Generic Homomorphic Undeniable Signatures

J. Monnerat S. Vaudenay

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Asiacrypt '04 - December 8, 2004

Outline

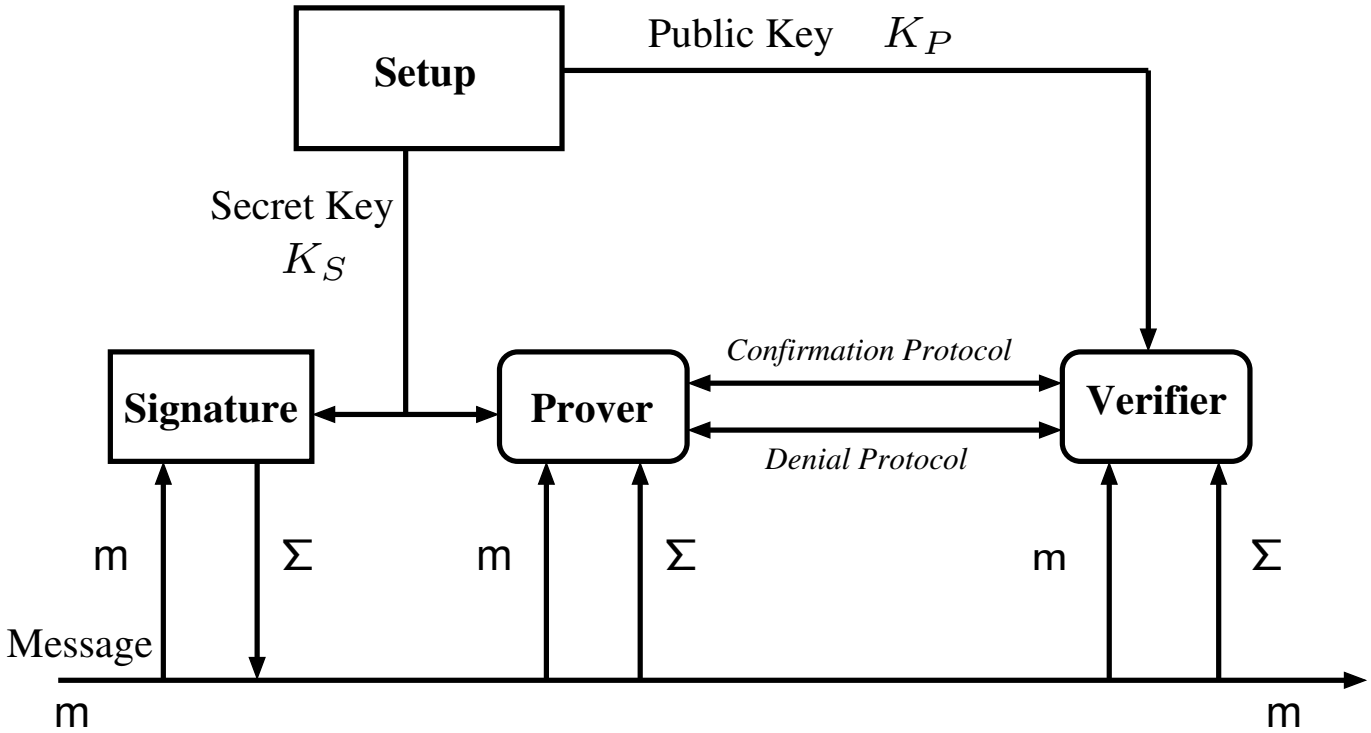
Introduction

Undeniable Signature (1)

Properties:

- Public key algorithm
- Binding some information or a document with an entity
- Verifiable only with the cooperation of the signer
- Non repudiation property still holds!

Undeniable Signature (2)



Related Work

- *Undeniable Signatures*, Chaum and van Antwerpen, Crypto'89.
- *Zero-knowledge Undeniable Signatures*, Chaum, Eurocrypt '90.
- *New Convertible Undeniable Signatures*, Dåmgard and Pedersen, Eurocrypt '96.
- *RSA-Based Undeniable Signatures*, Gennaro, Rabin and Krawczyk, Crypto '97.
- *Identity Based Undeniable Signatures*, Libert and Quisquater, CT-RSA '04.
- *Undeniable Signatures Based on Characters*, Monnerat and Vaudenay, PKC '04. (MOVA Scheme)

Interpolation of Group Homomorphisms

Interpolation Problems

GHI Problem (Group Homomorphism Interpolation Problem)

Parameters: two Abelian groups G and H, a set of s points $S \subseteq G \times H$.

