Sampling Lattice Trapdoors

November 10, 2015

Today:

- 2 notions of lattice trapdoors
- Efficient sampling of trapdoors
- Application to digital signatures

Last class we saw one type of lattice trapdoor for a matrix A and that it was sufficient for solving LWE and ISIS with matrix A. The difficulty is in sampling uniform A along with a trapdoor. Today we will look at a particular matrix for which we can easily describe a trapdoor. With this matrix in hand, it will suffice to sample a different type of trapdoor – a task that will be simpler. Finally, we will demonstrate a simple digital signature scheme based on the above.

1 2 types of trapdoors

1.1 Type 1

This is the notion of a trapdoor that we saw last class.

Definition 1.1 ($L^{\perp}(A)$)**.** *For a matrix $A \in \mathbb{Z}_q^n \times m$, we denote by $L^{\perp}$ the* dual lattice of A *composed of all vectors in the kernal of A (arithmetic done $\mod q$):*

$$L^{\perp}(A) \triangleq \{x \in \mathbb{Z}^m \ : \ Ax = 0 \mod q\}$$

A trapdoor T for A is a short basis for the lattice $L^{\perp}(A)$.

Definition 1.2 ('Type 1' Trapdoor)**.** *For matrices $A \in \mathbb{Z}_q^n \times m$ and $T \in \mathbb{Z}_q^{m \times m}$, T is a* trapdoor for A *if:*

1. $AT \equiv 0^{n \times m} \mod q$
2. *T is full rank* over $\mathbb{Z}$.
3. *Each column t_i of* $T = \begin{bmatrix} t_1 \ t_2 \ \cdots t_m \end{bmatrix}$ *is 'short'.*

Last class, we saw that, given such a trapdoor T for A, one could efficiently solve LWE and ISIS.

1.2 The Gadget Matrix

A special matrix that will be important for us later is the "gadget matrix" G, whose trapdoor is very easily understood.

Definition 1.3 (Gadget Matrix G)**.** *Let* $g = [1\ 2\ 4\ \cdots\ 2^{\lceil \log q \rceil - 1}]$. *The* gadget matrix $G \in \mathbb{Z}_q^{n \times n \log q}$ *is* $G \triangleq g \otimes I_n$*:*

$$G = \begin{bmatrix} g & 0 & \cdots & 0 \\ 0 & g & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g \end{bmatrix}$$

The vector g can be thought of a binary recomposition[1] operator, taking the binary representation (as a column vector) $\mathsf{binary}(x) \in \mathbb{Z}_q^{n \log q}$ of an integer $x \in \mathbb{Z}_q$, and mapping it to $g^T \cdot \mathsf{binary}(x) = x$. Likewise, for integers $x_1, \ldots, x_n \in \mathbb{Z}_q$, G is the operator mapping $b \in \mathbb{Z}_q^{n^2 \log q}$ to Gb:

$$b = \begin{bmatrix} \mathsf{binary}(x_1) \\ \vdots \\ \mathsf{binary}(x_n) \end{bmatrix} \mapsto \quad Gb = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

One possible trapdoor T_g for g is:

$$T_g \triangleq \left[\begin{array}{cccccc} 2 & 0 & 0 & \cdots & 0 & \\ -1 & 2 & 0 & \cdots & 0 & \\ 0 & -1 & 2 & \cdots & 0 & \mathsf{binary}(q) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 2 & \\ 0 & 0 & 0 & \cdots & -1 & \end{array}\right]$$

where $\mathsf{binary}(q) \in \mathbb{Z}_q^{\lceil \log q \rceil - 1}$ is the binary expansion of q. It is easy to verify that $gT_g = 0$ and that each column has short length (all are either $\sqrt{5}$ or $O(\sqrt{\log q})$). Let $k = \lceil \log q \rceil - 1$ and $b = \mathsf{binary}(q)$. Then

$$\det(T_g) = \sum_{i=1}^{k} (-1)^{i-1} \cdot b[i] \cdot 2^{i-1} (-1)^{k-i} = (-1)^{k-1} \sum_{i=1}^{k} b[i] \cdot 2^{i-1} = (-1)^{k-1} q \neq 0$$

and therefore T_g is full rank over $\mathbb{Z}$ (though not full-rank over $\mathbb{Z}_q$).

