PAPER

An Identification Scheme with Tight Reduction

Seiko ARITA[†a)], ***Member and*** **Natsumi KAWASHIMA**[†], ***Nonmember***

SUMMARY There are three well-known identification schemes: the Fiat-Shamir, GQ and Schnorr identification schemes. All of them are proven secure against the passive or active attacks under some number-theoretic assumptions. However, efficiencies of the reductions in those proofs of security are not tight, because they require "rewinding" a cheating prover. We show an identification scheme IDKEA1, which is an enhanced version of the Schnorr scheme. Although it needs the four exchanges of messages and slightly more exponentiations, the IDKEA1 is proved to be secure under the KEA1 and DLA assumptions with *tight reduction*. The idea underlying the IDKEA1 is to use an *extractable commitment* for prover's commitment. In the proof of security, the simulator can open the commitment in two different ways: one by the non-black-box extractor of the KEA1 assumption and the other through the simulated transcript. This means that we don't need to rewind a cheating prover and can prove the security without loss of the efficiency of reduction.

key words: *identification scheme, rewinding, KEA1 assumption, tight reduction*

1. Introduction

1.1 Zero-Knowledge Identification Schemes

The zero-knowledge identification scheme is a triple $(\mathcal{K}, P, V)$ of probabilistic polynomial-time algorithms. A key-generator $\mathcal{K}$ generates a pair (pk, sk) of public and private keys on input of the security parameter k. A prover P with the secret key sk (and the public key pk) proves its identity to a verifier V (with the public key pk) through interactions showing its possession of sk in (honest-verifier) zero-knowledge.

The major security goal of identification schemes is to prevent an adversary A with no secret key sk from impersonating the authentic prover P. Such an adversary A is called passive if A only eavesdrops the message-flow between honest P and V (to impersonate P after that). If A acts as a cheating prover or verifier beyond eavesdropping, A is called an active adversary. In particular, if A can act as a cheating verifier concurrently against plural prover clones with the same secret key, it is called a concurrent attack, in which interest has been growing.

There are three well-known identification schemes: the Fiat-Shamir [6], GQ [9] and Schnorr [11] identification schemes. The Fiat-Shamir scheme is proven to be secure against impersonation by an active adversary based on the difficulty of the integer factorization problem. However, it needs rather long secret keys. The GQ identification scheme is an extension of the Fiat-Shamir scheme, which reduces both the number of messages exchanged and memory requirements for secret keys. The GQ identification scheme is proven to be secure against the passive and concurrent attacks under the RSA and One-More-Inversion assumptions, respectively [3]. The Schnorr identification scheme is an alternative to the Fiat-Shamir and GQ schemes. It is also proven to be secure against the passive and concurrent attacks under the DLA (Discrete Logarithmic Assumption) and One-More-DL assumptions, respectively [3].

1.2 Provable Security of Identification Schemes

Let us briefly recall how the proof of the security against impersonation does work in the case of the Schnorr scheme. In the Schnorr scheme, P has a secret key x and V has a public key $q, g, h(= g^x)$. First, P randomly chooses a from $\mathbb{Z}_q$, computes a commitment $t = g^a$ and sends t to V, which, in turn, randomly chooses a challenge c from $\mathbb{Z}_q$ and sends c to P. Then, P responds $y = a + xc$ to V. Finally, V sees whether y can correctly open $h^c t$ or not, that is, it checks the equality of $g^y = h^c t$.

Suppose there is an adversary A that can impersonate a prover in the Schnorr scheme with a non-negligible success probability. Using A we can construct the following simulator S which computes the discrete logarithm x of a given element $h(= g^x)$. A simulator S invokes a copy of A, gives h to A as a public key of a prover in the Schnorr scheme, and plays the role of a verifier against A. That is, the simulator S, receiving a cheating commitment t^* from A, sends a random challenge c to A and gets a cheating response y^*. Then, we have $g^{y^*} = h^c t^*$ with probability of A's success. Now, S rewinds A to the point receiving a challenge and sends a new random challenge c_1 once more to get a new response y_1^* from A. We have $g^{y_1^*} = h^{c_1} t^*$ also with probability of A's success. Using the two equations, S can compute the discrete logarithm x of h by $x = (y^* - y_1^*)(c - c_1)^{-1}$ with probability of the *square* of A's success. Thus, we have the following relation between the advantage $\mathbf{Adv}^{\mathrm{imp}}_{ID,A}$ of A against ID and the advantage $\mathbf{Adv}^{\mathrm{dl}}_{G,S}$ of S against DLP (Discrete Logarithmic Problem) (on $G = \langle g \rangle$):

$$\mathbf{Adv}^{\mathrm{imp}}_{ID,A}(k) \;\leq\; \sqrt{\mathbf{Adv}^{\mathrm{dl}}_{G,S}(k)} + \eta(k)$$

Manuscript received November 17, 2006.
Manuscript revised March 5, 2007.
Final manuscript received May 16, 2007.

