Security Notions for Unconditionally Secure Signature Schemes

Junji Shikata[1], Goichiro Hanaoka[1], Yuliang Zheng[2] and Hideki Imai[1]

[1] Institute of Industrial Science, University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
{shikata,hanaoka}@imailab.iis.u-tokyo.ac.jp, imai@iis.u-tokyo.ac.jp

[2] Department of Software and Information Systems, UNC Charlotte
9201 University City Blvd. Charlotte, NC 28223, USA
yzheng@uncc.edu

Abstract. This paper focuses on notions for the security of digital signature schemes whose resistance against forgery is not dependent on unproven computational assumptions. We establish successfully a sound and strong notion for such signature schemes. We arrive at the sound notion by examining carefully the more established security notions for digital signatures based on public-key cryptography, and taking into account desirable requirements of signature schemes in the unconditional security setting. We also reveal an interesting relation among relevant security notions which have appeared in the unconditionally setting, and significantly, prove that our new security notion is the strongest among all those for unconditionally secure authentication and signature schemes known to date. Furthermore, we show that our security notion encompasses that for public-key signature schemes, namely, existential unforgeability under adaptive chosen-message attack. Finally we propose a construction method for signature schemes that are provably secure in our strong security notion.

1 Introduction

In this paper, we address security notions for signature schemes that do not depend on any computational assumption.

Since the discovery of public-key cryptography [10], significant advances have been reported on digital signature schemes [21][11]. Although it is shown in [10] that a trapdoor function allows to create digital signature schemes in the public-key setting, a number of technical problems arise if digital signatures are implemented using a general trapdoor function as suggested in [10]. Thus it is important to have a formal notion of what a secure digital signature scheme is, and to construct a digital signature scheme which can be proven to be secure in the formal notion. The current standard security notion was established by Goldwasser, Micali and Rivest [14]. In the same paper the authors also demonstrated the first digital signature scheme that was proven to be secure against a very general attack, called adaptive chosen message attack. Since then, many provable secure digital signature schemes have been proposed by researchers [2][23][7][12][1].

These schemes and the infrastructure within which they operate have a limitation in that their underlying security relies on the presumed computational difficulty of certain number-theoretic problems such as the integer factoring problem and the (elliptic curve) discrete logarithm problem. Thus should future progress in computers as well as discoveries of revolutionary algorithms make it computationally feasible to solve larger size number-theoretic problems, such a presumption would not be able to assure the security of current digital signatures. This situation is disturbing considering that there are many cases where documents, such as court and government records, long-term leases and contracts, are required by law to be kept intact for a long period of time, say over 50 years.

In attempting to solve this problem, researchers have introduced unconditionally secure digital signature schemes and authentication codes which do not rely on any unproven assumption such as the discrete logarithm problem. Like many other areas in security, there is clearly a need to identify a kind of benchmarks that one can employ to analyze and compare various signature schemes in the unconditional security setting. A major contribution of this research is to establish a strong security notion for all digital signature schemes including unconditionally secure ones. Additionally, we will show a concrete construction of unconditionally secure digital signature schemes which satisfies the requirements of the strong security notion.

Let us briefly survey existing unconditionally secure schemes. The first unconditionally secure signature was proposed by Chaum and Roijakkers [5]. There have been many attempts to enhance conventional unconditionally secure authentication codes [13] [27] with extra security-properties that are required by signature schemes. Major extensions of conventional authentication codes include the so-called A^2-codes [28] [29] [19] [20] [18], A^3-codes [3] [8] [30] [17] [18] [31] and multi-receiver authentication codes (with dynamic senders) [9] [24] [25] [26] [18]. Recently, the first unconditionally secure signature scheme that admits provably secure transfer of signatures has been proposed in [15]. These schemes, however, have all been proven to be secure against some specific attacks. This raises a number of interesting questions: what are other possible attacks? More importantly, are these signature schemes secure against other yet to be identified attacks?

As mentioned earlier, the focus of this research is to establish a strong security notion for signature schemes whose security does not depend on any computational assumption. It is discussed by taking into account the security notions for public-key signature schemes and additional requirements for signature schemes in the unconditional security setting. Furthermore we examine relations among all the security notions which have been proposed in the context of unconditionally secure signature schemes. It turns out that our security notion is the strongest among all the security notions for unconditionally secure authentication and signature schemes known so far, and it encompasses the security notion for public-key signature schemes, namely existential unforgeability under adaptive chosen-message attack. Finally we propose a construction method for signature schemes that are secure in our strong security notion.

2 Approaches to the notion of unconditional security

2.1 Discussion

In this section, we consider how *unconditionally secure signature schemes* should be defined. By *unconditionally secure* one generally means that security must not depend on any computational assumption. To address the question, there are two issues to be discussed. The first is how to establish a proper model for signature schemes, and the second is to define, in a formal way, unconditional security notion in that model.

When introducing a model for unconditionally secure signature schemes, care should be taken so that properties of public-key signature schemes are captured. In addition, the model should be as simple as possible.

We start with the following typical model for signature schemes.

Definition 1 A *signature scheme* $\Pi = (Gen, Sig, Ver)$ consists of a key generation algorithm, Gen, a signing algorithm, Sig, and a verification algorithm, Ver.

1. **Key Generation:** The key generation algorithm outputs a signing-key x for a signer and a verification-key y for a verifier, respectively.
2. **Signature Generation:** For a message m, the signer creates a signature $a := Sig(x, m)$ using his signing key x. The pair (m, a) is a resultant signed message.
3. **Verification:** The verifier checks whether (m, a) is created by the signer using his verification key. More precisely, the verifier accepts it as having originated from the signer if $Ver(y, m, a) = true$, and rejects it if $Ver(y, m, a) = false$.

Definition 2 Let x be a signing-key of a signer. A signed message (m, a) is said to be *valid* if $a = Sig(x, m)$. Likewise, a signature a of a message m is said to be *valid* if $a = Sig(x, m)$. Otherwise, (m, a) is said to be *invalid*.

