Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures

Dan Boneh and **David Mandell Freeman**

Stanford University, USA

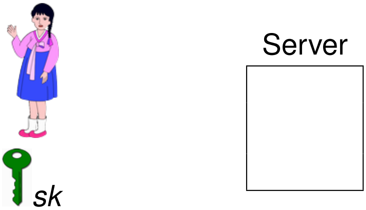
PKC 2011
Taormina, Italy
7 March 2011

Linearly Homomorphic Signatures

Linearly homomorphic signatures allow users to *authenticate vector subspaces* of a given ambient space.

Linearly Homomorphic Signatures

Linearly homomorphic signatures allow users to *authenticate vector subspaces* of a given ambient space.



Linearly homomorphic signatures allow users to *authenticate vector subspaces* of a given ambient space.

sk

signed $\longrightarrow$ vectors

Server

$\mathbf{v}_1$	σ_1
$\vdots$	$\vdots$
$\mathbf{v}_k$	σ_k

$\mathbf{v}_i \in \mathbb{F}_p^n$

$\sigma_i =$ signature on $\mathbf{v}_i$

Linearly Homomorphic Signatures

Linearly homomorphic signatures allow users to *authenticate vector subspaces* of a given ambient space.

sk

signed vectors $\longrightarrow$

Server

$\mathbf{v}_1$	σ_1
$\vdots$	$\vdots$
$\mathbf{v}_k$	σ_k

$$\xrightarrow[\sigma = f(\{c_i, \sigma_i\})]{\mathbf{v} = \sum c_i \mathbf{v}_i}$$

pk

$\mathbf{v}_i \in \mathbb{F}_p^n$
$\sigma_i =$ signature on $\mathbf{v}_i$

$\mathbf{v} \in \mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$
$\sigma =$ signature on $\mathbf{v}$

Linearly Homomorphic Signatures

Linearly homomorphic signatures allow users to *authenticate vector subspaces* of a given ambient space.

sk

$\xrightarrow{\text{signed}}$ vectors

Server

$\mathbf{v}_1$	σ_1
$\vdots$	$\vdots$
$\mathbf{v}_k$	σ_k

$$\xrightarrow[\sigma = f(\{c_i, \sigma_i\})]{\mathbf{v} = \sum c_i \mathbf{v}_i}$$

pk

$\mathbf{v}_i \in \mathbb{F}_p^n$
$\sigma_i =$ signature on $\mathbf{v}_i$

$\mathbf{v} \in \text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$
$\sigma =$ signature on $\mathbf{v}$

- Security: no adversary can authenticate any vector $\mathbf{v}^*$ outside $\text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$.

Motivation: Network Coding

Network coding routing mechanism [ACLY00]:

- Interpret data as vectors in $\mathbb{F}_p^n$.
- Routers send random linear combinations of received vectors, along with coefficients.
- Recipient reconstructs file from full-rank system.

Motivation: Network Coding

Network coding routing mechanism [ACLY00]:

- Interpret data as vectors in $\mathbb{F}_p^n$.
- Routers send random linear combinations of received vectors, along with coefficients.
- Recipient reconstructs file from full-rank system.

Problem: susceptible to pollution attacks.

- Recipient can't distinguish good packets from bad ones.

Motivation: Network Coding

Network coding routing mechanism [ACLY00]:

- Interpret data as vectors in $\mathbb{F}_p^n$.
- Routers send random linear combinations of received vectors, along with coefficients.
- Recipient reconstructs file from full-rank system.

Problem: susceptible to pollution attacks.

- Recipient can't distinguish good packets from bad ones.

Solution: linearly homomorphic signatures
[KFM04,ZKMH07,CJL09,BFKW09,GKKR10]

- Routers derive signature on lin. combinations; recipient verifies.

Motivation: Network Coding

Network coding routing mechanism [ACLY00]:

- Interpret data as vectors in $\mathbb{F}_p^n$.
- Routers send random linear combinations of received vectors, along with coefficients.
- Recipient reconstructs file from full-rank system.

Problem: susceptible to pollution attacks.

- Recipient can't distinguish good packets from bad ones.

Solution: linearly homomorphic signatures
[KFM04,ZKMH07,CJL09,BFKW09,GKKR10]

- Routers derive signature on lin. combinations; recipient verifies.

Current solutions authenticate vectors over $\mathbb{F}_p$ for large p.
For efficiency, we want to use vectors defined over $\mathbb{F}_2$.

Our Contributions

- Linearly homomorphic signatures over $\mathbb{F}_2$.
  - Secure under lattice assumptions, private unconditionally.
  - Primitive that can be constructed via lattice techniques, but not (currently) via dlog or factoring.