[†]The authors are with Institute of Information Security, Yokohama-shi, 221-0835 Japan.
a) E-mail: arita@iisec.ac.jp

DOI: 10.1093/ietfec/e90–a.9.1949

with some negligible function η (of security parameter k). This means a contradiction to DLA. Here, the running time t_S of S is around the twice the running time t_A of A: $t_S(k) \leq 2t_A(k) + O(k^3)$.

The above (standard) proof depends on the well-known technique "rewinding to extract." As seen above, the technique sacrifices efficiency of the reduction. The probability to extract the secret is only the square of probability of the successful attack.

The similar situation holds also for the Fiat-Shamir and GQ schemes.

1.3 Our Results

We show an identification scheme IDKEA1, which is an enhanced version of the Schnorr scheme. Although it needs four messages exchanged and slightly more exponentiations for both a prover and a verifier than the Schnorr scheme, the IDKEA1 is proved to be secure under the two assumptions of KEA1 [4], [10] and DLA with *tight reduction*. Here, by the term "tight reduction" under two assumptions A_1 and A_2 we mean a reduction in which an adversary who breaks the scheme with probability ϵ in time t can be used to break the underlying problems of the assumption A_i with probability ϵ_i in time t_i ($i = 1, 2$), and we have $\epsilon \approx \epsilon_1 + \epsilon_2$ and $t \leq \mathrm{Min}(O(t_1), O(t_2))$.

The idea underlying the IDKEA1, which is inspired by Barak's generic non-back-box techniques [1], [2], is to use an *extractable commitment* for prover's commitment. The extractable commitment is actually extractable only by the simulator who can use the non-black-box extractor of the KEA1 assumption. In the proof of security, the simulator can open the commitment in two different ways: one by the non-black-box extractor and the other through the simulated transcript. This means that we don't need to depend on the rewind technique and can prove the security without loss of the efficiency of reduction.

Our first theorem is as follows.

Theorem 1 *If the generator G is both (t', ϵ')-DLA and (t'', ϵ'')-KEA1, then the* IDKEA1 *scheme ID with the generator G is (t, ϵ)-secure under the passive attack with*

$$t \leq \mathrm{Min}\left\{\frac{1}{2}(t' - 6.4t_{\mathrm{exp}}),\ t''\right\}$$
$$\epsilon \geq \epsilon' + \epsilon''$$

(t_{exp} denotes the time to compute an exponentiation in the group generated by G).

In addition, using a variant OMDL+ of the OMDL assumption [3], we can prove the IDKEA1 is secure even under the concurrent attack also with tight reduction:

Theorem 2 *If the generator G is both $(t', n+1, \epsilon')$-OMDL+ and (t'', ϵ'')-KEA1, then the* IDKEA1 *scheme ID with the generator G is (t, n, ϵ)-secure under the concurrent attack with*

$$t \leq \mathrm{Min}\left\{\frac{1}{2}(t' - (4.2 + n)t_{\mathrm{exp}}),\ t''\right\}$$
$$\epsilon \geq \epsilon' + \epsilon''.$$

1.4 Related Works

A signature scheme whose security can be tightly reduced to difficulty of the discrete logarithm problem in the standard model is proposed by Cramer and Damgard [5]. [5] built the signature scheme based on Σ-protocol, which can be viewed as a generalization of identification schemes treated in the presented paper. However, note that the security of Σ-protocol itself is not tightly reduced to difficulty of the discrete logarithm problem in [5]. Our aim here is at the security of an identification scheme itself, not at the resulting signature scheme.

Bellare and Palacio [4] show a 3-Round Zero-Knowledge protocol in which the KEA3 assumption (a variant of KEA1) is used to prove its soundness. The role played by the KEA3 assumption is different from ours. In fact, the proof of the soundness in [4] needs the rewinding technique to extract the secret of the cheating prover.

Fischlin [7] shows a non-interactive proof of knowledge with online extractors. The online extractor plays the similar role as extractors (without rewinding) in the proof of our scheme. The online extractor needs the random oracle model and unfortunately the communication complexity (i.e., the length of the proof) in the scheme is rather high although it can be said feasible. Our scheme can be viewed as an interactive and practical (but restricted) version of [7] based on the non-black-box assumption instead of the random oracle.

2. Definitions and Assumptions

In this section, following [3], we state the definitions of the security of identification schemes under the passive and concurrent attacks and introduce the KEA1 and OMDL+ assumptions.

