Two-Move and Setup-Free Blind Signatures with Perfect Blindness

Lucjan Hanzlik
Faculty of Fundamental Problems of Technology
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27
50-370 Wrocław, Poland
lucjan.hanzlik@pwr.edu.pl

Kamil Kluczniak
Faculty of Fundamental Problems of Technology
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27
50-370 Wrocław, Poland
kamil.kluczniak@pwr.edu.pl

ABSTRACT

At ISC'12 Ghadafi and Smart proposed a two-move blind signature scheme based on a new variant of the interactive LRSW assumption and the perfect hiding property of Pedersen commitments. In this paper we show that there exists a simple algorithm that always breaks the blindness property of their scheme. We show how to fix this issue and propose the first two-move blind signature scheme with perfect blindness in the dishonest-key model, but without random oracles and a common-reference string. The efficiency of our solution is comparable with the previous best scheme proposed by Fuchsbauer et al. at Crypto'15. In particular our blind signature scheme is more efficient in terms of communication complexity and the size of signatures is 50% shorter. What is more, the instantiations given by Fuchsbauer et al., for its security proof, require interactive assumptions for both blindness and unforgeability, where for our constructions we only use such assumptions for unforgeability.

CCS Concepts

•Security and privacy → Digital signatures; Cryptanalysis and other attacks;

Keywords

Blind Signatures, Standard Model, Interactive Assumptions

1. INTRODUCTION

The concept of blind signatures was first introduced by David Chaum in his work [6]. He also gave the first application for this primitive, namely e-cash. The idea was to protect privacy of users 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 its signature request (*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. Over 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, in particular the number of moves a user and signer must perform during the issuance procedure. Two-move blind signatures (also called *round-optimal* [9]) are of particular interest as they directly yield concurrent security.

The papers [4, 7] give efficient and round-optimal blind signatures with security in the random oracle model. 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 [17]. 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 [9]. 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. [2] using structure-preserving signatures and Groth-Sahai proofs [18].

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. Unfortunately, in many applications such a trusted party may not be acceptable. Moreover, in the 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. [10] it is impossible to construct a blind signature scheme whose 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 these limitations.

APKC'17, April 02 2017, Abu Dhabi, United Arab Emirates

DOI: http://dx.doi.org/10.1145/3055504.3055505

Garg et al. [16] were the first to propose a generic construction in the standard model. However, their 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 [15] 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 [18], where the common reference string is a part of the signer's 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 the use of a technique called complexity leveraging. As a consequence, the computational and communication complexity limits the usage in many practical applications.

Recently, Fuchsbauer et al. [12] 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 prove 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 [11], whose security is based on an interactive assumption.

Our Contribution.. The first contribution of this paper is the security analysis of the scheme proposed by Ghadafi and Smart [17]. In particular we show that one can construct a simple malicious signer that always breaks blindness of their scheme. We show that this is possible because there exists an alternative signing procedure that yields valid signatures if the signer guesses the correct message in a Pedersen commitment (in the blindness game there are only two possible messages). We also discuss this issue and its implications for a real-world implementation.

The main contribution of this paper is a blind signature scheme based on the scheme from [17]. First, we propose how to fix the problem using a modified version of the E-LRSW assumption called E'-LRSW. Then we show that the common reference string is not necessary since Pedersen commitments are perfectly hiding, even if the commitment key is generated by the receiver. Moreover, we can trust the group parameters in the signer's public key if we use a deterministic parameter generator.

Combining these ideas we obtain surprising results: a blind signature scheme that is *the first* two-move, setup-free and is *perfectly blind* in the malicious-key model without random oracles. Moreover, our solution is comparable in terms of efficiency with the previously best results from [12], presented at Crypto'15. Our blind signature scheme is more efficient in terms of communication complexity. However, the most interesting improvement is the reduced signature size. The scheme proposed in this paper has signatures that are 50% shorter than the ones in [12] (at a 256-bit (resp. 512-bit) representation of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$)). What is more, we use an interactive assumption only to prove unforgeability, where the instantiation given in [12] requires such assumptions for both unforgeability and blindness.

Finally, if we instantiate the underlying bilinear groups with the popular BN-curves [3], then all procedures in our solution, beside verification, use operations in prime order multiplicative groups and on standard elliptic curves. It follows that our construction could in practice be used with standard smart cards (e.g. in the Java Card [24] or Multos [22] technology). However, the verification procedure would have to be delegated to readers but this may not be an issue as it only requires public values. Note that practical application like e-voting, e-cash and anonymous credentials usually take advantage of security tokens such as smart cards.

2. PRELIMINARIES

Before presenting our contribution we briefly review a few facts about bilinear maps, Pedersen commitments and assumptions used in this paper.

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:

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

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 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* $\mathbb{G}_T$,

According to the classification from [14], 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 there are no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

Bilinear map groups are known to be instantiable with ordinary elliptic curves such as MNT curves [21] or curves introduced by Barreto and Naehrig [3] (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 [12] we assume that* BGGen *is deterministic (which is the case for BN-curves [3]).*

2.2 LRSW Assumption and CL-Signatures

We now recall the well-known LRSW assumption [19] and its variation called E-LRSW. The latter was used by Ghadafi and Smart in their blind signature scheme [17]. Both problems were shown to hold in the generic group model and are interactive assumptions. Then we introduce a modification of the E-LRSW assumption which we call E'-LRSW and show that it also holds in the generic group model.

Definition 4 (LRSW Assumption [19]). *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. *Let* $X = [x]P_2$, $Y = [y]P_2$ *and define* $O_{X,Y}(\cdot)$ *to be an oracle that, on input a value* $m \in \mathbb{Z}_q$, *outputs* $(A, [y]A, [x + m \cdot x \cdot y]A) \in \mathbb{G}_1^3$, *where* $A \xleftarrow{\$} \mathbb{G}_1$.

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

$$\Pr[\mathsf{BG} \leftarrow \mathsf{BGGen}(\lambda), x \xleftarrow{\$} \mathbb{Z}_q, y \xleftarrow{\$} \mathbb{Z}_q, X = [x]P_2, Y = [y]P_2,$$
$$(m, A, [y]A, [x + m \cdot x \cdot y]A) \leftarrow \mathcal{A}^{O_{X,Y}(\cdot)}(\mathsf{BG}, X, Y) :$$
$$m \notin Q \ \wedge \ m \in \mathbb{Z}_q \backslash \{0\} \ \wedge \ A \in \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}],$$

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

Definition 5 (E-LRSW Assumption [17]). *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. *Let* $X = [x]P_2$, $Y = [y]P_2$, $Z = [z]P_1$ *and define* $O_{X,Y,Z}(\cdot)$ *to be an oracle that, on input of a value* $M = [m]P_1 \in \mathbb{G}_1$ *(with unknown* m*), outputs* $(A, [y]A, [x + m \cdot x \cdot y]A, [x \cdot y \cdot z]A) \in \mathbb{G}_1^4$, *where* $A \xleftarrow{\$} \mathbb{G}_1$.

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

$$\Pr[\mathsf{BG} \leftarrow \mathsf{BGGen}(\lambda), x \xleftarrow{\$} \mathbb{Z}_q, y \xleftarrow{\$} \mathbb{Z}_q, z \xleftarrow{\$} \mathbb{Z}_q,$$
$$X = [x]P_2, Y = [y]P_2, Z = [z]P_1,$$
$$(\{m_i, A_i, [y]A_i, [x + m_i \cdot xy]A_i\}_{i=1}^{k+1}) \leftarrow \mathcal{A}^{O_{X,Y,Z}(\cdot)}(\mathsf{BG}, X,$$
$$Y, Z) : \forall_{i=1}^{k+1} m_i \in \mathbb{Z}_q \backslash \{0\} \ \wedge \ A_i \in \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}$$
$$\wedge (\textit{if } i \neq j, \textit{ then } m_i \neq m_j)],$$

where k *denotes the number of queries made by* $\mathcal{A}$ *to oracle* $O_{X,Y,Z}(\cdot)$.

