

# Efficient Round Optimal Blind Signatures

Sanjam Garg<sup>1,\*</sup> and Divya Gupta<sup>2</sup>

<sup>1</sup> IBM T. J. Watson  
sanjamg@cs.ucla.edu  
<sup>2</sup> UCLA  
divyag@cs.ucla.edu

**Abstract.** Known constructions of blind signature schemes suffer from at least one of the following limitations: (1) rely on parties having access to a common reference string or a random oracle, (2) are not round-optimal, or (3) are prohibitively expensive.

In this work, we construct the *first* blind-signature scheme that does not suffer from any of these limitations. In other words, besides being round optimal and having a standard model proof of security, our scheme is very efficient. Specifically, in our scheme, one signature is of size 6.5 KB and the communication complexity of the signing protocol is roughly 100 KB. An amortized variant of our scheme has communication complexity less than 1 KB.

## 1 Introduction

Blind signatures, introduced by Chaum [10], allow users to obtain signatures on messages of their choice without revealing the messages itself to the signer. Additionally, the blind signature scheme should satisfy unforgeability, i.e. no user can produce additional signatures on messages without interacting with the signer. Blind signatures have widespread applications such as e-cash, e-voting, and anonymous credentials.

Even after 30 years of research, and with 50+ candidate schemes in the literature, the state of the art is not completely satisfactory. Essentially, all schemes in the literature can be partitioned into two categories – (1) the schemes that rely on a random oracle or a setup, or (2) the schemes which are round inefficient. Examples of constructions argued to be secure using the random oracle methodology [7] include [26,27,25,1,5,8] and using a setup such as a shared random string include [4,3,11,13,21,23,22]. On the other hand, essentially all schemes that avoid the use of the random oracle methodology or a setup [20,9,23,19] are not round optimal.

The only scheme that does not fall in the above two categories is the recent construction of Garg et al. [16]. Unfortunately, this scheme is prohibitively expensive. For example, the communication complexity of this protocol is a

---

\* Research conducted while at the IBM Research, T.J.Watson funded by NSF Grant No.1017660.

**Table 1.** Comparing the Efficiency of Different Round Optimal Blind Signature Schemes.  $\kappa$  is the security parameter of the scheme.  $\epsilon > 1$  is an appropriate constant. The concrete parameters above correspond to the setting for 80 bits of security.

Scheme	Communication Complexity		Signature Size	
	Asymptotic	Concrete	Asymptotic	Concrete
Garg et al. [16]	$\text{poly}(\kappa)$		small <sup>2</sup>	
DLIN (This work)	$O(\kappa^{1+\epsilon})$	100.6KB	$O(\kappa^\epsilon)$	6.5KB
Amortized (This work)	$O(\kappa^\epsilon)$	836 Bytes	$O(\kappa^\epsilon)$	6.5KB
$q$ -SFP (This work)	$O(\kappa^{1+\epsilon})$	100.2KB	$O(\kappa^\epsilon)$	3.2KB
Amortized (This work)	$O(\kappa^\epsilon)$	472 bytes	$O(\kappa^\epsilon)$	3.2KB

large polynomial in the security parameter<sup>1</sup>. In this work, we ask the following question:

*Can we construct a very efficient round optimal blind signature scheme without relying on a random oracle or a setup?*

## 1.1 Our Results

We construct the *first* blind signature scheme that avoids all of the above limitations, namely it is very efficient, round optimal and does not rely on a random oracle or a setup. We obtain parameters for our scheme by using the concept of work factors from [14,6]. A summary of the results is highlighted in Table 1.

- **Standard Setting:** We assume the sub-exponential hardness of Decisional Linear (DLIN) Assumption and a variant of the discrete-log assumption. Then our signature scheme has one signature of size 6.5 KB and the communication complexity of signing protocol roughly 100 KB.
- **Amortized Setting:** A number of applications require a user to obtain multiple signatures from the same signer. In such a setting, for our scheme almost all of the communication costs can be avoided. More specifically an amortized variant of our scheme has communication cost roughly 100 KB when obtaining the first signature. However, for every subsequent signature obtained the communication cost is less than 1 KB.
- **Stronger assumption:** Assuming a stronger assumption, sub-exponentially hard  $q$ -Simultaneous Flexible Pairing Assumption (SFP) from [3], we can improve the size of a signature and the amortized communication complexity of our signing protocol by roughly a factor of 2.

<sup>1</sup> To give an estimate on how big this polynomial is, we instantiate the proofs being given in their construction with Dwork-Naor Zaps using Kilian-Petrank NIZKs and get communication complexity of at least  $O(\kappa^9)$  bits. One can also use asymptotically more efficient ZAPs instantiated with PCP based Groth NIZKs with ultimate proof size being  $O(\kappa^5 \text{poly log}(\kappa))$ . Note that  $\text{polylog}(\kappa)$  factor is quite large and for reasonable security parameters proof size would be comparable to  $O(\kappa^7)$ .

<sup>2</sup> This scheme uses general MPC techniques and can be instantiated using arbitrary signature scheme and thus has small signatures.

**Qualitative Improvements.** [16] uses *complexity leveraging* to obtain *standard model* round optimal blind signature scheme, and it is the use of these techniques which makes this scheme so inefficient. However, unfortunately, impossibility results of Fischlin et al. [12] and Pass [24] roughly indicate that the use of these techniques is essential for getting round optimal scheme in the standard model. Nonetheless, in this work, we introduce new techniques to *reduce* and *optimize* the use of complex leveraging, and thereby obtain a significantly more efficient scheme.

- **Reducing the use of complexity leveraging.** The technique of complexity leveraging works by creating a gap between the power of an adversarial entity and the reduction proving security. However, many a times this gap needs to be created multiple times in a layered fashion leading to larger parameters. The previous scheme of Garg et al. [16] needed to create this gap twice. However, in our scheme, we only need to create this gap once and this allows us to get smaller parameters.
- **Optimizing the use of complexity leveraging.** Complexity leveraging techniques (particularly for our application) inherently make non-black-box use of the underlying primitives. [16] in their construction end up rolling out the cryptographic primitive and viewing it as circuit. This leads to prohibitively inefficient schemes. We also make non-black-box use of the underlying primitive but avoid viewing it as a circuit. Instead, we cast it directly as a set of very structured equations which fit the framework of Groth-Sahai proofs, drastically improving the communication complexity of our protocol.

The techniques developed here are very general and we believe that they should be applicable to other settings. We leave this exploration for future work.

## 1.2 Technical Difficulties and New Ideas

Now we will describe the key ideas behind our scheme. We assume some familiarity with Groth-Sahai proofs. Lets us start by reminding the reader that GS proofs come in two modes – the hiding mode and the binding mode. In hiding mode, proofs reveal nothing about the witness used in the generation of a proof, and in binding mode, no fake proof exists.

**Starting Point.** The starting point for our construction is to use a blind signature scheme in the common reference string (CRS) model and remove the need for the CRS by letting the signer generate it. Of course this is problematic because a malicious signer can generate the CRS dishonestly (e.g. in a way such that it knows the trapdoors associated with the CRS) and use that to break the blindness property of the scheme. We solve this problem by using a special blind signature scheme for which blindness is statistical as long as the CRS is sampled from a certain “honest” distribution. In this setting, it is enough for the signer to prove that the CRS is sampled from the “honest” distribution. Looking ahead, this “honest” distribution is actually the CRS distribution for GS proofs in the hiding mode. However, we are faced with the following three issues.

**Issue 1)** First, in order to ensure blindness, the signer needs to prove to the user that the CRS was indeed sampled from the “honest” distribution.

**Issue 2)** Secondly, for proving unforgeability we will need that the reduction playing as the signer can “simulate” this proof. In other words, we need that the proof does not leak anything to the user.

**Issue 3)** The third issue is more subtle and arises as an interleaving of the first two issues. Specifically, the reduction for arguing unforgeability should be able to “extract” the messages on which the signatures are being issued and simulate the view of the attacking user. In other words, this extraction and simulation process should go unnoticed in the view of the attacking user. However, if a cheating signer could replicate the same behavior then this would go unnoticed as well. Hence, we certainly need to rule this out.

