FORMALIZING GROUP BLIND SIGNATURES AND PRACTICAL CONSTRUCTIONS WITHOUT RANDOM ORACLES

Essam Ghadafi

ghadafi@cs.bris.ac.uk
University of Bristol

ACISP 2013
1 Background

2 Security Model

3 Building Blocks

4 Our Constructions

5 Full Anonymity

6 Summary & Open Problems
Outline

1 Background

2 Security Model

3 Building Blocks

4 Our Constructions

5 Full Anonymity

6 Summary & Open Problems

Formalizing Group Blind Signatures...
1  **Background**

2  **Security Model**

3  **Building Blocks**

4  **Our Constructions**

5  **Full Anonymity**

6  **Summary & Open Problems**
1. Background

2. Security Model

3. Building Blocks

4. Our Constructions

5. Full Anonymity

6. Summary & Open Problems
1 Background

2 Security Model

3 Building Blocks

4 Our Constructions

5 Full Anonymity

6 Summary & Open Problems
1 Background

2 Security Model

3 Building Blocks

4 Our Constructions

5 Full Anonymity

6 Summary & Open Problems
**Group Blind Signatures**

- **Group Signatures [CH91]** preserve the anonymity of the signer.
- **Blind Signatures [Cha83]** preserve the privacy of the message to be signed.
- **Group Blind Signatures [LZ98]** combine properties of the above and thus preserve both the anonymity of the signer + the privacy of the message.
GROUP BLIND SIGNATURES

Opener

Issuer

User

Group

ok

ik

gpk
GROUP BLIND SIGNATURES

FORMALIZING GROUP BLIND SIGNATURES . . .
GROUP BLIND SIGNATURES

Opener

Issuer

User

Group

ok

ik

gpk

Sig

FORMALIZING GROUP BLIND SIGNATURES . . .
GROUP BLIND SIGNATURES

Opener

Issuer

User

Group

ok

ik

gpk

Sig

FORMALIZING GROUP BLIND SIGNATURES...
The primitive which combines the properties of a blind signature [Cha83] and a group signature [CH91] was introduced by Lysyanskaya and Zulfikar [LZ98].

Existing constructions:

  - Based on Camenisch-Stadler group signatures [CL97].
  - Uses divertible zero-knowledge proofs [OO90].
Group blind signatures provide bi-directional privacy and are thus useful for applications where such a requirement is needed.

**Example applications:**

- **Distributed e-cash, e.g. [LZ98]:** The e-coin reveals neither the identity of its holder nor that of the issuing bank/branch.
- **Other applications include:** multi-authority e-voting and e-auction systems.
Our Contribution

- A formal security model for the primitive.
- A generic construction.
- The first instantiations without random oracles.
- Other useful building blocks and observations.
The signing protocol is blind, i.e. the message is not known to the signer and the signature is not well defined, so:

1. How to define Full Anonymity, i.e. CCA2 Anonymity?
   - How to identify the challenge signature?
2. How to define non-frameability?
3. How to extend blindness to the group setting?
SYNTAX OF GROUP BLIND SIGNATURES

A GROUP BLIND SIGNATURE

- $\text{GKg}(1^\lambda)$: Outputs gpk, ik and ok.
- $\text{UKg}$: Outputs a pair of personal secret/public keys $(\text{ssk}[i], \text{spk}[i])$ for a signer.
- $\langle \text{Join}(\text{gpk}, i, \text{ssk}[i]), \text{Issue}(\text{ik}, i, \text{spk}[i]) \rangle$: If successful, Signer$_i$ becomes a member and obtains a group signing key gsk$[i]$.
- $\langle \text{Obtain}(\text{gpk}, m), \text{Sign}(\text{gsk}[i]) \rangle$: If successful, the user obtains a signature $\Sigma$; Otherwise, it outputs ⊥.
- $\text{GVf}(\text{gpk}, m, \Sigma)$: Verifies if $\Sigma$ is valid on the message $m$.
- $\text{Open}(\text{gpk}, \text{ok}, \text{reg}, m, \Sigma)$: Returns the identity of the signer plus a proof $\tau$.
- $\text{Judge}(\text{gpk}, i, \text{spk}[i], m, \Sigma, \tau)$: Verifies the Opener’s decision.
Correctness: If all parties are honest, we have that:

- Signatures are accepted by the GVf algorithm.
- The Opener can identify the signer.
- The Judge algorithm accepts the Opener’s decision.
Anonymity: Signatures do not reveal who signed them.