The LRSW Assumption was used by Camenisch and Lysyanskaya (or in short CL-Signatures) in their signature scheme [5].

Definition 6 (CL-Signatures [5]). *The CL signature scheme is given by the following triple of algorithms given an output* BG *of* $\mathsf{BGGen}(\lambda)$.

$\mathsf{KeyGen}_{\mathsf{CL}}(\mathsf{BG})$**:**
Choose the private key $\mathsf{sk}_{\mathsf{CL}} = (x, y) \xleftarrow{\$} (\mathbb{Z}_q^*)^2$ *and* $\mathsf{pk}_{\mathsf{CL}} = (X, Y) = ([x]P_2, [y]P_2)$.

$\mathsf{Sign}_{\mathsf{CL}}(m, \mathsf{sk}_{\mathsf{CL}})$**:**
Select $A \xleftarrow{\$} \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}$ *and compute* $B = [y]A$, $C = [x + m \cdot x \cdot y]A$. *Output* (A, B, C).

$\mathsf{Verify}_{\mathsf{CL}}(m, (A, B, C), \mathsf{pk}_{\mathsf{CL}})$**:**
Output 1 *if and only if* $A \neq 1_{\mathbb{G}_1}$, $e(B, P_2) = e(A, Y)$ *and* $e(C, P_2) = e(A, X) \cdot e(B, X)^m$.

Definition 7 (Randomization of CL Signatures). *For all tuples* $(\mathsf{pk}_{\mathsf{CL}}, m, (A, B, C))$, *where*

$$\mathsf{Verify}_{\mathsf{CL}}(m, (A, B, C), \mathsf{pk}_{\mathsf{CL}}) = 1 \quad \textit{and} \quad m \in \mathbb{Z}_q,$$

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

Definition 8 (E'-LRSW Assumption). *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. *Let* $X = [x]P_2$, $Y = [y]P_2$, $Z = [z]P_1$, $W = [z]X$, *and define* $O_{X,Y,Z,W}(\cdot)$ *to be an oracle that, on input a value* $M = [m]P_1 \in \mathbb{G}_1$ *(with unknown* m*), outputs* $(A, [y]A, [x + m \cdot x \cdot y]A, [x \cdot y \cdot z]A) \in \mathbb{G}_1^4$, *where* $A \xleftarrow{\$} \mathbb{G}_1$.

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

$$\Pr[\mathsf{BG} \leftarrow \mathsf{BGGen}(\lambda), x \xleftarrow{\$} \mathbb{Z}_q, y \xleftarrow{\$} \mathbb{Z}_q, z \xleftarrow{\$} \mathbb{Z}_q,$$
$$X = [x]P_2, Y = [y]P_2, Z = [z]P_1, W = [z]X, (\{m_i, A_i, [y]A_i,$$
$$[x + m_i \cdot x \cdot y]A_i\}_{i=1}^{k+1}) \leftarrow \mathcal{A}^{O_{X,Y,Z,W}(\cdot)}(\mathsf{BG}, X, Y, Z, W) :$$
$$\forall_{i=1}^{k+1} m_i \in \mathbb{Z}_q \backslash \{0\} \ \wedge \ A_i \in \mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}$$
$$\wedge (\textit{if } i \neq j \textit{ then } m_i \neq m_j)],$$

k *denotes the number of queries made by* $\mathcal{A}$ *to* $O_{X,Y,Z,W}(\cdot)$.

Theorem 1. *Let* $\mathcal{A}$ *denote an adversary in the generic group model (or GGM [20] in short) against the E'-LRSW assumption. Assume* $\mathcal{A}$ *makes* $q_{\mathbb{G}}$ *queries to the group operations,* q_e *queries to the pairing and* q_O *queries to the E'-LRSW oracle. If we set* $n = 6 + q_{\mathbb{G}} + 4q_O + q_e$, *then the adversaries probability of winning the E'-LRSW game is* $O(n^2 \cdot q_O/q)$, *where* q *is the (prime) order of the generic group.*

3. BLIND SIGNATURES

In this section we recall the syntax for blind signature schemes. Moreover, we define security properties that such schemes must satisfy. Since the focus of this paper is put on round optimal (two-move) blind signatures, we simplify the syntax and security games for this case only. For blind signatures with more rounds, the signer and the user are modelled as interactive algorithms.

3.1 Syntax of Blind Signatures Schemes

DEFINITION 9. *A (two-move) blind signature scheme consists of the following algorithms* $\mathsf{BS} = (\mathsf{Setup}_{\mathsf{BS}}, \mathsf{KeyGen}_{\mathsf{BS}}, \mathsf{Request}_{\mathsf{BS}}, \mathsf{Issue}_{\mathsf{BS}}, \mathsf{Unblind}_{\mathsf{BS}}, \mathsf{Verify}_{\mathsf{BS}})$ *defined as follows:*

$\mathsf{Setup}_{\mathsf{BS}}(\lambda)$**:** *on input a security parameter, this procedure outputs a common reference string* $\mathsf{CRS}_{\mathsf{BS}}$ *(we assume that this reference string is an implicit parameter for all later algorithms).*

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

$\mathsf{Request}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}})$**:** *on input a message* m*, from the message space* $\mathcal{M}$*, and the signer public key* $\mathsf{pk}_{\mathsf{BS}}$*, this procedure, run by the user, produces a signature request* ρ *and some state information* $\mathsf{St}_{\mathsf{BS}}$*.*

$\mathsf{Issue}_{\mathsf{BS}}(\rho, \mathsf{sk}_{\mathsf{BS}})$**:** *on input a signature request* ρ *and a secret key* $\mathsf{sk}_{\mathsf{BS}}$*, this algorithm, run by the signer, produces a pre-signature* β*.*

$\mathsf{Unblind}_{\mathsf{BS}}(\beta, \mathsf{St}_{\mathsf{BS}}, \mathsf{pk}_{\mathsf{BS}})$**:** *on input a pre-signature* β*, state information* $\mathsf{St}_{\mathsf{BS}}$ *and the signer's public key* $\mathsf{pk}_{\mathsf{BS}}$*, this algorithm produces a blind signature* σ *on* m*, or outputs* $\perp$*.*

$\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}})$**:** *on input a message* m*, a 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.*

Note that the $\mathsf{Setup}_{\mathsf{BS}}$ algorithm is redundant in the setup-free setting but we use this more general definition to describe the blind signature scheme proposed in [17] which relies on a CRS.

We now describe the security properties that need to be satisfied by a blind signature scheme. The following definitions, if a CRS is used, must hold for all CRS's returned by the $\mathsf{Setup}_{\mathsf{BS}}$ procedure.

3.1.1 Correctness.

Correctness is what one would expect, i.e. we have:

$$\Pr[(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(\lambda), m \xleftarrow{\$} \mathcal{M}, (\rho, \mathsf{St}_{\mathsf{BS}}) \leftarrow \mathsf{Request}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}}), \beta \leftarrow \mathsf{Issue}_{\mathsf{BS}}(\rho, \mathsf{sk}_{\mathsf{BS}}), \sigma \leftarrow \mathsf{Unblind}_{\mathsf{BS}}(\beta, \mathsf{St}_{\mathsf{BS}}, \mathsf{pk}_{\mathsf{BS}}) : \mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}}) = 1] = 1.$$

Informally, this means that if both parties (user and signer) behave honestly, the signature should verify.

3.1.2 Unforgeability.

Informally, a blind signature is unforgeable, if there exists no efficient adversary $\mathcal{A}$, that given oracle access to the procedure $\mathsf{Issue}_{\mathsf{BS}}(\cdot, \mathsf{sk}_{\mathsf{BS}})$ returns $k+1$ valid message/signature pairs with different messages after at most k queries to the oracle.