2.1 Security Definitions of Identification Schemes

Let a triple $ID = (\mathcal{K}, P, V)$ of probabilistic polynomial-time algorithms be an identification scheme. A key-generator $\mathcal{K}$ generates a pair (pk, sk) of public and private keys on input of the security parameter k. A prover P with the secret key sk (and the public key pk) proves its identity to verifier V (with the public key pk) through interactions showing its possession of sk.

The security of an identification scheme ID under the passive attack is defined as follows. In the following A_1 acts as an eavesdropper of conversations between an honest prover and an honest verifier. After halting A_1 with an output St, A_2 tries to impersonate the prover using St.

Definition 1 (Security under passive attacks) *Let a triple*

$ID = (\mathcal{K}, P, V)$ of probabilistic polynomial-time algorithms be an identification scheme. Let $A = (A_1, A_2)$ be any probabilistic polynomial-time adversary. For ID and A, an experiment $\mathbf{Exp}^{\mathsf{imp-pa}}_{ID,A}$ is defined as follows (λ denotes the empty string).

$\mathbf{Exp}^{\mathsf{imp-pa}}_{ID,A}$:
- $(pk, sk) \leftarrow \mathcal{K}$;
- *Invoke* $A_1(pk)$;
  - *When A_1 makes a query λ, reply with an honest transcript between $P(sk, pk)$ and $V(pk)$;*
- *Let St be an output of* A_1;
- *Invoke* $A_2(St)$;
  - *Play the role of honest verifier $V(pk)$ to $A_2(St)$;*
- *Output* 1 *if the V accepts; otherwise output* 0.

In the above, A is assumed to make a query λ at most once.

The advantage of adversary A against ID under the passive attack is defined as

$$\mathbf{Adv}^{\mathsf{imp-pa}}_{ID,A} = \mathbf{Pr}[\mathbf{Exp}^{\mathsf{imp-pa}}_{ID,A} = 1].$$

ID is called (t, ϵ)-secure under the passive attack *if for any probabilistic polynomial-time adversary A that runs in time at most t, the advantage $\mathbf{Adv}^{\mathsf{imp-pa}}_{ID,A}$ is upper bounded by ϵ.*

The security of an identification scheme ID under the concurrent attack is defined as follows. In the following A_1 acts as a cheating verifier that can take place in concurrent sessions with plural prover clones $P_i(sk)$ with the same secret key sk. Those sessions can be interleaved in any ways. After halting A_1 with an output St, A_2 with St tries to impersonate the prover $P(sk)$.

Definition 2 (Security under concurrent attacks) *Let a triple $ID = (\mathcal{K}, P, V)$ of probabilistic polynomial-time algorithms be an identification scheme. Let $A = (A_1, A_2)$ be any probabilistic polynomial-time adversary. For ID and $A = (A_1, A_2)$, an experiment $\mathbf{Exp}^{\mathsf{imp-ca}}_{ID,A}$ is defined as follows.*

$\mathbf{Exp}^{\mathsf{imp-ca}}_{ID,A}$:
- $(pk, sk) \leftarrow \mathcal{K}$;
- *Invoke* $A_1(pk)$;
  - *Play the role of prover clones $P_i(sk)$ concurrently to A_1;*
- *Let St be an output of* A_1;
- *Invoke* $A_2(St)$;
  - *Play the role of honest verifier $V(pk)$ to A_2;*
- *Output* 1 *if the V accepts; otherwise output* 0.

(In the above, A_1 is assumed to play the role of verifier at most once with each of prover clone P_i. So, if there are n prover clones invoked, A_1 plays the verifier at most n times.) The advantage of adversary A against ID under the concurrent attack is defined as

$$\mathbf{Adv}^{\mathsf{imp-ca}}_{ID,A} = \mathbf{Pr}[\mathbf{Exp}^{\mathsf{imp-ca}}_{ID,A} = 1].$$

ID is called (t, n, ϵ)-secure under the concurrent attack *if for any probabilistic polynomial-time adversary A that runs in time at most t, and makes interactions with at most n prover clones, the advantage $\mathbf{Adv}^{\mathsf{imp-ca}}_{ID,A}$ is upper bounded by ϵ.*

2.2 Assumptions on Groups

We use three number-theoretic assumptions on groups: the DLA, KEA1 and OMDL+ assumptions. The DLA is the standard Discrete Logarithmic Assumption. We use the DLA in the concrete manner as follows. The group generator G that outputs a generator g of a group of order q, is called to satisfy (t, ϵ)-*DLA* if for any adversary A that runs in time at most t, we have

$$\mathbf{Pr}[(q, g) \leftarrow G; x \stackrel{\$}{\leftarrow} \mathbb{Z}_q; y = g^x; \hat{x} \leftarrow A(q, g, y) \mid x = \hat{x}] < \epsilon.$$

The probability is taken over the coins of G, randomness choosing x and the coins of A as usual.

