Efficient Two-Move Blind Signatures in the Common Reference String Model

E. Ghadafi and N.P. Smart

Dept. Computer Science,
University of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB,
United Kingdom.
{ghadafi,nigel}@cs.bris.ac.uk

Abstract. Blind signatures provide a mechanism for achieving privacy and anonymity whereby a user gets the signer to sign a message of his choice without the signer learning the content of the message, nor linking message/signature request pairs when he sees the final signature. In this paper, we construct a blind signature that requires minimal interaction (two rounds) between the user and the signer. The signature request protocol is akin to the classic, blind-unblind methodology used for RSA blind signatures in the random oracle model. The output signature is a standard Camenisch-Lysyanskaya signature in pairing groups. The scheme is secure in the common reference string model, assuming two discrete logarithm related assumptions in bilinear groups; namely a new variant of the LRSW assumption and the SXDH problem. We provide evidence for the hardness of our new variant of the LRSW by showing it is intractable in the generic group model.

1 Introduction

Background: Since their introduction by Chaum [11] blind signatures have been used in a number of cryptographic applications that require one party (a signer) to authenticate a message for another party (the user), whilst maintaining privacy of the user's message. The classic example of their use is in e-cash protocols in which the signer acts as a bank and the message is a representation of digital cash; the privacy requirement coming from the non-traceability requirement of cash.

A blind signature must satisfy two security requirements, blindness and unforgeability. By blindness we mean that the signer can not learn what message he has signed and in addition, when he later sees the message at the verification process, he cannot link it to its corresponding signature request. Unforgeability on the other hand guarantees that the user cannot output any new signatures that he has not asked the signer to sign for him, or in other words, the number of signatures the user can compute is equal to the number of completed interactions he was involved in with the signer.

Since their introduction a number of authors have presented blind signature algorithms based on different assumptions and in different models. For example, schemes based on factoring related assumptions have been given in the random oracle model (ROM) [3], and in the common reference string (CRS) model [9]; schemes based on discrete logarithm related assumptions have been given in the ROM [1, 5, 26] and in the CRS model [2, 14, 25]; schemes based on a combination of discrete logarithm and factoring based assumption have been given in the CRS model [19]; finally in [13, 18] schemes in the CRS are given under general assumptions.

A blind signature consists of two distinct phases. In the first phase, which we shall call the signature request phase, the user obtains from the signer the signature on the message he requires. In the second phase, the signature and message are made public and anyone can apply the public verification algorithm to verify the message/signature pair. The signature request phase is the most complex of all phases. One could consider such a phase as a general secure two-party computation, where the user has the message as his private input and the signer has a secret key as his private input. After such a secure computation, the user outputs a valid blind signature on his secret message.

Lindell [20, 21] shows that, in the standard model, the number of rounds in the signature request phase is bounded from below by the number of concurrent executions of the signature request phase one allows. Such a lower bound can be circumvented by making a stronger assumption, for example in the ROM one can obtain two-move signature request protocols under the one-more RSA [3] and the one-more discrete logarithm [5] assumptions.

Prior work in the CRS: Due to the round optimal nature of a two-move signature request phase, and the desire to avoid the use of the random oracle, much recent work has focused on developing round optimal blind signatures in the CRS model.

In [13] Fischlin presented a scheme in the CRS which has a two move signature request protocol. The scheme is a generic construction from basic primitives, namely schemes for commitment, encryption and signatures as well as generic non-interactive zero knowledge (NIZK) proofs for NP-languages. The signature request protocol consists of the user sending a commitment to the message to the signer, who responds with a signature on the commitment. The user then uses this signature on the commitment to construct the blind signature, by first encrypting the commitment and the signature, and then adding a NIZK proof that the encrypted signature is a valid signature on the encrypted commitment, and that the encrypted commitment is a commitment to the specific message.

Using the notion of automorphic signatures Fuchsbauer [14] (see also [2]), presents a variant of the construction of Fischlin, using specific efficient components. In particular he makes use of the efficient NIZK proofs of Groth and Sahai [17, 16] which hold for only specific NP-statements in pairing groups. In Fuchsbauer's scheme the blind signature is constructed by providing a Groth–Sahai proof of knowledge of a signature on a message (as opposed to a signature on a commitment as in Fischlin's generic construction). This makes the underlying NIZK proofs simpler, but makes use of a different signature request phase. The resulting blind signature consists of around 30 group elements, and is the most efficient round optimal blind signature scheme in the CRS known to date.

Fuchsbauer's scheme is based on a new intractibility assumption called the ADH-SDH problem, which he shows holds in the Generic Group Model (GGM). This is a falsifiable assumption, in the sense of Naor [24], which is closely related to the q-SDH problem lying behind the Boneh–Boyen signature scheme [6]. However, the resulting blind signature is not a standard signature, e.g. it is not a true Boneh–Boyen signature.

In this paper we present a round optimal blind signature scheme in the CRS model which is significantly more efficient than Fuchsbauer's scheme; a signature only consists of three group elements. Indeed the resulting signature is a standard Camenisch–Lysyanskaya (CL) signature on the message m [10]. We note that our required hardness assumption, being interactive, is not falsifiable. However, this property is inherited from the underlying CL signature where the underlying hardness assumption is the LRSW assumption.

Our Scheme: The scheme we present has a number of similarities to previous work, yet a number of distinct advantages. The "classic" blind signature schemes are in the ROM and are essentially full-domain-hash style signature schemes. In these schemes the hash function is applied to the message, the result is then blinded and sent to the signer. The signer signs the blinded message as he would sign a normal message. On receiving the output from the signer the user then unblinds the signature using a homomorphic property of the underlying signature scheme. Such a mechanism is the basis of the original RSA based scheme of Chaum [11], which was proved secure in [3]; Chaum's scheme outputs standard RSA-FDH signatures [4]. It also forms a basis of the BLS signature [7] based blind signature scheme of Boldyreva [5]. The advantage of such schemes is that the output signature corresponds to a standard signature scheme; in these two cases RSA-FDH and BLS respectively. Our construction has a similar flavour in that the signature generation protocol is of the blind/unblind variant and that the output signature is a "standard" signature, namely a Camensich–Lysyanskaya (CL) signature, however, our scheme does not require random oracles.

Security of the RSA-FDH based blind signature is not based on the same assumption as the original RSA-FDH signature, indeed it is based on a stronger one, the same holds true for the BLS based blind signature. In our construction the same property holds; security of the standard CL signature scheme is based on the LRSW assumption [22], our schemes security rests on assuming a stronger assumption, which we call the Extended-LRSW assumption (E-LRSW), which is itself related to the Blind-LRSW assumption

(B-LRSW) previously proposed in [12]. We justify the hardness of this new assumption by presenting a proof in the GGM. We note, our proof can be modified to also justify the B-LRSW assumption in the GGM.

We note that the CRS based scheme in [19] also outputs standard CL signatures, however the signature request protocol requires factoring based assumptions to provide security. Our signature request protocol is significantly simpler.

Paper Organization: The rest of the paper is organized as follows; In Section 2 we recap on the syntax and security definitions for blind signatures. In Section 3 we recap on the basic properties of bilinear groups that we shall need, and we will present the underlying hard problems on which the security of our scheme rests. We present our scheme in Section 4, with the security proof provided in Section 5. Finally, we provide evidence for the hardness of our new intractability assumption in the GGM in Section 6.

2 Syntax and Security of Blind Signatures

In this section we define the syntax of blind signatures that we shall use, as well as recap on the standard security model. Since we are focusing on signature request phases which are two-move, we specialise the syntax for this case. This is purely to make the description of the security model and our scheme more transparent.