DEFINITION 10. *A blind signature scheme* $\mathsf{BS} = (\mathsf{KeyGen}_{\mathsf{BS}}, \mathsf{Request}_{\mathsf{BS}}, \mathsf{Issue}_{\mathsf{BS}}, \mathsf{Unblind}_{\mathsf{BS}}, \mathsf{Verify}_{\mathsf{BS}})$ *is called* unforgeable *if for any efficient algorithm* $\mathcal{A}$ *the probability that experiment* $\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ *(Figure 1) evaluates to* 1 *is negligible in the security parameter* λ*.*

Experiment $\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$**:**
$(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(\lambda)$
$((m_1^*, \sigma_1^*), \ldots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow \mathcal{A}^{\mathsf{Issue}_{\mathsf{BS}}(\cdot, \mathsf{sk}_{\mathsf{BS}})}(\mathsf{pk}_{\mathsf{BS}})$
Return 1 iff
$m_i^* \neq m_j^*$ for all $i, j \in \{1, \ldots, k+1\}$ with $i \neq j$, and
$\mathsf{Verify}_{\mathsf{BS}}(m_i^*, \sigma_i^*, \mathsf{pk}_{\mathsf{BS}}) = 1$ for all $i \in \{1, \ldots, k+1\}$, and
$\mathcal{A}$ called the oracle less than $k+1$ times.

Figure 1: Experiment $\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$

We will denote $\mathbf{Adv}^{\mathsf{Unforge}}_{\mathcal{A}}(\lambda) = \Pr[\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1]$ *as the adversary's* $\mathcal{A}$ *advantage in forging a signature.*

3.1.3 Blindness.

Informally, a blind signature scheme satisfies blindness if, it is infeasible for a malicious signer $\mathcal{S}^*$ to decide which of two messages m_0 and m_1 have been signed first in two executions with an honest user. We distinguish two definitions: the classical (honest key model) and the stronger dishonest-key blindness [1] (malicious key model). In the classical definition the key pair $(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}})$ is given to the malicious signer $\mathcal{S}^*$, whereas in the latter we allow $\mathcal{S}^*$ to generate the signer's public key maliciously.

DEFINITION 11. *A blind signature scheme* $\mathsf{BS} = (\mathsf{KeyGen}_{\mathsf{BS}}, \mathsf{Request}_{\mathsf{BS}}, \mathsf{Issue}_{\mathsf{BS}}, \mathsf{Unblind}_{\mathsf{BS}}, \mathsf{Verify}_{\mathsf{BS}})$ *is called* blind*, in the classical sense, if for any efficient algorithm* $\mathcal{S}^*$ *(working in* find*,* issue *and* guess *modes) the probability that experiment* $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$ *(Figure 2) evaluates to* 1 *is negligible close to* $1/2$*. Moreover, we call the scheme* dishonest-key blind *if for any efficient algorithm* $\mathcal{S}^*$ *(working in* find*,* issue *and* guess *modes) the probability that experiment* $\mathsf{DisKeyBlind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$ *(Figure 3) evaluates to* 1 *is close to* $1/2$*.*

Experiment $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$**:**
$(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(\lambda)$
$(m_0, m_1, \mathsf{St}_{\mathsf{find}}) \leftarrow \mathcal{S}^*(\mathsf{find}, \mathsf{sk}_{\mathsf{BS}}, \mathsf{pk}_{\mathsf{BS}})$
$b \xleftarrow{\$} \{0, 1\}$
$(\rho_b, \mathsf{St}_b) \leftarrow \mathsf{Request}_{\mathsf{BS}}(m_0, \mathsf{pk}_{\mathsf{BS}})$
$(\rho_{1-b}, \mathsf{St}_{1-b}) \leftarrow \mathsf{Request}_{\mathsf{BS}}(m_1, \mathsf{pk}_{\mathsf{BS}})$
$(\beta_0, \beta_1, \mathsf{St}_{\mathsf{issue}}) \leftarrow \mathcal{S}^*(\mathsf{issue}, \rho_0, \rho_1, \mathsf{St}_{\mathsf{find}})$
$(\sigma_0) \leftarrow \mathsf{Unblind}_{\mathsf{BS}}(\beta_b, \mathsf{St}_b, \mathsf{pk}_{\mathsf{BS}})$
$(\sigma_1) \leftarrow \mathsf{Unblind}_{\mathsf{BS}}(\beta_{1-b}, \mathsf{St}_{1-b}, \mathsf{pk}_{\mathsf{BS}})$
if $\sigma_0 = \perp$ or $\sigma_1 = \perp$ then set $\sigma_0 = \perp$ and $\sigma_1 = \perp$
$\hat{b} \leftarrow \mathcal{S}^*(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{St}_{\mathsf{issue}})$
Return 1 if $\hat{b} = b$ else return 0

Figure 2: Experiment $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$

Experiment $\mathsf{DisKeyBlind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$**:**
$(m_0, m_1, \mathsf{pk_{BS}}, \mathsf{St_{find}}) \leftarrow \mathcal{S}^*(\mathsf{find}, \lambda)$
$b \xleftarrow{\$} \{0,1\}$
$(\rho_b, \mathsf{St}_b) \leftarrow \mathsf{Request_{BS}}(m_0, \mathsf{pk_{BS}})$
$(\rho_{1-b}, \mathsf{St}_{1-b}) \leftarrow \mathsf{Request_{BS}}(m_1, \mathsf{pk_{BS}})$
$(\beta_0, \beta_1, \mathsf{St_{issue}}) \leftarrow \mathcal{S}^*(\mathsf{issue}, \rho_0, \rho_1, \mathsf{St_{find}})$
$(\sigma_0) \leftarrow \mathsf{Unblind_{BS}}(\beta_b, \mathsf{St}_b, \mathsf{pk_{BS}})$
$(\sigma_1) \leftarrow \mathsf{Unblind_{BS}}(\beta_{1-b}, \mathsf{St}_{1-b}, \mathsf{pk_{BS}})$
if $\sigma_0 = \perp$ or $\sigma_1 = \perp$ then set $\sigma_0 = \perp$ and $\sigma_1 = \perp$
$\hat{b} \leftarrow \mathcal{S}^*(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{St_{issue}})$
Return 1 if $\hat{b} = b$ else return 0

Figure 3: Experiment $\mathsf{DisKeyBlind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$

We will use

$$\mathbf{Adv}^{\mathsf{Blind}}_{\mathcal{A}}(\lambda) = |2 \cdot \Pr[\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1] - 1|$$

and

$$\mathbf{Adv}^{\mathsf{DisKeyBlind}}_{\mathcal{A}}(\lambda) = |2 \cdot \Pr[\mathsf{DisKeyBlind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1] - 1|$$

to denote the adversary's $\mathcal{A}$ advantage in winning the blindness experiment in the honest key model and the malicious key model, respectively.

4. GHADAFI'S AND SMART'S BLIND SIGNATURES - REVISITED

In Section 4.1 we introduce the blind signature scheme presented by Ghadafi and Smart from [17], and recall the theorems on the security of their scheme. Then, in Section 4.2 we give the security analysis of their scheme.

4.1 Construction

The detailed construction by Ghadafi and Smart can be found in Scheme 1.

THEOREM 2 (UNFORGEABILITY [17]). *If the E-LRSW assumption holds then Scheme 1 is unforgeable. In particular, if $\mathcal{A}$ is an adversary against the unforgeability of Scheme 1, then there is an adversary $\mathcal{B}$ which solves the E-LRSW problem such that*

$$\mathbf{Adv}^{\mathsf{Unforge}}_{\mathcal{A}}(\lambda) = \mathbf{Adv}^{E-LRSW}_{\mathcal{B}}(\lambda)$$

THEOREM 3 (BLINDNESS [17]). *Scheme 1 is perfectly blind. In particular, if $\mathcal{A}$ is an adversary against the blindness of Scheme 1, then*

$$\mathbf{Adv}^{\mathsf{Blind}}_{\mathcal{A}}(\lambda) = 0$$

4.2 Security Analysis

Consider the following adversary $\mathcal{S}^*$ playing the blindness game for Scheme 1. This algorithms works in three modes $\mathsf{find}, \mathsf{issue}$ and guess defined as follows:

On input $(\mathsf{find}, \mathsf{sk_{BS}}, \mathsf{pk_{BS}})$**:**

the algorithm chooses the messages $m_0 \xleftarrow{\$} \mathcal{M}$, $m_1 \xleftarrow{\$} \mathcal{M}$ at random, sets $\mathsf{St_{find}} = (m_0, m_1, \mathsf{sk_{BS}}, \mathsf{pk_{BS}})$ and outputs $(m_0, m_1, \mathsf{St_{find}})$.

