Lattice-Based Signature Scheme with Verifier Local Revocation

Adeline Langlois[1] San Ling[2]
Khoa Nguyen[2] Huaxiong Wang[2]

[1]LIP, ENS de Lyon, France

[2]Nanyang Technological University, Singapore

March 27, 2014

Our main result

with N members

First lattice-based group signature with verifier-local revocation, logarithmic signature size, and security under the SIS assumption in the Random Oracle Model.

logarithmic in N

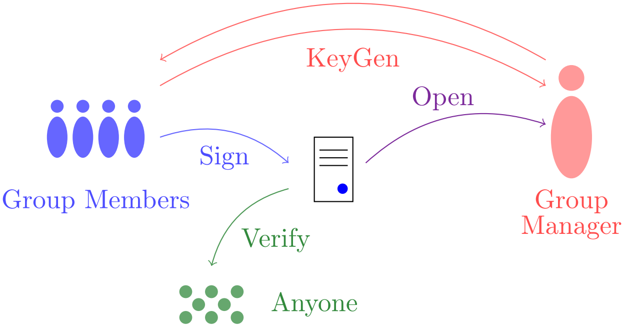
hard problem on lattices

Group signatures

[ChaumVanHeyst91]

Group signatures allow any member of a group to anonymously and accountably sign on behalf of this group.

- Group manager gpk, gsk_i — KeyGen, Open
- Group members (gsk_i) — Sign
- Anyone — Verify



Security:

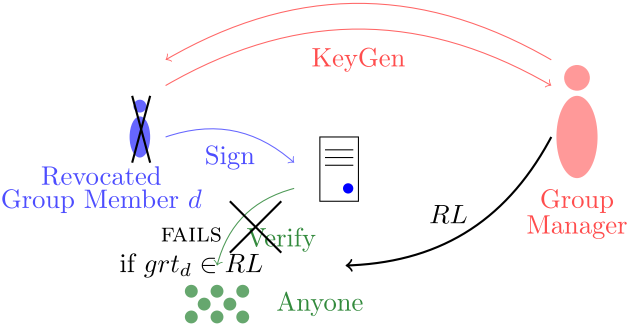
- Anonymity
- Traceability

Group signatures with verifier-local revocation

[ChaumVanHeyst91] [BonehShacham04]

Group signatures allow any member of a group to anonymously and accountably sign on behalf of this group.

- Group manager gpk, gsk_i, grt_i — KeyGen
- Group members (gsk_i) — Sign
- Anyone — Verify



Security:

- Anonymity
- Traceability

Security: anonymity and traceability

Security requirements [BonehShacham04]

- Correctness
  $\forall(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt}) \leftarrow \mathsf{KeyGen}$, $\forall i \in [N-1]$, $\forall M \in \{0,1\}^*$,

$$\mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}_i, M), M) = \mathsf{Valid} \;\Leftrightarrow\; \mathsf{grt}_i \notin RL.$$

- Selfless-anonymity

A given signature does not leak the identity of its originator.

Given	gpk and Sign, Corruption and Revocation queries,
Goal	find which of the two adaptively chosen keys generates the signature.

- Traceability

No collusion of malicious users can produce a valid signature that cannot be traced to one of them.

Given	gpk, grt_i for all i, and gsk_i of users in the collusion,
Goal	create a valid signature that doesn't trace to someone in the collusion (or that fails).

Applications

Need for authenticity *and* anonymity

- Anonymous credentials: anonymous use of certified attributes
  - E.g.: student card - name, picture, date, grade...
- Traffic management (Vehicle Safety Communications project of the U.S. Dept. of Transportation).
- Restrictive area access.

Prior works

- Group signature introduced by [ChaumVanHest91],
- Group signature with verifier local revocation introduced by [Brickell03] and [KiayiasTsiounisYung04],
- Formalized by [BonehShacham04],
- Number of realizations in bilinear map setting : [NakanishiFunabiki05 and 06], [LibertVergnaud09], [BichselCamenishNevenSmartWarinschi10].