Syntax: A blind signature scheme BS (with a two-move signature request phase) in the CRS model consists of six probabilistic polynomial time algorithms

$$\mathsf{BS} = (\mathsf{Setup}_{\mathsf{BS}}, \mathsf{KeyGen}_{\mathsf{BS}}, \mathsf{Request}_{\mathsf{BS}}, \mathsf{Issue}_{\mathsf{BS}}, \mathsf{Unblind}_{\mathsf{BS}}, \mathsf{Verify}_{\mathsf{BS}}).$$

The syntax of these algorithms is defined as follows; where to aid notation all algorithms (bar $\mathsf{Setup}_{\mathsf{BS}}$) are assumed to take as implicit input $\mathsf{CRS}_{\mathsf{BS}}$ as output by $\mathsf{Setup}_{\mathsf{BS}}$;

- $\mathsf{Setup}_{\mathsf{BS}}(1^\lambda)$: Takes as input a security parameter λ and outputs a common string $\mathsf{CRS}_{\mathsf{BS}}$. We assume $\mathsf{CRS}_{\mathsf{BS}}$ contains a description of the key and message spaces for the scheme.
- $\mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$: Takes as input the security parameter and outputs a pair $(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}})$ of public/private keys for the signer.
- $\mathsf{Request}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}})$: This algorithm, run by the user, takes a message m in the space of messages $\mathcal{M}$ and produces a signature request ρ, plus some state St (which is assumed to contain m).
- $\mathsf{Issue}_{\mathsf{BS}}(\rho, \mathsf{sk}_{\mathsf{BS}})$: This algorithm, run by the signer, takes the signature request ρ and produces a pre-signature β.
- $\mathsf{Unblind}_{\mathsf{BS}}(\beta, \mathsf{St}, \mathsf{pk}_{\mathsf{BS}})$: On input of β and St this algorithm produces a signature σ on m, or it outputs $\perp$.
- $\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}})$: This is the public signature verification algorithm. This should output 1 if σ is a valid signature on m and 0 otherwise.

Correctness of the blind signature algorithm is that if both parties behave honestly then signatures should verify, i.e. for all CRS's output by $\mathsf{Setup}_{\mathsf{BS}}$ we have,

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

Security: The standard security model for blind signatures [18, 26] consists of two properties, blindness and unforgeability. Intuitively blindness says that an adversarial signer, given two messages m_0 and m_1 by an honest user, for which the signer produces valid blind signatures, is unable to tell which order the messages were signed in upon being presented with the final unblinded signatures. On the other hand unforgeability deals with an adversarial user whose goal is to obtain $k + 1$ distinct message/signature pairs given only k interactions with the honest signer.

To define blindness we consider an honest-but-curious adversary $\mathcal{A}$ which has three modes $\mathsf{find}, \mathsf{issue}$ and guess, running in an experiment as follows:

Experiment: $\mathsf{Exp}^{blind}_{\mathsf{BS},\mathcal{A}}(\lambda)$

- $\mathsf{CRS}_{\mathsf{BS}} \leftarrow \mathsf{Setup}_{\mathsf{BS}}(1^\lambda)$.
- $(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$.
- $(m_0, m_1, \mathsf{St}_{\mathsf{find}}) \leftarrow \mathcal{A}(\mathsf{find}, \mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}, \mathsf{CRS}_{\mathsf{BS}})$.
- $b \leftarrow \{0,1\}$.
- $(\rho_b, \mathsf{St}_b) \leftarrow \mathsf{Request}_{\mathsf{BS}}(m_0, \mathsf{pk}_{\mathsf{BS}})$:
- $(\rho_{1-b}, \mathsf{St}_{1-b}) \leftarrow \mathsf{Request}_{\mathsf{BS}}(m_1, \mathsf{pk}_{\mathsf{BS}})$:
- $(\beta_0, \beta_1, \mathsf{St}_{\mathsf{issue}}) \leftarrow \mathcal{A}(\mathsf{issue}, \rho_0, \rho_1, \mathsf{St}_{\mathsf{find}})$.
- $\sigma_0 \leftarrow \mathsf{Unblind}_{\mathsf{BS}}(\beta_b, \mathsf{St}_b, \mathsf{pk}_{\mathsf{BS}})$.
- $\sigma_1 \leftarrow \mathsf{Unblind}_{\mathsf{BS}}(\beta_{1-b}, \mathsf{St}_{1-b}, \mathsf{pk}_{\mathsf{BS}})$.
- If $\sigma_0 =\perp$ or $\sigma_1 =\perp$ then abort.
- $b^* \leftarrow \mathcal{A}(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{St}_{\mathsf{issue}})$.
- Return 1 if $b = b^*$ else return 0.

We define

$$\mathsf{Adv}^{blind}_{\mathsf{BS},\mathcal{A}}(\lambda) = 2 \cdot \left| \Pr[\mathsf{Exp}^{blind}_{\mathsf{BS},\mathcal{A}}(\lambda) = 1] - 1/2 \right|$$

and we say that the scheme is *blind* if $\mathsf{Adv}^{blind}_{\mathsf{BS},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{A}$.

To define unforgeability we consider an adversary $\mathcal{A}$, having oracle access to the function $\mathsf{Issue}_{\mathsf{BS}}(\rho, \mathsf{sk}_{\mathsf{BS}})$, running in an experiment as follows:

Experiment: $\mathsf{Exp}^{forge}_{\mathsf{BS},\mathcal{A}}(\lambda)$

- $\mathsf{CRS}_{\mathsf{BS}} \leftarrow \mathsf{Setup}_{\mathsf{BS}}(1^\lambda)$.
- $(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$.
- $((m_1, \sigma_1), \ldots, (m_{k+1}, \sigma_{k+1})) \leftarrow \mathcal{A}^{\mathsf{Issue}_{\mathsf{BS}}(\cdot,\cdot)}(\mathsf{pk}_{\mathsf{BS}}, \mathsf{CRS}_{\mathsf{BS}})$.
- If $\mathcal{A}$ called its oracle more than k times then return 0.
- If $\exists i, j \in \{1, \ldots, k+1\}$, with $i \neq j$, such that $m_i = m_j$ then return 0.
- If $\exists i \in \{1, \ldots, k+1\}$ such that $\mathsf{Verify}_{\mathsf{BS}}(m_i, \sigma_i, \mathsf{pk}_{\mathsf{BS}}) = 0$ then return 0.
- Return 1.

We define

$$\mathsf{Adv}^{forge}_{\mathsf{BS},\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{forge}_{\mathsf{BS},\mathcal{A}}(\lambda) = 1]$$

and we say that the scheme is *unforgeable* if $\mathsf{Adv}^{forge}_{\mathsf{BS},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{A}$.

Note, our definition of forgery is not that of strong unforgeability; we do not require the adversary to not be able to output a new signature on an old message. This is because the final signature in our scheme will be a CL signature, which is always randomizable. Hence, our blind signature scheme may not be suitable in applications which require the strong unforgeability definition.

3 Bilinear Groups and Associated Hard Problems

In this section we introduce the basic mathematical constructs needed to present our scheme. Namely abelian groups admitting a bilinear pairing and the associated hard problems.

