SCIS 2006 The 2006 Symposium on Cryptography and Information Security
Hiroshima, Japan, Jan. 17-20, 2006
The Institute of Electronics, Information and Communication Engineers

Universally Composable Blind Signatures

Seiji Doi * Yoshifumi Manabe * † Tatsuaki Okamoto * †

Abstract— This paper shows that the security of blind signatures is, as defined by Juels, Luby and Ostrovsky, truly weaker than the security in the universal composability (UC) framework (i.e., define the ideal functionality of blind signatures), which was introduced by Canetti. That is, we formulate the security of blind signatures in the UC framework, and show that the class of UC-secure blind signatures is a proper subset of that of secure (in the sense of Juels et al.) blind signatures. In addition, we introduce a stronger security definition (stronger blindness; SB-security) of blind signatures than that by Juels et al. and show that SB-security is more suitable in many applications than Juels et al's. This paper then shows that SB-security of blind signatures is also truly weaker than the security in the UC framework.

Keywords: universal composition, blind signatures, two-party protocols

1 Introduction

1.1 Background

The concept of blind signatures was first introduced by Chaum [6]. In blind signatures, a user interacts with a signer and gets a signature. During the interaction the signer cannot see the content of the document that he is signing.

Pointcheval and Stern introduced a security notion called *one-more forgery* [12] to define the security of blind signatures. In a blind signature scheme that is secure against one-more forgery, a user cannot get more signatures than the number of interactions with the signer [12]. Juels, Luby and Ostrovsky introduced a general definition of blind signatures and formulated two security properties: *blindness* and *non-forgeability* [9]. Blindness means that the adversary cannot link a signature to a corresponding message and non-forgeability means that the adversary cannot get more signatures than the number of interactions with the signer even if the adversary undertakes, with the signer, adaptive, parallel and arbitrarily interleaved interactive protocols. They called such an attack *an adaptive interleaved chosen-message attack*. They also showed that there exists a blind signature protocol that is secure (i.e. satisfies the two above security properties) against an adaptive interleaved chosen-message attack if one-way trapdoor permutations exist.

Canetti introduced the universal composability (UC) framework as a new approach for analyzing the security of cryptographic primitives and protocols [2]. In the UC framework, it is guaranteed that a secure primitive/protocol maintains its security even if other primitives/protocols run concurrently. It allows us to combine protocols and construct large systems easily.

Since UC security requirement is very strong, it raises the new question of whether conventional security notions satisfy UC security. Canetti gave a positive answer to this question on digital signatures and public-key encryption (PKE). He showed that a UC-secure signature scheme is equivalent to a secure (*existential unforgeable against chosen-message attacks*) signature scheme, and that UC-secure PKE is equivalent to secure (*semantically secure against chosen-ciphertext attacks*) PKE [2].

On the other hand, as a negative answer, Canetti, Kushilevitz and Lindell showed that no (non-trivial) two-party protocol can be UC-secure in the plain model where we use no setup assumptions except for authenticated communication [4].

Since a blind signature scheme is not just a two-party protocol nor a simple primitive like signatures and PKE, it is far from trivial to show the relationship between UC security and the conventional security of blind signatures.

1.2 Our results

In this paper, we show that the conventional security of blind signatures is truly weaker than UC security. That is, first we formulate the security of blind signatures in the UC framework (i.e., define the ideal functionality of blind signatures), and show that the class of UC-secure blind signatures is a proper subset of that of secure (in the sense of [9]) blind signatures, assuming a one-way trapdoor permutation.

We then introduce a stronger security definition (stronger blindness; SB-security) of blind signatures than that by Juels et al. [9]. SB-security is more appropriate in many applications (e.g., electronic cash and voting) than Juels et al.'s. We then show that SB-security of blind signatures is also truly weaker than security in the UC framework. That is, we show that the class of UC-secure blind signatures is a proper subset of that of

* Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501 Japan
† NTT Laboratories, 1-1 Hikari-no-oka, Yokosuka, 239-0847 Japan

SB-secure blind signatures, assuming a one-way trapdoor permutation.