In lattice-based cryptography:

- First one [GordonKatzVaikuntanathan10], then with signature size linear in N: [CamenischNevenRückert12].
- Signature size logarithmic in N (and full-anonymity): [LaguillaumieLangloisLibertStehlé13].
- Our result: first lattice-based group signature with verifier-local revocation (and we have signature size logarithmic in N).

Lattice-based cryptography

From basic to very advanced primitives

- Public key encryption [Regev05, ...],
- Lyubashevsky signature scheme [Lyubashevsky12],
- Identity-based encryption [GentryPeikertVaikuntanathan08, ...],
- Attribute-based encryption [Boyen13, GorbunovVaikuntanathanWee13],
- Fully homomorphic encryption [Gentry09, ...].

Advantages of lattice-based primitives

- (Asymptotically) efficient,
- Security proofs **from the hardness of LWE and SIS**,
- Likely to resist quantum attacks.

SIS$_\beta$ and ISIS$_\beta$

Parameters: n dimension, $m \geq n$, q modulus.

For $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n})$:

Short Integer Solution	**Inhomogeneous SIS**
$\mathbf{x}\,\mathbf{A} = \mathbf{0} \bmod q$	$\mathbf{x}\,\mathbf{A} = \mathbf{u} \bmod q$
Goal: Given $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n})$, find $\mathbf{x}$ s.t. $0 < \lVert \mathbf{x} \rVert \leq \beta$.	Goal: Given $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n})$, $\mathbf{u} \in \mathbb{Z}_q^n$, find $\mathbf{x}$ s.t. $0 < \lVert \mathbf{x} \rVert \leq \beta$.

Shown to be as hard as worst-case lattice problems, [GentryPeikertVaikuntanathan2008]

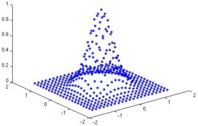
Lattice-based cryptography toolbox: trapdoors

- TrapGen $\rightsquigarrow$ $(\mathbf{A}, \mathbf{T_A})$ such that $\mathbf{T_A}$ is a short basis of the lattice

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T \cdot \mathbf{A} = \mathbf{0} \pmod{q}\}.$$

$$\begin{cases} \mathbf{A} \text{ public description of the lattice} \\ \mathbf{T_A} \text{ short basis, kept secret} \end{cases}$$

- Note that:
  1. Computing $\mathbf{T_A}$ *given* $\mathbf{A}$ is hard,
  2. Constructing $\mathbf{A}$ *together with* $\mathbf{T_A}$ is easy.



- With $\mathbf{T_A}$, we can sample short vectors in $\Lambda_q^{\perp}(\mathbf{A})$.

Our construction

Ingredients

- Certificate of users $\rightsquigarrow$ key to produce temporary certificate,
- Bonsai Tree signature [CashHofheinzKiltzPeikert12],
- ZKPoK using "Stern Extension" adapted from [LingNguyenStehléWang13].

Our scheme

- The member uses an interactive protocol to convince the verifier that he is a certified group member and he has not been revoked,
  - Repeated many times to make the soundness error negligibly small.
- Convert this protocol to a signature scheme via Fiat Shamir.

Generation of the keys

$N = 2^{\ell}$ group members

KeyGen

- Run TrapGen to get $\mathbf{A_0}$ together with a trapdoor $\mathbf{T_{A_0}}$,
- Sample $\mathbf{u}$ uniform in $\mathbb{Z}_q^n$,
- Sample 2ℓ public matrices $(\mathbf{A}_i^{(b)})$'s for $b \in \{0,1\}$, then define $\mathbf{A}$ and for each $d \in [N-1]$: $\mathbf{A}_d$ (as in a Bonsai signature),

