

Fair Partially Blind Signatures

Markus Rückert* and Dominique Schröder**

TU Darmstadt, Germany

markus.rueckert@cased.de schroeder@me.com

Abstract. It is well-known that blind signature schemes provide full anonymity for the receiving user. For many real-world applications, however, this leaves too much room for fraud. There are two generalizations of blind signature schemes that compensate this weakness: fair blind signatures and partially blind signatures. Fair blind signature schemes allow a trusted third party to revoke blindness in case of a dispute. In partially blind signature schemes, the signer retains a certain control over the signed message because signer and user have to agree on a specific part of the signed message.

In this work, we unify the previous well-studied models into a generalization, called fair partially blind signatures. We propose an instantiation that is secure in the standard model without any setup assumptions. With this construction, we also give a positive answer to the open question of whether fair blind signature schemes in the standard model exist.

Keywords Blind signatures, generic construction, security model

1 Introduction

Blind signatures, proposed by Chaum in 1982 [7], are interactive signature schemes between a signer and a user with the property that the message is hidden from the signer (blindness). Simultaneously, the user cannot produce more signatures than interactions with the signer took place (unforgeability). As one of the first applications of blind signatures, Chaum proposed untraceable digital payment (e-cash). In this context, Chaum pointed out that such a high degree of privacy enables an adversary to doubly spend an electronic coin if no countermeasures are taken. Further fraud scenarios are given in [25]. Another potential loophole in ordinary blind signature schemes, first mentioned by Abe and Okamoto [2], is that the signer entirely loses control over the signed message. Consider, e.g., vouchers with a predetermined expiration date. The signer is willing to blindly sign the voucher but wants to ensure that the receiving user cannot control the expiry date. For both weaknesses, two different countermeasures have been proposed, namely, fair blind and partially blind signatures.

* This work was supported by CASED (www.cased.de).

** Dominique Schröder was supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG).

Fair blind signatures, suggested by Stadler, Piveteau, and Camenisch [24], involve a trusted third party, which is able to revoke blindness in case of a dispute between signer and user. The revocation mechanism works in two directions: it either uniquely identifies an obtained signature from the signer’s view of a specific session (signature tracing), or connects a given signature with a specific session of the protocol (session tracing). The trusted party is *offline*, i.e., there is no initial setup phase and the signature issue protocol does not involve this party. It is only appealed to in case of irregularities, such as fraud or other crimes.

A second approach by Abe and Fujisaki [1], named partially blind signatures, compensates for the potential loophole that the signer entirely loses control over the signed message. Reconsider the example of vouchers with a predetermined expiration date. Signer and user both agree on some piece of information, such as the expiry date. Here, verification only works if user and signer agree on the same date. As soon as the user tries to change this auxiliary information, verification will fail.

The goal of our work is to establish a unified model of fair partially blind signatures that encompasses all previous concepts and their security models and to find provably secure instantiations. We motivate the need for the generalization of both concepts with the following simple example that can be transferred to other, more complex, application scenarios: consider the scenario where a bank issues electronic coins of different value. With ordinary blind signatures, the bank has to apply different keys for different values and cannot be sure that no malicious user doubly spends the coin. The use of partially blind signatures schemes allows the signer to work with a single key, while including the value of the coin as auxiliary information. Still, criminal investigations will be hindered by the customer’s irrevocable anonymity. On the other hand, using fair blind signatures allows for revocable anonymity but then it is again necessary for the signer to use multiple keys.

Thus, with fair partially blind signature schemes, we are able to get the best of both worlds. A bank that employs fair partially blind signatures only needs a single key pair, while simultaneously being able to remove blindness if necessary. Note that the revocation will probably not be done by the bank itself but by a law enforcement agency or a trusted notary. We believe that combining both concepts is most suitable for real-world applications, where the individual needs of customers (blindness), service providers (partial control), and those of the authorities (fairness) have to be satisfied.

RELATED WORK. Since Stadler, Piveteau, and Camenisch described the idea of *fair blind signatures* in 1995 [24], many constructions have been proposed, e.g., [14,19,21,20,3,17]. Unfortunately, some of these constructions cannot be considered “blind” in the sense of Juels et al. [18] and thus Abe and Ohkubo [3] developed a formal security model. Unfortunately, all previous results either provide only security arguments, or are provably secure in the random oracle model, which is discouraged by the work of Canetti, Goldreich, and Halevi [6]. We are not aware of any instantiation in the standard model. *Partially blind signatures*, due to Abe and Fujisaki [1], are also well-studied and several instantiations have

been proposed, e.g., [1,2,8,22]. Recently, the first instantiation without random oracles or setup assumptions has been proposed in [22].

In [17], Hufschmitt and Traoré consider dynamic fair blind signatures, a concept that is inspired by dynamic group signatures [5]. They require that each user of the blind signature scheme has to register before being able to obtain signatures. We also discuss the relation to our model.

CONTRIBUTION. We propose a novel security model, which is a generalization of the well-studied models of Juels, Luby, and Ostrovsky [18] and Pointcheval and Stern [23] for blind signatures, the model of Abe and Ohkubo [3] for fair blind signatures, and that of Abe and Fujisaki [1] for partially blind signatures. With our model for fair partially blind signatures, we provide a unified framework that can be used to instantiate blind, fair blind, and partially blind signature schemes under a strong security model. We present a provably secure instantiation within this model. The construction, which is inspired by the works of Fischlin [11] and Hazay et al. [16], relies on general assumptions and is provably secure in the standard model without any setup assumptions. By eliminating the auxiliary information, our scheme solves the longstanding problem of instantiability of fair blind signature schemes in the standard model [24]. Removing the trusted third party also yields the first partially blind signature scheme based on general assumption which is provably secure in the standard model, again without any setup assumptions. Independently of our work, Fuchsbauer and Vergnaud construct the first efficient instantiation of fair blind signatures in the standard model [15].

ORGANIZATION. After recalling basic definitions and notations, Section 2 introduces the concept of fair partially blind signatures along with a security model and a discussion of the various security properties. Section 3 is a warm-up for Section 4 to make it more accessible. Then, we provide a provably secure instantiation from general assumption in Section 4.

2 Fair Partially Blind Signatures

NOTATION. We use the following notation for interactive executions between algorithms \mathcal{X} and \mathcal{Y} . The joint execution between \mathcal{X} and \mathcal{Y} is denoted by $(a, b) \leftarrow \langle \mathcal{X}(x), \mathcal{Y}(y) \rangle$, where x is the private input of \mathcal{X} and y is the private input of \mathcal{Y} . The private output of algorithm \mathcal{X} is a and b is the private output of \mathcal{Y} . If an algorithm \mathcal{Y} can invoke an unbounded number of executions of the interactive protocol with \mathcal{X} , then we write $\mathcal{Y}^{\langle \mathcal{X}(x), \cdot \rangle^\infty}(y)$. Accordingly, $\mathcal{X}^{\langle \cdot, \mathcal{Y}(y_0) \rangle^1, \langle \cdot, \mathcal{Y}(y_1) \rangle^1}(x)$ means that \mathcal{X} can execute arbitrarily ordered executions with $\mathcal{Y}(y_0)$ and $\mathcal{Y}(y_1)$, but only once. An algorithm is efficient, if it runs in probabilistic polynomial-time.

2.1 Definition

The concept of fair partially blind signature schemes generalizes the idea of partially blind signatures and fair blind signatures. A partially blind signature

scheme is a blind signature scheme such that signer and user agree on some piece of information, denoted with *info*. The signature should only be valid if *both* parties use this specific *info* element during the signature issue protocol.

