International Journal of
Mathematics And its Applications
ISSN: 2347-1557

Int. J. Math. And Appl., **13(1)**(2025), 9–15
Available Online: http://ijmaa.in

R-LWE Based Secure Proxy Signature Scheme

Swati Thakur[1,*], Hemlal Sahu[2]

[1] *Department of Mathematics, Government Naveen College, Nawagaon, Naya Raipur, Raipur, Chhattisgarh, India*

[2] *Department of Mathematics, Government J. Yoganandam Chhattisgarh College, Raipur, Chhattisgarh, India*

Abstract

In the present scenario where thousands and millions of digital documents are transferred every day, the security of data has become a topic of global concern. Along with other cryptographic primitives, the proxy signature scheme is also widely used in various applications. The main part of our paper is a lattice-based proxy signature scheme. A Proxy Signature scheme is an arrangement that allows the transfer of singing rights from one entity or person to another on behalf of the original signer. Our scheme is based on the Ring-Learning with Error(R-LWE) problem which provides strong security with the worst-case to average-case reduction problem. Key construction and operations in our scheme are based on a ring of polynomials due to which both public and private keys are reduced and the size of the proxy signature also shrunk, while the security of our proposed scheme is based on the Short Integer Solution (SIS) problem.

Keywords: Lattice-based Cryptography; Proxy signature; LWE.

1. Introduction

Nowadays proxy signature schemes have been widely used in various applications like grid computing, mobile agent applications, global networks, mobile communications, and electronic payments. Initially, the concept of proxy signature was introduced by Mambo et.al. in 1996 [13], which is a special kind of digital signature. A proxy signature scheme is a system in which an entity or a person (original signer) gives his signing authority to another person(proxy signer) on his behalf in case of temporal absence or lack of time. The proxy signature generated by the proxy signer can be verified by the receiver with the public key of both the original signer and the proxy signer. According to the different ways of authorization, proxy signatures can be categorized as full proxy signature, partial proxy signature, and proxy signature with warrant.

- **Full delegation** – In this case, the original signer gives his private key to the proxy signer, later this secret key is used to sign documents by the proxy signer. In this case, a proxy signature

[*] Corresponding author (thswati211@gmail.com)

is the same as the digital signature scheme only difference is that it can be signed by the proxy signer.

- **Partial delegation** - In this case the original signer generates a proxy signature key from its private key and gives it to the proxy signer and then the proxy signer uses this proxy key to sign. The verification equation for the proxy signature is a little modified so that the proxy signature is not similar to the signature created by the original signer.
- **Delegation by warrant** - A warrant or certificate is the form of some message generated by the original signer and encrypted by his secret key, then sent to the proxy signer. This warrant is verified by the proxy signer and then he generates the proxy signature using his corresponding private key. The resulting signature consists of the created signature by the proxy signer and the warrant by the original signer.

Full delegation and Partial delegation may have security issues as it has a chance of misuse of a signature by the proxy signer on the other hand, they are also not unforgeable as the original signer can generate the proxy signature, while delegation by the warrant is more practical and secure and it does not have issues like previous two cases. Currently most proxy signature schemes are based on discrete logarithm problems or large integer factoring problems, but these classical hard problems which are currently in use for many applications will face security issues as quantum computers will be in common use. After quantum algorithms given by Shor in 1994 [18], by which discrete logarithm and integer factoring problems have polynomial time solutions. These classical hard problems will not be able to provide security against quantum computers. It's necessary to find a tool that can provide the required security in the presence of quantum computers in the future. Lattice-based cryptography is one of the most emerging and secure post-quantum cryptography. This attracts many researchers due to its proven security of worst-case to average-case reduction against a quantum computer. The first proven secure digital signature scheme was proposed by Genty et.al. in 2008 which was based on the trapdoor hash function [3]. In 2010, Jiang et.al. proposed a lattice-based proxy signature based on the Bonsai Tree Basis technique, but this scheme was successfully broken by Tian et. al. in 2011 [20]. In 2011 some Bonsai Tree Basis expansion technique-based proxy signature scheme was given by Wang et.al. and Xia differently [20]. In 2013, Yu Lei presented a high-efficiency proxy signature scheme based on a fixed dimension basis expansion technique. These schemes used lattice basis expansion techniques and two times primage sampling algorithm, but these schemes are very time-consuming. In 2015, Yang et.al proposed an efficient proxy signature scheme that was based on Lyubashevsky's digital signature scheme without trapdoor [20]. Following Ducas's Bimodal Gaussian Distribution Jiang et.al. gave another proxy signature scheme which claimed to have a smaller key size [17].