2 Preliminaries

2.1 Universal Composability

The Universal Composability security framework was proposed by Canetti [2]. The main concern was to create a new approach to the assessment of cryptographic protocols. In traditional approaches, protocols were defined for isolated execution. This is sufficient for small or simple protocols and evaluating their security is relatively easy. However, recent cryptographic systems consist of many protocols and thus these protocols must run in a complex environment where many protocols may run simultaneously or concurrently. In such a situation, analyzing protocols by the traditional approach is a virtually impossible. The security of the compound protocol must be verified even if every component protocol is secure by itself.

In the UC framework, a protocol is defined as standalone but it is guaranteed to run securely in any arbitrary environment. The framework guarantees that a compound protocol consisting of just secure protocols is also secure, which is called *the universal composition theorem*. Thus, when analyzing a multi-protocol system, we analyze each protocol in isolation, and the protocol maintains its security even if it is running within an environment wherein another protocol may be running concurrently. The universal composition theorem ensures that a cryptographic system consisting of secure protocols is secure. The advantage of using this framework is that protocols can be handled as modules. As mentioned above, we can easily construct large secure systems by combining small and simple secure protocols.

In the framework there are two worlds: the ideal world and the real world. In the real world, there are parties and the adversary. Parties interact with each other according to the protocol. The adversary can control the delivery of the messages that the parties send and may corrupt some parties during the execution. When a party is corrupted, it has to follow the instructions of the adversary. In the ideal world, there are parties, an ideal functionality, and the simulator. An ideal functionality is the ideal behavior of the function that a protocol should achieve. Parties cannot directly interact with each other. They exchange messages and interact via the functionality. The simulator can control message delivery and can corrupt parties as well as the adversary. Furthermore we also consider the middle world, called the hybrid world, which lies between the ideal world and the real world. In the hybrid world, parties interact according to a protocol as in the real world.

The UC framework also sets the special party called the environment, which interacts with each world and outputs a bit after the interaction. It can give as input to the parties arbitrary strings and can read all parties' outputs. Additionally it interacts with the adversary or simulator. It tries to distinguish these two worlds by the interactions. We say that the environment distinguishes the two worlds if the difference between the probability that the environment outputs 1 after the interaction with the real world and that after the interaction with the ideal world is non-negligible. We say a protocol is *UC-secure* if for any adversary there exists a simulator such that for any environment the ideal world and the real world are indistinguishable. Moreover, we say such a protocol *UC-realizes* the functionality in the ideal world.

2.2 Blind Signatures

The blind signature scheme was first introduced by Chaum as a significant cryptographic technique [6]. Using this technique, a signer can sign a document without seeing its contents, while in the standard signature scheme the signer can read the document before signing it.

A model of blind signatures

Juels, Luby and Ostrovsky formally defined the blind signature scheme [9]. In their definition, a blind signature scheme is the four-tuple (Signer, User, *Gen, Verify*). $Gen(1^k)$ is a probabilistic polynomial time algorithm which outputs a pair of public key and secret key, (pk, sk). Signer and User are polynomially-bounded probabilistic interactive Turing machines. pk produced by *Gen* is given to Signer and User as a common input. Additionally Signer is given a corresponding key sk, and User is given a message m. The length of all inputs must be polynomial in the security parameter 1^k. They interact with each other according to the protocol. After the computation, Signer outputs either *completed* or *non-completed*, and User outputs either *fail* or $\sigma(m)$. When verifying the signature, $Verify(pk, m, \sigma(m))$ is a signature verification algorithm which outputs either *accept* or *reject*. It is required that for any message m and random choice of *Gen* if both Signer and User follow the protocol then Signer always outputs *completed* and *Verify* always outputs *accept*.

Security properties

Jules, Luby and Ostrovsky formulated the security of blind signatures: blindness and non-forgeability. They defined that;

Definition 1. *A blind signature scheme is secure if for all constants c and for all probabilistic polynomial-time algorithms A, there exists a security parameter $k_{c,A}$ such that for all $k > k_{c,A}$ the following two properties hold:*

Blindness *Let $b \in_R \{0, 1\}$, where b is kept from A. A executes the following experiment:*