Before we describe our attempts to solve these issues, we note that for [16], this proof is the main reason for inefficiency.

**Attempt at Using Range Proofs.** As mentioned before, complexity leveraging makes non-black-box use of primitives essential making schemes prohibitively inefficient. In order to solve this issue we need to identify a problem such that: (1) the problem can be algebraically stated in groups of prime order  $p$  and has an efficient Groth-Sahai proof, (2) but solving the problem should be much easier than solving discrete log in the group of order  $p$ . The first property of the problem ensures efficiency of the proof. The second property as we will see later will be essential in making the complexity leveraging argument. We start by using a simple problem of solving discrete-log when the domain is restricted to some subspace. In particular, the problem we consider is: Given  $C = g^c$  such that  $c < q$  (where  $q \ll p$ ), one needs to find  $c$ . We then show that it satisfies both the above properties. In particular, we will show that this problem can be cast in the language of efficient Groth-Sahai proofs thus satisfying the first requirement. Secondly, improvement in the brute-force attack when the sample space is restricted to  $c < q$  is easy to see.

For the protocol, our idea is that user sends the value  $g^c$  for  $c < q$  to the signer. Further instead of having the signer prove that the CRS was sampled honestly we have him prove that either the CRS was honestly generated or that it is aware of  $c$ . This immediately solves our problems 1 and 2 from above. We know that a cheating signer will not be able to recover  $c$  and hence will not be able to cheat. At the same time we can have the reduction for unforgeability extract  $c$  and thereby generate simulated proofs.

However our solution to issues 1 and 2 has created a 4<sup>th</sup> issue. A cheating user may cheat by generating  $g^c$  such that  $c \geq q$ . Next, we will show how issues 3 and 4 can be solved.

**Solving Issue 3.** Very interestingly we can resolve issue 3 by requiring that the signer generates the proof above under the CRS he had sampled for the underlying blind signature scheme. This is very counter-intuitive as we are requiring the signer to generate a proof under a CRS that it generates on its own. The key idea is based on the observation that all we need is that the signer generates the

CRS from the hiding distribution for Groth-Sahai proofs. If this CRS is indeed hiding then the whole exercise of having a proof is redundant. On the other hand, if this CRS is actually generated dishonestly from the binding distribution then the signer is only hurting itself as it will not be able to generate his proof.<sup>3</sup>

**Solving Issue 4.** Recall that the 4<sup>th</sup> issue was that the user might generate  $g^c$  in a way such that  $c > q$ . We solve this problem by having the user provide a Groth-Sahai proof that the value  $c$  used is less than  $q$ . A question is under what CRS should this proof be given such that this proof does not leak  $c$  to the signer? Of course, we can not use the CRS that the signer generated for the underlying scheme. Our key observation here again is that we need to worry about this proof only if the original CRS has been generated maliciously, or in other words, if this CRS is binding. Recall that a binding CRS for Groth-Sahai proofs is a DLIN tuple. Our key idea here is that if  $(g, g_1, g_2, h_1, h_2, h)$  is a DLIN tuple then its shift  $(g, g_1, g_2, h_1, h_2, h \cdot g)$  can not be a DLIN tuple and hence the user can give his proof under this shifted CRS.<sup>4</sup>

## 2 Blind Signatures and Their Security

In this section we will recall the notion of blind signatures and define their security. Parts of this section have been taken verbatim from [16].

**Definition 1.** A blind signature scheme BS consists of PPT algorithms  $\text{Gen}, \text{Vrfy}$  along with interactive PPT algorithms  $\mathcal{S}, \mathcal{U}$  such that for any  $\lambda \in \mathbb{N}$ :

- $\text{Gen}(1^\lambda)$  generates a key pair  $(sk, vk)$ .
- The joint execution of  $\mathcal{S}(sk)$  and  $\mathcal{U}(vk, m)$ , where  $m \in \{0, 1\}^\lambda$ , generates an output  $\sigma$  for the user and no output for the signer. We write this as  $(\perp, \sigma) \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(vk, m) \rangle$ .
- Algorithm  $\text{Vrfy}(vk, m, \sigma)$  outputs a bit  $b$ .

We require **completeness** i.e., for any  $m \in \{0, 1\}^\lambda$ , and for  $(sk, vk) \leftarrow \text{Gen}(1^\lambda)$ , and  $\sigma$  output by  $\mathcal{U}$  in the joint execution of  $\mathcal{S}(sk)$  and  $\mathcal{U}(vk, m)$ , it holds that  $\text{Vrfy}(vk, m, \sigma) = 1$  with overwhelming probability in  $\lambda \in \mathbb{N}$ .

Blind signatures must satisfy unforgeability and blindness [20,28].

**Definition 2.** A blind signature scheme  $\text{BS} = (\text{Gen}, \mathcal{S}, \mathcal{U}, \text{Vrfy})$  is unforgeable if for any PPT algorithm  $\mathcal{U}^*$  the probability that experiment  $\text{Unforge}_{\mathcal{U}^*}^{\text{BS}}(\lambda)$  defined in Figure 1 evaluates to 1 is negligible in  $\lambda$ .

Blindness says that it should be infeasible for any malicious signer  $\mathcal{S}^*$  to decide which of two messages  $m_0$  and  $m_1$  has been signed first in two executions with

<sup>3</sup> In the final construction (Figure 2), the signer will prove under the CRS he had sampled that it is aware of  $c$ . An honest signer who generates a hiding CRS will be able to simulate this proof successfully.

<sup>4</sup> A similar idea was also used by [17] to get perfectly sound NIWI in the standard model using statistically sound NIZKs.

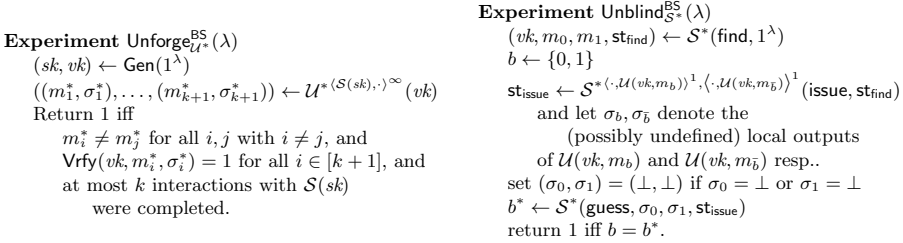


Fig. 1. Security games of blind signatures

an honest user  $\mathcal{U}$ . We define the advantage of  $\mathcal{S}^*$  in blindness game with respect to the experiment  $\text{Unblind}_{\mathcal{S}^*}^{\text{BS}}(\lambda)$  as

$$\text{Adv}_{\mathcal{S}^*, \text{BS}}^{\text{Unblind}}(\lambda) = \left| 2 \cdot \Pr[\text{Unblind}_{\mathcal{S}^*}^{\text{BS}}(\lambda) = 1] - 1 \right|$$

**Definition 3.** A blind signature scheme  $\text{BS} = (\text{Gen}, \mathcal{S}, \mathcal{U}, \text{Vrfy})$  satisfies blindness if the advantage function  $\text{Adv}_{\mathcal{S}^*, \text{BS}}^{\text{Unblind}}$  is negligible for any  $\mathcal{S}^*$  (working in modes find, issue, and guess) running in time  $\text{poly}(\lambda)$ .

A blind signature scheme is secure if it is unforgeable and blind.

### 3 Preliminaries

In this section, we recall and define basic notation and primitives used briefly. For a detailed description of primitives, see full version [15]. Let  $\lambda$  denote the security parameter. We call a function *negligible* in  $\lambda$  if it is asymptotically smaller than any inverse polynomial.

**Commitment Scheme on Groups.** We describe a perfectly binding commitment scheme based on the decisional linear (DLIN) assumption with the special property that both the message space and the commitment comprise only of group elements. Let  $(p, \mathbb{G}, \mathbb{G}_T, g, e)$  be a prime order bilinear pairing group. Then the function  $\text{Com}_{\mathbb{G}}(\cdot)$  generates a commitment to an element  $m \in \mathbb{G}$  by first sampling  $g_1, g_2 \xleftarrow{\$} \mathbb{G}, x, y \xleftarrow{\$} \mathbb{Z}_p$  and then outputting  $(g, g_1, g_2, g_1^x, g_2^y, m \cdot g^{x+y})$ .