In this paper we have proposed a post-quantum proxy signature scheme based on lattices, we have combined the Ring Tesla digital signature scheme given by Akleylek et.al. [1] and proxy signature scheme and proved that it is unforgeable, non-repudiation.

2. Preliminary

2.1 Notation

we know $\mathbb{Z}_q$ denotes the quotient ring Z/qZ having the coefficients in $(-q/2, q/2)$ for q be a prime $q = 1 \bmod 2n$ i.e., and $\mathbb{Z}_q[x]$ is polynomial ring modulo q and $R_q = Z_q[x]/x^n + 1$ is ring over integer. Other notation, rounding operator $\lfloor v \rceil_{d,q}$ and parameters are similar like ring Tesla signature scheme [1].

Key generation:- This algorithm takes as input the security parameter, system public parameter, original and proxy signer generate their own secret key and verification key separately.

Proxy Key generation:- The original signer A generates the signature (proxy key) z of warrant w using his secret key and sends it to B along with the warrant.

Proxy sign:- First proxy signer verifies the warrant sent by A , the proxy signer B generates the proxy signature and signs the message μ then outputs proxy signature (z, w, z', μ).

Proxy verify:- The verifier verifies the proxy signature by using the verification keys of both the original and proxy signer.

A proxy signature can be analyzed with the following security parameters:-

Unforgeability:- Unforgeability shows that a valid proxy signature can be generated by only the proxy signer, neither the original signer or any other third party can generate it and a valid proxy signature key can be generated by only original signer, nor the proxy signer or any other third party can generate it.

Undeniability:- Once the proxy signature is generated by proxy signer he can't deny from it.

key dependence:- The generation of the proxy key depends on the original signer's secrete key.

Verifiability:- Any receiver can verify that the original signer agreed on the signed message.

3. Proxy Signature Scheme

In this section, we have proposed our lattice-based proxy signature scheme with warrant. We have combined proxy signature and ring tesla signature to design a lattice-based proxy signature scheme. Our proxy signature has parameters $n, q, d, \omega, B, U, L,$ following like Ring-Tesla signature scheme, hash function $H : \{0,1\}^* \to \{v : v \in \{0,1\}^n, \|v^2\| = \omega\}$. Polynomial $a \in R_q$, $a = (a_0 + a_1 x + \cdot + a_{n-1}x^{n-1})$ is public parameter and choose e_1 from Gaussian distribution D_σ^n. Here original signer's signing key and verification key are s_1 and $t_1 = a.s_1 + e_1$ and proxy signer's signing key and verification key are s_2 and $t_2 = a.s_2 + e_2$. The original signer A generates a proxy signature key for proxy signer B using a, s_1 and warrant W.

- Proxy Key generation

  1. Choose a polynomial y_1 from D_σ^n

2. $v_1 = a.y_1 \bmod q$
3. $c = H(\lfloor v_1 \rceil_{d,q}, W)$
4. $z \leftarrow y_1 + s_1.c$
5. $w_1 \leftarrow v_1 - e_1.c \bmod q$
6. if $\lfloor w_1 \rceil_{d,q} \notin R_{2^d-L}$ and z $\notin R_{B-U}$

and send the proxy signature key (z, c) along with warrant W to B.

- When the proxy signer receives the signature key (z, W) from A, first he verifies by using the verification key of A, that the sender is the original signer by computing $w_1' \leftarrow a_1.z - t_1.c \bmod q$ and $c = H(\lfloor w_1 \rceil_{d,q}, W)$ if not verified proxy signer rejects the proxy key, if it succeeds in verification he signs the message with his secret key s_2
- Proxy Signature
  1. Choose a polynomial y_2 from D_σ^n
  2. $v_2 = a.y_2 \bmod q$
  3. $c_1 = H(\lfloor v_2 \rceil_{d,q}, \mu)$
  4. $z_1 \leftarrow y_2 + s_2.c_1$
  5. $u_1 \leftarrow v_2 - e_2.c_1 \bmod q$
  6. if $\lfloor u_1 \rceil_{d,q} \notin R_{2^d-L}$ and $z_1 \notin R_{B-U}$
  7. return proxy signature (z , W , z_1, μ)
- Proxy verify:- when the receiver receives the proxy signature (z, W, z_1, μ) from B, then he verifies the proxy signature by using both verification keys of A, B by computing $w_1 \leftarrow a_1.z - t_1.c \bmod q$, $c = H(\lfloor w_1 \rceil_{d,q}, W)$, $u_1' \leftarrow a_2.z_1 - t_2.c' \bmod q$, $c_1 = H(\lfloor u_1' \rceil_{d,q}, w)$ if verify then accept the signature otherwise rejects it.