The definitions of the KEA1 and OMDL+ assumptions are as follows.

2.2.1 The KEA1 Assumption

The KEA1 assumption [4], [10] for a group $G = \langle g \rangle$ means that it is possible only when one knows b to generate a DH-pair (g^b, g^{ab}) for a randomly selected g^a.

Definition 3 (The KEA1 Assumption [4]) *Let G be a probabilistic polynomial-time algorithm which on input of the security parameter k, outputs a prime number q of k bits and a generator g of a group of order q. Let H be any probabilistic polynomial-time algorithm which on input of $q, g, A (\in \langle g \rangle)$ and an auxiliary input w, outputs a pair (B, W) of elements in G. Let H^* be an extractor for H, that is, any probabilistic polynomial-time algorithm which on input of q, g, A and an auxiliary input w, outputs b.*

For any string w and such G, H, H^*, an experiment $\mathbf{Exp}^{w}_{G,H,H^*}$ is defined as follows.

$\mathbf{Exp}^{w}_{G,H,H^*}$:
- $(q, g) \leftarrow G(1^k)$; $a \stackrel{\$}{\leftarrow} \mathbb{Z}_q$; $A = g^a$;
- $(B, W) \leftarrow H(q, g, A, w)$;
- $b \leftarrow H^*(q, g, A, w)$;
- *If* $W = B^a$ *and* $B \neq g^b$ *then return* 1;
- *Otherwise return* 0.

Then, the advantage of adversary H in G for extractor H^* is defined as

$$\mathbf{Adv}^{w}_{G,H,H^*}(k) = \mathbf{Pr}[\mathbf{Exp}^{w}_{G,H,H^*}(k) = 1].$$

G is called to satisfy (t, ϵ)-KEA1 *if any adversary H that runs in time at most t, there exists an extractor H^* that runs also in time at most t and the $\mathbf{Adv}^{w}_{G,H,H^*}$ is upper bounded by ϵ for any w.*

The KEA1 assumption in [4] is stated only in terms of the asymptotic behavior. The above concrete version of the definition seems to be natural with respect to the intrinsic meaning of the assumption.

2.2.2 The OMDL+ Assumption

The OMDL+ assumption is a stronger version of the OMDL assumption used in [3]. The OMDL assumption means that it is difficult to solve one more DLP (Discrete Logarithmic Problem) instance even if one is provided with several randomly selected DLP instances with their answers, all sharing the same base element. The OMDL+ assumption is stronger in the sense that the challenge problems are given with some hints.

Definition 4 (OMDL+ Assumption) *Let G be a probabilistic polynomial-time algorithm which on input of the security parameter k, outputs a prime number q of k bits and a generator g of a group of order q. Let I be any probabilistic polynomial-time algorithm which on input of q, g outputs $x_1, \ldots, x_n$ using the challenge oracle $\mathcal{CO}$ and the DL oracle $\mathcal{DL}$. The challenge oracle $\mathcal{CO}_g$ given query h answers with a pair of g^x and h^x, where x is randomly chosen from $\mathbb{Z}_q$ independently by every query. The DL oracle $\mathcal{DL}_g$, given query h, answers with x satisfying $h = g^x$.*

For such G and I, an experiment $\mathbf{Exp}^{\mathsf{omdl+}}_{G,I}$ is defined as follows.

$\mathbf{Exp}^{\mathsf{omdl+}}_{G,I}$:
  $(q, g) \leftarrow G(1^k)$;
  Invoke $I^{\mathcal{CO}_g, \mathcal{DL}_g}(q, g)$;
    When I makes a query h to $\mathcal{CO}_g$,
    let $\mathcal{CO}_g$ reply with such (g_i, h_i);
    When I makes a query h to $\mathcal{DL}_g$,
    let $\mathcal{DL}_g$ reply with $\log_g(h)$;
  Let $(x_1, \ldots, x_n)$ be an output of I;
  Let $(g_1, h_1), \ldots, (g_n, h_n)$ be challenges issued by $\mathcal{CO}_g$;
  Let m be the number of answers given by $\mathcal{DL}_g$;
  If $g_i = g^{x_i}$ for all $i = 1, \ldots, n$ and $m < n$
    then return 1;
  Otherwise return 0.

The advantage of adversary I against G is defined as

$$\mathbf{Adv}^{\mathsf{omdl+}}_{G,I} = \mathbf{Pr}[\mathbf{Exp}^{\mathsf{omdl+}}_{G,I} = 1].$$

G is called to satisfy (t, n, ϵ)-OMDL+ if for any probabilistic polynomial-time adversary I that runs in time at most t and makes at most n queries to the challenge oracle $\mathcal{CO}$, the advantage $\mathbf{Adv}^{\mathsf{omdl+}}_{G,I}$ is upper bounded by ϵ.

