2-2014

Identity based identification from algebraic coding theory

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Chik How TAN

Yi MU

Willy SUSILO

Duncan S. WONG

Theoretical Computer Science 520 (2014) 51–61

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Identity based identification from algebraic coding theory

Guomin Yang [a,*,1], Chik How Tan [b], Yi Mu [a], Willy Susilo [a,2], Duncan S. Wong [c,2]

[a] School of Computer Science and Software Engineering, University of Wollongong, Australia
[b] Temasek Laboratories, National University of Singapore, Singapore
[c] Department of Computer Science, City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 19 October 2012
Received in revised form 17 April 2013
Accepted 8 September 2013
Communicated by X. Deng

Keywords:
Identity based cryptography
Identification
Error-correcting codes
Syndrome decoding

ABSTRACT

Cryptographic identification schemes allow a remote user to prove his/her identity to a verifier who holds some public information of the user, such as the user public key or identity. Most of the existing cryptographic identification schemes are based on number-theoretic hard problems such as Discrete Log and Factorization. This paper focuses on the design and analysis of identity based identification (IBI) schemes based on algebraic coding theory. We first revisit an existing code-based IBI scheme which is derived by combining the Courtois–Finiasz–Sendrier signature scheme and the Stern zero-knowledge identification scheme. Previous results have shown that this IBI scheme is secure under passive attacks. In this paper, we prove that the scheme in fact can resist active attacks. However, whether the scheme can be proven secure under concurrent attacks (the most powerful attacks against identification schemes) remains open. In addition, we show that it is difficult to apply the conventional OR-proof approach to this particular IBI scheme in order to obtain concurrent security. We then construct a special OR-proof variant of this scheme and prove that the resulting IBI scheme is secure under concurrent attacks.

1. Introduction

Remote user identification is one of the fundamental research topics in cryptography, and is very useful in practice. We can separate public key user identification schemes into two categories: standard identification (SI), and identity based identification (IBI). In a standard identification scheme, the verifier has the public key of the prover and uses this public key to verify the genuineness of the remote user, while in an identity based identification scheme, the verifier can perform the verification just based on the prover's identity.

Most of the existing identification schemes follow a three-move (or Σ-type) structure: the prover P initiates an identification protocol by sending a *commitment* Cmt, then the verifier V replies with a *challenge* Ch, and finally P generates a *response* Rsp and sends it to V who makes a final *decision* which is either 'accept' or 'reject'. In [1], Bellare et al. called identification schemes following such a structure *canonical* identification schemes. Many canonical SI and IBI schemes have been proposed in the literature (e.g. [1,11,5,10,14,19,21,20]). The security of these schemes are based on the intractability of several number-theoretic problems such as factorization, discrete log, and RSA. One important application of canonical

* Corresponding author.
E-mail addresses: gyang@uow.edu.au (G. Yang), tsltch@nus.edu.sg (C.H. Tan), ymu@uow.edu.au (Y. Mu), wsusilo@uow.edu.au (W. Susilo), duncan@cityu.edu.hk (D.S. Wong).

[1] Part of the work was done when G. Yang was with Temasek Laboratories, National University of Singapore.

[2] W. Susilo is supported by the ARC Future Fellowship (FT0991397). D.S. Wong is supported by a grant from the RGC of the HKSAR, China (Project No. CityU 121512).

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.09.008

identification schemes is that we can derive a standard signature (SS) (or identity based signature (IBS), resp.) scheme from an SI (or IBI, resp.) scheme via the Fiat–Shamir transformation [11].

From SI/SS to IBI. In [1], Bellare, Namprempre and Neven presented a generic framework to transform any SI scheme satisfying certain conditions into an IBI scheme. The derived IBI scheme will inherit the security of the underlying SI scheme. Independent to Bellare et al.'s work, in [15], Kurosawa and Heng proposed another generic framework that transforms any standard signature scheme, which is existentially unforgeable under adaptive chosen message attacks [13], into an IBI scheme secure against passive adversaries. In [24], Yang et al. further showed that in order to achieve passive security, a standard signature scheme secure under *known* message attacks suffices.

Code-based Cryptography. The first code-based public key cryptosystem was proposed by McEliece [17] in 1978. A variant of the McEliece cryptosystem was later proposed by Niederreiter in [18]. In Asiacrypt 2001, Courtois, Finiasz and Sendrier [8] proposed the first practical code-based digital signature scheme by applying the Full Domain Hash [2,3] to the Niederreiter cryptosystem. The advantage of using algebraic coding theory to construct cryptographic schemes is that these schemes may remain secure even in the post-quantum age.

SI/IBI Based on Algebraic Coding Theory. In [23], Stern proposed a standard identification scheme based on the syndrome decoding problem from algebraic coding theory. However, the Stern identification scheme is not canonical. It requires $3r$ communications rounds between the prover and the verifier where r is a system parameter. Several variants of the scheme are also introduced in [23], including an identity based one. However, no formal security proof was provided for this IBI scheme. In [7,6], Cayrel et al. proposed a new IBI scheme which can be regarded as the combination of a modified version of the Courtois–Finiasz–Sendrier (CFS) digital signature scheme [9] and the Stern identification scheme [23]. In this paper, we refer to this IBI scheme as mCFS-Stern-IBI. In [6], Cayrel et al. proved that mCFS-Stern-IBI is secure under *passive* attacks.

Our contributions. In this paper, we revisit several existing identification schemes based on algebraic coding theory, including the Stern identification scheme and the mCFS-Stern-IBI scheme. We also provide a new security analysis for the mCFS-Stern-IBI scheme by showing that it can in fact achieve *active* security. However, we show that it is difficult to extend the proof to obtain the *concurrent* security (i.e. the highest level of security) of the scheme.

One widely used approach to transform a passive secure IBI scheme into a concurrent secure one is to use the OR-proof technique. However, due to the special design of the mCFS-Stern-IBI scheme, the conventional OR-proof transformation does not work. We then design a new OR-proof system for this particular IBI and obtain a new scheme which is proven secure under concurrent attacks.

2. Preliminaries

In this section, we review the definition and security model for identity based identification schemes. We follow the IBI definition and security model in [1].

2.1. IBI definition

Definition 1. An identity based identification (IBI) scheme consists of four probabilistic polynomial-time (PPT) algorithms $(\mathsf{MKGen}, \mathsf{UKGen}, \mathsf{P}, \mathsf{V})$.

1. MKGen: On input 1^k, where $k \in \mathbb{N}$ is a security parameter, it generates a master public/secret key pair (mpk, msk).
2. UKGen: On input msk and some identity $I \in \{0,1\}^*$, it outputs a user secret key $usk[I]$.
3. (P, V) – User Identification Protocol: The prover with identity I runs algorithm P with initial state $usk[I]$, and the verifier runs V with initial state (mpk, I). The first and last messages of the protocol belong to the prover. The protocol ends when V outputs either 'accept' or 'reject'.