To simplify our discussions, we consider a model of signature schemes in which there are a single signer S and multiple verifiers $V_1, V_2, \ldots$. We wish a signature scheme to fulfill the following requirement.

Requirement 1

1. *Verifiability*: Any verifier can non-interactively check whether a signed message received from a signer is valid with his own verification-algorithm. In other words, he can check the validity of a received signed message without

communicating with others after receiving the signed message. More precisely, for any verifier V with his verification-key y, (m, a) is regarded as a valid signed message if and only if $Ver(y, m, a) = true$. In other words, if (m, a) is valid, $Ver(y, m, a) = true$; and if (m, a) is invalid, $Ver(y, m, a) = false$.

2. *Resolution for Dispute by a Third Party*: If a dispute occurs among users, a third party (called an arbiter) can resolve the dispute in a reasonable way: The third party has his own verification-key, and he resolves a dispute among users following the resolution-rule below.
   - *Resolution-Rule*: Let T be the third party and y_T be his verification-key. If a signer S denies the fact that he has created a signed message (m, a) held by a verifier V, then V should be able to present (m, a) to T. T rules in favor of V if $Ver(y_T, m, a,) = true$ and in favor of S otherwise.

   Here, we assume that the third party honestly follows the resolution-rule and honestly outputs its result when a dispute occurs. However, we assume that the third party is not always fully trusted. Namely, we assume that the third party might forge a signature.
3. *Security (unforgeability)*: It is infeasible for any adversary to forge a signature. Here, we assume that not only a verifier may be dishonest but also the signer and a third party may be dishonest. Each of them may become an adversary who may wish to forge a signature.

The level of security we require will be discussed in greater details in Section 2.2.

Requirement 1 can be relaxed in such a way that a *small error probability* is allowed.

Requirement 2 *Verifiability* and *Resolution for Disputes by a Third Party* in Requirement 1 can be relaxed as follows:

1. *Verifiability*: For any verifier V with his verification-key y, if (m, a) is valid, the verifier always accepts it (i.e. $Ver(y, m, a) = true$); and if (m, a) is invalid, the probability that the verifier erroneously accepts it is at most ϵ_1, where ϵ_1 is a very small quantity.
2. *Resolution for Disputes by a Third Party*: If a dispute between a signer and a verifier occurs, the resolution-rule in Requirement 1 is applied. However, we admit the following: If (m, a) is valid, T always accepts it (i.e. $Ver(y_T, m, a,) = true$); and if (m, a) is invalid, the probability that T erroneously accepts it is at most ϵ_2, where ϵ_2 is a very small quantity.

In a digital signature scheme based on public-key cryptography, a verification-key for a verifier can be public and shared among all verifiers. The following theorem indicates that such a signature scheme cannot be secure against an adversary with unlimited computing power.

Theorem 1 *Consider a signature scheme which satisfies Requirement 1. If it is infeasible for an adversary with unlimited computing power to succeed in forging* *a signature, then the verification-key for each verifier must be kept secret from all other verifiers. Similarly, consider a signature scheme which satisfies Requirement 2 with* $\epsilon_i \neq 0$ $(i = 1, 2)$. *If it is infeasible for an adversary with unlimited computing power to succeed in forging a signature, then the verification-key for each verifier must be kept secret not only from all other verifiers but also from a signer.*

A proof for the above theorem will be provided in the full version of this paper.

A consequence of Theorem 1 is that with a signature scheme that allows an adversary to have unlimited computing power, its key generation algorithm must generate verification-keys for all verifiers, and more importantly, distribute the verification-keys to verifiers separately in a secure way. For this reason we have to assume that the number of verifiers is limited. This is in contrast with a public-key signature scheme in which a single public verification-key is adequate and there is no limit placed on the number of verifiers.

To further simplify our discussions, we introduce into our model a trusted authority, denoted by TA. The roles of TA are to generate a signing-key and verification-keys by using a key generation algorithm, and to distribute the signing-key to the signer and verification-keys to each verifier, in a secure way.

2.2 Unforgeability

We now discuss security notions in our signature model. Let $\mathcal{U} := \{S, V_1, V_2, \ldots, V_n\}$ be a set of users, where S is a signer and V_i $(1 \leq i \leq n)$ are verifiers.

We note that the signer has information-theoretic advantage over other verifiers since the signing-key is secret information known only to the signer. We also note that each verifier has information-theoretic advantage over other users, since his verification-key is secret information known only to the verifier. From these facts it follows that we should take into account not only the secrecy of the signer's signing-key but also the secrecy of each verifier's verification-key. This is different from public-key signature schemes in which we need not to consider information-theoretic advantages of a verifier.

On the secrecy of the signer's signing-key, the following security notion can be considered, in conjunction with security notions for public-key signature schemes [14]:

Definition 3 (Forgery and Attacks against a Signer)[14]: Consider an *adversary* who can be either a dishonest verifier or an outsider in our model.

- Types of Forgery:
  1. *Total Break*: An adversary is able either to extract the signing key, or to find an efficient signing algorithm that is functionally equivalent to the signing algorithm equipped with the genuine signing key.
  2. *Selective Forgery*: An adversary is able to create a valid signature for a particular message or a class of messages chosen a priori.

3. *Existential Forgery*: An adversary is able to forge a valid signed message that signer has not created, but the adversary has little or no control over which message will be the target.

– Types of Attacks:
  1. *Key-Only Attack*: If a dishonest receiver is an adversary, the only key information he knows is the information on his verification-key. If an outsider is an adversary, he knows no secret key information, other than publicly available information on the scheme.
  2. *Message Attacks*: An adversary is able to examine signatures corresponding either to known or chosen messages. Message attacks can be further subdivided into three classes:
     (a) *Known-Message Attack*: An adversary has valid signatures for a set of messages which are known to the adversary but not chosen by him.
     (b) *Chosen-Message Attack*: An adversary obtains valid signatures from a chosen list of messages before attempting to forge another signed message.
     (c) *Adaptive Chosen-Message Attack*: An adversary is allowed to use the signer as an oracle; the adversary may request signatures of messages which may depend on the signer's signing key and previously obtained signed messages. That is, at any time the adversary can query the signer with messages chosen at his will, except for the target message.

