Conditional Blind Signatures

Alexandros Zacharakis, Panagiotis Grontas, and Aris Pagourtzis

School of Electrical and Computer Engineering
National Technical University of Athens, 15780 Athens, Greece
azach@corelab.ntua.gr, pgrontas@corelab.ntua.gr, pagour@cs.ntua.gr

Abstract. We propose a novel cryptographic primitive called *conditional blind signatures*. Our primitive allows a user to request blind signatures on messages of her choice. The signer has a secret Boolean input which determines if the supplied signature is valid or not. The user should not be able to distinguish between valid and invalid signatures. A designated verifier, however, can tell which signatures verify correctly, and is in fact the only entity who can learn the secret input associated with the (unblinded) signed message. We instantiate our primitive as an extension of the Okamoto-Schnorr blind signature scheme and provide variations to fit different usage scenarios. Finally, we analyze and prove the security properties of the new scheme and explore potential applications.

Keywords: digital signatures, blind signatures, designated verifier signatures

1 Introduction

Digital signatures, proposed in [1], are one of the most successful public key cryptographic primitives. A *user* $\mathcal{U}$ submits a message to a *signer* $\mathcal{S}$, who applies a function of his secret signing key sk and generates a signature that can be verified by everybody that possesses the corresponding public verification key vk. They allow message integrity, authenticity and non repudiation in a publicly verifiable manner. A digital signature scheme is secure if no probabilistic adversary $\mathcal{A}$, running in polynomial time (PPT), can output a forgery of a signature, i.e. a valid signature without possessing the signing key. Instantiations of digital signatures schemes base their security on well known cryptographic problems such as the RSA problem [2], the Discrete Log problem ([7] [10]) with its many variations and many more. Moreover, in [5] a method is given to construct digital signatures from interactive proofs of knowledge.

Digital signatures are quite versatile, as attested by the plethora of variations that have been proposed in the literature. Indeed many useful schemes can arise, if one fiddles with the basic setting of a digital signature scheme. For instance, blind signatures [3], hide the message to be signed from the signer, thus allowing the user to maintain her privacy while keeping the signature publicly verifiable. The security of blind signatures has been studied in [14], [17] and [20]. The relevant security properties are *blindness* or *unlinkability*, which models the fact that the signer cannot have access to the message, and resistance to *one more forgery*, which states that the user cannot herself create more signatures than the signer provided. Blind signatures have many important applications such as electronic cash [3] and electronic voting [9].

Another variation of digital signatures is group signatures [8]. They aim to provide *signer anonymity within a group*. This means that the signature is validated as coming from the group as a whole, without giving evidence as to which member of the group actually signed. Of course in the case of a dispute, the *traceability* property allows the group manager to specify which group member actually signed.

A less studied primitive related to signatures via zero knowledge proofs, are *designated verifier proofs*, proposed in [12]. They sacrifice the public verifiability of a proof and propose a scheme where its validity can only be verified by an entity that has a specific piece of knowledge (e.g. a private key). This entity is designated by the prover. Such a scheme might seem of no particular use, but this is not the case, since in [12], the authors propose a very interesting application in the context of receipt free and coercion resistant electronic voting. In particular, they propose that a voting authority use designated verifier proofs, in order to convince a voter that her ballot was correctly counted. However this proof is only verifiable by the voter herself and not by any third party. As a result it cannot be presented voluntarily or involuntarily to a coercer.

1.1 Motivation

In this paper we aim to create a primitive that can be used as a building block for protocols that require strong guarantees for coercion resistance and privacy. Such a primitive can be used, for example, in auction and payment systems, but the primary application we have in mind is remote electronic voting, where the lack of a controlled environment for vote casting, leaves the voters vulnerable to coercion attacks. The most well known way to defeat such attacks was proposed in the JCJ framework [19]. Its main idea, stems from the fact that the coercer has no incentive to carry out his attack if he cannot tell whether it has succeeded or not. This can be achieved, if we allow the voter to cast multiple ballots, by attaching a different but indistinguishable anonymous credential to each vote. One of these credentials is *valid* and it is used to cast the vote when the coercer is not present - JCJ assumes that each voter has a moment of privacy. Only the votes cast with valid credentials are included in the election tally. In order to filter out the invalid credentials the authors of [19] propose a quadratic number of checks in the number of votes cast. Such a complexity is not practical for real large scale elections.

A more practical solution, would involve a signer that uses voter identification information to efficiently retrieve the valid credential and check it against the one that is provided during the voting process. If the credentials are different, then the voter is under coercion and the vote should not be counted. This bit of information has to be conveyed to the counter in a manner undetectable by the coercer. Of course the signer should not have any access to the contents of the vote to maintain voter privacy. As a result the signatures have to be blind, as well. A well known voting scheme built on blind signatures was given in [9], but it is not coercion resistant. What is needed, is a primitive that can integrate the coercion resistance property of [19] and the increased privacy guarantees of [9].

1.2 Our contribution

Our approach is based on the observation that a combination of a simple group signature scheme with a designated verifier proof can be used to convey a piece of secret information from a signer to a specified verifier. For instance if we imagine the group members, as possible responses to the message to be signed, a designated group signature is equivalent to sending a particular response to the verifier. This can be achieved even when the message is blinded, thus enabling the signer to authorize a secret message and to simultaneously attach an extra bit of information to it. To this end, we propose a new primitive, called *conditional blind signatures*, that implements this functionality. We define its security properties, by extending blindness and unforgeability - the standard properties of blind signatures - with a new property, *conditional verifiability*, that incorporates the extra bit of information to the validation procedure. Moreover, we provide an instantiation that is based on the well known Okamoto-Schnorr blind signatures [10]. We use our definitions and the particular instantiation to provide proofs for the security properties. Despite the fact that the motivation behind our primitive is specific, we believe that it can stand on its own and be used in many applications apart from electronic voting.