Our Contributions

- Linearly homomorphic signatures over $\mathbb{F}_2$.
  - Secure under lattice assumptions, private unconditionally.
  - Primitive that can be constructed via lattice techniques, but not (currently) via dlog or factoring.
- New tools for lattice-based cryptography.
  - New k-SIS assumption; reduction to worst-case lattice assumptions (used for security result).
  - Result on distributions of sums of discrete Gaussian samples (used for privacy result).
  - Tight length bounds for discrete Gaussian samples.

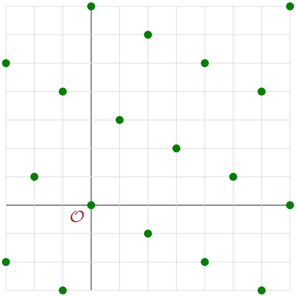
Our Contributions

- Linearly homomorphic signatures over $\mathbb{F}_2$.
  - Secure under lattice assumptions, private unconditionally.
  - Primitive that can be constructed via lattice techniques, but not (currently) via dlog or factoring.
- New tools for lattice-based cryptography.
  - New k-SIS assumption; reduction to worst-case lattice assumptions (used for security result).
  - Result on distributions of sums of discrete Gaussian samples (used for privacy result).
  - Tight length bounds for discrete Gaussian samples.
- k-time signature scheme without random oracles.
  - Application of new k-SIS assumption.

Building Block: GPV Trapdoor Function

- $\Lambda \subset \mathbb{Z}^m$ a lattice (full-rank subgroup), defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ $(n < m)$:

$$\Lambda = \Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{v} = 0 \bmod q\}$$

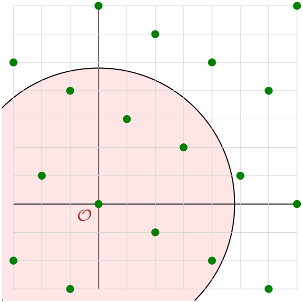


Building Block: GPV Trapdoor Function

- $\Lambda \subset \mathbb{Z}^m$ a lattice (full-rank subgroup), defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ ($n < m$):

$$\Lambda = \Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{v} = 0 \bmod q\}$$

- $D := \{\text{short vectors in } \mathbb{Z}^m\}$,

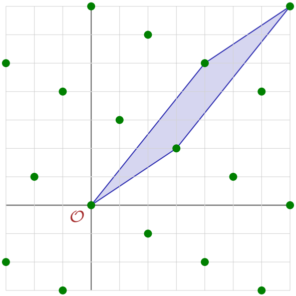


Building Block: GPV Trapdoor Function

- $\Lambda \subset \mathbb{Z}^m$ a lattice (full-rank subgroup), defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ $(n < m)$:

$$\Lambda = \Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{v} = 0 \bmod q\}$$

- $D :=$ {short vectors in $\mathbb{Z}^m$}, $R := \mathbb{Z}^m \bmod \Lambda \quad \cong \mathbb{Z}_q^n$.



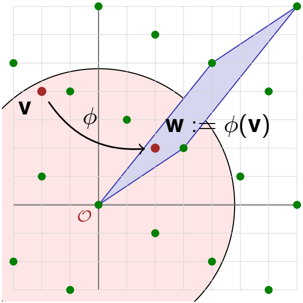
Building Block: GPV Trapdoor Function

- $\Lambda \subset \mathbb{Z}^m$ a lattice (full-rank subgroup), defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ ($n < m$):

$$\Lambda = \Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{v} = 0 \bmod q\}$$

- $D := \{\text{short vectors in } \mathbb{Z}^m\}$, $R := \mathbb{Z}^m \bmod \Lambda \quad \cong \mathbb{Z}_q^n$.
- GPV: define a *preimage-samplable trapdoor function* $\phi \colon D \to R$ by

$$\phi(\mathbf{v}) := \mathbf{v} \bmod \Lambda \;=\; \mathbf{A} \cdot \mathbf{v} \bmod q$$



Building Block: GPV Trapdoor Function

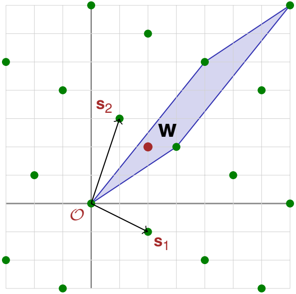
- $\Lambda \subset \mathbb{Z}^m$ a lattice (full-rank subgroup), defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ $(n < m)$:

$$\Lambda = \Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{v} = 0 \bmod q\}$$

- $D := \{\text{short vectors in } \mathbb{Z}^m\}$, $R := \mathbb{Z}^m \bmod \Lambda \quad \cong \mathbb{Z}_q^n$.
- GPV: define a *preimage-samplable trapdoor function* $\phi \colon D \to R$ by

$$\phi(\mathbf{v}) := \mathbf{v} \bmod \Lambda \;=\; \mathbf{A} \cdot \mathbf{v} \bmod q$$

- For any $\mathbf{w} \in R$, can sample short vectors in $\phi^{-1}(\mathbf{w}) = \Lambda + \mathbf{w}$ given a "short" basis of Λ.