Input: $x \in G$.

Problem: find $y \in H$ such that $S \cup \{(x, y)\}$ interpolates in a group homomorphism i.e., for $S = \{(x_1, y_1), \ldots, (x_s, y_s)\}$ there exists a group homomorphism Hom such that $\mathrm{Hom}(x_i) = y_i$, $i = 1, \ldots, s$ and $\mathrm{Hom}(x) = y$.

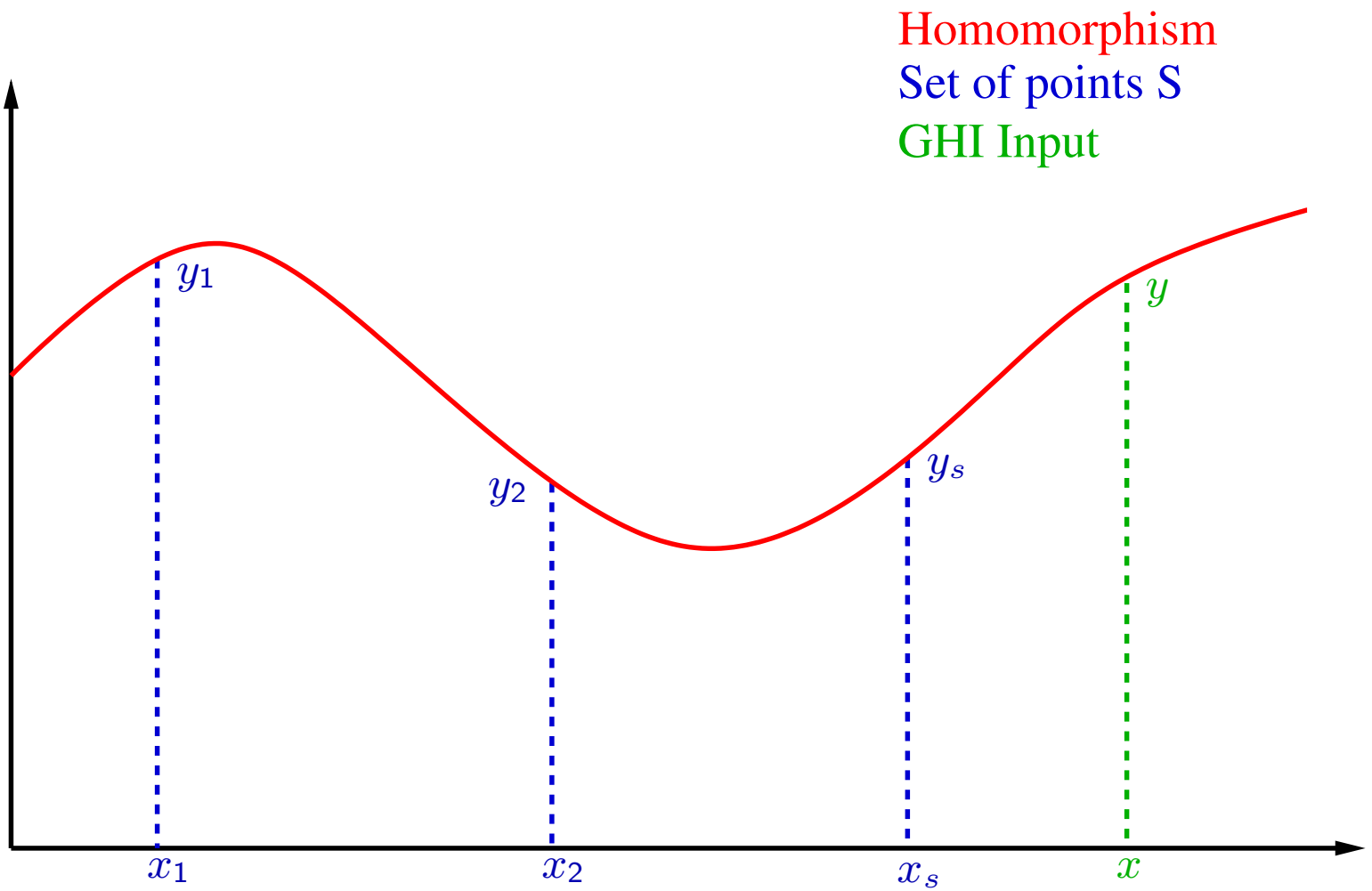
GHID Problem (Group Homomorphism Interpolation Decisional Problem)

Parameters: two Abelian groups G and H, a set of s points $S \subseteq G \times H$.