Bilinear Groups: Bilinear groups are a set of three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, of prime order p, along with a bilinear map (a deterministic function) $\hat{t}$ which takes as input one element in $\mathbb{G}_1$ and one element in $\mathbb{G}_2$ and outputs an element in $\mathbb{G}_T$. We shall write $\mathbb{G}_1$ and $\mathbb{G}_2$ additively (with identity element 0), and $\mathbb{G}_T$ multiplicatively (with identity element 1), and write $\mathbb{G}_1 = \langle P_1 \rangle, \mathbb{G}_2 = \langle P_2 \rangle$, for two explicitly given generators P_1 and P_2. We define $\mathcal{P} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{t}, P_1, P_2)$ to be the set of pairing group parameters.

The function $\hat{t}$ must have the following three properties:

1. Bilinearity: $\forall Q_1 \in \mathbb{G}_1$, $\forall Q_2 \in \mathbb{G}_2$, $\forall x, y \in \mathbb{Z}$, we have
$$\hat{t}([x]Q_1, [y]Q_2) = \hat{t}(Q_1, Q_2)^{xy}.$$
2. Non-Degeneracy: The value $\hat{t}(P_1, P_2)$ generates $\mathbb{G}_T$.
3. The function $\hat{t}$ is efficiently computable.

In practice there are a number of different types of bilinear groups one can take, each giving rise to different algorithmic properties and different hard problems. Following [15] we categorize pairings into three distinct types (other types are possible, but the following three are the main ones utilized in practical protocols).

- **Type-1**: This is the symmetric pairing setting in which $\mathbb{G}_1 = \mathbb{G}_2$.
- **Type-2**: Here we have $\mathbb{G}_1 \neq \mathbb{G}_2$, but there is an efficiently computable isomorphism $\psi : \mathbb{G}_2 \longrightarrow \mathbb{G}_1$ where $\psi(P_2) = P_1$.
- **Type-3**: Again $\mathbb{G}_1 \neq \mathbb{G}_2$, but now there is no known efficiently computable isomorphism.

In this paper we shall always consider Type-3 pairings. Such pairings can be efficiently realised; by taking $\mathbb{G}_1$ to be the set of points of order p of an elliptic curve over $\mathbb{F}_q$ with "small" embedding degree k; by taking $\mathbb{G}_2$ to be the set of points of order p on a twist of the same elliptic curve over $\mathbb{F}_{q^e}$, for some divisor e of k; and $\mathbb{G}_T$ to be the subgroup of order p in the finite field $\mathbb{F}_{q^k}$.

For a security parameter λ we let $\mathsf{Setup}_{\mathsf{Grp}}(1^\lambda)$ denote an algorithm which produces a pairing group instance $\mathcal{P}$ of Type-3.

Hard Problems: The security of our scheme rests on two assumptions which hold in pairing groups; the first assumption is particular to Type-3 pairings, whereas the second assumption can be defined in any type of pairing group.

Definition 1 (Decision Diffie-Hellman Assumption in $\mathbb{G}_1$). *The DDH assumption in $\mathbb{G}_1$ is said to hold if the following differences of probabilities is negligible in the security parameter λ, for all adversaries $\mathcal{A}$ and all parameter sets $\mathcal{P}$ output by $\mathsf{Setup}_{\mathsf{Grp}}(1^\lambda)$,*

$$\begin{aligned} &\mathsf{Adv}_{\mathsf{DDH},\mathcal{A}}(\lambda) = \\ &\quad \Pr\left[x, y, z \leftarrow \mathbb{F}_p, X = [x]P_1, Y = [y]P_1, Z = [z]P_1 : \mathcal{A}(X, Y, Z, \mathcal{P}) = 1\right] \\ &\quad - \Pr\left[x, y \leftarrow \mathbb{F}_p, X = [x]P_1, Y = [y]P_1, Z = [x \cdot y]P_1 : \mathcal{A}(X, Y, Z, \mathcal{P}) = 1\right]. \end{aligned}$$

Note, the assumption is about the outputs of the specific algorithm $\mathsf{Setup}_{\mathsf{Grp}}$, since one can have a Type-3 pairing group in which DDH in $\mathbb{G}_1$ does not hold. The DDH assumption in $\mathbb{G}_1$ is believed to hold for appropriate outputs of $\mathsf{Setup}_{\mathsf{Grp}}$ in both the Type-2 and Type-3 pairings situations. Our scheme we present in the Type-3 setting as this is the most efficient setting, and in addition our generic group model proof of the hardness of our other assumption we give only in the Type-3 setting.

Our second hard problem is related to the following problem introduced in [22] in the case of Type-1 pairings; we present this problem in it generality for all pairings,

Definition 2 (LRSW Assumption). *If $\mathcal{A}$ is an algorithm which is given access to an oracle $\mathcal{O}_{[x]P_2,[y]P_2}(\cdot)$ that on input of $f \in \mathbb{F}_p$ outputs $(A, B, C) = (A, [y]A, [x + f \cdot x \cdot y]A)$, for some random $A \in \mathbb{G}_1 \setminus \{0\}$, we let Q denote the set of queries made by $\mathcal{A}$ to $\mathcal{O}_{[x]P_2,[y]P_2}(\cdot)$.*

The LRSW assumption is said to hold for the output of $\mathsf{Setup}_{\mathsf{Grp}}$ if for all probabilistic polynomial time adversaries $\mathcal{A}$, and all outputs of $\mathsf{Setup}_{\mathsf{Grp}}$, the following probability is negligible in the security parameter λ,

$$\begin{aligned} \Pr\,[\,&x \leftarrow \mathbb{F}_p,\ y \leftarrow \mathbb{F}_p,\ X \leftarrow [x]P_2,\ Y \leftarrow [y]P_2, \\ &(Q, m, A, B, C) \leftarrow \mathcal{A}^{\mathcal{O}_{X,Y}(\cdot)}(\mathcal{P}, X, Y) : \\ &m \notin Q\ \wedge\ m \in \mathbb{F}_p \setminus \{0\}\ \wedge \\ &A \in \mathbb{G}_1 \setminus \{0\}\ \wedge\ B = [y]A\ \wedge\ C = [x + m \cdot x \cdot y]A\ ] \end{aligned}$$

In [22] it was shown that the LRSW assumption holds in the GGM and is independent of the DDH assumption. The LRSW assumption is the underlying hard problem behind the Camenisch–Lysyanskaya (CL) signature scheme [10].

Definition 3 (Camenisch–Lysyanskaya Signature Scheme). *The CL signature scheme is given by the following triple of algorithms given an output $\mathcal{P}$ of $\mathsf{Setup}_{\mathsf{Grp}}(1^\lambda)$.*

- $\mathsf{KeyGen}(\mathcal{P})$*: Set* $\mathsf{sk}_{CL} \leftarrow (x, y) \in \mathbb{F}_p^2$ *and* $\mathsf{pk}_{CL} \leftarrow (X, Y) = ([x]P_2, [y]P_2)$.
- $\mathsf{Sign}(m, \mathsf{sk}_{CL})$*: Select* $A \in \mathbb{G}_1 \setminus \{0\}$*, and then set* $B \leftarrow [y]A$*,* $C = [x + m \cdot x \cdot y]A$*. Output* (A, B, C).
- $\mathsf{Verify}(m, (A, B, C), \mathsf{pk}_{CL})$*: Output true if and only if* $\hat{t}(B, P_2) = \hat{t}(A, Y)$ *and* $\hat{t}(C, P_2) = \hat{t}(A, X) \cdot \hat{t}(B, X)^m$.

Indeed the LRSW problem and the EF-CMA security of the CL signature scheme are equivalent since the oracle in the LRSW assumption produces CL signatures, and the output of the adversary against the LRSW assumption corresponds precisely to a forger who can construct one more CL signature.