It is easily seen that the OMDL+ assumption means the OMDL assumption and that the OMDL assumption means the CDH assumption or the OMDL+ assumption.

3. The Identification Scheme IDKEA1

We describe our identification scheme IDKEA1= $\{\mathcal{K}, P, V\}$. Let G be a probabilistic polynomial-time algorithm which given the security parameter k outputs a prime number q of k bits and a generator g of a group of order q.

A key-generation algorithm $\mathcal{K}$ of the IDKEA1 on input k runs $G(k)$ to get q, g_1, chooses x randomly from $\mathbb{Z}_q$, computes $h_1 = g_1^x$ and outputs x and q, g_1, h_1 as a secret key and a public key, respectively.

In the IDKEA1, a prover P (with a secret key x) proves its identity to a verifier V (with a public key q, g_1, h_1) as follows.

1° V randomly selects a from $\mathbb{Z}_q$ and computes $g_2 = g_1^a$. V sends g_2 to P.
2° P randomly selects m_0 from $\mathbb{Z}_q$ and computes $c_1 = g_1^{m_0}$, $c_2 = g_2^{m_0}$. P sends c_1, c_2 to V.
3° V sees whether $c_2 = c_1^a$ or not. If not, V aborts. Otherwise V randomly selects r from $\mathbb{Z}_q$ and sends r to P.
4° P computes $m = m_0 - rx$ and sends m to V.
5° V sees whether $c_1 = g_1^m h_1^r$ does hold or not. If it does, V accepts. Otherwise V rejects.

As seen above, the IDKEA1 needs two exponentiations for a prover and two exponentiations and a two-exponent multi-exponentiation for a verifier, and it needs four messages exchanged. (Assuming (as in [8]) that a two-exponent multi-exponentiation takes $1.2\,t_{\mathrm{exp}}$, the time for a verifier is dominated by $3.2\,t_{\mathrm{exp}}$, where t_{exp} denotes the time to compute an exponentiation.) Thus, the IDKEA1 is not so efficient as the Schnorr scheme in computations and communications. However, the IDKEA1 has the security proof with tight reduction without loss of the security.

4. Security of the IDKEA1

We prove the security of the IDKEA1. In the proof, the simulator can open the cheating prover's commitment in two different ways: one by the non-black-box extractor of the KEA1 assumption and the other through the simulated transcript. This means we don't need to rewind the cheating provers.

4.1 Security under the Passive Attack

Security of the IDKEA1 under the passive attack is proven under the DLA and KEA1 assumptions with tight reduction.

Theorem 1 *If the generator G is both (t', ϵ')-DLA and (t'', ϵ'')-KEA1, then the IDKEA1 scheme ID with the generator G is (t, ϵ)-secure under the passive attack with*

$$t \leq \mathrm{Min}\left\{\frac{1}{2}(t' - 6.4t_{\mathrm{exp}}),\ t''\right\}$$
$$\epsilon \geq \epsilon' + \epsilon''$$

(t_{exp} denotes the time to compute an exponentiation in the group generated by G).

Proof Assume we have a passive adversary $A = (A_1, A_2)$ against $ID = \{\mathcal{K}, P, V\}$, running in time at most t, which succeeds in impersonating the honest prover P with probability at least ϵ. We use A to construct a KEA1-adversary H

running in time at most t'' and we use A and H to construct a DL-extractor E running in time at most t' with the advantage $\mathbf{Adv}_E$ at least $\epsilon - \epsilon''(\geq \epsilon')$. The stated result follows.

Let q, g_1, h_1 be generated as in $\mathcal{K}$:

$$q, g_1 \leftarrow G;\ x \xleftarrow{\$} \mathbb{Z}_q;\ h_1 = g_1^x.$$

On inputs q, g_1, h_1, the DL-extractor E proceeds as follows.

1° E starts the adversary A_1 with inputs q, g_1, h_1. When A_1 makes a query ϵ, E computes

$$a' \xleftarrow{\$} \mathbb{Z}_q;\ g_2' = g_1^{a'}$$
$$m', r' \xleftarrow{\$} \mathbb{Z}_q$$
$$c_1' = g_1^{m'} h_1^{r'};\ c_2' = {c_1'}^{a'}$$

and answers A_1 with a transcript $(g_2', (c_1', c_2'), r', m')$. Note the simulated transcript is distributed just as the real one between an honest P and V. Suppose A_1 halts and outputs a string St.