Completeness: For all $k \in \mathbb{N}$, $I \in \{0,1\}^*$, $(mpk, msk) \leftarrow \mathsf{MKGen}(1^k)$, and $usk[I] \leftarrow \mathsf{UKGen}(msk, I)$, an honest V who is initialized with (mpk, I) always outputs 'accept' at the end of the identification protocol after communicating with P who is honest and initialized with $usk[I]$.

2.2. IBI security model

There are three security notions for IBI schemes: impersonation under passive (id-imp-pa), active (id-imp-aa) and concurrent (id-imp-ca) attacks.

Definition 2 *(id-imp-pa).* For an IBI scheme $(\mathsf{MKGen}, \mathsf{UKGen}, \mathsf{P}, \mathsf{V})$, consider the following game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$.

1. $\mathcal{S}$ generates a master key pair $(mpk, msk) \leftarrow \mathsf{MKGen}(1^k)$ and gives mpk to $\mathcal{A}$. $\mathcal{S}$ also maintains two user sets: HU and CU, which stand for Honest Users and Corrupted Users, respectively. Initially, both HU and CU are empty.

2. $\mathcal{A}$ can make queries to the following oracles:
   (a) INIT(I) – create a user with identity I: If $I \in HU \cup CU$, $\perp$ is returned indicating that I has already been created. Otherwise, $usk[I] \leftarrow \mathsf{UKGen}(msk, I)$ is executed and I is added into HU. A symbol '1' is returned to the adversary indicating that the creation is successful.
   (b) CORR(I) – corrupt a user with identity I: If $I \notin HU$, $\perp$ is returned, otherwise, I is deleted from HU and added into CU, and $usk[I]$ is returned to $\mathcal{A}$.
   (c) CONV(I) – get a conversation between user I (the prover) and a verifier: If $I \notin HU$, $\perp$ is returned, otherwise, a conversation between I (with initial state $usk[I]$) and a verifier (with initial state (mpk, I)) is returned to $\mathcal{A}$.
3. $\mathcal{A}$ can adaptively query INIT, CORR and CONV, and then outputs an identity $\mathrm{I_b} \in HU$, which corresponds to the user that $\mathcal{A}$ wants to impersonate. After receiving $\mathrm{I_b}$, the simulator removes $\mathrm{I_b}$ from HU and adds it into CU.
4. $\mathcal{A}$ runs the user identification protocol with a verifier V (initialized with $(mpk, \mathrm{I_b})$). $\mathcal{A}$ can continue to make INIT, CORR and CONV queries. The simulator halts when V outputs 'accept' or 'reject'.

The advantage of the adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}_{\mathcal{A}}^{\mathsf{id\text{-}imp\text{-}pa}}(k) = \Pr\big[\mathsf{V} \text{ outputs 'accept'}\big].$$

An IBI scheme $(\mathsf{MKGen}, \mathsf{UKGen}, \mathsf{P}, \mathsf{V})$ is said to be id-imp-pa secure if $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{id\text{-}imp\text{-}pa}}(k)$ is negligible[3] for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$.

The id-imp-aa and id-imp-ca security. The id-imp-aa security is defined via a similar game, except that the conversation oracle CONV is replaced by a proving oracle PROV. When querying this oracle, $\mathcal{A}$ provides an identity $I \in \mathrm{HU}$ to the simulator and then acts as a (malicious) verifier to communicate with the prover $\mathsf{P}(usk[I])$ simulated by $\mathcal{S}$. The difference between active (id-imp-aa) and concurrent (id-imp-ca) attack is that in the former adversarial model, $\mathcal{A}$ can only have one ongoing session with one proving oracle at a time, but in the concurrent model, $\mathcal{A}$ can have parallel and concurrent sessions with one proving oracle.

3. Public key cryptosystems based on algebraic coding theory

Let $\mathbb{F}_2$ denote the finite field with two elements $\{0, 1\}$. In this paper, we use C to denote a binary linear code of length n and dimension k, which is a subspace of dimension k of the vector space $\mathbb{F}_2^n$. We call elements of F_2^n words, and elements of C *codewords*. A code is usually given in the form of a generating matrix G, lines of which form a basis of the code. The parity check matrix H is a dual form of this generating matrix: it is the $(n-k) \times n$ matrix which is a generator matrix of the dual code of C. When we multiply a word by the parity check matrix we obtain what is called a *syndrome* which has the length of $n-k$ bits. The security of code-based cryptosystems are based on the intractability of the following problems.

Syndrome decoding problem. Given a random $r \times n$ binary matrix $\bar{H}$, an integer $\omega > 0$, and a binary vector $s \in \mathbb{F}_2^r$, is there a word $x \in \mathbb{F}_2^n$ of weight at most ω such that $\bar{H}x^T = s$?

The variant of the syndrome decoding problem in which we ask for an x with exactly ω 1's is NP-complete [4]. It is conjectured that the syndrome decoding problem is also NP-complete.

Bounded decoding problem. Given an integer d, a random $r \times n$ binary matrix $\bar{H}$ such that every $d-1$ columns of $\bar{H}$ are linearly independent, a binary vector $s \in \mathbb{F}_2^r$, and an integer $\omega \leqslant (d-1)/2$, is there a word $x \in \mathbb{F}_2^n$ of weight at most ω such that $\bar{H}x^T = s$?

Goppa code distinguishing problem. Given an $r \times n$ binary matrix $\bar{H}$, decide whether $\bar{H}$ is a random binary matrix or $\bar{H}$ is a random parity check matrix for a Goppa code.

3.1. The Niederreiter cryptosystem

In [18], Niederreiter proposed a public key cryptosystem which is a variant of the first code-based cryptosystem by McEliece [17]. Let C_0 denote a t-error correcting Goppa code, and H_0 a parity check matrix of C_0. The public key is obtained by $H = VH_0P$ where V is an $(n-k) \times (n-k)$ non-singular matrix and P is an $n \times n$ permutation matrix. The corresponding secret key is (V, H_0, P).

The encryption c of a message $m \in \mathbb{F}_2^n$ (m has Hamming weight at most t) is simply $c = Hm^T \in \mathbb{F}_2^{n-k}$. To decrypt a ciphertext c,

[3] A function $\epsilon : \mathbb{N} \to \mathbb{R}$ is negligible if for every constant $c \geqslant 0$, there exists an integer k_c such that $\epsilon(k) < k^{-c}$ for all $k \geqslant k_c$.