Fair blind signatures have a revocation mechanism, enforceable by a trusted third party, which can revoke blindness upon disputes between signer and user. Revocation works in both directions. Whenever we need to find the signature that corresponds to a certain session, we query the revocation authority with a view of a session and obtain a signature identifier id_{sig} . Now, when given a signature, we can verify whether it corresponds to id_{sig} or not. If, instead, we would like to find the session that resulted in a certain signature, we query the revocation authority with this signature to obtain a session identifier id_{ses} . Again, we can easily verify whether a given session corresponds to id_{ses} or not.

In both cases, the trusted party outputs an identifier, which *uniquely* associates an execution with a signature. In the following definition, we combine both types of blind signatures.

Definition 1 (Fair Partially Blind Signature Scheme). *A fair partially blind signature scheme FPBS consists of the following efficient algorithms:*

Key Generation. $\text{Kg}(1^n)$ generates a key pair (sk, pk) .

Revocation Key Generation. $\text{RevKg}(1^n)$ outputs a key pair $(rsk, rpik)$.

Signature Issuing. The joint execution of the algorithms $\mathcal{S}(sk, rpik, \text{info})$ and $\mathcal{U}(pk, rpik, m, \text{info})$ with message $m \in \{0, 1\}^n$ and information element $\text{info} \in \{0, 1\}^n$, generates the private output σ of the user and the private view *view* of the signer, $(\text{view}, \sigma) \leftarrow \langle \mathcal{S}(sk, rpik, \text{info}), \mathcal{U}(pk, rpik, m, \text{info}) \rangle$.

Verification. $\text{Vf}(pk, rpik, m, \text{info}, \sigma)$ outputs a bit, indicating the validity of σ .

Signature Revocation. SigRev takes as input the signer's view *view* of a session and the secret revocation key *rsk*. It outputs an identifier id_{sig} , which corresponds to the signature that the user obtained in this same session.

Session Revocation. When queried with a message *m*, with an information element *info*, with a (valid) signature σ and with the secret revocation key *rsk*, SesRev discloses the session identifier id_{ses} .

Signature Tracing. On input $(\text{id}_{\text{sig}}, \sigma)$, SigVf outputs a bit indicating whether id_{sig} matches σ .

Session Tracing. On input $(\text{id}_{\text{ses}}, \text{view})$, SesVf returns 1 if id_{ses} matches to *view* and 0 otherwise.

It is assumed that a fair partially blind signature scheme is complete:

- For any $n \in \mathbb{N}$, any $(sk, pk) \leftarrow \text{Kg}(1^n)$, any $(rsk, rpik) \leftarrow \text{RevKg}(1^n)$, any $\text{info} \in \{0, 1\}^n$, any message $m \in \{0, 1\}^n$ and any σ output by $\mathcal{U}(pk, rpik, m, \text{info})$ after the joint execution with $\mathcal{S}(sk, rpik, \text{info})$, we have $\text{Vf}(pk, rpik, m, \text{info}, \sigma) = 1$.
- For any $n \in \mathbb{N}$, any $(sk, pk) \leftarrow \text{Kg}(1^n)$, any $(rsk, rpik) \leftarrow \text{RevKg}(1^n)$, any message $m \in \{0, 1\}^n$, any $\text{info} \in \{0, 1\}^n$, any σ output by $\mathcal{U}(pk, rpik, m, \text{info})$ in the joint execution with $\mathcal{S}(sk, rpik, \text{info})$, and any *view*, as observed by the signer, we have $\text{SigVf}(\text{SigRev}(rsk, \text{view}), \sigma) = 1$ and $\text{SesVf}(\text{SesRev}(rsk, m, \text{info}, \sigma), \text{view}) = 1$.

Note that the algorithms `SigRev` and `SesRev` may get additional public parameters as input if needed. As for `info`, we point out that signer and user agree on it before engaging in the signature issue protocol. This process is application specific and will not be addressed here. Furthermore, note that restricting m and `info` to $\{0,1\}^n$ is no limitation in practice because one can always extend it to $\{0,1\}^*$ via a collision resistant hash function.

2.2 Security of Fair Partially Blind Signatures

Our security model for fair partially blind signature schemes is related to the security model for blind signatures [18,23,13], to the model for partially blind signatures [2], and to that for fair blind signatures [3]. In fact, it unifies the various types of blind signature schemes into a single model.

Security of blind signature schemes requires two properties, namely, unforgeability and blindness [18,23]. Unforgeability demands that a malicious user should not be able to produce more signatures than there were successful interactions with the signer. Here, the user is allowed to choose the messages as well as the information elements adaptively.

In the context of partially blind signatures, we want unforgeability to be stronger than in the classical case since “recombination” attacks should be considered. That is, an adversarial user should not be able to generate a valid signature for a *new* `info` instead of just for a new message. Depending on the application, one might even want to consider the stronger notion of “strong unforgeability”. There, the adversary also wins if it outputs a *new* signature.

Definition 2 (Unforgeability). *A fair partially blind signature scheme FPBS is called unforgeable if for any efficient algorithm \mathcal{U}^* the probability that experiment $\text{Forge}_{\mathcal{U}^*}^{\text{FPBS}}(n)$ evaluates to 1 is negligible (as a function of n) where*

Experiment $\text{Forge}_{\mathcal{U}^*}^{\text{FPBS}}(n)$
 $(rsk, rpki) \leftarrow \text{RevKg}(1^n)$
 $(sk, pk) \leftarrow \text{Kg}(1^n)$
For each `info`, let k_{info} denote the number of successful, complete interactions
 $((m_1, \text{info}, \sigma_1), \dots, (m_{k_{\text{info}}+1}, \text{info}, \sigma_{k_{\text{info}}+1})) \leftarrow \mathcal{U}^{*\langle \mathcal{S}(sk, rpki, \cdot), \cdot \rangle}^\infty(pk, rsk, rpki)$
Return 1 iff
 $m_i \neq m_j$ for $1 \leq i < j \leq k_{\text{info}} + 1$, and
 $\forall i (pk, rpki, m_i, \text{info}, \sigma_i) = 1$ for all $i = 1, 2, \dots, k_{\text{info}} + 1$.

Partial blindness is a generalization of blindness. It allows the malicious signer \mathcal{S}^* to choose a public key, two messages m_0, m_1 , and `info` on its own. The signer then interacts with two honest user instances. Based on a coin flip b , the first user obtains a signature for m_b and the second obtains one for m_{1-b} . If both protocol executions were successful, \mathcal{S}^* gets the signatures σ_0, σ_1 for m_0, m_1 , respectively, in the original order. The task is to guess b .

Definition 3 (Partial Blindness). *FPBS is partially blind if for any efficient algorithm \mathcal{S}^* (working in modes `find`, `issue`, and `guess`) the probability that the following experiment $\text{FPBlind}_{\mathcal{S}^*}^{\text{FPBS}}(n)$ outputs 1 is negligibly close to $1/2$, where*

Experiment $\text{FPBlind}_{\mathcal{S}^*}^{\text{FPBS}}(n)$

$(rsk, rp_k) \leftarrow \text{RevKg}(1^n)$

$(pk, m_0, m_1, \text{info}, \text{st}_{\text{find}}) \leftarrow \mathcal{S}^*(\text{find}, rp_k, 1^n)$

$b \leftarrow \{0, 1\}$

$\text{st}_{\text{issue}} \leftarrow \mathcal{S}^*(\langle \cdot, \mathcal{U}(\dots, m_b) \rangle^1, \langle \cdot, \mathcal{U}(\dots, m_{1-b}) \rangle^1)(\text{issue}, \text{st}_{\text{find}}, rp_k)$

Let σ_b and σ_{1-b} be the private outputs of $\mathcal{U}(pk, rp_k, m_b)$ and $\mathcal{U}(pk, rp_k, m_{1-b})$.
and let view_0 and view_1 be the corresponding views of \mathcal{S}^* .