**Structure-Preserving Signatures.** A signature scheme (SPGen, SPVerify) is said to be a *structure preserving signature scheme* over a prime order bilinear group  $(p, \mathbb{G}, \mathbb{G}_T, g, e)$ , if public keys, signatures and messages to be signed are vectors of group elements and verification only evaluates pairing product equations. Structure preserving signature schemes that sign a vector of group elements are known under different assumptions [17,3,2]. The first feasibility result was given by Groth [17]. This scheme is inefficient as the signature size grows linearly with the number of group elements in the message to be signed and the constants are quite big. In our scheme, we will use constant size structure preserving signatures [3,2]. Both of these results have been summarized in

**Table 2.** Efficiency of Structure Preserving Schemes

Scheme	$ msg $	$ gk  +  vk $	$ \sigma $	$\#(\text{PPE})$	Assumption
AHO10	$k$	$2k + 12$	7	2	q-SFP
ACDKNO12	$k$	$2k + 25$	17	9	DLIN

the table given below. The size of different parameters are in terms of number of group elements.

When  $k$  is a constant, a public key as well as a signature generated consist of a constant number of group elements only. Hence, these schemes are highly efficient for constant size messages. From the security of these schemes, it follows that under assumptions which are hard to break in time  $T \cdot \text{poly}(\lambda)$ , these schemes are secure against existential forgery under chosen message attack for adversaries running in time  $T \cdot \text{poly}(\lambda)$ . More precisely, these schemes are  $T$ -eu-cma-secure under hardness of  $T$ -q-SFP and  $T$ -DLIN, respectively.

### 3.1 Two-CRS Non-interactive Zero-Knowledge Proofs

In this section, we will define a special notion of NIZK proofs that work in the setting with two common reference strings.

Let  $R$  be an efficiently computable binary relation. For pairs  $(x, w) \in R$  we call  $x$  the statement and  $w$  the witness. Let  $L$  be the language consisting of statements in  $R$ . A *Two-CRS* non-interactive proof system for a relation  $R$  consists of three common reference string (CRS) generation algorithms  $K_B$ ,  $\text{Shift}$  and  $\text{Shift}^{-1}$ , a prover algorithm  $\mathcal{P}$  and a verification algorithm  $\mathcal{V}$ . We require that all these algorithms be efficient, i.e. polynomial time. The CRS generation algorithm  $K_B$  takes the security parameter  $1^\lambda$  as input and produces a common reference string  $\text{crs}$  along with an extraction key  $\tau$ . Both  $\text{Shift}$  and  $\text{Shift}^{-1}$  are deterministic algorithms. They take as input a string  $\text{crs}$  and output another string  $\text{crs}'$ . The prover algorithm  $\mathcal{P}$  takes as input  $(\text{crs}, x, w)$  and produces a proof  $\pi$ . The verification algorithm  $\mathcal{V}$  takes as input  $(\text{crs}, x, \pi)$  and outputs 1 or 0. We require that:

**CRS Indistinguishability.** For all PPT adversaries  $\mathcal{A}$ ,

$$\text{Adv}_{\mathcal{A}}^{\text{CRS-distinguish}}(1^\lambda) = 2 \cdot \Pr \left[ b = b' \left[ \begin{array}{l} (\text{crs}, \tau) \leftarrow K_B(1^\lambda); \text{crs}' \leftarrow \text{Shift}(\text{crs}); \text{crs}'' \leftarrow \text{Shift}^{-1}(\text{crs}) \\ b \xrightarrow{\$} \{0, 1\}; \text{ if } b = 0, (\text{crs}_1, \text{crs}_2) := (\text{crs}, \text{crs}') \\ \text{ else } (\text{crs}_1, \text{crs}_2) := (\text{crs}'', \text{crs}) \\ b' \leftarrow \mathcal{A}(\text{crs}_1, \text{crs}_2) \end{array} \right] - 1 \right].$$

We say that a Two-CRS NIZK system has CRS indistinguishability if for all PPT adversaries  $\mathcal{A}$ ,  $\text{Adv}_{\mathcal{A}}^{\text{CRS-distinguish}}$  is negligible in  $\lambda$ .

**Perfect Completeness.** Completeness requires that an honest prover with a valid witness can always make an honest verifier output 1. For  $K \in \{K_B, \text{Shift} \circ K_B, \text{Shift}^{-1} \circ K_B\}$ , where  $\circ$  is the the composition of functions, we require that for all  $x, w$  such that  $(x, w) \in R$ :

$$\Pr \left[ \mathcal{V}(\text{crs}, x, \pi) = 1 \mid \text{crs} \leftarrow K(1^\lambda); \pi \leftarrow \mathcal{P}(\text{crs}, x, w) \right] = 1.$$

**Perfect Knowledge Extraction.** We require that there exists a probabilistic polynomial time knowledge extractor  $\mathcal{E}$  such that for every  $(\text{crs}, \tau) \leftarrow K_B(1^\lambda)$ ,  $x$  and purported proof  $\pi$  such that  $\mathcal{V}(\text{crs}, x, \pi) = 1$  then we have

$$\Pr \left[ (x, w) \in R \mid w := \mathcal{E}(\text{crs}, \tau, x, \pi) \right] = 1.$$

Note that since perfect knowledge extraction implies the existence of a witness for the statement being proven, it implies *perfect soundness*.

**Perfect Zero-Knowledge.** A proof system is zero-knowledge if the proofs do not reveal any information about the witnesses. We require that there exists a polynomial time simulator  $\mathcal{S}$  such that for all  $(\text{crs}, \tau) \leftarrow K_B(1^\lambda)$ ,  $\text{crs}' := \text{Shift}(\text{crs})$  (or,  $\text{crs}' := \text{Shift}^{-1}(\text{crs})$ ) we have that for all  $x \in L$  the distributions  $\mathcal{P}(\text{crs}', x, w)$  and  $\mathcal{S}(\text{crs}', \tau, x)$  are identical.

**Efficient Realization of Two-CRS NIZKs Based on Groth-Sahai Proofs.**

Groth-Sahai proofs [18] can be used to give efficient Two-CRS NIZKs (under the DLIN assumption) for special languages, namely pairing product equations, multi-scalar multiplication equations, and quadratic equations (described below) in the setting of symmetric bilinear groups. We also show that the range equations also fit this framework. For details, refer to the full version [15].

- **Pairing Product Equation.** A pairing product equation (PPE) over the variables  $X_1, \dots, X_n \in \mathbb{G}$  is an equation of the form<sup>5</sup>

$$\prod_{i=1}^n e(\mathcal{A}_i, X_i) \cdot \prod_{i=1, j \geq i}^{n, n} e(X_i, X_j)^{\gamma_{i,j}} = 1,$$

determined by constants  $\mathcal{A}_i \in \mathbb{G}$  and  $\gamma_{i,j} \in \mathbb{Z}_p$ .

- **Multiscalar Multiplication Equation.** A multiscalar multiplication equation over the variables  $X_1, \dots, X_n \in \mathbb{G}$  and  $y_1, y_2, \dots, y_m \in \{0, 1\}$  is of the form

$$\prod_{j=1}^m \mathcal{A}_j^{y_j} \cdot \prod_{i=1}^n X_i^{b_i} \cdot \prod_{i=1}^n \prod_{j=1}^m X_i^{\gamma_{i,j} y_j} = \mathcal{T},$$

determined by constants  $\mathcal{A}_j \in \mathbb{G}$ ,  $b_i, \gamma_{i,j} \in \mathbb{Z}_p$ , and  $\mathcal{T} \in \mathbb{G}$ .

---

<sup>5</sup> General form of PPE can have any  $\mathcal{T} \in \mathbb{G}_T$  on the R.H.S. Since GS NIZKs are only known for PPE having 1 on the R.H.S., we use only such equations in our construction.