Adversary wins if: $b = b^*$. 

† Similarly to IND-RCCA [CKN03], the Open oracle returns $\bot$ if the signature opens to $i_0$ or $i_1$. 

Anonymity: Signatures do not reveal who signed them.

Adversary wins if: $b = b^*$. 

† Similarly to IND-RCCA [CKN03], the Open oracle returns $\bot$ if the signature opens to $i_0$ or $i_1$. 

Formalizing Group Blind Signatures...
**Traceability:** The adversary cannot output an untraceable signature.

Adversary wins if all the following holds:
- $\Sigma$ verifies on $m$.
- Either $\Sigma$ does not open to a signer in the group or Judge does not accept the Opener’s decision on $\Sigma$. 
Non-Frameability: The adversary cannot output a signature that traces to an honest member who did not produce it.

Adversary wins if all the following holds:

- \( \forall i \in \{1, \ldots, n+1\}, \Sigma_i \) verifies on \( m_i \), opens to id and the opening is accepted by Judge.
- The adversary asked for only \( n \) signatures by signer id.
- If weak unforgeability, the messages are distinct.
**Blindness:** Group members do not learn the message being signed.

Adversary wins if: $b = b^*$. 

- † If strong unforgeability, the $\text{Open}$ oracle returns ⊥ if $(m, \Sigma) = (m_b, \Sigma_b)$ or $(m, \Sigma) = (m_{1-b}, \Sigma_{1-b})$.
- If weak unforgeability, the $\text{Open}$ oracle returns ⊥ if $m \in \{m_0, m_1\}$. 
CONSTRUCTION CHALLENGES

How to realize the subtle dual privacy requirement and

1. Maintain round optimality
2. Avoid idealized assumptions

?
(Prime-Order) Bilinear Groups

$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are finite cyclic groups of prime order $p$, where $\mathbb{G}_1 := \langle G_1 \rangle$ and $\mathbb{G}_2 := \langle G_2 \rangle$.

**Pairing** ($e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$):

The function $e$ must have the following properties:

- **Bilinearity**: $\forall H_1 \in \mathbb{G}_1, H_2 \in \mathbb{G}_2, x, y \in \mathbb{Z}$, we have

  $$e(H_1^x, H_2^y) = e(H_1, H_2)^{xy}.$$  

- **Non-degeneracy**: The value $e(G_1, G_2) \neq 1$ generates $\mathbb{G}_T$.

- The function $e$ is efficiently computable.

**Type-3 [GPS08]**: $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. 

Groth-Sahai proofs [GS08]:

\[
\begin{align*}
G_1 & \times G_2 \xrightarrow{f} G_T \\
\iota_1 & \downarrow \uparrow \rho_1 \\
\iota_2 & \downarrow \uparrow \rho_2 \\
\iota_T & \downarrow \uparrow \rho_T \\
H_1 & := G_1^2 \times H_2 := G_2^2 \xrightarrow{F} H_T := G_T^4
\end{align*}
\]

The system work by first committing to (encrypting) the witness and then producing a proof for the statement.

The system can be instantiated in either:

- **The simulation setting** ⇒ perfectly hiding proofs.
- **The extraction setting** ⇒ perfectly sound proofs.

The limitations:

1. Can only extract one-way function (i.e. $G_i^w$) of an exponent witness $w$.
2. Cannot simulate and extract at the same time.
Useful Properties of Groth-Sahai Proofs:

- Independence of public terms (Also, independently observed by [Fuc11]):
  - Example:
    
    \[
    E := \prod_{j=1}^{n} e(A_j, Y_j) \prod_{i=1}^{m} e(X_i, B_i) \prod_{i=1}^{m} \prod_{j=1}^{n} e(X_i, Y_j)^{\gamma_{i,j}} = t_T, 
    \]

    a proof \( \Pi \) for \( E \) is independent of \( t_T \) \( \Rightarrow \) we can transform \( \Pi \) into a NIZK/NIWI proof for a related equation without knowledge of the original witness.

- Re-randomizability of proofs [BCCKLS09]:
  - Re-randomize the GS commitments and update the proofs \( \Rightarrow \) the new proof is unlinkable to the old one.
NCL is based on the CL signature scheme [CL04]:

THE NCL SIGNATURE SCHEME

- **KeyGen:** Choose \( x, y \leftarrow \mathbb{Z}_p \), set \( \text{sk} := (x, y) \) and \( \text{pk} := (X := G_2^x, Y := G_2^y) \).