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 \\ \hline \mathbf{A}_1^{(0)} \\ \hline \mathbf{A}_1^{(1)} \\ \hline \vdots \\ \hline \mathbf{A}_\ell^{(0)} \\ \hline \mathbf{A}_\ell^{(1)} \end{bmatrix} \in \mathbb{Z}_q^{(\ell+1)m\times n}, \text{ and } \mathbf{A}_d = \begin{bmatrix} \mathbf{A}_0 \\ \hline \mathbf{A}_1^{(d_1)} \\ \hline \dots \\ \hline \mathbf{A}_\ell^{(d_\ell)} \end{bmatrix} \in \mathbb{Z}_q^{(\ell+1)m\times n}.$$

Generation of the keys

$N = 2^{\ell}$ group members

KeyGen

- Run TrapGen to get $\mathbf{A_0}$ together with a trapdoor $\mathbf{T}_{\mathbf{A}_0}$,
- Sample $\mathbf{u}$ uniform in $\mathbb{Z}_q^n$,
- Sample 2ℓ public matrices $(\mathbf{A}_i^{(b)})$'s for $b \in \{0, 1\}$, then define $\mathbf{A}$ and for each $d \in [N-1]$: $\mathbf{A}_d$ (as in a Bonsai signature),
- For each d, sample a small $\mathbf{x}^d$ gaussian (using $\mathbf{T}_{\mathbf{A}_0}$), such that $(\mathbf{x}^d)^T \mathbf{A}_d = \mathbf{u}^T \bmod q$,

$$\left[\ (\mathbf{x}_0^{(d)})^T \ \Big\| \ (\mathbf{x}_1^{d_1})^T \ \Big\| \ \ldots \ \Big\| \ (\mathbf{x}_\ell^{d_\ell})^T \ \right] \left[\begin{array}{c} \mathbf{A}_0 \\ \hline \mathbf{A}_1^{(d_1)} \\ \hline \ldots \\ \hline \mathbf{A}_\ell^{(d_\ell)} \end{array}\right] = \mathbf{u}^T \bmod q$$

Generation of the keys

$N = 2^{\ell}$ group members

KeyGen

- Run TrapGen to get $\mathbf{A_0}$ together with a trapdoor $\mathbf{T}_{\mathbf{A}_0}$,
- Sample $\mathbf{u}$ uniform in $\mathbb{Z}_q^n$,
- Sample 2ℓ public matrices $(\mathbf{A}_i^{(b)})$'s for $b \in \{0,1\}$, then define $\mathbf{A}$ and for each $d \in [N-1]$: $\mathbf{A}_d$ (as in a Bonsai signature),
- For each d, sample a small $\mathbf{x}^d$ gaussian (using $\mathbf{T}_{\mathbf{A}_0}$), such that $(\mathbf{x}^d)^T \mathbf{A}_d = \mathbf{u}^T \bmod q$,

- Public key: $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$,
- Secret key for each d: $\mathsf{gsk}_d = \mathbf{x}^{(d)}$ such that $\mathbf{x}^{(d)} \mathbf{A}_d = \mathbf{u}^T \bmod q$,
$$\mathbf{x}^{(d)} = \left[\ (\mathbf{x}_0^{(d)})^T \ \middle\| \ (\mathbf{x}_1^{d_1})^T \ \middle\| \ \ldots \ \middle\| \ (\mathbf{x}_\ell^{d_\ell})^T \ \right].$$
- Revocation token for each d: $\mathsf{grt}_d = (\mathbf{x}_0^{(d)})^T \mathbf{A}_0$.

Sign

- To sign a message, the user must hide d
- $\Rightarrow$ he cannot convince a verifier that he knows $\mathbf{x}^{(d)}$ with $(\mathbf{x}^{(d)})^T \mathbf{A}_d = \mathbf{u}^T \bmod q$ if the verifier does not know $\mathbf{A}_d$.

Sign

- To sign a message, the user must hide d
- $\Rightarrow$ he cannot convince a verifier that he knows $\mathbf{x}^{(d)}$ with $(\mathbf{x}^{(d)})^T \mathbf{A}_d = \mathbf{u}^T \bmod q$ if the verifier does not know $\mathbf{A}_d$.