We shall require a strengthening of the LRSW assumption, a variant of which was first proposed by Chen et. al [12]. We first present the strengthening of Chen et. al, called the Blind-LRSW assumption, and then we present our modification which we call the Extended-LRSW assumption (E-LRSW).

In the B-LRSW problem the oracle provided to the adversary does not take as input an element $m \in \mathbb{F}_p$, but instead takes as input an element $M \in \mathbb{G}_1$. The output of the oracle is a CL signature on the discrete logarithm m of M with respect to P_1. The output of the adversary against this modified LRSW assumption is still a valid CL signature on a new element m. Following [12] we call this the Blind-LRSW (B-LRSW) assumption, since the adversary is given access to an oracle which can produce what are in effect blinded signatures. Formally we define

Definition 4 (B-LRSW Assumption). *If $\mathcal{A}$ is an algorithm which is given access to an oracle $\mathcal{O}^B_{[x]P_2,[y]P_2}(\cdot)$ that on input of $M = [m]P_1 \in \mathbb{G}_1$ outputs $(A, B, C) = (A, [y]A, [x+m\cdot x\cdot y]A)$, for some random $A \in \mathbb{G}_1 \setminus \{0\}$, we let Q denote the set of queries made by $\mathcal{A}$ to $\mathcal{O}^B_{[x]P_2,[y]P_2}(\cdot)$.*

The B-LRSW assumption is said to hold for the output of $\mathsf{Setup}_{\mathsf{Grp}}$ if for all probabilistic polynomial time adversaries $\mathcal{A}$, and all outputs of $\mathsf{Setup}_{\mathsf{Grp}}$, the following probability is negligible in the security parameter λ,

$$\begin{aligned}
\Pr\,[\,& x \leftarrow \mathbb{F}_p,\ y \leftarrow \mathbb{F}_p,\ X \leftarrow [x]P_2,\ Y \leftarrow [y]P_2, \\
& (Q, m, A, B, C) \leftarrow \mathcal{A}^{\mathcal{O}^B_{X,Y}(\cdot)}(\mathcal{P}, X, Y) : \\
& [m]P_1 \notin Q \ \wedge\ m \in \mathbb{F}_p \setminus \{0\} \ \wedge \\
& A \in \mathbb{G}_1 \setminus \{0\} \ \wedge\ B = [y]A \ \wedge\ C = [x + m \cdot x \cdot y]A\ ]
\end{aligned}$$

Note, that the oracle in the B-LRSW assumption, given access to x and y, can compute its response by generating $a \in \mathbb{F}_p$ at random and outputing the triple $([a]P_1, [a \cdot y]P_1, [a \cdot x]P_1 + [a \cdot x \cdot y]M)$, i.e. the oracle does not need to be able to solve discrete logarithms if it has access to x and y.

Also note, since an oracle query to $\mathcal{O}_{X,Y}(\cdot)$ can be simulated with an oracle query to $\mathcal{O}^B_{X,Y}(\cdot)$, and the output of the adversaries in the two problems is essentially identical, that an adversary $\mathcal{A}^B$ against LRSW can be turned into an adversary against B-LRSW (but it appears not vice-versa). Thus B-LRSW is a stronger assumption than LRSW.

In our hard problem, shown to hold in the GGM in Section 6, we extend the oracle in the B-LRSW assumption to provide some additional data about the values A, x and y, with respect to a new public key element $Z = [z]P_1$. The output of the adversary still being a Camenisch–Lysyanskaya signature.

Definition 5 (E-LRSW Assumption). *If $\mathcal{A}$ is an algorithm which is given access to an oracle $\mathcal{O}^E_{[x]P_2,[y]P_2,[z]P_1}(\cdot)$ that on input of $M = [m]P_1 \in \mathbb{G}_1$ (for some unknown value of f) outputs $(A, B, C, D) = (A, [y]A, [x + m \cdot x \cdot y]A, [x \cdot y \cdot z]A)$, for some random $A \in \mathbb{G}_1 \setminus \{0\}$, we let q denote the number of queries made by $\mathcal{A}$ to $\mathcal{O}^E_{[x]P_2,[y]P_2,[z]P_1}(\cdot)$.*

The E-LRSW assumption is said to hold for the output of $\mathsf{Setup}_{\mathsf{Grp}}$*, if for all probabilistic polynomial time adversaries* $\mathcal{A}$*, and all outputs of* $\mathsf{Setup}_{\mathsf{Grp}}$*, the following probability is negligible in the security parameter* λ*,*

$$\begin{aligned}\mathsf{Adv}_{\mathsf{E-LRSW},\mathcal{A}}(\lambda) = \Pr\,[\, & x \leftarrow \mathbb{F}_p,\ y \leftarrow \mathbb{F}_p,\ z \leftarrow \mathbb{F}_p,\\ & X \leftarrow [x]P_2,\ Y \leftarrow [y]P_2,\ Z \leftarrow [z]P_1\\ & (\{m_i, A_i, B_i, C_i\}_{i=1}^{q+1}) \leftarrow \mathcal{A}^{\mathcal{O}^E_{X,Y,Z}(\cdot)}(\mathcal{P}, X, Y, Z) :\\ & \textit{For all } 1 \leq i \leq q+1 \textit{ we have } m_i \in \mathbb{F}_p \setminus \{0\} \ \wedge\\ & A_i \in \mathbb{G}_1 \setminus \{0\} \ \wedge \ B_i = [y]A_i \ \wedge \ C_i = [x + m_i \cdot x \cdot y]A_i\\ & \ \wedge \ (\textit{if } i \neq j \textit{ then } m_i \neq m_j)\ ]\end{aligned}$$

Note, we present this problem in terms of a one-more problem rather than in the form of the B-LRSW assumption. This is because the extra item D allows the adversary to "open" the signature in one of a number of ways. However, the intuition is that he needs to commit to how he is going to open the signature before he sends the request to the oracle. We formalise this intuition when we present a proof in the GGM of the difficulty of the E-LRSW problem.

We note that our E-LRSW assumption is in some sense to the LRSW assumption, as the HSDH problem from [8] is to the SDH problem. In that the "queries" $(g^{1/(x+c_i)}, c_i)$ in the SDH problem are replaced by "blinded queries" $(g^{1/(x+c_i)}, g^{c_i}, h^{c_i})$ in the HSDH problem, and the output value $(g^{1/(x+c^*)}, c^*)$ in the SDH problem is replaced by the blinded output value $(g^{1/(x+c^*)}, g^{c^*}, h^{c^*})$ in the HSDH problem.