- **Quadratic Equation.** A quadratic equation in  $\mathbb{Z}_p$  over variables  $y_1, y_2, \dots, y_n \in \{0, 1\}$  is of the form

$$\sum_{i=1}^n a_i y_i + \sum_{i=1, j \geq i}^{n, n} \gamma_{i,j} y_i y_j = t,$$

determined by constants  $a_i \in \mathbb{Z}_p, \gamma_{i,j} \in \mathbb{Z}_p,$  and  $t \in \mathbb{Z}_p.$

- **Range Equation.** The range equation over the variable  $c \in \mathbb{Z}_p$  is of the form.

$$\exists c : g^c = C \wedge c < q,$$

determined by constants  $C \in \mathbb{G}$  and  $q < p.$  We note that the range equation is not explicitly a part of the Groth-Sahai framework but is implied by it.

*Remark 1.* We note that for the first three kinds of equations, under the above mentioned realization of Two-CRS NIZKs, the proof size grows only linearly with the number of variables and the number of equations. This follows directly from the GS proofs as explained in the full version [15].

*Remark 2.* As shown in the full version [15], a range equation can be expressed as one multiscalar multiplication equation and  $\log_2 q$  quadratic equations over  $\log_2 q$  variables in  $\mathbb{Z}_p.$

## 4 Blind Signature Scheme: Construction

We begin by giving an informal description of the scheme. In our scheme, we will use a bilinear group  $\mathbb{G}$  of prime order  $p,$  a structure preserving signature scheme for signing vectors of elements in this group, Two-CRS NIZKs, and commitment scheme  $\text{Com}_{\mathbb{G}}.$

During the key generation phase, the signer generates the verification key  $vk$  and the secret key  $sk$  for the blind signature scheme as follows.  $vk$  consists of a verification key  $vk_{\text{SP}}$  for the structure preserving signature scheme, two CRSes,  $\text{crs}_1$  and  $\text{crs}_2$  under Two-CRS NIZK proof system, and a parameter  $q = p^\epsilon$  for some constant  $\epsilon \in (0, 1).$   $\text{crs}_1$  is sampled from  $K_B$  and  $\text{crs}_2$  is set to be the shifted  $\text{crs}_1,$  i.e.  $\text{crs}_2 \leftarrow \text{Shift}(\text{crs}_1).$   $sk$  consists of the signing key  $sk_{\text{SP}}$  corresponding to  $vk_{\text{SP}}$  and the extraction key  $\tau$  for  $\text{crs}_1.$

Next, the two round blind signature scheme proceeds as follows: In the first round, the user generates its message as follows: It begins by checking whether  $\text{crs}_2$  equals  $\text{Shift}(\text{crs}_1).$  It aborts, if this is not the case. Next, it blinds its message  $m$  by generating a commitment  $m_{\text{blind}}$  using  $\text{Com}_{\mathbb{G}}$  under randomness  $r.$  Then, it samples a random  $c < q$  and sets  $C = g^c.$  Finally, it generates a proof  $\pi$  under  $\text{crs}_1$  for the NP-statement  $\Phi: \exists c \mid g^c = C \wedge c < q.$  It sends  $(m_{\text{blind}}, C, \pi)$  as the first round message to the signer.

In the second round, the signer generates its message as follows: It begins by checking if the proof  $\pi$  is valid under  $\text{crs}_1.$  It aborts, if this is not the case. Next, it extracts the witness  $c$  from the proof  $\pi$  using extraction key  $\tau.$  Then it

Recalling from Section 3, let  $(\text{SPGen}, \text{SPSign}, \text{SPVerify})$  be an existentially unforgeable structure preserving signature scheme,  $(K_B, \text{Shift}, \text{Shift}^{-1}, \mathcal{P}, \mathcal{V})$  be a Two-CRS NIZK proof system and  $\text{Com}_{\mathbb{G}}$  be a group based commitment scheme. And let  $0 < \epsilon < 1$  be an appropriate (specified later) constant parameter.

**Key Generation Gen:** On input  $1^\lambda$ , choose an appropriate bilinear group  $(p, \mathbb{G}, \mathbb{G}_T, g, e)^a$  and proceed as follows:

- Sample a key pair for the structure preserving signature scheme  $(\text{sk}_{\text{SP}}, \text{vk}_{\text{SP}}) \leftarrow \text{SPGen}(1^\lambda)$ .
- Sample a CRS  $(\text{crs}_1, \tau) \leftarrow K_B(1^\lambda)$  and generate its shift  $\text{crs}_2 \leftarrow \text{Shift}(\text{crs}_1)$ .
- Output the verification-key for the blind signature scheme as  $\text{vk} = (\text{vk}_{\text{SP}}, \text{crs}_1, \text{crs}_2, q = p^\epsilon)$  and the secret-key as  $\text{sk} = (\text{sk}_{\text{SP}}, \tau)$ .

**Signing Protocol:** The user  $\mathcal{U}$  with input  $m \in \mathbb{G}$ ,  $\text{vk}_{\text{SP}}$  and the signer  $\mathcal{S}$  with input  $\text{sk}_{\text{SP}}$  proceed as follows.

- **Round 1:** The user  $\mathcal{U}$  generates its first message as follows:
  - Abort if  $\text{crs}_2 \neq \text{Shift}(\text{crs}_1)$ .
  - Sample  $m_{\text{blind}} \leftarrow \text{Com}_{\mathbb{G}}(m; r)$ .
  - Samples a uniformly random  $c$  such that  $c < q$  and sets  $C := g^c$ . Next sample a proof  $\pi \leftarrow \mathcal{P}(\text{crs}_1, \Phi, c)$  where  $\Phi$  is the NP-statement:

$$\exists c \mid g^c = C \wedge c < q. \tag{1}$$

- Send  $(m_{\text{blind}}, C, \pi)$  to the signer.
- **Round 2:**  $\mathcal{S}$  generates the second round message as:
  - If  $\mathcal{V}(\text{crs}_1, \Phi, \pi) \neq 1$  then abort, otherwise obtain  $c := \mathcal{E}(\text{crs}_1, \tau, \Phi, \pi)$  and sample a proof  $\pi' \leftarrow \mathcal{P}(\text{crs}_2, \Phi, c)$ .
  - Sample a signature  $\sigma_{\text{SP}} := \text{SPSign}(\text{sk}_{\text{SP}}, m_{\text{blind}})$ .
  - Send  $(\pi', \sigma_{\text{SP}})$  to the user  $\mathcal{U}$ .
- **Signature Generation:**  $\mathcal{U}$  aborts if  $\mathcal{V}(\text{crs}_2, \Phi, \pi') \neq 1$ .  $\mathcal{U}$  also aborts if  $\text{SPVerify}(\text{vk}_{\text{SP}}, m_{\text{blind}}, \sigma_{\text{SP}}) \neq 1$  and otherwise outputs  $\sigma \leftarrow \mathcal{P}(\text{crs}_2, \Psi, (m_{\text{blind}}, r, \sigma_{\text{SP}}))$  where  $\Psi$  is the NP-statement:

$$\exists (m_{\text{blind}}, r, \sigma_{\text{SP}}) \mid m_{\text{blind}} = \text{Com}_{\mathbb{G}}(m; r) \wedge \text{SPVerify}(\text{vk}_{\text{SP}}, m_{\text{blind}}, \sigma_{\text{SP}}) = 1 \tag{2}$$

**Signature Verification Vrfy:** For input a claimed signature  $\sigma$  on message  $m$ , output  $\mathcal{V}(\text{crs}_2, \Psi, \sigma)$ .

<sup>a</sup> All algorithms take this bilinear group as an implicit input.

**Fig. 2.** Blind Signature Scheme

generates a fresh proof  $\pi'$  for the statement  $\Phi$  under  $\text{crs}_2$ . Finally, it generates a signature  $\sigma_{\text{SP}}$  on  $m_{\text{blind}}$  using signing key  $\text{sk}_{\text{SP}}$ . It sends  $(\pi', \sigma_{\text{SP}})$  as the second round message to the user.