Set $(\sigma_0, \sigma_1) = (\perp, \perp)$ if $\sigma_0 = \perp$ or $\sigma_1 = \perp$

$b^* \leftarrow \mathcal{S}^*(\text{guess}, \sigma_0, \sigma_1, \text{st}_{\text{issue}})$

Return 1 iff $b = b^*$.

As already mentioned in [3], it is possible to strengthen the notion of partial blindness even further by giving the adversarial signer access to conditional signature and session revocation oracles. There, the signer is allowed to query the oracles on any but the “relevant” signatures σ_0, σ_1 or sessions $\text{view}_0, \text{view}_1$. We call this stronger notion *strong partial blindness*.

Fairness of (partially) blind signatures consists of signature and session traceability. Signature traceability means that, given the revocation key rsk every message-signature pair can be related to exactly one execution of the signature issue protocol. This intuition is formalized in the following experiment, where a malicious user tries to output a valid message-signature tuple such that no matching protocol instance can be found. The second possibility to win the experiment is to output two message-signature pairs corresponding to the same protocol instance.

In the context of partially blind signatures, we give a strengthened definition in the following sense. The malicious user even succeeds with a valid message-signature tuple (m, info, σ) such that no session corresponding to the *same* info exists. Thus, the adversary merely needs to find a message-signature tuple, whose view matches that of $\text{info}' \neq \text{info}$. Observe that the adversary has access to rsk .

For each info , let k_{info} be the number of successful, complete interactions and let V_{info} be the set of corresponding protocol views. Let $V_* = \bigcup V_{\text{info}}$ contain all views and assume that the adversary \mathcal{U}^* asks its signature oracle at most $|V_*| \leq q$ times.

Definition 4 (Signature Traceability). *FPBS is signature traceable if for any efficient algorithm \mathcal{U}^* the probability that experiment $\text{SigTrace}_{\mathcal{U}^*}^{\text{FPBS}}(n)$ evaluates to 1 is negligible (as a function of n), where*

Experiment $\text{SigTrace}_{\mathcal{U}^*}^{\text{FPBS}}(n)$
 $(rsk, rp_k) \leftarrow \text{RevKg}(1^n)$
 $(sk, pk) \leftarrow \text{Kg}(1^n)$
 $((m_1, \text{info}_1, \sigma_1), (m_2, \text{info}_2, \sigma_2)) \leftarrow \mathcal{U}^{*(\mathcal{S}(sk, rp_k, \cdot), \cdot)^\infty}(pk, rsk, rp_k)$
Return 1 if
for one of the message-signature tuples, denoted by (m, info, σ) , we have:
 $\text{Vf}(pk, rp_k, m, \text{info}, \sigma) = 1$ and $\text{SigVf}(\text{id}_{\text{Sig}}, \sigma) = 0$
for all $\text{id}_{\text{Sig}} \leftarrow \text{SigRev}(\text{view}, rsk)$ with $\text{view} \in V_{\text{info}}$,
or if $(m_1, \text{info}_1) \neq (m_2, \text{info}_2)$ and
 $\text{Vf}(pk, rp_k, m_i, \text{info}_i, \sigma_i) = 1$, $i = 1, 2$, and there exists a $\text{view} \in V_*$ s.t.
 $\text{SigVf}(\text{id}_{\text{Sig}}, \sigma_0) = \text{SigVf}(\text{id}_{\text{Sig}}, \sigma_1) = 1$ where $\text{id}_{\text{Sig}} \leftarrow \text{SigRev}(rsk, \text{view})$.

Given, on the other hand, a view of a protocol execution, the trusted party can uniquely determine the message-signature tuple that was generated. This property is called session traceability. In the following experiment, the adversarial user tries to output a message-signature tuple such that either no session corresponds to this signature or such that at least two sessions can be associated to it. Again, we give a stronger definition by letting the attacker win if the output signature matches to a session for a different info and we let it have rsk .

Definition 5 (Session Traceability). FPBS is session traceable if for any efficient algorithm \mathcal{U}^* the probability that experiment $\text{SesTrace}_{\mathcal{U}^*}^{\text{FPBS}}(n)$ evaluates to 1 is negligible (as a function of n), where

Experiment $\text{SesTrace}_{\mathcal{U}^*}^{\text{FPBS}}(n)$
 $(rsk, rp_k) \leftarrow \text{RevKg}(1^n)$
 $(sk, pk) \leftarrow \text{Kg}(1^n)$
 $(m, \text{info}, \sigma) \leftarrow \mathcal{U}^{*(\mathcal{S}(sk, rp_k, \cdot), \cdot)^\infty}(pk, rsk, rp_k)$
Let $\text{id}_{\text{Ses}} \leftarrow \text{SesRev}(rsk, m, \text{info}, \sigma)$.
Return 1 iff
 $\text{Vf}(pk, rp_k, m, \text{info}, \sigma) = 1$ and $\text{SesVf}(\text{id}_{\text{Ses}}, \text{view}) = 0$ for all $\text{view} \in V_{\text{info}}$
or there exist distinct $\text{view}_1, \text{view}_2 \in V_*$ such that
 $\text{SesVf}(\text{id}_{\text{Ses}}, \text{view}_1) = \text{SesVf}(\text{id}_{\text{Ses}}, \text{view}_2) = 1$.

As already observed in [17] and others, unforgeability is implied by traceability. Thereby, we obtain the following definition of “security”.

Definition 6. FPBS is secure if it is partially blind, signature traceable, and session traceable.

RELATION TO THE SECURITY NOTION OF HUFSCMITT AND TRAORÉ. As already mentioned in the introduction, the security model of [17] for fair blind signatures is inspired by dynamic group signatures [5]. In contrast to blind signatures, group signature consider a fixed group where each member is statically registered. Dynamic group signatures allow in addition members to dynamically join and leave the group. Although this may be common in many practical scenarios, it complicates the definitions. In our work, we want to keep the definitions as simple as possible (and we want to follow the widely accepted notion

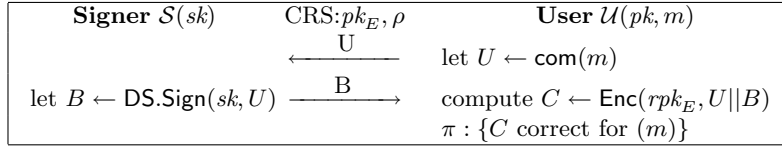


Fig. 1. Fischlin’s protocol.

of blind signatures). With our approach, we also demonstrate that registration procedures are not at all necessary to construct provably secure and flexible fair and partially blind signatures. Therefore, we do not consider their alternative security model.

3 A Warm-Up — Fischlin’s Blind Signature Scheme

Our blind signature scheme relies on an elegant construction due to Fischlin [11] that is provably secure in the *common reference string* (CRS) model. We review a simplified version of Fischlin’s scheme ([11] presents a *strongly* unforgeable scheme) and then propose a partial solution to fair partially blind signatures. Fischlin’s scheme performs the following steps:

Setup. The CRS contains a public key pk_E for a semantically secure public-key encryption scheme, and a string ρ used as a CRS for the non-interactive zero-knowledge proof.

Key Generation. The key generator runs the key generation algorithm of a standard signature scheme $(pk, sk) \leftarrow \text{DS.Kg}(1^n)$ and returns both keys.

Signing. The interactive protocol in which the user \mathcal{U} derives a signature on a message m is as follows:

- \mathcal{U} computes a commitment $U \leftarrow \text{com}(m)$ and sends it to the signer.
- \mathcal{S} signs the commitment $B \leftarrow \text{DS.Sign}(sk, U)$ and sends B .
- \mathcal{U} checks whether the signature is valid and aborts if it is invalid. Otherwise, the user computes $C \leftarrow \text{Enc}(rpk_E, U||B)$ together with a NIZK proof π that a valid signature is encrypted properly. Then it outputs the signature (C, π) .