1. $(pk, sk) \leftarrow Gen(1^k)$.
2. $\{m_0, m_1\} \leftarrow A(1^k, pk, sk)$.
3. *We denote by $\{m_b, m_{1-b}\}$ the same two documents $\{m_0, m_1\}$, ordered according to the value of b, which is still kept from A. $A(1^k, pk, sk, m_0, m_1)$ engages in two parallel interactive protocols, the first with User(pk, m_b) and the second with User(pk, m_{1-b}).*

4. *If the first User outputs on her private tape $\sigma(m_b)$ and the second User outputs $\sigma(m_{1-b})$ then A is given as an additional input $\{\sigma(m_b), \sigma(m_{1-b})\}$ ordered according to the corresponding (m_0, m_1) order.*
5. *A outputs bit $\tilde{b}$.*

The probability, taken over the choice of b, over coin-flips of Gen, the coin-flips of A, and coin-flips of both users in step 3, that $\tilde{b} = b$ is at most $\frac{1}{2} + \frac{1}{k^c}$.

Non-forgeability *A executes the following experiment:*

1. *$(pk, sk) \leftarrow Gen(1^k)$.*
2. *$A(pk)$ engages in polynomially many adaptive, parallel and arbitrarily interleaved interactive protocols with polynomially many copies of Signer, where A decides in an adaptive fashion when to stop. Let l denote the number of executions, that Signer outputs completed at the end of step 2.*
3. *A outputs a collection $\{(m_1, \sigma(m_1)), \cdots, (m_j \; \sigma(m_j))\}$ subject to the constraint that $\forall i_1, i_2$, $m_{i_1} \neq m_{i_2}$ $(i_1, i_2 \in \{1, \cdots, j\}, i_1 \neq i_2)$, and that that all $(m_i, \sigma(m_i))$ for $1 \leq i \leq j$ are accepted by Verify$(pk, m_i, \sigma(m_i))$.*

It follows that the probability, taken over, coin-flips of Gen, the coin-flips of A and the coin-flips of Signer, that $j > l$ is at most $\frac{1}{k^c}$.

Notice that we here introduced non-forgeability as normal non-forgeability (i.e., not strong non-forgeability) while in [9], strong non-forgeability is defined as one of the security properties. Jules, Luby and Ostrovsky proved the following proposition [9].

Proposition 1. *Assume that one-way trapdoor permutations exist. Then there exists a polynomial-time blind signature scheme, secure against an adaptive interleaved chosen-message attack.*

2.3 Stronger security definition

In Definition 1, adversaries use pk output by Gen. When such a trusted party that issues pk does not exist, another party which is potentially corrupted may generate pk. This situation does not match blindness in Definition 1. Thus, we define as follows a stronger security definition where we allow adversaries to generate pk.

Strong Blindness Let $b \in_R \{0, 1\}$, where b is kept from A. A executes the following experiment:

1. $\{pk, m_0, m_1\} \leftarrow A(1^k)$
2. We denote by $\{m_b, m_{1-b}\}$ the same two documents $\{m_0, m_1\}$, ordered according to the value of b, which is still kept from A. $A(1^k, pk, m_0, m_1)$ engages in two parallel interactive protocols, the first with User(pk, m_b) and the second with User(pk, m_{1-b}).
3. If the first User outputs on her private tape $\sigma(m_b)$ and the second User outputs $\sigma(m_{1-b})$ then A is given as an additional input $\{\sigma(m_b)$ $\sigma(m_{1-b})\}$ ordered according to the corresponding (m_0, m_1) order.
4. A outputs a bit $\tilde{b}$.

The probability, taken over the choice of b, the coin-flips of A, and coin-flips of both users in step 2, that $\tilde{b} = b$ is at most $\frac{1}{2} + \frac{1}{k^c}$.