On receiving the above message from the user, it computes the signature on  $m$  as follows: User aborts if  $\pi'$  is not a valid proof under  $\text{crs}_2$ . It then checks

if  $\sigma_{\text{SP}}$  is a valid signature on  $m_{\text{blind}}$  under  $\text{vk}_{\text{SP}}$ . It aborts if this is not the case. Otherwise, it outputs  $\sigma$  as the proof under  $\text{crs}_2$  of the NP-statement  $\Psi$ :  $\exists (m_{\text{blind}}, r, \sigma_{\text{SP}}) \mid m_{\text{blind}} = \text{Com}_{\mathbb{G}}(m; r) \wedge \text{SPVerify}(\text{vk}_{\text{SP}}, m_{\text{blind}}, \sigma_{\text{SP}}) = 1$ . In other words, the user proves that there exists  $(m_{\text{blind}}, r, \sigma_{\text{SP}})$  such that  $m_{\text{blind}}$  is the commitment of  $m$  using randomness  $r$  under commitment scheme  $\text{Com}_{\mathbb{G}}$  and  $\sigma_{\text{SP}}$  is a valid signature on  $m_{\text{blind}}$ .

To verify a signature  $\sigma$  on message  $m$ , check whether  $\sigma$  is a valid proof for the statement  $\Psi$  under  $\text{crs}_2$ .

**Formal Description.** Let  $\text{SPSig} = (\text{SPGen}, \text{SPSign}, \text{SPVerify})$  be any structure preserving signature scheme which is existentially unforgeable,  $(K_B, \text{Shift}, \text{Shift}^{-1}, \mathcal{P}, \mathcal{V})$  be a Two-CRS NIZK proof system,  $\text{Com}_{\mathbb{G}}$  be the DLIN based commitment scheme for elements in  $\mathbb{G}$  (Section 3). Formal description of the blind signature scheme  $(\text{Gen}, \mathcal{S}, \mathcal{U}, \text{Vrfy})$  is given in Figure 2.

### 5 Proof of Unforgeability

Let  $\mathbf{T}_{\mathbb{G},q}^{\text{dlog}}$  be the time it takes to break the discrete log problem in  $\mathbb{G}$  when exponents are chosen from  $\mathbb{Z}_q$ .

**Theorem 1.** *For any PPT malicious user  $\mathcal{U}^*$  for the unforgeability game against the blind signature scheme given in Section 4 the following holds:*

$$\text{Adv}_{\mathcal{U}^*, \text{BS}}^{\text{Unforge}}(\lambda) \leq \text{Adv}_{\mathcal{B}}^{\text{CRS-distinguish}}(\lambda) + \text{Adv}_{\widehat{\mathcal{U}}^*, \text{SPSig}}^{\text{Unforge}}(\lambda),$$

where  $\mathcal{B}$  is an adversary against the CRS indistinguishability property of the two-CRS NIZK proof system such that  $\mathbf{T}(\mathcal{B}) = k \cdot \mathbf{T}_{\mathbb{G},q}^{\text{dlog}} + \mathbf{T}(\mathcal{U}^*) + \text{poly}(\lambda)$  and  $\widehat{\mathcal{U}}^*$  is the adversary against the unforgeability of the underlying structure preserving signature scheme  $\text{SPSig}$  such that  $\mathbf{T}(\widehat{\mathcal{U}}^*) = k \cdot \mathbf{T}_{\mathbb{G},q}^{\text{dlog}} + \mathbf{T}(\mathcal{U}^*) + \text{poly}(\lambda)$ . Also,  $\mathcal{U}^*$  and  $\widehat{\mathcal{U}}^*$  make at most  $k$  signing queries.

If we use GS proof system based Two-CRS NIZKs in our construction, the above theorem immediately implies the following corollary:

**Corollary 1.** *For any PPT malicious user  $\mathcal{U}^*$  for the unforgeability game against the blind signature scheme given in Section 4 the following holds:*

$$\text{Adv}_{\mathcal{U}^*, \text{BS}}^{\text{Unforge}}(\lambda) \leq 2 \cdot \text{Adv}_{\mathcal{B}, \mathbb{G}}^{\text{dlin}}(\lambda) + \text{Adv}_{\widehat{\mathcal{U}}^*, \text{SPSig}}^{\text{Unforge}}(\lambda),$$

where  $\mathcal{B}$  is an adversary against the DLIN assumption in  $\mathbb{G}$  such that  $\mathbf{T}(\mathcal{B}) = k \cdot \mathbf{T}_{\mathbb{G},q}^{\text{dlog}} + \mathbf{T}(\mathcal{U}^*) + \text{poly}(\lambda)$  and  $\widehat{\mathcal{U}}^*$  is the adversary against the unforgeability of the underlying structure preserving signature scheme  $\text{SPSig}$  such that  $\mathbf{T}(\widehat{\mathcal{U}}^*) = k \cdot \mathbf{T}_{\mathbb{G},q}^{\text{dlog}} + \mathbf{T}(\mathcal{U}^*) + \text{poly}(\lambda)$ . Also,  $\mathcal{U}^*$  and  $\widehat{\mathcal{U}}^*$  make at most  $k$  signing queries.

Following is a corollary of the above theorem:

**Theorem 2.** *Assume that  $\mathbf{T}_{\mathbb{G},q}^{\text{dlog}}$ -DLIN holds in  $\mathbb{G}$  and  $\text{SPSig}$  is  $\mathbf{T}_{\mathbb{G},q}^{\text{dlog}}$ -eu-cma-unforgeable. Then the blind signature scheme in Section 4 is unforgeable.*

*Proof.* (of Theorem 1) Let  $\mathcal{U}^*$  be any PPT malicious user then we will prove our theorem by considering a sequence of games starting with the unforgeability game from Definition 2 (see Section 2).

- **Game<sub>0</sub>**: This is the challenger-adversary game between the challenger following the honest signer  $\mathcal{S}$  specification and the malicious user  $\mathcal{U}^*$ . More specifically, the game starts with the challenger generating a key pair  $(sk, vk)$ . The challenger then sends  $vk$  to  $\mathcal{U}^*$ . At this point the challenger (playing as the honest signer) and  $\mathcal{U}^*$  proceed by interacting in  $k$  executions of the signing protocol. Note that the challenger knows the secret key  $sk$  and uses it to participate as the signer in the executions of the signing protocol. Finally  $\mathcal{U}^*$  outputs  $k + 1$  message/signature pairs  $(m_i, \sigma_i)$ .  $\mathcal{U}^*$  is said to win if all the messages are distinct and all signatures verify under  $vk$ .
- **Game<sub>1</sub>**: Recall that in the second round of the signing protocol the challenger (acting as the signer) obtains the secret value  $c$  using the extraction algorithm  $\mathcal{E}$ . **Game<sub>1</sub>** is same as the **Game<sub>0</sub>** except that in each of the  $k$  instances of the signing protocol, instead of extracting the secret  $c$  using the extraction algorithm, the challenger obtains  $c$  by evaluating the discrete log of  $C$  assuming that it is less than  $q$ . (The challenger aborts if no values less than  $q$  is a valid dlog of  $C$ .)

Note that since  $\text{crs}_1$  is sampled from  $K_B$ , proofs under  $\text{crs}_1$  are perfectly sound. This implies that the value  $c$  that challenger extracts by solving discrete log is exactly the same as the one that challenger would have extracted using the extraction algorithm in **Game<sub>0</sub>**.

Note that the views of the malicious user  $\mathcal{U}^*$  in games **Game<sub>0</sub>** and **Game<sub>1</sub>** are identical.

It also follows from the perfect soundness of the two-CRS NIZK proof system that the challenger in **Game<sub>1</sub>** runs in time  $k \cdot \mathbf{T}_{\mathbb{G},q}^{\text{dlog}} + \text{poly}(\lambda)$ , where  $\mathbf{T}_{\mathbb{G},q}^{\text{dlog}}$  is the time it takes to break discrete log problem in  $\mathbb{G}$  when the exponent is chosen from  $\mathbb{Z}_q$ .