Finally, we define the gadget matrix T_G fo the matrix G:

$$T_G \triangleq T_g \oplus I_n = \begin{bmatrix} T_g & 0 & \cdots & 0 \\ 0 & T_g & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_g \end{bmatrix}$$

Just as with T_g, T_G is full-rank over $\mathbb{Z}$, has short entries, and is in the null space of G.

Last class, we saw that a (type 1) trapdoor for A suffices to solve LWE and ISIS. Because of the structure of G, solving these problems with respect to G is particularly efficient.

[1] The inverse of binary decomposition

1.3 Type 2

Our goal is to sample a uniform matrix A along with some auxiliary information that makes solving LWE (recovering s^T given $s^T A + e$) with matrix A tractable. The trapdoor T_G for G enabling us to solve LWE instances with matrix G efficiently. Therefore it suffices to have auxiliary information about A that allows us to transform LWE samples for A to LWE samples for G with the same secret. This information will be a 'type 2' trapdoor for A.

Definition 1.4 ('Type 2' Trapdoor)**.** *For matrices $A \in \mathbb{Z}_q^n \times m$ and $R \in \mathbb{Z}_q^{m \times n \log q}$, R is a* trapdoor for A *if:*

1. $AR = G$
2. *Each column r_i of $R = \begin{bmatrix} r_1 \; r_2 \; \cdots r_m \end{bmatrix}$ is 'short'.*

Remark. We could similarly define a type 2 trapdoor with respect to any full rank matrix G for which we knew a trapdoor, and the reductions below would suffice. The reason for preferring this particular gadget matrix G is that its structure makes solving LWE and ISIS conceptually simple and concretely efficient.

Solving LWE and ISIS.
Given an LWE sample $s^T A + e^T$ and a (type 2) trapdoor R for A:

$$(s^T A + e^T)R = s^T G + (e^T R)$$

This is an LWE sample for G with error $e' = e^T R$.

Similarly, given an ISIS challenge (A, v), we can find a short e such that $Ae = v$ by solving the corresponding problem with respect to G, and multiplying the solution by R

$$Ae = A(Re') = Ge' = v$$

Since G is the binary recomposition matrix, a solution to the above is $e' = [\mathsf{binary}(v_1) \| \cdots \| \mathsf{binary}(v_n)]^T$. This e' is composed only of 0s and 1s and thus $\|e'\| < \sqrt{n}$.

2 Sampling Trapdoors

Though we've seen that sampling matrices A and associated type 2 trapdoors R suffices for solving lattice problems, how do we sample such matrices?

1. Sample uniformly $A_0 \in \mathbb{Z}_q^{n \times m_0}$ [2]
2. Sample random $R_0 \in \mathbb{Z}_q^{m_0 \times n \log(q)}$ with each entry $R_{i,j} \leftarrow \mathsf{Bernoulli}(\frac{1}{2})$ is 0 or 1 with equal probability.

[2]We shall see below that $m_0 = n \log q + 2n$ suffices.

3.

$$A \triangleq \begin{bmatrix} A_0 \parallel & -A_0R_0 + G \end{bmatrix}$$

$$R \triangleq \begin{bmatrix} R_0 \\ I \end{bmatrix}$$

It is easy to verify that $AR = G$, and that the columns r_i of R are short with high probability (in particular,) We now demonstrate that A sampled as above is statistically-close to uniform. To prove this, we use the Leftover Hash Lemma. Each column of R_0 has $m_0 := n \log q + 2n$ bits of entropy. This implies that for each column r_i, $(A_0, -A_0 r_i) \approx_{2^-n} (A_0, u)$ where $u \leftarrow \mathbb{Z}_q^n$ is sampled uniformly. Combining for all the columns, we get that $[A_0 \| - A_0R_0 + G]$ is statistically-close to $[A_0 \| U]$, a completely uniform $n \times m$-matrix.

