\mathrm{SKU}}\ (1 \leq i \leq m),$$

where $S_i^{(r_i)} = (S_i^{(0)}, S_i^{(1)}, \ldots, S_i^{(r_i)})$, $e_i^{(r_i)} = (e_i^{(0)}, e_i^{(1)}, \ldots, e_i^{(r_i)})$ $(1 \le i \le m)$. Let R be m tuple of integers, $R = (r_1, r_2, \ldots, r_m)$, and each r_i $(1 \le i \le m)$ is not greater than $U(|n|)$. Then we use $INV_{\text{SKU}}^{(m,R)}$ to denote any algorithm that, on input

$$n \in \mathcal{SC}_k;\ e_u^{(r_u)} = e'_0 e'_1 \cdots e'_{r_u},\ e'_i \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)\ (0 \le i \le r_u);$$
$$x \in \mathcal{Z}_n^*\ (x^2 \not\equiv 1 \pmod{n});\ \langle n, ID_i, S_i^{(r_i)}, e_i^{(r_i)}\rangle \in \mathcal{C}_{\text{SKU}}\ (1 \le i \le m),$$

outputs $y \in \mathcal{Z}_n^*$ such that $x \equiv y^e \pmod{n}$ for a non-negligible fraction of $x \in \mathcal{Z}_n^*$, and we use $INV_{\text{PKU}}^{(m,R)}$ to denote any algorithm that, on input

$$n \in \mathcal{SC}_k;\ f \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor);\ x \in \mathcal{Z}_n^*\ (x^2 \not\equiv 1 \pmod{n});$$
$$\langle n, ID_i, T_i^{(r_i)}, f_i^{(r_i)}\rangle \in \mathcal{C}_{\text{PKU}}\ (1 \le i \le m),$$

outputs $y \in \mathcal{Z}_n^*$ such that $x \equiv y^f \pmod{n}$ for a non-negligible fraction of $x \in \mathcal{Z}_n^*$, where $T = (T_i^{(0)}, T_i^{(1)}, \ldots, T_i^{(r_i)})$, $f_i^{(r_i)} = (f_i^{(0)}, f_i^{(1)}, \ldots, f_i^{(r_i)})$ $(1 \le i \le m)$. In addition, we use INV to denote any algorithm that, on input $n \in \mathcal{SC}_k$, $e \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$, and $x \in \mathcal{Z}_n^*$, outputs $y \in \mathcal{Z}_n^*$ such that $x \equiv y^e \pmod{n}$ for a non-negligible fraction of $x \in \mathcal{Z}_n^*$.

From technical reasons, we assume, throughout the rest of this paper, that each ID_i such that $ID_i^2 \not\equiv 1 \pmod{n}$ is randomly chosen (by the trusted center) with uniform probability over $\mathcal{Z}_n^*$, but once assigned it is unchanged forever.

Theorem 4.4: *Given an expected polynomial time algorithm $INV_{\text{SKU}}^{(m,R)}$, there exists an expected polynomial time algorithm INV using $INV_{\text{SKU}}^{(m,R)}$ as an oracle.*

Proof: It suffices to show that, given $n \in \mathcal{SC}_k$, $e = e'_0 e'_1 \cdots e'_{r_u}$, and $x \in \mathcal{Z}_n^*$ such that $x^2 \not\equiv 1 \pmod{n}$, there exists an expected polynomial time $INV\left(INV_{\text{SKU}}^{(m,R)}\right)$ using an expected polynomial time algorithm $INV_{\text{SKU}}^{(m,R)}$ as an oracle.

Let $E(|n|)$ and $U(|n|)$ be any fixed polynomials in $|n|$. Let $m < E(|n|)$ and let $R = (r_1, r_2, \ldots, r_m)$, where $r_i \le U(|n|)$ $(1 \le i \le m)$. Note that for $\langle n, ID_i, S_i^{(r_i)}, e_i^{(r_i)}\rangle$ $(1 \le i \le m)$, the oracle $INV_{\text{SKU}}^{(m,R)}$ returns a correct answer if $\langle n, ID_i, S_i^{(r_i)}, e_i^{(r_i)}\rangle \in \mathcal{C}_{\text{SKU}}$ for all i $(1 \le i \le m)$; it might return garbage or something otherwise.

Algorithm $INV\left(INV_{\text{SKU}}^{(m,R)}\right)$:

Input. $n \in \mathcal{SC}_k$; $e = e'_0 e'_1 \cdots e'_{r_u}$, where $e'_i \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ $(0 \le i \le r_u)$;
$x \in \mathcal{Z}_n^*$ such that $x^2 \not\equiv 1 \pmod{n}$.