2° E starts the adversary A_2 with the input St and with random coins R. E randomly selects a from $\mathbb{Z}_q$, computes $g_2 = g_1^a$, and gives g_2 to A_2. Suppose A_2 replies with the message c_1^*, c_2^*. If ${c_1^*}^a \neq c_2^*$, then E aborts. Otherwise E randomly selects r from $\mathbb{Z}_q$ and sends r to A_2. Then, A_2 is supposed to output m^* and halt. If $c_1^* \neq g_1^{m^*} h_1^r$, E aborts. Note the messages given to A_2 are distributed just as the real ones given by an honest V.

3° Consider the following KEA1 adversary H on inputs (of the above) q, g_1, g_2 and $w = St, R$:

KEA1 adversary $H(q, g_1, g_2, (St, R))$:
  Invoke $A_2(St; R)$;
  Give g_2 to A_2;
  Get c_1^*, c_2^* from A_2;
  Output c_1^*, c_2^*.

E invokes the corresponding extractor H^* to H on the same inputs and gets m_0^*:

$$m_0^* \leftarrow H^*(q, g_1, g_2, (St, R));$$

4° E outputs $\hat{x} = (m_0^* - m^*)r^{-1}$.

In the above, note that the extractor E opens the commitment c_1^* made by the adversary A_2 in the two different ways: the one is m_0^* obtained by the KEA1-extractor H^* and the other is m^* through the simulated transcript.

Now we evaluate the advantage $\mathbf{Adv}_E = \mathbf{Pr}[\hat{x} = x]$ of the DL extractor E. Let Imp be an event that A_2 successfully impersonates P in the above simulation by E and Ext be an event the equation $c_1^* = g_1^{m_0^*}$ does hold. Note that $\mathbf{Pr}[\mathsf{Imp}] \geq \epsilon$ and if Imp holds, we have

$$c_2^* = {c_1^*}^a,\ c_1^* = g_1^{m^*} h_1^r. \tag{1}$$

If Ext holds, we have

$$c_1^* = g_1^{m_0^*}. \tag{2}$$

By the second equation of Eq. (1) and Eq. (2), we see

$$x = (m_0^* - m^*)r^{-1} = \hat{x}.$$

Thus,

$$\begin{aligned}\mathbf{Adv}_E = \mathbf{Pr}[\hat{x} = x] &\geq \mathbf{Pr}[\mathsf{Imp} \wedge \mathsf{Ext}] \\ &\geq \mathbf{Pr}[\mathsf{Imp}] - \mathbf{Pr}[\neg\mathsf{Ext} \wedge \mathsf{Imp}].\end{aligned}$$

So,

$$\mathbf{Pr}[\mathsf{Imp}] \leq \mathbf{Adv}_E + \mathbf{Pr}[\neg\mathsf{Ext} \wedge \mathsf{Imp}]. \tag{3}$$

By the definition of Ext and Imp,

$$\mathbf{Pr}[\neg\mathsf{Ext} \wedge \mathsf{Imp}] \leq \mathbf{Pr}[c_2^* = {c_1^*}^a \wedge c_1^* \neq g_1^{m_0^*}] \leq \mathbf{Adv}_H. \tag{4}$$

Now since the running time of H is bounded by the running time of A_2, it is not greater than t''. So, by the assumption of G being (t'', ϵ'')-KEA1, we have

$$\mathbf{Adv}_H < \epsilon''. \tag{5}$$

Then, by Eqs. (3), (4) and (5), we have

$$\epsilon \leq \mathbf{Pr}[\mathsf{Imp}] \leq \mathbf{Adv}_E + \epsilon'',$$

and

$$\mathbf{Adv}_E \geq \epsilon - \epsilon''.$$

Here, as easily seen from the description of E, the running time $time(E)$ of E includes the running time of A, the running time of H (which is less than the one of A) and is otherwise dominated by the four exponentiations and the two two-exponent multi-exponentiations. Assuming (as in [8]) that a two-exponent multi-exponentiation takes time $1.2t_{\exp}$, we have $time(E) \leq 2 \cdot t + (4 + 2.4)t_{\exp} \leq t'$, as desired. □

4.2 Security under the Concurrent Attack

Under the OMDL+ and KEA1 assumptions, $\mathsf{IDKEA1}$ is proven to be secure even under the concurrent attack also with tight reduction.

Theorem 2 *If the generator G is both $(t', n+1, \epsilon')$-OMDL+ and (t'', ϵ'')-KEA1, then the* $\mathsf{IDKEA1}$ *scheme ID with the generator G is (t, n, ϵ)-secure under the concurrent attack with*

$$t \leq \mathrm{Min}\left\{\frac{1}{2}(t' - (4.2 + n)t_{\exp}),\ t''\right\}$$
$$\epsilon \geq \epsilon' + \epsilon''.$$

Proof Assume we have an adversary $A = (A_1, A_2)$ against $ID = \{\mathcal{K}, P, V\}$ in the concurrent attack, running in time at most t and making interactions with at most n prover clones, which succeeds in impersonating the honest prover

P with probability at least ϵ. We use A to construct a KEA-adversary H running in time at most t'' and we use A and H to construct an OMDL+ solver I that runs in time at most t', making at most $n+1$ queries to the challenge oracle, with the advantage $\mathbf{Adv}_I$ at least $\epsilon - \epsilon''(\geq \epsilon')$. The stated result follows.