- **Game<sub>2</sub>**: **Game<sub>2</sub>** is same as **Game<sub>1</sub>** except that the challenger generates the CRSes differently. Instead of generating the CRSes by first sampling  $(\text{crs}_1, \tau) \leftarrow K_B(1^\lambda)$  and then generating its shift  $\text{crs}_2 \leftarrow \text{Shift}(\text{crs}_1)$ , it reverses the order in which the CRSes are generated. This reverses the security properties of proofs under the two CRSes. More specifically the challenger first samples  $(\text{crs}_2, \tau) \leftarrow K_B(1^\lambda)$  and then sets  $\text{crs}_1 := \text{Shift}^{-1}(\text{crs}_2)$ . Note that now we get perfect zero-knowledge for  $\text{crs}_1$  and perfect soundness for  $\text{crs}_2$ .

Indistinguishability of **Game<sub>1</sub>** and **Game<sub>2</sub>** follows from the CRS-Indistinguishability property of the two-CRS NIZK proof system. More precisely, the success probability of  $\mathcal{U}^*$  can change by at most  $\text{Adv}_{\mathcal{B}}^{\text{CRS-distinguish}}$ , where  $\mathcal{B}$  is an adversary against the CRS indistinguishability property of the two-CRS NIZK proof system such that  $\mathbf{T}(\mathcal{B}) = k \cdot \mathbf{T}_{\mathbb{G},q}^{\text{dlog}} + \mathbf{T}(\mathcal{U}^*) + \text{poly}(\lambda)$ .

Now we will show how  $\mathcal{U}^*$  who wins in **Game<sub>2</sub>** can be used to construct a malicious user  $\widehat{\mathcal{U}}^*$  that winning the existential unforgeability game of the underlying structure preserving signature scheme.

$\widehat{U}^*$  starts by obtaining the verification key  $\text{vk}_{\text{SP}}$  from the challenger of the structure preserving signature scheme  $(\text{SPGen}, \text{SPSign}, \text{SPVerify})$ . Furthermore, it samples  $(\text{crs}_2, \tau) \leftarrow K_B(1^\lambda)$ , sets  $\text{crs}_1 := \text{Shift}^{-1}(\text{crs}_2)$  and invokes  $U^*$  with  $(\text{vk}_{\text{SP}}, \text{crs}_1, \text{crs}_2, q)$  as input. At this point, the user  $U^*$  expects to interact in  $k$  instances of the signing protocol. In each of these executions, it provides its challenger (the adversary  $\widehat{U}^*$  in our case) with its first round message  $(m_{\text{blind}}, C, \pi)$ . Our adversary  $\widehat{U}^*$  obtains  $c$  by solving the discrete log problem (aborting if  $c \geq q$ ) and uses the extracted value to generate the response proof  $\pi'$ . Additionally, it obtains the signature  $\sigma_{\text{SP}}$  on  $m_{\text{blind}}$  from the signing oracle and passes  $(\pi', \sigma_{\text{SP}})$  to  $U^*$ . After  $k$  such executions,  $U^*$  returns  $k + 1$  pairs  $(m_j, \sigma_j)$ . Note that each  $\sigma_j$  given by  $U^*$  is a proof of knowledge of  $(m_{\text{blind},j}, r_j, \sigma_{\text{SP},j})$  under  $\text{crs}_2$ . Furthermore, since  $\widehat{U}^*$  generates  $\text{crs}_2$  in the binding setting, therefore  $\tau$  can be used to extract  $(m_{\text{blind},j}, r_j, \sigma_{\text{SP},j})$  for each  $j$  by invoking  $\mathcal{E}(\text{crs}_2, \tau, \Psi, \sigma_j)$ . Since all messages  $m_j$  are distinct and  $\text{Com}_{\mathbb{G}}$  is perfectly binding, all  $m_{\text{blind},j}$  will also be distinct. Since all  $m_{\text{blind},j}$  are distinct there exists at least one  $m_{\text{blind},j^*}$  among these that  $\widehat{U}^*$  never queried its challenger.  $\widehat{U}^*$  outputs  $(m_{\text{blind},j^*}, \sigma_{\text{SP},j^*})$  as its output.

Hence, the advantage of  $U^*$  in producing a valid forgery in  $\text{Game}_3$  is at most the advantage of  $\widehat{U}^*$  in producing a valid forgery against the underlying structure preserving signature scheme, i.e.  $\text{Adv}_{U^*, \text{BS}, \text{Game}_3}^{\text{Unforge}} \leq \text{Adv}_{\widehat{U}^*, \text{SPSig}}^{\text{Unforge}}(\lambda)$ , where  $\widehat{U}^*$  runs in time  $k \cdot \mathbf{T}_{\mathbb{G},q}^{\text{dlog}} + \mathbf{T}(U^*) + \text{poly}(\lambda)$ .

## 6 Proof of Blindness

**Theorem 3.** *For any PPT malicious signer  $S^*$  for the blindness game against the blind signature scheme given in Section 4, which successfully completes the blindness game, the following holds*

$$\text{Adv}_{S^*, \text{BS}}^{\text{Unblind}}(\lambda) < 2 \cdot \text{Adv}_{\mathcal{A}, \text{Com}_{\mathbb{G}}}^{\text{hid}} + \text{Adv}_{\mathcal{B}, \mathbb{G}, q}^{\text{dlog}}$$

where  $\mathcal{A}$  is an adversary against the non-uniform hiding property of  $\text{Com}_{\mathbb{G}}$  such that  $\mathbf{T}(\mathcal{A}) = \mathbf{T}(S^*) + \text{poly}(\lambda)$  and  $\mathcal{B}$  is an adversary against the non-uniform discrete log problem in  $\mathbb{G}$  when exponents are chosen uniformly randomly in  $\mathbb{Z}_q$  such that  $\mathbf{T}(\mathcal{B}) = \mathbf{T}(S^*) + \text{poly}(\lambda)$ .

Since the hiding property of the  $\text{Com}_{\mathbb{G}}$  holds under the DLIN assumption in  $\mathbb{G}$ , the above theorem immediately implies the following corollary.

**Corollary 2.** *For any PPT malicious signer  $S^*$  for the blindness game against the blind signature scheme given in Section 4, which successfully completes the blindness game, the following holds*

$$\text{Adv}_{S^*, \text{BS}}^{\text{Unblind}}(\lambda) < 4 \cdot \text{Adv}_{\mathcal{C}, \mathbb{G}}^{\text{DLIN}} + \text{Adv}_{\mathcal{B}, \mathbb{G}, q}^{\text{dlog}}$$

where  $\mathcal{C}$  is an adversary against the non-uniform DLIN assumption in  $\mathbb{G}$  such that  $\mathbf{T}(\mathcal{C}) = \mathbf{T}(S^*) + \text{poly}(\lambda)$  and  $\mathcal{B}$  is an adversary against the non-uniform

discrete log problem in  $\mathbb{G}$  when exponents are chosen uniformly randomly in  $\mathbb{Z}_q$  such that  $\mathbf{T}(\mathcal{B}) = \mathbf{T}(\mathcal{S}^*) + \text{poly}(\lambda)$ .

Following is a corollary of the above.

**Theorem 4.** *Assume that non-uniform DLIN assumption holds in  $\mathbb{G}$  and the non-uniform discrete log assumption holds in  $\mathbb{G}$  even when the exponents are chosen uniformly randomly from  $\mathbb{Z}_q$ . Then the blind signature scheme from Section 4 is blind.*

*Proof.* (of Theorem 3) Let  $\mathcal{S}^*$  be any PPT malicious signer then we will prove our theorem by considering a sequence of games starting with the blindness game from Definition 3 (see Section 2).