The strongest signature scheme is one that is secure against existential forgery under adaptive chosen message attack.

Next we consider the secrecy of a verifier's verification-key.

Definition 4 (Forgery and Attacks against a Verifier): Let V be a verifier. In the following, an *adversary* means a dishonest signer, a dishonest verifier, or an outsider in our model.

– Types of Forgery:
  1. *Total Acceptance Forgery for V*: An adversary is able either to compute the verification-key information of the verifier V, or find an efficient verification algorithm that is functionally equivalent to the verification algorithm equipped with the genuine verification-key.
  2. *Selective Acceptance Forgery for V*: An adversary is able to make a signature, which will be accepted by V, for a particular message or a class of messages chosen a priori.
  3. *Existential Acceptance Forgery for V*: An adversary is able to make a signed message that has not been created by the signer but will be accepted by V. The adversary has little or no control over which signed message will be targeted.

– Types of Attacks:

1. *Key-Only Attack*: The only key information which an adversary knows is the adversary's secret key. In a case that the adversary is a signer in our model, the only key information available to him is that of his signing key. Otherwise if the adversary is a verifier, the only key information known to him is that of his verification-key.
2. *Signature Attacks for V*: An adversary is able to examine verification results of V corresponding either to known or chosen signatures. Signature attacks can be further subdivided into three classes:
   (a) *Known-Signature Attack for V*: An adversary has some signed messages and he knows whether these will be accepted by the verifier V or not. However, these are not chosen by him.
   (b) *Chosen-Signature Attack for V*: An adversary obtains some signed messages whose verification results (i.e. the results whether these are accepted or not by V) are known to him. These are chosen before attempting to forge a signed message.
   (c) *Adaptive Chosen-Signature Attack for V*: An adversary is allowed to use the verifier V as an oracle; the adversary may request for an answer as to whether a signed message will be accepted by V. The signed message may be dependent on V's verification-key and verification-results obtained previously from V. That is, at any time the adversary can query the verifier with any signed messages, except for the target.

Finally, some clarifications on the types of forgery and attacks on verifiers follow.

Definition 5 (Forgery Range among Verifiers)

1. *Forgery for All Verifiers*: An adversary can forge a signature for all verifiers.
2. *Forgery for Selective Verifiers*: An adversary can forge a signature for a particular verifier selected by the adversary.
3. *Forgery for Existential Verifiers*: An adversary can forge a signature for a verifier, but the adversary has little or no control over which verifier will be the victim.

The above discussions suggest that a strong security notion be considered along the following line: Under adaptive chosen-message and adaptive chosen-signature attacks, it is infeasible for an adversary to succeed in not only existential forgery but also existential acceptance forgery against any verifier. The following theorem whose proof is straightforward is helpful, as it shows that it will be sufficient to consider only existential acceptance forgery, rather than both existential forgery and existential acceptance forgery.

Theorem 2 *Let Π be a signature scheme. If Π is existentially acceptance unforgeable for any verifier under adaptive chosen-message and adaptive chosen-signature attacks, then it is also existentially unforgeable under adaptive chosen-message and adaptive chosen-signature attacks.*

Based on Theorem 2, we can define a strong security notion as follows:

Definition 6 (Strong Security) Let Π be a signature scheme. Then Π is called *secure* if it is existential acceptance unforgeable for any verifier under adaptive chosen-message and adaptive chosen-signature attacks.

2.3 Some remarks on security notions

In this subsection we consider some conditions that should be met when discussing security notions for signature schemes in unconditional security setting.

- **The security parameter:** In signature schemes with computational security in public-key cryptography, the security parameter is introduced to govern the overall security of a scheme, the length and number of messages, and the running time of algorithms. Similarly, a security parameter k for unconditional secure signature schemes can be defined. This parameter determines the overall security, the key-length of signing-keys and that of verification-keys, the length of messages and that of signatures, and the running time of algorithms.
- **The number of colluders:** There may exist dishonest users, and some dishonest users might collude in order to succeed in forgery. In this paper we adopt the idea of threshold schemes. Namely, we assume that there exists at most ω colluders among the users $\mathcal{U} = \{S, V_1, V_2, \ldots, V_n\}$. In discussing signature schemes with unconditional security, at least from a theoretical viewpoint, introducing the pre-defined number of colluders does not pose a problem in practice when compared with digital signature schemes with computational security, because even in the latter case at most polynomially many colluders are implicitly assumed when discussing security.
- **The numbers of signing and verifying operations:** In order to describe security notions in a more formal way, we should introduce a number up to which an adversary can have access to the signing oracle, and a number up to which the adversary can have access to the verification oracle. We introduce a number up to which a signer is allowed to generate signatures, denoted by ψ, and a number up to which each verifier is allowed to check received signatures, denoted by ψ'. This implies that an adversary can obtain at most ψ valid signed message from the signer, and at most $\psi' - 1$ verification results on signed messages from the target verifier. This should be contrasted to public key signature schemes in which an adversary is allowed to obtain at most $poly(k)$, where k is a security parameter, valid signed messages, and an unlimited number of verification results using a publicly known verification-key.