3.1 Security

Only the proxy signer can generate a valid proxy signature, original signer nor any third party can generate the valid proxy signature:-

Case 1:- In this case the original signer has been considered as forger as he has more information than any other third party, as he knows the private key of the original signer so he can directly compute the proxy key, so the forgery of the signature (z, c, z_1, c_1) is the forgery of only (z_1, c_1). We introduce another signing way without knowing the secret key of the proxy signer, he firstly choose a vector c_1 uniformly from $\{v : v \in \{0, 1\}^n, \|v^2\| = \omega\}$ and choose z_1 according to discrete gaussian distribution and using rejection sampling output (z_1, c_1) with probability $1/M$ and $H(az_1 - tc_1, \mu) = c_1$.

Case 2:- Let the polynomial time forger A be for a signature scheme that takes a public key as input and makes h random oracle hash queries, s signing queries, and forges a valid signature with non-negligible probability δ then distinguisher D solving $R - LWE_{n,2,q,\sigma}$ in time $'$t$'$ with probability. When the challenger c receives the given, he randomly chooses a polynomial as the private key and computes the corresponding verification key. he sends a,t to Forger and keeps a private key secret.

- **Hash Queries:-** When a forger sends a hash query (v, μ) to the challenger $\mathcal{C}$, the challenger will first check whether this query (v, μ) was asked earlier or not, if it is, he returns the hash value c, otherwise $\mathcal{C}$ chooses uniformly $c \leftarrow \{v : v \in \{0,1\}^n, \|v^2\| = \omega\}$ and return it to the forger.

- **Sign Queries:-** The challenger $\mathcal{C}$ holds another list which contains a set of different messages with their respective signatures (μ, z, c), when the forger forges sign query for message μ, the challenger first check whether μ is in the list if it is, the challenger $\mathcal{C}$ returns the pre-exist signature (z,c) to forger, otherwise challenger $\mathcal{C}$ chooses uniformly $c \leftarrow \{v : v \in \{0,1\}^n, \|v^2\| = \omega\}$ and $z \leftarrow D_\sigma^n$ and return (z,c) to the forger.

- **Forge:-** When forger F outputs the forgeable signature (c', z') of the message μ' with $\|z\| \leq (B - U)$ and $H(az' - tc', \mu') = c'$.

Suppose that c was queried earlier then there exists a signature (c', z'') of the message μ' with $H(az'' - tc', \mu') = c'$ analyzing with forge signature $H(az'' - tc', \mu') = H(az' - tc', \mu')$ if $\mu \neq \mu'$ or $az'' - tc' \neq az' - tc'$ there will be hash collision then we have $\mu = \mu'$ or $az'' - tc' = az' - tc'$ then $a(z'' - z') = 0$ where $(z'' - z') \neq 0$ which is equivalent to $(z'' - z')$ is solution of SIS problem with $\|(z'' - z')\| \leq 2(B - U) = 28(n-1)\sqrt{\omega}\sigma$. Now to apply the Bellare and Neven version of the forking lemma, we take i^{th} component of random oracle output different c′ so, $c_i' \neq c_i*$ then we obtain a valid signature (z′, c′) on the same message μ. Then we have $\lfloor az - tc \rceil_{d,q} = \lfloor az' - tc' \rceil_{d,q}$, $az - tc = az' - tc'$ i.e. $a(z - z') - t(c - c') = 0 \bmod q$, $a((z - z') + s(c' - c)) + e(c - c') = 0 \bmod q$ taking $y_1 = ((z - z') + s(c' - c))$ and $y_2 = e(c - c')$ then we have a non zero solution of $ay_1 + y_2 = 0 \mod q$ with $\|y_1\| \leq 28(n-1)\sqrt{\omega}\sigma + 2\zeta\omega$ and $\|y_2\| \leq 2\omega$. Signature key sizes of original signer and proxy signer are $n \log 2(B - U)$ and size of public key is $2n \log q$ public and secrete key $3n \log 14\sigma$.

4. Conclusion

We have proposed a new post-quantum proxy signature scheme based on a ring of polynomials whose keys are secured by hard security provided by the learning With Errors problem. our scheme we have a multiplication of polynomials and a rejection sampling algorithm.