Building Block: GPV Trapdoor Function

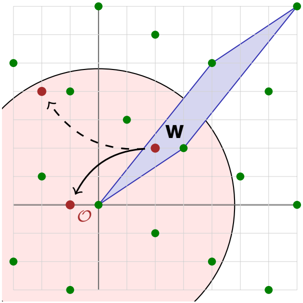
- $\Lambda \subset \mathbb{Z}^m$ a lattice (full-rank subgroup), defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ $(n < m)$:

$$\Lambda = \Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{v} = 0 \bmod q\}$$

- $D :=$ {short vectors in $\mathbb{Z}^m$},
  $R := \mathbb{Z}^m \bmod \Lambda \quad \cong \mathbb{Z}_q^n$.
- GPV: define a *preimage-samplable trapdoor function* $\phi \colon D \to R$ by

$$\phi(\mathbf{v}) := \mathbf{v} \bmod \Lambda \;=\; \mathbf{A} \cdot \mathbf{v} \bmod q$$

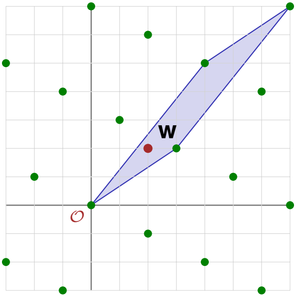
- For any $\mathbf{w} \in R$, can sample short vectors in $\phi^{-1}(\mathbf{w}) = \Lambda + \mathbf{w}$ given a "short" basis of Λ.



Building Block: GPV Trapdoor Function

- $\Lambda \subset \mathbb{Z}^m$ a lattice (full-rank subgroup), defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ ($n < m$):

$$\Lambda = \Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{v} = 0 \bmod q\}$$



- $D := \{\text{short vectors in } \mathbb{Z}^m\}$, $R := \mathbb{Z}^m \bmod \Lambda \quad \cong \mathbb{Z}_q^n$.
- GPV: define a *preimage-samplable trapdoor function* $\phi \colon D \to R$ by

$$\phi(\mathbf{v}) := \mathbf{v} \bmod \Lambda \ = \ \mathbf{A} \cdot \mathbf{v} \bmod q$$

- For any $\mathbf{w} \in R$, can sample short vectors in $\phi^{-1}(\mathbf{w}) = \Lambda + \mathbf{w}$ given a "short" basis of Λ.
- Sampling short vectors in $\Lambda + \mathbf{w}$ without short basis is hard.

Linearly Homomorphic Signatures: Key Ideas

GPV sign/verify algorithms: $H\colon \{0,1\}^* \to \mathbb{Z}_q^n$

$pk = \mathbf{A} \in \mathbb{Z}_q^{n\times m}$ $sk =$ short basis of $\Lambda_q^\perp(\mathbf{A})$

Sign($\mathbf{v}$) := short vector in $(\Lambda_q^\perp(\mathbf{A}) + H(\mathbf{v}))$

Verify(σ) := 1 iff σ is short, $\mathbf{A}\cdot\sigma \bmod q = H(\mathbf{v})$

Linearly Homomorphic Signatures: Key Ideas

GPV sign/verify algorithms: $H\colon \{0,1\}^* \to \mathbb{Z}_q^n$

$$pk = \mathbf{A} \in \mathbb{Z}_q^{n\times m} \quad sk = \text{short basis of } \Lambda_q^\perp(\mathbf{A})$$
$$\text{Sign}(\mathbf{v}) := \text{short vector in } (\Lambda_q^\perp(\mathbf{A}) + H(\mathbf{v}))$$
$$\text{Verify}(\sigma) := 1 \text{ iff } \sigma \text{ is short}, \ \mathbf{A}\cdot\sigma \bmod q = H(\mathbf{v})$$

Idea: instead of hashing, use lattice $\Lambda_{2q}^\perp(\mathbf{A})$ defined mod $2q$:

- mod 2 part encodes a vector $\mathbf{v} \in \mathbb{F}_2^n$.
- mod q part encodes solution to a hard problem.

Linearly Homomorphic Signatures: Key Ideas

New sign/verify algorithms: $\mathbf{v} \in \mathbb{F}_2^n$, q odd

$$pk = \mathbf{A} \in \mathbb{Z}_{2q}^{n \times m} \quad sk = \text{short basis of } \Lambda_{2q}^{\perp}(\mathbf{A})$$

$$\text{Sign}(\mathbf{v}) := \text{short vector in } (\Lambda_{2q}^{\perp}(\mathbf{A}) + q \cdot \mathbf{v})$$

$$\text{Verify}(\sigma) := 1 \quad \text{iff} \quad \sigma \text{ is short, } \mathbf{A} \cdot \sigma = \begin{cases} \mathbf{v} \bmod 2 \\ 0 \bmod q \end{cases}$$

Idea: instead of hashing, use lattice $\Lambda_{2q}^{\perp}(\mathbf{A})$ defined mod $2q$:

- mod 2 part encodes a vector $\mathbf{v} \in \mathbb{F}_2^n$.
- mod q part encodes solution to a hard problem.

