Blind Signatures from Knowledge Assumptions

Wojciech Wodo, Lucjan Hanzlik

Faculty of Fundamental Problems of Technology,
Wrocław University of Technology
{firstname.secondname}@pwr.edu.pl

Abstract. This paper concerns blind signature schemes. We focus on two moves constructions, which imply concurrent security. There are known efficient blind signature schemes based on the random oracle model and on the common reference string model. However, constructing two move blind signatures in the standard model is a challenging task, as shown by the impossibility results of Fischlin et al. The recent construction by Garg et al. (Eurocrypt'14) bypasses this result by using complexity leveraging, but it is impractical due to the signature size ($\approx$ 100 kB). Fuchsbauer et al. (Crypto'15) presented a more practical construction, but with a security argument based on interactive assumptions. We present a blind signature scheme that is two-move, setup-free and comparable in terms of efficiency with the results of Fuchsbauer et al. Its security is based on a knowledge assumption called knowledge-of-exponent assumption.

Keywords: blind signature, short randomizable signatures, knowledge assumption

1 Introduction

The idea of blind signatures was first introduced by David Chaum in his work [5]. He also gave the first application for this primitive, namely e-cash. The idea was to protect privacy of user's in such a way that the bank is not able to trace the usage of a signed banknote. In particular, this means that the signer should not be able to link a signature to the issuing protocol (*blindness*). Of course, we also require *unforgeability*, i.e. without the knowledge of the secret key, one cannot compute a valid signature. From this point on, blind signatures were the topic of many research papers. With time new applications such as e-voting and one-show anonymous credentials were developed.

Efficiency is one of the main topics in the research on blind signatures. This not only concerns the computational complexity, public key and signature size but also the communication complexity and the number of moves a user and signer must perform during the issuance procedure. Two-move blind signatures (also called *round-optimal* [7]) are of particular interest as they directly yield concurrent security.

There exist efficient and round-optimal blind signatures with security in the random oracle model [3,6]. Ghadafi and Smart proposed a two-move blind signature scheme in the common reference string model, based on a new variant of the interactive LRSW assumption [13]. However, those solutions assume that the public key is generated honestly, i.e. they use a weaker definition of blindness, where the signing key pair is generated honestly and then given to the adversary.

Non-interactive zero-knowledge (NIZK) proofs in the CRS model were used by Fischlin to fill this gap [7]. His generic construction of blind signatures is round-optimal and blind in the malicious key model that allows the signer to generate the public key in a malicious way. This construction was successfully instantiated by Abe et al. [1] using structure-preserving signatures and Groth-Sahai proofs [14].

The CRS model allows to construct efficient blind signatures under standard assumptions without random oracles. However, such construction requires users to perform a setup phase to receive the CRS. This string has to be computed by a trusted third party in order to be useful and to ensure security. Moreover, in a real world, it is a good practice to update the parameters of a system in order to keep a reasonable and constant security level.

Thus, *setup-free* and round-optimal blind signatures without random oracles are desired. However, as shown by Fischlin et al. [8] it is impossible to construct a blind signature scheme which unforgeability property would have a black-box reduction to a non-interactive problem instance. This impossibility result requires that the scheme admits so called signature derivation checks, i.e. the transcript of communication allows to verify whether the user is able to derive a valid signature in this execution. This leaves room for constructions that bypass this limitations.

Garg et al. [12] were the first to propose a generic construction in the standard model. However, the solution is not efficient from a practical point of view. The user uses fully homomorphic encryption to encrypt the message, which the signer evaluates using a signing circuit. To get rid of the CRS the author's use two-round witness-indistinguishable proofs (ZAPs).

At Eurocrypt'14 Garg and Gupta [11] proposed the first efficient round-optimal blind signature constructions in the standard model. They used a two-CRS NIZK proof system based on GS proofs [14], where the common reference string is a part of the signers public key. The construction forces the signer to either honestly compute the CRS or to solve a subexponential DL instance. The reduction algorithm for the unforgeability proof is able to compute this DL instance and compute a malicious CRS, which is used to break the underlying standard assumption. This requires to use a technique called complexity leveraging. As a consequence, the computational and communication complexity limits the usage in many practical applications.

Recently, Fuchsbauer et al. [10] proposed the first practical round-optimal blind signature scheme in the standard model. They also present how to extend their construction to a partially blind signature scheme and a blind signature scheme on a vector of messages (which yield one-show anonymous credentials in the standard model). Their construction is based on structure-preserving signatures on equivalence classes (SPS-EQ), which allows to sign a representative of an equivalence class and such signature can be transformed (even without the secret signing key) to a signature of a different member of the equivalence class. Unforgeability follows from the unforgeability of the SPS-EQ scheme. On the other hand, in order to proof blindness, an interactive version of the well-known decisional Diffie-Hellman problem is required. One of the disadvantages of this generic construction is that it cannot be instantiated with all SPS-EQ. Admissible instantiations must provide a feature called perfect adaptation under malicious keys. The authors instantiate their construction with the SPS-EQ from [9], which security is based on an interactive assumption.

Our Contribution. The main contribution of this paper is a blind signature scheme based on the single-message protocol for the short randomizable signatures presented by Pointcheval and Sanders [16]. The protocol allows the user to receive a signature under a message committed in a Pedersen commitment. However, the protocol requires the user to proof knowledge of the opening. Therefore, common instantiation for this type of proofs requires either the random oracle model, multiple rounds or the common reference string model.

In our construction we get rid this proof using the knowledge-of-exponent assumption [4]. Moreover, we use a deterministic parameter generator. Thus, we can 'trust' the group parameters in the signer's public key. The resulting blind signature scheme is not only two-move but works in the plain model, i.e. without random oracle or a common-reference string. Additionally, we propose how to extend this construction to a partially blind signature scheme. Both schemes are blind in the weaker, honestly generated public key model.

