Lattices and Homomorphic Encryption, Spring 2013	Instructors: Shai Halevi, Tal Malkin
SIS-based Signatures	
February 26, 2013	Scribe: Fernando Krell

Basics

We will use the following parameters:

- n, the security parameter.
- q=poly(n).
- $m \approx 2n \log q$
- $s \geq 2\sqrt{n \log q}$, the Gaussian parameter.

For a matrix $A \in \mathbb{Z}_q^{n\times m}$ and vector $\vec{u} \in \mathbb{Z}_q^n$ denote

$$\mathcal{L}^{\perp}(A) = \{\vec{x} \in \mathbb{Z}_q^m |\ A\vec{x} = 0 \text{ mod q}\}$$
$$\mathcal{L}^{\perp}_{\vec{u}}(A) = \{\vec{x} \in \mathbb{Z}_q^m |\ A\vec{x} = \vec{u} \text{ mod q}\}$$

Micciancio and Peikert 2012 [MP12] describe the following useful procedures $(A, t) \leftarrow \texttt{TrapSamp}(r, m, q, s)$ such that:

- A is nearly uniform in $\mathbb{Z}_q^{n\times m}$.
- Given A and t, it is easy to solve SIS in $\mathcal{L}^{\perp}(A)$. Even more, we have a procedure $\vec{y} \leftarrow \texttt{PreImageSamp}(A, t, s, \vec{u})$ such that $\vec{y}$ is close to discrete Gaussian in lattice $\mathcal{L}^{\perp}_{\vec{u}}$ with small parameter $s \approx m$. Namely $\vec{y} \sim \mathcal{D}_{\mathcal{L}^{\perp}_{\vec{u}}(A),s}$.

Note that $\mathcal{D}_{\mathcal{L}^{\perp}_{\vec{u}}(A),s} = \mathcal{D}_{\mathbb{Z}_q^n,s}|Ax = \vec{u} \bmod q$. That is, sampling from discrete Gaussian over the lattice $\mathcal{L}^{\perp}_{\vec{u}}(A)$, is the same as sampling from Gaussian in $\mathbb{Z}_q^n$ conditioned on that the sampled points $\vec{x}$ satisfy $Ax = \vec{u} \bmod q$.

Definition 1 (Signature Scheme)**.** A signature scheme *is a tuple of three polynomial time algorithms* `KeyGen`*,* `Sign`*, and* `Verify` *such that*

- $(pk, sk) \leftarrow \texttt{KeyGen}(1^n)$
- $\sigma \leftarrow \texttt{Sign}(m, sk)$.
- $\texttt{Accept/Reject} \leftarrow \texttt{Verify}(\sigma, m, pk)$. *For all* $(pk, sk) \leftarrow \texttt{KeyGen}(1^n)$*, and for every message* m *and for every possible* $\sigma \leftarrow \texttt{Sign}(m, sk)$*, it holds that*

$$\Pr[\texttt{Verify}(\sigma, m, pk)] = 1$$

Definition 2 ([GMR86])**.** *A signature scheme* $S = (\texttt{KeyGen}, \texttt{Sign}, \texttt{Verify})$ *is* strong existentially unforgeable under adaptive chosen message attack *is for every feasible attacker* F *that is given a public key* pk*, corresponding to secret key* sk*, and oracle access to* $\texttt{Sign}(\cdot, sk)$*, and outputs pair* (m^*, σ^*)

$$\Pr[\texttt{Verify}(\sigma^*, m^*, pk)|\ (m^*, \sigma^*) \neq (m_i, \sigma_i)\ \forall\ i] = negl(n)$$

Where m_i *denotes the* i*-th message queried to the oracle* $\texttt{Sign}(\cdot, sk)$*, and* σ_i *its answer.*

The GPV signature scheme presented next is secure in the sense of the above definition, in the Random Oracle Model (ROM), under the assumed hardness of SIS problem.