3 Security notions and their relations

3.1 The model

As mentioned in the previous section, we consider the following simplified model of signature schemes:

Definition 7 A *signature scheme* Π consists of ($\mathcal{U}$, TA, $\mathcal{M}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{A}$, Gen, Sig, Ver):

1. **Notation:**
   - $\mathcal{U} = \{S, V_1, V_2, \ldots, V_n\}$ is a finite set of users, where S is a signer and $V_i (1 \leq i \leq n)$ are verifiers,
   - TA is a trusted authority,
   - $\mathcal{M} = \{\mathcal{M}_k\}_{k \in \boldsymbol{N}}$ is a sequence of finite sets of possible messages, where $\mathcal{M}_k \subset \{0,1\}^{l_M(k)}$, and $l_M(k)$ is a polynomial of k. Hereafter, k means a security parameter.
   - $\mathcal{X} = \{\mathcal{X}_k\}_{k \in \boldsymbol{N}}$ is a sequence of finite sets of possible signing-keys. Here, $\mathcal{X}_k \subset \{0,1\}^{l_X(k)}$, and $l_X(k)$ is a polynomial of k,
   - $\mathcal{Y} = \{\mathcal{Y}_k\}_{k \in \boldsymbol{N}}$ is a sequence of finite sets of possible verification-keys. Here, $\mathcal{Y}_k \subset \{0,1\}^{l_Y(k)}$, and $l_Y(k)$ is a polynomial of k,
   - $\mathcal{A} = \{\mathcal{A}_k\}_{k \in \boldsymbol{N}}$ is a sequence of finite sets of possible signatures. Here, $\mathcal{A}_k \subset \{0,1\}^{l_A(k)}$, and $l_A(k)$ is a polynomial of k,
   - Gen is a *key generation algorithm* which on input a security parameter 1^k, outputs a signing-key and verification-keys,
   - $Sig : \mathcal{X} \times \mathcal{M} \longrightarrow \mathcal{A}$ is a *signing algorithm*,
   - $Ver : \mathcal{Y} \times \mathcal{M} \times \mathcal{A} \longrightarrow \{true, false\}$ is a *verification algorithm*.
2. **Key Generation and Distribution by TA:** The TA generates a signing-key x for the signer S, and a verification-key y_{V_i} for each verifier V_i using Gen. Here, Gen is a probabilistic algorithm which produces, on input 1^k, where k is a security parameter, keys $(x, y_{V_1}, y_{V_2}, \ldots, y_{V_n})$ of matching signing and verifying keys, where $x \in \mathcal{X}_k$ and $y_{V_i} \in \mathcal{Y}_k$ for $1 \leq i \leq n$. TA then transmits the signing-key x to the signer S and the verification-key y_{V_i} to the verifier V_i in a secure way. After delivering these keys, TA may erase the keys $(x, y_{V_1}, y_{V_2}, \ldots, y_{V_n})$ from his memory. The signer keeps secret his signing-key, and each verifier keeps secret his verification-key.
3. **Signature Generation:** For a message $m \in \mathcal{M}_k$, the signer S generates a signature $a = Sig(x, m) \in \mathcal{A}_k$ by using the signing-key x in conjunction with Sig. The pair (m, a) is regarded as a signed message. Here, we assume that Sig is deterministic, but in general it might be randomized. If it is deterministic, for a message m and a signing-key x, the signature $a = Sig(x, m)$ is uniquely determined, while in the case of a randomized algorithm, each time a different signature can be produced for the same message.
4. **Signature Verification:** On receiving (m, a) from the signer S, a verifier V_j checks whether a is valid by using his verification-key $y_{V_j} \in \mathcal{Y}_k$. More precisely, V_j accepts (m, a) as a valid signed message if and only if $Ver(y_{V_j}, m, a) = true$. Here, we assume that Ver is deterministic.

In addition, in the above model a trusted party (or an arbiter) is selected among verifiers. When a dispute occurs, the trusted party can resolve the dispute with his verification-key by following the resolution-rule described in Requirement 1.

Let ψ be a number up to which the signer is allowed to generate signatures, and ψ' be a number up to which each verifier is allowed to check received signatures, respectively, and let ω be the number of possible colluders among users. Let

$\mathcal{W} := \{W \subset \mathcal{U} | \ |W| \le \omega\}$. Each element of $\mathcal{W}$ represents a group of possibly collusive users. For a set $\mathcal{T}$ and a non-negative integer t, let $\wp_t^{\mathcal{T}} := \{T \subset \mathcal{T} \mid |T| \le t\}$ be the family of all subsets of $\mathcal{T}$ whose cardinalities are less than or equal to t. Of course, the empty set $\emptyset$ is always contained in $\wp_t^{\mathcal{T}}$.

3.2 A strong security notion

With notations above, we can now discuss security notions for unconditionally secure signature schemes. We start with introducing *exponentially negligible functions* in order to strictly describe a *small error probability* in Requirement 2.

Definition 8 (Exponentially Negligible Function) Let $\epsilon(k)$ be a function defined over the positive integers $k \in \boldsymbol{N}$ that takes non-negative real numbers. Then, $\epsilon(k)$ is called *exponentially negligible* if there exists an integer k_0 and some constant a $(1 < a)$ such that $\epsilon(k) \le \frac{1}{a^k}$ for all $k \ge k_0$.

Using notations we have introduced, we now formulate the strong security notion in our signature model as follows:

Definition 9 (Strong Security) Let k be a security parameter and $\epsilon(k)$ an exponentially negligible function. For simplicity, we will denote $\epsilon(k)$ by ϵ.

