A Simple Key Reuse Attack on LWE and Ring LWE Encryption Schemes as Key Encapsulation Mechanisms (KEMs)

Jintai Ding[1], Chi Cheng[2], Yue Qin[2]

University of Cincinnati
China University of Geosciences

Abstract. In this paper, we present a simple attack on LWE and Ring LWE encryption schemes used directly as Key Encapsulation Mechanisms (KEMs). This attack could work due to the fact that a key mismatch in a KEM is accessible to an adversary. Our method clearly indicates that any LWE or RLWE (or any similar type of construction) encryption directly used as KEM can be broken by modifying our attack method according to the respective cases.

keywords: LWE, RLWE, encryption, KEM

1 Introduction

Public key cryptography is the foundation of our modern communication systems. Public key cryptosystems, in secure communications, are not used to send usual messages but they are used to build a class of mechanism to secure short symmetric cryptographic key material for transmission, because public key systems are rather inefficient in transmitting long messages. Classically we use RSA and Diffie-Hellman to build Key encapsulation Mechanisms (KEMs). Recently, NSA has announced plans to transit to quantum resistant cryptographic primitives for its Suite B cryptographic algorithms and NIST started the standardization process in 2016.

LWE and Ring LWE problems are now considered as one of the most promising tools to build next generation post-quantum algorithms, which can resist quantum computer attacks. The point of this paper we want to make here is that LWE or RLWE encryption schemes are secure as encryption schemes alone, but they are not secure once they are used directly as KEM since a key mismatch in a KEM is accessible to an adversary. This work is based on the new key mismatch attack developed in [4].

Here first we will present a complete attack on standard LWE and RLWE based KEM by showing that an adversary can derive the secret key of the LWE or RLWE based encryption by performing a number of times of KEM. We provide a detailed description on how such an attack is performed.

2 Preliminaries

2.1 Learning with Errors and RLWE

The Learning with Errors (LWE) problem is a generalization of the parity-learning problem introduced by Oded Regev in 2005 [10]. Regev also showed a quantum reduction from solving LWE in the average case to solving worst case lattice problems such as the Shortest Vector Problem (SVP) and the Shortest Independent Vectors Problem (SIVP). In 2009, Peikert showed a classical reduction from variants of the shortest vector problem to corresponding versions of LWE [9].

The LWE problem is parameterized by a modulus q, dimension n and an error distribution χ on $\mathbb{Z}_q$. Then, the decision version of the LWE problem is to distinguish the following two distributions: $(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + e)$ and $(\mathbf{a}, \mathbf{b})$, where $\mathbf{a}$, $\mathbf{s}$, $\in \mathbb{Z}_q^n$ and $\mathbf{b} \in \mathbb{Z}_q$ are sampled uniformly at random and $e \leftarrow \chi$ from the error distribution. The search version is to find s given $poly(n)$ number of samples $(a_i, a_i \cdot s + e_i) = (a_i, b_i)$. The Ring Learning with Error (RLWE) problem is the version of LWE using polynomial rings and is preferred over LWE due to its efficiency and potentials for practical implementations. We provide the definition of the Discrete Gaussian distribution (error distribution) here:

Discrete Gaussian Distribution

Definition 1. *[11] For any positive real $\alpha \in \mathbb{R}$, and vectors $c \in \mathbb{R}^n$, the continuous Gaussian distribution over $\mathbb{R}^n$ with standard deviation centered at v is defined by the probability function $\rho_{\alpha,c}(x) = (\frac{1}{\sqrt{2\pi\alpha^2}})^n exp(\frac{-\|x-c\|^2}{2\alpha^2})$. For integer vectors $c \in \mathbb{R}^n$, let $\rho_{\alpha,c}(x) = \sum_{x \in \mathbb{Z}^n} \rho_{\alpha,c}(x)$. Then, we define the discrete Gaussian distribution over $\mathbb{Z}^n$ as $D_{\mathbb{Z}^n,\alpha,c}(x) = \frac{\rho_{\alpha,c}(x)}{\rho_{\alpha,c}(\mathbb{Z}^n)}$, where $x \in \mathbb{Z}^n$. The subscripts s and c are taken to be* 1 *and* 0 *(respectively) when omitted.*