The Random Oracle Model. Schemes in this model have access to a function $H : \Sigma^* \to \mathbb{Z}_q^n$. In the security analysis, we pretend that this function is a truly random function, that is, H assigns to every message a uniformly random vector $\vec{y} = H(m) \in \mathbb{Z}_q^n$.

The GPV Signature Scheme [GPV08]

Let $H : \Sigma^* \to \mathbb{Z}_q^n$ be a hash function. The GPV signature scheme using H consists of the following algorithms:

- $\texttt{KeyGen}(1^n)$: Run $\texttt{TrapSamp}(n, m, q, s)$ to get pair (A, t). Output $(pk = A, sk = (A, t))$.
- $\texttt{Sign}(m, sk = (A, t))$: Compute $\vec{y} = H(m)$, and output short vector $\vec{u} \leftarrow \texttt{PreImageSamp}(A, t, s, \vec{y}$
- $\texttt{Verify}(\vec{u}, m, pk = A)$: Compute $\vec{y} = H(m)$. Output $\texttt{Accept}$ if and only if $A\vec{u} = \vec{y}$ and $||\vec{u}|| < 6n \log q$.

Remark 1. We need to make sure that we always output the same signature for the same message. Otherwise, the scheme can be broken. We can do this by keeping a table of all signatures computed so far, or by using a pseudorandom function, computing the "random" coins for the $\texttt{PreImageSamp}$ procedure as $\mathrm{PRF}(m)$.

Correctness

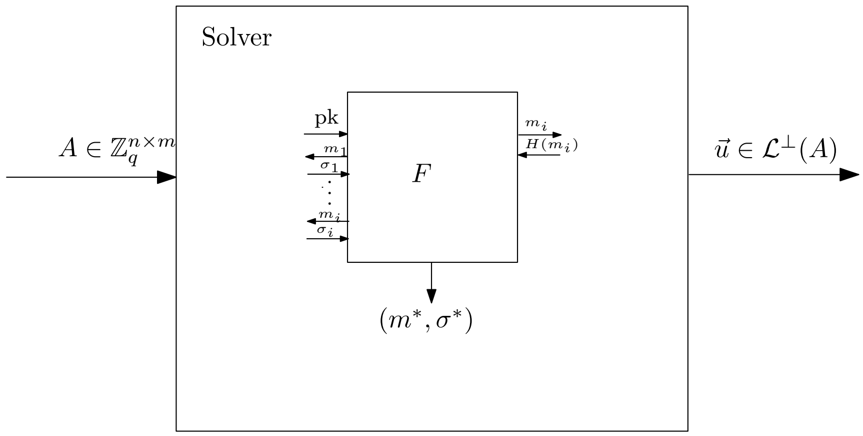
Since the vector $\vec{u}$ was computed using $\texttt{PreImageSamp}$ algorithm, it is distributed close to $D_{\mathcal{L}^\perp_{\vec{y}}, s}$, thus $A\vec{u} = \vec{y}$. Moreover, using $s = 2\sqrt{n \log q} > \eta_{2^{-n}}(\mathcal{L}^\perp(A))$ with high probability[1], so it's expected size is $\leq 2s\sqrt{m} = 4\sqrt{2}n \log q < 6n \log q$, and $\Pr[||\vec{u}|| > 6n \log q] < 2^{-n}$.

Remark 2. Recall from lecture 5 that the smoothing parameter η_α of a lattice $\mathcal{L}$ is the smallest Gaussian parameter s such that $\rho_{1/s}(\mathcal{L}^* \setminus \{0\}) \leq \alpha$. Where $\rho_s(\vec{x}) = e^{-\pi(||x||/||s||)^2}$ is the Gaussian probability density function with parameter s (centered at $\vec{0}$). If α is not too large then $\eta_\alpha \geq \frac{1}{\lambda_1(\mathcal{L}^*)} \geq \lambda_n(\mathcal{L})$. Also, a vector $\vec{x}$ sampled from this distribution has length $\leq s\sqrt{k}$ with high probability, where k is the dimension of the lattice.