1) For $W \in \mathcal{W}$ such that $V_j, S \notin W$, we define $P_1^{strong}(V_j, W)$ as

$$P_1^{strong}(V_j, W) := \max_{y_W} \max_{M_S = \{(m_S, a_S)\} \in \wp_{\psi}^{\mathcal{M}_k \times \mathcal{A}_k}} \max_{M_{V_j} = \{(m_{V_j}, a_{V_j})\} \in \wp_{\psi'-1}^{\mathcal{M}_k \times \mathcal{A}_k}}$$
$$\max_{M_{V_1}, \ldots, M_{V_l}, \ldots, M_{V_n} \in \wp_{\psi'}^{\mathcal{M}_k \times \mathcal{A}_k} (l \ne j)} \max_{(m,a)} \Pr(V_j \text{ accepts } (m, a)$$
$$\mid y_W, M_S, M_{V_j}, M_{V_l}, \{Ver(y_{V_l}, m_{V_l}, a_{V_l}) | (m_{V_l}, a_{V_l}) \in M_{V_l}\} \ (1 \le l \le n, l \ne j))$$

where M_S is taken over $\wp_{\psi}^{\mathcal{M}_k \times \mathcal{A}_k}$ such that any element of M_S is a valid signed message; M_{V_j} is taken over $\wp_{\psi'-1}^{\mathcal{M}_k \times \mathcal{A}_k}$ such that $Ver(y_{V_j}, m_{V_j}, a_{V_j}) = false$ for any $(m_{V_j}, a_{V_j}) \in M_{V_j}$; M_{V_l} is taken over $\wp_{\psi'}^{\mathcal{M}_k \times \mathcal{A}_k}$ for $1 \le l \le n$, $l \ne j$; and (m, a) runs over $\mathcal{M}_k \times \mathcal{A}_k$ such that $(m, a) \notin M_S$ and $(m, a) \notin M_{V_j}$. Note that the condition $(m, a) \notin M_S$ means that for any $(m_S, a_S) \in M_S$ either $m \ne m_S$, or $m = m_S$ and $a \ne a_S$ holds. Next we define

$$P_1^{strong} := \max_{V_j, W} P_1^{strong}(V_j, W).$$

2) For $W \in \mathcal{W}$ such that $V_j \notin W$ and $S \in W$, we define $P_2^{strong}(V_j, W)$ as

$$P_2^{strong}(V_j, W) := \max_{x} \max_{y_{W-\{S\}}} \max_{M_{V_j} = \{(m_{V_j}, a_{V_j})\} \in \wp_{\psi'-1}^{\mathcal{M}_k \times \mathcal{A}_k}}$$
$$\max_{M_{V_1}, \ldots, M_{V_l}, \ldots, M_{V_n} \in \wp_{\psi'}^{\mathcal{M}_k \times \mathcal{A}_k} (l \ne j)} \max_{(m,a)} \Pr(V_j \text{ accepts } (m, a)$$
$$\mid x, y_{W-\{S\}}, M_{V_j}, M_{V_l}, \{Ver(y_{V_l}, m_{V_l}, a_{V_l}) | (m_{V_l}, a_{V_l}) \in M_{V_l}\} \ (1 \le l \le n, l \ne j))$$

where $M_{V_j} = \{(m_{V_j}, a_{V_j})\}$ is taken over $\wp_{\psi'-1}^{\mathcal{M}_k \times \mathcal{A}_k}$ such that $Ver(y_{V_j}, m_{V_j}, a_{V_j}) = false$ for any $(m_{V_j}, a_{V_j}) \in M_{V_j}$; M_{V_l} is taken over $\wp_{\psi'}^{\mathcal{M}_k \times \mathcal{A}_k}$ for $1 \leq l \leq n$, $l \neq j$; and $(m, a) \in \mathcal{M}_k \times \mathcal{A}_k$ runs over invalid signed messages such that $(m, a) \notin M_{V_j}$. We define $P_2^{strong} := \max_{V_j, W} P_2^{strong}(V_j, W)$.

Then, a signature scheme Π is said to be (n, ω, ψ, ψ')*-secure* if

$$\max\{P_1^{strong},\ P_2^{strong}\} \leq \epsilon$$

3.3 Relations among security notions

One of the purposes in this paper is to clarify which is the strongest among all the security notions that have appeared in unconditionally secure authentication codes and signature schemes. We focus on security notions for the following notable schemes: multireceiver authentication codes (MRA) [9][24], Johansson's scheme [18], Wang and Safavi-Naini's scheme [31] and Hanaoka, Shikata, Zheng and Imai's scheme [15]. Specifically, we analyze a relation among our strong security notion and those of MRA, Johansson's scheme, Wang and Safavi-Naini's scheme, Hanaoka, Shikata, Zheng and Imai's scheme, respectively.

We describe security notions of those schemes as follows. Let Π be a signature scheme (or an authentication code) along with our signature model. Then, Π is said to be $(n, \omega, \psi)^{MRA}$*-secure* if the success probability of all attacks considered in MRA [9] [24] is exponentially negligible under the following conditions: there exists at most ω colluders among the users; and the number up to which a signer is allowed to generate signatures is ψ. Similarly, Π is said to be $(n, \omega, \psi)^{HSZI}$*-secure* if the success probability of all attacks considered in Hanaoka, Shikata, Zheng and Imai's scheme [15] is exponentially negligible under the same conditions. Also, we can define $(n, \omega, \psi)^{ext}$*-secure* by slightly modifying security notions of Johansson's scheme [18], and Wang and Safavi-Naini's scheme [31] so as to fit our signature model (the precise definition of $(n, \omega, \psi)^{ext}$*-secure* is described in Appendix).

¿From the definitions of security notions for the model in Definition 7, an interesting statement can be obtained:

Theorem 3 *The following relations among security notions hold:*

$$\begin{tikzcd}
(n,\omega,\psi,\psi')^{strong}\text{-secure} \arrow[r, shift left] \arrow[d, shift right] & (n,\omega,\psi)^{HSZI}\text{-secure} \arrow[l, shift left, "/" marking] \arrow[dl] \arrow[d, shift right] \\
(n,\omega,\psi)^{ext}\text{-secure} \arrow[u, shift right, "/" marking] \arrow[r, shift left] & (n,\omega,\psi)^{MRA}\text{-secure} \arrow[l, shift left, "/" marking] \arrow[u, shift right, "/" marking]
\end{tikzcd}$$

where "X-secure $\longrightarrow$ Y-secure" means that X-secure always implies Y-secure, while "X-secure $\not\longrightarrow$ Y-secure" means that there exists a signature scheme which is X-secure but not Y-secure.

A detailed proof will appear in the full version of this paper.