1.3 Related work

On a conceptual level our scheme bears similarities to *designated confirmer signatures* (DCS) [11] and *conditional disclosure of secrets* (CDS) [16].

Indeed, designated confirmer signatures were proposed as a combination of digital signatures and zero knowledge proofs, to solve a problem of undeniable signatures [6]. In undeniable signatures, if the signer becomes unavailable during the verification process, the signature cannot be validated. To fix this, a DCS scheme adds a third party to the protocol, a designated confirmer, that can also verify (confirm) the signature. Designated confirmer signatures have been studied extensively since their introduction and many variations have been proposed (e.g. see [21] and references therein).

On the other hand, conditional disclosure of secrets was proposed as a way for a client to obtain a secret held by a server if and only if the input of the client satisfies a certain condition. The client may hold a secret key and encrypt the input using the corresponding public key that is known to the server. In [18], a CDS scheme is used to build a protocol that enables a buyer (client) to purchase items from a vendor (server), without disclosing which item is bought. The secret here is the particular item that is purchased and the condition that must be met for disclosure is that the encrypted price sent by the buyer matches the price of the item on sale. Of course all comparisons are made on encrypted inputs by utilizing the homomorphic properties of the underlying cryptosystem. More recent work on CDS [22] has been able to use them as lightweight zero knowledge proofs.

Our proposal, resembles DCS in the basic usage scenario, since in our case, as well, the verifier can be a third party which is 'designated' during the signature creation process. However, the problem we want to solve is the secure passing of a single bit of information to the verifier through the signature and not the unavailability of the signer. This might seem similar to CDS, with the signer playing the role of the server and the client playing the role of the designated verifier. Our scheme, however, allows for a third participant, namely the user requesting signatures. This enables us to also use it in cases where the signer and the verifier have no conflict of interest, as is typically the case with CDS, and even in scenarios where the client can be the same entity as the server at a later time. Moreover, the most important difference of our scheme with both related proposals is that the user messages are perfectly blinded, so that the signer, contrary to DCS, cannot have access to their contents. This perfect blindness, in contrast to CDS combined with encryption schemes, results in messages signed with conditional blind signatures, being private even in an information theoretical sense. All these make conditional blind signatures a novel and interesting primitive.

2 Preliminaries

In this section we briefly review the necessary concepts for the construction and security analysis of our proposal.

2.1 Security Assumptions

The security of our scheme depends on the Computational Diffie Hellman (CDH) and the Decisional Diffie Hellman (DDH) assumptions, as we show in section 5. Informally, the CDH assumption states that given a group $\mathbb{G}$, a generator g and two group elements g^a, g^b,the group element g^{ab} cannot be efficiently computed. The DDH assumption states that the triples of group elements (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) where a, b, c are randomly selected from $\{1, \cdots, |\mathbb{G}|\}$ cannot be efficiently distinguished.

More formally, following [15], let G be a group family and g a generator of a particular member $\mathbb{G}$ of G.

Definition 1. Computational Diffie Hellman *Assumption.*
A CDH *algorithm A is a probabilistic polynomial time algorithm satisfying:*

$$\Pr[A(g, g^a, g^b) = g^{ab}] > \frac{1}{\lambda^k}$$

for some fixed $k > 0$ and sufficiently large n, where the probability is taken over the selection of $\mathbb{G}$, a, b and the random bits of A. The group family satisfies the CDH *assumption if there is no* CDH *algorithm for it.*

Definition 2. Decisional Diffie Hellman *Assumption.*
A DDH *algorithm A is a probabilistic polynomial time algorithm satisfying:*

$$|\Pr[A(g, g^a, g^b, g^{ab}) = 1] - \Pr[A(g, g^a, g^b, g^c) = 1]| > \frac{1}{\lambda^k}$$

for some fixed $k > 0$ and sufficiently large n, where the probability is taken over the selection of $\mathbb{G}$, a, b, c and the random bits of A. The group family satisfies the DDH *assumption if there is no* DDH *algorithm for it.*

There are many groups where the Decisional Diffie Hellman Assumption is believed to hold [15]. One such group is the q order subgroup of quadratic residues in $\mathbb{Z}_p^*$ where $q, p = 2q + 1$ are primes.

2.2 Okamoto-Schnorr Blind Signatures

In section 4 we provide an instantiation of our scheme built on the Okamoto-Schnorr blind signatures. For completeness, we repeat their definition here from [10].

The public parameters of the protocol are a group $\mathbb{G}$ with prime order q, two generators g_1, g_2 and a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_q$. The signer $\mathcal{S}$ selects the private signing key $s_1, s_2 \in \mathbb{Z}_q$ and computes the public verification key $v := g_1^{-s_1} g_2^{-s_2}$. The user $\mathcal{U}$ wants to sign the message m. The protocol is executed in the following phases:

1. In the commitment phase, $\mathcal{S}$ randomly selects $r_1, r_2 \in \mathbb{Z}_q$ and commits to the value $x := g_1^{r_1} g_2^{r_2}$.
2. In the blinding phase, $\mathcal{U}$ selects the blinding factors $u_1, u_2, d \in \mathbb{Z}_q$ and computes the following values:
   - $x^* := g_1^{u_1} g_2^{u_2} v^d x$
   - $e^* := \mathcal{H}(m, x^*)$
   - $e := e^* - d \bmod q$