Verification. To verify that (C, π) is a valid signature on m (w.r.t. pk), the verification algorithm checks that π is a valid proof (w.r.t. ρ).

We briefly discuss the security of the scheme and then how to derive a fair partially blind signature scheme out of Fischlin’s blind signature scheme.

Unforgeability of the scheme follows from the unforgeability of the regular signature scheme and from the binding property of the commitment. On the other hand, the hiding property of the commitment scheme prevents the malicious signer from learning any information about the message during the issue protocol. Furthermore, the semantically secure encryption scheme hides the encrypted message-signature pair and the non-interactive zero-knowledge proof discloses no information about the plaintext in C .

FAIR PARTIALLY BLIND SIGNATURES: A PARTIAL SOLUTION. In order to adapt Fischlin’s scheme and avoid the CRS, we have several difficulties to overcome. In the following, we describe these problems together with our solutions.

- Removing CRS ρ : We apply the technique of Hazay et al. [16] to remove the CRS. Thus, we rely on ZAPs [9] rather than on NIZKs (a ZAP is a two round witness-indistinguishable proof system, we give a formal definition in the following section).
- Achieving fairness: In order to obtain a “fair” scheme, we have to guarantee that each message-signature pair can, given a revocation key, be related to exactly one execution (and vice versa). To do so, we let the signer choose a random session identifier and (verifiably) include it in the signature as well as in the user’s encryption of the signature.
- Achieving partial message control: We allow the signer to control a certain part *info* of the signed message by including it in its signature. Simultaneously, the user must include the same *info* in the encryption of the message. Thus, one-sided changes to *info* result in an invalid signature.

4 An Instantiation from General Assumptions

Before presenting our generic construction, we review the required primitives.

INDISTINGUISHABLE ENCRYPTION UNDER CHOSEN-PLAINTEXT ATTACKS (IND-CPA). A public-key encryption scheme PKE is secure in the IND-CPA model if no efficient adversary \mathcal{A} can distinguish ciphertexts for messages of its choice. The corresponding experiment gives \mathcal{A} access to an encryption oracle, which is initialized with a public key rpk and a bit b . The encryption oracle takes as input two equally sized messages (m_0, m_1) and returns $\text{Enc}(rpk, m_b)$. The adversary wins if it is able to guess b with probability noticeable greater than $1/2$. We assume in this construction that the message expansion c is deterministic, i.e., the length of the ciphertext is $c(n)$ for all messages of length n .

STRONGLY UNFORGEABLE SIGNATURE SCHEME. A digital signature DS is strongly unforgeable under adaptive chosen message attacks if there is no efficient algorithm, which queries a signing oracle on messages of its choice, that manages to output a *new* signature for a (possibly) queried message. We assume all signature to be encoded as $l(n)$ bit strings.

PROOF OF KNOWLEDGE (PoK). Let’s consider an \mathcal{NP} -Language L with relation $R_L := \{(s, w) \mid s \in L \text{ and } w \text{ is a witness for } s\}$. Informally, we call an interactive proof protocol between a prover \mathcal{P} and a verifier \mathcal{V} a *proof of knowledge* if no malicious prover \mathcal{P}^* can cheat but with negligible probability. If \mathcal{P}^* convinces the verifier with noticeable probability then there exists an efficient algorithm *Extract*, the extractor, which is able to extract a value y from \mathcal{P}^* such that $(x, y) \in R_L$. We refer the interested reader to [4] for further details.

WITNESS-INDISTINGUISHABILITY In the case that an \mathcal{NP} -Language L has at least two witnesses w_1 and w_2 for some string $s \in L$, we can define witness-indistinguishable proofs. Basically, a witness-indistinguishable interactive proof system [12] allows the prover to prove some statement, without revealing the

Experiment $\text{WitInd}_{\mathcal{V}^*}^{\text{ZAP}}$

$b \leftarrow \{0, 1\}$
 $(\rho, s_1, \dots, s_\ell, (w_1^0, \dots, w_\ell^0), (w_1^1, \dots, w_\ell^1), \text{st}_{\text{find}}) \leftarrow \mathcal{V}^*(\text{find}, 1^n)$
 Return 0 if $(s_i, w_i^0), (s_i, w_i^1) \notin R_L$ for $i \in [1, \ell]$.
 $\pi_i \leftarrow \mathcal{P}(\rho, s_i, w_i^b)$ for $i \in [1, \ell]$.
 $b' \leftarrow \mathcal{V}^*(\text{guess}, \text{st}_{\text{find}}, \rho, (\pi_1, \dots, \pi_\ell), (s_1, \dots, s_\ell))$
 Return 1 iff $b' = b$.

Experiment $\text{AdSnd}_{\mathcal{P}^*}^{\text{ZAP}}$

$\rho \leftarrow \mathcal{V}(1^n)$
 $(s, \pi) \leftarrow \mathcal{P}^*(\rho, 1^n)$
 Return 1 iff $\mathcal{V}(\rho, s, \pi) = 1$ and $s \notin L$.

Fig. 2. Experiments describing witness-indistinguishability and adaptive soundness.

witness (w_1 or w_2) used during the proof. This condition even holds if the verifier knows both witnesses. Thus, a witness-indistinguishable proof of knowledge (WI-PoK) hides the used witness and simultaneously allows its extraction.

ZAP. Roughly speaking, a ZAP is a two round public coin witness-indistinguishable protocol [9] with the useful property that the first round (a message from verifier \mathcal{V} to prover \mathcal{P}) can be made universal for all executions and therefore be part of the public key of \mathcal{V} . Dwork and Naor showed that ZAPs can be build upon any trapdoor permutation [9].

Definition 7 (ZAP). Let $L_{p(n)} := L \cap \{0, 1\}^{\leq p(n)}$ for some polynomial p . A ZAP is 2-round public coin witness-indistinguishable protocol for some \mathcal{NP} -language L with associated relation R_L . It consists of two efficient interactive algorithms \mathcal{P}, \mathcal{V} such that

- The verifier \mathcal{V} outputs an initial message ρ on input 1^n ;
- The prover \mathcal{P} gets as input ρ , a statement $s \in L_{p(n)}$, and a witness w such that $(s, w) \in R_L$. It outputs a proof π ;
- The verifier \mathcal{V} outputs a decision bit when queried with ρ, s and π .

A ZAP is complete if for any $n \in \mathbb{N}$ and any $(s, w) \in R_L$, we have $\mathcal{V}(\rho, s, \mathcal{P}(\rho, s, w)) = 1$. Furthermore, ZAPs satisfy the following properties (see Figure 2 for the experiments):

Witness-Indistinguishability. A ZAP is witness-indistinguishable if for any efficient algorithm \mathcal{V}^* (working in modes *find* and *guess*) the probability that experiment $\text{WitInd}_{\mathcal{V}^*}^{\text{ZAP}}$ outputs 1 is negligibly close to $1/2$.

Adaptive Soundness. A ZAP satisfies adaptive soundness if for any algorithm \mathcal{P}^* the probability that experiment $\text{AdSnd}_{\mathcal{P}^*}^{\text{ZAP}}$ outputs 1 is negligible.