2 Preliminaries

2.1 Notation and Bilinear Groups

By $y \leftarrow \mathcal{A}(x)$ we denote the execution of algorithm $\mathcal{A}$ outputting y, on input x. In addition, the superscript $\mathcal{O}$ in $\mathcal{A}^{\mathcal{O}}$ means that algorithm $\mathcal{A}$ has access to oracle $\mathcal{O}$. We say that $\mathcal{A}$ is probabilistic polynomial-time (PPT) if $\mathcal{A}$ uses internal random coins and the computation for any input $x \in \{0,1\}^*$ terminates in polynomial time. By $r \xleftarrow{\$} S$ we mean that r is chosen uniformly at random over the set S. Furthermore, we will use $1_{\mathbb{G}}$ to denote the identity element in group $\mathbb{G}$ and $[k]P$ to denote point multiplication, where:

$$[k]P = \underbrace{P + \ldots + P}_{k\text{- times}}$$

and point $P = (x, y)$ lies on some curve E.

Definition 1 (Negligible Function). *A function $\epsilon(\lambda) : \mathbb{N} \to \mathbb{R}$ is negligible, if for every positive polynomial $poly(.)$ there exists an integer $N > 0$ such that for all security parameters $\lambda > N$ we have:*

$$|\epsilon(\lambda)| < \frac{1}{poly(\lambda)}$$

Definition 2 (Bilinear map). *Let us consider cyclic groups* $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$, $(\mathbb{G}_T, \cdot)$ *of a prime order* q. *Let* P_1, P_2 *be generators of respectively* $\mathbb{G}_1$ *and* $\mathbb{G}_2$. *We call* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *a* bilinear map *(pairing) if it is efficiently computable and the following holds:*

Bilinearity: $\forall (S, T) \in \mathbb{G}_1 \times \mathbb{G}_2$, $\forall a, b \in \mathbb{Z}_q$, *we have* $e([a]S, [b]T) = e(S, T)^{a \cdot b}$,
Non-degeneracy: $e(P_1, P_2) \neq 1$ *is a generator of group* $\mathbb{G}_T$,

Depending on the choice of groups we say that map e is of:

Type 1: if $\mathbb{G}_1 = \mathbb{G}_2$,
Type 2: if $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct groups and there exists an efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$,
Type 3: if $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct groups and no efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is known.

Bilinear map groups are known to be instantiable with ordinary elliptic curves such as MNT curves [15] or curves introduced by Barreto and Naehrig [2] (in short BN-curves).

Definition 3 (Bilinear-group generator). *A bilinear-group generator is a polynomial-time algorithm* BGGen *that on input of a security parameter* λ *returns a bilinear group* $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ *such that* $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$ *and* $\mathbb{G}_T$ *are groups of order* q *with* $\log_2 q \approx \lambda$ *and* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *is a bilinear map. Similar to [10] we assume that* BGGen *is deterministic (which is the case for BN-curves [2]).*

2.2 Pedersen Commitments

In our constructions we will make use of Pedersen commitments that work with the bilinear group generator BGGen described above.

Definition 4 (Pedersen Commitment in $\mathbb{G}_1$). *Pedersen commitments consist of the following algorithms:*

$\mathsf{Setup_P}(\lambda)$**:**
Compute a bilinear group $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow \mathsf{BGGen}(\lambda)$, *choose* $z \xleftarrow{\$} \mathbb{Z}_q$, *compute* $Q_1 = [z]P_1$ *and output the commitment key* $\mathsf{cpp} = (\mathsf{BG}, Q_1)$ *(which is an implicit parameter to the rest algorithms).*

$\mathsf{Commit_P}(m, r)$**:**
On input a message $m \in \mathbb{Z}_q$ *and a randomness* $r \in \mathbb{Z}_q$, *output the commitment* $Co = [m]P_1 + [r]Q_1$ *and opening* $O = (m, r)$.

$\mathsf{Open_P}(Co, O)$**:**
On input a commitment $Co \in \mathbb{G}_1$ *and an opening* O, *if* $Co = [m]P_1 + [r]Q_1$ *output* m, *else output* $\perp$.

The above commitments are perfectly hiding and computationally binding under the DLP assumption in $\mathbb{G}_1$. Which is a classical result.

2.3 Short Randomizable Signatures

We now recall the short randomizable signatures presented by Pointcheval and Sanders [16]. This signature scheme uses type 3 pairing, thus we will use describe it using the bilinear group generator BGGen. Moreover, we present the modified variant of the scheme that admits the signing of committed messages.

Definition 5 (Short Randomizable Signatures). *The signature scheme is given by the following triple of algorithms given an output* BG *of* $\mathsf{BGGen}(\lambda)$.

$\mathsf{KeyGen}_{\mathsf{PS}}(\mathsf{BG})$**:**

Choose $(x, y) \xleftarrow{\$} (\mathbb{Z}_q^*)^2$*, compute* $X_1 = [x]P_1$*,* $X_2 = [x]P_2$*,* $Y_1 = [y]P_1$*,* $Y_2 = [y]P_2$*, set the private key* $\mathsf{sk}_{\mathsf{PS}} = X_1$ *and the public key* $\mathsf{pk}_{\mathsf{PS}} = (\mathsf{BG}, X_2, Y_1, Y_2)$.

$\mathsf{Sign}_{\mathsf{PS}}(m, \mathsf{sk}_{\mathsf{PS}})$**:**

Select $u \xleftarrow{\$} \mathbb{Z}_q$ *and compute* $\sigma_1 = [u]P_1$*,* $\sigma_2 = [u](X_1 + [m]Y_1)$*. Output* (σ_1, σ_2).