Step 1. Set $\mathcal{R} := \phi$ and $i := 1$.

Step 2. Set $\mathcal{S}_i := \phi$ and $\ell_i = r_i$ and choose $S_i^{(r_i)} \in \mathcal{Z}_n^*$ such that $\left\{S_i^{(r_i)}\right\}^2 \not\equiv 1 \pmod{n}$.

Step 3. Choose $(r_i - \ell_i + 1)$ distinct $d_j \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$ such that $d_j \notin \mathcal{S}_i$, using primality testing. (see, e.g., [AH], [Ra], [SS].)

Step 4. For each $d_j \notin \mathcal{S}_i$ $(1 \le j \le r_i - \ell_i + 1)$, compute $S_i^{(\ell_i - 1)} \equiv \left\{S_i^{(\ell_i)}\right\}^{d_j} \pmod n$ until $S_i^{(\ell_i-1)} \not\equiv S_i^{(\ell)} \pmod n$ for all ℓ $(\ell_i \le \ell \le r_i)$.

Step 5. Set $e_i^{(\ell_i)} := d_j$ and $\mathcal{S}_i := \mathcal{S}_i \cup e_i^{(\ell_i)}$.

Step 6. If $\ell_i > 0$, then $\ell_i := \ell_i - 1$ and go to **Step 3**.

Step 7. Set $ID_i := S_i^{(-1)}$.

Step 7-1. If $ID_i \in \mathcal{R}$, then go to **Step 2**; otherwise set $\boldsymbol{S}_i^{(r_i)} = (S_i^{(0)}, S_i^{(1)}, \ldots, S_i^{(r_i)})$ and $\boldsymbol{e}_i^{(r_i)} = (e_i^{(0)}, e_i^{(1)}, \ldots, e_i^{(r_i)})$.

Step 7-2. If $i < m$, then $\mathcal{R} := \mathcal{R} \cup ID_i$, $i := i + 1$ and go to **Step 2**.

Step 8. Run the algorithm (or oracle) $INV_{\text{SKU}}^{(m,R)}$ on input $n \in \mathcal{SC}_k$, $e = e_0' e_1' \cdots e_{r_u}'$, $x \in \mathcal{Z}_n^*$ such that $x^2 \not\equiv 1 \pmod n$, and $\langle n, ID_i, \boldsymbol{S}_i^{(r_i)}, \boldsymbol{e}_i^{(r_i)} \rangle$ $(1 \le i \le m)$.

Output. $y \in \mathcal{Z}_n^*$ such that $x \equiv y^e \pmod n$.

Trivially, **Step 1** (resp. **Step 2**) runs in deterministic (resp. expected) polynomial time. From Lemma 4.1 and the facts that deciding primality is in $\mathcal{ZPP}$ (see [AH], [Ra], [SS].) and $(r_i - \ell_i + 1) \le U(|n|) + 1$, it follows that **Step 3** runs in expected polynomial time. From Lemmas 4.2 and 4.3, there must exist at least one $d_j \notin \mathcal{S}_i$ such that for all ℓ $(\ell_i \le \ell \le r_i)$, $S_i^{(\ell_i-1)} \not\equiv S_i^{(\ell)} \pmod n$, thus **Step 4** runs in deterministic polynomial time. Since **Step 5** runs in deterministic polynomial time and the iteration times of a loop from **Step 3** to **6** is $r_i + 1 \le U(|n|) + 1$, then the total running cost of the loop from **Step 3** to **6** is expected polynomial time.

In **Step 7-1**, the probability that $ID_i \in \mathcal{R}$ is negligibly small, because possible candidates of ID_i is *exponentially* many, while $\|\mathcal{R}\|$ is *polynomially* bounded, i.e., $\|\mathcal{R}\| \le m < E(|n|)$, and then the expected iteration times from **Step 2** to **7-1** or **7-2** is $O(E(|n|))$. Hence the algorithm $INV\left(INV_{\text{SKU}}^{(m,R)}\right)$ runs in expected polynomial time and outputs $y \in \mathcal{Z}_n^*$ such that $y \equiv x^e \pmod n$ for a non-negligible fraction of $x \in \mathcal{Z}_n^*$. □

Informally, Theorem 4.4 shows that when any polynomially many entities conspire in **SKU** even in polynomially many times key-updating, they can not invert $x \in \mathcal{Z}_n^*$ for a non-negligible fraction of $x \in \mathcal{Z}_n^*$. From the definition of *soundness* [FFS], [TW], this implies that in **SKU** any polynomially many conspiring entities can not misrepresent themselves for a non-negligible fraction of (possible) other entities, even in polynomially many times key-updating. A result similar to this holds for **PKU**.