Definition 2. *A blind signature scheme is SB-secure if for all constants c and for all probabilistic polynomial-time algorithms A, there exists a security parameter $k_{c,A}$ such that for all $k > k_{c,A}$ Strong Blindness and Non-forgeability hold.*

Lemma 1. *If a blind signature is SB-secure then it is secure (in the sense of Definition 1.).*

Proof. We show that if there exists an adversary that breaks security we can construct another adversary that breaks SB-security. Because non-forgeability is the same for both types of security, we concentrate on blindness. Assume that A^* breaks blindness. Here, we can construct A that breaks strong blindness as follows. Instead of Gen, A randomly generates sk and pk, and sends them to A^*. When A^* outputs $\{m_0, m_1\}$, A outputs $\{pk, m_0, m_1\}$. A delivers messages from Users to A^* and vice versa during the interaction with them. When A^* outputs $\tilde{b}$ A also outputs it. The probability that A succeeds in this guess is the same as that A^*'s guess succeeds. Therefore if A^* breaks blindness then A breaks strong blindness. □

3 UC Blind Signatures

In this section we define the ideal functionality of blind signatures. The functionality provides a verification key and at the request to issue a signature it chooses a value from the distribution of the output of a signature scheme. Thus, the behavior of the functionality depends on the signature scheme used in it. We describe $\mathcal{F}_{\text{BSIG}}^{\Pi}$ in Figure 1, which is the ideal functionality of blind signatures parameterized by a signature scheme Π. In Figure 1, Gen and Σ is specified by Π. At the request of Signer, $\mathcal{F}_{\text{BSIG}}^{\Pi}$ sends a verification key to Signer and all Users. When User sends $\mathcal{F}_{\text{BSIG}}^{\Pi}$ a request with a verification key to issue a signature, $\mathcal{F}_{\text{BSIG}}^{\Pi}$ checks that the verification key is valid. If it is valid, then $\mathcal{F}_{\text{BSIG}}^{\Pi}$ randomly chooses σ from Σ and sends it to User. At the request of a verifier, it returns an output according to the following scenario. In Figure 1, 2-(a) is the scenario where v' and (m', σ') are valid. 2-(b) occurs when Signer is not corrupted and the document has not been signed before. 2-(c) means that if some result for this input is already stored, then $\mathcal{F}_{\text{BSIG}}^{\Pi}$ returns the same result. 2-(d) occurs when Signer is corrupted and no result is stored. In this case $\mathcal{F}_{\text{BSIG}}^{\Pi}$ returns the result decided by the simulator because Signer is corrupted and the simulator can control the verification result. The main role of $\mathcal{F}_{\text{BSIG}}^{\Pi}$ is to act as an anonymous message storage. It stores pairs of messages and signatures. Each signature can be considered as a tag. At each signature generation, Signer can know the signature but cannot know the message signed. Upon receiving a request for the verification of a signature and a message, $\mathcal{F}_{\text{BSIG}}^{\Pi}$ checks if the pair of the message and the signature is already stored. Surely, $\mathcal{F}_{\text{BSIG}}^{\Pi}$ is designed to have strong blindness and non-forgeability.

$\mathcal{F}_{\mathrm{BSIG}}^{\Pi}$

With User $P_i (i = 1, \cdots, n)$, Signer Q, Simulator S.

- **Key Generation**
  1. In the first activation, expect to receive (`KeyGen`) from Q; upon receipt send it to S.
  2. Upon receiving (`Key`, v) from S, store (Q, v), send (`Verification Key`, v) to all User and Q.
- **Signature Generation**
  1. Upon receiving (`Sign`, m, v) from P_i, check that v is already stored. If so (`Request`, v, P_i) to S. Otherwise, send (`Reject`) to P_i.
  2. If (`Signature`, `Completed`) is received from S, randomly choose σ from $\Pi(m, v)$, store $(m, \sigma, v, 1)$, send (`Signature`, σ, m) to P_i, and send (`Completed`) to Q (Given pk and m, $\Pi(m, pk)$ is the random variable of outputs of honest User with input pk after interacting with honest Signer with input sk, where (pk, sk) is an output of $Gen(1^k)$ and the probability of $\Pi(m, pk)$ is over the randomness of User, Signer and Gen). Otherwise, send (`Fail`) to P_i and send (`Not-completed`) to Q.
- **Signature Verification**
  1. Upon receiving (`Verify`, m', σ', v') from P_j, send it to S.
  2. Assume (`Verified`, m', ϕ) is received from S.
     (a) if $v' = v$ and $(m', \sigma', v', 1)$ is stored, then set $f = 1$.
     (b) else if $v' = v$, m' is never signed and Q is not corrupted, then store $(m', \sigma', v', 0)$ and set $f = 0$.
     (c) else if (m', σ', v', f') is stored, then set $f = f'$.
     (d) else set $f = \phi$ and record (m', σ', v', ϕ)
  3. If $f = 1$ then send (`Accept`) to P_j, otherwise send (`Reject`) to P_j.