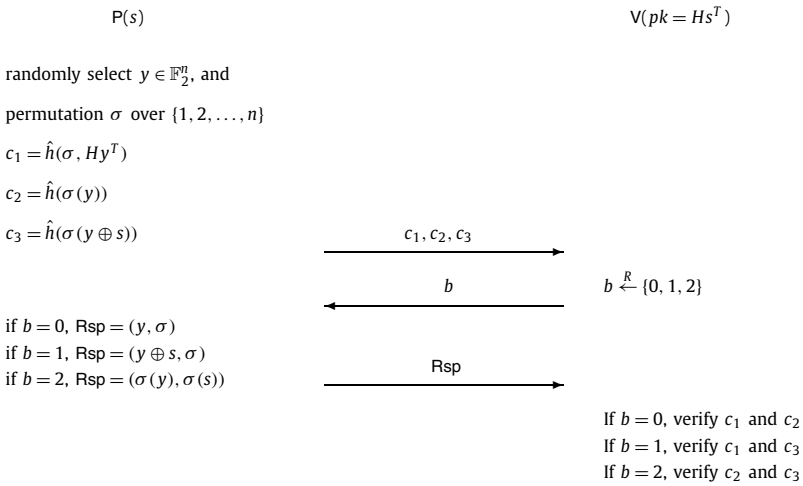

Fig. 1. The Stern identification protocol.

1. compute $\alpha = V^{-1}c = H_0 P m^T$;
2. apply the syndrome decoding procedure for C_0 to α and obtain $\beta = Pm^T$;
3. retrieve the message m via $m^T = P^{-1}\beta$.

3.2. The Courtois–Finiasz–Sendrier signature scheme

In [8], Courtois, Finiasz and Sendrier proposed the first practical code-based signature scheme based on the Niederreiter cryptosystem presented above. The idea is to apply the Full Domain Hash [2,3] to the Niederreiter cryptosystem.

Let H and (V, H_0, P) denote the user public and private key in the Niederreiter cryptosystem. To generate a signature for a message m, the signing algorithm works as follows:

- Set an initial counter $i = 0$;
- Compute $s = h(m, i)$ using a cryptographic hash function $h : \{0,1\}^* \to \mathbb{F}_2^{n-k}$;
- Use the decryption algorithm to find x such that $Hx^T = s$. If no such x is found, set $i = i + 1$ and go back to step 2.
- Output (x, i) as the signature.

To verify a signature (x, i) for a message m:

- Compute $s = Hx^T$ and $s' = h(m, i)$;
- If $s = s'$, return *true*; otherwise, return *false*.

In [8], it has been shown that if we use t-error correcting Goppa code, then the probability that a random syndrome is decodable is about $\frac{1}{t!}$.

A variant. In [9], Dallot proposed a slight modified version of the CFS scheme where the counter i is randomly selected from $\{1, 2, \ldots, 2^{n-k}\}$. It has been proven in [9] that the modified CFS (mCFS) scheme is *strongly* unforgeable under adaptive chosen message attacks.

Theorem 1. *(See [9].) The modified CFS digital signature scheme is strongly unforgeable under adaptive chosen message attacks.*

3.3. The Stern identification scheme

In [22,23], Stern introduced a standard identification scheme based on the syndrome decoding problem.

Let H denote a random $(n-k) \times n$ matrix over $\mathbb{F}_2$. This matrix is public. Each user receives an n-bit secret key s of weight t. The user public key is $pk = Hs^T$. To identify him/herself to a verifier who has pk, the prover runs the identification protocol as shown in Fig. 1.

1. Generating y and σ according to the scheme and replacing the unknown secret key s by some arbitrary vector v of weight t, then computing the various commitments. By doing this, the false prover hopes that b is 0 or 2. In the first case, he can simply disclose y and σ and in the second case, he returns $\sigma(y)$ and $\sigma(v)$. On the other hand, he is unable to answer when $b = 1$.
2. A similar strategy can be defined with $y \oplus s$ in place of y. In this case, the false prover hopes that b is 1 or 2.
3. Having σ and both y and $y \oplus v$ ready where v is some element such that $Hv^T = pk$ (there is no requirement on the weight of v). This strategy expects that b is 0 or 1.

Fig. 2. Impersonation strategies for the Stern identification scheme.

The Stern Identification Protocol

1. The prover chooses randomly a word $y \in \mathbb{F}_2^n$ and a permutation σ of $\{1, 2, \ldots, n\}$, and sends to the verifier c_1, c_2, c_3 such that

$$c_1 = \hat{h}\big(\sigma, Hy^T\big), \qquad c_2 = \hat{h}\big(\sigma(y)\big), \qquad c_3 = \hat{h}\big(\sigma(y \oplus s)\big),$$

   where $\hat{h}$ denotes a cryptographic hash function.
2. Upon receiving (c_1, c_2, c_3), the verifier randomly selects $b \in \{0, 1, 2\}$ and sends b to the prover.
3. Based on the value of b, the prover responds as follows
   - If $b = 0$, the prover reveals y and σ;
   - If $b = 1$, the prover reveals $y \oplus s$ and σ;
   - If $b = 2$, the prover reveals $\sigma(y)$ and $\sigma(s)$.
4. Upon receiving the response,
   - If $b = 0$, the verifier verifies c_1 and c_2;
   - If $b = 1$, the verifier verifies c_1 and c_3;
   - If $b = 2$, the verifier verifies c_2 and c_3.
5. Repeat the above steps for r times.

During the fourth step, when $b = 1$, it can be noticed that Hy^T can be derived from $H(y \oplus s)^T$ by

$$Hy^T = H(y \oplus s)^T \oplus Hs^T = H(y \oplus s)^T \oplus pk.$$

It has been shown in [23] that the above identification scheme is a zero-knowledge Proof-of-Knowledge (PoK) with knowledge error $(2/3)^r$. In particular, three strategies (Fig. 2) have been presented in [23] for an adversary to impersonate a given identity without knowing the secret key. Each impersonation strategy has a success probability of 2/3.

3.4. Identity based variants of the Stern identification scheme

In [23], Stern also introduced an identity based variant of his identification scheme. But no formal security analysis has been provided in [23] for the IBI scheme. In [7,6], Cayrel et al. combined the modified CFS signature scheme [9] and the Stern identification scheme to construct a new IBI scheme (i.e. mCFS-Stern-IBI).

The mCFS-Stern-IBI scheme

- MKGen: run the key generation algorithm of the modified CFS signature scheme to generate a signing and verification key pair (sk, vk). Set $mpk = vk$, and $msk = sk$.
- UKGen: run the signing algorithm of the modified CFS signature scheme to generate a signature (x, i) on a user identity I. Set $usk[I] = (x, i)$.
- P, V: run the Stern identification protocol where P is initialized with x and V is initialized with $h(I, i)$ (i is sent to the verifier in the first message of the protocol).

We can see that $h(I, i)$ in fact serves as the "public key" of the user in the Stern identification scheme. Cayrel et al. [6] proved that the above IBI scheme is id-imp-pa secure.