Let n be an integer and a power of 2. Define $f(x) = x^n + 1$ and consider the ring $R := \mathbb{Z}[x]/\langle f(x) \rangle$. For any positive integer q, we define the ring $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$ analogously, where the ring of polynomials over $\mathbb{Z}$ (respectively $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$) we denote by $\mathbb{Z}[x]$ (respectively $\mathbb{Z}_q[x]$). Let χ_α denote the discrete Gaussian distribution on R_q with parameter α. Let the norm $\|p\|$ of a polynomial $p \in R$ (or R_q) be defined as the norm of the corresponding coefficient vector in $\mathbb{Z}$ (or $\mathbb{Z}_q$)

We recall two useful lemmas here:

Lemma 1 ([11]). *Let $f(x)$ and R be defined as above. Then, for any $s, t \in R$, we have $\|s \cdot t\| \leq \sqrt{n} \cdot \|s\| \cdot \|t\|$ and $\|s \cdot t\|_\infty \leq n \cdot \|s\|_\infty \cdot \|t\|_\infty$.*

Lemma 2 ([8, 5]). *For any real number $\alpha = \omega(\sqrt{\log n})$, we have* $\Pr_{\boldsymbol{x} \leftarrow \chi_\alpha}[\|\boldsymbol{x}\| > \alpha\sqrt{n}] \leq 2^{-n+1}$.

Let $s \leftarrow R_q$ be a uniformly chosen element of the ring R_q, as defined above. We define A_{s,χ_α} to be the distribution of the pair $(a, as + e) \in R_q \times R_q$, where $a \leftarrow R_q$ is uniformly chosen and $e \leftarrow \chi_\alpha$ is independent of a.

Definition 2 (Ring-LWE Assumption[7]). *Let R_q, χ_α be defined as above, and let $s \leftarrow R_q$ be uniformly chosen. The (special case) ring-LWE assumption $RLWE_{q,\alpha}$ states that it is hard for any PPT algorithm to distinguish A_{s,χ_α} from the uniform distribution on $R_q \times R_q$ with only polynomial samples.*

The search version of RLWE is for a PPT algorithm to find s rather than distinguish the two distributions. For certain parameter choices, the two forms are polynomially equivalent [7]. The *normal form* [3, 2] of the RLWE problem is by modifying the above definition by choosing s from the error distribution χ_α rather than uniformly. It has been proven that the ring-LWE assumption still holds even with this variant [1, 7].

Proposition 1 ([7]). *Let n be a power of 2, let α be a real number in $(0,1)$, and q a prime such that $q \bmod 2n = 1$ and $\alpha q > \omega(\sqrt{\log n})$. Define $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ as above. Then there exists a polynomial time quantum reduction from $\widetilde{O}(\sqrt{n}/\alpha)$-SIVP (Short Independent Vectors Problem) in the worst case to average-case RLWE$_{q,\beta}$ with ℓ samples, where $\beta = \alpha q \cdot (n\ell/\log(n\ell))^{1/4}$.*

2.2 LWE and RLWE encryption schemes

The LWE case. Suppose we have m LWE samples, where m is bigger than n, $(a_i, a_i \cdot s + e_i) = (a_i, b_i)$ and we will rewrite theses samples in a matrix form. We assume that a_i and s are column vectors. Then we have a matrix A and two column vector B and E such that

$$A = \begin{pmatrix} a_1^t \\ a_2^t \\ \cdot \\ \cdot \\ a_m^t \end{pmatrix}, B = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_m \end{pmatrix}, E = \begin{pmatrix} e_1 \\ e_2 \\ \cdot \\ \cdot \\ e_m \end{pmatrix},$$

where

$$B = A \times s + E.$$

In an LWE encryption scheme, the public key is given as A and B and the secret key is s.

To encrypt one bit message t (being 0 or 1), one finds a row vector M of length m where the entries are uniformly independently chosen from 0 and 1. Then compute

$$a' = M \times A$$

and

$$b' = M \cdot B + t(q-1)/2$$

and send (a', b') as the cipher-text.

To decrypt, we calculate

$$c' = b' - a' \cdot s = M \cdot E + t(q-1)/2,$$

and if c' is near zero, we decrypt as 0 and if it is near $(q-1)/2$, we decrypt it as 1. Here we assume p to be an odd number, otherwise we can round it to be an integer.