- **Game<sub>0</sub>**: This is a challenger-adversary game between the challenger following the honest user strategy and the malicious signer  $\mathcal{S}^*$ . The malicious signer  $\mathcal{S}^*$  has full control over the scheduling of instances of the user in an arbitrary order. Since our scheme is only two round, we can fix it to be the worst case ordering. Since  $\mathcal{S}^*$  does not receive any response to the message it sends to the user, we can assume that  $\mathcal{S}^*$  first gathers all the incoming messages from the user and then sends its responses. Thus, without loss of generality, the **Game<sub>0</sub>** proceeds as follows:  $\mathcal{S}^*$  first outputs the public key  $vk$  and the challenge messages  $m_0, m_1$ .  $\mathcal{S}^*$  then expects the two incoming blinded messages  $m_{blind,0}$  and  $m_{blind,1}$  from the user corresponding to  $m_b, m_{1-b}$  for a random bit  $b$ . After receiving both the messages,  $\mathcal{S}^*$  outputs its responses to the challenger. Our challenger at this point outputs the signature on  $(m_0, m_1)$  generated in the two protocol executions. Finally the malicious signer  $\mathcal{S}^*$  outputs a bit  $b'$  and its advantage  $\text{Adv}_{\mathcal{S}^*, \text{BS}}^{\text{Unblind}}$  is equal to  $|2 \cdot \Pr[b = b'] - 1|$ .
- **Game<sub>1</sub>**: Same as **Game<sub>0</sub>** except the following: The challenger after receiving the public key  $vk$ , figures out whether  $\text{crs}_2$  is in the range of  $K_B$  or not. The challenger may execute in unbounded time when figuring this out; storing the extraction key  $\tau$  for later use. Now it proceeds as follows:
  - $\text{crs}_2$  is in the range of  $K_B$ : In this case, our challenger proceeds just as in **Game<sub>0</sub>**, except that if the first instance of the signing protocol completes successfully then our challenger outputs **DL-Abort**.
  - $\text{crs}_2$  is not in the range of  $K_B$ : Proceed as in **Game<sub>0</sub>**.

Note that conditioned on the fact that **DL-Abort** does not happen, we have that **Game<sub>0</sub>** and **Game<sub>1</sub>** are identical. Next we will show that the probability of **DL-Abort** happening is bounded by  $\text{Adv}_{\mathcal{B}, \mathbb{G}, q}^{\text{dlog}}$ .

**Lemma 1.** *The probability of DL-Abort happening is bounded by  $\text{Adv}_{\mathcal{B}, \mathbb{G}, q}^{\text{dlog}}$ , with  $\mathbf{T}(\mathcal{B}) = \mathbf{T}(\mathcal{S}^*) + \text{poly}(\lambda)$ ,  $\mathcal{B}$  is an adversary against the non-uniform discrete log problem in  $\mathbb{G}$  when exponents are chosen uniformly randomly in  $\mathbb{Z}_q$ .*

*Proof.* We will show that an  $\mathcal{S}^*$  that can make our challenger output **DL-Abort** can be used to construct an adversary  $\mathcal{B}$  that breaks the non-uniform discrete log problem in  $\mathbb{G}$  when the exponent is restricted to  $< q$ .

*Constructing the adversary  $\mathcal{B}$ .* Given this cheating signer  $\mathcal{S}^*$ , there exists random coins for  $\mathcal{S}^*$  such that our challenger in **Game-1** outputs **DL-Abort**. We will hard-code the random coins of  $\mathcal{S}^*$  such that our challenger outputs **DL-Abort** with maximum probability. Note that we are in the case when  $\text{crs}_2$  is binding and hence  $\text{crs}_1$  is hiding. Next, our adversary  $\mathcal{B}$  or the challenger of the blindness game on receiving this public key  $vk$  will run in unbounded time to compute the extraction key  $\tau$  for  $\text{crs}_2$ . Thus, the adversary  $\mathcal{B}$  we constructed is a non-uniform adversary with auxiliary input as the random coins of  $\mathcal{S}^*$  (specified above) and the extraction key  $\tau$  corresponding to  $vk$ . Our adversary  $\mathcal{B}$  obtains as input  $D$  (such that  $D = g^d$  with  $d < q$ ) and it wins if it outputs  $d$ . On receiving  $D$ ,  $\mathcal{B}$  proceeds as the challenger does in **Game<sub>1</sub>** except that it sets  $C := D$  instead of choosing a fresh value for  $C$ . Also, invoking perfect zero-knowledge property of  $\text{crs}_1$ ,  $\mathcal{B}$  generates  $\pi$  as  $\mathcal{S}(\text{crs}_1, \tau, \Phi)$ , where  $\mathcal{S}$  is the zero-knowledge simulator. At this point  $\mathcal{S}^*$  must output a proof  $\pi'$  such that  $\mathcal{V}(\text{crs}_2, \Phi, \pi') = 1$  for the challenger in **Game<sub>1</sub>** to output **DL-Abort**. On obtaining the proof  $\pi'$ ,  $\mathcal{B}$  outputs  $\mathcal{E}(\text{crs}_2, \tau, \Phi, \pi')$  as the discrete log of  $D$ . By perfect extraction under  $\text{crs}_2$ , the extracted value will be the discrete log of  $D$ .

Note that after receiving the challenge  $D$ ,  $\mathcal{B}$  runs in polynomial time. Thus, the probability of **DL-Abort** when we fix the worst case random coins of  $\mathcal{S}^*$  (as described above) is bounded by  $\text{Adv}_{\mathcal{B}, \mathbb{G}, q}^{\text{dlog}}$ . Hence, it holds that the probability of **DL-Abort** in **Game<sub>1</sub>** is bounded by  $\text{Adv}_{\mathcal{B}, \mathbb{G}, q}^{\text{dlog}}$ .

- **Game<sub>2</sub>**: **Game<sub>2</sub>** is identical to **Game<sub>1</sub>** except for the following modifications. Instead of generating the final signatures honestly, the challenger simulates them. More specifically, instead of generating the signatures as  $\mathcal{P}(\text{crs}_2, \Psi, (m_{\text{blind}}, r, \sigma_{\text{SP}}))$ , in **Game<sub>2</sub>** the challenger generates signatures as  $\mathcal{S}(\text{crs}_2, \tau, \Psi)$ . **Game<sub>2</sub>** and **Game<sub>1</sub>** are perfectly indistinguishable based on the non-uniform perfect zero-knowledge property of the two-CRS NIZK proof system.
- **Game<sub>3</sub>**: Now, we modify **Game<sub>2</sub>** and remove all dependencies on the input messages  $m_0$  and  $m_1$ . That is, we let the user algorithm compute the blinded message  $m_{\text{blind}, 0}$  as  $\text{Com}_{\mathbb{G}}(0)$  instead of  $\text{Com}_{\mathbb{G}}(m_b)$ . We proceed similarly for  $m_{1-b}$ .  
The indistinguishability between **Game<sub>3</sub>** and **Game<sub>2</sub>** follows from the non-uniform computational hiding property of the commitment scheme  $\text{Com}_{\mathbb{G}}$ .

In **Game<sub>3</sub>** the entire transcript is independent of the message:  $\text{Adv}_{\mathcal{S}^*, \text{BS}, \text{Game}_3}^{\text{Unblind}} = 0$ .

## 7 Concrete Efficiency

In this section we will compute the communication complexity and the size of the final blind signature for our scheme. First we need to compute the group size  $p$  and number  $q$  which will give us the desired level of security. For this we will calculate the work factors for different adversaries as discussed below.

**WORK FACTORS.** These have been used in [14,6] to calculate concrete parameters. This text has been taken verbatim from [6]. For any adversary running

in time  $\mathbf{T}(\mathcal{A})$  and gaining advantage  $\epsilon$ , we define the *work factor* of  $\mathcal{A}$  to be  $\mathbf{WF}(\mathcal{A}) \leq \mathbf{T}(\mathcal{A})/\epsilon$ . The ratio of  $\mathcal{A}$ 's running time to its advantage provides a measure of efficiency of the adversary. Generally speaking, to resist an adversary with work factor  $\mathbf{WF}(\mathcal{A})$ , a scheme should have its security parameter (bits of security) be  $\kappa \geq \log \mathbf{WF}(\mathcal{A})$ . Note that for a particular  $\epsilon$ , this means a run time of  $\mathbf{T}(\mathcal{A}) \leq \epsilon 2^\kappa$ .