Fig. 1. Ideal functionality of blind signatures

4 UC-secure blind signatures are not equivalent to secure blind signatures

In this section we show;

Theorem 1. *The class of UC-secure blind signatures is a proper subset of that of secure blind signatures for static adversaries, assuming one-way trapdoor permutations.*

First, we show;

Lemma 2. *There exists a protocol that is secure but not UC-secure for static adversaries, assuming the one-way trapdoor permutations.*

Proof. We present an instance of the JLO protocol and show that it is not UC-secure. The JLO protocol uses *the two-party completeness theorem* for realizing blindness. This theorem was shown by Yao and Goldreich, Micali and Wigderson [13, 8]. They said that for any two parties A and B where A is given secret input x and B is given secret input y, and for any polynomial-time computable function $g(\cdot, \cdot)$ there exists a protocol for computing $g(x, y)$ such that nothing except for the output of the function is revealed to the parties. In [8], Goldreich, Micali and Wigderson showed at first that assuming the existence of a trapdoor permutation, any functionality can be securely realized by a protocol in the semi-honest adversary model. In the semi-honest adversary model, even if a party is corrupted it works according to a prescribed protocol and the adversary only has access to the state of corrupted parties. Next, they constructed *a protocol compiler* that transforms a protocol in the semi-honest model to work securely in the malicious adversary model. The compiler makes each party prove, in zero-knowledge manner, that each message it sends was honestly made from its input, its random choice, and the messages it has received so far. It prevents a corrupted party from diverging from the protocol without being detected by another honest party and the adversary is limited to semi-honest behavior. Goldreich, Micali and Wigderson showed that zero-knowledge proof protocol can be constructed assuming one-way permutations [7].

We denote by Σ_1 the instance of the JLO protocol that uses as a black box the Blum's zero-knowledge proof protocol [1, 3]. The zero-knowledge proof protocol uses commitments that do not require common strings, namely that exists in the plain model. We then construct Σ_2 as follows. We replace the commitment protocol in the zero-knowledge proof protocol in Σ_1 by $\mathcal{F}_{\mathrm{COM}}$, the ideal functionality of commitments defined in [3]. Here, we show that Σ_2 UC-realizes $\mathcal{F}_{\mathrm{BSIG}}$. In [3], it was proved that the Blum's zero-knowledge proof protocol in which commitment protocols are replaced by $\mathcal{F}_{\mathrm{COM}}$ UC-realizes $\mathcal{F}_{\mathrm{ZK}}$, the ideal functionality of zero-knowledge proofs [5]. Moreover, Canetti, Lindell, Ostrovsky and Sahai proved that by using $\mathcal{F}_{\mathrm{ZK}}$, any two-party functionality can be UC-realized [5]. They introduced *the universally composable protocol compiler* which is a UC version of the protocol compiler [8]. Notice that if no party is corrupted and the execution runs honestly then any environment cannot distinguish. Thus, Σ_2 UC-realizes $\mathcal{F}_{\mathrm{BSIG}}$.

Here, we assume that Σ_1 UC-realizes $\mathcal{F}_{\mathrm{BSIG}}$. Then it holds that Σ_1 and Σ_2 are indistinguishable. Recall that the difference between Σ_1 and Σ_2 is the part of commitments; Σ_1 uses an actual commitment protocol and Σ_2 uses $\mathcal{F}_{\mathrm{COM}}$. Thus, the fact that Σ_1 and Σ_2 are indistinguishable means that the commitment protocol in the zero-knowledge proof protocol in Σ_1 UC-realizes $\mathcal{F}_{\mathrm{COM}}$. If the commitment protocol does not UC-realize $\mathcal{F}_{\mathrm{COM}}$, that is, there exist Z^* that distinguishes the

interaction with the protocol and that with $\mathcal{F}_{\mathrm{COM}}$, we can construct Z that distinguishes the interaction with Σ_1 and that with Σ_2 by using Z^*'s output. Canetti and Fischlin, however, proved that no two-party protocol can UC-realize $\mathcal{F}_{\mathrm{COM}}$ in the plain model [3]. From the contradiction we can say that Σ_1 does not UC-realize $\mathcal{F}_{\mathrm{BSIG}}$.