$\mathsf{Verify}_{\mathsf{PS}}(m, (\sigma_1, \sigma_2), \mathsf{pk}_{\mathsf{PS}})$**:**

Output 1 *if and only if* $\sigma_1 \neq 1_{\mathbb{G}_1}$ *and* $e(\sigma_1, X_2 + [m]Y_2) = e(\sigma_2, P_2)$.

Definition 6 (Randomization of Signatures). *For all tuples* $(\mathsf{pk}_{\mathsf{PS}}, m, (\sigma_1, \sigma_2))$*, where*

$$\mathsf{Verify}_{\mathsf{PS}}(m, (\sigma_1, \sigma_2), \mathsf{pk}_{\mathsf{PS}}) = 1 \quad \text{and} \quad m \in \mathbb{Z}_q,$$

we have that $([t]\sigma_1, [t]\sigma_2)$*, where* $t \in \mathbb{Z}_q^*$*, is a random element in the signature space, conditioned on* $\mathsf{Verify}_{\mathsf{PS}}(m, ([t]\sigma_1, [t]\sigma_2), \mathsf{pk}_{\mathsf{PS}}) = 1$.

Definition 7 (Assumption 1 [16]). *Given a security parameter* λ *and a bilinear group* $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ *returned by algorithm* BGGen*. For* $X_1 = [x]P_1$*,* $Y_1 = [y]P_1$*,* $X_2 = [x]P_2$*,* $Y_2 = [y]P_2$*, where* x *and* y *are random scalars in* $\mathbb{Z}_q$*, we define oracle* $\mathcal{O}(\cdot)$ *that, on input a value* $m \in \mathbb{Z}_q$*, outputs* $(h, [x + y \cdot m]h) \in \mathbb{G}_1^2$*, where* h *is a random element in* $\mathbb{G}_1$.

The Assumption 1 is said to hold for BG *if for all PPT adversaries* $\mathcal{A}$ *the following probability is negligible in the security parameter* λ*:*

$$\begin{aligned}\Pr[&\mathsf{BG} \leftarrow \mathsf{BGGen}(\lambda), x \xleftarrow{\$} \mathbb{Z}_q, y \xleftarrow{\$} \mathbb{Z}_q, Y_1 = [y]P_1, X_2 = [x]P_2, Y_2 = [y]P_2,\\ &(m, h, [x + y \cdot m]h) \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(\mathsf{BG}, X_1, Y_1, Y_2) :\\ &m \notin Q \ \wedge \ m \in \mathbb{Z}_q \backslash \{0\} \ \wedge \ h \in \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}],\end{aligned}$$

where Q *denotes the set of queries made by* $\mathcal{A}$ *to oracle* $\mathcal{O}(\cdot)$.

Theorem 1 ([16]). *The above Assumption 1 holds in the generic bilinear group model: after* $q_{\mathcal{O}}$ *oracle queries and* $q_{\mathbb{G}}$ *group-oracle queries, no adversary can generate a valid pair for a new scalar with probability greater than* $6 \cdot (q_{\mathcal{O}} + q_{\mathbb{G}})^2 / q$.

Proof. The proof is given in [16].

Theorem 2. *The above signature scheme is EUF-CMA secure under Assumption 1.*

Proof. The proof is given in [16].

```
**Experiment** $\mathsf{Exp}^{kea1}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g)$**:**
  $b \xleftarrow{\$} \mathbb{Z}_q$; $B \leftarrow g^b$
  $(C,Y) \leftarrow \mathbf{A}_n(q,g,B)$ ; $c \leftarrow \bar{\mathbf{A}}_n(q,g,B)$
  If $(Y = C^b$ AND $g^c \neq C)$ then return 1 else return 0
```

Fig. 1. Experiment $\mathsf{Exp}^{kea1}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g)$

```
**Experiment** $\mathsf{Exp}^{kea3}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A)$**:**
  $b \xleftarrow{\$} \mathbb{Z}_q$; $B \leftarrow g^b$ ; $X \leftarrow A^b$
  $(C,Y) \leftarrow \mathbf{A}_n(q,g,A,B,X)$ ; $c_1, c_2 \leftarrow \bar{\mathbf{A}}_n(q,g,A,B,X)$
  If $(Y = C^b$ AND $g^{c_1}A^{c_2} \neq C)$ then return 1 else return 0
```

Fig. 2. Experiment $\mathsf{Exp}^{kea3}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A)$

2.4 Knowledge of Exponent Assumption

We now recall two non-uniform definitions for knowledge of exponent assumptions [4]. Let $GL = \{(q,g) : q, 2q+1$ are primes and g is a generator of $G_q\}$ and let $GL_n = \{(q,g) \in GL : |2q+1| = n\}$. Of course, the definition can be adapted any type of groups. In particular, to the groups returned by the generator BGGen.

Definition 8. *KEA1 Let $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ and $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}_n\}_{n\in\mathbb{N}}$ be families of circuits, and $v : \mathbb{N} \to [0,1]$ a function. We associate to any $n \in \mathbb{N}$, any $(q,g) \in GL_n$, and any $A \in G_q$ the following experiment:*

We let

$$\mathbf{Adv}^{kea1}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g) = \Pr[\mathsf{Exp}^{kea1}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g) = 1]$$

denote the advantage of $\mathbf{A}$ relative to $\bar{\mathbf{A}}$ on inputs n, q, g. We say that $\bar{\mathbf{A}}$ is a kea1-extractor for $\mathbf{A}$ with error bound v if

$$\forall n \in \mathbb{N}\ \forall (q,g) \in GL_n : \mathbf{Adv}^{kea1}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g) \leq v(n).$$

We say that KEA1 holds if for every poly-size family of circuits $\mathbf{A}$ there exists a poly-size family of circuits $\bar{\mathbf{A}}$ and a negligible function v such that $\bar{\mathbf{A}}$ is a kea1-extractor for $\mathbf{A}$ with error bound v.