   Finally she sends the value of e to $\mathcal{S}$.
3. In the signing phase $\mathcal{S}$ computes the values $y_1 := r_1 + es_1 \bmod q$ and $y_2 := r_2 + es_2 \bmod q$. The blind signature is (x, e, y_1, y_2).
4. In the unblinding phase $\mathcal{U}$ computes the values $y_1^* := y_1 + u_1 \bmod q$ and $y_2^* := y_2 + u_2 \bmod q$. The plain signature is (x^*, e^*, y_1^*, y_2^*).
5. To verify the signature the following two relations are checked:
   - $e^* = \mathcal{H}(m, x^*)$
   - $x^* = g_1^{y_1^*} g_2^{y_2^*} v^{e^*}$

3 Definitions for Conditional Blind Signatures

Our new primitive can be abstractly viewed as a protocol implementing the following functionality f:

$$\text{signature} := f(b, sk, pk, c)$$

where:

- b is the secret information to be conveyed from the signer to the verifier. We restrict the secret information to a single bit.
- sk is the signing key.
- pk is the corresponding public key.
- c is the message to be signed.

The participants of the protocol are:

- The user $\mathcal{U}$ is the entity that requests blind signatures on messages of her choice.
- The signer $\mathcal{S}$ is the entity that creates the signatures on the message provided by the user. The signer wants to use the signature to convey the secret information to the verifier.
- The verifier $\mathcal{V}$ is the entity that checks the validity of the signatures and learns the secret information of $\mathcal{S}$. The verifier can be the signer himself at a future time.

The adversary $\mathcal{A}$ may be any entity except the designated verifier. This means that, apart from external attackers, both the signer and the user may have incentive to attack our scheme. For instance, as the signer is presented with a blinded message, he might want to learn its contents. The user, or an agent acting on her behalf, on the other hand, might want to retrieve the signer's secret information.

The desired security properties of our scheme extend the security properties of digital and blind signatures:

- The signatures given must be perfectly blind.
- The scheme must be secure against one more forgery.
- No PPT adversary can check the validity of the produced signatures nor can he extract the secret information, but with probability negligible in relation to a security parameter.

Concretely, our primitive can be defined as follows:

Definition 3. *A conditional blind signature scheme is a triple* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *with the following properties:*

- Gen *is an algorithm that takes as input the security parameter* 1^λ *and outputs two pairs of keys* $(sk_\mathcal{S}, pk_\mathcal{S})$ *for signing and* $(sk_\mathcal{V}, pk_\mathcal{V})$ *for verification, the message space* $\mathbb{M}$ *and the signature space* $\mathbb{S}$*. These sets are described by a set of parameters (e.g. group generators) collectively denoted as* params*. We also refer to the set of public keys as* $pk = (pk_\mathcal{S}, pk_\mathcal{V})$ *and to the set of secret keys as* $sk = (sk_\mathcal{S}, sk_\mathcal{V})$*.*
- $\mathsf{Sign}(\text{params}, pk) = \langle \mathcal{S}(sk_\mathcal{S}, b), \mathcal{U}(m) \rangle$ *is a protocol executed between the signer and the user. The public input to the signing protocol consists of the parameters and the public keys. The secret input of the signer is the signing key* $sk_\mathcal{S}$ *and the secret information bit* b*, while the secret input of the user is the message* m *to be signed. The protocol outputs a signature* sig *of* m *to* $\mathcal{U}$*.*

- Vrfy *is an algorithm which on input* $(sk_{\mathcal{V}}, m, sig)$ *outputs a single bit representing the validity of the signature.*

We require that correctness holds, that is $\mathsf{Vrfy}(sk_{\mathcal{V}}, m, sig)$ *outputs* 1 *if and only if sig is the output of the execution of the protocol* Sign *on message* m *and the secret information bit of* $\mathcal{S}$ *is* $b = 1$*, except with negligible probability.*

Conditional blind signatures inherit the Blindness and the security against One More Forgery properties from the conventional blind signatures schemes. We extend the respective definitions of [20], to accommodate for the secret bit b.

We first formally define the blindness property using the BlindExp game presented in algorithm 1, which states that a malicious signer cannot tell which of the two messages m_0, m_1 was signed first except with negligible probability. In particular, at first the (adaptive) adversary, operating in **find** mode, is allowed to (maliciously) generate the public parameters pk and the messages m_0, m_1. Subsequently, in **issue** mode he is given two blinded versions of the messages in random order, according to the value of c. These correspond to two executions of the Sign protocol. If these interactions produce results, denoted by (σ_0, σ_1), he tries to guess which signature corresponds to which message. Since, the secret bit b can be used to distinguish between two messages, we restrict the adversary in issuing the two signatures with the same secret bit, which is provided as input. The BlindExp game returns 1 if and only if he succeeds.

