Lattice Signature Schemes

Vadim Lyubashevsky
INRIA / ENS Paris
DIGITAL SIGNATURE SCHEMES
Digital Signatures

\[(sk, pk) \leftarrow \text{KeyGen}\]
\[\text{Sign}(sk, m_i) = s_i\]
\[\text{Verify}(pk, m_i, s_i) = \text{YES} / \text{NO}\]

Correctness: \[\text{Verify}(pk, m_i, \text{Sign}(sk, m_i)) = \text{YES}\]

Security: Unforgeability
1. Adversary gets pk
2. Adversary asks for signatures of \[m_1, m_2, \ldots\]
3. Adversary outputs \((m, s)\) where \(m \neq m_i\) and wins if \[\text{Verify}(pk, m, s) = \text{YES}\]
Signature Schemes

• Hash-and-Sign
  – Requires a trap-door function

• Fiat-Shamir transformation
  – Conversion from an identification scheme
  – No trap-door function needed
FIAT-SHAMIR SIGNATURE SCHEMES
Signature Scheme (Main Idea)

Secret Key: $S$
Public Key: $A, T = AS \mod q$

**Sign($\mu$)**
- Pick a random $y$
- Compute $c = H(Ay \mod q, \mu)$
- $z = Sc + y$
- Output$(z, c)$

**Verify($z, c$)**
- Check that $z$ is “small” and
- $c = H(Az - Tc \mod q, \mu)$
Main Security Intuition

Secret Key: $S$
Public Key: $A$, $T = AS \mod q$

**Sign($\mu$)**
- Pick a random $y$
- Compute $c = H(Ay \mod q, \mu)$
- $z = Sc + y$
- Output($z, c$)

**Verify($z, c$)**
- Check that $z$ is “small” and $c = H(Az - Tc \mod q, \mu)$

*Signature is independent of the secret key*
Signature Scheme

Secret Key: \( S \)
Public Key: \( A, T = AS \mod q \)

**Sign(\( \mu \))**

- Pick a random \( y \) make \( y \) uniformly random mod \( q \)?
- Compute \( c = H(Ay \mod q, \mu) \)
- \( z = Sc + y \)
- **Output(\( z, c \))** then \( z \) is too big and forging is easy 😞
Signature Scheme

Secret Key: $S$
Public Key: $A$, $T = AS \mod q$

\textbf{Sign}(\mu)

Pick a random $y$
Compute $c = H(Ay \mod q, \mu)$
$z = Sc + y$
Output($z, c$) then $z$ will not be independent of $S$ 😞
Rejection Sampling

Secret Key: $S$
Public Key: $A$, $T = AS \mod q$

**Sign($\mu$)**
Pick a random $y$  make $y$ small
Compute $c = H(Ay \mod q, \mu)$
$z = Sc + y$
Output($z, c$) *if* $z$ meets certain criteria, *else* repeat
Rejection Sampling

Have access to samples from \( g(x) \)

Want \( f(x) \)
Rejection Sampling

Have access to samples from $g(x)$

Want $f(x)$

Sample from $g(x)$, accept $x$ with probability $\frac{f(x)}{Mg(x)} \leq 1$

$$\Pr[x] = g(x) \cdot \left( \frac{f(x)}{Mg(x)} \right) = \frac{f(x)}{M}$$

Something is output with probability $\frac{1}{M}$
Rejection Sampling

Impossible to tell whether $g(x)$ or $h(x)$ was the original distribution

Have access to samples from $g(x)$

Want $f(x)$

Sample from $g(x)$, accept $x$ with probability $f(x)/Mg(x) \leq 1$

or ... Sample from $h(x)$, accept $x$ with probability $f(x)/Mh(x) \leq 1$

$Pr[x] = g(x) \cdot (f(x)/Mg(x)) = f(x)/M = h(x) \cdot (f(x)/Mh(x))$

Something is output with probability $1/M$
Rejection Sampling

Pick a random $y$
Compute $c = H(Ay \mod q, \mu)$
$z = Sc + y$
Output $(z, c)$ w.p. ...
Normal Distribution

1-dimensional Normal distribution:

\[ \rho_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-x^2/2\sigma^2} \]

It is:

Centered at 0
Standard deviation: \( \sigma \)
Examples
Shifted Normal Distribution