4.1 Construction

From a high-level point of view, the idea of our construction is as follows. In the first step, the user encrypts a message m together with *info* under the public key rp_k of the trusted third party and sends this encryption to the signer. The signer then concatenates the encryption with *info* and a unique session identifier *ssid*, signs the resulting string using an ordinary signature scheme and sends the

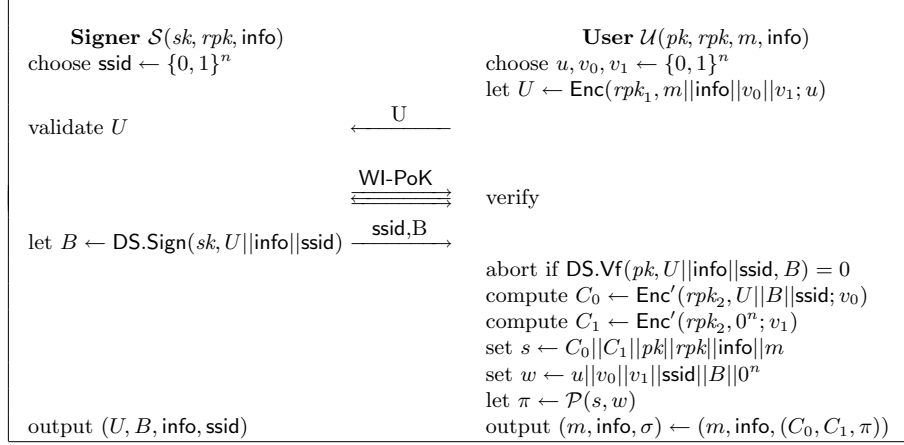


Fig. 3. Issue protocol of the fair partially blind signature scheme FPBS.

signature back to the user. Finally, the user encrypts (again under the public key of the trusted party) the first message as well as the signature and proves that the valid signature is properly encrypted. Note that we use two (not necessarily distinct) encryption schemes in order to handle the different input/output lengths.

Key Generation. $\text{Kg}(1^n)$ executes the key generation algorithm of the signature scheme $(sk, pk) \leftarrow \text{DS.Kg}(1^n)$. It chooses a suitable one-way function f , two elements $x_0, x_1 \leftarrow \{0, 1\}^n$ and sets $y_0 \leftarrow f(x_0)$ and $y_1 \leftarrow f(x_1)$. Kg also computes the verifier's first message $\rho \leftarrow \mathcal{V}(1^n)$ for the ZAP. The public key is $pk \leftarrow (pk, y_0, y_1, \rho)$ and the corresponding secret key is $sk \leftarrow (sk, x_0, x_1)$.

Revocation Key Generation. $\text{RevKg}(1^n)$ runs the key generation algorithm of both public-key encryption schemes $(rsk_1, rp_{k_1}) \leftarrow \text{EncKg}(1^n)$, $(rsk_2, rp_{k_2}) \leftarrow \text{EncKg}'(1^n)$ and outputs the secret revocation key $rsk \leftarrow (rsk_1, rsk_2)$ and the corresponding public key $rp_k \leftarrow (rp_{k_1}, rp_{k_2})$.

Signature Issue Protocol. The signature issue protocol is shown in Figure 3. Note that user and signer both include the same info in their computations, thus if one of the parties changes their mind about info during the execution then no valid signature can be produced.

- The user encrypts the message m together with the info element and the randomness for the second encryption under the public key of the trusted party, and sends U to the signer.
- The signer checks that the components U is of the correct form, namely a block-wise encryption of four fields of length n (each). Such an encryption scheme can be realized by concatenating IND-CPA secure encryptions of the individual fields. This check is essential. Otherwise, a malicious user

can easily break the scheme.¹ It then initiates an interactive witness-indistinguishable proof of knowledge, where the signer proves knowledge of either $f^{-1}(y_0)$ or $f^{-1}(y_1)$.²

- The signer sends its signature B and the randomly chosen session identifier ssid .
- \mathcal{U} encrypts the signature B and computes a ZAP proving that the encryption *either* contains a valid message-signature pair *or* a preimage of y_0 or y_1 . We now define the corresponding \mathcal{NP} -relation. Recall that L is an \mathcal{NP} -language and that a ZAP is witness-indistinguishable. Thus, we can include a second witness (the “or ...” part below), which allows full simulation during the security reduction. To compute the final proof, the user computes the ZAP π with respect to the first message ρ (stored in the public key) for $s \leftarrow C_0 \| C_1 \| pk \| rp_k \| \text{info} \| m \in L$ with witness $w \leftarrow u \| v_0 \| v_1 \| \text{ssid} \| B \| x$ such that membership in L is established by:

$$U \leftarrow \text{Enc}(rp_{k_1}, m \| \text{info} \| v_0 \| v_1; u) \wedge C_0 = \text{Enc}'(rp_{k_2}, U \| B \| \text{ssid}; v_0) \\ \wedge \text{DS.Vf}(pk, U \| \text{info} \| \text{ssid}, B) = 1$$

$$\text{or } C_1 = \text{Enc}'(rp_{k_2}, x; v_1) \wedge f(x) \in \{y_0, y_1\}.$$

Signature Verification. $\text{Vf}(pk, rp_k, m, \text{info}, \sigma)$ parses the signature σ as (C_0, C_1, π) . It returns the result of $\mathcal{V}(s, \pi)$ for $s \leftarrow C_0 \| C_1 \| pk \| rp_k \| \text{info} \| m$.

Signature Revocation. $\text{SigRev}(rsk, \text{view})$ parses the view view as $(U, B, \text{info}, \text{ssid})$. It decrypts U , computing $m \| \text{info} \| v_0 \| v_1 \leftarrow \text{Dec}(rsk_1, U)$ and extracts the randomness v_0 . SigRev then encrypts $U \| B \| \text{ssid}$ using the randomness v_0 , obtaining $C'_0 \leftarrow \text{Enc}'(rp_{k_2}, U \| B \| \text{ssid}; v_0)$ and outputs $\text{id}_{\text{Sig}} \leftarrow C'_0$.

Session Revocation. $\text{SesRev}(rsk, m, \text{info}, \sigma)$ takes as input a message-signature tuple $(m, \text{info}, (C_0, C_1, \pi))$ and the secret revocation key $rsk = (rsk_1, rsk_2)$. It extracts B' , computing $U \| B' \| \text{ssid} \leftarrow \text{Dec}(rsk_2, C_0)$ and returns $\text{id}_{\text{Ses}} \leftarrow (B', \text{ssid})$.

Signature Tracing. $\text{SigVf}(\text{id}_{\text{Sig}}, \sigma)$ parses id_{Sig} as C'_0 and σ as (C_0, C_1, π) . It outputs 1 iff $C'_0 = C_0$.

Session Tracing. $\text{SesVf}(\text{id}_{\text{Ses}}, \text{view})$ parses id_{Ses} as (B', ssid') and view as $(U, B, \text{info}, \text{ssid})$. It returns 1 iff $B' = B$ and $\text{ssid}' = \text{ssid}$.

Theorem 1 (Partial Blindness). *Let DS be a strongly unforgeable signature scheme, $(\text{EncKg}, \text{Enc}, \text{Dec})$ and $(\text{EncKg}', \text{Enc}', \text{Dec}')$ be two IND-CPA secure encryption schemes, and $(\mathcal{P}, \mathcal{V})$ be a ZAP. Then FPBS is partially blind.*

Proof. We prove partial blindness similar to [11,16]. That is, we transform the way the signatures are computed such that both signatures are, in the end,

¹ The malicious user \mathcal{U}^* in the unforgeability experiment executes the signer on a random message $m \in \{0, 1\}^n$ and an arbitrary $\text{info} \in \{0, 1\}^{n/2}$. It selects another $\text{info}' \in \{0, 1\}^{n/2}$, computes U according to the protocol and sends $U' \leftarrow U \| \text{info}'$ to the signer. Following the protocol, \mathcal{U}^* receives the signature B , computes $\sigma \leftarrow (C, \pi)$ and returns the tuple $(m, \text{info}' \| \text{info}, \sigma)$. This tuple is a valid forgery because \mathcal{U}^* never queried \mathcal{S} on $\text{info}'' \leftarrow \text{info}' \| \text{info}$.

² This proof allows the simulator in the blindness experiment to extract the witness and to use it instead of the signature.

completely independent of the bit b . We then argue that the success probability, while transforming the experiment, does not change the success probability of the adversary more than in a negligible amount.