Algorithm 1: $\mathrm{BlindExp}_{\mathcal{A},\Pi}$

```
Input  : security parameter $\lambda$, secret bit $b$
Output: $o \in \{0,1\}$
$(pk, m_0, m_1) \leftarrow \mathcal{A}(\mathbf{find}, 1^{\lambda})$
$c \leftarrow_R \{0,1\}$
$(\sigma_0, \sigma_1) \leftarrow \mathcal{A}^{\langle \cdot, U(m_c) \rangle, \langle \cdot, U(m_{1-c}) \rangle}(\mathbf{issue}, b)$
if $\sigma_0 = \perp \vee \sigma_1 = \perp$ then
    $(\sigma_0, \sigma_1) := (\perp, \perp)$
else
    $c^* \leftarrow \mathcal{A}(\mathbf{guess}, \sigma_0, \sigma_1)$
end
if $c = c^*$ then
    return 1
else
    return 0
end
```

Definition 4. *A blind signature scheme Π is perfectly blind if for every (unbounded) $\mathcal{A}$:*

$$\Pr[\mathrm{BlindExp}_{\mathcal{A},\Pi}(\lambda)=1]=\frac{1}{2}$$

The unforgeability property is captured using the notion of *One More Forgery* [17]. It states, that if l is an integer, polynomial in the security parameter λ, an attacker cannot produce $l+1$ valid signatures, after fewer than l successful interactions with the signer. The *Strong* One More Forgery [17] is a variation of the above case, where l is *polylogarithmically* bound to the security parameter. As far as the secret bit is concerned, invalid signatures might assist the aspiring forger, so we allow him to get signatures with a b of his choice for each invocation of the signing protocol. A more formal description is given in algorithm 2:

Algorithm 2: $\mathrm{OneMoreForge}_{\mathcal{A},\Pi}$

```
Input  : security parameter $\lambda$
Output: $o \in \{0,1\}$
$(sk, pk) \leftarrow \mathrm{Gen}(1^{\lambda})$
$\{(m_i, \sigma_i) \leftarrow \mathcal{A}^{\langle S(sk_{\mathcal{S}}, b_i), \cdot\rangle}(pk)\}_{i=1}^{l+1}$
/* $k$: number of successful interactions for the Sign protocol      */
if $(\forall i, j \; \textit{with} \; i \neq j \Rightarrow m_i \neq m_j) \wedge (\forall i \;\; \mathrm{Vrfy}(vk, m_i, \sigma_i) = 1) \wedge k \leq l$ then
    return 1
else
    return 0
end
```

Definition 5. *A blind signature scheme Π is one more unforgeable if for every* PPT $\mathcal{A}$ *there is a negligible function of λ where:*

$$\Pr[\mathrm{OneMoreForge}_{\mathcal{A},\Pi}(\lambda)=1]\leq negl(\lambda)$$

Additionally, for our primitive we require an extra property which is called *Conditional Verifiability* and defined in the game CondVerExp presented in algorithm 3.

In the particular game, after the parameter generation the adversary $\mathcal{A}$ executes the Sign protocol in place of the user and adaptively gets valid and invalid signatures on messages of his choice. Then he creates a challenge message and receives a valid or invalid signature on this message based on a coin flip. Afterwards he tries to determine the value of the coin flip. He may continue to get signatures of his choice. The scheme is secure with respect to conditional verifiability if there is no PPT adversary who can succeed in guessing the result of the coin flip with non negligible advantage. More formally:

Algorithm 3: $\mathrm{CondVerExp}_{\mathcal{A},\Pi}$

```
Input  : security parameter $\lambda$
Output: $o \in \{0,1\}$
$c \leftarrow_R \{0,1\}$
$(sk, pk, \text{params}) \leftarrow \mathrm{Gen}(1^\lambda)$
$\{(m_i, sig_i) \leftarrow \mathrm{Sign}\langle S(sk_{\mathcal{S}}, b_i), \mathcal{A}(\text{params}, pk, \{m_j, sig_j\}_{j=1}^{i-1}, b_i)\rangle\}_{i=1}^{l_1}$
$m_c \leftarrow \mathcal{A}(\text{params}, pk, \{(m_i, sig_i)\}_{i=1}^{l_1}, \textbf{challenge})$
$(\epsilon, sig_c) \leftarrow \mathrm{Sign}\langle S(sk, c), \mathcal{A}(\text{params}, pk, m_c)\rangle$
/* $\epsilon$ is the empty string                                      */
$\{(m_i, sig_i) \leftarrow \mathrm{Sign}\langle S(sk_{\mathcal{S}}, b_i), \mathcal{A}(\text{params}, pk, \{m_j, sig_j\}_{j=1}^{i-1}, b_i)\rangle\}_{i=l_1+1}^{l_2}$
$c' \leftarrow \mathcal{A}(\{m_i, sig_i\}_{i=1}^{l_1+l_2}, m_c, sig_c, \textbf{guess})$
if $c = c'$ then
    return 1
else
    return 0
end
```

Definition 6. *A conditional blind signature scheme Π has the Conditional Verifiability property if for each* PPT *adversary $\mathcal{A}$ there is a negligible function* negl *with regard to the security parameter λ such that:*

$$\Pr[\mathrm{CondVerExp}_{\mathcal{A},\Pi} = 1] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

Definition 7. *A conditional blind signature scheme Π is secure if it has the properties* Perfect Blindness*,* One More Forgery *and* Conditional Verifiability*.*

We must note here that the user cannot validate the signature she receives, since she does not have access to the secret bit b. Although this seems counter intuitive with respect to traditional signatures, in our setting it is the exact property we want to capture.

4 An instantiation based on Okamoto-Schnorr Blind Signatures

In this section we propose an instantiation of our scheme based on the Okamoto-Schnorr Blind Signatures [10]. The intuition behind our construction is that we replace the elements (y_1, y_2) of the Okamoto-Schnorr blind signature with a 'lifted' signature (k^{y_1}, y_2) where k is some element of the underlying group with logarithm known only to the verifier.

Firstly we define the key generation algorithm for the 3 participating entities, namely the user $\mathcal{U}$, the signer $\mathcal{S}$ and the verifier $\mathcal{V}$. The details are presented in algorithm 4.