4 Our Scheme

```
$\mathsf{Setup}_{\mathsf{BS}}(1^\lambda)$:
  – $\mathcal{P} \leftarrow \mathsf{Setup}_{\mathsf{Grp}}(1^\lambda)$.
  – $z \leftarrow \mathbb{F}_p$.
  – $Z \leftarrow [z]P_1$.
  – $\mathcal{M} := \mathbb{F}_p \setminus \{0\}$.
  – $\mathsf{CRS}_{\mathsf{BS}} \leftarrow (\mathcal{P}, Z, \mathcal{M})$.
  – Output $\mathsf{CRS}_{\mathsf{BS}}$.

$\mathsf{Request}_{\mathsf{BS}}(m, \mathsf{pk}_{\mathsf{BS}})$:
  – $r \leftarrow \mathbb{F}_p$.
  – $\mathsf{Co} \leftarrow [m]P_1 + [r]Z$.
  – $\rho \leftarrow Co$, $\mathsf{St} \leftarrow (m, r)$.
  – Output $(\rho, \mathsf{St})$.

$\mathsf{Unblind}_{\mathsf{BS}}(\beta, \mathsf{St}, \mathsf{pk}_{\mathsf{BS}})$:
  – Parse $\beta$ as $(A, B, C, D)$.
  – Parse $\mathsf{St}$ as $(m, r)$.
  – $C \leftarrow C - [r]D$.
  – If $\mathsf{Verify}_{\mathsf{BS}}(m, (A, B, C), \mathsf{pk}_{\mathsf{BS}}) = 0$
      • Return $\perp$.
  – $t \leftarrow \mathbb{F}_p \setminus \{0\}$.
  – $A \leftarrow [t]A$, $B \leftarrow [t]B$, $C \leftarrow [t]C$.
  – $\sigma \leftarrow (A, B, C)$.
  – Output $\sigma$.
```

```
$\mathsf{KeyGen}_{\mathsf{BS}}(1^\lambda)$:
  – $x, y \leftarrow \mathbb{F}_p$.
  – $X \leftarrow [x]P_2$.
  – $Y \leftarrow [y]P_2$.
  – $\mathsf{sk}_{\mathsf{BS}} \leftarrow (x, y)$, $\mathsf{pk}_{\mathsf{BS}} \leftarrow (X, Y)$.
  – Output $(\mathsf{pk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}})$.

$\mathsf{Issue}_{\mathsf{BS}}(\rho, \mathsf{sk}_{\mathsf{BS}})$:
  – Parse $\rho$ as $Co$.
  – $a \leftarrow \mathbb{F}_p \setminus \{0\}$.
  – $A \leftarrow [a]P_1$
  – $B \leftarrow [a \cdot y]P_1$
  – $C \leftarrow [a \cdot x]P_1 + [a \cdot x \cdot y]Co$
  – $D \leftarrow [a \cdot x \cdot y]Z$.
  – $\beta \leftarrow (A, B, C, D)$.
  – Output $\beta$.

$\mathsf{Verify}_{\mathsf{BS}}(m, \sigma, \mathsf{pk}_{\mathsf{BS}})$:
  – Parse $\sigma$ as $(A, B, C)$.
  – If $A = 0$ or $\hat{t}(A, Y) \neq \hat{t}(B, P_2)$
      or $\hat{t}(C, P_2) \neq \hat{t}(A, X) \cdot \hat{t}(B, X)^m$
        • Return 0.
  – Return 1.
```

Fig. 1. Our blind signature scheme

In Figure we 1 define the algorithms which make up our blind signature scheme. Notice, that β contains a CL signature on the "hidden" message $m + z \cdot r$, but no party knows the value of z. The signer is signing a message he does not know. Indeed the user also does not know the message (unless he picks $r = 0$), but he is able to unblind this signature to produce a valid CL signature on m using his value r. To see that the unblinded signature is valid, notice that the value of C (before multiplication by t) is equal to

$$\begin{aligned} C &= ([a \cdot x]P_1 + [a \cdot x \cdot y]\mathsf{Co}) - [r]D, \\ &= [x]A + [a \cdot x \cdot y] \cdot ([m]P_1 + [r]Z) - [r]D, \\ &= [x + m \cdot x \cdot y]A + [a \cdot x \cdot y \cdot r]Z - [r]D, \\ &= [x + m \cdot x \cdot y]A. \end{aligned}$$

Then notice, that even the revealed signature provides no linkage with the values signed, due to the fact that CL signatures for Type-3 pairings are unlinkable once randomized. In our security proof we will confirm this intuition.

5 Proof of Security

First we prove that the scheme is blind, then we show that it is unforgeable.

Theorem 1. *If the DDH assumption holds in $\mathbb{G}_1$ then the above blind signature scheme is blind. In particular if $\mathcal{A}$ is an adversary against the blindness of the above blind signature scheme, then there is an adversaries $\mathcal{B}$ such that*

$$\mathsf{Adv}^{blind}_{\mathsf{BS},\mathcal{A}}(\lambda) \leq \mathsf{Adv}_{\mathsf{DDH},\mathcal{B}}(\lambda).$$

Proof. The proof proceeds via two games, G_0 and G_1. We let G_0 denote the experiment played by the adversary $\mathcal{A}$ against the blindness property played against the scheme, see Figure 2. By the definition of the blindness game if the adversary $\mathcal{A}$ does not return valid values of β_0 and β_1 then it has no advantage in winning the game, i.e. the game definition assumes an adversary that is honest-but-curious. This property allows us to make our game hop to game G_1, also presented in Figure 2.

```
Game $\mathsf{G}_0$:
  – $z \leftarrow \mathbb{F}_p$, $Z \leftarrow [z]P_1$.
  – $x, y \leftarrow F_p$, $(X, Y) \leftarrow ([x]P_2, [y]P_2)$.
  – $(m_0, m_1, \mathsf{St}_{\mathsf{find}}) \leftarrow \mathcal{A}(\mathsf{find}, (X, Y), (x, y), Z)$.
  – $b \leftarrow \{0, 1\}$.
  – $r_0, r_1 \leftarrow \mathbb{F}_p$.
  – $\mathsf{Co}_b \leftarrow [m_0]P_1 + [r_0]Z$.
  – $\mathsf{Co}_{1-b} \leftarrow [m_1]P_1 + [r_1]Z$.
  – $(\beta_0, \beta_1, \mathsf{St}_{\mathsf{issue}}) \leftarrow \mathcal{A}(\mathsf{issue}, \mathsf{Co}_0, \mathsf{Co}_1, \mathsf{St}_{\mathsf{find}})$.
  – Parse $\beta_0$ as $(A_b, B_b, C_b, D_b)$.
  – Parse $\beta_1$ as $(A_{1-b}, B_{1-b}, C_{1-b}, D_{1-b})$.
  – $C_0 \leftarrow C_0 - [r_0]D_0$, $C_1 \leftarrow C_1 - [r_1]D_1$.
  – $t_0, t_1 \leftarrow \mathbb{F}_p \setminus \{0\}$.
  – $\sigma_0 \leftarrow ([t_0]A_0, [t_0]B_0, [t_0]C_0)$.
  – $\sigma_1 \leftarrow ([t_1]A_1, [t_1]B_1, [t_1]C_1)$.
  – $b^* \leftarrow \mathcal{A}(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{St}_{\mathsf{issue}})$.
  – Return 1 if $b = b^*$ else return 0.
```

```
Game $\mathsf{G}_1$:
  – $z \leftarrow \mathbb{F}_p$, $Z \leftarrow [z]P_1$.
  – $x, y \leftarrow F_p$, $(X, Y) \leftarrow ([x]P_2, [y]P_2)$.
  – $(m_0, m_1, \mathsf{St}_{\mathsf{find}}) \leftarrow \mathcal{A}(\mathsf{find}, (X, Y), (x, y), Z)$.
  – $b \leftarrow \{0, 1\}$.
  – $r_0, r_1 \leftarrow \mathbb{F}_p$.
  – $\mathsf{Co}_b \leftarrow [m_0]P_1 + [r_0]Z$.
  – $\mathsf{Co}_{1-b} \leftarrow [m_1]P_1 + [r_1]Z$.
  – $(\beta_0, \beta_1, \mathsf{St}_{\mathsf{issue}}) \leftarrow \mathcal{A}(\mathsf{issue}, \mathsf{Co}_0, \mathsf{Co}_1, \mathsf{St}_{\mathsf{find}})$.
  – $a_0, a_1 \leftarrow \mathbb{F}_p \setminus \{0\}$.
  – $\sigma_0 \leftarrow ([a_0]P_1, [a_0 \cdot y]P_1, [a_0 \cdot x + a_0 \cdot x \cdot y \cdot m_0]P_1)$.
  – $\sigma_1 \leftarrow ([a_1]P_1, [a_1 \cdot y]P_1, [a_1 \cdot x + a_1 \cdot x \cdot y \cdot m_1]P_1)$.
  – $b^* \leftarrow \mathcal{A}(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{St}_{\mathsf{issue}})$.
  – Return 1 if $b = b^*$ else return 0.
```

Fig. 2. The two games G_0 and G_1.