In the first step, we modify the experiment of fair partially blind signature scheme in the following way: During signature issue, the user algorithm \mathcal{U}_0 executes the extraction algorithm **Extract** of the witness-indistinguishable proof of knowledge (WI-PoK) and computes the signature using the extracted witness. In case the proof is valid, but extraction fails, the experiment outputs a random bit. At the end of the issue protocol, the user computes the signature using the extracted witness (all other algorithms remain unchanged). More precisely:

Signature Issue Protocol (FPBS'). The issuing protocol is as before, except for the user \mathcal{U}_0 . It runs the extraction algorithm $x' \leftarrow \text{Extract}$ of the proof of knowledge. It aborts if $f(x') \notin \{y_0, y_1\}$. Otherwise, \mathcal{U}_0 executes \mathcal{P} on input $(s, w) \leftarrow (C_0 \| C_1 \| pk \| rpk \| \text{info} \| m, 0^{4n} \| 0^{l(n)} \| x')$, where $l(n)$ is the length of a signature. It outputs $\sigma \leftarrow (C_0, C_1, \pi)$. \mathcal{U}_1 remains unchanged.

It follows easily from the witness-indistinguishability property that both, experiment $\text{Blind}_{\mathcal{S}^*}^{\text{FPBS}}$ and $\text{Blind}_{\mathcal{S}^*}^{\text{FPBS}'}$, output 1 with the same probability (except for a negligible deviation).

In the next step, we modify the signature issue protocol of FPBS' such that the user algorithm sends an encryption to all-zero strings in the first round and computes the proof with the previously extracted witness again:

Signature Issue Protocol (FPBS''). The signature issue protocol remains the same, except for the user algorithm that computes $U \leftarrow \text{Enc}(rpk_1, 0^{4n}; u)$, and $C_1 \leftarrow \text{Enc}'(rpk_2, x'; v_1)$.

We denote the modified scheme with FPBS'' . The IND-CPA property of the encryption scheme guarantees that this modification goes unnoticed (negligible change in \mathcal{S}^* success probability).

In the last step, the user algorithm encrypts only all-zero strings in C_0 and the previously extracted witness x in C_1 .

Signature Issue Protocol (FPBS'''). The user \mathcal{U} calculates $C_0 \leftarrow \text{Enc}'(rpk_2, 0^{4c(n)} \| 0^{l(n)} \| 0^n; v_0)$ and $C_1 \leftarrow \text{Enc}'(rpk_2, x'; v_1)$.

Denoting the scheme with FPBS''' , we argue by the IND-CPA property that the success probability of \mathcal{S}^* stays essentially the same (except for a negligible deviation). In this protocol, the signature $\sigma = (C_0, C_1, \pi)$ is completely independent of U and B . We conclude that the malicious signer \mathcal{S}^* (against FPBS) cannot predict the bit b with noticeable advantage over $1/2$. \square

Theorem 2 (Signature Traceability). *Let DS be a strongly unforgeable signature scheme, $(\text{EncKg}, \text{Enc}, \text{Dec})$ and $(\text{EncKg}', \text{Enc}', \text{Dec}')$ be two IND-CPA secure encryption schemes, and $(\mathcal{P}, \mathcal{V})$ be a ZAP. Then FPBS is signature traceable.*

Proof. First of all recall that an adversary \mathcal{A} has three possibilities to win the signature traceability experiment. The first is to compute a message-signature tuple (m, info, σ) such that there exists no matching view. Note that since only views with respect to info are considered, the second possibility is to output a message-signature tuple corresponding to an execution with a different information element info' . The third way the attacker \mathcal{A} may win the game is to return two message-signatures related to the same view. In the following proof, we distinguish these three cases:

- An adversary is called a type-1 attacker, denoted by \mathcal{A}_1 , if it manages to output a valid message-signature tuple (m, info, σ) , where $\sigma = (C, \pi)$, such that $\text{SigVf}(\text{id}_{\text{Sig}}, \text{view}) = 0$ for all $\text{id}_{\text{Sig}} \leftarrow \text{SigRev}(rsk, \text{view})$ with $\text{view} \in V_*$.
- An algorithm is called a type-2 attacker, denoted by \mathcal{A}_2 , if it outputs a valid message-signature tuple (m, info, σ) such that there exists a $\text{view} \in V_* \setminus V_{\text{info}}$ with $\text{SigVf}(\text{id}_{\text{Sig}}, \sigma) = 1$ for $\text{id}_{\text{Sig}} \leftarrow \text{SigRev}(rsk, m, \text{view})$.
- An attacker is called a type-3 attacker, denoted by \mathcal{A}_3 , if it returns two valid message-signature tuple $(m_1, \text{info}_1, \sigma_1)$ and $(m_2, \text{info}_2, \sigma_2)$ with $m_1 \neq m_2$ such that $\text{SigVf}(\text{id}_{\text{Sig}}, \sigma_1) = \text{SigVf}(\text{id}_{\text{Sig}}, \sigma_2) = 1$ for $\text{id}_{\text{Sig}} \leftarrow \text{SigRev}(rsk, \text{view})$ and some $\text{view} \in V_*$.

Before proving the theorem, recall that the ZAP π only guarantees that *either* a valid signature *or* a value x , with $f(x) \in \{y_0, y_1\}$, has been encrypted. Thus, we have to distinguish these two cases throughout all proofs. In the first part of the proof, we show how to invert the one-way function f if a preimage x of y_0 or y_1 is encrypted. We stress that we show the reduction only once because it is the same in all proofs.

TYPE-1 ATTACKER. For the first type of adversaries, we have to distinguish, again, two cases: a) the encryption C contains a value x such that $f(x) \in \{y_0, y_1\}$ and b) C contains $U||B||\text{ssid}$. We begin the proof with the type-1a adversaries, denoted with \mathcal{A}_{1a} , showing how to invert the one-way function f if x is encrypted by applying the technique of Feige and Shamir [12]. We cover the type-1b adversaries, denoted with \mathcal{A}_{1b} , showing how to forge the underlying signature scheme if $U||B||\text{ssid}$ is encrypted.

Let \mathcal{A}_{1a} be an efficient algorithm, which succeeds in the signature traceability experiment with noticeable probability. We show how to build an algorithm \mathcal{B}_{1a} that efficiently inverts the one-way function f . Note that \mathcal{A}_{1a} may either encrypt the preimage of y_0 or the preimage of y_1 . Thus, we describe the algorithm that extracts the preimage of y_0 and argue, in the analysis, that the same algorithm can be used to extract the preimage of y_1 by “switching” y_0 and y_1 .

Setup. Algorithm \mathcal{B}_{1a} gets as input an image y of a one-way function f . It selects a value $x_1 \in \{0, 1\}^n$ at random, sets $y_1 \leftarrow f(x_1)$, generates a key pair for the signature scheme $(sk, pk) \leftarrow \text{DS.Kg}(1^n)$, as well as a key pair for the simulation of the trusted party $(rsk, rpk) \leftarrow \text{RevKg}(1^n)$, and computes the first message of the ZAP $\rho \leftarrow \mathcal{V}(1^n)$. \mathcal{B}_{1a} runs \mathcal{A}_{1a} on input $((pk, y, y_1, \rho), rsk, rpk)$ in a black-box simulation.

Signing Queries. If \mathcal{A}_{1a} initiates the signing protocol with its signing oracle then \mathcal{B}_{1a} behaves in the following way. It receives the first message U and info from \mathcal{A}_{1a} . It first checks that U has four fields, each of length $c(n)$. If so, it executes the witness-indistinguishable protocol (using the witness x_1) and, after terminating the protocol, \mathcal{B}_{1a} randomly selects a session identifier ssid . It returns ssid and $B \leftarrow \text{DS.Sign}(sk, U || \text{info} || \text{ssid})$.