For the signing protocol we assume a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_q$, which is modelled as a random oracle. The protocol begins as in the Okamoto-Schnorr

Algorithm 4: Key Generation Algorithm

```
Input  : security parameter $\lambda$
Output: $(sk_S, vk_S), (sk_V, vk_V), params$
/* Select a group $\mathbb{G}$ with prime order $q$ where $2^{\lambda-1} \leq q < 2^{\lambda}$ where the
   DDH assumption holds                                                  */
$(q, \mathbb{G}) \leftarrow \text{GroupGen}(1^{\lambda})$
/* Select the appropriate generators                                     */
$(g_1, g_2) \leftarrow_R \mathbb{G}$
$\text{params} := (q, \mathbb{G}, g_1, g_2)$
/* Select the secret $sk_S$ and public signing keys $vk_S$ for $\mathcal{S}$          */
$s_1, s_2 \leftarrow_R \mathbb{Z}_q$
$v := g_1^{-s_1} g_2^{-s_2}$
$(sk_S, vk_S) := ((s_1, s_2), v)$
/* Select secret $sk_V$ and public verification keys $vk_V$ for $\mathcal{V}$      */
$s \leftarrow_R \mathbb{Z}_q$
$k := g_1^{s}$
$(sk_V, pk_V) := (s, k)$
```

blind signatures [10]. The signer commits to a random value. The user selects the blinding factors and blinds the commitment and the hash to be signed. Our variation actually begins when the signer is ready to sign the blinded values. We consider 2 cases:

- If the hidden bit of $\mathcal{S}$ is 1, instead of generating the standard Okamoto-Schnorr tuple, the signer raises the public key of the verifier to the first part of the signature.
- If the hidden bit of $\mathcal{S}$ is 0, then the signature is invalidated merely by selecting a random element from the underlying group.

In both cases, the second part of the tuple is calculated in the standard way. The details are given in Figure 1. Note that the unblinding of the first part of the signature, occurs on the exponent.

For the verification algorithm, the verifier checks the verification equation using the hash of the message and the commitment. If the secret signer bit is 1, then the signature will be valid, otherwise the verification equation will not hold. Thus the verifier will learn the secret bit of the signer. Details are presented in algorithm 5.

Note that in this specific instantiation of conditional blind signatures, the verifier can issue valid signatures by choosing a random $bsig_2$ and calculating the corresponding $bsig_1$ by the verification equation. This capability is very useful in cases where the signer wants to send the secret bit to his future self (who is posing as a verifier), something that would not be otherwise possible due to the blinding of the messages. Despite the fact, we assume that the signer and the verifier are the same entity, we intentionally distinguish them to comply with the broader definition of conditional blind signatures.

Sign

Signer		**Recipient**
input:(params, sk_S, vk_S, b)		input:(params, vk_S, m)
$r_1, r_2 \leftarrow_R \mathbb{Z}_q$		
$x := g_1^{r_1} g_2^{r_2}$		
	$\xrightarrow{x}$	
		Select Blinding Factors
		$u_1, u_2, d \leftarrow_R \mathbb{Z}_q$
		Blind
		$x^* := x g_1^{u_1} g_2^{u_2} v^d$
		$e^* := \mathcal{H}(m, x^*)$
		$e := e^* - d$
	$\xleftarrow{e}$	
$y_1 := r_1 + es_1$		
$y_2 := r_2 + es_2$		
if $b = 1$ **then**		
$(bsig_1, bsig_2) := (k^{y_1}, y_2)$		
else		
$bsig_1 \leftarrow_R \mathbb{G}$		
$bsig_2 := y_2$		
end		
The blind signature is		
$(x, e, bsig_1, bsig_2)$		
	$\xrightarrow{(bsig_1, bsig_2)}$	
		Unblind
		$sig_1 := bsig_1 \cdot k^{u_1}$
		$sig_2 := bsig_2 + u_2$
		output $(m, x^*, e^*, sig_1, sig_2)$

Fig. 1. Signing protocol for conditional blind signatures

Algorithm 5: Signature Verification

```
Input  : $sk_V, pk_V, vk_S, \text{params}, \mathcal{H}, m, sig = (x^*, e^*, sig_1, sig_2)$
Output: $b \in \{0,1\}$
if $m \neq m'$ then
    return 0
end
$e^{*'} := \mathcal{H}(m, x^*)$
${y_1}' := sig_1$
${y_2}' := sig_2$
if $x^{*s} = {y_1}'{g_2}^{{y_2}' \cdot s} v^{e^* \cdot s}$ and $e^{*'} = e^*$ then
    return 1
else
    return 0
end
```

In the case of $b = 1$ the signatures are valid and can be verified by the designated verifier. The correctness property follows easily from the verification equation:

$$\begin{aligned}
x^{*s} = {y_1}'{g_2}^{{y_2}' \cdot s} v^{e^* \cdot s} &\Leftrightarrow (x g_1^{u_1} g_2^{u_2} v^d)^s = k^{y_1+u_1} {g_2}^{(y_2+u_2)\cdot s} v^{(e+d)\cdot s} \\
&\Leftrightarrow x^s k^{u_1} g_2^{s \cdot u_2} v^{s \cdot d} = k^{y_1} k^{u_1} {g_2}^{sy_2} g_2^{su_2} v^{se} v^{sd} \\
&\Leftrightarrow x^s = k^{y_1} {g_2}^{sy_2} v^{se} \\
&\Leftrightarrow x^s = {g_1}^{sy_1} {g_2}^{sy_2} v^{se} \\
&\Leftrightarrow x = {g_1}^{y_1} {g_2}^{y_2} v^e \\
&\Leftrightarrow g_1^{r_1} g_2^{r_2} = g_1^{r_1+es_1} g_2^{r_2+es_2} (g_1^{-s_1} {g_2}^{-s_2})^e \\
&\Leftrightarrow g_1^{r_1} g_2^{r_2} = g_1^{r_1} g_1^{es_1} g_2^{r_2} g_2^{es_2} g_1^{-es_1} {g_2}^{-es_2} \\
&\Leftrightarrow g_1^{r_1} g_2^{r_2} = g_1^{r_1} g_2^{r_2}
\end{aligned}$$