4. A new security proof for the mCFS-Stern-IBI scheme

We provide a new security proof for the mCFS-Stern-IBI scheme [7,6] reviewed in the previous section. We show that the scheme is in fact id-imp-aa secure.

Theorem 2. *The mCFS-Stern-IBI is secure under active attacks.*

Proof. The proof is by contradiction. Given an adversary $\mathcal{A}$ against the mCFS-Stern-IBI in id-imp-aa game, we construct a forger $\mathcal{F}$ against the mCFS signature scheme. $\mathcal{F}$ is given the public key vk_{mCFS} of the mCFS signature scheme. $\mathcal{F}$ sets $mpk = vk_{\mathsf{mCFS}}$ and passes mpk to $\mathcal{A}$. $\mathcal{F}$ then simulates the id-imp-aa game as follows.

Suppose $\mathcal{A}$ issues at most q_{inti} INTI queries. $\mathcal{F}$ randomly selects an index ℓ in $\{1, 2, \ldots, q_{inti}\}$. For the j-th INTI query made by $\mathcal{A}$ where $j \neq \ell$, $\mathcal{F}$ queries the signing oracle to generate a signature (x_j, i_j) for ID_j. For the ℓ-th INTI query, $\mathcal{F}$ randomly selects $i_\ell \in \{1, 2, \ldots, 2^{n-k}\}$. Notice that according to the analysis given in [8],

$$\Pr\big[h(ID_\ell, i_\ell) \text{ is decodable}\big] = \frac{1}{t!}.$$

When $\mathcal{A}$ issues a corruption query to any ID_j such that $ID_j \neq ID_\ell$, $\mathcal{F}$ returns (x_j, i_j) to $\mathcal{A}$. If $\mathcal{A}$ issues a corruption query to ID_ℓ (denote this event by Abort_1), $\mathcal{F}$ aborts the game without any output. Then we have

$$\Pr[\mathsf{Abort}_1] \leqslant \frac{1}{q_{inti}}.$$

To simulate the PROV oracle for user ID_ℓ, $\mathcal{F}$ works as follows. At the beginning of each round, $\mathcal{F}$ chooses at random one of the three cheating strategies described in Section 3.3 and prepares the initial commitments c_1, c_2, c_3 according to the chosen strategy. Now, each strategy allows to successfully answer two of the three challenges issued by $\mathcal{A}$. In case $\mathcal{A}$ asks a challenge which $\mathcal{F}$ cannot answer, $\mathcal{F}$ resets $\mathcal{A}$ for the current round. The reset will continue until $\mathcal{F}$ can successfully answer $\mathcal{A}$'s challenge, or number of reset reaches a limit λ (to be determined shortly). In the latter case, $\mathcal{F}$ aborts the game without any output. Denote the event that $\mathcal{F}$ aborts the game when simulating a PROV oracle for ID_ℓ by Abort_2. Then by the union bound we have

$$\Pr[\mathsf{Abort}_2] \leqslant q_{prov} r \left(\frac{1}{3}\right)^{\lambda}$$

where q_{prov} denotes the number of PROV queries $\mathcal{A}$ would ask. In order to make $\Pr[\mathsf{Abort}_2] \leqslant 1/3$, we can set $\lambda = \log_3 q_{prov} r + 1$.

Under the condition that $\mathcal{F}$ does not abort the game in the simulation, and $h(ID_\ell, i_\ell)$ is decodable (which happens with probability $\frac{1}{t!}$), then the simulation is perfect.

Now suppose the adversary $\mathcal{A}$ can impersonate the user ID_ℓ with a non-negligible probability, the following lemma by Stern shows that there exists a polynomial-time algorithm which can extract the user secret key x_ℓ of ID_ℓ also with a non-negligible probability.

Lemma 1. *(See Lemma 1 of [23].) Assume there exists a PPT adversary which can impersonate an uncorrupted user with probability $(2/3)^r + \epsilon$, then there exists a polynomial-time algorithm which can extract the user secret key with probability at least $\epsilon^3/10$.*

After obtaining the valid user secret key x_ℓ for ID_ℓ, $\mathcal{F}$ outputs (x_ℓ, i_ℓ) as the forgery for the message ID_ℓ. □

On the concurrent security. The above proof for active security cannot be easily extended to prove the concurrent security of the scheme. When we simulate the PROV oracle in the concurrent security game, *recursive rewinding* may occur. We leave the id-imp-ca security of the mCFS-Stern-IBI scheme as an open problem.

5. A new code-based IBI scheme secure under concurrent attacks

Given that we are unable to directly prove the id-imp-ca security of the mCFS-Stern-IBI scheme, a natural question is if we can construct an id-imp-ca secure variant of it. A popular way to transform an id-imp-pa secure identification scheme into an id-imp-ca secure one is to use the OR-proof technique [16,12]. So we start with an OR-proof variant of the mCFS-Stern-IBI scheme.

5.1. The first OR-proof variant

To apply the OR-proof technique as shown in [16,12], we modify Cayrel et al.'s IBI [7,6] scheme in the following way: we first generate two valid signatures (x_0, i_0) and (x_1, i_1) for a user identity I, and then toss a coin ϖ and set $usk[I] = (\varpi, x_\varpi, i_0, i_1)$. During the identification phase, the user proves that he knows at least one valid secret key with respect to $h(I, i_0)$ or $h(I, i_1)$. The detailed scheme is presented below.

The OR-proof protocol

1. The prover first randomly chooses $b_{1-\varpi} \in \{0,1,2\}$. Based on the values of $b_{1-\varpi}$ and $h(I, i_{1-\varpi})$, randomly select an impersonation strategy given in Fig. 2 (e.g. if $b_{1-\varpi} = 0$, choose either strategy 1 or strategy 3) and prepare the commitment $(c_1^{1-\varpi}, c_2^{1-\varpi}, c_3^{1-\varpi})$. Based on $h(I, i_\varpi)$ and x_ϖ, prepare $(c_1^{\varpi}, c_2^{\varpi}, c_3^{\varpi})$ according to the original Stern identification protocol. Send $(i_0, c_1^0, c_2^0, c_3^0, i_1, c_1^1, c_2^1, c_3^1)$ to the verifier.
2. Upon receiving $(i_0, c_1^0, c_2^0, c_3^0, i_1, c_1^1, c_2^1, c_3^1)$, the verifier randomly selects $b \in \{0,1,2\}$ and sends b to the prover.
3. The prover computes the response for $b_{1-\varpi}$ w.r.t. $h(I, i_{1-\varpi})$ based on the impersonation strategy chosen in the first step. The prover then computes $b_\varpi = b - b_{1-\varpi} \bmod 3$, and computes the response for b_ϖ w.r.t. $h(I, i_\varpi)$ according to the original Stern identification protocol.
4. Upon receiving the response, the verifier checks that $b_0 + b_1 = b \bmod 3$ and the response Rsp_0 for $h(I, i_0)$ w.r.t. b_0, and the response Rsp_1 for $h(I, i_1)$ w.r.t. b_1, are both correct.
5. Repeat the above steps for r times.