Output. Finally, \mathcal{A}_{1a} stops, outputting a tuple (m, info, σ) with $\sigma = (C, \pi)$. Algorithm \mathcal{B}_{1a} computes $x' \leftarrow \text{Dec}(rsk, C)$ and outputs x' .

For the analysis, it is assumed that \mathcal{A}_{1a} outputs a valid message-signature tuple, either containing a witness for $f^{-1}(y_0)$ or for $f^{-1}(y_1)$, with noticeable probability $\epsilon(n)$. With $\epsilon_b(n)$, we denote the probability that \mathcal{A}_{1a} encrypts a witness x for $f^{-1}(y_b)$. Obviously, either $\epsilon_0(n) \geq \epsilon(n)/2$ or $\epsilon_1(n) \geq \epsilon(n)/2$ and, according to the description of \mathcal{B}_{1a} , we assume w.l.o.g. that the former holds. But if $\epsilon_0(n)$ is not negligible then we already have a contradiction because \mathcal{B}_{1a} outputs a preimage $x' = f^{-1}(y_0)$. Thus, $\epsilon_0(n)$ must be negligible and $\epsilon_1(n)$ has to be noticeable.

In this case, however, we can execute the same algorithm with switched public keys, i.e., we run \mathcal{A}_{1a} on input y_1, y instead of y, y_1 . Thanks to the witness-indistinguishability of the proof system, one can show (cf. [10]) that switching y and y_1 changes the success probability of \mathcal{A}_{1a} only in a negligible amount. Therefore, $|\epsilon_0(n) - \epsilon_1(n)|$ is negligible.

Next, we show that the second type of adversary (\mathcal{A}_{1b}) directly contradicts the unforgeability of the underlying signature scheme. We build an algorithm \mathcal{B}_{1b} , which uses \mathcal{A}_{1b} in a black-box simulation.

Setup. \mathcal{B}_{1b} takes as input a public verification key pk and has access to a signing oracle. It executes the key generation algorithm of the encryption scheme $(rsk, rp_k) \leftarrow \text{EncKg}(1^n)$, selects two elements x_0, x_1 at random and sets $y_0 \leftarrow f(x_0), y_1 \leftarrow f(x_1)$. The algorithm then computes the first round of the ZAP $\rho \leftarrow \mathcal{V}(1^n)$ and executes \mathcal{A}_{1b} on input $((pk, y_0, y_1, \rho), rsk, rp_k)$.

Signing Queries. Whenever \mathcal{A}_{1b} invokes the interactive signing protocol on some message U and some info element info , algorithm \mathcal{B}_{1b} behaves as follows. It first checks that U comprises four fields of length $c(n)$. If the assertion holds, it follows the witness-indistinguishable protocol using the witness x_1 . Afterwards, \mathcal{B}_{1b} selects a session identifier ssid at random and invokes its signing oracle on $m \leftarrow U || \text{info} || \text{ssid}$, receiving the signature σ . Algorithm \mathcal{B}_{1b} sends $\text{ssid}, B \leftarrow \sigma$ back to \mathcal{A}_{1b} .

Output. Finally, \mathcal{A}_{1b} stops, outputting a possibly valid tuple $(m', \text{info}', \sigma')$ with $\sigma' = (C', \pi')$. \mathcal{B}_{1b} decrypts C' obtaining $U' || B' || \text{ssid}'$. It outputs $(m^*, \sigma^*) \leftarrow (U' || \text{info}' || \text{ssid}', B')$.

For the analysis, first note that \mathcal{B}_{1b} is efficient since \mathcal{A}_{1b} is and that \mathcal{B}_{1b} performs a perfect simulation from \mathcal{A}_{1b} 's point of view. We assume that \mathcal{A}_{1b} is successful and that C' does *not* contain the encryption of a witness. Since there is no matching view $\text{view} = (U, B, \text{info}, \text{ssid})$, it follows that either \mathcal{B}_{1b} never queried m^* to its oracle or never received the same signature B for the message m^* . Thus, \mathcal{B}_{1b} succeeds whenever \mathcal{A}_{1b} does.

TYPE-2 ATTACKER. The second class of attackers (\mathcal{A}_2) manages to output a message-signature tuple (m, info, σ) with $\sigma = (C, \pi)$ such that a “foreign” matching view exists, i.e., $\text{SigVf}(\text{id}_{\text{sig}}, \sigma) = 1$ for $\text{id}_{\text{sig}} \leftarrow \text{SigRev}(rsk, \text{view})$ with $\text{view} \in V_* \setminus V_{\text{info}}$. We show that if a matching view exists, then the message, the information element, and the session id have to be the same.

Let the matching view be $\text{view} = (U', B', \text{info}', \text{ssid}')$ and let $U||B||\text{ssid} \leftarrow \text{Dec}(rsk, C)$ be the decryption of the signature received from the malicious user. Since the verification algorithm evaluates to 1, and because x such that $f(x) \in \{y_0, y_1\}$ is *not* contained in C , it follows that $C' = C$ (otherwise $\text{SigVf}(\text{id}_{\text{sig}}, \sigma) = 0$). This, however, implies that $U = U'$, $\text{ssid} = \text{ssid}'$, and $B = B'$. Thus $m' = m$, $\text{info}' = \text{info}$, and $v' = v$ where $m'||\text{info}'||v' \leftarrow \text{Dec}(rsk, U')$ is the decryption of U' contained in view . Thus, such an attacker does not exist.

TYPE-3 ATTACKER. Let \mathcal{A}_3 be an adversary, which outputs two valid message-signature tuples $(m_1, \text{info}_1, \sigma_1)$ and $(m_2, \text{info}_2, \sigma_2)$ with $\sigma_i = (C_i, \pi_i)$, for $i = 1, 2$, such that $(m_1, \text{info}_1) \neq (m_2, \text{info}_2)$. We first show that the tuples $(m_1, \text{info}_1, \sigma_1)$ and $(m_2, \text{info}_2, \sigma_2)$ cannot stem from the same view.

W.l.o.g., suppose that $\text{view} = (U, B, \text{info}, \text{ssid}) \in V_*$ is the corresponding view to the message-signature tuple $(m_1, \text{view}_1, \sigma_1)$. Recall that the signature revocation algorithm SigRev reveals the randomness v from view , computing $m||\text{info}||v \leftarrow \text{Dec}(rsk, U)$, and generates $C' \leftarrow \text{Enc}(rpk, U||B||\text{ssid}; v)$. Since view is also a matching view for the first message-signature tuple, we infer that $C' = C_1$ and consequently $C' = C_2$. Otherwise, the verification algorithm would not return 1 on both inputs. Since $\sigma_2 = (C_2, \pi_2)$ is a valid signature, the prove π_2 is valid and it follows that C_2 is the ciphertext for $U_2||B_2||\text{ssid}_2$. But this string equals $U_1||B_1||\text{ssid}_1$ and therefore $m_1 = m_2$, $\text{info}_1 = \text{info}_2$, and $v_1 = v_2$. This, however, contradicts the assumption that $(m_1, \text{info}_1) \neq (m_2, \text{info}_2)$ and therefore such a pair cannot exist. \square

Theorem 3 (Session Traceability). *Let DS be a strongly unforgeable signature scheme, $(\text{EncKg}, \text{Enc}, \text{Dec})$ and $(\text{EncKg}', \text{Enc}', \text{Dec}')$ be two IND-CPA secure encryption schemes and $(\mathcal{P}, \mathcal{V})$ be a ZAP. Then FPBS is session traceable.*

Proof. We first show that if a signature verifies then it is always possible to disclose the corresponding session and second, that each signature has a unique identifier and therefore a unique session. Analogously to the proof of signature traceability, we distinguish three cases:

- An algorithm \mathcal{A}_1 , which outputs a valid message-signature tuple (m, info, σ) , where $\sigma = (C, \pi)$, such that $\text{SesVf}(\text{id}_{\text{ses}}, \text{view}) = 0$ for all $\text{view} \in V_*$ with $\text{id}_{\text{ses}} \leftarrow \text{SesRev}(rsk, m, \text{info}, \sigma)$.
- An attacker \mathcal{A}_2 , which returns a valid message-signature tuple (m, info, σ) , such that there exists a $\text{view} \in V_* \setminus V_{\text{info}}$ with $\text{SesVf}(\text{id}_{\text{ses}}, \sigma) = 1$ for $\text{id}_{\text{ses}} \leftarrow \text{SesRev}(rsk, m, \text{info}, \sigma)$.
- An adversary \mathcal{A}_3 , which manages to output a valid message-signature tuple (m, info, σ) , such that there exist two distinct views $\text{view}_1, \text{view}_2 \in V_*$ with $\text{SesVf}(\text{id}_{\text{ses}}, \text{view}_1) = \text{SesVf}(\text{id}_{\text{ses}}, \text{view}_2) = 1$ for $\text{id}_{\text{ses}} \leftarrow \text{SesRev}(rsk, m, \text{info}, \sigma)$.