Linearly Homomorphic Signatures: Key Ideas

New sign/verify algorithms: $\mathbf{v} \in \mathbb{F}_2^n$, q odd

$$pk = \mathbf{A} \in \mathbb{Z}_{2q}^{n \times m} \qquad sk = \text{short basis of } \Lambda_{2q}^{\perp}(\mathbf{A})$$

$$\text{Sign}(\mathbf{v}) \quad := \quad \text{short vector in} \quad (\Lambda_{2q}^{\perp}(\mathbf{A}) + q \cdot \mathbf{v})$$

$$\text{Verify}(\sigma) \quad := \quad 1 \quad \text{iff} \quad \sigma \text{ is short}, \ \mathbf{A} \cdot \sigma = \begin{cases} \mathbf{v} \bmod 2 \\ 0 \bmod q \end{cases}$$

Idea: instead of hashing, use lattice $\Lambda_{2q}^{\perp}(\mathbf{A})$ defined mod $2q$:

- mod 2 part encodes a vector $\mathbf{v} \in \mathbb{F}_2^n$.
- mod q part encodes solution to a hard problem.

Homomorphic property: "mod $2q$" is a linear map, so adding signatures corresponds to adding messages.

Linearly Homomorphic Signatures: Key Ideas

New sign/verify algorithms: $\mathbf{v} \in \mathbb{F}_2^n$, q odd

$$pk = \mathbf{A} \in \mathbb{Z}_{2q}^{n\times m} \qquad sk = \text{short basis of } \Lambda_{2q}^{\perp}(\mathbf{A})$$

$$\text{Sign}(\mathbf{v}) \quad := \quad \text{short vector in} \quad (\Lambda_{2q}^{\perp}(\mathbf{A}) + q \cdot \mathbf{v})$$

$$\text{Verify}(\sigma) \quad := \quad 1 \quad \text{iff} \quad \sigma \text{ is short, } \ \mathbf{A} \cdot \sigma = \begin{cases} \mathbf{v} \bmod 2 \\ 0 \bmod q \end{cases}$$

Idea: instead of hashing, use lattice $\Lambda_{2q}^{\perp}(\mathbf{A})$ defined mod $2q$:

- mod 2 part encodes a vector $\mathbf{v} \in \mathbb{F}_2^n$.
- mod q part encodes solution to a hard problem.

Homomorphic property: "mod $2q$" is a linear map, so adding signatures corresponds to adding messages.

- Suppose σ_1, σ_2 are signatures on $\mathbf{v}_1, \mathbf{v}_2$
  $\Rightarrow \sigma_i$ short, $\mathbf{A} \cdot \sigma_i \bmod 2q = q \cdot \mathbf{v}_i$.

Linearly Homomorphic Signatures: Key Ideas

New sign/verify algorithms: $\mathbf{v} \in \mathbb{F}_2^n$, q odd

$$pk = \mathbf{A} \in \mathbb{Z}_{2q}^{n\times m} \qquad sk = \text{short basis of } \Lambda_{2q}^{\perp}(\mathbf{A})$$

$$\text{Sign}(\mathbf{v}) := \text{short vector in } (\Lambda_{2q}^{\perp}(\mathbf{A}) + q\cdot\mathbf{v})$$

$$\text{Verify}(\sigma) := 1 \quad \text{iff} \quad \sigma \text{ is short, } \mathbf{A}\cdot\sigma = \begin{cases} \mathbf{v} \bmod 2 \\ 0 \bmod q \end{cases}$$

Idea: instead of hashing, use lattice $\Lambda_{2q}^{\perp}(\mathbf{A})$ defined mod $2q$:

- mod 2 part encodes a vector $\mathbf{v} \in \mathbb{F}_2^n$.
- mod q part encodes solution to a hard problem.

Homomorphic property: "mod $2q$" is a linear map, so adding signatures corresponds to adding messages.

- Suppose σ_1, σ_2 are signatures on $\mathbf{v}_1, \mathbf{v}_2$
  $\Rightarrow \sigma_i$ short, $\mathbf{A}\cdot\sigma_i \bmod 2q = q\cdot\mathbf{v}_i$.
- Define signature on $\mathbf{v}_1 + \mathbf{v}_2$ to be $\sigma := \sigma_1 + \sigma_2$.
  $\Rightarrow \sigma$ is short, $\mathbf{A}\cdot\sigma \bmod 2q = q\cdot(\mathbf{v}_1 + \mathbf{v}_2)$.