Security analysis. The above OR-proof approach has been widely applied in the design of id-imp-ca secure IBI schemes [16,24,12]. However, when we apply this approach to the mCFS-Stern-IBI scheme, we found that the derived IBI scheme is insecure at all. An adversary can impersonate the prover by selecting one impersonation strategy (Fig. 2) for $h(I, i_0)$, and another one for $h(I, i_1)$. Take as an example, the adversary chooses strategy 1 for $h(I, i_0)$, and strategy 2 for $h(I, i_1)$. That means the attacker can pass the verification if $b_0 \in \{0,2\}$ and $b_1 \in \{1,2\}$. The problem is that for any $b \in \{0,1,2\}$, the adversary can simply find a pair of $b_0 \in \{0,2\}$ and $b_1 \in \{1,2\}$ such that $b_0 + b_1 = b \bmod 3$. It is easy to verify that the attack works no matter which two impersonation strategies the adversary chooses.

5.2. A new OR-proof variant

Difficulties in obtaining a three-move protocol. The problem of the first OR-proof scheme is that the adversary has too much "freedom" in answering the challenge sent by the verifier. To solve the problem, we need a way to restrict the adversary's freedom while at the same time preserve completeness (i.e. the real prover can always complete the protocol successfully). If we revisit the attack against the first OR-proof protocol, we observe that when the adversary chooses two impersonation strategies for $h(I, i_0)$ and $h(I, i_1)$, for example $b_0 \in \{0,2\}$ and $b_1 \in \{1,2\}$, then $b_0 + b_1 \in \{1,2,0,1\}$. If we require the adversary to provide responses for two different b_0 and b_0', and also responses for two different b_1 and b_1', such that $b_0 + b_1 = b$ and $b_0' + b_1' = b$, then the probability that the adversary can cheat is $1/3$ (in the example, the adversary can provide the responses only when $b = 1$). On the other hand, a real prover who has one valid secret key can provide responses for any challenge $b \in \{0,1,2\}$.

The above observation shows that by requiring the prover to provide two responses for one challenge, we can enhance the "soundness" of the protocol. However, this approach is insecure either, since it will allow the verifier to easily obtain the user secret key (i.e., the protocol is not "witness hiding"). It is easy to see that if the prover provides the responses for both b_0 and b_0' (or b_1 and b_1'), then the verifier is able to derive the secret key w.r.t. $h(I, i_0)$ (or $h(I, i_1)$). In the Stern identification scheme, the responses have the form (y, σ), $(y \oplus s, \sigma)$, and $(\sigma(y), \sigma(s))$. If the prover sends any two responses to the verifier, then the verifier can derive the value of the secret key s easily.

In summary, to provide the soundness property, we must use two response pairs in the protocol. However, if the prover sends both response pairs in one move, then the protocol will lose the witness hiding property. It seems difficult to reconcile the conflict in a three-move protocol. To resolve the problem, we should let the prover only send either the response pair for b_0 and b_1 or the response pair for b_0' and b_1', but at the same time demonstrate that he/she can produce valid response pairs for both cases. This can be done by applying an additional challenge-response phase, that is, the verifier will select another random challenge bit $\rho \in \{0,1\}$, and based on the value of ρ, the prover reveals one of the two response pairs to the verifier. The details are given below.

Code-IBI: A new variant of mCFS-Stern-IBI

- MKGen: run the key generation algorithm of the modified CFS signature scheme to generate a signing and verification key pair (sk, vk). Set $mpk = vk$, and $msk = sk$.
- UKGen: run the signing algorithm of the modified CFS signature scheme twice to generate two different signatures (x_0, i_0) and (x_1, i_1) $(i_0 \neq i_1)$ for a user identity I, and then toss a coin ϖ and set $usk[I] = (\varpi, x_\varpi, i_0, i_1)$.
- (P, V): initialize P with $usk[I]$, and V with I. Then run the following identification protocol.
  1. The prover first randomly chooses two different $b_{1-\varpi}, b'_{1-\varpi} \in \{0,1,2\}$. Based on the values of $b_{1-\varpi}, b'_{1-\varpi}$, select for $h(I, i_{1-\varpi})$ the impersonation strategy given in Fig. 2 (e.g. if $b_{1-\varpi} = 0$ and $b'_{1-\varpi} = 1$, choose strategy 3) and prepare the commitment $(c_1^{1-\varpi}, c_2^{1-\varpi}, c_3^{1-\varpi})$. Based on $h(I, i_\varpi)$ and x_ϖ, prepare $(c_1^{\varpi}, c_2^{\varpi}, c_3^{\varpi})$ according to the original Stern identification protocol. Send $(i_0, c_1^0, c_2^0, c_3^0, i_1, c_1^1, c_2^1, c_3^1)$ to the verifier.
  2. Upon receiving $(i_0, c_1^0, c_2^0, c_3^0, i_1, c_1^1, c_2^1, c_3^1)$, the verifier randomly selects $b \in \{0,1,2\}$ and sends b to the prover.