Definition 9. *KEA3 according to [4] Let $\mathbf{A} = \{\mathbf{A}_n\}_{n\in\mathbb{N}}$ and $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}_n\}_{n\in\mathbb{N}}$ be families of circuits, and $v : \mathbb{N} \to [0,1]$ a function. We associate to any $n \in \mathbb{N}$, any $(q,g) \in GL_n$, and any $A \in G_q$ the following experiment:*

We let

$$\mathbf{Adv}^{kea3}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g) = \Pr[\mathsf{Exp}^{kea3}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A) = 1]$$

denote the advantage of $\mathbf{A}$ relative to $\bar{\mathbf{A}}$ on inputs n, q, g, A. We say that $\bar{\mathbf{A}}$ is a kea1-extractor for $\mathbf{A}$ with error bound v if

$$\forall n \in \mathbb{N}\ \forall (q,g) \in GL_n : \mathbf{Adv}^{kea3}_{\mathbf{A},\bar{\mathbf{A}}}(n,q,g,A) \leq v(n).$$

We say that KEA3 holds if for every poly-size family of circuits $\mathbf{A}$ *there exists a poly-size family of circuits* $\bar{\mathbf{A}}$ *and a negligible function* v *such that* $\bar{\mathbf{A}}$ *is a kea3-extractor for* $\mathbf{A}$ *with error bound* v.

Remark 1. Informally, if we assume that the KEA3 assumption holds for the generator BGGen. Then if there exists an algorithm that on input receives $(g, \hat{g} = [k]g, h, \hat{h} = [k]h)$ and outputs $c = [m]g + [r]h$ and $c' = [m]\hat{g} + [r]\hat{h}$ then there must exists an extraction algorithm that outputs m, r, receiving the same input.

3 (Partially) Blind Signatures

In this section we recall the syntax and security of blind signature and partially blind signature schemes.

3.1 Blind Signature Scheme

Definition 10. *A blind signature scheme consists of the following PPT algorithms* $\mathsf{BS} = (\mathsf{KeyGen}_{\mathsf{BS}}, \mathcal{U}_{\mathsf{BS}}, \mathcal{S}_{\mathsf{BS}}, \mathsf{Verify}_{\mathsf{BS}})$ *defined as follows:*

$\mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$**:** *on input a security parameter, this algorithm outputs a pair of public/secret key* $(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}})$ *of the signer.*

$\langle \mathcal{U}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}}), \mathcal{S}_{\mathsf{BS}}(\mathsf{sk}_{\mathsf{BS}}) \rangle$**:** *are executed by a user and a signer. On input the signer's secret key* $\mathsf{sk}_{\mathsf{BS}}$ *algorithm* $\mathcal{S}_{\mathsf{BS}}$ *interacts with algorithm* $\mathcal{U}_{\mathsf{BS}}$. *On input a message* m, *from message space* $\mathcal{M}$, *and the signer public key* $\mathsf{pk}_{\mathsf{BS}}$, *algorithm* $\mathcal{U}_{\mathsf{BS}}$ *outputs a signature* σ *on* m, *or* $\perp$, *if the interaction was not successful.*

$\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}})$**:** *on input a message* m, *signature* σ *and the signer's public key* $\mathsf{pk}_{\mathsf{BS}}$, *this algorithm outputs* 1, *if* σ *is a valid signature and* 0 *otherwise.*

A blind signature scheme BS must satisfy correctness, unforgeability and blindness as defined below.

Correctness. A blind signature scheme BS is *correct*, if for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$, all messages $m \in \mathcal{M}$ and $\sigma \leftarrow \langle \mathcal{U}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}}), \mathcal{S}_{\mathsf{BS}}(\mathsf{sk}_{\mathsf{BS}}) \rangle$ it holds that $\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}}) = 1$.

Unforgeability. A blind signature scheme BS is *unforgeable*, if for all PPT algorithms $\mathcal{A}$ having access to a signer oracle, we have:

$$\begin{aligned}\Pr\Big[&(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda), (m_i^*, \sigma_i^*)_{i=1}^{k+1} \leftarrow \mathcal{A}(\mathsf{pk}_{\mathsf{BS}})^{\langle \cdot\ , \mathcal{S}_{\mathsf{BS}}(\mathsf{sk}_{\mathsf{BS}}) \rangle} : \\ &m_i^* \neq m_j^* \quad \text{for } i, j \in \{1, \ldots, k+1\}, i \neq j \qquad \text{and} \\ &\mathsf{Verify}_{\mathsf{BS}}(m_i^*, \sigma_i^*, \mathsf{pk}_{\mathsf{BS}}) = 1 \quad \text{for } i \in \{1, \ldots, k+1\}\Big] \leq \epsilon(\lambda),\end{aligned}$$

where k is the number of oracle queries.

Blindness. A blind signature scheme BS is *blind* in the *honest-signer* model, if for all PPT algorithms $\mathcal{A}$ with one-time access to two user oracles, we have:

$$\begin{aligned}\Pr\Big[&b \xleftarrow{\$} \{0,1\}, (\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda), (\mathsf{St}_1, m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}),\\ &(\mathsf{St}_2) \leftarrow \mathcal{A}(\mathsf{St}_1)^{\langle \mathcal{U}_{\mathsf{BS}}(m_b, \mathsf{pk}_{\mathsf{BS}}),\ \cdot\rangle^{(1)}, \langle \mathcal{U}_{\mathsf{BS}}(m_{1-b}, \mathsf{pk}_{\mathsf{BS}}),\ \cdot\rangle^{(1)}},\\ &\text{Let } \sigma_b \text{ and } \sigma_{1-b} \text{ be the resp. outputs of } \mathcal{U}_{\mathsf{BS}},\\ &\text{If } \sigma_0 = \bot \text{ or } \sigma_1 = \bot \text{ then } (\sigma_0, \sigma_1) = (\bot, \bot),\\ &b^* \leftarrow \mathcal{A}(\mathsf{St}_2, \sigma_0, \sigma_1) : b = b^*\Big] - \tfrac{1}{2} \leq \epsilon(\lambda).\end{aligned}$$