Security Analysis

Goal: Reduce system's security to the following problem.

$\text{SIS}_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$,
find an $\mathbf{v}^* \in \Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$.

Security Analysis

Goal: Reduce system's security to the following problem.

$\text{SIS}_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$,
find an $\mathbf{v}^* \in \Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$.

Theorem [MR04,GPV08]: An algorithm that solves SIS can be used to solve worst-case lattice problems (e.g., GapSVP, SIVP).

Security Analysis

Goal: Reduce system's security to the following problem.

$\mathsf{SIS}_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$,
find an $\mathbf{v}^* \in \Lambda_q^\perp(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$.

Theorem [MR04,GPV08]: An algorithm that solves SIS can be used to solve worst-case lattice problems (e.g., GapSVP, SIVP).

- Problem: signatures are already short vectors in $\Lambda_q^\perp(\mathbf{A})$, so can't simulate in a reduction.

Security Analysis

Goal: Reduce system's security to the following problem.

$\mathsf{SIS}_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$,
find an $\mathbf{v}^* \in \Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$.

Theorem [MR04,GPV08]: An algorithm that solves SIS can be used to solve worst-case lattice problems (e.g., GapSVP, SIVP).

- Problem: signatures are already short vectors in $\Lambda_q^{\perp}(\mathbf{A})$, so can't simulate in a reduction.
- Solution: Make a new assumption!
  (and then reduce it to a standard assumption).

Goal: Reduce system's security to the following problem.

k-$\text{SIS}_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and k short vectors $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \Lambda_q^\perp(\mathbf{A})$ find an $\mathbf{v}^* \in \Lambda_q^\perp(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$ and $\mathbf{e}^* \notin \mathbb{Q}\text{-span}(\mathbf{e}_1, \ldots, \mathbf{e}_k)$.

Security Analysis

Goal: Reduce system's security to the following problem.

k-SIS$_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and k short vectors $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \Lambda_q^{\perp}(\mathbf{A})$ find an $\mathbf{v}^* \in \Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$ and $\mathbf{e}^* \notin \mathbb{Q}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k)$.

Theorem: An adversary that forges a signature (in the random oracle model) can be used to solve the k-SIS$_{q,m,\beta}$ problem.

Security Analysis

Goal: Reduce system's security to the following problem.

k-SIS$_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and k short vectors $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \Lambda_q^{\perp}(\mathbf{A})$ find an $\mathbf{v}^* \in \Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$ and $\mathbf{e}^* \notin \mathbb{Q}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k)$.

Theorem: An adversary that forges a signature (in the random oracle model) can be used to solve the k-SIS$_{q,m,\beta}$ problem.

Theorem: An algorithm that solves the k-SIS$_{q,m,\beta}$ problem can be used to solve SIS$_{q,m-k,\beta'}$.

Goal: Reduce system's security to the following problem.

k-SIS$_{q,m,\beta}$ Problem

Given random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and k short vectors $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \Lambda_q^{\perp}(\mathbf{A})$ find an $\mathbf{v}^* \in \Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{v}^*\| < \beta$ and $\mathbf{e}^* \notin \mathbb{Q}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k)$.

Theorem: An adversary that forges a signature (in the random oracle model) can be used to solve the k-SIS$_{q,m,\beta}$ problem.

Theorem: An algorithm that solves the k-SIS$_{q,m,\beta}$ problem can be used to solve SIS$_{q,m-k,\beta'}$.

Sadly, the k-SIS-to-SIS reduction is exponential in k:

$$\beta' \approx k! \cdot n^{k/2} \cdot \beta.$$

But this is OK if $k = O(1)$.

Idea of the k-SIS-to-SIS Reduction

Given SIS challenge $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, do:

- Choose $\mathbf{e}_1, \ldots, \mathbf{e}_k$ from Gaussians over $\mathbb{Z}^{m+k}$.
- Define $\mathbf{B}$ by appending k random columns to $\mathbf{A}$ such that

$$\underbrace{\left(\begin{array}{c|ccc} \mathbf{A} & \mathbf{b}_1 & \cdots & \mathbf{b}_k \end{array} \right)}_{\mathbf{B}} \cdot \begin{pmatrix} | \\ \mathbf{e}_i \\ | \end{pmatrix} = 0 \bmod q \quad \text{for all } i$$

Idea of the k-SIS-to-SIS Reduction

Given SIS challenge $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, do:

- Choose $\mathbf{e}_1, \ldots, \mathbf{e}_k$ from Gaussians over $\mathbb{Z}^{m+k}$.
- Define $\mathbf{B}$ by appending k random columns to $\mathbf{A}$ such that

$$\underbrace{\left(\begin{array}{c|ccc} \mathbf{A} & \mathbf{b}_1 & \cdots & \mathbf{b}_k \end{array} \right)}_{\mathbf{B}} \cdot \left(\mathbf{e}_i \right) = 0 \bmod q \quad \text{for all } i$$