As one can see, in G_1 the game ignores the values β_0 and β_1 returned by the adversary and simply generates a new CL signature σ_0 and σ_1, using the secret key $\mathsf{sk}_{\mathsf{BS}}$ available to the environment, rather than performing the $\mathsf{Unblind}_{\mathsf{BS}}$ operation of the scheme. If we let $\mathcal{S}_0(m)$ denote the distribution of signatures returned to the final stage of the adversary in game G_0 on message m, and $\mathcal{S}_1(m)$ the distribution in game G_1, we see that $\mathcal{S}_0(m)$ is identically distributed to $\mathcal{S}_1(m)$. This is because in G_0 we randomize a specific CL signature, whereas in G_1 we produce a new independent CL signature. It is easy to see that these two operations have the same effect on the distribution of the final signature passed to the final stage of the adversary. If we let $\Pr[\mathsf{G}_i]$ denote the probability of the adversary winning in game G_i, we have $\Pr[\mathsf{G}_0] = \Pr[\mathsf{G}_1]$.

We now show that an adversary playing G_1, which wins with non-negligible advantage, can be used to solve the DDH problem in $\mathbb{G}_1$. To see this note that the value Co returned by $\mathsf{Request}_{\mathsf{BS}}$ is a Pedersen commitment to the value m, hence an adversary in game G_1 can be used to solve the following blindness property of Pedersen commitment:

Experiment: $\mathsf{Exp}^{blind}_{\mathsf{Pedersen},\mathcal{C}}(\lambda)$

- $\mathsf{sk} \leftarrow \mathbb{F}_p$.
- $\mathsf{pk} \leftarrow [\mathsf{sk}]P_1$.
- $(m_0, m_1, \mathsf{St}) \leftarrow \mathcal{C}_1(\mathsf{pk})$, with $m_i \in \mathbb{F}_p$.
- $b \leftarrow \{0, 1\}$.
- $r_0, r_1 \leftarrow \mathbb{F}_p$.
- $C_0 \leftarrow [m_b]P_1 + [r_0]\mathsf{pk}$.
- $C_1 \leftarrow [m_{1-b}]P_1 + [r_1]\mathsf{pk}$.
- $b' \leftarrow \mathcal{C}_2(C_0, C_1, \mathsf{St})$.
- If $b = b'$ then return 1, else return 0.

If $\mathcal{C}$ is an adversary in the above experiment then we define

$$\mathsf{Adv}^{blind}_{\mathsf{Pedersen},\mathcal{C}}(\lambda) = 2|\Pr[\mathsf{Exp}^{blind}_{\mathsf{Pedersen},\mathcal{C}}(\lambda)] - 1/2|.$$

That we have

$$\Pr[\mathsf{G}_1] = \Pr[\mathsf{Exp}^{blind}_{\mathsf{Pedersen},\mathcal{C}}(\lambda)]$$

follows immediately; we simply translate the (m_0, m_1) output by the first stage of adversary $\mathcal{A}$ in game G_1, into the challenge for the Pedersen blindness game; this provides the input into the second stage of adversary $\mathcal{A}$; the input to the third stage of adversary $\mathcal{A}$ is then independent of the values returned, by our first game hop.

That Pedersen commitments are computationally blind as long as the DDH problem holds in the associated group is a classic result. Thus there is an adversary $\mathcal{B}$ against DDH in $\mathbb{G}_1$ such that the advantage of $\mathcal{C}$ against the blindness of Pedersen commitments is bounded by the advantage of $\mathcal{B}$ against DDH in $\mathbb{G}_1$. The theorem statement then follows immediately.

We now turn to show that our scheme is is unforgeable.

Theorem 2. *If the E-LRSW assumption holds then the above blind signature scheme is unforgeable. In particular if $\mathcal{A}$ is an adversary against the unforgeability of the above blind signature scheme, then there is an adversary $\mathcal{B}$ which solves the E-LRSW problem such that*

$$\mathsf{Adv}^{forge}_{\mathsf{BS},\mathcal{A}}(\lambda) \leq \mathsf{Adv}_{\mathsf{E-LRSW},\mathcal{B}}(\lambda).$$

Proof. Let $\mathcal{A}$ denote an adversary against the unforgeability experiment of the blind signature scheme. We shall use $\mathcal{A}$ to construct an algorithm which solves the E-LRSW problem. Let $(\mathcal{P}, X, Y, Z)$ denote the input to the E-LRSW problem instance. Algorithm $\mathcal{B}$ sets up the CRS and the public keys for the blind signature scheme by setting $\mathsf{CRS}_{\mathsf{BS}} \leftarrow (\mathcal{P}, Z, \mathbb{F}_p \setminus \{0\})$ and $\mathsf{pk}_{\mathsf{BS}}(X, Y)$.

Algorithm $\mathcal{B}$ now calls adversary $\mathcal{A}$. At some point $\mathcal{A}$ will make one of q queries to its $\mathsf{Issue}_{\mathsf{BS}}$ oracle. Algorithm $\mathcal{B}$ responds to a query of Co as follows: It passes the value Co to its E-LRSW oracle $\mathcal{O}^E_{X,Y,Z}$ so as to obtain a tuple

$$(A, B, C, D) = (A, [y]A, [x + t \cdot x \cdot y]A, [x \cdot y \cdot z]A),$$

where $\mathsf{Co} = [t]P_1$. Notice that if $t = m + r \cdot z$ then (A, B, C, D) is a valid response for the commitment to the message m. The tuple (A, B, C, D) is now passed back to algorithm $\mathcal{A}$.

At some point $\mathcal{A}$ terminates and outputs a set of $q + 1$ values (m_i, A_i, B_i, C_i) where (A_i, B_i, C_i) is a valid CL signature on m_i. By returning this list to its challenger the adversary thereby solves the E-LRSW assumption.

6 E-LRSW in the Generic Group Model

We now show that the E-LRSW assumption holds in the GGM. Whilst not providing complete evidence that a given problem is hard, a GGM proof can be used to gain confidence in an underlying assumption. For more details of the GGM and its application to other problems, see [27, 23].

Theorem 3. *Let $\mathcal{A}$ denote an adversary in the generic group model against the E-LRSW assumption. Assume $\mathcal{A}$ makes q_G queries to the group operations, q_P queries to the pairing and q_O queries to the E-LRSW oracle $\mathcal{O}^E$. If we set $m = 3 + q_G + 4q_O + q_P$ then the probability of the adversary winning the E-LRSW game is $O(m^2 \cdot q_O/p)$, where p is the (prime) order of the generic group.*

Proof. A GGM adversary against E-LRSW works by interacting with the group, pairing and $\mathcal{O}^E$ oracle via group handles. We first present how the adversary uses these handles to provide access to these operations. Different authors use slightly different notions as to how this is done, but all are essentially equivalent. The challenger keeps three lists $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T$ of pairs (σ, P) where σ is a "random" encoding of the group element chosen from some set $\mathcal{S}$ with $\#\mathcal{S} > 3 \cdot p$, and P is some polynomial in $\mathbb{F}_p[X, Y, Z, A_1, \ldots, A_{q_O}]$.