3.2 Partially Blind Signature Scheme

Definition 11. *A partially blind signature scheme consists of the following PPT algorithms* $\mathsf{PBS} = (\mathsf{KeyGen}_{\mathsf{PBS}}, \mathcal{U}_{\mathsf{PBS}}, \mathcal{S}_{\mathsf{PBS}}, \mathsf{Verify}_{\mathsf{PBS}})$ *defined as follows:*

$\mathsf{KeyGen}_{\mathsf{PBS}}(1^\lambda)$**:** *on input a security parameter, this algorithm outputs a pair of public/secret key* $(\mathsf{pk}_{\mathsf{PBS}}, \mathsf{sk}_{\mathsf{PBS}})$ *of the signer.*

$\langle \mathcal{U}_{\mathsf{PBS}}(m, \gamma \mathsf{pk}_{\mathsf{PBS}}), \mathcal{S}_{\mathsf{PBS}}(\gamma, \mathsf{sk}_{\mathsf{PBS}})\rangle$**:** *are executed by a user and a signer. On input common information γ, the signer's secret key* $\mathsf{sk}_{\mathsf{PBS}}$ *algorithm* $\mathcal{S}_{\mathsf{PBS}}$ *interacts with algorithm* $\mathcal{U}_{\mathsf{PBS}}$*. On input a message m and a common information γ, both from message space $\mathcal{M}$, and the signer public key* $\mathsf{pk}_{\mathsf{PBS}}$*, algorithm* $\mathcal{U}_{\mathsf{PBS}}$ *outputs a signature σ on m, or $\bot$, if the interaction was not successful.*

$\mathsf{Verify}_{\mathsf{PBS}}(m, \gamma, \sigma, \mathsf{pk}_{\mathsf{PBS}})$**:** *on input the message m, the common information γ, the signature σ and the signers public key* $\mathsf{pk}_{\mathsf{PBS}}$*, this procedure outputs* 1 *if σ is a valid signature and* 0 *otherwise.*

A partially blind signature scheme PBS must satisfy correctness, unforgeability and partial blindness as defined below.

Correctness. A partially blind signature scheme PBS is *correct*, if for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}_{\mathsf{PBS}}, \mathsf{sk}_{\mathsf{PBS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$, all messages $m \in \mathcal{M}$, all common information $\gamma \in \mathcal{M}$ and $\sigma \leftarrow \langle \mathcal{U}_{\mathsf{PBS}}(m, \gamma, \mathsf{pk}_{\mathsf{BS}}), \mathcal{S}_{\mathsf{PBS}}(\gamma, \mathsf{sk}_{\mathsf{PBS}})\rangle$ it holds that $\mathsf{Verify}_{\mathsf{PBS}}(m, \gamma, \sigma, \mathsf{pk}_{\mathsf{PBS}}) = 1$.

Unforgeability. A partially blind signature scheme PBS is *strongly unforgeable*, if for all PPT algorithms $\mathcal{A}$ having access to a signer oracle, we have:

$$\begin{aligned}\Pr\Big[&(\mathsf{pk}_{\mathsf{PBS}}, \mathsf{sk}_{\mathsf{PBS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{PBS}}(1^\lambda), (\gamma^*, (m_i^*, \sigma_i^*)_{i=1}^{k+1}) \leftarrow \mathcal{A}(\mathsf{pk}_{\mathsf{PBS}})^{\langle \cdot\ , \mathcal{S}_{\mathsf{PBS}}(\mathsf{sk}_{\mathsf{PBS}})\rangle} :\\ &m_i^* \neq m_j^* \quad \text{for } i, j \in \{1, \ldots, k+1\}, i \neq j \qquad \text{and}\\ &\mathsf{Verify}_{\mathsf{PBS}}(m_i^*, \gamma^*, \sigma_i^*, \mathsf{pk}_{\mathsf{PBS}}) = 1 \quad \text{for } i \in \{1, \ldots, k+1\}\Big] \leq \epsilon(\lambda),\end{aligned}$$

where k is the number of oracle queries.

Blindness. A partially blind signature scheme PBS is *blind* in the *honest-signer* model, if for all PPT algorithms $\mathcal{A}$ with one-time access to two user oracles, we have:

$$\begin{aligned}\Pr\Big[& b \xleftarrow{\$} \{0,1\}, (\mathsf{pk}_{\mathsf{PBS}}, \mathsf{sk}_{\mathsf{PBS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{PBS}}(1^\lambda), (\mathsf{St}_1, \gamma, m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk}_{\mathsf{PBS}}, \mathsf{sk}_{\mathsf{PBS}}),\\ & (\mathsf{St}_2) \leftarrow \mathcal{A}(\mathsf{St}_1)^{\langle \mathcal{U}_{\mathsf{PBS}}(m_b,\gamma,\mathsf{pk}_{\mathsf{PBS}}),\ \cdot\rangle^{(1)}, \langle \mathcal{U}_{\mathsf{PBS}}(m_{1-b},\gamma,\mathsf{pk}_{\mathsf{PBS}}),\ \cdot\rangle^{(1)}},\\ & \text{Let } \sigma_b \text{ and } \sigma_{1-b} \text{ be the resp. outputs of } \mathcal{U}_{\mathsf{PBS}},\\ & \text{If } \sigma_0 = \perp \text{ or } \sigma_1 = \perp \text{ then } (\sigma_0, \sigma_1) = (\perp, \perp),\\ & b^* \leftarrow \mathcal{A}(\mathsf{St}_2, \sigma_0, \sigma_1) : b = b^*\Big] - \tfrac{1}{2} \leq \epsilon(\lambda).\end{aligned}$$

4 Blind Signatures from Knowledge Assumptions

In this section we present our blind signature scheme. The construction is in fact the single-message protocol presented by Pointcheval and Sanders, for their randomizable signature scheme [16]. It allows to sign committed messages but requires a proof of knowledge protocol of commitment opening. Thus a concrete instantiation requires additional rounds (Schnorr like sigma protocol), random oracles (via Fiat-Shamir transformation) or a common reference string (non-interactive proof systems e.g. Groth-Sahai).