5 Security Analysis

5.1 Blindness

For the blindness property we can apply the arguments of the original Okamoto-Schnorr scheme [10]. More specifically, the commitment is blinded in exactly the same way in both schemes and the second parts of the signatures are identical in both cases. In addition, the message hash is hidden using the value d exactly as in [10]. The first part of the signature is 'lifted' in our case, but the mapping from y_1 to k^{y_1} is one to one and onto. As a result in the blindness game in algorithm 1, the probability that an unbounded adversary succeeds in linking two protocol executions to the corresponding messages and signature pairs is exactly 1/2.

5.2 Strong One More Forgery

Our system is also secure against the strong version of the one more forgery assumption [17]. We note here that an adversary can create invalid signatures by randomly choosing $y_2 \in \mathbb{Z}_q$ and a random element of $\mathbb{G}$. As a result, in the security proof, an interaction with the signer for an invalid signature does not provide any advantage, so we may assume that the adversary only interacts with the signer to obtain valid signatures.

The following theorem demonstrates that the system is secure under the (strong) one more forgery definition.

Theorem 1. *Suppose there exists a* PPT *adversary $\mathcal{A}$ that wins the* OneMore-Forge *experiment, for l polylogarithmic in the security parameter λ, with non negligible probability. Then there exists a* PPT *algorithm $\mathcal{B}$ that solves the* Computational Diffie Hellman *problem with non negligible probability.*

Proof. Let $\mathcal{A}$ be such an adversary. If $\mathcal{A}$ produces two signatures (m, x, e, k^{y_1}, y_2), $(m, x, \bar{e}, k^{\bar{y_1}}, \bar{y_2})$ for the same message with the same initial commitment and $y_2 - s_2 e \neq \bar{y_2} - s_2 \bar{e}$ then the CDH problem can be efficiently solved, i.e. g^{ab} can be computed from g, g^a, g^b.

In order to obtain these signatures we apply a Replay Attack as in ([13], [17]). More specifically we run the algorithm with a random oracle $\mathcal{H}_1$ and then we repeat the same process with a random oracle $\mathcal{H}_2$ such that $\mathcal{H}_2$ yields the same answers to the first $i-1$ questions. We expect that with non negligible probability, we will achieve the collision in the i-th query.

This follows because there is an one to one and onto correspondence between y_1 and k^{y_1} and so the probabilistic analysis presented in ([13],[17]) also holds for our scheme.

In more detail the reduction is as follows:

- Our input is g, g^a, g^b and we want to compute g^{ab}.
- We set $g_1 = g, g_2 = g^a, k = g^b = g_1^s = g^s$. We select s_1, s_2 and compute the public key v.
- We supply the public values g_1, g_2, k, v to $\mathcal{A}$.
- We execute the forgery game with $\mathcal{A}$ and random oracle $\mathcal{H}_1$.
- We repeat the attack substituting $\mathcal{H}_1$ with $\mathcal{H}_2$.
- If we receive the required signatures since they are valid: $x^s = k^{y_1} {g_2}^{y_2 s} v^{es}$ and $x^s = k^{\bar{y_1}} {g_2}^{\bar{y_2} s} v^{\bar{e}s}$.
- This means that:
$$k^{y_1} = x^s {g_2}^{-y_2 s} v^{-es}$$
$$k^{\bar{y_1}} = x^s {g_2}^{-\bar{y_2} s} v^{-\bar{e}s}$$
- In turn we have:
$$k^{y_1} k^{-\bar{y_1}} = {g_2}^{(-y_2+\bar{y_2})s} v^{(-e+\bar{e})s}$$
- We know:
  - $t = k^{y_1} k^{-\bar{y_1}}$
  - $y = (-y_2 + \bar{y_2})$

- $c = (-e + \bar{e})$
- the values s_1, s_2 from the secret key we generated

Now we can calculate g^{ab} from the above known values and g_1, g_2, k:

$$\begin{aligned} t = g_2^{ys} v^{cs} &\Rightarrow t = (g^a)^{ys}(g^{-s_1} g_2^{-s_2})^{cs} \\ &\Rightarrow t = g^{ays}(g^{-s_1} g^{-s_2 a})^{cs} \\ &\Rightarrow t = g^{ays} g^{-s_1 cs} g^{-s_2 acs} \\ &\Rightarrow t g^{s_1 cs} = g^{as(y - s_2 c)} \\ &\Rightarrow g^{as} = (t \cdot (g^s)^{s_1 c})^{(y - s_2 c)^{-1}} \\ &\Rightarrow g^{ab} = (t \cdot (g^b)^{s_1 c})^{(y - s_2 c)^{-1}} \end{aligned}$$

By using the same techniques as in ([13], [17]) it follows that the probability that this attack succeeds is non-negligible. □

5.3 Conditional Verifiability

Finally, we show that the system is conditionally verifiable by a reduction from the DDH problem:

Theorem 2. *Suppose there exist a* PPT *adversary $\mathcal{A}$ that wins the* CondVerExp *with non negligible probability. Then there exists a* PPT *algorithm $\mathcal{B}$ that solves* DECISIONAL DIFFIE HELLMAN *problem with non negligible probability.*

Proof. We will construct $\mathcal{B}$.