Theorem

$(\mathbf{B}, \mathbf{e}_1, \ldots, \mathbf{e}_k)$ *produced in this way is statistically indistinguishable from a k-SIS challenge in dimension $m + k$.*

Real k-SIS challenge: fix $\mathbf{B}$, then choose $\mathbf{e}_i \in \Lambda_q^{\perp}(\mathbf{B})$.

k-SIS-to-SIS Reduction, Continued

Given simulated k-SIS challenge $(\mathbf{B}, \mathbf{e}_1, \ldots, \mathbf{e}_k)$

$$\underbrace{\left(\begin{array}{c||ccc} \mathbf{A} & \mathbf{b}_1 & \cdots & \mathbf{b}_k \end{array} \right)}_{\mathbf{B}} \cdot \left(\begin{array}{ccc} \mathbf{e}_1 & \cdots & \mathbf{e}_k \end{array} \right) = 0 \bmod q$$

k-SIS-to-SIS Reduction, Continued

Given simulated k-SIS challenge $(\mathbf{B}, \mathbf{e}_1, \ldots, \mathbf{e}_k)$

$$\underbrace{\left(\begin{array}{c||ccc} \mathbf{A} & \mathbf{b}_1 & \cdots & \mathbf{b}_k \end{array} \right)}_{\mathbf{B}} \cdot \left(\begin{array}{cccc} \mathbf{e}_1 & \cdots & \mathbf{e}_k & \mathbf{e}^* \end{array} \right) = 0 \bmod q$$

k-SIS adversary produces $\mathbf{e}^* \in \Lambda_q^\perp(\mathbf{B})$ not in $\mathbb{Q}$-span$(\mathbf{e}_1 \ldots, \mathbf{e}_k)$.

k-SIS-to-SIS Reduction, Continued

Given simulated k-SIS challenge $(\mathbf{B}, \mathbf{e}_1, \ldots, \mathbf{e}_k)$

$$\underbrace{\left(\begin{array}{c||ccc} \mathbf{A} & \mathbf{b}_1 & \cdots & \mathbf{b}_k \end{array} \right)}_{\mathbf{B}} \cdot \begin{pmatrix} \mathbf{v}^* \\ 0 \\ 0 \end{pmatrix} = 0 \bmod q$$

k-SIS adversary produces $\mathbf{e}^* \in \Lambda_q^{\perp}(\mathbf{B})$ not in $\mathbb{Q}$-span$(\mathbf{e}_1 \ldots, \mathbf{e}_k)$.

- Use Gaussian elimination over $\mathbb{Z}$ to find short nonzero $\mathbf{v}^* \in \mathbb{Z}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k, \mathbf{e}^*)$ with last k entries 0.

k-SIS-to-SIS Reduction, Continued

Given simulated k-SIS challenge $(\mathbf{B}, \mathbf{e}_1, \ldots, \mathbf{e}_k)$

$$\underbrace{\left(\begin{array}{c||ccc} \mathbf{A} & \mathbf{b}_1 & \cdots & \mathbf{b}_k \end{array} \right)}_{\mathbf{B}} \cdot \begin{pmatrix} \mathbf{v}^* \\ 0 \\ 0 \end{pmatrix} = 0 \bmod q$$

k-SIS adversary produces $\mathbf{e}^* \in \Lambda_q^{\perp}(\mathbf{B})$ not in $\mathbb{Q}$-span$(\mathbf{e}_1 \ldots, \mathbf{e}_k)$.

- Use Gaussian elimination over $\mathbb{Z}$ to find short nonzero $\mathbf{v}^* \in \mathbb{Z}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k, \mathbf{e}^*)$ with last k entries 0.
- First m entries of $\mathbf{v}^*$ are in $\Lambda_q^{\perp}(\mathbf{A})$ — solves SIS problem!

k-SIS-to-SIS Reduction, Continued

Given simulated k-SIS challenge $(\mathbf{B}, \mathbf{e}_1, \ldots, \mathbf{e}_k)$

$$\underbrace{\left(\begin{array}{c||ccc} \mathbf{A} & \mathbf{b}_1 & \cdots & \mathbf{b}_k \end{array} \right)}_{\mathbf{B}} \cdot \begin{pmatrix} \mathbf{v}^* \\ 0 \\ 0 \end{pmatrix} = 0 \bmod q$$

k-SIS adversary produces $\mathbf{e}^* \in \Lambda_q^\perp(\mathbf{B})$ not in $\mathbb{Q}$-span$(\mathbf{e}_1 \ldots, \mathbf{e}_k)$.

- Use Gaussian elimination over $\mathbb{Z}$ to find short nonzero $\mathbf{v}^* \in \mathbb{Z}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k, \mathbf{e}^*)$ with last k entries 0.
- First m entries of $\mathbf{v}^*$ are in $\Lambda_q^\perp(\mathbf{A})$ — solves SIS problem!