1-dimensional shifted Normal distribution:

\[ \rho_{\sigma,\nu}(x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\nu)^2}{2\sigma^2}} \]

It is:

Centered at \( \nu \)

Standard deviation: \( \sigma \)
n-Dimensional Normal Distribution

n-dimensional shifted Normal distribution:

\[ \rho_{\sigma, v}(x) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{||x-v||^2}{2\sigma^2}} \]

It is:

- Centered at \( v \)
- Standard deviation: \( \sigma \)
2-Dimensional Example
n-Dimensional Normal Distribution

n-dimensional shifted Normal distribution:

\[ \rho_{\sigma, v}(x) = \frac{1}{(\sqrt{2\pi\sigma})^n} e^{-\frac{|x-v|^2}{2\sigma^2}} \]

It is:

Centered at \( v \)

Standard deviation: \( \sigma \)

Discrete Normal: for \( x \) in \( \mathbb{Z}^n \),

\[ D_{\sigma, v}(x) = \frac{\rho_{\sigma, v}(x)}{\rho_{\sigma, v}(\mathbb{Z}^n)} \]
Rejection Sampling

Pick a random $y$
Compute $c = H(Ay \mod q, \mu)$
$z = Sc + y$
Output $(z, c)$ w.p. $D_{\sigma,0}(z) / (MD_{\sigma,sc}(z))$

for $\sigma = 12v$,
$D_{\sigma,0}(z) / (MD_{\sigma,sc}(z)) \approx e/M$

$v = \max ||Sc||$
Security Reduction

Adversary Simulator

A

Pick random $S$

$(z_i, c_i) = \text{Sign}(\mu_i)$

$A(z-z') + T(c' - c) = 0$

$A(z-z' + Sc' - Sc) = 0$

If this is not 0, then $\textbf{SIS}$ is solved.

Important for adversary to not know $S$. 

$\mu_i$

$(z_i, c_i)$

$\mu$, $(z, c)$

$\mu$, $(z', c')$
INTERLUDE: THE SIS PROBLEM
The SIS Problem

Given a random $A$ in $\mathbb{Z}_q^{n \times m}$, find a “small” $s$ such that $As = 0 \mod q$
The LWE Problem

\[ \mathbf{A} \mathbf{s} + \mathbf{e} = \mathbf{b} \mod q \]

- Find \( \mathbf{s} \)
- \( \| \mathbf{e} \| \) is small
Decision LWE

Valid LWE Distribution: \( A + s + e = b \)

Uniformly Random: \( A = b \)
Solve SIS to Solve LWE

\[ v \equiv 0 \mod q \]
Solve SIS to Solve LWE
Solve SIS to Solve LWE
Compute $\mathbf{v} \cdot \mathbf{b} \mod q$. If $\mathbf{b} = \mathbf{A}s + \mathbf{e}$, then $\mathbf{v} \cdot \mathbf{b} = \mathbf{v} \cdot \mathbf{e}$ is small. If $\mathbf{b}$ is uniform, then $\mathbf{v} \cdot \mathbf{b} \mod q$ is uniform.
BACK TO SIGNATURES...
Improving the Rejection Sampling

Pick a random $y$
Compute $c=H(Ay \mod q, \mu)$
$z = Sc + y$
Output $(z, c)$ w.p. $D_{\sigma,0}(z) / (MD_{\sigma,Sc}(z))$
Bimodal Gaussians [DDLL ‘13]

Pick a random \( y \)
Compute \( c = H(Ay \mod q, \mu) \)
Pick a random \( b \) in \(-1,1\)
\( z = bSc + y \)
Output \((z, c)\) w.p. \( D_{\sigma,0}(z) / M(\frac{1}{2}D_{\sigma,Sc}(z) + \frac{1}{2}D_{\sigma,-Sc}(z)) \approx \frac{e}{M} \)

Verify \((z, c)\)
Check that \( z \) is “small” and
\( c = H(Az - Tc \mod q, \mu) \)

\( Az - Tc = A(bSc + y) - Tc = bTc - Tc + Ay \)

Want: \( Tc = -Tc \)
Optimizations

• Base problem on the hardness of the NTRU problem
• Compress the signature $\rightarrow$ not all of $z$ needs to be output if $H$ only acts on the high order bits
• A few other small tricks
# Performance of the Bimodal Lattice Signature Scheme