4 Construction

In this section we propose a construction method for signature schemes which is secure in terms of our strong security notion. We describe the key generation algorithm, *Gen*, signing algorithm, *Sig*, and verification algorithm, *Ver*, using the notations introduced in Section 3.1.

- **Key Generation Algorithm**: The key generation algorithm, *Gen*, which, on input 1^k, picks a k-bit prime power q, constructs a finite field $\boldsymbol{F}_q$ with q elements. It also picks uniformly at random $2n$ elements $\mathbf{v}_1^{(1)}, \mathbf{v}_1^{(2)}, \mathbf{v}_2^{(1)}, \mathbf{v}_2^{(2)}, \ldots, \mathbf{v}_n^{(1)}, \mathbf{v}_n^{(2)}$ in ${\boldsymbol{F}_q}^{\omega+\psi'}$ for verifiers $V_1, V_2, \ldots, V_n$, respectively, and constructs two polynomials $F_d(Y_1, Y_2, \ldots, Y_{\omega+\psi'}, Z)$ $(d = 1, 2)$ over $\boldsymbol{F}_q$ with $\omega + \psi' + 1$ variables $Y_1, Y_2, \ldots, Y_{\omega+\psi'}, Z$ as follows:

$$F_d(Y_1, \ldots, Y_{\omega+\psi'}, Z) = \sum_{i=0}^{\psi} \sum_{j=1}^{\omega+\psi'} a_{ij}^{(d)} Z^i Y_j + \sum_{i=0}^{\psi} a_{i0}^{(d)} Z^i \ (d = 1, 2),$$

  where the coefficients $a_{ij}^{(d)}$ are chosen uniformly at random from $\boldsymbol{F}_q$. Then, a signing-key for the signer S is $x := (F_1(Y_1, \ldots, Y_{\omega+\psi'}, Z), F_2(Y_1, \ldots, Y_{\omega+\psi'}, Z))$ and a verification-key for the verifier V_i is $y_{V_i} := (\mathbf{v}_i^{(1)}, \mathbf{v}_i^{(2)}, F_1(\mathbf{v}_i^{(1)}, Z), F_2(\mathbf{v}_i^{(2)}, Z))$ for $1 \leq i \leq n$. The algorithm *Gen* returns $(\boldsymbol{F}_q, x, y_{V_1}, y_{V_2}, \ldots, y_{V_n})$.

We consider the case where $\mathcal{M}_k \subset \boldsymbol{F}_q$.

- **Signing Algorithm**: The signing algorithm Sig which, on input the signing-key $x = (F_1(Y_1, \ldots, Y_{\omega+\psi'}, Z), F_2(Y_1, \ldots, Y_{\omega+\psi'}, Z))$ and a message m, returns a signature $a := (F_1(Y_1, \ldots, Y_{\omega+\psi'}, m), F_2(Y_1, \ldots, Y_{\omega+\psi'}, m))$.
- **Verification Algorithm**: The verification algorithm Ver which, on input (y_{V_i}, m, a), where $a = (F_1(Y_1, \ldots, Y_{\omega+\psi'}, m), F_2(Y_1, \ldots, Y_{\omega+\psi'}, m))$ and $y_{V_i} = (\mathbf{v}_i^{(1)}, \mathbf{v}_i^{(2)}, F_1(\mathbf{v}_i^{(1)}, Z), F_2(\mathbf{v}_i^{(2)}, Z))$, computes evaluation values $e_1^{(d)}, e_2^{(d)}$ $(d = 1, 2)$ as follows:

$$\begin{aligned} e_1^{(d)} &:= F_d(Y_1, \ldots, Y_{\omega+\psi'}, m)|_{(Y_1, \ldots, Y_{\omega+\psi'}) = \mathbf{v}_i^{(d)}} \\ e_2^{(d)} &:= F_d(\mathbf{v}_i^{(d)}, Z)|_{Z=m} \qquad\qquad (d = 1, 2). \end{aligned}$$

  Ver then returns *"true"* if $e_1^{(d)} = e_2^{(d)}$ for $d = 1, 2$, and *"false"* otherwise.

The following theorem proves the security of the above construction in our strong security notion.

Theorem 4 *The above construction results in an (n, ω, ψ, ψ')-secure signature scheme, where ω, ψ, ψ' can be taken in such a way that*

$$0 \leq \omega \leq n, \qquad 0 < \psi < q, \qquad 0 < \psi' \leq q + 1 - \sqrt{q},$$

and the success probability of attacks is less that $1/q$.

Once again a proof for the theorem will be provided in the full version of this paper.

5 Concluding remarks

In this paper, we have established a sound security notion, which is likely to be the strongest possible, by taking into account the security notion for public-key signature schemes and some desirable requirements for signature schemes in the unconditional security setting. And we have examined relationships among security notions which have appeared in unconditionally secure schemes both for authentication and signature. We have demonstrated that our security notion is the strongest among all the notions proposed so far. An interesting aspect is that our security notion includes that of public-key signature schemes. We have further presented a construction method for unconditionally secure signature schemes which is provable secure in our strong security notion.

Acknowledgement

The authors wish to thank Tatsuaki Okamoto for helpful comments on the previous version. We also thank anonymous referees for their helpful comments.