Gaussian elimination blows up length by a factor $\approx k! \cdot n^{k/2}$.

Privacy

Privacy property: derived signature on $\mathbf{v} = \sum c_i \mathbf{v}_i$ reveals nothing about $\mathbf{v}_1, \ldots, \mathbf{v}_k$ beyond value of $\mathbf{v}$.

Privacy

Privacy property: derived signature on $\mathbf{v} = \sum c_i \mathbf{v}_i$ reveals nothing about $\mathbf{v}_1, \ldots, \mathbf{v}_k$ beyond value of $\mathbf{v}$.

Specifically: given two vector spaces

$$V = \text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k), \qquad W = \text{span}(\mathbf{w}_1, \ldots, \mathbf{w}_k)$$

and a set of coefficients $\{c_i\}$ with

$$\sum c_i \mathbf{v}_i = \sum c_i \mathbf{w}_i,$$

even unbounded adversary cannot distinguish derived signature on $\sum c_i \mathbf{v}_i$ from derived signature on $\sum c_i \mathbf{w}_i$.

New Tool Used to Prove Privacy

Theorem

Let $\mathbf{e}_i \in \mathbb{Z}^m$ be sampled from a discrete Gaussian over $\Lambda + \mathbf{t}_i$ with parameter σ. Let $c_i \in \{0, 1\}$. Then for sufficiently large σ, the distribution of $\sum c_i \mathbf{e}_i$ is a discrete Gaussian over $\Lambda + \sum c_i \mathbf{t}_i$.*

*up to negligible statistical distance

New Tool Used to Prove Privacy

Theorem

Let $\mathbf{e}_i \in \mathbb{Z}^m$ be sampled from a discrete Gaussian over $\Lambda + \mathbf{t}_i$ with parameter σ. Let $c_i \in \{0, 1\}$. Then for sufficiently large σ, the distribution of $\sum c_i \mathbf{e}_i$ is a discrete Gaussian over $\Lambda + \sum c_i \mathbf{t}_i$.*

Corollary: Linearly homomorphic signatures over $\mathbb{F}_2$ are private.

Proof idea:

- Sigs on $\mathbf{v}_i$ sampled from discrete Gaussian distribution, derived sigs are linear combinations.
- By theorem, distribution of derived signature on $\mathbf{v} = \sum c_i \mathbf{v}_i$ depends only on $\{c_i\}$ and $\mathbf{v}$, not on the $\mathbf{v}_i$.
- If $\sum c_i \mathbf{v}_i = \sum c_i \mathbf{w}_i$, derived sig distributions are identical*.

*up to negligible statistical distance

New Tool Used to Prove Privacy

Theorem

Let $\mathbf{e}_i \in \mathbb{Z}^m$ be sampled from a discrete Gaussian over $\Lambda + \mathbf{t}_i$ with parameter σ. Let $c_i \in \{0, 1\}$. Then for sufficiently large σ, the distribution of $\sum c_i \mathbf{e}_i$ is a discrete Gaussian over $\Lambda + \sum c_i \mathbf{t}_i$.*

Corollary: Linearly homomorphic signatures over $\mathbb{F}_2$ are private.

Proof idea:

- Sigs on $\mathbf{v}_i$ sampled from discrete Gaussian distribution, derived sigs are linear combinations.
- By theorem, distribution of derived signature on $\mathbf{v} = \sum c_i \mathbf{v}_i$ depends only on $\{c_i\}$ and $\mathbf{v}$, not on the $\mathbf{v}_i$.
- If $\sum c_i \mathbf{v}_i = \sum c_i \mathbf{w}_i$, derived sig distributions are identical*.

Theorem generalizes to tuples of discrete Gaussians.

*up to negligible statistical distance

Other Contributions

A k-time signature scheme without random oracles:

Other Contributions

A *k*-time signature scheme without random oracles:

- Sign/Verify algorithms same as in homomorphic scheme:

$$\begin{aligned} \mathrm{Sign}(\mathbf{v}) &:= \text{Gaussian sample from} \quad (\Lambda_{2q}^{\perp}(\mathbf{A}) + q \cdot \mathbf{v}) \\ \mathrm{Verify}(\sigma) &:= 1 \quad \text{iff} \quad \|\sigma\| < \beta, \; \mathbf{A} \cdot \sigma = q \cdot \mathbf{v} \bmod 2q. \end{aligned}$$

Other Contributions

A *k*-time signature scheme without random oracles:

- Sign/Verify algorithms same as in homomorphic scheme:

$$\begin{aligned} \text{Sign}(\mathbf{v}) &:= \text{Gaussian sample from } (\Lambda^{\perp}_{2q}(\mathbf{A}) + q \cdot \mathbf{v}) \\ \text{Verify}(\sigma) &:= 1 \quad \text{iff} \quad \|\sigma\| < \beta, \;\; \mathbf{A} \cdot \sigma = q \cdot \mathbf{v} \bmod 2q. \end{aligned}$$

- Eliminate homomorphic property by choosing small β:
  $\sigma_1 + \sigma_2$ now too long to verify for $\mathbf{v}_1 + \mathbf{v}_2$.