$\mathsf{Setup_{BS}}(\lambda)$**:**
generate bilinear group parameters $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow \mathsf{BGGen}(\lambda)$, choose $z \xleftarrow{\$} \mathbb{Z}_q$, compute $Z = [z]P_1$ and set $\mathcal{M} = \mathbb{Z}_q^*$. Return $\mathsf{CRS_{BS}} = (\mathsf{BG}, Z, \mathcal{M})$. We assume that $\mathsf{CRS_{BS}}$ is an argument for all algorithms.

$\mathsf{KeyGen_{BS}}(\lambda)$**:**
choose $x, y \xleftarrow{\$} \mathbb{Z}_q$, and compute $X = [x]P_2$ and $Y = [y]P_2$. Return $\mathsf{sk_{BS}} = (x, y)$ and $\mathsf{pk_{BS}} = (X, Y)$.

$\mathsf{Request_{BS}}(m, \mathsf{pk_{BS}})$**:**
compute the Pedersen commitment $Co = [m]P_1 + [r]Z$ for a random $r \xleftarrow{\$} \mathbb{Z}_q$. Set $\rho = (Co)$ and $\mathsf{St_{BS}} = (m, r)$. Return $(\rho, \mathsf{St_{BS}})$.

$\mathsf{Issue_{BS}}(\rho, \mathsf{sk_{BS}})$**:**
choose $a \xleftarrow{\$} \mathbb{Z}_q^*$, $A = [a]P_1$, $B = [a \cdot y]P_1$, $C = [a \cdot x]P_1 + [a \cdot x \cdot y]Co$, $D = [a \cdot x \cdot y]Z$. Return $\beta = (A, B, C, D)$.

$\mathsf{Unblind_{BS}}(\beta, \mathsf{St_{BS}}, \mathsf{pk_{BS}})$**:**
compute $C = C - [r]D$. If $\mathsf{Verify_{BS}}(m, (A, B, C), \mathsf{pk_{BS}}) = 0$, then return $\perp$. Choose $t \xleftarrow{\$} \mathbb{Z}_q^*$ and return $\sigma = (A, B, C) = ([t]A, [t]B, [t]C)$.

$\mathsf{Verify_{BS}}(m, \sigma, \mathsf{pk_{BS}})$**:**
if $A = 1_{\mathbb{G}_1}$ or $e(A, Y) \neq e(B, P_2)$ or $e(C, P_2) \neq e(A, X) \cdot e(B, X)^m$, then return 0. Otherwise, return 1.

Scheme 1: Ghadafi's and Smart's Blind Signature Scheme

On input $(\mathsf{issue}, \rho_0 = Co_0, \rho_1 = Co_1, \mathsf{St_{find}})$**:**

$\mathcal{S}^*$ computes two valid signatures (verifiable with $\mathsf{Verify_{BS}}$), i.e.

(A_0, B_0, C_0) where $\mathsf{Verify_{BS}}(m_0, (A_0, B_0, C_0), \mathsf{pk_{BS}}) = 1$

and

(A_1, B_1, C_1) where $\mathsf{Verify_{BS}}(m_1, (A_1, B_1, C_1), \mathsf{pk_{BS}}) = 1$.

Note that those are standard CL signatures on m_0 and m_1. Then, it chooses $b_0, b_1 \xleftarrow{\$} \mathbb{Z}_q$, $D_0 = [b_0]Z$, $D_1 = [b_1]Z$ and updates

$$C_0' = C_0 + [b_0]Co_0 - [m_0 \cdot b_0]P_1,$$

$$C_1' = C_1 + [b_1]Co_1 - [m_1 \cdot b_1]P_1.$$

It sets $\beta_0 = (A_0, B_0, C_0', D_0)$, $\beta_1 = (A_1, B_1, C_1', D_1)$, $\mathsf{St_{issue}} = ()$ and outputs $(\beta_0, \beta_1, \mathsf{St_{issue}})$.

On input $(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{St_{issue}})$**:**

outputs 1 if $(\sigma_0, \sigma_1) = (\perp, \perp)$ and 0 otherwise.

LEMMA 1. *Let $\mathcal{S}^*$ be a PPT adversary against blindness game of Scheme 1. If $\mathcal{S}^*$ works as described above, then we have:*

$$\mathbf{Adv}^{\mathsf{Blind}}_{\mathcal{S}^*}(\lambda) = 1$$

PROOF. Note that only if $b = 0$ the tuples β_0 and β_1 will yield valid CL signatures, i.e. in case of β_0 the user will compute $C_0'' = C_0' - [r_b]D_0 = C_0 + [b_0]Co_0 - [m_0 \cdot b_0]P_1 - [r_b]([b_0]Z)$ and $C_0'' = C_0$ only if $r_b = r_0$, which implies that $b = 0$ (otherwise, (A_0, B_0, C_0'') is not a valid CL signature). □

COROLLARY 1. *Theorem 3 must be false.*

REMARK 1. *Note that $\mathcal{S}^*$ uses an alternative signing algorithm, i.e. knowing the secret keys x, y, z, the input m and output $Co = [m]P_1 + [r]Z$ of the $\mathsf{Request}_{\mathsf{BS}}$ algorithm, one can first compute a valid CL signature (A, B, C) on m and return $\beta = (A, B, C + [b]Co - [m \cdot b]P_1, [b]Z)$, for a random $b \in \mathbb{Z}_q \backslash \{0\}$. For the returned tuple β and $\mathsf{St}_{\mathsf{BS}} = (m, r)$, the $\mathsf{Unblind}_{\mathsf{BS}}(\beta, \mathsf{St}_{\mathsf{BS}}, \mathsf{pk}_{\mathsf{BS}})$ algorithm will return a signature $\sigma \neq \perp$.*

REMARK 2. *We notice that this attack works in the honestly generated public key model. In particular, it does not require that the public key is generated maliciously. Thus, the attack applies to the scheme in [17].*

4.2.1 Source and Consequences of the Problem.

We will now take a look at the proof of blindness presented for this scheme. The core idea of the security proof is the following part: *"We let E denote the event that the* guess *stage of the adversary is passed the pair $(\sigma_0, \sigma_1) = (\perp, \perp)$. We clearly have*

$$\Pr[\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1] \leq \Pr[E] \cdot \Pr[\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1|E] + \Pr[\neg E] \cdot \Pr[\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1|\neg E].\text{"}$$

Using this observation the authors show that

$$\Pr[\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1|E] = 1/2$$

because of the perfect hiding property of Pedersen commitments. Note that in this case the adversary only receives commitments Co_0 and Co_1. In case of event $\neg E$, the reduction uses the known signing key (since this proof is in the honestly-generated key model) and signs messages m_0 and m_1 by itself and omits (β_0, β_1). Again, because of the perfect hiding property of Pedersen commitments

$$\Pr[\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1|\neg E] = 1/2.$$

Combining these results we have:

$$\Pr[\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = 1] \leq \Pr[E] \cdot 1/2 + \Pr[\neg E] \cdot 1/2 = 1/2.$$

This of course is the result stated in Theorem 3, which we showed is false.

We argue that this idea can only be used in case the malicious signer knows which event, i.e. E or $\neg E$ will occur for its returned values β_0, β_1. This could for example be checked using a signature-derivation check. The formal definition can be found in [10]. Informally, such function SDCh takes as input the signer's public key $\mathsf{pk}_{\mathsf{BS}}$, a transcript of communication (ρ, β) and outputs 0 if and only if algorithm $\mathsf{Unblind}_{\mathsf{BS}}$ will return $\perp$ and not a valid signature in this communication.