3 Application to Digital Signatures

Now we'll see an application of lattice trapdoors in building a simple signature scheme. Observe that we can already build a signature from lattices assuming that there is a lattice-based one-way function a la Ajtai [?], using generic constructions of digital signatures from OWFs [?]. In contrast, we will see a relatively simple and efficient scheme based on [?]. Previously, there were a number of flawed proposals .

We consider a weak form of security, where the adversary is required to forge a signature on a specific challenge message msg^* and the adversary receives signatures on random messages rather than messages of his choice.

Definition 3.1 (Digital Signatures)**.** *A digital signature scheme for messages in* $\mathcal{M}$ *is a tuple of algorithms* $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ *satisfying the following properties:*

Syntax:

- $\mathsf{Keygen}(1^n) \to sk, vk$ *is a randomized algorithm taking a security parameter* $\lambda \in \mathbb{N}$ *in unary, and outputting a signing key* sk *and a verification key* vk.
- $\mathsf{Sign}(sk, msg) \to sig$ *is a randomized algorithm taking a signing key* sk *and a message* $msg \in \mathcal{M}$ *as inputs, and outputting a signature* sig.
- $\mathsf{Verify}(vk, sig, msg) \to \{0,1\}$ *is a deterministic algorithm taking a verification key* vk*, a signature* sig*, and a message* msg*, and outputting* $b \in \{0,1\}$.

Correctness: *A digital signature is correct if validly generated signatures verify. Namely, for all* $\lambda \in \mathbb{N}$*, for all* $sk, vk \leftarrow \mathsf{Keygen}(1^n)$*, and for all* $msg \in \mathcal{M}$*:*

$$\mathsf{Verify}(vk, \mathsf{Sign}(sk, msg), msg) = 1$$

Security:[3] *A digital signature scheme is* secure *if for large enough* $n \in \mathbb{N}$*, all probabilistic polynomial-time algorithms* $\mathcal{A}$ *have negligible probability of winning in the following game:*

- $sk, vk \leftarrow \mathsf{Keygen}(1^n)$
- $msg^* \leftarrow \mathcal{M}$ `// The challenge message`
- $\mathsf{state}_0 \leftarrow \mathcal{A}(vk, msg^*)$*;*

- *While* $\mathsf{done} \not\leftarrow \mathcal{A}(\mathsf{state}_i, sig_i)$
  - $\mathsf{state}_{i+1} \leftarrow \mathcal{A}(\mathsf{state}_i, sig_i)$
  - $msg_{i+1} \leftarrow \mathcal{M}$
  - $sig_{i+1} \leftarrow \mathsf{Sign}(sk, msg_{i+1})$
- $sig^* \leftarrow \mathcal{A}(\mathsf{state}_{\mathsf{final}})$
- $\mathcal{A}$ *wins if* $\big(\mathsf{Verify}(vk, sig^*, msg^*) = 1\big)$

3.1 A First Attempt

Suppose we have an algorithm $A, R \leftarrow \mathsf{TrapSamp}(n, m, q)$ that samples matrices $A \in \mathbb{Z}_q^{n\times m}$ along with an associated (type 2) trapdoor $R \in \mathbb{Z}_q^{m\times n\log q}$. Let $\mathcal{M} = \mathbb{Z}_q^m$.

- $\mathsf{Keygen}(1^n)$: Sample $A, R \leftarrow \mathsf{TrapSamp}(n, ??, ??)$. Output $sk = R$ and $vk = A$.
- $\mathsf{Sign}(sk, msg)$: Solve $Ae = y$ for $y := msg$, and a short $e \in \mathbb{Z}_q^m$. Output $sig = e$.
- $\mathsf{Verify}(vk, sig, msg)$: Output 1 if $\big(Ae = msg\big) \wedge \big(\|e\| \le \mathrm{poly}(n)\big)$. Otherwise, output 0.