- Solution: prove that he knows $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A} = \mathbf{u}^T \bmod q$, and that for every two consecutive blocks of $\mathbf{x}^{(d)}$, one is a zero block.

Sign

- To sign a message, the user must hide d
- $\Rightarrow$ he cannot convince a verifier that he knows $\mathbf{x}^{(d)}$ with $(\mathbf{x}^{(d)})^T \mathbf{A}_d = \mathbf{u}^T \bmod q$ if the verifier does not know $\mathbf{A}_d$.

- Solution: prove that he knows $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A} = \mathbf{u}^T \bmod q$, and that for every two consecutive blocks of $\mathbf{x}^{(d)}$, one is a zero block.

- Recall that $\mathbf{x}^{(d)} = \left[\ (\mathbf{x}_0^{(d)})^T \ \| \ (\mathbf{x}_1^{d_1})^T \ \| \ \ldots \ \| \ (\mathbf{x}_\ell^{d_\ell})^T \ \right]$,
  Construct $\mathbf{x}$:

$$\left[\ (\mathbf{x}_0^{(d)})^T \ \| \ \underbrace{(\mathbf{x}_1^{d_1})^T \ \| \quad \mathbf{0} \quad}_{\text{if } d_1=0} \| \quad \ldots \quad \| \qquad\qquad \| \qquad\qquad \right]$$

Sign

- To sign a message, the user must hide d
- $\Rightarrow$ he cannot convince a verifier that he knows $\mathbf{x}^{(d)}$ with $(\mathbf{x}^{(d)})^T \mathbf{A}_d = \mathbf{u}^T \bmod q$ if the verifier does not know $\mathbf{A}_d$.

- Solution: prove that he knows $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A} = \mathbf{u}^T \bmod q$, and that for every two consecutive blocks of $\mathbf{x}^{(d)}$, one is a zero block.

- Recall that $\mathbf{x}^{(d)} = \left[\; (\mathbf{x}_0^{(d)})^T \,\|\, (\mathbf{x}_1^{d_1})^T \,\|\, \ldots \,\|\, (\mathbf{x}_\ell^{d_\ell})^T \;\right]$,
  Construct $\mathbf{x}$:
  $$\left[\; (\mathbf{x}_0^{(d)})^T \,\|\, \underbrace{\mathbf{0} \,\|\, (\mathbf{x}_1^{d_1})^T}_{\text{if } d_1=1} \,\|\, \ldots \,\|\qquad\|\qquad \;\right]$$

Sign

- To sign a message, the user must hide d
- $\Rightarrow$ he cannot convince a verifier that he knows $\mathbf{x}^{(d)}$ with $(\mathbf{x}^{(d)})^T \mathbf{A}_d = \mathbf{u}^T \bmod q$ if the verifier does not know $\mathbf{A}_d$.

- Solution: prove that he knows $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A} = \mathbf{u}^T \bmod q$, and that for every two consecutive blocks of $\mathbf{x}^{(d)}$, one is a zero block.

- Recall that $\mathbf{x}^{(d)} = \left[\ (\mathbf{x}_0^{(d)})^T \,\|\, (\mathbf{x}_1^{d_1})^T \,\|\, \ldots \,\|\, (\mathbf{x}_\ell^{d_\ell})^T\ \right]$,
  Construct $\mathbf{x}$:
  $$\left[\ (\mathbf{x}_0^{(d)})^T \,\|\, \underbrace{\mathbf{0} \,\|\, (\mathbf{x}_1^{d_1})^T}_{\text{if } d_1=1} \,\|\, \ldots \,\|\, \underbrace{(\mathbf{x}_\ell^{d_\ell})^T \,\|\, \mathbf{0}}_{\text{if } d_\ell=0}\ \right]$$

Sign

- To sign a message, the user must hide d
- $\Rightarrow$ he cannot convince a verifier that he knows $\mathbf{x}^{(d)}$ with $(\mathbf{x}^{(d)})^T \mathbf{A}_d = \mathbf{u}^T \bmod q$ if the verifier does not know $\mathbf{A}_d$.