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Security</th>
<th>Signature Size</th>
<th>SK Size</th>
<th>PK Size</th>
<th>Sign (ms)</th>
<th>Sign/s</th>
<th>Verify (ms)</th>
<th>Verify/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLISS-0</td>
<td>≤ 60 bits</td>
<td>3.3 kb</td>
<td>1.5 kb</td>
<td>3.3 kb</td>
<td>0.241</td>
<td>4k</td>
<td>0.017</td>
<td>59k</td>
</tr>
<tr>
<td>BLISS-I</td>
<td>128 bits</td>
<td>5.6 kb</td>
<td>2 kb</td>
<td>7 kb</td>
<td>0.124</td>
<td>8k</td>
<td>0.030</td>
<td>33k</td>
</tr>
<tr>
<td>BLISS-II</td>
<td>128 bits</td>
<td>5 kb</td>
<td>2 kb</td>
<td>7 kb</td>
<td>0.480</td>
<td>2k</td>
<td>0.030</td>
<td>33k</td>
</tr>
<tr>
<td>BLISS-III</td>
<td>160 bits</td>
<td>6 kb</td>
<td>3 kb</td>
<td>7 kb</td>
<td>0.203</td>
<td>5k</td>
<td>0.031</td>
<td>32k</td>
</tr>
<tr>
<td>BLISS-IV</td>
<td>192 bits</td>
<td>6.5 kb</td>
<td>3 kb</td>
<td>7 kb</td>
<td>0.375</td>
<td>2.5k</td>
<td>0.032</td>
<td>31k</td>
</tr>
<tr>
<td>RSA 1024</td>
<td>72-80 bits</td>
<td>1 kb</td>
<td>1 kb</td>
<td>1 kb</td>
<td>0.167</td>
<td>6k</td>
<td>0.004</td>
<td>91k</td>
</tr>
<tr>
<td>RSA 2048</td>
<td>103-112 bits</td>
<td>2 kb</td>
<td>2 kb</td>
<td>2 kb</td>
<td>1.180</td>
<td>0.8k</td>
<td>0.038</td>
<td>27k</td>
</tr>
<tr>
<td>RSA 4096</td>
<td>≥ 128 bits</td>
<td>4 kb</td>
<td>4 kb</td>
<td>4 kb</td>
<td>8.660</td>
<td>0.1k</td>
<td>0.138</td>
<td>7.5k</td>
</tr>
<tr>
<td>ECDSA (^1) 160</td>
<td>80 bits</td>
<td>0.32 kb</td>
<td>0.16 kb</td>
<td>0.16 kb</td>
<td>0.058</td>
<td>17k</td>
<td>0.205</td>
<td>5k</td>
</tr>
<tr>
<td>ECDSA 256</td>
<td>128 bits</td>
<td>0.5 kb</td>
<td>0.25 kb</td>
<td>0.25 kb</td>
<td>0.106</td>
<td>9.5k</td>
<td>0.384</td>
<td>2.5k</td>
</tr>
<tr>
<td>ECDSA 384</td>
<td>192 bits</td>
<td>0.75 kb</td>
<td>0.37 kb</td>
<td>0.37 kb</td>
<td>0.195</td>
<td>5k</td>
<td>0.853</td>
<td>1k</td>
</tr>
</tbody>
</table>
HASH-AND-SIGN SIGNATURE SCHEMES
Constructing the Trapdoor

\[ A' \quad A' \quad R \quad + \quad G \]

- Random matrix
- Random matrix with small coefficients
- Special matrix that is easy to invert
Easily-Invertible Matrix

Want: Matrix $\mathbf{G}$ such that:

For any $\mathbf{b}$ in $\mathbb{Z}_q^n$, you can find a 0/1 vector $\mathbf{s}$ such that $\mathbf{Gs} = \mathbf{b} \mod q$

$$
\mathbf{G} = \begin{bmatrix}
1 & 2 & 4 & 8 & \ldots & \log_2 q \\
1 & 2 & 4 & 8 & \ldots & \log_2 q \\
1 & 2 & 4 & 8 & \ldots & \log_2 q \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & 2 & 4 & 8 & \ldots & \log_2 q
\end{bmatrix}
$$
Inverting with a Trapdoor

\[ A = [A' \mid A'R+G] \]

Want to find a small \( s \) such that \( As = b \)

\[ s = (s_1,s_2) \]

\[ b = As = A's_1 + (A'R+G)s_2 \]

\[ = A'(s_1 + Rs_2) + Gs_2 \]

set to 0

\[ b = Gs_2 \quad s_1 = -Rs_2 \]

Revealing \( R \)!