References

1. M. Abe and T. Okamoto, "A signature scheme with message recovery as secure as discrete logarithm", Advances in Cryptology - ASIACRYPT '99, LNCS 1716, pp.378-389, Springer, 1999.
2. M. Bellare and P. Rogaway, "The exact security of digital signatures - How to sign with RSA and Rabin", Advances in Cryptology - EUROCRYPT '96, LNCS 1070, Springer, 1996.
3. E. F. Brickell and D. R. Stinson, "Authentication codes with multiple arbiters," Advances in Cryptology - EUROCRYPT '88, LNCS 330, Springer, pp.51-55, 1988.
4. D. Chaum and H. van Antwerpen, "Undeniable signatures", Advances in Cryptology - CRYPTO '89, Springer, pp. 212-216, 1990.
5. D. Chaum and S. Roijakkers, "Unconditionally secure digital signatures," Advances in Cryptology - CRYPTO'90, LNCS 537, Springer, pp.206-215, 1990.
6. D. Chaum, E. Heijst and B. Pfitzmann, "Cryptographically strong undeniable signatures, unconditionally secure for the signer," Advances in Cryptology - CRYPTO '91, LNCS 576, Springer, pp.470-484, 1991.
7. R. Cramer and V. Shoup, "Signature schemes based on the strong RSA assumption", Proc. of the 6th ACM Conference in Computer and Communication Security, 1999.
8. Y. Desmedt and M. Yung, "Arbitrated unconditionally secure authentication can be unconditionally protected against arbiter's attack," Advances in Cryptology - CRYPTO '90, LNCS 537, Springer, pp.177-188, 1990.
9. Y. Desmedt, Y. Frankel and M. Yung, "Multi-receiver/Multi-sender network security: efficient authenticated multicast/feedback," Proc. of IEEE Infocom'92, pp.2045-2054, 1992.
10. W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory 22, 6, pp. 644-654, 1976.

11. T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” IEEE Transactions on Information Theory, 31, 4, pp.469-472, 1985.
12. R. Gennaro, S. Halevi, and T. Rabin “Secure hash-and-sign signatures without the random oracle”, Advances in Cryptology - EUROCRYPT ’99, LNCS 1592, pp.123-139, Springer, 1999.
13. E. N. Gilbert, F. J. MacWilliams and N. J. A. Sloane, “Codes which detect deception,” Bell System Technical Journal, 53, pp.405-425, 1974.
14. S. Goldwasser, S. Micali and R. Rivest, “A digital signature scheme secure against adaptive chosen message attacks”, SIAM J. Comput. 17, 2, pp. 281-308, 1988.
15. G. Hanaoka, J. Shikata, Y. Zheng, and H. Imai, “Unconditionally secure digital signature schemes admitting transferability”, Advances in Cryptology - ASIACRYPT 2000, LNCS 1976, Springer, pp.130-142, 2000.
16. G. Hanaoka, J. Shikata, Y. Zheng, and H. Imai, “Efficient and Unconditionally Secure Digital Signatures and a Security Analysis of a Multireceiver Authentication Code”, to appear in Proc. of Public Key Cryptography, Springer, 2002.
17. T. Johansson, “Lower bounds on the probability of deception in authentication with arbitration”, IEEE Trans. Inform. Theory 40, 5, pp.1573-1585, 1994.
18. T. Johansson, “Further results on asymmetric authentication schemes,” Information and Computation, 151, pp.100-133, 1999.
19. K. Kurosawa, “New bound on authentication code with arbitration,” Advances in Cryptology - CRYPTO ’94, LNCS 839, Springer, pp.140-149, 1994.
20. K. Kurosawa and S. Obana, “Combinatorial bounds for authentication codes with arbitration,” Advances in Cryptology - EUROCRYPT ’95, LNCS 921, Springer, pp.289-300, 1995.
21. R. Rivest, A. Shamir and L. Adleman, “A method for obtaining digital signature and public-key cryptosystems,” Communication of the ACM, vol.21, no.2, pp.120-126, 1978.
22. B. Pfitzmann, “Sorting out signature schemes”, Proc. of the First ACM Conference on Computer and Communications Security, ACM Press, pp.74-86, 1993.
23. D. Pointcheval and J. Stern, “Security proofs for signature schemes”, Advances in Cryptology - EUROCRYPT ’96, LNCS 1070, Springer, 1996.
24. R. Safavi-Naini and H. Wang, “New results on multi-receiver authentication codes,” Advances in Cryptology - EUROCRYPT ’98, LNCS 1403, pp.527-541, Springer, 1998.
25. R. Safavi-Naini and H. Wang, “Broadcast authentication in group communication,” Advances in Cryptology - ASIACRYPT ’99, LNCS 1716, Springer, pp.399-411, 1999.
26. R. Safavi-Naini and H. Wang, “Multireceiver authentication codes: models, bounds, constructions and extensions,” Information and Computation, 151, pp.148-172, 1999.
27. G. J. Simmons, “Authentication theory/coding theory,” Advances in Cryptology - CRYPTO ’84, LNCS 196, Springer, pp.411-431, 1984.
28. G. J. Simmons, “Message authentication with arbitration of transmitter/ receiver disputes,” Advances in Cryptology - EUROCRYPT ’87, Springer, pp.151-165, 1987.
29. G. J. Simmons, “A Cartesian construction for unconditionally secure authentication codes that permit arbitration,” Journal of Cryptology 2, pp.77-104, 1990.
30. R. Taylor, “Near optimal unconditionally secure authentication,” Advances in Cryptology - EUROCRYPT ’94, LNCS 950, Springer, pp.244-253, 1994.
31. Y. Wang and R. Safavi-Naini, “A^3-codes under collusion attacks” Advances in Cryptology - ASIACRYPT ’99, LNCS 1716, Springer, pp.390-398, 1999.

Appendix: A security notion for extended A^2 and A^3-codes

Johansson's model [18] for a class of broadcast authentication scheme is an extension of that of A^2-codes. Also, Wang and Safavi-Naini's model [31] is an extension of that of A^3-codes. Taking into account security notions of these models, we arrive at the following security notion by modifying their notions so as to fit our signature model. In that sense, the following security notion can also be regarded as that of an extension of A^2 and A^3-codes.

Definition 10 Let k be a security parameter and $V_{arb} \in \mathcal{U} - \{S\}$ an arbiter (or a trusted party).