Let us look at the $\mathsf{Unblind}_{\mathsf{BS}}$ algorithm. The user receives the tuple (A, B, C, D), transforms it into a valid (or not) CL signature and verifies if the transformed signature is valid on the given message. However, the consistency of (A, B, C, D) is not verified (which could be done for example with SDCh). We will now show that for the considered blind signatures such a SDCh does not exist.

LEMMA 2. *There exists no signature derivation check* SDCh *for the above blind signature scheme.*

PROOF. We will show that, if such a signature derivation check exists, then we can construct an algorithm $\mathcal{R}$ for which $\mathbf{Adv}^{\mathsf{Hiding}}_{\mathcal{R}}(\lambda) = 1$.

First, the reduction $\mathcal{R}$ sets up the blind signature CRS and signing key, i.e. it runs

$$\mathsf{CRS}_{\mathsf{BS}} = (\mathsf{BG}, Z, \mathcal{M}) \leftarrow \mathsf{Setup}_{\mathsf{BS}}(\lambda)$$

and

$$(\mathsf{sk}_{\mathsf{BS}} = (x, y), \mathsf{pk}_{\mathsf{BS}} = (X, Y)) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(\lambda).$$

Then the reduction $\mathcal{R}$ chooses two messages $m_0, m_1 \xleftarrow{\$} \mathcal{M}$, sets $\mathsf{cpp} = (\mathsf{BG}, Z)$ and outputs (m_0, m_1, cpp) to the challenger in the hiding experiment. As a result $\mathcal{R}$ receives Co_0, Co_1, sets $\rho = Co_0$ and computes $\beta = (A, B, C, D)$, where $A = [a]P_1$, $B = [y]A$, $C = [a \cdot x]P_1 + [a \cdot x \cdot y]Co_0 + [b]Co_0 - [m_1 \cdot b]P_1$, $D = [r]Z$ and $a, r \xleftarrow{\$} \mathbb{Z}_q$. Finally, $\mathcal{R}$ returns the output of $\mathsf{SDCh}(\mathsf{pk}_{\mathsf{BS}}, \rho, \beta)$. □

REMARK 3. *Note that, in the proof given above, the reduction $\mathcal{R}$ knows the signer's secret key. It follows that even a malicious signer cannot know which event will occur for the returned values β_0, β_1.*

This in fact is the source of the problem. The adversary can use the alternative signing algorithm used in the proof and described in Remark 1. In other words, this means that the adversary can guess the signed message and use the user as an oracle and verify his guess (i.e. the guess is correct only if the user yields a valid signature).

We have already shown that this problem allows the adversary to win the blindness game for Scheme 1. We now describe how this problem relates to real-world applications. We will focus on two popular use cases for blind signatures, namely eCash and eVoting systems.

Let us first consider eCash systems as in [8]. Typically, such systems work as follows. In the mint transaction the user chooses a random coin identifier and the bank uses the blind signature scheme to sign this identifier. The user may then spend the coin by revealing the signature and the corresponding identifier. Double spending is prevented due to registering the coin identifier by the bank. Here the blindness property guaranties that the bank cannot link the spending transaction with the mint transaction. Note, that our attack cannot be directly applied to this scenario since the message space for coin identifiers must be large to prevent double spending.

Now let us consider eVoting schemes which utilize blind signature schemes to sign ballots as in [13] or [23]. In such schemes, voters receive a blind signature from an administration authority on ballots. Voters may then presents the ballot and the signature to a counting authority. The number of choices in a election is a reasonably small set. Let us here assume that a voter may choose "Yes" and "No". Moreover, assume that the user's client retries communication in

case of an error (e.g. because the signature is invalid). The administration authority can guess the signed message and wait if the same IP (or user) connects again. This way the administration authority may distinguish the signed message i.e. what was the user's vote.

REMARK 4. *To use the alternative signing algorithm, we have to know the message committed in Co. In the blindness game the adversary receives Co_0, Co_1 and knows that one of them corresponds to message m_0 and one to m_1. Note that there are only two possible associations and using the input in the* guess *mode, the adversary can cross out one of them (if he receives $(\perp, \perp)$). This obviously makes the other one correct.*

5. OUR BLIND SIGNATURES

5.1 Construction

Our construction is built on top of the blind signature scheme from [17] and requires pairings of type 3. However, we extend it using a modified unblind and verification procedure. The idea is to include a new element $W = [z]X$ into the signer's public key. This new element can be used by the user, who is given a pre-signature (A', B', C', D'), to check that (A', B', C') is a valid signature on $m + z \cdot s$ and $D' = [z \cdot x]B'$. In fact, the user checks that the signature is on $\log_{P_1} Co$. It follows, that given a different message m' and opening s' such that $m+z\cdot s = m'+z\cdot s'$, an unbounded user could also derive a signature on message m'. Note that this is not the case for the original scheme from [17], where the user computes the actual signature using the element D' and then checks if the resulting signature is on the message m. Finally, we use a deterministic bilinear group generator BGGen to get rid of the common reference string. As a result we obtain a scheme that is two-move, setup-free and perfectly blind in the malicious key model. Details on the construction are given in Scheme 2.

$\mathsf{KeyGen}_{\mathsf{BS}}(\lambda)$:
compute bilinear group parameters $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow \mathsf{BGGen}(\lambda)$ with type 3 pairings, $x, y, z, w \xleftarrow{\$} \mathbb{Z}_q \backslash \{0\}$, $X = [x]P_2$ and $Y = [y]P_2$, $Z = [z]P_1$, $W = [z]X$. Return $\mathsf{sk}_{\mathsf{BS}} = (x, y, z)$ and $\mathsf{pk}_{\mathsf{BS}} = (\lambda, X, Y, Z, W)$.

$\mathsf{Request}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}})$:
compute $\mathsf{BG} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow \mathsf{BGGen}(\lambda)$, compute the Pedersen commitment $Co = [m]P_1 + [s]Z$ for $s \xleftarrow{\$} \mathbb{Z}_q \backslash \{0\}$. Set $\rho = (Co)$ and $\mathsf{St}_{\mathsf{BS}} = (m, s)$. Return $(\rho, \mathsf{St}_{\mathsf{BS}})$.

$\mathsf{Issue}_{\mathsf{BS}}(\rho, \mathsf{sk}_{\mathsf{BS}})$:
choose $a \xleftarrow{\$} \mathbb{Z}_q^*$, $A' = [a]P_1$, $B' = [a \cdot y]P_1$, $C' = [a \cdot x]P_1 + [a \cdot x \cdot y]Co$, $D' = [a \cdot x \cdot y]Z$. Return $\beta = (A', B', C', D')$.

$\mathsf{Unblind}_{\mathsf{BS}}(\beta, \mathsf{St}_{\mathsf{BS}}, \mathsf{pk}_{\mathsf{BS}})$:
return $\perp$ if $A' = 1_{\mathbb{G}_1}$ or $Z = 1_{\mathbb{G}_1}$ or $e(A', Y) \neq e(B', P_2)$ or $e(C', P_2) \neq e(A', X) \cdot e(B', X)^m \cdot e(B', W)^s$ or $e(Z, X) \neq e(P_1, W)$ or $e(B', W) \neq e(D', P_2)$. Choose $t \xleftarrow{\$} \mathbb{Z}_q^*$ and return $\sigma = (A, B, C) = ([t]A', [t]B', [t](C' - [s]D'))$.

$\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}})$:
if $A = 1_{\mathbb{G}_1}$ or $e(A, Y) \neq e(B, P_2)$ or $e(C, P_2) \neq e(A, X) \cdot e(B, X)^m$, then return 0. Otherwise, return 1.