Similar to [14,6], in the discussion that follows we will assume that Pollard Rho's algorithm for finding discrete logs in  $\mathbb{G}$  is the best known attack<sup>6</sup> against DLIN in group  $\mathbb{G}$  of prime order  $p$ . The work factor of Pollard's algorithm is

$$\mathbf{WF}(\mathcal{P}) = \frac{T(\mathcal{P})}{\epsilon_p} = \frac{0.88}{e} \sqrt{p} \frac{\log^2(p)}{10^3}$$

For security we require that the work factor of any adversary  $\mathcal{A}$  against DLIN is at most the work factor of Pollard's algorithm, i.e.  $\mathbf{WF}(\mathcal{A}) \leq \mathbf{WF}(\mathcal{P})$ .

**Parameters.** In the full version [15], we calculate the values of  $p$  and  $q$  using the work factors for adversaries against the blindness game and unforgeability game. We summarize the parameters obtained in Table 3.

**Table 3.** Suggested parameters, where  $k$  is the number of signature queries and the adversary is allowed to run in time  $t \cdot T_R$  where  $T_R$  is the time taken by the reduction.

$k$	$t$	$\log q$	$\log  \mathbb{G} $
$2^{20}$	$2^{30}$	155	291
$2^{20}$	$2^{40}$	155	311
$2^{30}$	$2^{30}$	155	331
$2^{30}$	$2^{40}$	155	351

## 7.1 Efficiency

**Verification Key Size.** In our blind signature scheme, the verification key is  $vk = (vk_{\text{SP}}, \text{crs}_1, \text{crs}_2, q = p^\epsilon)$ , where  $vk_{\text{SP}}$  is the verification key of the structure preserving signature scheme in  $\mathbb{G}$  and  $\text{crs}_1$  and  $\text{crs}_2$  are two CRSes for Two-CRS NIZK. Furthermore, as can be seen in Table 2, to sign  $k$  group elements,  $vk_{\text{SP}}$  has  $2k + 25$  group elements. Since in our case  $k = 6$ , there are 37 group elements in  $vk_{\text{SP}}$ . In GS proof system, we need 6 group elements in  $\mathbb{G}$  to represent  $\text{crs}_1$  and  $\text{crs}_2$ . Hence, the size of the verification key for our scheme is 43 group elements. Taking the number of bits to represent a group element as 291 bits, we get the key size to be 1.6KB.

**Signature Size.** The final signature is a Groth-Sahai [18] proof of knowledge in  $\mathbb{G}$  using  $\text{crs}_2$  as the common reference string. Under the DLIN assumption, the proof size is three group elements for each variable and nine group elements for each pairing product equation (see Figure 2 in [18]) that is proved. The variables

<sup>6</sup> If there is a faster attack against discrete log or DLIN problem for prime order groups, it can be used to obtain the parameters for our blind signature scheme.



are  $m_{blind}, \sigma_{SP}, r$ . By  $\text{Com}_{\mathbb{G}}$ ,  $m_{blind}$  has six group elements and in order to prove  $m_{blind} = \text{Com}_{\mathbb{G}}(m; r)$ , we will have two additional variables (which capture the randomness  $r$  used in commitment) and three pairing product equations in total. Furthermore, as can be seen in Table 2,  $\sigma_{SP}$  has 17 group elements and nine pairing product equations in verification algorithm. Hence, the size of the final blind signature will be 183 group elements in  $\mathbb{G}$ . Taking the number of bits to represent a group element as 291 bits, we get the signature size to be 6.5KB.

**Communication Complexity.** We begin by computing the communication complexity of the user step by step as follows:

- $\mathcal{U}$  computes a commitment  $m_{blind}$  in  $\mathbb{G}$  which consists of six group elements.
- It computes a range proof  $\pi$  for an NP-statement which consists of  $\log_2 q$  quadratic equations and one multiscalar multiplication equation over  $\log_2 q$  variables in  $\mathbb{Z}_p$  (Remark 2). In GS proof system, each quadratic equations adds six group elements, multiscalar multiplication equation adds nine group elements and each variable in  $\mathbb{Z}_p$  adds three group elements to the proof ([18], Figure 2). Using this,  $\pi$  consists of  $9 \log_2 q + 9$  group elements of  $\mathbb{G}$ .

Now we compute the communication complexity of signer as follows:

- It computes  $\sigma_{SP}$  consisting of 17 elements in  $\mathbb{G}$  as explained above.
- It also computes a range proof  $\pi'$  for the same NP-statement as the user. As above,  $\pi'$  consists of  $9 \log_2 q + 9$  group elements of  $\mathbb{G}$ .

Hence, the overall communication complexity of our blind signature protocol is  $18 \log_2 q + 41$  elements in  $\mathbb{G}$ . Taking  $\log_2 q$  as 155 and  $\log_2 p$  as 291, the communication complexity is 100.56KB.

**Acknowledgements.** We thank Jens Groth for useful discussions relating to this work. We also thank the anonymous reviewers of EUROCRYPT 2014 for their insightful comments and an observation that helped improve the communication complexity of our signing protocol from 200 KB to 100 KB.

## References

1. Abe, M.: A secure three-move blind signature scheme for polynomially many signatures. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 136–151. Springer, Heidelberg (2001)
2. Abe, M., Chase, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 4–24. Springer, Heidelberg (2012)
3. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on elements in bilinear groups for modular protocol design. IACR ePrint 2010/133 (2010)
4. Abe, M., Ohkubo, M.: A framework for universally composable non-committing blind signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 435–450. Springer, Heidelberg (2009)

5. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The power of rsa inversion oracles and the security of chaum's rsa-based blind signature scheme. In: Syverson, P.F. (ed.) FC 2001. LNCS, vol. 2339, pp. 309–328. Springer, Heidelberg (2002)
6. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: Simplified proof and improved concrete security for waters' ible scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (2009)
7. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
8. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)
9. Camenisch, J.L., Koprowski, M., Warinschi, B.: Efficient blind signatures without random oracles. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 134–148. Springer, Heidelberg (2005)
10. Chaum, D.: Blind signatures for untraceable payments. In: CRYPTO, pp. 199–203 (1982)
11. Fischlin, M.: Round-optimal composable blind signatures in the common reference string model. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 60–77. Springer, Heidelberg (2006)
12. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer, Heidelberg (2010)
13. Fuchsbauer, G.: Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. IACR ePrint 2009/320 (2009)
14. Galindo, D.: The exact security of pairing based encryption and signature schemes. In: Based on a talk at Workshop on Provable Security, INRIA, Paris (2004), <http://www.dgalindo.es/galindoEncrypt.pdf>
15. Garg, S., Gupta, D.: Efficient round optimal blind signatures. Cryptology ePrint Archive, Report 2014/081 (2014), <http://eprint.iacr.org/>
16. Garg, S., Rao, V., Sahai, A., Schröder, D., Unruh, D.: Round optimal blind signatures. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 630–648. Springer, Heidelberg (2011)
17. Groth, J.: Simulation-sound nizek proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
18. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
19. Hazay, C., Katz, J., Koo, C.-Y., Lindell, Y.: Concurrently-secure blind signatures without random oracles or setup assumptions. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 323–341. Springer, Heidelberg (2007)
20. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (1997)
21. Kiayias, A., Zhou, H.-S.: Concurrent blind signatures without random oracles. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 49–62. Springer, Heidelberg (2006)

22. Meiklejohn, S., Shacham, H., Freeman, D.M.: Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 519–538. Springer, Heidelberg (2010)
23. Okamoto, T.: Efficient blind and partially blind signatures without random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 80–99. Springer, Heidelberg (2006)
24. Pass, R.: Limits of provable security from standard assumptions. In: STOC, pp. 109–118 (2011)
25. Pointcheval, D.: Strengthened security for blind signatures. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 391–405. Springer, Heidelberg (1998)
26. Pointcheval, D., Stern, J.: Provably secure blind signature schemes. In: Kim, K.-C., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 252–265. Springer, Heidelberg (1996)
27. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 387–398. Springer, Heidelberg (1996)
28. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. *Journal of Cryptology* 13(3), 361–396 (2000)