Theorem 4.5: *Given an expected polynomial time algorithm $INV_{\text{PKU}}^{(m,R)}$, there exists an expected polynomial time algorithm INV using $INV_{\text{PKU}}^{(m,R)}$ as an oracle.*

Proof: It suffices to show that, given $n \in \mathcal{SC}_k$, $f \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$, and $x \in \mathcal{Z}_n^*$ such that $x^2 \not\equiv 1 \pmod n$, there exists an expected polynomial time $INV\left(INV_{\mathrm{PKU}}^{(m,R)}\right)$ using an expected polynomial time algorithm $INV_{\mathrm{PKU}}^{(m,R)}$ as an oracle.

Let $E(|n|)$, $U(|n|)$, m, and $R = (r_1, r_2 \ldots, r_m)$ be defined in the same way as the proof of Theorem 4.4. It should be noted that for $\langle n, ID_i, T_i^{(r_i)}, f_i^{(r_i)} \rangle$ $(1 \le i \le m)$, the oracle $INV_{\mathrm{PKU}}^{(m,R)}$ returns a correct answer if $\langle n, ID_i, T_i^{(r_i)}, f_i^{(r_i)} \rangle \in \mathcal{C}_{\mathrm{PKU}}$ for all i $(1 \le i \le m)$; it might return garbage or something otherwise.

Algorithm $INV\left(INV_{\mathrm{PKU}}^{(m,R)}\right)$:

Input. $n \in \mathcal{SC}_k$; $f \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$; $x \in \mathcal{Z}_n^*$ such that $x^2 \not\equiv 1 \pmod n$.

Step 1. Run each **Step** from **1** to **7** in the algorithm $INV\left(INV_{\mathrm{SKU}}^{(m,R)}\right)$.

Step 2. Run the algorithm $A_{\mathrm{SKU}\to\mathrm{PKU}}$ on input $\langle n, ID_i, S_i^{(r_i)}, e_i^{(r_i)} \rangle \in \mathcal{C}_{\mathrm{SKU}}$ for each i $(1 \le i \le m)$, and output $\langle n, ID_i, T_i^{(r_i)}, f_i^{(r_i)} \rangle \in \mathcal{C}_{\mathrm{PKU}}$.

Step 3. Run the algorithm (oracle) $INV_{\mathrm{PKU}}^{(m,R)}$ on input $n \in \mathcal{SC}_k$, $f \in \mathcal{OP}(\lfloor\sqrt{n}/4\rfloor)$, $x \in \mathcal{Z}_n^*$ such that $x^2 \not\equiv 1 \pmod n$, and $\langle n, ID_i, T_i^{(r_i)}, f_i^{(r_i)} \rangle$ $(1 \le i \le m)$.

Output. $y \in \mathcal{Z}_n^*$ such that $x \equiv y^f \pmod n$.

From the proof of Theorem 4.4, it follows that **Step 1** runs in expected polynomial time. Since $m < E(|n|)$ and Theorem 3.5 guarantees that the algorithm $A_{\mathrm{SKU}\to\mathrm{PKU}}$ runs in deterministic polynomial time, **Step 2** runs in deterministic polynomial time.

Hence the algorithm $INV\left(INV_{\mathrm{SKU}}^{(m,R)}\right)$ runs in expected polynomial time and outputs $y \in \mathcal{Z}_n^*$ such that $y \equiv x^f \pmod n$ for a non-negligible fraction of $x \in \mathcal{Z}_n^*$. □

5 Conclusion and Remarks

In this paper, we showed two kinds of secure key-updating schemes **SKU** and **PKU** in the extended Fiat-Shamir scheme. Here we define more general schemes **SKU'** and **PKU'**:

Let $n \in \mathcal{SC}_k$ and let $E(|n|)$ and $U(|n|)$ be any fixed polynomial in $|n|$. Then the key-updating scheme **SKU'** is completely the same as **SKU** except that for each entity i, $e_i^{(j)} < \lfloor\sqrt{n}/4\rfloor$ $(0 \le j \le r_i \le U(|n|))$ is an odd number and is not necessarily distinct from each other, and the key-updating scheme **PKU'** is also completely the same as **PKU** except that for each entity i, $f_i^{(j)} \not| f_i^{(k)}$ $(0 \le k < j \le r_i)$. Using a technique similar to the proofs of Theorems 4.4 and 4.5, we can show that both **SKU'** and **PKU'** are provably secure if decrypting RSA is hard for $n \in \mathcal{SC}_k$.