The reductions for the first two cases are identical to the first two reductions in the proof of Theorem 2. Thus, we omit it here.

What is left to show is that two different views cannot match to a single execution. This follows from the fact that the signer generates a new session identifier ssid during each execution. Thus, with overwhelming probability over the choice of ssid , each execution yields a new view. \square

By combining the above results with an observation in Section 2, we get the following corollary.

Corollary 1 (Unforgeability). *Let DS be a strongly unforgeable signature scheme, $(\text{EncKg}, \text{Enc}, \text{Dec})$ be an IND-CPA secure encryption scheme with message expansion function c , and $(\mathcal{P}, \mathcal{V})$ be a ZAP. Then FPBS is unforgeable.*

Acknowledgments

The authors thank the anonymous reviewers of Africacrypt 2010 for their valuable comments. They are indebted to one particular reviewer who provided exceptionally detailed comments and helped in improving the presentation as well as eliminating inconsistencies. The MD5 hash of the first sentence in his or her review was `3641f58bff4934c06d9e71afcc3e14fe`.

References

1. Masayuki Abe and Eiichiro Fujisaki. *How to Date Blind Signatures*. Advances in Cryptology — Asiacrypt'96, Volume 1163 of Lecture Notes in Computer Science, pages 244–251. Springer-Verlag, 1996.
2. Masayuki Abe and Tatsuaki Okamoto. *Provably Secure Partially Blind Signatures*. Advances in Cryptology — Crypto 2000, Lecture Notes in Computer Science, pages 271–286. Springer-Verlag, 2000.
3. Masayuki Abe and Miyako Ohkubo. *Provably Secure Fair Blind Signatures with Tight Revocation*. Advances in Cryptology — Asiacrypt 2001, Lecture Notes in Computer Science, pages 583–602. Springer-Verlag, 2001.
4. Mihir Bellare and Oded Goldreich. *On Defining Proofs of Knowledge*. Advances in Cryptology — Crypto'92, Lecture Notes in Computer Science, pages 390–420. Springer-Verlag, 1992.
5. Mihir Bellare, Haixia Shi, and Chong Zhang. *Foundations of Group Signatures: The Case of Dynamic Groups*. Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA) 2005, Lecture Notes in Computer Science, pages 136–153. Springer-Verlag, 2005.
6. Ran Canetti, Oded Goldreich, and Shai Halevi. *The random oracle methodology, revisited*. *J. ACM*, 51(4):557–594, 2004.
7. David Chaum. *Blind Signatures for Untraceable Payments*. Advances in Cryptology — Crypto'82, pages 199–203. Plenum, New York, 1983.
8. Sherman S.M. Chow, Lucas C.K. Hui, S.M. Yiu, and K.P. Chow. *Two Improved Partially Blind Signature Schemes from Bilinear Pairings*. Cryptology ePrint Archive, Report 2004/108, 2004. <http://eprint.iacr.org/>.

9. Cynthia Dwork and Moni Naor. *Zaps and Their Applications*. *SIAM Journal on Computing*, 36(6):1513–1543, 2007.
10. Uriel Feige. *Alternative Models for Zero-Knowledge Interactive Proofs*. PhD Thesis. Weizmann Institute of Science. Dept. of Computer Science and Applied Mathematics, 1990. <http://www.wisdom.weizmann.ac.il/~feige>.
11. Marc Fischlin. *Round-Optimal Composable Blind Signatures in the Common Reference String Model*. Advances in Cryptology — Crypto'06, Volume 4117 of Lecture Notes in Computer Science, pages 60–77. Springer-Verlag, 2006.
12. Uriel Feige and Adi Shamir. *Zero Knowledge Proofs of Knowledge in two Rounds*. Advances in Cryptology — Crypto'89, Lecture Notes in Computer Science, pages 526–544. Springer-Verlag, 1989.
13. Marc Fischlin and Dominique Schröder. *Security of Blind Signatures Under Aborts*. Public-Key Cryptography (PKC) 2009, Volume 5443 of Lecture Notes in Computer Science, pages 297–316. Springer-Verlag, 2009.
14. Yair Frankel, Yiannis Tsiounis, and Moti Yung. *Indirect Discourse Proof: Achieving Efficient Fair Off-Line E-cash*. Advances in Cryptology — Asiacrypt'96, Lecture Notes in Computer Science, pages 286–300. Springer-Verlag, 1996.
15. G. Fuchsbauer and D. Vergnaud. *Fair Blind Signatures without Random Oracles*. To appear at The 3rd International Conference on Cryptology in Africa (AFRICACRYPT '10).
16. Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. *Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions*. Theory of Cryptography Conference (TCC)'07, Volume 4392 of Lecture Notes in Computer Science, pages 323–341. Springer-Verlag, 2007.
17. Emeline Hufschmitt and Jacques Traoré. *Fair Blind Signatures Revisited*. Pairing-Based Cryptography - Pairing 2007, Volume 4575 of Lecture Notes in Computer Science, pages 268–292. Springer, 2007.
18. Ari Juels, Michael Luby, and Rafail Ostrovsky. *Security of Blind Digital Signatures*. Advances in Cryptology — Crypto'97, Volume 1294 of Lecture Notes in Computer Science, pages 150–164. Springer-Verlag, 1997.
19. Markus Jakobsson and Moti Yung. *Distributed "Magic Ink" Signatures*. Advances in Cryptology — Eurocrypt'97, Lecture Notes in Computer Science, pages 119–135. Springer-Verlag, 1997.
20. Hyung-Woo Lee and Tai-Yun Kim. *Message Recovery Fair Blind Signature*. Public-Key Cryptography (PKC)'99, Lecture Notes in Computer Science, pages 97–111. Springer-Verlag, 1999.
21. Shingo Miyazaki and Kouichi Sakurai. *A More Efficient Untraceable E-Cash System with Partially Blind Signatures Based on the Discrete Logarithm Problem*. Financial Cryptography (FC)'98, Lecture Notes in Computer Science, pages 296–308. Springer-Verlag, 1998.
22. Tatsuaki Okamoto. *Efficient Blind and Partially Blind Signatures Without Random Oracles*. Theory of Cryptography Conference (TCC)'06, Volume 3876 of Lecture Notes in Computer Science, pages 80–99. Springer-Verlag, 2006.
23. David Pointcheval and Jacques Stern. *Security Arguments for Digital Signatures and Blind Signatures*. *Journal of Cryptology*, 13(3):361–396, 2000.
24. Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch. *Fair Blind Signatures*. Advances in Cryptology — Eurocrypt'95, Lecture Notes in Computer Science, pages 209–219. Springer-Verlag, 1995.
25. Sebastiaan H. von Solms and David Naccache. *On blind signatures and perfect crimes*. *Computers & Security*, 11(6):581–583, 1992.