Scheme 2: Our Blind Signature Scheme

5.2 Security

THEOREM 4 (UNFORGEABILITY). *If the E'-LRSW assumption holds for* BGGen *then Scheme 2 is unforgeable.*

PROOF. Let $\mathcal{A}$ be an adversary against the unforgeability of Scheme 2. We shall use $\mathcal{A}$ to construct an algorithm $\mathcal{R}$ which solves the E'-LRSW problem. Let $(\mathsf{BG}, \hat{X}, \hat{Y}, \hat{Z}, \hat{W})$ be an instance of the E'-LRSW problem for a security parameter λ. Algorithm $\mathcal{R}$ sets the public key $\mathsf{pk}_{\mathsf{BS}} = (\lambda, \hat{X}, \hat{Y}, \hat{Z}, \hat{W})$. Now $\mathcal{R}$ executes the algorithm $\mathcal{A}$, which at some point will make k queries to its signing oracle $\mathsf{Issue}_{\mathsf{BS}}$. The algorithm $\mathcal{R}$ responds to a query on $\rho = (Co)$ by passing $Co = [t]P_1$ to its $O_{\hat{X},\hat{Y},\hat{Z},\hat{W}}$ oracle. Note that t is unknown to $\mathcal{R}$, as it is equal to $(m+s\cdot z)$ for some m, s chosen by $\mathcal{A}$. In the process $\mathcal{R}$ receives a tuple $(A, [y]A, [x+t\cdot x\cdot y]A, [x\cdot y\cdot z]A)$, which is then passed back to $\mathcal{A}$. Eventually, $\mathcal{A}$ will output $k+1$ tuples $(m_i, (A_i, B_i, C_i))$ where (m_i, A_i, B_i, C_i) are valid CL signatures. By returning this list, $\mathcal{R}$ solves the given E'-LRSW problem instance. □

THEOREM 5 (BLINDNESS). *Scheme 2 is perfectly blind in the dishonest-key model.*

PROOF. Let $\beta = (A', B', C', D')$ be the pre-signature sent by the signer after receiving a commitment $Co = [m]P_1 + [s]Z$. We begin the proof with the simple observation that the verifications performed in the $\mathsf{Unblind}_{\mathsf{BS}}$ algorithm check whether:

1. (A', B', C') is a valid CL signature on message $\hat{m} = m + s \cdot z$, this in particular means that:
   - $B' = [y]A'$,
   - $C' = [x + x \cdot y(m + s \cdot z)]A'$,
2. $D' = [x \cdot y \cdot z]A'$.

To see that this is true, let us take a look at the verifications performed. $\mathsf{Unblind}_{\mathsf{BS}}$ returns $\perp$ if:

$$A' = 1_{\mathbb{G}_1} \tag{1}$$
$$e(A', Y) \neq e(B', P_2) \tag{2}$$
$$e(C', P_2) \neq e(A', X) \cdot e(B', X)^m \cdot e(B', W)^s \tag{3}$$
$$e(Z, X) \neq e(P_1, W) \tag{4}$$
$$e(B', W) \neq e(D', P_2). \tag{5}$$

If any of these equations don't hold, then the adversary obtains $\perp$ in the guess stage and his advantage of winning the game is 0. Let us now assume, these equations hold. From the equation 1 we have that $A' = [a]P_1$ for some $a \in \mathbb{Z}_q^*$. Then, from 2 we have that $B' = [y]A'$ and from

$e(Z, X) = e(P_1, W)$ we have $W = [z \cdot x]P_2$. Now, from 3 we can write

$$\begin{aligned} e(C', P_2) =& e(A', X) \cdot e(B', X)^m \cdot e(B', W)^s \\ =& e(A', X) \cdot e(B', X)^{m+z \cdot s}, \end{aligned}$$

and what follows $C' = [x + \hat{m} \cdot x \cdot y]A'$ for $\hat{m} = m + z \cdot s$. Thus, the triple $(A',\ B',\ C')$ forms a CL signature on the message $\hat{m}$. Moreover, since $Co = [m+z \cdot s]P_1$, we have that $C' = [a \cdot x \cdot y]Co$. Finally, from the equation 5, we know that $D' = [y \cdot z \cdot x]A$, thus $C = C' - [s]D' = [x + m \cdot x \cdot y]A$ and the triple $(A, B, C) = (A', B', C)$ form a valid CL signature on m.

Since, we know that $C' = [a \cdot x \cdot y]Co$ and Co is a Pedersen commitment, the final value C could be computed as $C = C' - [s']D'$, and in effect yield a valid CL signature on a message $m' = (m+z(s-s'))$, i.e. any message in the message space. Obviously this does not mean we can compute this message.

In contrast to [17] we might further assume that all values (i.e. A', B', C', D') returned by the adversary are of the form as described above. This is motivated by the fact that the $\mathsf{Unblind}_{\mathsf{BS}}$ algorithm first verifies all these values before performing any other action and, as noticed above, the adversary at the guess stage obtains $\perp$ for both signatures in case he outputs incorrect signatures. This is not the case for the scheme in [17] as the $\mathsf{Unblind}_{\mathsf{BS}}$ algorithm first updates C before performing any verification and there is only a consistency check for the final signature which additionally omits the correctness of the intermediate value D. In particular the checks do not imply that $C' = [a \cdot x \cdot y]Co$.

Now, the crucial observation is the fact that the partial signatures may be unblinded to any message and that the CL signatures might be randomized into any signature in the signature space. Let $Co_0 = [m_0]P_1 + [s_0]Z$ and $Co_1 = [m_1]P_1 + [s_1]Z$ denote the Pedersen commitment used in the blindness game and let (A_0, B_0, C_0) and (A_1, B_1, C_1) be the signatures returned by the unblind procedures. As showed above, we may always find s_0' and s_1' such that $Co_0 = [m_1]P_1 + [s_0']Z$ and $Co_1 = [m_0]P_1 + [s_1']Z$, thus it may be that $C_0 = [x \cdot y \cdot m_0]A_0$ and $C_1 = [x \cdot y \cdot m_1]A_1$, but, from the adversary's view, it is equally probable that, $C_0 = [x \cdot y \cdot m_1]A_0$ and $C_1 = [x \cdot y \cdot m_1]A_1$. Then, from the randomization property of CL signatures we have that the CL signature itself gives the adversary no useful information. To sum up, we have that the CL signature gives no information and the pre-signatures could be unblinded to any message (in particular m_0 or m_1), therefore the adversary can only guess the bit in the blindness game and, what follows, his advantage in breaking the blindness property is equal to 0. □

5.3 Efficiency of Our Constructions.

In Figure 4 we compare our construction to the recent results given by Fuchsbauer et al. at Crypto'15 [12] and Garg et al. at Eurocrypt'14 [15], as they are the most efficient schemes from standard assumptions.

We base our results on the calculations from [12]. The assumption is that the blind signatures by Fuchsbauer et al. are based on SPS-EQ from [11] and the construction by Garg et al. is instantiated using the DLIN assumption.

It is easy to see from Figure 4 that Scheme 2 has the lowest communication complexity. Moreover, the signature size of our construction is 50% shorter than of the best previous scheme.

Overall, our construction is comparable, in terms of efficiency, with the previous best scheme proposed by Fuchsbauer et al. at Crypto'15. However, their instantiation requires interactive assumptions for both unforgeability and blindness. Our scheme is perfectly blind but requires an interactive assumption for unforgeability.

One may think that this efficiency comes at a price of security, since we use a different interactive assumption than in [12]. Obviously, we do not have to worry about blindness, since we do not depend on any assumption. From the original paper on structure-preserving signatures on equivalence classes [11], we learn that an adversary can output a forged signature in the GGM with probability $O(\frac{q_O^3}{q})$, where the adversary can issue $O(q_O)$ queries to group oracles and ask for q_O signatures. This means that the same probability holds for the blind signature scheme in [12]. The unforgeability of our scheme is based on the E'-LRSW assumption for which the probability of a forgery in the GGM is $O(n^2 \cdot q_O/q)$, where $n = 6 + q_{\mathbb{G}} + 4q_O + q_e$. Thus, we may conclude that both schemes ensure a similar level of security in terms of unforgeability.