To each list we associate an Update operation, that takes as input the specific list $\mathcal{G}$ and a polynomial P. The function $\mathsf{Update}(\mathcal{G}, P)$ searches the list for a pair whose second component is equal to P, if it finds one then it returns the first component as a result. Otherwise a new element σ is pulled from $\mathcal{S}$, distinct from all other elements in $\mathcal{S}$ used so far, and the pair $(\sigma, \mathcal{P})$ is added to the list. The value σ is then returned. The values σ are the handles used by the adversary to represent the group elements; note the adversary at no point gets access to the second element in the pairs.

The lists are first initialized by executing the following six operations:

$$\begin{aligned}&\mathsf{Update}(\mathcal{G}_1, 1), \mathsf{Update}(\mathcal{G}_2, 1), \mathsf{Update}(\mathcal{G}_T, 1),\\ &\mathsf{Update}(\mathcal{G}_1, Z), \mathsf{Update}(\mathcal{G}_2, X), \mathsf{Update}(\mathcal{G}_2, Y).\end{aligned}$$

The adversary now interacts with these lists via the following operations;

<u>Group Operations</u>: The adversary has access to three oracles $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_T$ which allow him access to the group operations, via a subtraction operation. On a call to $\mathcal{O}_i(\sigma_1, \sigma_2)$ the challenger searches list $\mathcal{G}_i$ for pairs of the form (σ_1, P_1) and (σ_2, P_2). If two such pairs do not exist then the challenger returns $\perp$. Otherwise, the challenger returns to the adversary the output of $\mathsf{Update}(\mathcal{G}_i, P_1 - P_2)$.

<u>Pairing Operation</u>: The adversary has access to an oracle $\mathcal{O}_P$. When called with $\mathcal{O}_P(\sigma_1, \sigma_2)$ the challenger searches the list $\mathcal{G}_1$ for a pair of the form (σ_1, P_1), and the list $\mathcal{G}_2$ for a pair of the form (σ_2, P_2). Again if no such pairs are found to exist then the challenger returns $\perp$. Otherwise, the challenger returns to the adversary the output of $\mathsf{Update}(\mathcal{G}_T, P_1 \cdot P_2)$.

<u>E-LRSW Oracle</u>: The adversary may make up to q_O queries of the oracle $\mathcal{O}^E(\sigma)$. We let i denote the ith such query. Upon receiving such a request, the challenger searches the $\mathcal{G}_1$ list for a pair (σ, P). If no such pair exists then the challenger returns $\perp$. Otherwise, the challenger executes the operations, where A_i, X, Y and Z are the indeterminants introduced above,

$$\begin{aligned}\sigma_A &\leftarrow \mathsf{Update}(\mathcal{G}_1, A_i),\\ \sigma_B &\leftarrow \mathsf{Update}(\mathcal{G}_1, A_i \cdot Y),\\ \sigma_C &\leftarrow \mathsf{Update}(\mathcal{G}_1, A_i \cdot X \cdot (1 + Y \cdot P)),\\ \sigma_D &\leftarrow \mathsf{Update}(\mathcal{G}_1, A_i \cdot X \cdot Y \cdot Z).\end{aligned}$$

Returning the tuple $(\sigma_A, \sigma_B, \sigma_C, \sigma_D)$ to the adversary.

Notice after executing these operations the total number of non-constant polynomials contained in the three lists $\mathcal{G}_1, \mathcal{G}_2$ and $\mathcal{G}_T$ is bounded from above by $m = 3 + q_G + 4q_O + q_P$.

Using these oracles we can simulate the entire run of the adversary in the GGM, i.e. the adversary may make no decision which depends on the particular encoding of group elements used. We wish to show that the probability of such an adversary being successful is negligibly small. Thus assume the adversary is successful in solving the E-LRSW problem. It will therefore eventually output a set of $q_O + 1$ tuples

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

where $m_i \in \mathbb{F}_p \setminus \{0\}$ are distinct and $\sigma_A^{(i)}, \sigma_B^{(i)}, \sigma_C^{(i)}$ are handles on group elements in $\mathbb{G}_1$. We let $P_A^{(i)}, P_B^{(i)}, P_C^{(i)}$ denote the associated polynomials related to these handles in the list $\mathcal{G}_1$.

As the output is supposed to correspond to a solution to the E-LRSW problem, these output polynomials can be assumed to satisfy, for some assignment $(x, y, z, a_1, \ldots, a_{q_O}) \in \mathbb{F}_p^{3+q_O}$ to the variables $(X, Y, Z, A_1, \ldots, A_{q_O})$, the two equations

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

From this we wish to derive a contradiction, i.e. conclude that the adversary cannot solve the E-LRSW assumption in the GGM.

Typically a proof in the GGM proceeds by showing that the probability of an assignment to the variables resulting in either the output equations being satisfied, or two polynomials on the lists being equal at this assignment, is negligible. In most proofs it is obvious that the output polynomials do not result in the checking equations which are *identically zero*, however in our situation this is not the obvious. In other words it is not obvious that we do not have, for all i, that

$$Y \cdot P_A^{(i)} - P_B^{(i)} \equiv X \cdot P_A^{(i)} + X \cdot m_i \cdot P_B^{(i)} - P_C^{(i)} \equiv 0.$$

Thus we divide the proof into demonstrating three properties:

- Firstly, that the above set of equations is not identically zero.
- Secondly, that the adversary does not learn, for a specific assignment to the variables, that it is not interacting with genuine group oracles.
- Thirdly, that the chance of a specific assignment to the variables, satisfying the above set of (non-trivial) equalities is negligible.

Before examining these properties we notice that the degree of the polynomials in $\mathcal{G}_1, \mathcal{G}_2$ and $\mathcal{G}_T$ can only be raised by application of the $\mathcal{O}_P$ and $\mathcal{O}^E$ oracles. Indeed if q_O is the total number of queries to $\mathcal{O}^E$ made by the adversary, then the total degree of all polynomials is bounded from above by $\mathsf{d}_{\mathsf{max}} = 9 \cdot q_O + 1$.

Property One: The first of these properties is often not addressed in GGM proofs since it is usually obvious that the required equation cannot hold for all assignments; consider for example the proof of difficulty of CDH in the GGM the output equation is a quadratic equation yet only linear operations on linear equations are allowed. Our situation is different, the oracle $\mathcal{O}^E$ allows us to obtain solutions which hold for all assignments.

We first observe that if we are trying to construct polynomials $P_A^{(i)}$ such that $P_B^{(i)} = Y \cdot P_A^{(i)}$, for polynomials $P_A^{(i)}, P_B^{(i)}$ in $\mathcal{G}_1$, then neither $P_A^{(i)}$ nor $P_B^{(i)}$ can involve any terms in X. This fact follows from the nature of the oracle queries available to the adversary. To see this we proceed by induction, assume no such polynomial $P_A^{(i)}$ exists which has positive degree in X at the point where $\mathcal{A}$ makes its jth query to the $\mathcal{O}_E$ oracle. Clearly this property holds when $j = 0$, so assume the property holds when at an arbitrary j.

