A Key Exchange Protocol Based on LWE

August 7, 2015

In [Pei14] Peikert for an even modulus q defines two functions:

$$\lfloor \cdot \rceil_2 : \mathbb{Z}_q \to \{0,1\}, \quad \lfloor v \rceil_2 = \left\lfloor \frac{2}{q} v \right\rceil$$
$$\langle \rangle_2 : \mathbb{Z}_q \to \{0,1\}, \quad \langle v \rangle_2 = \left\lfloor \frac{4}{q} v \right\rfloor \mod 2$$

Peikert shows that if v is uniformly random, then $\langle v \rangle_2$ is uniformly random and $\lfloor v \rceil_2$ is uniformly random given $\langle v \rangle_2$.

Given a value w that is close to v and given a binary value $c = \langle v \rangle_2$ Peikert shows how to find $\lfloor v \rceil_2 \leftarrow rec(w, c)$. The procedure is called “reconciliation”. This procedure allows two parties to agree exactly on the value of $\lfloor v \rceil_2$ (which will become the key), getting at first the approximations of v.

1 First Protocol. Parameters Estimation.

The first protocol for key exchange based on LWE uses LWE assumption on the server side to generate a Server’s KeyExchange message and uses the left-over hash lemma on the client side to generate a Client’s KeyExchange message.

$$A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$$

Alice (Server)		Bob (Client)
$S \xleftarrow{\$} \mathbb{Z}_q^{n \times \overline{n}}$		
$E \xleftarrow{\chi} \mathbb{Z}_q^{m \times \overline{n}}$		
$B := AS + E$		
	$\xrightarrow{B \in \mathbb{Z}_q^{m \times \overline{n}}}$	
		$R \xleftarrow{\$} \{-1, 1\}^{\overline{m} \times m}$
		$B' := RA$
		$V := RB$
		$C := \langle V \rangle_2$
	$\xleftarrow{B' \in \mathbb{Z}_q^{\overline{m} \times n}, C}$	
$K := rec(B'S = RAS, C)$		$K := \lfloor V \rceil_2 = \lfloor RAS + RE \rceil_2$

Figure 1: First LWE-based key exchange protocol.

Lower bound on m. The LWE-based key exchange protocol is depicted on Figure 1. The proof of the protocol requires for B' and V to be computationally close to being uniformly random, independent of A and B. To achieve this we apply the leftover hash lemma to argue that if

$$m > (n + \overline{n} + 1)\log_2(q) + \omega(\log n) \tag{1}$$

given a uniformly random matrix $(A||B) \in \mathbb{Z}_q^{m\times(n+\overline{n})}$ the matrix $R(A||B) \in \mathbb{Z}_q^{\overline{m}\times(n+\overline{n})}$ will be statistically close to uniform.

Choosing $\overline{n}$ and $\overline{m}$. We have freedom in choosing the parameters $\overline{m}$ and $\overline{n}$. We set LWE parameters to 128-bits security level, which means it is enough to choose $\overline{m}, \overline{n}$ such that at the end we get a $\lambda = \overline{n}\cdot\overline{m} = 128$ bits key.

The size of matrix $|B| = m\overline{n}\log q$, $|B'| = \overline{m}n\log q$, the size of $|c| = \overline{n}\cdot\overline{m}$ which can be neglected. To optimize total communication we need to find $\overline{n}, \overline{m} = argmin(|B| + |B'|)$, setting $\overline{m} = 128/\overline{n}$, we need $\overline{n} = \underset{\overline{n}}{argmin}(\frac{128n}{\overline{n}} + \overline{n}m)$. Taking derivative, setting it to be equal to zero we get

$$\overline{n} = \sqrt{\frac{128n}{m}}, \overline{m} = 128/\overline{n} \tag{2}$$

Correctness. For correctness we require for all pairs of indices (i,j), $|(R\cdot E)_{ij}| < \frac{q}{8} - \frac{1}{2}$. This way both parties will get the same key. For a fixed pair (i,j), we bound the probability p_{ij} of $|(R\cdot E)_{ij}| > \frac{q}{8} - \frac{1}{2}$ as the probability of the sum of m independent Gaussians variables with standart deviation σ to exceed $\frac{q}{8} - \frac{1}{2}$. The sum of m independet Gaussians can be approximated with a Gaussian with standart deviation $\sqrt{m}$ times bigger. Therefore the probability can be approximated by

$$p_{ij} \le \int_{\frac{q}{8}-\frac{1}{2}}^{\infty} D_{\mathbb{Z},\sqrt{m}\sigma}(x)dx \le \frac{1}{2}\cdot\exp\left(-\frac{\left(\frac{q}{8}-\frac{1}{2}\right)^2}{2m\sigma^2}\right)$$