The RLWE case In the case of RLWE encryption, we use the normalized version of RLWE, namely the public key is given as a pair: $(a, b) = (a, as+e) \in R_q \times R_q$, where $a \leftarrow R_q$ is uniformly chosen and $s, e \leftarrow \chi_\alpha$ and they are all independent. Note here this means that both coefficients of s and e are small. Here

$$a = a_0 + a_1 x + ... + a_{n-1}x^{n-1},$$

$$s = s_0 + s_1 x + ... + s_{n-1}x^{n-1},$$
$$e = e_0 + e_1 x + ... + e_{n-1}x^{n-1}.$$

To encrypt a message $m = m_0 + m_1 x + ... + m_{n-1}x^{n-1}$, where m_i are either 0 or 1, one finds independently $e', e'', c \leftarrow \chi_\alpha$, and computer the ciphertext pair to be

$$(a', b') = (ca + e', cb + e'' + \frac{(q-1)}{2}m).$$

To decrypt the message, one computes

$$c' = b' - a' \times s = cas + ce + e'' - csa - e's + \frac{(q-1)}{2}m = ce + e'' - e's + \frac{(q-1)}{2}m,$$

then look at the each coefficient to see if it is near 0 or $(q-1)/2$, which gives the correct answer due to the fact that s, e', e'', c are all small. Here we would like to fix the decryption algorithm such that coefficients from the interval $(-\lfloor q/4 \rceil, ..., \lfloor q/4 \rceil)$ will be rounded to 0 and the rest will be round to 1.

2.3 LWE and RLWE encryption schemes as KEM

Let us assume that we will use LWE encryption as a KEM. Namely one party Alice will announce its public key and then anyone wants to communication securely with Alice would first try to build a key length of L (for example 256) bits by encrypting L bits to Alice and then perform the secure communications. We also know if such a session is a failure, namely if the keys do not match, then such a communication will break down immediately, since no one could read each other's messages, therefore an adversary will know immediately if such a case occurs.

In the case, RLWE, Alice can have a public key and anyone can use one encryption session to derive n bits shared keys.

3 The Attack – on an LWE KEM and on RLWE KEM

3.1 The LWE case

Suppose someone wants to attack Alice's KEM, it will go through the following steps.

1. This adversary will first perform a KEM session, which consists of L sessions of encryption of a single bit to get L bits shared keys.
   In the first encryption session, he will choose the first M to be:
   $$M = (y, 0, 0, .., 0),$$
   where $y = 1$ and set t to be 0, in this case, ciphertext will be
   $$(a', b') = (M \times A, M \times (A \times s + E)) = (ya_1^t, ya_1 \cdot s + ye_1),$$
   Alice will use $c' = ye_1 = e_1$, since we know that e_1 is close to 0, the decryption will be 0
   Then we will do the rest of $L-1$ encryption sessions to be honest random encrytption sessions, namely choose M_i as he or she should. In this case, the keys will match clearly.

2. Then in the next KEM session, we will do the same but in the first ecryption session we choose
$$M = (y, 0, 0, .., 0),$$
where $y = 2$ and set t to be 0, in this case, ciphertext will be
$$(a', b') = (ya_1, ya_1 \cdot s + ye_1),$$
Alice will use $c' = ye_1 = 2e_1$, since we know that e_1 is close to 0, the decryption will be 0 again in general.
3. In the following KEM session, we will do the same but we choose
$$M_1 = (y, 0, 0, .., 0),$$
where $y = 3$. Then next will be the same but increase y by 1.
By doing this, for the KEM sessions, the shared key will start to mismatch eventually, since
$$c' = ye_1,$$
and the absolute value of c' will be closer to $(q-1)/2$ as y increases, but not 0. We can find the absolute value of e_1. For example, if the e_1 is equal to 1 or -1, then the mismatch should happen at about $y = \lfloor q/4 \rceil$. If the e_1 is equal to 2 or -2, the mismatch should happen at about $y = \lfloor q/8 \rceil$.
4. Suppose we figure out the absolute value of e_1, for simplicity, let us assume that e_1 is 1 or -1 then we can easily figure out the sign of e_1 by doing the same things except that the ciphertext will be
$$(a', b') = (ya_1, ya_1 \cdot s + ye_1 + \lfloor q/4 \rceil - \epsilon),$$
where ϵ is a small positive integer like 3. In this case, if e_1 is 1, the mismatch will happen in at most $\epsilon + 1$ steps and if it is -1, it will not happen in $2\epsilon + 1$ steps. Therefore, the above process allows us to find the value of e_1.
5. Similarly if we use $M = (0, y, 0, ..., 0, 0)$, we can find e_2.
6. In sum, we can use $M = (0, ..., 0, y, 0, .., 0, .., 0)$, where y is at the i-th position to find e_i. This will allow us to find E, which then allows us to find s by solving a linear system. Then we find the secret key for the encryption scheme.