(probably bad...)
Inverting with a Trapdoor

$A = [A' \mid A'R+G]$

Want to find a small $s$ such that $As = b$

$s = (s_1, s_2)$

$b = As = A's_1 + (A'R+G)s_2$

$= A'(s_1 + Rs_2) + Gs_2$

Maybe $y$ hides $R$?

$b - A'y = Gs_2$

$s_1 = y - Rs_2$
Signature Scheme

Secret (Signing) Key: $R$

Public (Verification) Key: $A = [A' \mid A'R+G]$

Random Oracle $H: \{0,1\}^* \rightarrow \mathbb{Z}_q^n$

Sign($m$):

Find short $s$ such that $As = H(m,u)$

Verify ($s,u,m$)

Check that $s$ is short, and $As=H(m,u)$
Security Proof Sketch

\[ A \text{ pick from D} = H(m_i, u_i) \]

sign \( m_i \)
Security Proof Sketch

A pick from D

= H(m_i,u_j)

program the random oracle

give me H(m_i,u_j)
Security Proof Sketch

To forge on m, the Adversary needs $H(m,u)$

So m is one of the $m_j$ he asked for $H(m_i,u_j)$

Thus we know an $s_j$ such that $A s_j = H(m_i,u_j)$
Security Proof Sketch

\[
A - \begin{bmatrix}
\text{short and hopefully non-zero}
\end{bmatrix} = 0
\]

if it’s non-zero, then we have a solution to SIS
Properties Needed

1. Can sample the distribution $D$ of $s$ without knowing the trapdoor.

2. The following produce the same distribution of $(s,b)$

   (a) Choose $s \sim D$. Set $b = As$

   (b) Choose random $b$. Use the trapdoor to find an $s$ such that $As = b$.

3. For a random $b$, there is more than one likely possible output $s$ such that $b = As$. 
Inverting with a Trapdoor

\[ A = [A' \ | \ A'R+G] \]

Want to find a small \( s \) such that \( As = b \)

\[ s = (s_1, s_2) \]

\[ H(m, u) = As = A's_1 + (A'R + G)s_2 \]

\[ = A'(s_1 + Rs_2) + Gs_2 \]

Maybe \( y \) hides \( R \)?

\[ H(m, u) - A'y = Gs_2 \]

\[ s_1 = y - Rs_2 \]
Rejection Sampling

\[ H(m, u) - A'y = Gs_2 \]

\[ s_1 = y - Rs_2 \]

Choose \( y \) to be a Gaussian.

If \( y \) has a lot of entropy, the distribution of \( s_2 \) is uniform and does not depend on the exact value of \( y \).

\( s_1 = y - Rs_2 \) is now a shifted Gaussian. Use rejection sampling as before (this requires a proof).
Another Approach for Sampling

Suppose we have a matrix $A$ and a trapdoor $R$ such that $AR=G$.

Here is another way to generate an $s$ such that $As=b$

Sample some vector $p$
Sample a $z$ such that $Gz = b - Ap$
Output $s = p + Rz$ (so $As=Ap+Gz=b$)
Correcting the Distribution

Sample some vector $p$
Sample a $z$ such that $Gz = b - Ap$
Output $s = p + Rz$ (so $As = Ap + Gz = b$)

How to make the distribution of $s$ independent of $R$?
Tailor the distribution of $p$ to $R$
1. Continuous Gaussian $\leftrightarrow$ positive definite covariance matrix $\Sigma$. 
   (pos def means: $u^T \Sigma u > 0$ for all unit $u$.)
   Spherical Gaussian $\leftrightarrow$ covariance $s^2 I$.