The probability that at least one coefficient of k_A and k_B disagree is clearly bounded above by the sum of all the p_{ij}, so we get

$$\Pr(k_A \ne k_B) \le \sum_{i=0}^{\overline{n}}\sum_{j=0}^{\overline{m}} p_{ij} \le \frac{\overline{n}\cdot\overline{m}}{2}\cdot\exp\left(-\frac{\left(\frac{q}{8}-\frac{1}{2}\right)^2}{2m\sigma^2}\right) \tag{3}$$

For our choice of parameters we need the quantity in Eq. 3 to be much smaller than the security advantage 2^{-128}.

Real parameters for 128bits security. The hardness of LWE depends on the magnitude of the noise with respect to the modulus of the scheme. The smaller the ratio q/r, the easier the problem is, that's why q can not be too big. Sample parameters from [vdPS13] paper gives an upper bound on q based on σ and n for security level 128:

$$q < 2^{41}, \sigma = 3.2, n = 1024 \tag{4}$$

Taking parameters estimates from Eq.4 ($n = 2^{10}$, $q = 2^{16}$), we get that the optimal m for communication that satisfies the requirement in Eq. 1 is $m = 2^{14}$. The total communication is therefore equal to

$$\overline{n} = \sqrt{\frac{128n}{m}} \approx 3 \tag{5}$$

$$\overline{m} = 128/\overline{n} = 46 \tag{6}$$

$$(\overline{m}n + m\overline{n})\log q = 16(1024\cdot 46 + 2^{14}\cdot 3) = 188KB \tag{7}$$

Comparing to RLWE-TLS paper, the total communication there is 8KB.

The correctness requirement from Eq. 3 is satisfied with a big enough margin: $64 * \exp(-(2^{16}/8 - 0.5)^2/(2 \cdot 2^{14} \cdot 3.2^2)) \approx 2^{-282}$.

Summarizing our parameters for this protocol will be:

$$\boxed{\begin{aligned} q &= 2^{16} \\ n &= 2^{10} \\ m &= 2^{14} \\ \sigma &= 3.2 \\ \overline{n} &= 3 \\ \overline{m} &= 46 \end{aligned}}$$

2 Second Protocol. Parameters Estimation.

In this protocol instead of using a leftover hash lemma on the Client's side to generate a random matrix B' we apply LWE another time on another side of the matrix A. Note that the matrix A is square and all the secret matrices S, S' are coming from a bounded noise distribution, as opposed to from a uniformly random distribution as in the previous protocol.

TODO: Verify the parameters of the reduction for short non-uniform secrets.

$$A \xleftarrow{\$} \mathbb{Z}_q^{n \times n}$$

Alice (Server)		Bob (Client)
$S, E \xleftarrow{\chi} \mathbb{Z}_q^{n \times \overline{n}}$		
$B := AS + E$		
	$\xrightarrow{B \in \mathbb{Z}_q^{n \times \overline{n}}}$	
		$S', E' \xleftarrow{\chi} \mathbb{Z}_q^{\overline{m} \times n}$
		$B' := S'A + E'$
		$E'' \xleftarrow{\chi} \mathbb{Z}_q^{\overline{m} \times \overline{n}}$
		$V := S'B + E'' =$
		$= S'AS + S'E + E''$
		$C := \langle V \rangle_2$
	$\xleftarrow{B' \in \mathbb{Z}_q^{\overline{m} \times n}, C \in \mathbb{Z}_2^{\overline{m} \times \overline{n}}}$	
$K := rec(B'S = S'AS + E'S, C)$		$K := \lfloor V \rceil_2 = \lfloor S'AS + S'E + E'' \rceil_2$

Figure 2: Second LWE-based key exchange protocol.

Choosing $\overline{n}$ and $\overline{m}$. To get the key size to be equal to 128 bits we set $\overline{n} \cdot \overline{m} = 128$ and to minimize the communication we set $\overline{n} = \overline{m} = \sqrt{128} \approx 12$.