In our construction we get rid of the proof of knowledge and require the user to compute some additional value. In particular, the signer publishes $(P_1, [y]P_1, [k]P_1, [k \cdot y]P_1)$ and for a given message m, the user chooses t and computes the Pedersen commitment $C_1 = [t]P_1 + [m]([y]P_1)$. In the original protocol from [16], the user must proof that it knows m and t. However, in our construction it is only required that the user additionally computes and sends $C_2 = [t]([k]P_1) + [m]([k \cdot y]P_1)$ to the signer. The proof of security follows then from the knowledge of exponent assumption (KEA3).

Theorem 3 (Correctness). *Scheme 1 is correct.*

Proof. Correctness of the scheme follows directly from the correctness of the randomizable signature scheme in [16]. Note that the additional element $C_2 = [t]\hat{P}_1 + [m]\hat{Y}_1$ and the equation $[k]C_1 = C_2$ is always satisfied if the user behaves according to protocol. However, for a sense of completeness we will sketch the idea behind correctness. First see that the user computes the commitment $C_1 = [t]P_1 + [m]Y_1$, which is used by the signer to compute $(\sigma_1, \sigma_2) = ([u]P_1, [u](X + C_1))$. Thus, $[u](X + C_1) = [u](X + [m]Y_1 + [t]P_1))$ and by computing $\sigma = (\sigma_1, \sigma_2 - [t]\sigma_1)$ the user receives $\sigma = ([u]P_1, [u](X + [m]Y_1))$, which is a valid signature under m.

Theorem 4 (Unforgeability). *Scheme 1 is unforgeable.*

Proof (Sketch). We will show that if there exists an algorithm $\mathcal{A}$ that has non-negligible advantage in breaking unforgeability of scheme 1, then we can construct an algorithm $\mathcal{R}$ that breaks the EUF-CMA security of the used short randomizable signatures.

$\mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$**:** Generate bilinear group parameters $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow \mathsf{BGGen}(1^\lambda)$. Compute the short randomizable signature scheme key pair $(\mathsf{sk}_{\mathsf{PS}}, \mathsf{pk}_{\mathsf{PS}}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathsf{PS}}(\mathsf{BG})$, where $\mathsf{pk}_{\mathsf{PS}} = (\mathsf{BG}, X_2, Y_1, Y_2)$. Compute random $k \xleftarrow{\$} \mathbb{Z}_q$ and set the secret key $\mathsf{sk}_{\mathsf{BS}} = (\mathsf{sk}_{\mathsf{PS}}, k)$. Compute $\hat{P}_1 = [k]P_1$, $\hat{Y}_1 = [k]Y_1$ and the public key $\mathsf{pk}_{\mathsf{BS}} = (\mathsf{pk}_{\mathsf{PS}}, \hat{P}_1, \hat{Y}_1)$.

$\mathcal{U}^{(1)}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}})$**:** generate the parameters $\mathsf{BG} \leftarrow \mathsf{BGGen}(1^\lambda)$. Parse $\mathsf{pk}_{\mathsf{BS}}$ as $((\mathsf{BG}, X_2, Y_1, Y_2), \hat{P}_1, \hat{Y}_1)$, choose $t \xleftarrow{\$} \mathbb{Z}_q$ and compute $\rho = ([t]P_1 + [m]Y_1, [t]\hat{P}_1 + [m]\hat{Y}_1)$. Set $\mathsf{St}_{\mathsf{BS}} = (m, t)$ and send ρ to the signer.

$\mathcal{S}_{\mathsf{BS}}(\rho, \mathsf{sk}_{\mathsf{BS}})$**:** Parse $\mathsf{sk}_{\mathsf{BS}}$ as (X, k) and ρ as (C_1, C_2). Abort if $[k]C_1 \neq C_2$. Compute $u \xleftarrow{\$} \mathbb{Z}_q$ and send $\beta = ([u]P_1, [u](X + C_1))$ to the user.

$\mathcal{U}^{(2)}_{\mathsf{BS}}(\beta, \mathsf{St}_{\mathsf{BS}}, \mathsf{pk}_{\mathsf{BS}})$**:** Parse β as (σ_1, σ_2), $\mathsf{St}_{\mathsf{BS}}$ as (m, t) $\mathsf{pk}_{\mathsf{BS}}$ as $(\mathsf{pk}_{\mathsf{PS}}, \cdot, \cdot)$ and compute $\sigma = (\sigma_1, \sigma_2 - [t]\sigma_1)$. Return $\perp$ if $\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}}) = 0$; otherwise return σ.

$\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}})$**:** Parse $\mathsf{pk}_{\mathsf{BS}}$ as $(\mathsf{pk}_{\mathsf{PS}}, \cdot, \cdot)$ and return 1 iff $\mathsf{Verify}_{\mathsf{PS}}(m, \sigma, \mathsf{pk}_{\mathsf{PS}}) = 1$ and $e(Y_1, P_2) = e(P_1, Y_2)$ and $e(\hat{P}_1, Y_2) = e(\hat{Y}_1, P_2)$.