Observing the results in this paper, we can say that **SKU** and **PKU** have the same security with each other in a polynomial time sense, and seemingly so do **SKU'** and **PKU'**. The scheme **PKU**, however, seems to be better one than **SKU** in the light of efficiency, because in the r_i-th ($1 \leq r_i \leq U(|n|)$) key-updating of **PKU**, a public-key of each entity i is only a prime $f_i^{(r_i)}$, while in the r_i-th ($1 \leq r_i \leq U(|n|)$) key-updating of **SKU**, a public-key of each entity i is $\prod_{j=0}^{r_i} e_i^{(j)}$. This is also the case for **PKU'** and **SKU'**.

Our results can be generalized to more theoretical form — For any transitive trapdoor random self-reducible uniform relation (see [IST].), there exists a perfect zero-knowledge (identity-based) identification system with provably secure key-updating schemes, i.e., if any polynomially many entities conspire in polynomially many times key-updating, they can not find a secret-key of a non-negligible fraction of (possible) other entities, or they can not misrepresent themselves for a non-negligible fraction of (possible) other entities.

Acknowledgments

The authors wish to thank Mitsunori Ogiwara, Yuliang Zheng, Kouichi Sakurai, Kenji Koyama, Tsutomu Matsumoto, and Hiroki Shizuya for their valuable comments and suggestions. The authors also would like to thank Osamu Watanabe for his cooperative and suggestive discussion on the early version of this work.

References

[AH] Adleman, L.M. and Huang, M.D.A., "Recognizing Primes in Random Polynomial Time," Proc. of 19*th* Annual ACM Symposium on Theory of Computing, pp.462-469 (May, 1987).

[AL] Angluin, D. and Lichtenstein, D., "Provable Security of Cryptosystems: a Survey," Technical Report TR-288, Yale University (October, 1983).

[FFS] Feige, U., Fiat, A., and Shamir, A., "Zero Knowledge Proofs of Identity," *Journal of Cryptology*, Vol.1, No.1, pp.77-94 (1988).

[FS] Fiat, A. and Shamir, A., "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," in *Advances in Cryptology* — Crypto'86, Lecture Notes in Computer Science 263, Springer-Verlag, Berlin, pp.186-194 (1987).

[FT] Fujioka, A. and Tsujii, S., "An ID-Based Identification System with Simple Key-Updating," Technical Report of IEICE, ISEC89-25 (November, 1989).

[GQ] Guillou, L.C. and Quisquater, J.J., "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessors Minimizing both Transmission and Memory," in *Advances*

[GQ] Guillou, L.C. and Quisquater, J.J., "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessors Minimizing both Transmission and Memory," in *Advances in Cryptology* — Eurocrypt'88, Lecture Notes in Computer Science 330, Springer-Verlag, Berlin, pp.123-128 (1988).

[HW] Hardy, G.H. and Wright, E.M., *An Introduction to the Theory of Numbers*, Oxford University Press, 5*th* Edition (1979).

[IST] Itoh, T., Shinozaki, S., and Tsujii, S., "Secure Key-Updating Schemes in Identification Protocols," manuscript (March, 1990).

[Kr] Kranakis, E., *Primality and Cryptography*, Wiley-Teubner Series in Computer Science, John Wiley & Sons, Chishester (1986).

[OO] Ohta, K. and Okamoto, T., "A Modification of the Fiat-Shamir Scheme," in *Advances in Cryptology* — Crypto'88, Lecture Notes in Computer Science 403, Springer-Verlag, Berlin, pp.232-243 (1989).

[Ra] Rabin, M.O., "Probabilistic Algorithm for Primality Testing," *Journal of Number Theory*, Vol.12, pp.128-138 (1980).

[RSA] Rivest, R.L., Shamir, A., and Adleman, L.M., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communication of the ACM*, Vol.21, No.2, pp.120-126 (February, 1978).

[SI] Shizuya, H. and Itoh, T., "A Group-Theoretic Interface to Random Self-Reducibility," to appear in The Transactions of the IEICE, Vol.E 73, No.7 (July, 1990).

[SS] Solovay, R. and Strassen, V., "A Fast Monte Calro Test for Primality," *SIAM Journal on Computing*, Vol.6, No.1, pp.84-85 (March, 1977).

[TW] Tompa, M. and Woll, H., "Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information," Proc. of 28*th* Annual IEEE Symposium on Foundations of Computer Science, pp.472-482 (October, 1987).