- Solution: prove that he knows $\mathbf{x}$ such that $\mathbf{x}^T \mathbf{A} = \mathbf{u}^T \bmod q$, and that for every two consecutive blocks of $\mathbf{x}^{(d)}$, one is a zero block.

- Recall that $\mathbf{x}^{(d)} = \left[\ (\mathbf{x}_0^{(d)})^T \ \| \ (\mathbf{x}_1^{d_1})^T \ \| \ \ldots \ \| \ (\mathbf{x}_\ell^{d_\ell})^T \ \right]$,
  Construct $\mathbf{x}$:
$$\left[\ (\mathbf{x}_0^{(d)})^T \ \| \ \underbrace{\mathbf{0} \ \| \ (\mathbf{x}_1^{d_1})^T}_{\text{if } d_1=1} \| \ \ldots \ \| \ \underbrace{\mathbf{0} \ \| \ (\mathbf{x}_\ell^{d_\ell})^T}_{\text{if } d_\ell=1} \right]$$

  for example, if $d = 111\ldots1$:
$$\left[\ (\mathbf{x}_0^{(d)})^T \ \| \ \mathbf{0} \ \| \ (\mathbf{x}_1^{d_1})^T \ \| \ \ldots \ \| \ \mathbf{0} \ \| \ (\mathbf{x}_\ell^{d_\ell})^T \ \right] \begin{bmatrix} \mathbf{A}_0 \\ \hline \mathbf{A}_1^{(0)} \\ \hline \mathbf{A}_1^{(1)} \\ \hline \ldots \\ \hline \mathbf{A}_\ell^{(0)} \\ \hline \mathbf{A}_\ell^{(1)} \end{bmatrix} = \mathbf{u}^T \bmod q$$

Our construction

- Public parameters $\mathbf{A} \in \mathbb{Z}^{(\ell+1)m \times n}$ and $\mathbf{u} \in \mathbb{Z}_q^n$,
- Secret key $\mathbf{x}^{(d)}$.

- We propose an interactive Zero Knowledge protocol π which allows the user to prove knowledge of $\mathbf{x}^{(d)}$ (using $\mathbf{x}$),

- Verifier additional input: set $RL = \{(\mathbf{x}_0^{(d)})^T \mathbf{A}_0)_d\}$, for some d's.
- Prove that:
  - $\mathbf{x}^T \mathbf{A} = \mathbf{u}^T \bmod q$ and $\mathbf{x}$ of good shape,
  - $(\mathbf{x}_0^{(d)})^T \mathbf{A}_0 \notin RL$.

- ZKPoK $\rightsquigarrow$ made non-interactive *via* Fiat-Shamir, as a triple $\big(\{\mathrm{CMT}^{(k)}\}_{k=1}^t,\ \mathrm{CH},\ \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big)$, where

$$\mathrm{CH} = \big(\{Ch^{(k)}\}_{k=1}^t\big) = \mathcal{H}\big(M,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big) \in \{1,2,3\}^t.$$

(incorporating **the message** in π)

Performance and security

Size

- Size of the signatures: $\tilde{\mathcal{O}}(\lambda \cdot \log(N))$.
- Size of group public key : $\tilde{\mathcal{O}}(\lambda^2 \cdot \log(N))$.
- $\lambda = \Theta(n)$ is the security parameter.

Security in the Random Oracle Model:

Selfless anonymity

Simulation of the ZKPoK.

Traceability

Traceability under SIS, and extraction of information in the ZKPoK.

Conclusion

Our result

- We give the first lattice-based signature with verifier local revocation,
- We achieve logarithmic signature and public key sizes,
- Selfless anonymity and traceability (SIS).

Open problems

- Practice,
- Ring variants of SIS,
- Improving the sizes of the signature and public key,
- Removing the random oracle model.