- $\mathcal{B}$ gets as input g, g^a, g^s, g^c. She tries to find whether $c = as$ or c is uniformly distributed in $\mathbb{Z}_q$
- $\mathcal{B}$ sets $g_1 = g$, $g_2 = g^a$ and $k = g_1^s$. She randomly chooses s_1, s_2 and sets $v := g_1^{-s_1} g_2^{-s_2}$. She gives g_1, g_2, k, v to $\mathcal{A}$.
- Using the secret key (s_1, s_2) $\mathcal{B}$ can answer $\mathcal{A}$'s valid signature requests.
- When $\mathcal{B}$ gets a challenge request from $\mathcal{A}$ she does the following:
  - She randomly chooses r_1, r_2 and sends $x := g_1^{r_1} g_2^{r_2}$ to $\mathcal{A}$.
  - $\mathcal{A}$ responds with e.
  - $\mathcal{B}$ chooses a random y_2 and sets

    $$k^{y_1} := (g^s)^{r_1} (g^c)^{r_2} (g^c)^{-y_2} (g^s)^{s_1 e} (g^c)^{s_2 e}$$

  - $\mathcal{B}$ sends the signature pair $(bsig_1, bsig_2) := (k^{y_1}, y_2)$
- As before $\mathcal{B}$ responds to $\mathcal{A}$'s signing requests using the secret key (s_1, s_2).
- $\mathcal{B}$ outputs 1 (the input is a DDH tuple) iff $\mathcal{A}$ outputs 1 (valid signature).

The validity of the signature is fully defined by the message $(bsig_1, bsig_2)$ sent by the signer. The signature is valid iff $k^{y_1} = x^s {g_2}^{-y_2 s} v^{-es}$. Now we have:

$$\begin{aligned} k^{y_1} = x^s g_2^{-y_2 s} v^{-es} &\Leftrightarrow (g^s)^{r_1} (g^c)^{r_2} (g^c)^{-y_2} (g^s)^{s_1 e} (g^c)^{s_2 e} = x^s g_2^{-y_2 s} v^{-es} \\ &\Leftrightarrow g^{sr_1} g^{cr_2} g^{-cy_2} g^{ss_1 e} g^{cs_2 e} = g^{sr_1} g_2^{sr_2} g_2^{-sy_2} g^{ss_1 e} g_2^{ss_2 e} \\ &\Leftrightarrow (g^c)^{(r_2 - y_2 + s_2 e)} = (g^{as})^{(r_2 - y_2 + s_2 e)} \end{aligned}$$

This means that if $r_2 - y_2 + s_2 e \neq 0$ then the signature is valid iff $g^c = g^{as}$ which means that the input is a DDH tuple. Since y_2 is chosen randomly, $r_2 - y_2 + s_2 e = 0$ holds only with negligible probability which yields the result. □

The theorems above prove that the system is secure according to the definitions. We must note however that security for one more forgery depends on the fact that the number of valid signatures is poly logarithmic to the security parameter, which is not strong enough. We leave it as future work to strengthen our scheme to attacks that require a polynomial number of signatures.

6 Variations

In this section we shall provide variations on the instantiation of conditional blind signatures to make them more usable in practice.

6.1 Reduced Round Conditional Blind Signatures

The signing protocol in Figure 1 requires three rounds of communication to produce a signature. At first the signer commits to a random group element x. Subsequently the user provides the blinded message, which depends on x. Finally the signer supplies the signature. It would be more practical for the user to be able to request signatures directly, without having to wait for the commitment.

A simple method to produce a signature in two rounds, can be obtained by determining a way for both the signer and the user to independently compute x. For instance, x could be computed from some adequately random fact, like the hash of a session id in a practical implementation of our scheme.

Moreover, we note that the verification key can be used to issue signatures through the verification equation, so for simplicity the signer and the verifier can be the same entity.

We present the reduced round Sign protocol in Figure 2.

The common inputs to both participants are the group parameters and the public keys $pk_v = k$ from algorithm 5, $vk_s = v$ from Figure 1 and x. Now v is a random group element, since the keys s_1, s_2 are not required to produce a signature. The signer private input is the secret key $sk_v = s$ from algorithm 5 and the secret bit b. The recipient's private input is the message m.

It is easy to see that the signature is valid since it satisfies the verification equation in algorithm 5. Moreover the view of the user $\mathcal{U}$ is the same for x random. As a result the security of the reduced round signing protocol is equivalent to the original.

Sign

Signer		**Recipient**
input:(params, s, b, k, v, x)		input:(params, m, k, v, x)
		Select Blinding Factors $u_1, u_2, d \leftarrow_R \mathbb{Z}_q$ Blind $x^* := x g_1^{u_1} g_2^{u_2} v^d$ $e^* := \mathcal{H}(m, x^*)$ $e := e^* - d$
	$\xleftarrow{\quad e \quad}$	
$y_2 \leftarrow_R \mathbb{Z}_q$ $y_1 := (x \cdot g_2^{-y_2} \cdot v^{-e})^s$ **if** $b = 1$ **then** $(bsig_1, bsig_2) := (y_1, y_2)$ **else** $(bsig_1, bsig_2) \leftarrow_R (\mathbb{G} \times \mathbb{Z}_q)$ **end** The blind signature is $(x, e, bsig_1, bsig_2)$		
	$\xrightarrow{(bsig_1, bsig_2)}$	
		Unblind $sig_1 := bsig_1 \cdot k^{u_1}$ $sig_2 := bsig_2 + u_2$ output $(m, x^*, e^*, sig_1, sig_2)$

Fig. 2. Signing protocol for reduced round conditional blind signatures

6.2 Conditional Blind Signatures with homomorphic encryption

Another useful application of our scheme, comes from the fact that it can be combined with a homomorphic encryption scheme like El Gamal [4]. Recall that in [4] two ciphertexts created with the same public key can be multiplied to obtain an encryption of the product of the corresponding plaintexts.