2. Convolution of Gaussians:

$$
\Sigma_1 + \Sigma_2 = \Sigma = s^2 I
$$

$$
Rz + p = s
$$

$$
RR^t + (s^2 I - RR^t) = s^2 I
$$
IDENTITY-BASED ENCRYPTION
“Dual” Cryptosystem

\[ A = t \]

Public Key

Secret Key (short)

\[ r + A \]

\[ 0 + m \]

\[ = u + v \]
“Dual” Cryptosystem

A - s = t

r + A + s = t

v - u = m

represent 0 by m=0
represent 1 by m=(q-1)/2
“Dual” Cryptosystem Security

A = t

Random

Pseudorandom

r

A + 0

m

= u + v

v
Identity-Based Encryption

Key Authority
Master Public Key
Master Secret Key

Secret Key = s
Public Key = Bob
Identity-Based Encryption

- Key Authority
  - Master Public Key
  - Master Secret Key

- Secret Key = $s_{\text{Bob}}$
  - Public Key = Bob

- Secret Key = $s_{\text{Chris}}$
  - Public Key = Chris

- Secret Key = $s_{\text{Dave}}$
  - Public Key = Dave

Encrypt(Chris, msg)
Security for IBE

Key Authority
  Master Public Key
  Master Secret Key

Secret Key = $s_{Bob}$
Public Key = Bob

Secret Key = $s_{Chris}$
Public Key = Chris

Secret Key = $s_{Dave}$
Public Key = Dave

Encrypt(Chris, msg)

CPA-Security: For all $m_i$ Encrypt(Chris, $m_i$) are **computationally indistinguishable** from each other
IBE Based on LWE

Master Public Key: \( A \)
Master Secret Key: \( R \)

Identity = “Bob”
\( b = H(Bob) \)

Use the sampling algorithm to find a short \( s \) such that \( As = b \) mod \( q \)

Use “Dual” LWE encryption to Encrypt to Bob
Security Proof Sketch

Show that breaking IBE implies breaking the “Dual” cryptosystem.

- **Public key**: 
  - \(A\)
  - \(t\)
- **Ciphertext**: 
  - \(u\)
  - \(v\)
- **Master public key**: 
  - \(A\)
  - \(=\) \(= H(Bob)\)

Bob picks \(A\) from \(D\)

Program the random oracle.
Show that breaking IBE implies breaking the “Dual” cryptosystem

A

A

Bob

H(Bob)

program the random oracle
Security Proof Sketch

Show that breaking IBE implies breaking the “Dual” cryptosystem

I will break an encryption to Dave

\[ \text{master public key} = \text{H(Dave)} \]

Decryption
SIGNATURES WITHOUT RANDOM ORACLES
Signatures without Random Oracles

Public Key: \([ A \mid AR ], b\)

Messages will be square invertible matrices \(M\)

To sign \(M\), find a short \(s\) such that \([ A \mid AR+MG ]s = b\)

\[ A(s_1+Rs_2)+MGs_2 = b \]
\[ Gs_2 = M^{-1}(b - A(s_1+Rs_2)) \]
\[ s_1 = s_1+Rs_2 - Rs_2 \]
Proof of “Selective” Security

In a selectively secure scheme, the adversary declares the message he will forge on before seeing the public key.
Proof of “Selective” Security

Given \( A, b \), want to find a short \( r \) such that \( Ar = b \).

Adversary gives message \( M' \) on which he will forge

We want to construct a public key \([A | B]\) so that a signature of \( M' \) will be an \( s \) that \([A | B + M'G]s = b \) will let us find the short \( r \).

Pick a trap-door \( R \) and let \( B = AR - M'G \)

Then if the adversary can forge, we will get \([A | AR]s = b\)

So \( A(s_1 + Rs_2) = b \), and so \( s_1 + Rs_2 \) is the \( r \) we need.
Proof of “Selective” Security

Adversary designed to work on public key

\[ A | AR \]

We give it public key \[ A | AR - M’G \]

Will it still succeed in forging?

Yes, if \[ A | AR \] is uniformly random, then

\[ A | AR \] has the same distribution as \[ A | AR - M’G \]

Adversary may ask us to sign other messages \( M \).

How do we do it?
Proof of “Selective” Security

To sign $\mathbf{M}$, we need to find an $s$ such that
\[
[A \mid AR - M'G + MG]s = b
\]
\[
A(s_1 + Rs_2) + (M-M')Gs_2 = b
\]
\[
Gs_2 = (M-M')^{-1}(b - A(s_1 + Rs_2))
\]
\[
s_1 = s_1 + Rs_2 - Rs_2
\]

Need $\mathbf{M}-\mathbf{M}'$ to always be invertible.

Such a set of size $q^n$ is easy to construct.

Hint: consider a field $F$ of order $q^n$. Every difference of polynomials in the field is invertible.

How do you map this to matrices?