1. Success probability of impersonation and substitution by verifiers: For $W \in \mathcal{W}$ such that $V_j, V_{arb}, S \notin W$, we define $P_{I,S}^{ext}(V_j, W)$ as

$$P_{I,S}^{ext}(V_j, W) := \max_{y_W} \max_{M \in \wp_{\psi}^{\mathcal{M}_k}, \{(m,a)\}_{m \in M}} \max_{(m',a')} \Pr(V_j \text{ accepts } (m', a') \mid y_W, \{(m,a)\}_{m \in M})$$

   where M is taken over $\wp_{\psi}^{\mathcal{M}_k}$, $\{(m,a)\}_{m \in M}$ is a set of $|M|$ valid signed messages with $m \in M$, and m' is taken over $\mathcal{M}_k$ satisfying $m' \notin M$. Then, $P_{I,S}^{ext}$ is defined as

$$P_{I,S}^{ext} := \max_{V_j, W} P_{I,S}^{ext}(V_j, W)$$

   where V_j is taken over all receivers including V_{arb} and W is taken over $\mathcal{W}$ satisfying $S, V_j, V_{arb} \notin W$.
2. Success probability of attack by colluders including the signer: For $W \in \mathcal{W}$ such that $V_j, V_{arb} \notin W$ and $S \in W$, we define

$$P_{signer}^{ext}(V_j, W) := \max_{x} \max_{y_{W-\{S\}}} \max_{(m,a)} \Pr(V_j \text{ accepts } (m, a) \mid x, y_{W-\{S\}})$$

   where m is taken over $\mathcal{M}_k$ and $a \in \mathcal{A}_k$ is taken such that (m, a) is an invalid signed message, i.e. $a \neq Sig(x, m)$. Then, P_{signer}^{ext} is defined as follows:

$$P_{signer}^{ext} := \max_{V_j, W} P_{signer}^{ext}(V_j, W),$$

   where V_j is taken over all receivers including V_{arb} and W is taken over $\mathcal{W}$ satisfying $V_j, V_{arb} \notin W$ and $S \in W$.
3. Success probability of attack against the sender: For $W \in \mathcal{W}$ such that $S \notin W$, we define

$$P_{arbiter\_1}^{ext}(W) := \max_{y_W} \max_{M \in \wp_{\psi}^{\mathcal{M}_k}, \{(m,a)\}_{m \in M}} \max_{(m',a')} \Pr((m', a') \text{ is a valid signed message generated by } S \mid y_W, \{(m,a)\}_{m \in M})$$

where M is taken over $\wp_{\psi}^{\mathcal{M}_k}$, $\{(m,a)\}_{m\in M}$ is a set of $|M|$ valid signed messages with $m \in M$ and m' is taken over $\mathcal{M}_k$ satisfying $m' \notin M$. Then, $P^{ext}_{arbiter\_1}$ is defined as

$$P^{ext}_{arbiter\_1} := \max_{W} \ P^{ext}_{arbiter\_1}(W),$$

where W is taken over $\mathcal{W}$ such that $S \notin W$. Here, we note that W runs over $\mathcal{W}$ including the cases $V_{arb} \in W$.

4. Success probability of attack against a verifier by colluders including the arbiter: For $W \in \mathcal{W}$ such that $V_{arb} \in W$ and $S, V_j \notin W$, we define

$$P^{ext}_{arbiter\_2}(V_j, W) := \max_{y_{V_{arb}}}, \max_{y_{W-\{V_{arb}\}}} \max_{M\in\wp_{\psi}^{\mathcal{M}_k}, \{(m,a)\}_{m\in M}} \max_{(m',a')} \Pr(V_j \text{ accepts } (m',a') \mid y_{V_{arb}}, y_{W-\{V_{arb}\}}, \{(m,a)\}_{m\in M}),$$

where M is taken over $\wp_{\psi}^{\mathcal{M}_k}$, $\{(m,a)\}_{m\in M}$ is a set of $|M|$ valid signed messages with $m \in M$ and m' is taken over $\mathcal{M}_k$ satisfying $m' \notin M$. Then, $P^{ext}_{arbiter\_2}$ is defined as

$$P^{ext}_{arbiter\_2} := \max_{V_j, W} \ P^{ext}_{arbiter\_2}(V_j, W),$$

where V_j is taken over all receivers except V_{arb}, and W is taken over $\mathcal{W}$ satisfying $V_{arb} \in W$ and $S, V_j \notin W$.

5. Success probability of attack against a verifier by colluders including both the arbiter and the sender: For $W \in \mathcal{W}$ such that $V_{arb}, S \in W$ and $V_j \notin W$, we define

$$P^{ext}_{arbiter\_3}(V_j, W) := \max_{x} \max_{y_{V_{arb}}} \max_{y_{W-\{V_{arb},S\}}} \max_{(m,a)} \Pr(V_j \text{ accepts } (m,a) \mid x, y_{V_{arb}}, y_{W-\{V_{arb},S\}}),$$

where (m,a) is taken over $\mathcal{M}_k \times \mathcal{A}_k$ such that (m,a) is not accepted by V_{arb}, i.e. $Ver(m, a, y_{V_{arb}}) = false$. Then, $P^{ext}_{arbiter\_3}$ is defined as

$$P^{ext}_{arbiter\_3} := \max_{V_j, W} \ P^{ext}_{arbiter\_3}(V_j, W),$$

where V_j is taken over all verifiers except V_{arb}, and W is taken over $\mathcal{W}$ such that $V_{arb}, S \in W$ and $V_j \notin W$.

Let $\epsilon(k)$ be an exponentially negligible function. For simplicity, we denote $\epsilon(k)$ by ϵ. A signature scheme Π along with our signature model is called $(n, \omega, \psi)^{ext}$-*secure* if the following condition is satisfied: under the conditions that there exists at most ω colluders and that the signer is allowed to generate at most ψ signatures, the inequality below holds.

$$\max\{P^{ext}_{I,S}, P^{ext}_{signer}, P^{ext}_{arbiter\_1}, P^{ext}_{arbiter\_2}, P^{ext}_{arbiter\_3}\} \le \epsilon$$