Other Contributions

A *k*-time signature scheme without random oracles:

- Sign/Verify algorithms same as in homomorphic scheme:

$$\begin{aligned} \text{Sign}(\mathbf{v}) &:= \text{Gaussian sample from} \quad (\Lambda_{2q}^{\perp}(\mathbf{A}) + q \cdot \mathbf{v}) \\ \text{Verify}(\sigma) &:= 1 \quad \text{iff} \quad \|\sigma\| < \beta, \ \ \mathbf{A} \cdot \sigma = q \cdot \mathbf{v} \bmod 2q. \end{aligned}$$

- Eliminate homomorphic property by choosing small β: $\sigma_1 + \sigma_2$ now too long to verify for $\mathbf{v}_1 + \mathbf{v}_2$.
- Requires tight bound on length of Gaussian samples.

Other Contributions

A k-time signature scheme without random oracles:

- Sign/Verify algorithms same as in homomorphic scheme:

$$\begin{aligned} \text{Sign}(\mathbf{v}) &:= \text{Gaussian sample from} \quad (\Lambda_{2q}^{\perp}(\mathbf{A}) + q \cdot \mathbf{v}) \\ \text{Verify}(\sigma) &:= 1 \quad \text{iff} \quad \|\sigma\| < \beta, \;\; \mathbf{A} \cdot \sigma = q \cdot \mathbf{v} \bmod 2q. \end{aligned}$$

- Eliminate homomorphic property by choosing small β: $\sigma_1 + \sigma_2$ now too long to verify for $\mathbf{v}_1 + \mathbf{v}_2$.
- Requires tight bound on length of Gaussian samples.

Theorem

Let $\mathbf{e} \in \mathbb{Z}^n$ be sampled from a discrete Gaussian with parameter σ. Then for any $\epsilon > 0$ we have w.h.p.

$$(1 - \epsilon) \cdot \sigma \sqrt{n/2\pi} \leq \|\mathbf{e}\| \leq (1 + \epsilon) \cdot \sigma \sqrt{n/2\pi}.$$

Best previous result was $\|\mathbf{e}\| \leq \sigma\sqrt{n}$.

Open Problems

1. Find a better k-SIS $\rightarrow$ SIS reduction.
   - Current reduction is exponential in k.
   - System can only sign $k = O(1)$ vectors while maintaining security based on worst-case problems.

Open Problems

1. Find a better k-SIS $\rightarrow$ SIS reduction.
   - Current reduction is exponential in k.
   - System can only sign $k = O(1)$ vectors while maintaining security based on worst-case problems.

2. Homomorphic signatures over $\mathbb{F}_2$ with worst-case security for $k = \text{poly}(n)$.
   - Achieved in BF eprint 2011/018:
     "Homomorphic Signatures for Polynomial Functions."

Open Problems

1. Find a better k-SIS $\to$ SIS reduction.
   - Current reduction is exponential in k.
   - System can only sign $k = O(1)$ vectors while maintaining security based on worst-case problems.

2. Homomorphic signatures over $\mathbb{F}_2$ with worst-case security for $k = \text{poly}(n)$.
   - Achieved in BF eprint 2011/018:
     "Homomorphic Signatures for Polynomial Functions."

3. Remove random oracle from security proof.
   - Adapt techniques from the next talk to lattice setting?

Open Problems

1. Find a better k-SIS $\rightarrow$ SIS reduction.
   - Current reduction is exponential in k.
   - System can only sign $k = O(1)$ vectors while maintaining security based on worst-case problems.

2. Homomorphic signatures over $\mathbb{F}_2$ with worst-case security for $k = \text{poly}(n)$.
   - Achieved in BF eprint 2011/018:
     "Homomorphic Signatures for Polynomial Functions."

3. Remove random oracle from security proof.
   - Adapt techniques from the next talk to lattice setting?

4. Find other applications of the k-SIS tool.

Open Problems

1. Find a better k-SIS $\to$ SIS reduction.
   - Current reduction is exponential in k.
   - System can only sign $k = O(1)$ vectors while maintaining security based on worst-case problems.
2. Homomorphic signatures over $\mathbb{F}_2$ with worst-case security for $k = \text{poly}(n)$.
   - Achieved in BF eprint 2011/018:
     "Homomorphic Signatures for Polynomial Functions."
3. Remove random oracle from security proof.
   - Adapt techniques from the next talk to lattice setting?
4. Find other applications of the k-SIS tool.

Thank you!