Next, we show;

Lemma 3. *If a blind signature protocol is UC-secure then it is also secure in the sense of Definition 1.*

Proof. We show that if a protocol is not secure it is not UC-secure. Assume that a protocol is not secure, that is, there exists an adversary that breaks blindness or non-forgeability. First, consider the case that blindness is broken and denote by A^* the adversary that breaks blindness with the probability ε_1. We then can construct an environment Z^* as follows. Z^* corrupts a signer and gets a verification key and a corresponding secret key (v^*, sk^*). Z^* then sends (v^*, sk^*) to A^*. When A^* generates (m_0, m_1), Z^* activates a user with input $(\mathtt{Sign}, m_0, v^*)$ and with $(\mathtt{Sign}, m_1, v^*)$ and gets $\sigma(m_0)$ and $\sigma(m_1)$. Z^* sends $\{\sigma(m_0), \sigma(m_1)\}$ to A^*, which means that Z^* sets $b = 0$. When A^* outputs $\tilde{b}$, Z^* outputs it. If Z^* interacts with the ideal world then A^* has no advantage in terms of the guess so that the probability of $\tilde{b} = 0$ is $\frac{1}{2}$. On the other hand, if Z^* interacts with the real world then the probability of $\tilde{b} = 0$ is $\frac{1}{2} + \varepsilon_1$. Therefore, Z^* can distinguish the two worlds with probability ε_1.

Next, consider the case that non-forgeability is broken. We then can construct Z^* that distinguishes the two worlds as follows. Let A^* be the adversary that breaks non-forgeability with the probability ε_2. First, Z^* corrupts a user after v^* is issued by activating a signer. Z^* then simulates the interaction with Signer for A^* by delivering the outputs of A^* to corrupted User and vice versa. When A^* outputs a list $\{(m_1, \sigma(m_1)), \cdots, (m_j, \sigma(m_j))\}$, where j is truly greater than the number of interactions between the user and the signer, Z^* verifies all pairs with v^* and gets the results. If all results are (`Accepted`), Z^* outputs 1 and otherwise outputs 0. If Z^* interacts with the ideal world then for a forged pair the result is (`Reject`). Thus, the probability that Z^* outputs 1 is always 0. On the other hand, if Z^* interacts the real world then the probability that Z^* outputs 1 is ε_2. Therefore, Z^* can distinguish with ε_2.

Consequently, we conclude that if a protocol is not secure then the protocol is not UC-secure. □

Lemma 2 and Lemma 3 lead Theorem 1.

5 UC-secure blind signatures are not equivalent to SB-secure blind signatures

In this section we show;

Theorem 2. *The class of UC-secure blind signatures is a proper subset of that of SB-secure blind signatures for static adversaries, assuming one-way permutations.*

First, we consider the following protocol π_1. As input, a user is given m and a signer is given 1^k where k is a security parameter. Before starting the computations, the signer sends a public key, pk, to the user and they write pk on their own output tapes. At the end of the computations, the user additionally writes $\sigma(m)$ on its output tape. The user learns nothing about the signer's input and the signer learns nothing about m. We can actually construct such a protocol by using the two-party completeness theorem (see section 4) [8], assuming one-way permutations.

Claim. π_1 is SB-secure.

Proof. We show if π_1 is not SB-secure then the two-party completeness theorem is broken. First, consider the case that strong blindness is broken. This case implies that a malicious signer can get some information about m, which is clearly contradicts the property of zero-knowledge proofs. Next, consider the case that non-forgeability is broken. It contradicts the two-party completeness theorem unless π_1 is a protocol that a randomly generated signature passes the verification with the non-negligible probability. □

Here, we show;

Lemma 4. *π_1 is not UC-secure.*

Proof. We show that π_1 satisfies the condition under that no two-party protocol can be UC-secure. Canetti, Kushilevitz and Lindell defined *unpredictability* and proved that an unpredictable probabilistic two-party function f cannot be UC-realized [4]. Let $f_k : X \times X \to \{0,1\}^*$ be a probabilistic function that is parameterized by k. They showed the following two definitions;

Definition 3. *$x_1(x_2) \in X$ is said a $P_1(P_2)$-safe value for $p(\cdot)$ and k if for every $x_2(x_1) \in X$ and all possible output values $v \in \{0,1\}^*$ it holds that* $\Pr[f_k(x_1, x_2) \neq v] > \frac{1}{p(k)}$.