Scheme 1: Our Blind Signature Scheme

First $\mathcal{R}$ sets up the system using the public key $\mathsf{pk}_{\mathsf{PS}}$, i.e. creates the blind signature public key $\mathsf{pk}_{\mathsf{BS}}$. Then $\mathcal{R}$ starts interacting with $\mathcal{A}$. After each signature query of $\mathcal{A}$, $\mathcal{R}$ runs the KEA3 extractor and receives m, t. $\mathcal{R}$ then asks the EUF-CMA oracle for a signature $(\sigma_1, \sigma 2)$ under the message m. To answer $\mathcal{A}$'s query, the algorithm $\mathcal{R}$ returns $(\sigma_1, \sigma_2 + [t]\sigma_1)$. In order to win, $\mathcal{A}$ must output a message m^* and a signature σ^*, such that m^* was now queried to $\mathcal{R}$. Thus, m^* was also not queried by $\mathcal{R}$ to the EUF-CMA oracle and $\mathcal{R}$ can return (m^*, σ^*) as a valid forge.

Theorem 5 (Blindness). *Scheme 1 is blind in the honest-signer model.*

Proof (Sketch). We will show that there exists no algorithm $\mathcal{A}$ that has non-negligible advantage in breaking the blindness of scheme 1. It is easy to see that given the secret key $\mathsf{sk}_{\mathsf{BS}}$, we can open the user commitments to arbitrary messages and even sign arbitrarily messages. What is more, the resulting signatures are randomizable. Thus, from the point of view of any adversary $\mathcal{A}$ the blindness experiment in case $b = 0$ is indistinguishable from the blindness experiment in case $b = 1$.

5 Partially Blind Signatures from Knowledge Assumptions

In this section we propose an extension of our blind signature scheme to partially blind signatures. We use the idea of Pedersen commitments, i.e. instead of signing the message m, the signed message is $m + \gamma \cdot r$, where r is a secret chosen by the signer. Similar to the binding property, this approach protects against the changing of the signed message (as long as the co-DLP assumption holds).

$\mathsf{KeyGen}_{\mathsf{PBS}}(1^\lambda)$**:** Generate bilinear group parameters $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow \mathsf{BGGen}(1^\lambda)$. Compute the short randomizable signature scheme key pair $(\mathsf{sk}_{\mathsf{PS}}, \mathsf{pk}_{\mathsf{PS}}) \xleftarrow{\$} \mathsf{KeyGen}_{\mathsf{PS}}(\mathsf{BG})$, where $\mathsf{pk}_{\mathsf{PS}} = (\mathsf{BG}, X_2, Y_1, Y_2)$. Compute random $k, r \xleftarrow{\$} \mathbb{Z}_q$ and set the secret key $\mathsf{sk}_{\mathsf{PBS}} = (\mathsf{sk}_{\mathsf{PS}}, k, r)$. Compute $\hat{P}_1 = [k]P_1$, $\hat{Y}_1 = [k]Y_1$, $Y_3 = [r]Y_2$ and set the public key $\mathsf{pk}_{\mathsf{PBS}} = (\mathsf{pk}_{\mathsf{PS}}, \hat{P}_1, \hat{Y}_1, Y_3)$.

$\mathcal{U}^{(1)}_{\mathsf{BS}}(m, \gamma, \mathsf{pk}_{\mathsf{PBS}})$**:** generate the parameters $\mathsf{BG} \leftarrow \mathsf{BGGen}(1^\lambda)$. Parse $\mathsf{pk}_{\mathsf{PBS}}$ as $((\mathsf{BG}, X_2, Y_1, Y_2), \hat{P}_1, \hat{Y}_1, Y_3)$, choose $t \xleftarrow{\$} \mathbb{Z}_q$ and compute $\rho = ([t]P_1 + [m]Y_1, [t]\hat{P}_1 + [m]\hat{Y}_1)$. Set $\mathsf{St}_{\mathsf{PBS}} = (m, t)$ and send ρ to the signer.

$\mathsf{Issue}_{\mathsf{PBS}}(\rho, \gamma, \mathsf{sk}_{\mathsf{PBS}})$**:** Parse $\mathsf{sk}_{\mathsf{PBS}}$ as (X, k) and ρ as (C_1, C_2). Abort if $[k]C_1 \neq C_2$. Compute $u \xleftarrow{\$} \mathbb{Z}_q$ and send $\beta = ([u]P_1, [u](X + C_1 + [\gamma \cdot r]Y_1))$ to the user.

$\mathsf{Unblind}_{\mathsf{PBS}}(\beta, \mathsf{St}_{\mathsf{PBS}}, \mathsf{pk}_{\mathsf{PBS}})$**:** Parse β as (σ_1, σ_2), $\mathsf{St}_{\mathsf{PBS}}$ as (m, t) $\mathsf{pk}_{\mathsf{PBS}}$ as $(\mathsf{pk}_{\mathsf{PS}}, \cdot, \cdot)$ and compute $\sigma = (\sigma_1, \sigma_2 - [t]\sigma_1)$. Return $\perp$ if $\mathsf{Verify}_{\mathsf{PBS}}(m, \gamma, \sigma, \mathsf{pk}_{\mathsf{BS}}) = 0$; otherwise return σ.

$\mathsf{Verify}_{\mathsf{PBS}}(m, \gamma, \sigma, \mathsf{pk}_{\mathsf{BS}})$**:** Parse $\mathsf{pk}_{\mathsf{PBS}}$ as $(\mathsf{pk}_{\mathsf{PS}}, \cdot, \cdot)$ and return 1 iff $\sigma_1 \neq 1_{\mathbb{G}_1}$ and $e(\sigma_1, X_2 + [m]Y_2 + [\gamma]Y_3) = e(\sigma_2, P_2)$ and $e(Y_1, P_2) = e(P_1, Y_2)$ and $e(\hat{P}_1, Y_2) = e(\hat{Y}_1, P_2)$.