At the point of making the $j+1$ query, the adversary has been able to produce linear combinations of the polyomials available before the jth query, and those obtained from the jth query. This is because only the $\mathcal{O}_E$ oracle allows the adversary to produce non-linear operations on polynomials in the $\mathcal{G}_1$ list. But it is clear that the jth query (followed by linear operations) cannot result in two polynomial with $\deg_X(P_A^{(i)}) \geq 1$ and $P_B^{(i)} = Y \cdot P_A^{(i)}$. Similarly we can conclude that $\deg_Z(P_A^{(i)}) = 0$.

From this we conclude that we must have

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

for some integer values $r_{i,j}$. From which it follows that

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

But the only way to produce a polynomial $P_C^{(i)}$ in the list $\mathcal{G}_1$, using our provided operations, which is divisible by a single power of X, would be to add together multiples of the polynomials associated with the σ_C and σ_D values output by the $\mathcal{O}^E$ oracle. In addition the queries to this oracle would have been associated to polynomials of the form $f_j + g_j \cdot Z$, otherwise they could not then be used in constructing the output signatures. This means we have values $s_{i,j}, t_{i,j}, f_j, g_j \in \mathbb{Z}_p$ such that

$$P_C^{(i)} = \sum_{j=1}^{q} \left(s_{i,j} \cdot A_j \cdot X \cdot (1 + f_j \cdot Y + g_j \cdot Y \cdot Z) + t_{i,j} \cdot A_j \cdot X \cdot Y \cdot Z \right). \tag{2}$$

Equating coefficients in equations (1) and (2) we find that we must have

$$\begin{aligned} r_{i,j} &= s_{i,j}, \\ r_{i,j} \cdot m_i &= s_{i,j} \cdot f_j, \\ 0 &= s_{i,j} \cdot g_j + t_{i,j}, \end{aligned}$$

for all values $1 \leq i \leq q_O + 1$ and $1 \leq j \leq q_O$. If, for any specific value of i, we have $s_{i,j} = 0$ for all values of j, then the ith output signature is of the form (A, B, C) where $A = 0$, which is disallowed. Hence, we must have, for each i, an index j_i such that $s_{i,j_i} \neq 0$. But this implies, by the above equations, that $m_i = f_{j_i}$, which itself implies that we must have two values i_0 and i_1 such that $m_{i_0} = m_{i_1}$. This contradicts the fact that all the messages should be distinct.

We conclude that the output of the adversary cannot correspond to a set of polynomial equations which hold for all possible variable assignments. Thus the adversary must win, or tell it is in a simulation, via a specific assignment to the variables.

Property Two: If the adversary learned it was interacting in a simulated game there would be two polynomials P_1 and P_2 on one of the lists such that $P_1 = P_2$ for a specific assignment of values to the variables. We note, that P_1 cannot equal P_2 identically by construction of the oracles. Since $\deg(P_1), \deg(P_2) \leq \mathsf{d_{max}}$ we find that the probability of an assignment being such that $P_1 = P_2$ is bounded by $\mathsf{d_{max}}/p$ [27][Lemma 1]. Thus the probability that this happens is bounded by $O(m^2 \cdot \mathsf{d_{max}}/p)$, since the size of the lists are bounded by m.

Property Three: In a similar way we need to bound the probability of the adversary outputing a set of elements, whose associated polynomials satisfy the required equations, for a specific assignment of values to the variables. We have that the total degree of the set of equations is bounded by $9q_O + 2 = \mathsf{d_{max}} + 1$, and we have $2 \cdot (q_O + 1)$ such equations to satisfy. Thus the probability of Property Three holding is bounded by $O((q_O/p)^q_O)$, i.e. vanishingly small.

In conclusion the probability that an adversary breaks the E-LRSW assumption in the GGM is bounded by $O(m^2 \cdot q_O/p)$.

7 Acknowledgements

The authors would like to thank the EU projects CACE and eCrypt-2 for partially funding the work in this paper. The second author was supported by the EPSRC under grant EP/H043454/1, and was also partially funded by a Royal Society Wolfson Merit Award, and Google Inc. The authors would like to thank Bogdan Warinschi for useful conversations whilst the work was carried out.

References

1. M. Abe. A secure three-move blind signature scheme for polynomially many signatures. In *Advances in Cryptology – Eurocrypt 2001*, Springer LNCS 2045 , 136–151, 2001.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology – Crypto 2010*, Springer LNCS 6223, 209–236, 2010.
3. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, **16(3)**, 185–215, 2003.
4. M. Bellare and P. Rogaway. The exact security of digital signatures - How to sign with RSA and Rabin. In *Advances in Cryptology – Eurocrypt '96*, Springer LNCS 1070, 399–416, 1996.
5. A. Boldyreva. Efficient threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-Group signature scheme. In *Public Key Cryptography – PKC 2003*, Springer LNCS 2567, 31–46, 2003.
6. D. Boneh and X. Boyen. Short signatures without random oracles. *Journal of Cryptology*, **21(2)**, 149-177, 2008.
7. D. Boneh, B. Lynn and H. Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, **17(4)**, 297–319, 2004.
8. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *Public Key Cryptography – PKC 2007*, Springer LNCS 4450, 1–15, 2007.
9. J. Camensich, M. Koprowski and B. Warinschi. Efficient blind signatures without random oracles. In *Security in Communication Networks – SCN 2004*, Springer LNCS 3352, 134–148, 2004.
10. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology – CRYPTO 2004*, Springer LNCS 3152, 56–72, 2004.
11. D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology – CRYPTO 1982*, Plenum Press, 199–203, 1983.
12. L. Chen, P. Morrissey and N.P. Smart. DAA: Fixing the pairing based protocols. IACR e-print 2009/198. `http://eprint.iacr.org/2009/198`.
13. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In *Advances in Cryptology – CRYPTO 2006*, Springer LNCS 4117, 60–77, 2006.
14. G. Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. IACR e-print 2009/320. `http://eprint.iacr.org/2009/320`.
15. S. Galbraith, K. Paterson and N.P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, **156**, 3113–3121, 2008
16. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Advances in Cryptology – EUROCRYPT 2008*, Springer LNCS 4965, 415–432, 2008.
17. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups (full version). `http://www.brics.dk/~jg/WImoduleFull.pdf`
18. A. Juels, M. Luby and R. Ostrovsky. Security of blind digital signatures. In *Advances in Cryptology – CRYPTO '97*, Springer LNCS 1294, 150–164, 1997.
19. A. Kiayias and H.-S. Zhou. Concurrent blind signatures without random oracles. In *Security and Cryptography for Networks – SCN06*, Springer LNCS 4116, 49–62, 2006.
20. Y. Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *Symposium on the Theory of Computation – STOC 2003*, ACM Press, 683–692, 2003.
21. Y. Lindell. Lower bounds for concurrent self composition. In *Theory of Cryptography Conference – TCC 2004*, Springer LNCS 2951, 203–222, 2004.
22. A. Lysyanskaya, R. Rivest, A. Sahai and S. Wolf. Pseudonym systems. In *Selected Areas in Cryptography – SAC 1999*, Springer LNCS 1758, 184–199, 1999.
23. U. Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding 2005*, Springer LNCS 3796, 1–12, 2005.
24. M. Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology – Crypto 2003*, Springer LNCS 2729, 96–109, 2003.

25. T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *Theory of Cryptography Conference – TCC 2006*, Springer LNCS 3876, 80–99, 2006.
26. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, **13(3)**, 361–396, 2000.
27. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology – Eurocrypt 1997*, Springer LNCS 1233, 256–266, 1997.