Input: $(x, y) \in G \times H$.

Problem: does $S \cup \{(x, y)\}$ interpolate in a group homomorphism?

Geometrical Interpretation



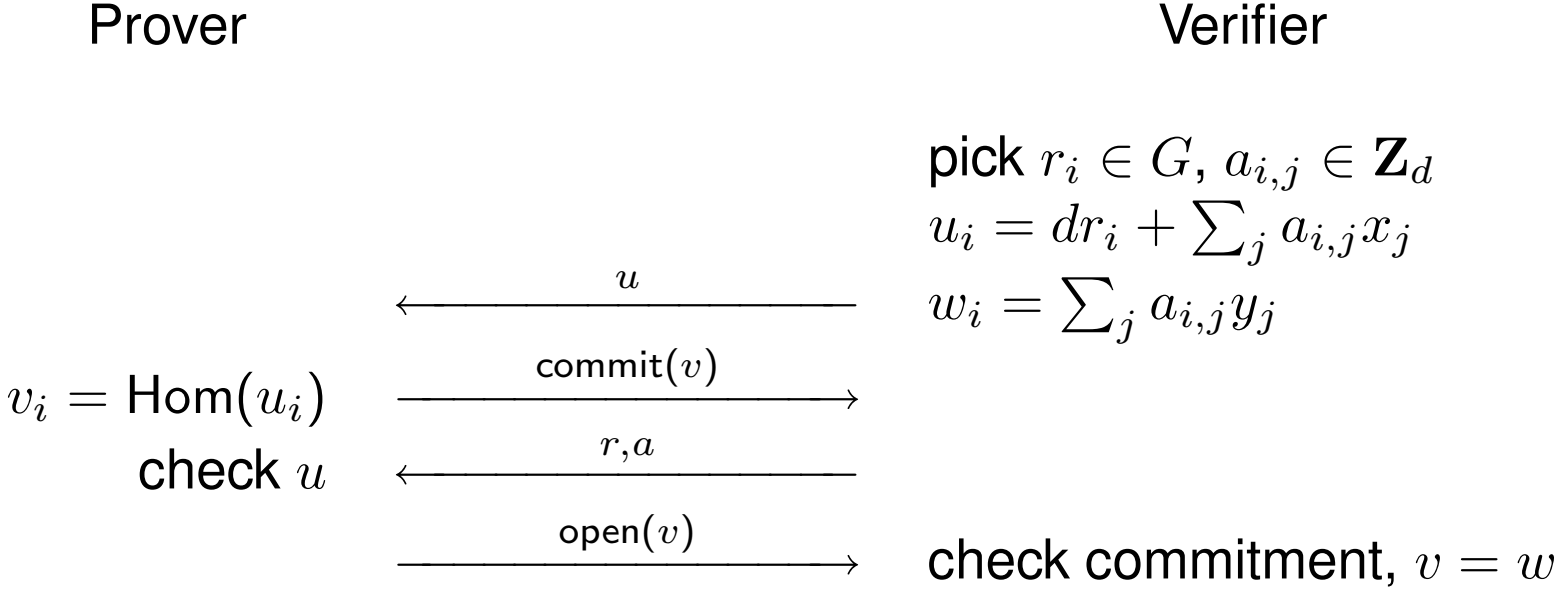
Relation to Well-known Problems

- **DLP.** $G :=< g >$ cyclic group of order q, $H := \mathbf{Z}_q$. $S = \{(g, 1)\}$ interpolates in a unique homomorphism, namely the discrete logarithm w.r.t. g.
- **RSA.** Let $n = pq$ be a RSA modulus, $e \in \mathbf{Z}^*_{\varphi(n)}$ the encryption exponent and $G = H = \mathbf{Z}^*_n$. Let $S := \{(x_i^e \bmod n, x_i)_{i=1,\ldots,s}\}$ such that the first coordinates generate $\mathbf{Z}^*_n$. The RSA decryption problem corresponds to the GHIP.
- Other examples such as, the quadratic residuosity problem, Diffie-Hellman problem, bilinear Diffie-Hellman problem, MOVA problem, ...