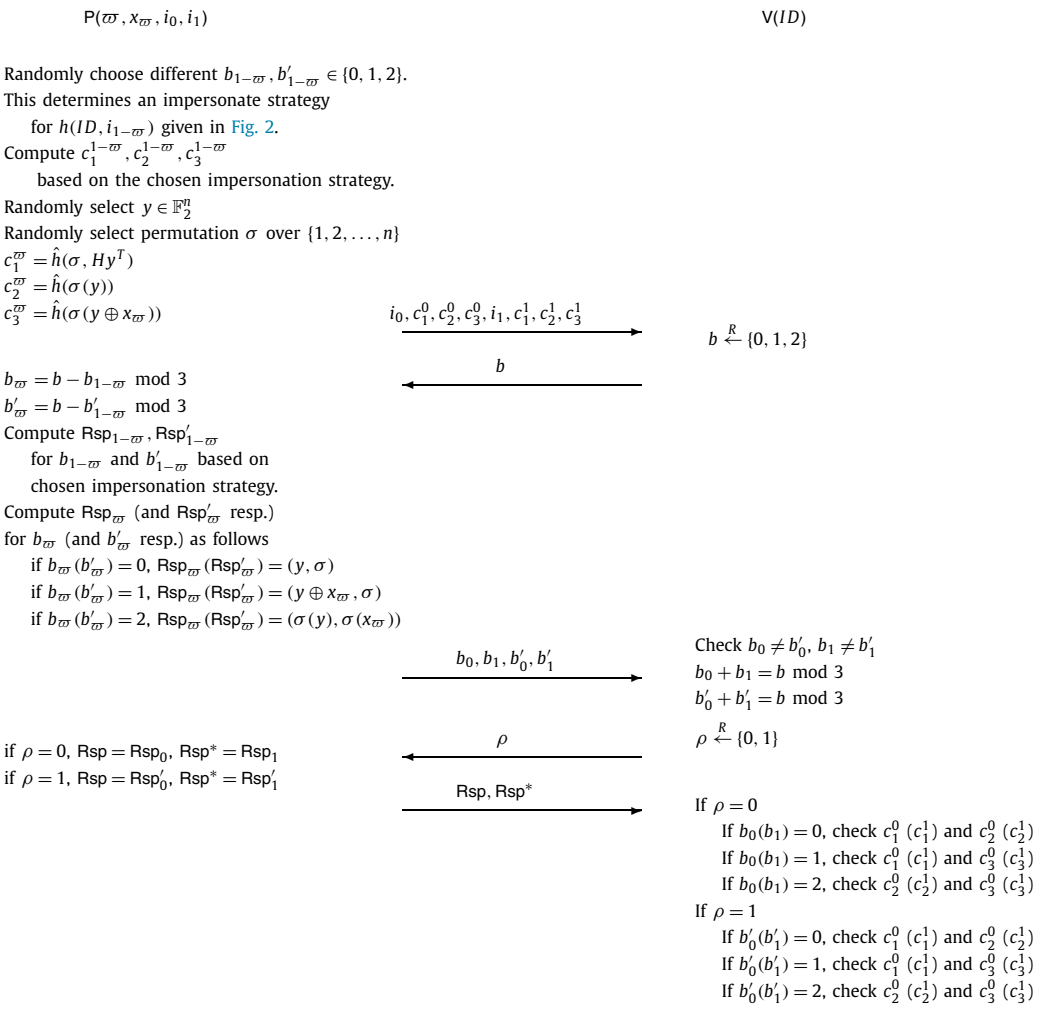

Fig. 3. Code-IBI: A variant of mCFS-Stern-IBI with concurrent security.

3. The prover computes the responses $\mathsf{Rsp}_{1-\varpi}$ and $\mathsf{Rsp}'_{1-\varpi}$ for $h(I, i_{1-\varpi})$ w.r.t. $b_{1-\varpi}$ and $b'_{1-\varpi}$ based on the impersonation strategy chosen in the first step. The prover then computes $b_\varpi = b - b_{1-\varpi} \mod 3$ and $b'_\varpi = b - b'_{1-\varpi} \mod 3$, and computes the responses Rsp_ϖ and Rsp'_ϖ for $h(I, i_\varpi)$ w.r.t. b_ϖ and b'_ϖ according to the original Stern identification protocol.
4. The prover sends (b_0, b_1) and (b'_0, b'_1) to the verifier.
5. The verifier checks that $b_0 \neq b'_0$, $b_1 \neq b'_1$, $b_0 + b_1 = b \mod 3$, $b'_0 + b'_1 = b \mod 3$. Then the verifier randomly chooses a bit $\rho \in \{0, 1\}$ and sends it to the prover.
6. If $\rho = 0$, the prover sends Rsp_0 and Rsp_1 to the verifier; otherwise, if $\rho = 1$, the prover sends Rsp'_0 and Rsp'_1 to the verifier.
7. Upon receiving the response, the verifier checks that Rsp_0 and Rsp_1 w.r.t. b_0 and b_1 (if $\rho = 0$), or Rsp'_0 and Rsp'_1 w.r.t. b'_0 and b'_1 (if $\rho = 1$), are correct.
8. Repeat the above steps for r times.

Now let's revisit the impersonation attack against the first OR-proof given in Section 5.1. Assume the adversary chooses impersonation strategy 1 (Fig. 2) for $h(I, i_0)$, and strategy 2 for $h(I, i_1)$. That means the attacker can pass the verification if $b_0, b'_0 \in \{0, 2\}$ and $b_1, b'_1 \in \{1, 2\}$. So if the challenge $b = 1$, then the adversary can successfully pass the verification by setting $b_0 = 0, b'_0 = 2$ and $b_1 = 1, b'_1 = 2$. However, if $b = 0$ or $b = 2$, then the adversary only has probability $1/2$ to pass the verification. That means the probability that the adversary can impersonate an uncorrupted user in one round is bounded by $\frac{1}{3} \cdot 1 + \frac{2}{3} \cdot \frac{1}{2} = \frac{2}{3}$. The result shown in the following theorem coincides with this informal analysis.

Theorem 3. *The new identification protocol* (P, V) *is a proof of knowledge system with knowledge error* $(2/3)^r$.

Proof. In the proposed Code-IBI scheme (Fig. 3), the challenge space becomes $(b, \rho) \in \{(0,0), (0,1), (1,0), (1,1), (2,0), (2,1)\}$. Consider the tree $T(\omega)$ of all 6^r executions corresponding to all possible questions of the verifier when the adversary has a fixed random tape ω. Now a vertex with at least 5 children corresponds to a situation where a commitment $(i_0, c_1^0, c_2^0, c_3^0, i_1, c_1^1, c_2^1, c_3^1)$ has been made and the adversary can provide answers to at least 5 possible challenges of the verifier.

Lemma 2. *If there exists an adversary $\mathcal{A}$ which can impersonate an uncorrupted user with probability $(2/3)^r + \epsilon$, then there exists a polynomial-time algorithm which can find a vertex with five children with probability at least $\epsilon^3/10$.*

Proof. The proof follows the same technique as used in [23]. The difference is that now we are dealing with a 6-ary tree instead of a 3-ary tree. Suppose there exists an adversary $\mathcal{A}$ that can impersonate an uncorrupted user with probability $(2/3)^r + \epsilon$, we construct a polynomial-time algorithm to find a vertex with at least five children as follows:

1. Randomly select a random tape ω for the adversary $\mathcal{A}$. This defines an execution tree $T(\omega)$.
2. Randomly select a sequence for the verifier's queries. This defines a branch B of $T(\omega)$.
3. Visit all the vertices along the selected branch B. If a vertex with at least five children is found at level i, return (ω, B, i); else, return $\perp$.

Analysis. Consider the set X defined by

$$X = \left\{ \omega \;\middle|\; T(\omega) \text{ has at least } 4^r + \frac{\epsilon}{2} 6^r \text{ branches} \right\}.$$

Then the probability that a random ω falls in X is at least $\epsilon/2$. Otherwise, we can bound the overall successful probability of $\mathcal{A}$ by

$$\begin{aligned}
\Pr[\mathcal{A} \text{ is successful}] &= \Pr[\mathcal{A} \text{ is successful} \wedge \omega \in X] + \Pr[\mathcal{A} \text{ is successful} \wedge \omega \notin X] \\
&\leqslant \Pr[\omega \in X] + \Pr[\mathcal{A} \text{ is successful} \mid \omega \notin X] \\
&< \frac{\epsilon}{2} + \left(\frac{4}{6}\right)^r + \frac{\epsilon}{2} \\
&= \left(\frac{2}{3}\right)^r + \epsilon.
\end{aligned}$$

So by contradiction, the probability that a random ω falls in X is at least $\epsilon/2$. Now, since $T(\omega)$ has at least $4^r + \epsilon/2 \cdot 6^r$ branches, the probability that the branch B we selected in step 2 of our algorithm corresponds to a successful execution is at least $(4/6)^r + \epsilon/2$.