Let q, g_1 be generated by

$$q, g_1 \leftarrow G(k).$$

On inputs q, g_1, the OMDL+ solver I proceeds as follows.

1° I randomly chooses a' from $\mathbb{Z}_q$ and computes $g_2' = g_1^{a'}$. I sends g_2' to the challenge oracle CO_{g_1} to get the response $h_1(= g_1^{x_0}), h_2'(= g_2'^{x_0})$. I starts the adversary A_1 with inputs q, g_1, h_1.

2° When A_1 sends the message $g_2^{*(i)}$ to some prover clone (in the i-th session), I forwards the message $g_2^{*(i)}$ to the challenge oracle CO_{g_1} and get the response $c_1^{(i)}(= g_1^{x_i}), c_2^{(i)}(= g_2^{*(i)x_i})$. I delivers $c_1^{(i)}, c_2^{(i)}$ to A_1. When A_1 sends the response $r^{*(i)}$ to the prover clone, I makes a query $c_1^{(i)} h_1^{-r^{*(i)}}$ to the DL oracle $\mathcal{DL}_{g_1}$ and gets the response $m^{(i)}$, which is transferred to A_1 by I. Suppose A_1 halts and outputs a string St after performing concurrently such n sessions ($i = 1, \ldots, n$) with the simulated prover clones. It is easy to see that the above simulation of prover clones for A_1 is perfect.

3° I starts the adversary A_2 with the input St and with random coins R. I randomly selects a from $\mathbb{Z}_q$, computes $g_2 = g_1^a$, and gives g_2 to A_2. Suppose A_2 replies with the message c_1^*, c_2^*. If $c_1^{*a} \neq c_2^*$, then I aborts. Otherwise I randomly selects r from $\mathbb{Z}_q$ and sends r to A_2. Then, A_2 is supposed to output m^* and halt. If $c_1^* \neq g_1^{m^*} h_1^r$, I aborts. Note the above simulation of the honest verifier is perfect for A_2.

4° Consider the following KEA1 adversary H on inputs (of the above) q, g_1, g_2 and $w = St, R$:

KEA1 adversary $H(q, g_1, g_2, (St, R))$:
Invoke $A_2(St; R)$;
Give g_2 to A_2;
Get c_1^*, c_2^* from A_2;
Output c_1^*, c_2^*.

I invokes the corresponding extractor H^* to H on the same inputs and gets m_0^*:

$$m_0^* \leftarrow H^*(q, g_1, g_2, (St, R));$$

5° I outputs $\hat{x}_0 = (m_0^* - m^*)r^{-1}$ and $\hat{x}_i = m^{(i)} + \hat{x}_0 r^{*(i)}$ for $i = 1, \ldots, n$.

In the above, note that the solver I opens the commitment c_1^* made by the adversary A_2 in the two different ways: the one is m_0^* obtained by the KEA1-extractor H^* and the other is m^* through the simulated transcript.

Now we evaluate the advantage $\mathbf{Adv}_I = \mathbf{Pr}[\hat{x}_i = x_i\ (i = 0, 1, \ldots, n)]$ of the OMDL+ solver I. Let Imp be an event that A_2 successfully impersonates P in the above simulation by I and Ext be an event the equation $c_1^* = g_1^{m_0^*}$ does hold. Note that $\mathbf{Pr}[\mathsf{Imp}] \geq \epsilon$ and if Imp holds, we have

$$c_2^* = c_1^{*a},\ c_1^* = g_1^{m^*} h_1^r. \tag{6}$$

If Ext holds, we have

$$c_1^* = g_1^{m_0^*}. \tag{7}$$

By the second equation of Eq. (6) and Eq. (7), we see

$$x_0 = (m_0^* - m^*)r^{-1} = \hat{x}_0,$$

and since ${g_1}^{m^{(i)}} = c_1^{(i)} h_1^{-r^{*(i)}}$, we have

$$x_i = \mathrm{DL}_{g_1}(c_1^{(i)}) = m^{(i)} + x_0 r^{*(i)} = \hat{x}_i.$$

Thus,

$$\begin{aligned} \mathbf{Adv}_I &= \mathbf{Pr}[\hat{x}_i = x_i\ (i = 0, 1, \ldots, n)] \\ &\geq \mathbf{Pr}[\mathsf{Imp} \wedge \mathsf{Ext}] \\ &\geq \mathbf{Pr}[\mathsf{Imp}] - \mathbf{Pr}[\neg\mathsf{Ext} \wedge \mathsf{Imp}] \end{aligned}$$