Conditional blind signatures preserve this multiplicative homomorphic property. In the signing phase of the protocol in Figure 1 the signer can produce the encryption of $bsig_1$ namely: $\texttt{Enc}(k^{y_1})$. During unblinding the user will multiply the encrypted part with $\texttt{Enc}(k^{u_1})$. This produces:

$$\texttt{Enc}(k^{y_1})\texttt{Enc}(k^{u_1}) = \texttt{Enc}(k^{y_1}k^{u_1}) = \texttt{Enc}(k^{y_1+u_1})$$

which means that the unblinding of the first component of the signature can be performed in encrypted form. This property will be useful in protocols that operate on messages in encrypted form.

7 Conclusion and Discussion

In this paper, we introduced a new digital signature primitive which we call *conditional blind signatures*. We defined its security properties by extending the blindness and unforgeability properties of standard blind signatures with conditional verifiability. This new property captures the fact that conditional blind signatures are verified by a designated verifier if and only if a certain condition holds. Moreover, we provided an instantiation of this new primitive by extending the Okamoto-Schnorr blind signatures and proved that it is secure by reductions from the CDH and DDH problems. Finally, we presented variations that are applicable to different usage scenarios.

Conditional blind signatures are a generic primitive and as such, they can be used in every case where the act of signing a message, must convey some additional piece of information to a designated verifier. For instance, the signer might be a registrar that checks if a user can participate in a process executed by the designated verifier. Our scheme allows for such information to be conveyed in an anonymous manner, without explicitly rejecting user requests that are not eligible for processing.

A major application of this new primitive can be found in coercion resistant electronic voting, where a voter must defeat a strong adversary that wishes to dictate the vote and threatens with countermeasures if the voter does not comply. Conditional blind signatures can be used to design an efficient protocol that utilizes the JCJ coercion resistance framework [19], where the voter can cast many ballots authenticated using indistinguishable anonymous credentials. Before the election one such credential is registered as authentic with the registration authority. A vote should be counted only if it is accompanied by this specific credential. As a result, when the voter is under coercion she can provide a different one, in effect cancelling this ballot. When she gets her moment of privacy, as required by the JCJ framework, she can cast a vote with the registered credential. Of course the coercer should not be able to distinguish the two cases. This does not apply to the tallier, who must be able to tell which votes should be counted and which should not. A protocol that utilizes our primitive to implement the above scenario involves a registration authority that compares the credentials supplied during voting with the one that is registered before. The secret bit of the signer is the result of this comparison and indicates whether the credential is valid or not. The comparison can be easily carried out by the registrar, since he has access to the voter identity. By applying our primitive he can convey this bit of information to the tallier, informing her about the validity of the vote, without leaking it to the coercer. Thus, the tallier will learn if the votes are under coercion or not and proceed to count them in the former case. Similar cases can arise in other contexts as well, e.g. in anonymous surveys used in assessments for courses, services etc. where there can be conflicts of interest that can lead to coercion.

The design of protocols for applying conditional blind signatures to electronic voting and anonymous surveys is a natural direction for further research. Moreover, it would be interesting to investigate alternative instantiations of the primitive. Another important research goal would be to extend the secret information to more than a single bit. Finally, it would be ideal to design instantiations of conditional blind signatures that can be proved unforgeable against stronger adversaries.

References

1. Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.
2. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
3. David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982.*, pages 199–203, 1982.
4. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Information Theory*, 31(4):469–472, 1985.
5. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 186–194, 1986.
6. David Chaum and Hans van Antwerpen. Undeniable signatures. In *Proceedings on Advances in Cryptology*, CRYPTO '89, pages 212–216, New York, NY, USA, 1989. Springer-Verlag New York, Inc.
7. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 239–252, 1989.
8. David Chaum and Eugène van Heyst. Group signatures. In *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, pages 257–265, 1991.

9. Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992, Proceedings*, pages 244–251, 1992.
10. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, pages 31–53, 1992.
11. David Chaum. Designated confirmer signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 86–91. Springer, 1994.
12. Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, pages 143–154, 1996.
13. David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In *Advances in Cryptology - ASIACRYPT '96, International Conference on the Theory and Applications of Cryptology and Information Security, Kyongju, Korea, November 3-7, 1996, Proceedings*, pages 252–265, 1996.
14. Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 150–164, 1997.
15. Dan Boneh. The decision diffie-hellman problem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 48–63, 1998.
16. Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *Journal of Computer and System Sciences*, 60(3):592 – 629, 2000.
17. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
18. Bill Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *In Birgit Pfitzmann, editor, Advances in Cryptology EUROCRYPT 2001, volume 2045 of Lecture Notes in Computer Science*, pages 119–135. Springer-Verlag, 2001.
19. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES 2005, Alexandria, VA, USA, November 7, 2005*, pages 61–70, 2005.
20. Dominique Schröder and Dominique Unruh. Security of blind signatures revisited. *IACR Cryptology ePrint Archive*, 2011:316, 2011.
21. Fubiao Xia. *Designated confirmer signatures : modelling, design and analysis*. PhD thesis, University of Birmingham, UK, 2013.
22. Sven Laur and Bingsheng Zhang. *Lightweight Zero-Knowledge Proofs for Crypto-Computing Protocols*, pages 140–157. Springer International Publishing, 2014.