Someone may say that such an attack can be easily prevented if Alice sees that the first component of the ciphertext is a multiple of a row of A, Alice will reject it immediately. However, one can see that if we choose
$$M = (y, 0, 0, ..0, 1, 0.., 0, 1, 0.., 0, 1, 0.., 0),$$
where we have a very few 1s in random positions, then we can still can make it work just like what is explained in [4].

3.2 The attack on a RLWE KEM

Suppose someone wants to attack Alice's KEM, the attacker will got through the following steps. Let us assume that all the entries of e_i is bounded by $q/8$, which is true in general.

1. In the KEM sessions, the attacker will first choose m to be 0 (as a polynomial in R_q) and e' to be zero (as a polynomial in R_q) and c to be 2 (as a polynomial in R_q). He chooses
$$e'' = \lfloor q/4 \rceil + y,$$
where
$$y = -\lfloor q/8 \rceil, -\lfloor q/8 \rceil + 1, ... - 1, 0, 1, 2, ..., \lfloor q/8 \rceil.$$
In this case, the ciphertext will be
$$(a', b') = (2a, 2b + e') = (2a, 2as + 2e + e'').$$
Alice will use
$$c' = 2e+e'' = 2e+\lfloor q/4 \rceil+y = (2e_0+\lfloor q/4 \rceil+y+2e_1x+2e_2x^2+...+2e_{n-1}x^{n-1},$$
since we know that e_i is very close to 0, the decryption will be 0 if $y = -\lfloor q/8 \rceil$ and 1 if $y = \lfloor q/8 \rceil$ the decryption will be 1. Then by observing when the mismatch will happen, we can decide the value of e_0, since for each different value the mismatch will happen at different point of y. For example, if e_0 is zero, the mismatch happens when y changes from $y = 0$ to $y = 1$, and if $e_0 = 1$, the mismatch happens when y changes from $y = -1$ to $y = 0$.
2. In the next KEM session, we will do the same except that we will choose
$$c = -2x^{n-1},$$
then we have
$$c' = -2x^{n-1}e + e" = -2x^{n-1}e + \lfloor q/4 \rceil + y.$$
This will give us the value of e_1 since the constant term of $-x^{n-1}$ is exactly e_1.
3. Next, in the following KEM session, we will do the same but we choose
$$c = -2x^{n-2},$$
then we can get the value of e_2.
4. More generally by using
$$c = -2x^{n-i},$$
then we can find e_i. This will allow us to find e, which then allows us to find s by solving a linear system. Finally, we find the secret key for the encryption scheme.

3.3 The attack on other LWE or RLWE KEM

One can see that all existing LWE or RLWE based encryption schemes are nothing but minor modifications of the two fundamental schemes above. Therefore we can easily adapt what we have done and the techniques developed in [4].

We note here that the attacks here can be easily optimized.

3.4 Conclusion

We have demonstrated how we can break the two fundamental LWE and RLWE encryption schemes used directly as KEM. The reason this attack could work is due to the extra information we can derive from key mismatch. Though, the origin of such an attack in spirit can trace all the way back to the work of researchers in NSA [6]. Also from the works of [4] and [6] we know that simple preventive measures to stop such attacks are futile since attacker can easily adapt the attacks. But the preventive measures will also make the schemes more or less unusable if we do a lot of such things.

4 Acknowledgement

We would like to thank Tsuyoshi Takagi for useful discussions.

Bibliography

[1] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer Berlin Heidelberg, 2009.

[2] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

[3] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer Berlin Heidelberg, 2011.

[4] J. Ding, S. R. Fluhrer, and S. RV. Complete attack on RLWE key exchange with reused keys, without signal leakage. In *Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings*, pages 467–486, 2018.

[5] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[6] D. Kirkwood, B. C. Lackey, J. McVey, M. Motley, J. A. Solinas, and D. Tuller. Failure is not an option: Standardization issues for post-quantum key agreement, 2016. http://csrc.nist.gov/groups/ST/post-quantum-2015/presentations/session7-motley-mark.pdf.

[7] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer Berlin / Heidelberg, 2010.

[8] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37:267–302, April 2007.

[9] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 333–342, New York, NY, USA, 2009. ACM.

[10] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[11] J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen. Authenticated key exchange from ideal lattices. In *Advances in Cryptology-EUROCRYPT 2015*, pages 719–751. Springer, 2015.