References

[1] S. Akleylek, N. Bindel, J. Buchmann, J. Krämer and G. A. Marson, *An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation*, In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds) Progress in

Cryptology – AFRICACRYPT 2016, AFRICACRYPT 2016, Lecture Notes in Computer Science(), Vol 9646, Springer, Cham. https://doi.org/10.1007/978-3-319-31517-1_3

[2] M. Ajtai, *Generating hard instances of lattice problems*, STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing, (1996), 99-108.

[3] D. J. Bernstein, J. Buchmann and E. Dahmen, *Introduction to post quantum cryptography*, Post-Quantum Cryptography, Springer, Berlin, Heidelberg, (2009).

[4] M. Bellare and P. Rogaway, *The exact security of digital signatures - How to sign with RSA and Rabin*, In U. M. Maurer, Editor, EUROCRYPT'96, Volume 1070 of LNCS, pages 399–416, Saragossa, Spain, May 12–16, 1996, Springer, Heidelberg, Germany.

[5] A. Boldyreva, A. Palacio and B. Warinschi, *Secure Proxy Signature Schemes for Delegation of Signing Rights*, J Cryptol., 25(2012), 57–115.

[6] D. Cash, D. Hofheinz and E. Kiltz, *Bonsai trees or how to delegate a lattice basis*, Journal of Cryptology, 25(4)(2012), 601-639.

[7] L. Ducas, A. Durmus, T. Lepoint and V. Lyubashevsky, *Lattice Signatures and bimodal Gaussians*, IN CRYPTO, (2013), 40-56.

[8] N. C. Dwarakanath and S. D. Galbraith, *Sampling from discrete Gaussians for lattice-based cryptography on a constrained device*, Applicable Algebra in Engineering, Communication and Computing, 25(2014), 159–180.

[9] C. Gentry, C. Peikert and V. Vaikuntanathan, *Trapdoors for Hard Lattices and New Cryptographic Constructions*, STOC '08: Proceedings of the fortieth annual ACM symposium on Theory of computing, (2008), 197-206.

[10] V. Lyubashevsky, *Lattice Signatures without Trapdoors*, In: Pointcheval, D., Johansson, T. (eds) Advances in Cryptology – EUROCRYPT 2012. EUROCRYPT 2012. Lecture Notes in Computer Science, vol 7237. Springer, Berlin, Heidelberg.

[11] V. Lyubashevsky, C. Peikert and O. Regev, *On Ideal Lattices and Learning with Errors over Rings*, In: Gilbert H. (eds) Advances in Cryptology – EUROCRYPT 2010. EUROCRYPT 2010. Lecture Notes in Computer Science, vol 6110. Springer, Berlin, Heidelberg.

[12] V. Lyubashevsky, C. Peikert and O. Regev, *A Toolkit for Ring-LWE Cryptography*, In: Johansson T., Nguyen P.Q.(eds) Advances in Cryptology – EUROCRYPT 2013. Lecture Notes in Computer Science, vol 7881. Springer, Berlin, Heidelberg.

[13] M. Mambo, K. Usuda and E. Okamoto, *Proxy signatures Delegation of the power to sign messages*, IEICE transactions on fundamentals of electronics, communications, and computer sciences, 79(9)(1996), 1338-1354.

[14] C. Peikert, *A Decade of Lattice Cryptography*, Foundation and Trends in Theoretical Computer Science, 10(4)(2016), 283-424.

[15] O. Regev, *On Lattices, learning with errors, random linear codes, and cryptography*, Journal of the ACM, 56(6)(2009), 1-34.

[16] Y. Jiang, F. Kong and X. Ju, *Lattice-based Proxy Signature [C]*, Computational Intelligence and Security (CIS), 2010 International Conference on IEEE, (2010), 382-385.

[17] Z. L. Jiang, Y. Liang, Z. Liu and X. Wang, *Lattice-based proxy signature scheme with reject sampling method*, 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Shenzhen, China, (2017), 558-563.

[18] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, 26(5)(1997), 1484-1509.

[19] F. Wu, W. Yao, X. Zhang and Z. Zheng, *An Efficient Lattice-Based Proxy Signature with Message Recovery*, In: Wang, G., Atiquzzaman, M., Yan, Z., Choo, KK. (eds) Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2017. Lecture Notes in Computer Science(), vol 10656. Springer, Cham.

[20] C. Yang, P. Qiu, S. Zheng and L. Wang, *An Efficient Lattice-Based Proxy Signature Scheme without Trapdoor*, 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Adelaide, SA, Australia, 2015, pp. 189-194.