So,

$$\mathbf{Pr}[\mathsf{Imp}] \leq \mathbf{Adv}_I + \mathbf{Pr}[\neg\mathsf{Ext} \wedge \mathsf{Imp}]. \tag{8}$$

By the definition of Ext and Imp,

$$\mathbf{Pr}[\neg\mathsf{Ext} \wedge \mathsf{Imp}] \leq \mathbf{Pr}[c_2^* = c_1^{*a} \wedge c_1^* \neq g_1^{m_0^*}] \leq \mathbf{Adv}_H. \tag{9}$$

Now since the running time of H is bounded by the running time of A_2, it is not greater than t''. So, by the assumption of G being (t'', ϵ'')-KEA1, we have

$$\mathbf{Adv}_H < \epsilon''. \tag{10}$$

Then, by Eqs. (8), (9) and (10), we have

$$\epsilon \leq \mathbf{Pr}[\mathsf{Imp}] \leq \mathbf{Adv}_I + \epsilon'',$$

and

$$\mathbf{Adv}_I \geq \epsilon - \epsilon''.$$

Here, as easily seen from the description of I, the number of queries made by I to the challenge oracle is at most $n+1$ (one in generating the simulated public key and n in generating the commitments for A_1). The running time $time(I)$ of I includes the running time of A, the running time of H (which is less than the one of A) and is otherwise dominated by the (3+n) exponentiations and the one two-exponent multi-exponentiation. Assuming (as in [8]) that a two-exponent multi-exponentiation takes time $1.2t_{\text{exp}}$, we have $time(I) \leq 2 \cdot t + (4.2 + n)t_{\text{exp}} \leq t'$, as desired. □

5. Conclusion

The paper has shown an identification scheme IDKEA1 which is an enhanced version of the Schnorr scheme by making the prover's commitment extractable. Although it needs four exchanges of messages and slightly more exponentiations than the Schnorr scheme, IDKEA1 is proved to be secure under the KEA1 and DLA assumptions with tight reduction. Moreover, using the variant OMDL+ of the OMDL assumption, we proved IDKEA1 is secure even under the concurrent attack also with tight reduction.

References

[1] B. Barak, "How to go beyond the black-box simulation barrier," Proc. 42nd FOCS, pp.106–115, IEEE, 2001.

[2] B. Barak and Y. Lindell, "Strict polynomial-time in simulation and extraction," Proc. 34th Annual Symposium on Theory of Computing, ACM, pp.484–493, 2002.

[3] M. Bellare and A. Palacio, "GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks," Proc. Crypto 2002, LNCS 2442, pp.162–177, 2002.

[4] M. Bellare and A. Palacio, "The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols," Proc. Crypto 2004, LNCS 3152, pp.273–289, 2004.

[5] R. Cramer and I. Damgard, "Secure signature schemes based on interactive protocols," CRYPTO 1995, pp.297–310, 1995.

[6] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," Proc. Crypto'86, LNCS 263, pp.171–185, 1986.

[7] M. Fischlin, "Communication-efficient non-interactive proofs of knowledge with online extractors," CRYPTO 2005, pp.152–168, 2005.

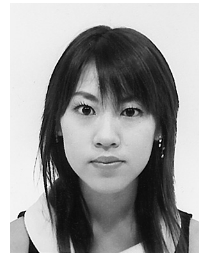
[8] E.-J. Goh and S. Jarecki, "A signature scheme as secure as the Diffie-Hellman problem," EUROCRYPT 2003, pp.401–415, 2003.

[9] L. Guillou and J.J. Quisquater, "A "paradoxical" identity-based signature scheme resulting from zero-knowledge," Proc. Crypto'88, LNCS 403, pp.216–231, 1988.

[10] S. Hada and T. Tanaka, "On the existence of 3-round zero-knowledge protocols," Proc. Crypto'98, LNCS 1462, pp.408–423, 1998.

[11] C.P. Schnorr, "Efficient signature generation by smart cards," J. Cryptol., vol.4, no.3, pp.161–174, 1991.

Seiko Arita received his B.E. and M.E. from Kyoto University, and Ph.D. from Chuo Universesity in 2000. He has been interested in prime numbers, algebraic curves and cryptography. He is with Institute of Information Security, Kanagawa, Japan. He is a member of JMS.



Natsumi Kawashima received her B.E. from Hosei University in 2004 and M.E. from Institute of Information Security in 2007.