Notes written by Daniel Malinowski

LWE and SIS problems

Definition 1 LWE (Learning with errors) Problem:

- Parameters:
  - q: prime number
  - n: dimension of the space
  - m: number of repetitions
  - χ: distribution over $\mathbb{Z}_q$
  - $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
- Problem: let $s \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \leftarrow \chi^n$. Given (A, v) distinguish whether $v = As + e$ or $v \xleftarrow{\$} \mathbb{Z}_q^m$.

LWE is hard for a proper choice of the distribution χ (usually Gaussian): the best attacks known are exponential in n. Note that in some sense it also represents an extension of LPN (Learning Parity with Noise) to a non-binary setting (although LPN uses a different metric). It is also connected to the decoding of linear codes, if we consider the noise term e as an "error vector". Most importantly, however, LWE is directly connected to lattices: there is in fact a *quantum* reduction to lattice problems (namely the approximate SVP problem).

Definition 2 SIS (Short independent solution) Problem:

- Parameters:
  - q: prime number
  - n: dimension of the space
  - m: number of repetitions
  - $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
- Problem: find "small" non-zero s such that $As = 0$.

LWE and SIS are two distinct instances of knapsack problems. The security of the problems is depicted in Figure 1. If the value $||s||$ is really small, then there might be no solution to the SIS problem, while for LWE, the smaller the values of $||s||$, the easier the problem gets. When the value of $||s||$ increases, on the other hand, there are going to be a lot of collisions, and not a unique solution anymore, hence SIS gets progressively easier.

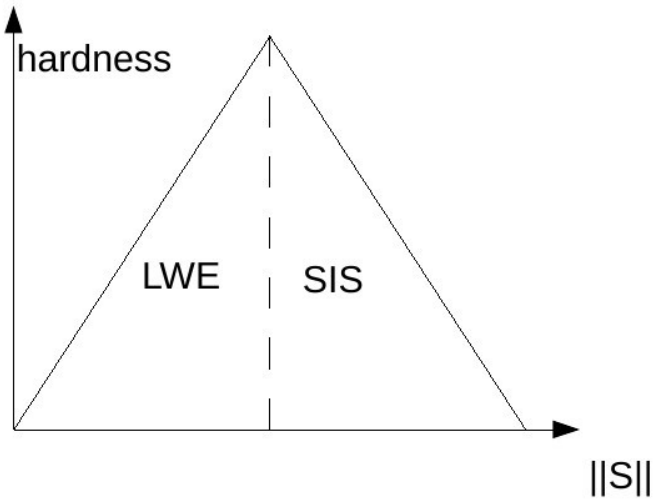

Figure 1: Hardness of LWE and SIS

Cryptographic schemes using LWE/SIS

We will show two signatures schemes based on the LWE/SIS problems. The first one follows the hash-and-sign paradigm, and thus requires a trapdoor one-way function, while the second one makes use of the Fiat-Shamir transform, that turns an identification scheme into a signature.

An intuitive idea would be to use GGH-like algorithms for signatures schemes. However, it has been shown that the security can be easily compromised in this case. The problem is that the signature algorithm leaks information about the private key. In fact, after seeing many message/signature pairs: (point P, lattice point Q closest to P) it becomes easy find the fundamental parallelepiped (i.e. the secret basis), leading to a total break of the scheme.

A first proposal to deal with this issue is due to Gentry, Peikert and Vaikuntanathan.

Definition 3 GPV '08 signature scheme:

- Keygen:
  - μ: message
  - h: hash function
  - q, n, m: integers
  - pk: a matrix $A \in \mathbb{Z}_q^{m \times n}$ defining a lattice
  - sk: a basis T of lattice A consisting of short vectors
- Sign:
  - let $b = h(\mu)$
  - Use pre-image sampling algorithm to find the signature σ s.t. $A\sigma = b$
- Verify:
  - check if $A\sigma \stackrel{?}{=} h(\mu) \pmod q$ and if σ is "small"

The security of the scheme is based on the SIS problem. The main issue is how to calculate the signature without leaking T. The idea of GPV is that, instead of using plain Babai's rounding techniques, we can use a sort of "randomized" version. The chosen point is now determined according to a spherical Gaussian distribution, that depends solely on the length of the vectors in the basis. Thus, with very high probability, this would correspond to the point found via the usual Babai's algorithm, but there is a non-zero chance to obtain another point. This shows that the process only leaks the length of the basis vectors, but not the basis itself.