[1] We prove this later

Security

We prove security by showing that if a forger F, running in time T, has success probability ε relative to a random function H, then there is a solver S that uses F to solve the SIS problem with probability $\sim \varepsilon$ in time $\sim T$.



The solver S: gets as input matrix A. To run the forger F, S need to provide a public key, oracle response to hash function and oracle response to signatures queries.

- Set A as the public key.
- For each random oracle query $H(m_i)$, sample $x_i \leftarrow D_{\mathbb{Z}^n,s}$. set $\vec{y}_i = A\vec{x}_i \bmod q$, reply with $H(m_i) = y_i$. Record tuple $(m_i, \vec{x}_i, \vec{y}_i)$ for future queries.
- For each signature query $\mathrm{Sign}(m_i)$. get $H(m_i)$ executing the procedure above. Finds tuple $(m_i, \vec{x}_i, \vec{y}_i)$ and outputs $\vec{x}_i$ as the signature.

At the end of the interaction F outputs a forgery $(m^*, \vec{u}^*)$. S now execute one more random oracle query on m^* to get $\vec{x}^*$ and returns $\vec{x} - \vec{u}$ as the SIS solution.

To prove that S solves SIS with probability $\sim \varepsilon$, we need to show two things:

1. The answers that F gets from S are distributed close to the same distribution as when F interacts with the scheme.

   Proof. In the scheme $H(m) = \vec{y}_m \in_R \mathbb{Z}_q^n \Rightarrow \vec{x} \leftarrow \mathcal{D}_{\mathcal{L}_{\vec{y}}^{\perp},s}$. In contrast, the solver chooses first $\vec{x}$ from $\mathcal{D}_{\mathbb{Z}^m,s}$ and compute $\vec{y}$ as $Ax \bmod q$. Conditioned in $\vec{y}$, we can see this as sampling $\vec{x}$ from $\mathcal{D}_{\mathcal{L}_{\vec{y}}^{\perp},s}$

   If $s > \eta_{2^{-n}}(\mathcal{L}^{\perp}(A))$, then $\vec{x}$ reduced modulo basic cell of $\mathcal{L}^{\perp}(A)$ is nearly uniform. In problem set 4, problem 2, we prove that this implies that $A\vec{x} = \vec{y}$ is nearly uniform in $\mathbb{Z}_q^n$. Thus, the distribution of $\vec{y}$ is teh same as in the scheme.

   At the same time, the distribution of $\vec{x}$ conditioned on $\vec{y}$ are also the same in the scheme and in the solver since $\mathcal{D}_{\mathcal{L}_{\vec{y}}^{\perp}(A),s}$ is the same as $\mathcal{D}_{\mathbb{Z}^m,s}$ conditioned on the outcome satisfying $A\vec{x} = \vec{y}$. □

2. If (m^*, u^*) is a valid forgery, then S outputs a solution to SIS (with high probability).

Proof. By the proof above F outputs a valid forgery with probability $\sim \varepsilon$ when interacting with S. This implies that for $y^* = H(m^*)$, it holds that $\vec{y} = A\vec{u}^* = Ax^*$, hence $A(\vec{x}^* - \vec{u}^*) = 0 \mod q$, and thus $(\vec{x}^* - \vec{u}^*) \in \mathcal{L}^\perp(A)$.

Also, $||\vec{u}^*|| < 6n \log q$ because $\vec{u}^*$ is a valid forgery. Now, $||\vec{x}^*|| < 6n \log q$ because $\vec{x}^*$ was sampled from a Gaussian distribution with parameter s. Therefore, $||\vec{x}^* - \vec{y}^*|| < 12n \log q$.