Forging a signature on msg requires finding a short solution to $Ae = b$ for a random value $b = msg$. Given no (msg_i, sig_i) pairs, this is infeasible by the difficulty of ISIS. But what about when given many pairs, all with respect to the same signing key A?

Suppose we use the solver for ISIS from Section 1.3 with the trapdoors from Section 2. Then the adversary receives many signatures of the form $sig = Re' = \begin{bmatrix} R_0 \\ I \end{bmatrix} e' = \begin{bmatrix} R_0 e' \\ e' \end{bmatrix}$, where e' is the binary decomposition of msg . With sufficiently many samples, the adversary could efficiently solve for R_0 the system of equations $(sig_1\|\cdots\|sig_m) = R_0 E$ where $E = (e'_1\|\cdots\|e'_m)$ can be easily constructed by the adversary. Given enough signatures, the adversary can reconstruct the trapdoor R, let alone forge signatures.

3.2 A better "Solve" step

We will fix the above scheme by requiring that "solve" step – hereafter named $\mathsf{Solve}(A, R, y)$ – in the Sign algorithm satisfies some additional property. After defining what a good solver is, we demonstrate that it suffices to prove existential unforgeability under chosen-message attack. Next lecture, we will look at such a good Solve algorithm, based on discrete Guassian sampling.

We will call Solve good if its outputs statistically hide the trapdoor R in the following strong sense.

Definition 3.2 (Good Solve). *Let $\mathsf{Solve}(A, R, y)$ be an algorithm that outputs short solutions to $Ae = y$. Solve is* good *if for some $\sigma > 0$, for every short trapdoor R:*

$$\Big(A \leftarrow \mathbb{Z}_q^{n\times m},\ e \leftarrow D_{\mathbb{Z}^m,\sigma},\ y := Ae\Big) \approx_s \Big(A \leftarrow \mathbb{Z}_q^{n\times m},\ \mathsf{Solve}(A, R, y),\ y \leftarrow \mathbb{Z}_q^m\Big)$$

where $\approx_s$ denotes statistical closeness 2^{-n}.[4] *and $D_{\mathbb{Z}^m,\sigma}$ denotes the discrete Guassian distribution on $\mathbb{Z}^m$ with standard deviation σ.*

[4] Computational, rather than statistical, indistinguishability would suffice.

That is, the output distribution of $e \leftarrow \mathsf{Solve}(A, R, y)$ is statistically close to sampling e from discrete Guassian conditioned on $Ae = y$.

We now reduce solving SIS to violating the security of the signature scheme in Section 3.1 instantiated with a "good" Solve algorithm.

Reducing SIS **to forging:** Given an SIS challenge $A \in \mathbb{Z}_q^{n\times m}$, our goal is to find a short $e \in \mathbb{Z}_q^m$ such that $Ae \equiv 0^n \mod q$. Run the adversary $\mathcal{A}$ for the signature scheme, but generate message-signature pairs by sampling $sig := e \leftarrow D_{\mathbb{Z}^m,\sigma}$ from a discrete Guassian and setting $msg := y = Ae$. Additionally, choose $e^* \leftarrow D_{\mathbb{Z}^m,\sigma}$ and set the challenge message $msg^* = Ae^*$.

By the goodness of Solve, this distribution on messages and signatures is statistically indistinguishable from the real distribution (random messages and signatures generated by $\mathsf{Solve}(A, R, y)$). Any $\mathcal{A}$ that forges when given samples from the latter must also forge when given samples from the former. Thus, with noticeable probability, $\mathcal{A}$ will output a signature $sig^{**} = e^{**}$ such that $Ae^{**} = Ae^*$ and $\|e^{**}\|$ is short.

With high probability $e^{**} \neq e^*$. Therefore, $e^{**} - e^* \neq 0$ is a solution to SIS for matrix A.