6. ANALYSIS OF THE E'-LRSW ASSUMPTION

In this section we give the proof of Theorem 1.

Proof. The difference between the E-LRSW and the E'-LRSW assumptions is the additional element $[z]X \in \mathbb{G}_2$ given as part of the problem. Thus, both assumptions are closely related. What is more, the intuition behind this proof is that the additional element in $\mathbb{G}_2$ does not help the adversary to break the assumption since the adversary must output only values in $\mathbb{G}_1$.

To prove that the E'-LRSW assumption holds in the GGM, we use the same reasoning that Ghadafi and Smart used in [17] to prove that the E-LRSW assumption holds in the GGM. In particular, the proof will only differ in the initialization step in which we include the element $[z]X$ on the list of elements in $\mathbb{G}_2$, the bound n on the number of non-constant polynomials and the maximum degree of all polynomials d_{max}. However, for a sense of completeness we will now rephrase the original proof and by applying the same reasoning, show that the E'-LRSW assumption holds in the GGM.

We begin the proof by defining three lists G_1, G_2, G_T of pairs (σ, P) kept by the challenger. The value σ corresponds to a "random" encoding of the group element chosen from some set S with $|S| > 3 \cdot q$ and P is some polynomial in $\mathbb{F}_q[X, Y, Z, A_1, \ldots, A_{q_O}]$. Similar to [17] we allow the adversary to introduce new elements only using the algorithms defined below. However, this does not change the arguments below but simplifies the reasoning.

We also introduce an $\mathsf{Update}(G, P)$ operation that searches the list G for an element (σ, P). If such element is found then this operation returns σ. Otherwise a new element σ is chosen from S in such a way that it is distinct from all other element from S used so far. Finally, the operation adds (σ, P) to list G and returns σ. The values σ are used by the adversary as handles to group elements.

The three lists are initialized by executing the below six operations:

Construction	Public key size	Communication	Signature size*	
[15]	$43\mathbb{G}_1$	$18\log_2 q + 41\mathbb{G}_1$	$183\mathbb{G}_1$	46848 bits
Scheme 3 [12]	$1\mathbb{G}_1 + 3\mathbb{G}_2$	$4\mathbb{G}_1 + 1\mathbb{G}_2$	$4\mathbb{G}_1 + 1\mathbb{G}_2$	1536 bits
Scheme 2	$1\mathbb{G}_1 + 3\mathbb{G}_2$	$5\mathbb{G}_1$	$3\mathbb{G}_1$	768 bits

* At a 256-bit (resp. 512-bit) representation of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$)

Figure 4: Comparison of (Partially) Blind Signature Schemes

- $\mathsf{Update}(G_1, 1)$,
- $\mathsf{Update}(G_2, 1)$,
- $\mathsf{Update}(G_1, Z)$,
- $\mathsf{Update}(G_2, X)$,
- $\mathsf{Update}(G_2, Y)$,
- $\mathsf{Update}(G_2, X \cdot Z)$.

The adversary is now allowed to perform the following operations:

Group Operations: The adversary receives access to three oracles O_1, O_2, O_T. By calling $O_i(\sigma_1, \sigma_2)$ the challenger searches list G_i for (σ_1, P_1) and (σ_2, P_2). If two such pairs does not exist, the challenger returns $\perp$. Otherwise, he returns $\mathsf{Update}(G_i, P_1 - P_2)$ to the adversary. Note that using this subtraction operation we can define the identity element O_{G_i} as $O_i(\sigma, \sigma)$, compute the negation $-\sigma$ as $O_i(O_{G_i}, \sigma)$ and addition as $O_i(\sigma_1, -\sigma_2)$.

Pairing Operation: Additionally, the adversary is given access to a pairing oracle O_e. By calling $O_e(\sigma_1, \sigma_2)$ the challenger searches list G_1 for (σ_1, P_1) and list G_2 for (σ_2, P_2). Again, if such pairs does not exist, the challenger returns $\perp$. Otherwise, he returns $\mathsf{Update}(G_T, P_1 \cdot P_2)$.

E'-LRSW Oracle: Finally, the adversary is allowed to make up to q_O queries to the E'-LRSW oracle $O^{E'}(\sigma)$. On the i-th call of oracle $O^{E'}(\sigma_i)$, the challenger returns $\perp$ if there exists no pair (σ, P) on list G_1. Otherwise, the challenger computes $\sigma_A \leftarrow \mathsf{Update}(G_1, A_i)$, $\sigma_B \leftarrow (G_1, A_i \cdot Y)$, $\sigma_C \leftarrow \mathsf{Update}(G_1, A_i \cdot X \cdot (1 + Y \cdot P))$, $\sigma_D \leftarrow \mathsf{Update}(G_1, A_i \cdot X \cdot Y \cdot Z)$, where A_i, X, Y and Z are the indeterminants introduced above. Finally, the challenger returns the tuple $(\sigma_A, \sigma_B, \sigma_C, \sigma_D)$ to the adversary.

We will now show that the probability that an adversary solves the E'-LRSW problem in the GGM is negligibly small. Let us assume by contradiction that there exists a successful adversary $\mathcal{A}$ that solves the problem. $\mathcal{A}$ will eventually output a set of values

$$\{m_i, \sigma_A^{(i)}, \sigma_B^{(i)}, \sigma_C^{(i)}\}_{i=1}^{q_O+1}$$

where $m_i \in \mathbb{F}_q \backslash \{0\}$ are distinct and $\sigma_A^{(i)}, \sigma_B^{(i)}, \sigma_C^{(i)}$ are handles to group elements in $\mathbb{G}_1$. In order to solve the problem, there must exist elements $(\sigma_A^{(i)}, P_A^{(i)})$, $(\sigma_B^{(i)}, P_B^{(i)})$, $(\sigma_C^{(i)}, P_C^{(i)})$ in list G_1. Moreover, as solution to the E'-LRSW problem, for some assignment $(x, y, z, a_1, \ldots, a_{q_O}) \in \mathbb{F}_q^{3+q_O}$ to the variables $(X, Y, Z, A_1, \ldots, A_{q_O})$ the following two equations must hold:

$$(Y \cdot P_A^{(i)} - P_B^{(i)})(x, y, z, a_1, \ldots, a_{q_O}) = 0, \tag{6}$$

$$(X \cdot P_A^{(i)} + X \cdot m_i \cdot P_B^{(i)} - P_C^{(i)})(x, y, z, a_1, \ldots, a_{q_O}) = 0. \tag{7}$$

In most GGM proofs it is obvious that the resulting equations are not identically zero, i.e. satisfied for all assignments. As already noted by Ghadafi and Smart in [17] for this kind of problems we have to divide the proof into three cases that can be shown separately.

Case 1: equations 6 and 7 are not identically zero,

Case 2: the adversary cannot distinguish if for a specific assignment to the variables it is interacting with genuine group oracles,

Case 3: non-identically zero equations 6 and 7 are satisfied by a specific assignment.

Before we examinate each case, we notice that the total number of non-constant polynomials on lists G_1, G_2, G_T is bounded from above by $n = 6 + q_{\mathbb{G}} + 4 \cdot q_O + q_e$.

Case 1

The first point is to notice that neither $P_A^{(i)}$ nor $P_B^{(i)} = Y \cdot P_A^{(i)}$ can involve any terms in X. This follows directly from the definitions of the used oracles. The proof follows from induction. The term X is not used on list G_1 in the initialization phase, so clearly this holds $j = 0$, where j is the number of queries made by the adversary to oracle $O^{E'}$. Now assume that this holds up to an arbitrary j. When making the $j+1$-the query, the adversary has been able to produce linear combinations of the polynomials available before the j-th query, and those obtained from the j-th query. This is possible because only oracle $O^{E'}$ allows non-linear combinations in $\mathbb{G}_1$. However, the j-th query, with further linear combinations, cannot result in two polynomials with $\deg_X(P_A^{(i)}) \geq 1$ and $P_B^{(i)} = Y \cdot P_A^{(i)}$. The same can be applied to the term Z. Thus, we can conclude that $\deg_Z(P_A^i) = 0$.