Proof of Interpolation

Let $d := \#H$.

GHIproof $(\{(x_j, y_j); j = 1, \ldots, J\})$ with parameter I



Security of GHIproof

The $\mathsf{GHIproof}_I(S)$ protocol satisfies the following properties:

- **Completeness.** The protocol always succeeds when the prover and the verifier follow the protocol.
- **Zero-knowledge** The protocol is perfectly black-box zero-knowledge.
- **Proof of membership.** If the protocol succeeds, then S interpolates in a group homomorphism.
- **Proof of knowledge.** If the protocol succeeds, there exists an extractor which computes an interpolating homomorphism.

Proof of Non-Interpolation

Let p be the smallest prime factor of $d = \#H$.

coGHIproof $\left(\{(x_j, y_j); j = 1, \ldots, J\}, \{(x'_k, z_k); k = 1, \ldots, K\}\right)$ with parameter I

Prover		Verifier
		pick $r_{i,k} \in G$, $a_{i,j,k} \in \mathbf{Z}_d$, $\lambda_i \in \mathbf{Z}_p$ $u_{i,k} = dr_{i,k} + \sum_j a_{i,j,k} x_j + \lambda_i x'_k$ $w_{i,k} = \sum_j a_{i,j,k} y_j + \lambda_i z_k$
compute $v_{i,k} = \mathsf{Hom}(u_{i,k})$ deduce λ_i from	$\xleftarrow{u,w}$	
$w_{i,k} - v_{i,k} = \lambda_i(z_k - \mathsf{Hom}(x'_k))$	$\xrightarrow{\mathsf{commit}(\lambda)}$	
check u, w	$\xleftarrow{r,a}$	
	$\xrightarrow{\mathsf{open}(\lambda)}$	check commitment, λ

Uniqueness of the Homomorphism

Theorem

Let G, H be two finite Abelian groups. We denote d the order of H. Let $x_1, \ldots, x_s \in G$ which span G'. The following properties are equivalent. In this case, we say that $x_1, \ldots, x_s$ H-generate G.

1. *For all $y_1, \ldots, y_s \in H$, there exists at most one group homomorphism* $\mathsf{Hom} : G \longrightarrow H$ *such that* $\mathsf{Hom}(x_i) = y_i$ *for all* $i = 1, \ldots s$.
2. $G' + dG = G$.