- **Sign:** To sign \( (M_1, M_2) \in \mathbb{G}_1 \times \mathbb{G}_2 \), return \( \perp \) if \( e(M_1, G_2) \neq e(G_1, M_2) \); otherwise, compute \( \sigma := (A := G_1^a, B := A^y, C := M_1^{ay}, D := (A \cdot C)^x) \in \mathbb{G}_1^4 \).

- **Verify:** Check that \( A \neq 1_{\mathbb{G}_1} \) and

\[
\begin{align*}
e(B, G_2) &= e(A, Y) \\
e(C, G_2) &= e(B, M_2) \\
e(D, G_2) &= e(A \cdot C, X)
\end{align*}
\]
NCL is secure under the (interactive) DH-LRSW assumption:

**Definition (Dual-Hidden LRSW (DH-LRSW) Assumption)**

Given \((G_2^x, G_2^y)\) for \(x, y \leftarrow \mathbb{Z}_p\) and an oracle that on input a pair \((M_1, M_2) \in G_1 \times G_2\) outputs:

- \(\bot\) if \(e(M_1, G_2) \neq e(G_1, M_2)\).
- A DH-LRSW tuple \((G_1^a, G_1^{ay}, M_1^{ay}, A^x \cdot M_1^{axy})\) for \(a \leftarrow \mathbb{Z}_p\) otherwise.

, it is infeasible to compute a DH-LRSW tuple for \((M_1', M_2')\) that was never queried to the oracle.
The Blind Signature Scheme [Fuc09]

As an example, we use the blind signature scheme by [Fuc09] based on the following automorphic signature scheme:

The Automorphic Signature Scheme [Fuc09]

- **Setup**: Given \((e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G_1, G_2, p)\), choose \(F, K, T \leftarrow \mathbb{G}_1\).
- **KeyGen**: Choose \(s \leftarrow \mathbb{Z}_p\) and set \(pk := (S_1, S_2) = (G_1^s, G_2^s)\).
- **Sign**: To sign \((M_1, M_2) \in \mathbb{G}_1 \times \mathbb{G}_2\) where \(e(M_1, G_2) = e(G_1, M_2)\), choose \(r, c \leftarrow \mathbb{Z}_p\) and compute:
  
  \[ H := (K \cdot T^r \cdot M_1)^{\frac{1}{x+c}}, \]
  
  \[ R_1 := G_1^r, R_2 := G_2^r, C_1 := F^c, C_2 := G_2^c. \]

  The signature is \(\sigma := (H, R_1, C_1, R_2, C_2) \in \mathbb{G}_1^3 \times \mathbb{G}_2^2\).
- **Verify**: Check that
  
  \[ e(H, S_2 \cdot C_2) = e(K \cdot M_1, G_2)e(T, R_2) \]
  \[ e(C_1, G_2) = e(F, C_2) \]
  \[ e(R_1, G_2) = e(G_1, R_2) \]
The automorphic signature scheme is secure under:

**Definition (AWFCDH Assumption)**

Given \((G_1, G_1^a, G_2) \in \mathbb{G}_1^\times \times \mathbb{G}_2^\times\) for \(a \leftarrow \mathbb{Z}_p\), it is infeasible to output a tuple \((G_1^b, G_1^{ab}, G_2^b, G_2^{ab}) \in \mathbb{G}_1^\times \times \mathbb{G}_2^\times\) for an arbitrary \(b \in \mathbb{Z}_p\).

**Definition (q-ADHSDH Assumption)**

Given \((G_1, F, K, G_1^x, G_2, G_2^x) \in \mathbb{G}_1^\times^4 \times \mathbb{G}_2^\times^2\) for \(x \leftarrow \mathbb{Z}_p\), and \(q - 1\) tuples \((A_i := (K \cdot G_1^{r_i})^{x+c_i}, C_{1,i} := F^{c_i}, C_{2,i} := G_2^{c_i}, R_{1,i} := G_1^{r_i}, R_{2,i} := G_2^{r_i})\) for \(i = 1, \ldots, q - 1\), where \(c_i, r_i \leftarrow \mathbb{Z}_p\), it is infeasible to output a new tuple \((A^*, C_{1}^*, C_{2}^*, R_{1}^*, R_{2}^*)\).
To get a round-optimal blind signature, Fischlin’s framework [Fis06]:

1. The user sends a commitment $C$ to the message $M$ to the signer.
2. The signer responds with a signature $\sigma$ on the commitment $C$.
3. The final blind signature $\Sigma$ is a NIZK PoK $\Pi$ of $C$ and $\sigma$ s.t.
   1. $\sigma$ is a valid signature on $C$ w.r.t. $pk$.
   2. $C$ is a commitment to $M$. 
In Fischlin’s framework If:

1. The PoK used is re-randomizable and independent of the public terms, e.g. Groth-Sahai proofs.
2. The signature scheme has the property that all the terms involving the message in the verification equations are independent of the signature components relying on the signing key, e.g. [Fuc09].

, we obtain a signer-anonymous blind signature.
Modify the Fischlin’s framework as follows:

1. The user sends a commitment $C$ to the message $M$.
2. The signer signs $C$ and responds with a PoK $\Pi'$ of $\sigma$ and $pk$ s.t.
   1. $\sigma$ is a valid signature on $C$ w.r.t. $pk$.
3. The user proceeds as follows:
   1. Re-randomizes $\Pi'$ into a fresh proof $\Pi$.
   2. Adds to $\Pi$ a proof that $C$ is a commitment to $M$.

The final signer-anonymous blind signature $\Sigma$ is $\Pi$. 
SIGNER-ANONYMOUS BLIND SIGNATURES

Security:

- **Unforgeability**: As in Fischlin’s framework.
- **Blindness**: As in Fischlin’s framework + Re-randomizability of the PoK.
- **Anonymity**: The hiding (i.e. NIWI/NIZK) properties of the PoK.
To extend the signer-anonymous BS to a GBS, we need:

1. A signature scheme $\text{CERT}$ to certify members’ public keys when they join.
   - $\Rightarrow$ Give $\text{sk}_{\text{CERT}}$ to the Issuer as $\text{ik}$.

2. Give the PoK extraction key to the Opener as $\text{ok}$.

3. The signer additionally needs to prove that he has a certificate on his public key w.r.t. $\text{pk}_{\text{CERT}}$, i.e. that he is a member of the group.
Instantiation I

- **The Join Protocol**: Uses the NCL signature scheme.
- **The Signing Protocol**: Based on the BS by [Fuc09] (i.e. The automorphic signature scheme + GS proofs).

- **Assumptions**: DH-LRSW, SXDH, AWFCDH and q-ADHSDH.

- **The Pros**: More efficient (signature size is $\mathbb{G}_1^{38} + \mathbb{G}_2^{36}$).
- **The Cons**: Involves some interactive intractability assumptions (i.e. the DH-LRSW assumption).
Instantiation II

- **The Join Protocol:** Uses the automorphic signature scheme [Fuc09].
- **The Signing Protocol:** Based on the BS by [Fuc09] (i.e. the automorphic signature scheme [Fuc09] + GS proofs).
- **Assumptions:** SXDH, AWFCDH and $q$-ADHSDH.

- The Pros 🙂: Only relies on falsifiable intractability assumptions.
- The Cons 😞: Less efficient (signature size is $\mathbb{G}_1^{42} + \mathbb{G}_2^{38}$).
Groth-Sahai proofs are not simulation-sound $\Rightarrow$ when simulating, we can no longer answer Open queries.

**Q**: How to achieve full anonymity?

**A**: One way is to combine Groth-Sahai proofs with an IND-CCA2 encryption scheme.

$\Rightarrow$ The signer additionally encrypts the witness and proves that it was done correctly. When simulating, decrypt the ciphertext to recover the witness.

The encryption scheme needs to be re-randomizable, e.g. maybe we could use an IND-RCCA encryption scheme.
Groth-Sahai proofs are not simulation-sound \(\Rightarrow\) when simulating, we can no longer answer Open queries.

**Q:** How to achieve full anonymity?

**A:** One way is to combine Groth-Sahai proofs with an IND-CCA2 encryption scheme.

\[\Rightarrow\] The signer additionally encrypts the witness and proves that it was done correctly. When simulating, decrypt the ciphertext to recover the witness.

The encryption scheme needs to be re-randomizable, e.g. maybe we could use an IND-RCCA encryption scheme.
A formal security model for the primitive.
Round-optimal constructions.
The first constructions without idealized assumptions.
Open Problems

- Efficient fully-anonymous instantiations.
- More efficient constructions without idealized assumptions.
Thank you for your attention!
Questions?