The above reasoning and results ensure that we must have:

$$P_A^{(i)} = \sum_{j=1}^{q_O} r_{i,j} \cdot A_j \text{ and } P_B^{(i)} = \sum_{j=1}^{q_O} r_{i,j} \cdot Y \cdot A_j,$$

for some integers $r_{i,j}$. This also means that:

$$P_C^{(i)} = \sum_{j=1}^{q_O} r_{i,j} \cdot A_j \cdot X \cdot (1 + m_i \cdot Y). \tag{8}$$

To produce a polynomial $P_C^{(i)}$, which only has a single power of X, one would have to add together multiplies of the polynomials σ_C and σ_D outputted by the $O^{E'}$ oracle. Thus, we have values $s_{i,j}, t_{i,j}, f_j, f_j$ such that:

$$P_C^{(i)} = \sum_{j=1}^{q_O} (s_{i,j} \cdot A_j \cdot X \cdot (1 + f_j \cdot Y + g_j \cdot Y \cdot Z) + t_{i,j} \cdot A_j \cdot X \cdot Y \cdot Z). \quad (9)$$

From equations 8 and 9 we must have

$$r_{i,j} = s_{i,j}, \quad r_{i,j} \cdot m_i = s_{i,j} \cdot f_j, \quad 0 = s_{i,j} \cdot g_j + t_{i,j},$$

for all values $1 \leq i \leq q_O + 1$ and $1 \leq j \leq q_O$. Note now that in order to be a valid solution, for a given i and one j we must have $s_{i,j} \neq 0$. Otherwise, this would imply that $P_A^{(i)} = 0$, which is disallowed (the value A must be in $\mathbb{G}_1 \backslash \{1_{\mathbb{G}_1}\}$). What is more, this means that $m_i = f_j$ and that there must exist two values i_0 and i_1 such that $m_{i_0} = m_{i_1}$. This obviously contradicts the fact that all the messages should be distinct. Finally, we conclude that the output of the adversary cannot correspond to a set of polynomial equations which hold for all assignments.

Case 2
To learn that its interacting with non-genuine oracles, the adversary must produce two different polynomials P_1 and P_2, such that $P_1 = P_2$ for a specific assignment. First we notice that by the construction of the oracles, there cannot exist identical P_1 and P_2. Moreover, since the maximum degree of polynomials we can get is d_{max}, the probability of an assignment being such that $P_1 = P_2$ is bounded by d_{max}/q. Thus, the overall probability that this happens is bounded by $O(n^2 \cdot d_{max}/q)$ (for each pair on lists of maximal n elements). Therefore, we have $O(n^2 \cdot q_O/q)$ as d_{max} is bounded by a multiply of q_O.

Case 3
We now show that the probability of the adversary outputting a set of elements, which satisfy the required equations for a specific assignment is negligibly small. First, we notice that the total degree of the set of equation and the number of equations that must be satisfied are of order q_O. Thus, the probability is bounded by $O((q_O/q)^{q_O})$. Which is negligible.

It is easy to see that the probability of an adversary breaking the E'-LSRW assumption in the GGM is bounded by $O(n^2 \cdot q_O/q)$. □

7. CONCLUSIONS

In this paper we proposed two important results. First, we have shown that the two-move blind signature scheme proposed by Ghadafi and Smart [17] is not blind. Then, we fixed this issue and proposed a new construction. Our solution remains two-move but requires no common reference string and is perfectly blind in the malicious-key model. Moreover, our construction is the most efficient, in terms of communication complexity and signature size, in comparison with existing setup-free and two-move schemes in the standard model.

To sum up, our construction is efficient. However, unforgeability of our solution is based on an interactive assumption called E'-LRSW. Although we presented a proof that this assumption holds in the generic group model, this may not provide satisfiable evidence that this problem is hard. This issue limits the practicality of the solution.

Acknowledgments

This material is based on work supported by the National Research Center under grant OPUS no 2014/15/B/ST6/02837.

8. REFERENCES

[1] M. Abdalla, C. Namprempre, and G. Neven. On the (Im)possibility of Blind Message Authentication Codes. In D. Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2006.

[2] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In T. Rabin, editor, *Advances in Cryptology âĂŞ CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 209–236. Springer Berlin Heidelberg, 2010.

[3] P. S. L. M. Barreto and M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In B. Preneel and S. E. Tavares, editors, *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005.

[4] Bellare, Namprempre, Pointcheval, and Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. *Journal of Cryptology*, 16(3):185–215, 2003.

[5] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Crypto 2004, LNCS 3152*, pages 56 – 72. Springer Verlag, 2004.

[6] D. Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 199–203. Plenum, 1982.

[7] D. Chaum. Blind Signature System. In D. Chaum, editor, *Advances in Cryptology*, pages 153–153. Springer US, 1984.

[8] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88: Proceedings*, pages 319–327, New York, NY, 1990. Springer New York.

[9] M. Fischlin. Round-Optimal Composable Blind Signatures in the Common Reference String Model. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77. Springer Berlin Heidelberg, 2006.

[10] M. Fischlin and D. Schröder. On the Impossibility of Three-Move Blind Signature Schemes. In H. Gilbert, editor, *Advances in Cryptology âĂŞ EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 197–215. Springer Berlin Heidelberg, 2010.

[11] G. Fuchsbauer, C. Hanser, and D. Slamanig. EUF-CMA-Secure Structure-Preserving Signatures on Equivalence Classes. Cryptology ePrint Archive, Report 2014/944, 2014. http://eprint.iacr.org/.

[12] G. Fuchsbauer, C. Hanser, and D. Slamanig. Practical Round-Optimal Blind Signatures in the Standard

Model. In R. Gennaro and M. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 233–253, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[13] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology — AUSCRYPT '92: Workshop on the Theory and Application of Cryptographic Techniques Gold Coast, Queensland, Australia, December 13–16, 1992 Proceedings*, pages 244–251, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[14] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Appl. Math.*, 156(16):3113–3121, Sept. 2008.

[15] S. Garg and D. Gupta. Efficient Round Optimal Blind Signatures. In P. Nguyen and E. Oswald, editors, *Advances in Cryptology âĂŞ EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 477–495. Springer Berlin Heidelberg, 2014.

[16] S. Garg, V. Rao, A. Sahai, D. SchrÃűder, and D. Unruh. Round Optimal Blind Signatures. In P. Rogaway, editor, *Advances in Cryptology âĂŞ CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 630–648. Springer Berlin Heidelberg, 2011.

[17] E. Ghadafi and N. Smart. Efficient Two-Move Blind Signatures in the Common Reference String Model. In D. Gollmann and F. Freiling, editors, *Information Security*, volume 7483 of *Lecture Notes in Computer Science*, pages 274–289. Springer Berlin Heidelberg, 2012.

[18] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. Smart, editor, *Advances in Cryptology âĂŞ EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer Berlin Heidelberg, 2008.

[19] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym Systems. In H. Heys and C. Adams, editors, *Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*. Springer Verlag, 1999.

[20] U. Maurer. Abstract models of computation in cryptography. In N. P. Smart, editor, *Cryptography and Coding: 10th IMA International Conference, Cirencester, UK, December 19-21, 2005. Proceedings*, pages 1–12, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[21] A. Miyaji, M. Nakabayashi, and S. Takano. 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.

[22] Multos Consortium. MULTOS Developer's Guide. MAO-DOC-TEC-005 v1.40, 2015. https://www.multos.com/uploads/MDG.pdf.

[23] M. Ohkubo, F. Miura, M. Abe, A. Fujioka, and T. Okamoto. An improvement on a practical secret voting scheme. In *Information Security: Second InternationalWorkshop, ISW'99 Kuala Lumpur, Malaysia, November 6-7, 1999 Proceedings*, pages 225–234, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[24] Oracle. Java Card Technology. http://www.oracle.com/technetwork/java/embedded/javacard/overview/index.html.