Correctness. For correctness we require that for all pairs of indices (i, j), $|(E'S + S'E + E'')_{ij}| < \frac{q}{8} - \frac{1}{2}$. For a fixed pair (i, j), we bound the probability of $|(E'S + S'E + E'')_{ij}| > \frac{q}{8} - \frac{1}{2}$ as follows. There are $2n + 1$

terms in the sum, if the (ij) element is greater than $\frac{q}{8}-\frac{1}{2}$, then at least one of the elements of the gaussian matrix must exceed $z=\sqrt{\frac{q-4}{8(2n+1)}}$ ($z\approx 511$ for $q=2^{32}$) in absolute value. The probability of individual gaussian coefficient exceeding z in absolute value can be bounded by $e^{-(z/(\sqrt{2}\sigma))^2}$. The probability that one out of (4n + 1) exceeds z is bounded above by the sum $(4n+1)e^{-(z/(\sqrt{2}\sigma))^2}$. Similarly, the probability that at least one coefficient of k_A and k_B disagree is clearly bounded above by the sum of all the p_{ij}, so we get

$$\Pr(k_A \neq k_B) \leq \sum_{i=0}^{\overline{n}} \sum_{j=0}^{\overline{n}} p_{ij} \leq \overline{n}^2(4n+1)e^{-(q-4)/16(2n+1)\sigma^2} \tag{8}$$

For our choice of parameters we need the quantity in Eq. 8 to be much smaller than the security advantage 2^{-128}.

Real parameters for 128bits security. Choosing (for correctness) $q=2^{32}$, $n=1024$, $\sigma=3.2$ (as in [BCNS14]), having $\overline{n}=\overline{m}=12$, we get the communication to be equal to $(2n\overline{n}\log q)$ bits = 96KB.

$$\boxed{\begin{aligned} q &= 2^{32} \\ n &= 2^{10} \\ \sigma &= 3.2 \\ \overline{n} &= \overline{m} = 12 \end{aligned}}$$

The correctness requirement from Eq. 8 is satisfied with a big margin: $144\cdot(4\cdot 1024+1)\cdot\exp(-(2^{32}-4)/16/2049/3.2^2)\leq 2^{-2^{14}}$.

3 Side Results.

Estimating the running time of the existing algorithms for LWE also shows that for given parameters ($n=1024, q=2^{16}$ or $q=2^{32}$ the number of operations required to solve a decision problem is more than 2^{128}. For that see [APS15] and their script (`https://bitbucket.org/malb/lwe-estimator`) with the code below:

```
load("https://bitbucket.org/malb/lwe-estimator/raw/HEAD/estimator.py")
n, alpha, q = 1024, alphaf(8,2^32-1), 2^32-1
set_verbose(1)
_ = estimate_lwe(n, alpha, q, skip=["arora-gb"])
```

From [BLP+13]: *"Combined with our modulus reduction, this has the following interesting consequence: the hardness of n-dimensional LWE with modulus q is a function of the quantity $n\log_2 q$. In other words, varying n and q individually while keeping $n\log_2 q$ fixed essentially preserves the hardness of LWE."*

From [BLP+13], Section 2.3: *"It follows from our results that (decision) LWE is hard not just for a smooth modulus q..., but actually for all moduli q, including prime moduli..."* (they operate with a power of two moduli q).

From [MP12], *"We also mention that the simplest and most practically efficient choices of G work for a modulus q that is a power of a small prime, such as $q=2^k$, but a crucial search/decision reduction for LWE was not previously known for such q, despite its obvious practical utility."* Provide a very general reduction for q like $q=2^k$ that are divisible by powers of very small primes. *"Altogether, for any n and typical values of $q\geq 2^{16}$..."*

See https://www.math.auckland.ac.nz/∼sgal018/gen-gaussians.pdf on how to compute the discrete Gaussian distribution (Section 4.2).

Having $\sigma\sqrt{2\pi} > \sqrt{n}$ allows the reduction of GapSVP to LWE to go through [Reg09] (as stated in Page 2, [ARS15]. (There $\sigma = \alpha q/\sqrt{2\pi}$).

The LWE problem is characterized by n, α, q, ψ, where ψ is the distribution of the elements of the secret vector.

References

[APS15] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Technical report, Cryptology ePrint Archive, Report 2015/046, 2015. http://eprint. iacr. org/2015/046, 2015.

[BCNS14] Joppe W Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the tls protocol from the ring learning with errors problem. Technical report, 2014.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology–EUROCRYPT 2012*, pages 700–718. Springer, 2012.

[Pei14] Chris Peikert. Lattice cryptography for the internet. In *Post-Quantum Cryptography*, pages 197–219. Springer, 2014.