We need to prove that $\vec{x}^* \neq \vec{u}^*$. Two cases to analyze:

- If F asked for a signature of m^*, then it received $\vec{x}^*$, thus $\vec{u}^* \neq \vec{x}^*$, since $\vec{u}^*$ is a valid "new" forgery.
- If F did not asked for a signature on m^* then F can only know about $\vec{x}^*$ what's implied by $\vec{y}^*$. So from F point of view, the min-entropy of $\vec{x}^*$ is $H_\infty(\mathcal{D}_{\mathcal{L}^\perp_{\vec{y}}(A)})$. If $s > \eta_{2^{-n}}(\mathcal{L}^\perp_{\vec{y}}(A))$, then $\vec{x}$ has min entropy $\geq n$ bits. Hence, $\Pr[\vec{x}^* = \vec{u}^*] < 2^{-n}$.

□

We end the security proof by showing that s is lager than the smoothness parameter with parameter $\alpha = 2^{-n}$.

Claim 1. $s > \eta_{2^{-n}}(\mathcal{L}^\perp(A))$.

Proof. Denote $\mathcal{L}(A) = \{\vec{u} \in \mathbb{Z}^n | \ \exists \vec{v} \in \mathbb{Z}^n \text{ such that } \vec{u} = \vec{v}A\}$. Observe that this lattice is almost dual of $\mathcal{L}^\perp(A)$. In fact, $(\mathcal{L}^\perp(A))^* = \mathcal{L}(A)/q$. We show that with high probability over $A \in_R \mathbb{Z}_q^{n\times m}$, $\lambda_1^\infty(\mathcal{L}(A)) > \frac{q}{4}$. Where λ_1^∞ denotes the successive minima in infinity norm.

Fix any short non-zero vector $\vec{u} \in \mathbb{Z}^n$ such that $||\vec{u}||_\infty < \frac{q}{4}$. What is the probability that $\vec{u} \in \mathcal{L}(A)$?

$$\begin{aligned} \Pr_A[\vec{u} \in \mathcal{L}(A)] &= \Pr_A[\exists \vec{v} \in \mathbb{Z}_q^n \text{ such that } \vec{v}A = \vec{u} \mod q] \\ &\leq \sum_{\vec{v} \in \mathbb{Z}_q^n \setminus \{\vec{0}\}} \Pr_A[\vec{v}A = \vec{u} \mod q] \\ &\leq q^{-m} q^n \end{aligned}$$

There are $\leq \left(\frac{q}{2}\right)^m$ possible vectors $\vec{u} \neq 0$ with $||\vec{u}||_\infty \leq \frac{q}{4}$ (coordinates between -q/4 and q/4). Hence

$$\begin{aligned} \Pr[\exists \vec{u} \neq 0 \wedge ||\vec{u}||_\infty \leq \frac{q}{4} \wedge \vec{u} \in \mathcal{L}(A)] &\leq \left(\frac{q}{2}\right)^m q^{n-m} \\ &\leq \frac{q^n}{2^m} \\ &\leq 2^{-n} \end{aligned}$$

Where last inequality holds since $m \geq 2n \log q$. This implies that $\lambda_1^\infty(\mathcal{L}^\perp(A)^*) = \frac{\lambda_1(\mathcal{L}(A))}{q} > \frac{1}{4}$. Thus,

$$
\begin{aligned}
\eta_{2^{-n}}(\mathcal{L}^{\perp}(A)) &\leq \frac{1}{\lambda_1^{\infty}((\mathcal{L}^{\perp}(A))^*)}\sqrt{\frac{\log(2n(1+2^n))}{\pi}} \\
&\leq 4\sqrt{\frac{\log n + n + 2}{\pi}} \\
&\leq 4\sqrt{\log n + n} < s
\end{aligned}
$$

□

[GMR86] S. Goldwasser, S. Micali, R. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. In *SIAM Journal of Computing, 1988.*

[GPV08] C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC, 2008.*

[MP12] D. Micciancio, C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *Advances in Cryptology - EUROCRYPT 2012.*