Our Signature Scheme

Using Group Homomorphisms in Cryptography

DL-based cryptography $y = g^x$

secret input $\xrightarrow{\text{fixed homomorphism}}$ public key

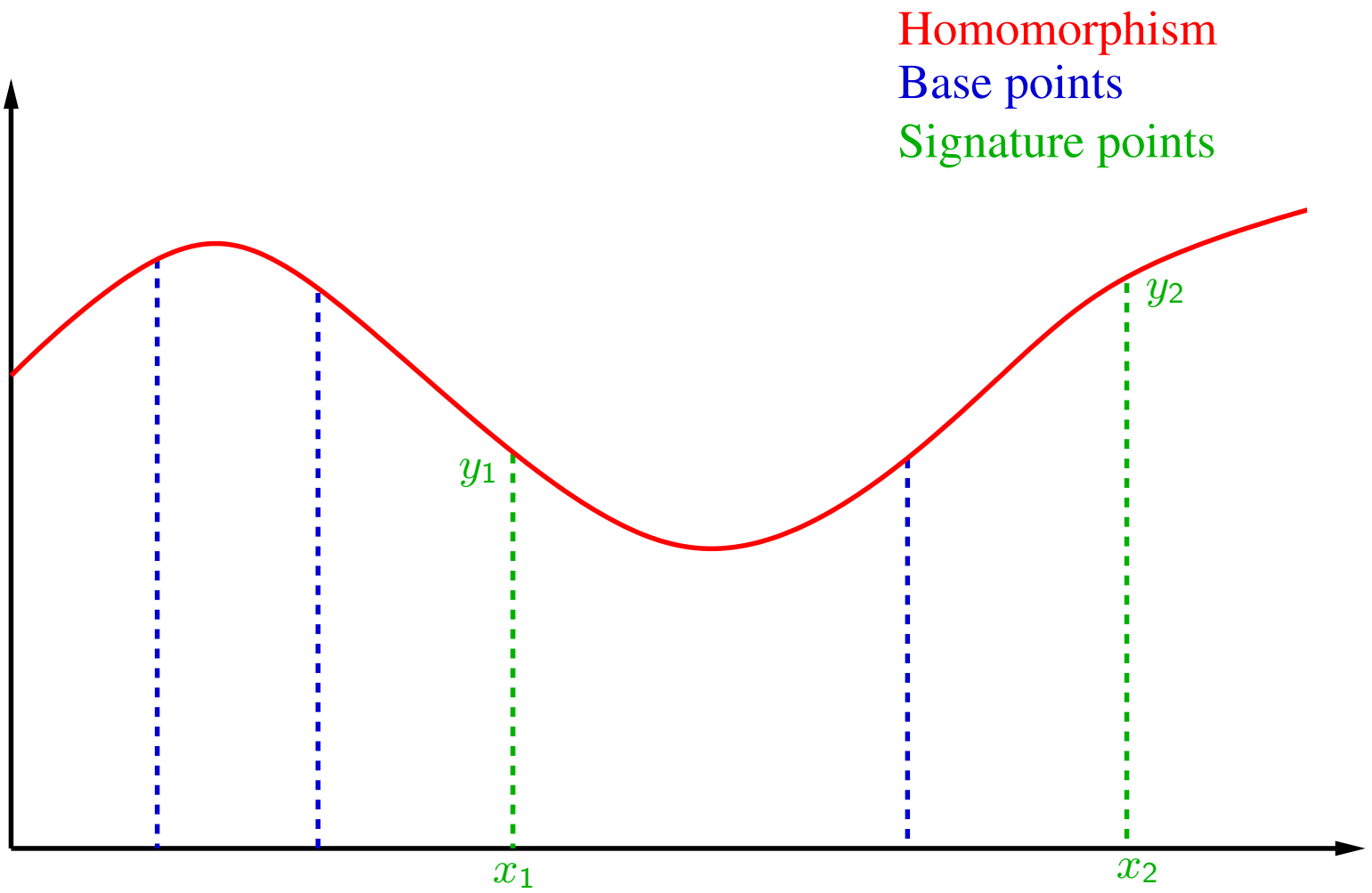
Our approach $y = \mathrm{Hom}(x)$

fixed input $\xrightarrow{\text{secret homomorphism}}$ public key

Basic Description

- Setup
  - Select two groups Xgroup and Ygroup (Ygroup small)
  - Select a secret group homomorphism Hom : Xgroup $\longrightarrow$ Ygroup
  - Select some base points to characterize Hom
- Signature
  - Generate some x_i's from the message
  - Compute the group homomorphism on the x_i's
- Verification: prove/disprove the interpolation

Geometrical Interpretation



Setups without Validation

- Setup Variant 1. The signer selects Abelian groups Xgroup, Ygroup and an homomorphism Hom. He computes the order d of Ygroup. He then picks a random string SeedK and computes the Lkey first values Xkey_j from $\mathsf{Gen}_1(\mathsf{SeedK})$ and $\mathsf{Ykey}_j = \mathsf{Hom}(\mathsf{Xkey}_j)$, $j = 1, \ldots, \mathsf{Lkey}$.
- Setup Variant 2. (signer with a Registration Authority) The role of RA consists of making sure that a key was randomly selected. This works similarly as the variant 1 except that RA picks SeedK at random after the signer have sent his identity Id. The RA sends SeedK with a signature C for

$$(\mathsf{Id}, \mathsf{Xgroup}, \mathsf{Ygroup}, d, \mathsf{SeedK}).$$

Signature Generation

Let M be a message to be signed.

- Compute $\mathsf{Gen}_2(M) \rightarrow (\mathsf{Xsig}_1, \ldots, \mathsf{Xsig}_{\mathsf{Lsig}})$
- Compute $\mathsf{Ysig}_1 = \mathsf{Hom}(\mathsf{Xsig}_1), \ldots, \mathsf{Ysig}_{\mathsf{Lsig}} = \mathsf{Hom}(\mathsf{Xsig}_{\mathsf{Lsig}})$
- The signature is $[\mathsf{Ysig}_1, \ldots, \mathsf{Ysig}_{\mathsf{Lsig}}]$