For any level i, $0 \leqslant i \leqslant r$, we let n_i denote the number of vertices at level i, and for $0 \leqslant i < r$, we define $\alpha_i = n_{i+1}/n_i$. Then we have

$$\prod_{i=0}^{r-1} \alpha_i \geqslant 4^r + \frac{\epsilon}{2} 6^r.$$

Taking logarithms, this yields

$$\sum_{i=0}^{r-1} \log_4(\alpha_i) \geqslant \log_4\left(4^r + \frac{\epsilon}{2} 6^r\right) \geqslant \log_4\left(\left(1 - \frac{\epsilon}{2}\right) 4^r + \frac{\epsilon}{2} 6^r\right) \geqslant \left(1 - \frac{\epsilon}{2}\right) r + \frac{\epsilon}{2} r \log_4 6$$

where the last inequality is based on the convexity inequality. Hence, one of the $\log_4(\alpha_i)$'s must exceed

$$1 - \frac{\epsilon}{2} + \frac{\epsilon}{2} \log_4 6 = 1 + \frac{\log 3 - 1}{4} \epsilon$$

which implies

$$\alpha_i \geqslant 4^{1 + \frac{\log 3 - 1}{4} \epsilon} = 2^{2 + \frac{\log 3 - 1}{2} \epsilon} = 4 \cdot 2^{\frac{\log 3 - 1}{2} \epsilon} = 4 \cdot e^{\ln 2 \frac{\log 3 - 1}{2} \epsilon}.$$

From the inequality $e^x \geqslant 1 + x$ for all $x \geqslant 0$ we have

$$\alpha_i \geqslant 4 \cdot \left(1 + \ln 2 \frac{\log 3 - 1}{2} \epsilon\right).$$

This indicates

$$n_{i+1} - 4n_i \geqslant \big(2\ln 2(\log 3 - 1)\epsilon\big)n_i.$$

Since each vertex at level i has at most 6 children, we have

$$\Pr[\text{a randomly selected vertex at level } i \text{ has more than 4 child}] \geqslant \ln 2(\log 3 - 1)\epsilon.$$

Hence, the probability that we find a vertex with at least 5 children in step 3 of our algorithm is at least $\ln 2(\log 3 - 1)\epsilon$.

Combining all together, the overall probability of our algorithm to find a vertex with at least 5 children is at least

$$\epsilon/2 \cdot \big((4/6)^r + \epsilon/2\big) \cdot \big(\ln 2(\log 3 - 1)\epsilon\big) \geqslant \epsilon^3/10. \qquad \square$$

Lemma 3. *For any vertex in the tree $T(\omega)$, if the adversary can provide answers for at least 5 possible challenges of the verifier, then the adversary can provide answers for all possible challenges w.r.t. either $h(I, i_0)$ and commitment c_1^0, c_2^0, c_3^0 or $h(I, i_1)$ and commitment c_1^1, c_2^1, c_3^1 in the Stern identification scheme.*

Proof. The proof is by contradiction. Suppose the adversary can only provide answers for two different challenges $x, x' \in \{0, 1, 2\}$ w.r.t. $h(I, i_0)$ and c_1^0, c_2^0, c_3^0 and answers for two different challenges $y, y' \in \{0, 1, 2\}$ w.r.t. $h(I, i_1)$ and c_1^1, c_2^1, c_3^1. Then $\{x+y, x+y', x'+y, x'+y'\}$ (mod 3) will cover the set $\{0, 1, 2\}$ and have exactly one number repeated once. Wlog, suppose $(x, x') = (0, 1)$ and $(y, y') = (1, 2)$, then $(x + y, x + y', x' + y, x' + y') = (1, 2, 2, 0)$ (mod 3), which means the adversary is able to answer at most 4 different challenges among the challenge set $\{(0, 0), (0, 1), (1, 0), (1, 1), (2, 0), (2, 1)\}$ defined by the challenge (b, ρ) sent by the verifier. More specifically, in this case the adversary is able to produces responses for one of $\{(0, 0), (0, 1)\}$, one of $\{(1, 0), (1, 1)\}$, and both of $\{(2, 0), (2, 1)\}$. A similar analysis can be done for other possible combinations of (x, x') and (y, y'). $\square$

Lemma 4. *If there exists an adversary $\mathcal{A}$ which can provide answers for all possible challenges w.r.t. $h(I, i)$ and commitment c_1, c_2, c_3 in the Stern identification protocol, then we can extract the secret key x for $pk = h(I, i)$.*

The proof for Lemma 4 can be found in [23] (Theorem 1). Theorem 3 can be obtained immediately from Lemmas 2–4. $\square$

Now, by combining Theorem 1 with Theorem 3, we can show that the Code-IBI scheme (Fig. 3) is secure under concurrent attacks.

Theorem 4. *The Code-IBI scheme is secure under concurrent attacks in the random oracle model.*

Proof. The proof is very similar to the proof of Theorem 2. Given an adversary $\mathcal{A}$ against the Code-IBI scheme in the id-imp-ca security model, we construction a new adversary $\mathcal{F}$ against the mCFS digital signature scheme in the strong unforgeability model.

Suppose $\mathcal{A}$ issues at most q_{inti} INTI queries. $\mathcal{F}$ randomly selects an index ℓ in $\{1, 2, \ldots, q_{inti}\}$. For the j-th INTI query made by $\mathcal{A}$ where $j \neq \ell$, $\mathcal{F}$ queries the signing oracle to generate two signatures (x_0^j, i_0^j) and (x_1^j, i_1^j) where $(i_0^j \neq i_1^j)$ for ID_j, and then set the secret key of ID_j by following the normal procedures described in the UKGen algorithm. For the ℓ-th INTI query, $\mathcal{F}$ first queries the signing oracle to obtain a valid signature (x, i) for ID_ℓ. Then $\mathcal{F}$ randomly selects $i' \in \{1, 2, \ldots, 2^{n-k}\}$. Notice that according to the analysis given in [8],

$$\Pr\big[h\big(ID_\ell, i'\big) \text{ is decodable}\big] = \frac{1}{t!}.$$

$\mathcal{F}$ then tosses a random coin ϖ and sets $x_\varpi^\ell = x, i_\varpi^\ell = i$ and $i_{1-\varpi}^\ell = i'$.