Definition 4. *$f = f_k$ is unpredictable if there exists a polynomial $p(\cdot)$ such that for infinity $k's$, there exist P_1-safe values and P_2-safe values for $p(\cdot)$ and k.*

Now, the output of π_1 includes pk which is generated by a probabilistic algorithm Gen so that π_1 is a probabilistic protocol. We consider f_k, P_1, x_1, P_2, x_2 and v as π_1, a user, m, a signer, 1^k and $(pk, \sigma(m))$ respectively. If a security parameter k is specified 1^k is fixed. First, we show that there exists P_1-safe value for $p(k) = k^c$ (where, c is a constant). Assume that there exists no P_1-safe value, that is for any m there exist $1^{\hat{k}}$ and $(\hat{\sigma}, \hat{pk})$ such that $\Pr[\pi_1(m, 1^{\hat{k}}) \neq (\hat{\sigma}, \hat{pk})] < \frac{1}{p(k)}$. Now,

$$\Pr[\pi_1(m, 1^{\hat{k}}) \neq (\hat{\sigma}, \hat{pk})] < \frac{1}{p(k)}$$
$$\Leftrightarrow \Pr[\pi_1(m, 1^{\hat{k}}) = (\hat{\sigma}, \hat{pk})] > 1 - \frac{1}{p(k)} \qquad (1)$$

Here, we take the strong blindness experiment. After choosing (m_0, m_1) A randomly chooses k^*. A then simulates $\pi_1(m_0, 1^{k^*})$ and gets $(\sigma^*(m_0), pk^*)$. A outputs

(pk^*, m_0, m_1). When he is given (σ_b, σ_{1-b}), if $k^* = \hat{k}$, that is, $(k^*, \sigma^*(m_0)) = (\hat{k}, \hat{\sigma})$, then, from the expression (1), A succeeds the guess with probability over $1 - \frac{1}{p(k)}$. Here, the probability of $k^* = \hat{k}$ is $\frac{1}{k}$. Thus, the probability that A succeeds the guess is greater than $\frac{1}{k}\left(1 - \frac{1}{p(k)}\right)$, which contradicts to strong blindness.

Next, we show that there exists a P_2-safe value. As the same with above, we assume that 1^k is not a P_2-safe value, that is, for any 1^k there exist $\hat{m}$ and $(\hat{\sigma}, \hat{pk})$ such that

$$\Pr[\pi_1(\hat{m}, 1^k) = (\hat{\sigma}, \hat{pk})] > 1 - \frac{1}{p(k)}. \qquad (2)$$

Here, we take non-forgeability experiment. A randomly chooses k^*, locally runs $Gen(1^{k^*})$ and gets (pk', sk'). A randomly generates m^*, simulates the behavior of the signer using (pk', sk') and gets $(m^*, \sigma(m^*))$. After one interaction with the signer A gets $\{(m_0, \sigma(m_0))\}$, where $\pi_1(m_0, 1^{k_0}) = (\sigma(m_0), pk_0)$. Now, pk_0 is output by $Gen(1^k)$ regardless of m, thus, from the expression (2), the probability of $pk_0 = \hat{pk}$, that is, $(\sigma(m_0), pk_0) = (\hat{\sigma}, \hat{pk})$ is greater than $1 - \frac{1}{p(k)}$. Thus, if $k^* = k_0$ the probability of $pk' = \hat{pk}$, that is, $(\sigma(m^*), pk') = (\hat{\sigma}, \hat{pk})$ is greater than $1 - \frac{1}{p(k)}$. Here, $Verify(m_0, \sigma(m_0), pk_0)$ surely passes. The probability that $Verify(m^*, \sigma(m^*), pk')$, is greater than $\left(1 - \frac{1}{p(k)}\right)^2$. Because k^* is randomly chosen the probability of $k^* = k_0$ is $\frac{1}{k}$. Thus, the probability that $(m^*, \sigma(m^*))$ passes the verification is greater than $\frac{1}{k}\left(1 - \frac{1}{p(k)}\right)^2$, which contradicts to non-forgeability.