Confirmation Protocol

Let M be the message and $[\mathsf{Ysig}_1, \ldots, \mathsf{Ysig}_{\mathsf{Lsig}}]$ be the signature

$$K_p = (\mathsf{Xgroup}, \mathsf{Ygroup}, d, \mathsf{param}, \mathsf{SeedK}, (\mathsf{Ykey}_1, \ldots, \mathsf{Ykey}_{\mathsf{Lkey}}), \mathsf{opt})$$

- Compute $\mathsf{Gen}_1(\mathsf{SeedK}) \rightarrow (\mathsf{Xkey}_1, \ldots, \mathsf{Xkey}_{\mathsf{Lkey}})$
- Compute $\mathsf{Gen}_2(M) \rightarrow (\mathsf{Xsig}_1, \ldots, \mathsf{Xsig}_{\mathsf{Lsig}})$
- Set

$$S = \{(\mathsf{Xkey}_j, \mathsf{Ykey}_j); j = 1, \ldots, \mathsf{Lkey}\} \cup \{(\mathsf{Xsig}_k, \mathsf{Ysig}_k); k = 1, \ldots, \mathsf{Lsig}\}$$

- Run $\mathsf{GHIproof}(S)$ with parameter Icon.

Denial Protocol

Let M be the message and $[\mathsf{Zsig}_1, \ldots, \mathsf{Zsig}_{\mathsf{Lsig}}]$ be the alleged non-signature

$$K_p = (\mathsf{Xgroup}, \mathsf{Ygroup}, d, \mathsf{param}, \mathsf{SeedK}, (\mathsf{Ykey}_1, \ldots, \mathsf{Ykey}_{\mathsf{Lkey}}), \mathsf{opt})$$

- Compute $\mathsf{Gen}_1(\mathsf{SeedK}) \rightarrow (\mathsf{Xkey}_1, \ldots, \mathsf{Xkey}_{\mathsf{Lkey}})$
- Compute $\mathsf{Gen}_2(M) \rightarrow (\mathsf{Xsig}_1, \ldots, \mathsf{Xsig}_{\mathsf{Lsig}})$
- Set

$$\begin{aligned} S &= \{(\mathsf{Xkey}_j, \mathsf{Ykey}_j); j = 1, \ldots, \mathsf{Lkey}\} \\ T &= \{(\mathsf{Xsig}_k, \mathsf{Zsig}_k); k = 1, \ldots, \mathsf{Lsig}\} \end{aligned}$$

- Run $\mathsf{coGHIproof}(S, T)$ with parameter Iden

MGGD Problem and Key Validity

MGGD Problem (Modular Group Generation Decisional Problem)

Parameters: an Abelian group G, an integer d.

Input: a set of values $S_1 = \{x_1, \ldots, x_s\} \subseteq G$.

Problem: Is $< S_1 > + dG = G$?

- We say that the public key is valid if the answer of the MGGD Problem is positive with $G = \mathsf{Xgroup}$ and $S_1 = \{\mathsf{Xkey}_1, \ldots, \mathsf{Xkey}_{\mathsf{Lkey}}\}$, i.e, S_1 Ygroup-generate Xgroup.
- Otherwise, the signer might be able to repudiate his signature.

Representation Problem

Expert group knowledge = ability to solve MSR and Root problems in Xgroup.

MSR Problem (Modular System Representation Problem)

Parameters: an Abelian group G, a set $S_1 = \{x_1, \ldots, x_s\} \subseteq G$, an integer d.

Input: $x \in G$.

Problem: find $a_1, \ldots, a_s \in \mathbf{Z}$ such that $x \in a_1 x_1 + \cdots + a_s x_s + dG$.

Root Problem (dth Root Problem)

Parameters: an Abelian group G, an integer d.

Input: $x \in G$.

Problem: find $r \in G$ such that $x = dr$.

Group Homomorphism Uniqueness Proof

MGGDproof $(\{x_j; j = 1, \ldots, J\})$ with parameter I

Prover		Verifier
pick $\alpha_i \in$ Xgroup	$\xrightarrow{\text{commit}(\alpha)}$	
	$\xleftarrow{\beta}$	pick $\beta_i \in$ Xgroup
solve $\alpha_i + \beta_i = dr_i + \sum_j a_{i,j} x_j$		
	$\xrightarrow{\text{open}(\alpha), r, a}$	check commitment, r, a

$\rightarrow$ all Xgroup elements can be written $dr_i + \sum_j a_{i,j} x_j$...