GPV constitutes an excellent solution to the problems of signature schemes based on lattices; however, the kind of trapdoor needed for the scheme is usually not easy to find and rather unwieldy. The following idea from Lyubashevsky shows how to obtain a signature scheme without needing a trapdoor. The idea builds on previous work from the same author, using an identification scheme together with the Fiat-Shamir transform

Definition 4 Lyubashevsky '12 signature scheme:

- Keygen:[1]
  - μ: message
  - h: hash function
  - q, n, m: integers
  - $sk = S \in \mathbb{Z}_q^{n \times \ell}$ "small"
  - $pk = (A, T = AS)$ for $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$
- Sign:
  - Sample $y \xleftarrow{\$} \mathbb{Z}_q^n$
  - Calculate $Y = Ay$
  - Obtain the challenge $c = h(\mu, Y)$
  - Compute $z = Sc + y$ and send the signature $\sigma = (z, c)$ to the Verifier
- Verify:
  - Check if $h(\mu, Az - Tc) \stackrel{?}{=} c$ and if z is "small"

Proof The security of the scheme is again based on the SIS problem. We thus construct a simulator S and show that it can solve SIS by using the adversary A against the signature scheme as a subroutine, as follows:

1. S gets a SIS instance given by a matrix A. It then samples a random small $S \in \mathbb{Z}_q^{n \times \ell}$ and sends (A, AS) to A.
2. A performs signing queries, submitting messages μ_i. S uses the matrix S as private key and is thus able to sign, and replies with signatures (z_i, c_i).
3. Eventually A will output a forgery (z, c) for a certain message μ. Now, S can rewind the procedure and run the subroutine again with different responses to random oracle queries, and eventually will obtain another valid signature (z', c') for the message μ.
4. It is easy to check that $(Az - Tc) = (Az' - Tc')$, hence $A(z - z' + Sc' - Sc) = 0$ and so if $z - z' + Sc' - Sc \neq 0$ then this is a solution to SIS. $\triangle$

[1] Note that, given the public key, the private key is not unique. This is an important security requirement.

It follows that, in order for the proof to work, the scheme needs the signature to be independent of S, so $z - z' + Sc' - Sc \neq 0$, and $z - z' + Sc' - Sc$ to be small, so that SIS is hard. To satisfy these requirements, we could imagine choosing y uniformly at random, but then z would be too big and forging would become easy. On the other hand, if y is too small, then we lose the independence of z from S (the noise doesn't mask the secret well enough). The ideal proposal sits "in the middle", and consists of using rejection sampling: make y somewhat small, and only output signature if it meets certain requirements on the size. As long as the probability of rejection is small, this works well.

Unfortunately, with this formulation, breaking SIS is hard but forging signatures becomes easier, since the signature is considerably bigger than the private key (see Figure 2). A possible to solution is to base the scheme on LWE rather than SIS, moving from high-density knapsacks to low-density knapsacks. In this way, both tasks are reasonably hard, and we get an overall better security (see again Figure 2).

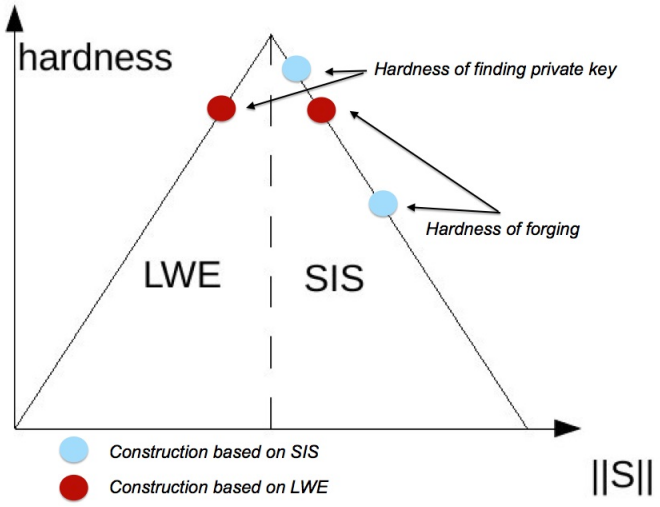

Figure 2: Hardness of signature schemes based on LWE and SIS

To do this, we change the requirement on the private key: it is now only required to be *computationally indistinguishable* to determine whether the private key is unique. However, this implies fixing the above proof, since now S cannot use his matrix S as a private key. Instead, first S plays a hybrid game swapping from his matrix S to a bigger one (this is indistinguishable by construction) and it can then play the previous security reduction game by programming the random oracle so to be able to reply to signature queries. This is possible since the distributions of c and z are known, and thus S can just sample from the joint distribution and output/reject signatures with the correct probability.

With this formulation, and due to the increased level of security, we get much smaller signature size: around 9000 bits.