Therefore, π_1 is unpredictable so that we can say that π_1 is not UC-secure. □

Next, we show;

Lemma 5. *If a blind signature protocol is UC-secure then it is also SB-secure.*

Proof. As in the proof of Lemma 3, we show that if there exist an adversary that breaks SB-security we can construct an environment that distinguishes the two worlds. Let A^* be an adversary that breaks SB-security with ε. We then construct Z^* that distinguishes the two worlds. Because non-forgeability is the same in Definition 1 and in Definition 2, if A^* breaks non-foregeability we can construct Z^* that distinguishes the two worlds as described in Lemma 3. Assume that A^* breaks strong blindness. We then construct Z^* as follows. Z^* runs A^* locally in itself. First, Z^* corrupts a signer and receives A^*'s output. When A^* outputs $\{m_0, m_1, pk^*\}$ then Z^* activates a user with m_0 and m_1. In each interaction, Z^* sends a pk^* as a verification key. Z^* delivers messages from the user to A^* and vice versa during each interaction. When the user outputs two signatures $\sigma(m_0)$ and $\sigma(m_1)$, Z^* sends $\{\sigma(m_0), \sigma(m_1)\}$ to A^* in this order. When A^* outputs $\tilde{b}$, Z^* outputs it. If Z^* interacts with the ideal world then A^* has no advantage in terms of the guess so that the probability of $\tilde{b} = 0$ is $\frac{1}{2}$. On the other hand, if Z^* interacts with the real world then the probability of $\tilde{b} = 0$ is $\frac{1}{2} + \varepsilon$. Therefore, Z^* can distinguish the two worlds with probability ε. □

Lemma 4 and Lemma 5 lead Theorem 2.

6 Conclusions

In this paper, we defined the functionality of blind signatures in the UC framework and showed that conventionally security of blind signatures is truly weaker than UC-security. we also defined stronger security properties for blind signatures that guarantee blindness under the condition that honest key generation does not exist, and showed that the stronger security is also truly weaker than UC-security.

References

1. Manuel Blum, 'Coin flipping by telephone', IEEE Spring COMPCOM, pp.133-137, Feb. 1982.
2. Ran Canetti, 'Universally Composable Security: A new paradigm for Cryptographic Protocols', 42nd FOCS, 2001. Full version available at *http://eprint.iacr.org/2000/067/*.
3. Ran Canetti and Marc Fischlin, 'Universally Composable Commitments', Proceedings of CRYPTO 2001.
4. Ran Canetti, Eyal Kushilevitz and Yeheuda Lindell, 'On the Limitations of Universally Composable Two-Party Computation Without Set-up Assumptions', Proceedings of EUROCRYPT 2003.
5. Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, Amit Sahai, 'Universally Composable Two-Party and Multi-Party Secure Composition', Proceedings of STOC 02.
6. David Chaum, 'Blind Signatures for Untraceable Payments', Proceedings of CRYPTO 82.
7. Oded Goldreich, Sivio Micali and Avi Wigderson, 'Proofs that Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems', Journal of ACM, Vol. 38, No. 1, pp. 691–729, 1991.
8. Oded Goldreich, Sivio Micali and Avi Wigderson, 'How to Play Any Mental Game', Proceedings of STOC 87.
9. Ari Juels, Michael Luby and Rafail Ostrovsky, 'Security of Blind Digital Signatures', Proceedings of CRYPTO 97.
10. Leslie Lamport, 'Constructing digital signatures from one-way functions', SRI intl. CSL-98, October 1979.
11. Moni Naor and Moti Yung, 'Universal One-Way Hash Functions and Their Cryptographic Applications', Proceedings of STOC 89.
12. David Pointcheval and Jacques Stern, 'Security Arguments for Digital Signatures and Blind Signatures', Proceedings of EUROCRUPT 96.
13. Andrew Chi-Chih Yao, 'How to Generate and Exchange Secrets', Proceeding of 27th FOCS, 1986, pp.162-167.