Setups with Validation

Setup Variant 3 (signer with an expert group knowledge)
Like the Setup Variant 1, but the signer also runs $\mathsf{MGGDproof}(\{\mathsf{Xkey}_1, \ldots, \mathsf{Xkey}_{\mathsf{Lkey}}\})$ with parameter Ival to validate the key.

Setup Variant 4 (signer with an expert group knowledge, non-interactive)
Like Setup Variant 3 except that MGGDproof is transformed into a non-interactive proof.

Public Key Content

$$K_p = (\mathsf{Xgroup}, \mathsf{Ygroup}, d, \mathsf{param}, \mathsf{SeedK}, (\mathsf{Ykey}_1, \ldots, \mathsf{Ykey}_{\mathsf{Lkey}}), \mathsf{opt})$$

- Variant 1: $\mathsf{opt} = \emptyset$
- Variant 2: $\mathsf{opt} = \mathsf{Id}, C$
- Variant 3: $\mathsf{opt} = \mathsf{Ival}$
- Variant 4: $\mathsf{opt} = \mathsf{Ival}, \mathsf{niMGGDproof}$

Security Results

Theorem

Assuming that the public key is valid, we have the following security results.

i. *Let* $S = \{(\mathsf{Xkey}_1, \mathsf{Ykey}_1), \ldots, (\mathsf{Xkey}_{\mathsf{Lkey}}, \mathsf{Ykey}_{\mathsf{Lkey}})\}$. *The scheme* **resists existential forgery attacks** *provided that* Gen_2 *is a random oracle and the* S*-GHI problem is intractable.*

ii. *The confirmation (resp. denial) protocol is* **sound**.

iii. *The confirmation protocol is* **private** *when the commitment scheme is extractable.*

iv. *The signatures are* **invisible**.

v. *The confirmation (resp. denial) protocol is* **perfectly black-box zero-knowledge** *when the commitment scheme is perfectly hiding.*

Setup Example

Let $n = p \times q$ such that $p = rd + 1$ and q are prime, $\gcd(r, d) = 1$, $\gcd(q - 1, d) = 1$. We take $G = \mathbf{Z}_n^*$ and $H = \mathbf{Z}_d$. We can easily compute a group homomorphism by first raising to the power $r(q - 1)$ then computing a discrete logarithm.

- Using a precomputed table (memory $\mathcal{O}(d)$, $\mathcal{O}(1)$ complexity)
- Time-memory tradeoffs (memory $\mathcal{O}(M)$, $\mathcal{O}(d/M)$ complexity)
- Using the Pollard algorithm (no memory, $\mathcal{O}(\sqrt{d})$ complexity)

Complexity

- We take $G = \mathbf{Z}_n^*$ with a standard RSA-modulus $n = pq$ and compare the setup example with MOVA adapted to our scheme ($d = 2$).
- We consider an online security of 2^{20} and offline security of 2^{80}.

Setup	d	Lkey	Lsig, Icon, Iden	Ival	Signature cost	Confirmation cost
1	2	80	20		20 Leg. symb.	20 Leg. symb., 730 mult.
2	2	20	20		20 Leg. symb.	20 Leg. symb., 280 mult.
3	2	2	20	20	20 Leg. symb.	20 Leg. symb., 145 mult.
4	2	2	20	80	20 Leg. symb.	20 Leg. symb., 145 mult.
1	$2^{20}+7$	4	1		1 Hom	1 Hom, 65 mult.
2	$2^{20}+7$	1	1		1 Hom	1 Hom, 35 mult.
3	$2^{20}+7$	1	1	1	1 Hom	1 Hom, 35 mult.
4	$2^{20}+7$	1	1	4	1 Hom	1 Hom, 35 mult.

Leg. symb. $\approx$ modular inversion

Hom $\approx$ exponentiation in $\mathbf{Z}_p^*$

Other properties

- We can have some 2-move variants for the confirmation and denial protocol.
- With expert group knowledge we can achieve selective convertibility.
- We can easily confirm a bunch of signatures and achieves batch verification.
- The non-transferability of the proofs is achieved using trapdoor commitment.

Conclusion

- We introduced the GHI and GHID problems
- We proposed efficient ZK proofs for GHID and co-GHID
- We devised a (generic) undeniable signature scheme
- Our scheme can achieve (very) short signatures and low computational costs
- Other nice properties: batch verification, selective convertibility, etc.