When $\mathcal{A}$ issues a corruption query to any ID_j such that $ID_j \neq ID_\ell$, $\mathcal{A}$ returns the secret key of ID_j to $\mathcal{A}$. If $\mathcal{A}$ issues a corruption query to ID_ℓ (denote this event by Abort_1), $\mathcal{F}$ aborts the game without any output. Then we have

$$\Pr[\mathsf{Abort}_1] \leqslant \frac{1}{q_{inti}}.$$

To simulate the PROV oracle for user ID_ℓ, $\mathcal{F}$ follows the protocol honestly since $\mathcal{F}$ has a valid secret key of ID_ℓ. Under the condition that $h(ID_\ell, i_{1-\varpi}^\ell)$ is decodable, the simulation is perfect.

Now suppose the adversary $\mathcal{A}$ can impersonate the user ID_ℓ with a non-negligible probability, then according to Theorem 3, $\mathcal{F}$ can extract a valid secret key of ID_ℓ also with a non-negligible probability. Since it has been shown in [23] that if we assume the hash function $\hat{h}(\cdot)$ is a random oracle, then the simulated transcript w.r.t. $h(ID_\ell, i_{1-\varpi}^\ell)$ is indistinguishable from the transcript generated by using a secret key corresponding to $h(ID_\ell, i_{1-\varpi}^\ell)$. Therefore, the Code-IBI scheme is witness indistinguishable, and with probability 1/2, the user secret key extracted by $\mathcal{F}$ from $\mathcal{A}$ constitutes a valid signature $(x_{1-\varpi}^\ell, i_{1-\varpi}^\ell)$ for ID_ℓ. Then $\mathcal{F}$ outputs $(x_{1-\varpi}^\ell, i_{1-\varpi}^\ell)$ as the forgery for the message ID_ℓ and wins the strong unforgeability game. $\square$

6. Conclusion

In this paper, we revisited the Stern identification scheme and the mCFS-Stern-IBI scheme based on algebraic coding theory. We provide a new security analysis for the mCFS-Stern-IBI scheme by showing that the scheme can be proven secure against active adversaries whereas the previous result only proves its passive security. We then further extend this IBI scheme to obtain concurrent security by using a special OR-proof system. One interesting open problem is: Can we directly prove the concurrent security of the mCFS-Stern-IBI scheme?

Acknowledgements

We thank the anonymous reviewers for their invaluable comments and suggestions.

References

[1] Mihir Bellare, Chanathip Namprempre, Gregory Neven, Security proofs for identity-based identification and signature schemes, in: Proc. EUROCRYPT 2004, Springer, 2004, pp. 268–286, full paper available at http://eprint.iacr.org/2004/252.
[2] Mihir Bellare, Phillip Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, in: Proc. ACM CCS, 1993, pp. 62–73.
[3] Mihir Bellare, Phillip Rogaway, The exact security of digital signatures – how to sign with RSA and Rabin, in: Proc. EUROCRYPT 96, Springer, 1996, pp. 399–416.
[4] E.R. Berlekamp, R.J. McEliece, H.C. van Tilborg, On the inherent intractability of certain coding problems, IEEE Trans. Inf. Theory 24 (1978).
[5] Thomas Beth, Efficient zero-knowledge identification scheme for smart cards, in: Proc. EUROCRYPT 88, Springer, 1988, pp. 77–84.
[6] Pierre-Louis Cayrel, Philippe Gaborit, David Galindo, Marc Girault, Improved identity-based identification using correcting codes, arXiv:0903.0069, 2009.
[7] Pierre-Louis Cayrel, Philippe Gaborit, Marc Girault, Identity-based identification and signature schemes using correcting codes, in: International Workshop on Coding and Cryptography, 2007.
[8] Nicolas Courtois, Matthieu Finiasz, Nicolas Sendrier, How to achieve a McEliece-based digital signature scheme, in: Proc. ASIACRYPT 2001, Springer, 2001, pp. 157–174.
[9] Léonard Dallot, Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme, in: Proc. WEWoRC, 2007, pp. 65–77.
[10] Uriel Feige, Amos Fiat, Adi Shamir, Zero-knowledge proofs of identity, J. Cryptol. 1 (2) (1988) 77–94.
[11] Amos Fiat, Adi Shamir, How to prove yourself: Practical solutions to identification and signature problems, in: Proc. CRYPTO 86, Springer, 1987, pp. 186–194.
[12] Atsushi Fujioka, Taiichi Saito, Keita Xagawa, Security enhancements by OR-proof in identity-based identification, in: Proc. ACNS, 2012, pp. 135–152.
[13] S. Goldwasser, S. Micali, R. Rivest, A digital signature scheme secure against adaptive chosen-message attack, SIAM J. Comput. 17 (2) (April 1988) 281–308.
[14] Louis C. Guillou, Jean-Jacques Quisquater, A "paradoxical" indentity-based signature scheme resulting from zero-knowledge, in: Proc. CRYPTO 88, Springer, 1990, pp. 216–231.
[15] Kaoru Kurosawa, Swee-Huay Heng, From digital signature to ID-based identification/signature, in: Public Key Cryptography 2004, Springer, 2004, pp. 248–261.
[16] Kaoru Kurosawa, Swee-Huay Heng, Identity-based identification without random oracles, in: Proc. ICCSA (2), 2005, pp. 603–613.
[17] Robert J. McEliece, A public-key cryptosystem based on algebraic coding theory, Technical report, DSN Progress report #42-44, Jet Propulsion Laboratory, Pasadena, California, 1978.
[18] H. Niederreiter, Knapsack-type cryptosystems and algebraic coding theory, Probl. Control Inf. Theory 15 (2) (1986) 159–166.
[19] Kazuo Ohta, Tatsuaki Okamoto, A modification of the Fiat–Shamir scheme, in: Proc. CRYPTO 88, Springer, 1990, pp. 232–243.
[20] Tatsuaki Okamoto, Provably secure and practical identification schemes and corresponding signature schemes, in: Proc. CRYPTO 92, Springer, 1993, pp. 31–53.
[21] H. Ong, Claus-Peter Schnorr, Fast signature generation with a Fiat–Shamir-like scheme, in: Proc. EUROCRYPT 90, Springer, 1990, pp. 432–440.
[22] Jacques Stern, A new identification scheme based on syndrome decoding, in: Proc. CRYPTO 93, Springer, 1993, pp. 13–21.
[23] Jacques Stern, A new paradigm for public key identification, IEEE Trans. Inf. Theory 42 (6) (1996) 1757–1768.
[24] Guomin Yang, Jing Chen, Duncan S. Wong, Xiaotie Deng, Dongsheng Wang, A new framework for the design and analysis of identity-based identification schemes, Theor. Comput. Sci. 407 (1–3) (2008) 370–388.