Scheme 2: Our Partially Blind Signature Scheme

Theorem 6 (Correctness). *Scheme 2 is correct.*

Proof. Similar to scheme 1, correctness of the scheme follows directly from the correctness of the randomizable signature scheme in [16].

Theorem 7 (Unforgeability). *Scheme 2 is unforgeable.*

Proof (Sketch). It is easy to see that the same proof as in case of theorem 4 can be applied. However, this time instead of sending m to the EUF-CMA oracle, the algorithm $\mathcal{R}$ queries $m + \gamma \cdot r$. Note that $\mathcal{R}$ fails if the adversary knows r. Therefore, we have to include an additional case in algorithm $\mathcal{R}$. Depending on a coin toss, $\mathcal{R}$ either works as in case of theorem 4 or solves the co-DLP problem (i.e given $[u]g_1 \in \mathbb{G}_1$, $[u]g_2 \in \mathbb{G}_2$, compute u). To do so, $\mathcal{R}$ sets Y_3 to be $[u]g_2$ for the solved co-DLP instance. $\mathcal{R}$ works according to protocol. However, instead of using $[r]Y_1$ it uses $[u]g_1$. Finally, the adversary outputs message-signature pairs. For one of them exists a message $m_i^* + \gamma_i^* \cdot u = m + \gamma \cdot u$, such that $\gamma_i^* \neq \gamma$ and m_i^*, γ_i^*, m, γ are known $\mathcal{R}$. Therefore, $\mathcal{R}$ is able to compute u and solve the co-DLP problem.

Theorem 8 (Blindness). *Scheme 2 is blind in the honest-signer model.*

Proof (Sketch). Note that since the secret key is computed in an honest way, the same reasoning as in the proof of theorem 5 can be used.

Conclusions

We have proposed a fairly practical two-move blind signature without random oracles and a common reference string. It is efficient in terms of signature size and communication complexity. For a future work we plan to extend blindness to the malicious-signer model, where the adversary generates the signing key. One promising approach is to use the knowledge of exponent assumption to extract the signer's secret key.

Acknowledgments. This research was supported by the National Science Centre (Poland) based on decision no 2014/15/N/ST6/04577.

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) Advances in Cryptology CRYPTO 2010, Lecture Notes in Computer Science, vol. 6223, pp. 209–236. Springer Berlin Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-14623-7_12
2. Barreto, P.S.L.M., Naehrig, M.: Pairing-Friendly Elliptic Curves of Prime Order. In: Preneel, B., Tavares, S.E. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 3897, pp. 319–331. Springer (2005), http://dblp.uni-trier.de/db/conf/sacrypt/sacrypt2005.html#BarretoN05
3. Bellare, Namprempre, Pointcheval, Semanko: The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. Journal of Cryptology 16(3), 185–215 (2003), http://dx.doi.org/10.1007/s00145-002-0120-1
4. Bellare, M., Palacio, A.: The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In: Advances in Cryptology - CRYPTO 2004, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3152, pp. 273–289. Springer (2004), http://www.iacr.org/cryptodb/archive/2004/CRYPTO/961/961.pdf

5. Chaum, D.: Blind Signatures for Untraceable Payments. In: Advances in Cryptology: Proceedings of CRYPTO '82. pp. 199–203. Plenum (1982)
6. Chaum, D.: Blind Signature System. In: Chaum, D. (ed.) Advances in Cryptology, pp. 153–153. Springer US (1984), http://dx.doi.org/10.1007/978-1-4684-4730-9_14
7. Fischlin, M.: Round-Optimal Composable Blind Signatures in the Common Reference String Model. In: Dwork, C. (ed.) Advances in Cryptology - CRYPTO 2006, Lecture Notes in Computer Science, vol. 4117, pp. 60–77. Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/11818175_4
8. Fischlin, M., Schrder, D.: On the Impossibility of Three-Move Blind Signature Schemes. In: Gilbert, H. (ed.) Advances in Cryptology EUROCRYPT 2010, Lecture Notes in Computer Science, vol. 6110, pp. 197–215. Springer Berlin Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-13190-5_10
9. Fuchsbauer, G., Hanser, C., Slamanig, D.: EUF-CMA-Secure Structure-Preserving Signatures on Equivalence Classes. Cryptology ePrint Archive, Report 2014/944 (2014), http://eprint.iacr.org/
10. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical Round-Optimal Blind Signatures in the Standard Model. Cryptology ePrint Archive, Report 2015/626 (2015), http://eprint.iacr.org/
11. Garg, S., Gupta, D.: Efficient Round Optimal Blind Signatures. In: Nguyen, P., Oswald, E. (eds.) Advances in Cryptology EUROCRYPT 2014, Lecture Notes in Computer Science, vol. 8441, pp. 477–495. Springer Berlin Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-55220-5_27
12. Garg, S., Rao, V., Sahai, A., Schrder, D., Unruh, D.: Round Optimal Blind Signatures. In: Rogaway, P. (ed.) Advances in Cryptology CRYPTO 2011, Lecture Notes in Computer Science, vol. 6841, pp. 630–648. Springer Berlin Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-22792-9_36
13. Ghadafi, E., Smart, N.: Efficient Two-Move Blind Signatures in the Common Reference String Model. In: Gollmann, D., Freiling, F. (eds.) Information Security, Lecture Notes in Computer Science, vol. 7483, pp. 274–289. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-33383-5_17
14. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) Advances in Cryptology EUROCRYPT 2008, Lecture Notes in Computer Science, vol. 4965, pp. 415–432. Springer Berlin Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-78967-3_24
15. Miyaji, A., Nakabayashi, M., Takano, S.: New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 84(5), 1234–1243 (2001)
16. Pointcheval, D., Sanders, O.: Short randomizable signatures. Cryptology ePrint Archive, Report 2015/525 (2015), http://eprint.iacr.org/