POST-QUANTUM PUBLIC KEY CRYPTOSYSTEM FROM SUBSET PRODUCT WITH ERRORS

TREY LI

ABSTRACT. We give a new post-quantum public key cryptosystem from the multiple modular subset product with errors problem.

1. INTRODUCTION

In [Li22e] we proposed a new post-quantum key exchange. In this paper we give a new post-quantum public key cryptosystem (PKC).

2. HARD PROBLEM

Our PKC is based on the multiple modular subset product with errors problem (M-MSPE) defined in [Li22e] with the change that the error prime in each MSPE instance is sampled from either the first half or the second half of the error set L depending on whether the message bit is 0 or 1. We repeat the settings in [Li22e] with this change.

Setup

Let $\ell_1,\dots,\ell_{2^n}$ be the first 2^n primes[1]; and $p_1,\dots,p_n$ be the next n primes. Denote $L = \{\ell_1,\dots,\ell_{2^n}\}$ and $P = \{p_1,\dots,p_n\}$.[2]

Choose a safe/Mersenne prime q in $[\ell_{2^n}^{2n+1}, p_1^{n^2/8}]$ (e.g., the smallest safe/Mersenne prime greater than $\ell_{2^n}^{2n+1}$).[3]

Let D_a be the distribution that samples a vector $v = (v_1,\dots,v_n) \leftarrow \{0,1\}^n$ uniformly at random and outputs the integer $a := \prod_{i=1}^n p_i^{v_i}$.

Let D_e be the distribution that keeps sampling vectors $v = (v_0,\dots,v_{n-1}) \leftarrow \{0,1\}^{\lceil \log(\ell_{2^n})\rceil}$ until finding one such that the integer $e := \sum_{i=0}^{\lceil \log(\ell_{2^n})\rceil - 1} \left(v_i \cdot 2^i\right)$ is a prime in L (i.e. a prime $\leq \ell_{2^n}$) and outputs e.

Let $D_e(b)$ be the distribution that takes as input a bit $b \in \{0,1\}$ and keeps sampling vectors $v = (v_0,\dots,v_{n-2}) \leftarrow \{0,1\}^{\lceil \log(\ell_{2^n})\rceil}$ until finding one such that the integer $e := b \cdot 2^{\lceil \log(\ell_{2^n})\rceil - 1} + \sum_{i=0}^{\lceil \log(\ell_{2^n})\rceil - 2} \left(v_i \cdot 2^i\right)$ is a prime in L and outputs e.

Let $D_e^n(v)$ be the distribution that takes as input a vector $v = (v_1,\dots,v_n) \in \{0,1\}^n$, samples a vector of primes $e = (e_1,\dots,e_n) \in L$ with $e_i \leftarrow D_e(v_i)$ for $i \in [n]$, and outputs e.[4]

This is the 6th paper of the series. Previously: [Li22a; Li22b; Li22c; Li22d; Li22e].

Date: October 6, 2022.

Email: treyquantum@gmail.com

[1] Potential improvement of efficiency of the PKC may be achieved by replacing the parameter 2^n by a smaller super-polynomial function. This can help reducing the size of q. The tradeoff is a slight dropping of the correctness probability of the PKC.

[2] The two sets of primes can be chosen more randomly as long as $L \cap P = \varnothing$.

[3] See [Li22e] for the reason and recommended parameter.

[4] This notation is not necessary for the definition of M-MSPE but will be used in the PKC.

Let $D_x(b)$ with respect to some $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ be the distribution that takes as input a bit $b \in \{0,1\}$, samples $a_1, \ldots, a_n \leftarrow D_a$ and $e \leftarrow D_e(b)$, computes $X = \prod_{i=1}^n a_i^{x_i} \cdot e^{\pm 1} \pmod q$, and outputs $(a_1, \ldots, a_n, X)$, where the exponent ± 1 of e is arbitrary.

Let O_x with respect to some $x \in \{0,1\}^n$ be the oracle that outputs instances $(a_1, \ldots, a_n, X)$ sampled from $D_x(b)$, where $b \in \{0,1\}$ is arbitrary for each instance.

Problems

Search M-MSPE (or M-MSPE) is given access to O_x, find x.

Decision M-MSPE is given access to either O_x for some $x \in \{0,1\}^n$, or O_{ran} which outputs MSPE instances $(a_1, \ldots, a_n, X)$ with $X = \prod_{i=1}^n a_i^{x_i} \cdot e^{\pm 1} \pmod q$ replaced by $X \leftarrow \mathbb{Z}_q^\times$, decide which is the oracle given.

Hardness

To inherit hardness from the original M-MSPE in [Li22e], one thing to notice is that the hardness of the problem does not seem to be effected by tuning the forms of the error terms, including changing the exponents of the error primes in an error term, changing the number of error primes in an error term, or even changing the distribution of the error primes — as long as the entropy of the distribution is sufficiently high so that one cannot brute force it. This gives us opportunity to store information in the error terms. In particular, we go with the third way for our PKC, i.e., we use error primes that belong to the first or second half of L to represent 0 or 1. One could also use one or two error primes for an error term to represent 0 or 1. But that requires a larger modulus q which reduces the efficiency of the PKC.

3. IDEA

The goal of PKC is to encrypt a plaintext into a cyphertext using a public key such that only having the corresponding private key one can recover the plaintext from the ciphertext.

Our PKC idea is illustrated by the following figure.

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} & e_1 \\ \vdots & & \vdots & \vdots \\ a_{n,1} & \ldots & a_{n,n} & e_n \\ f_1 & \ldots & f_n & 1 \end{bmatrix} \begin{matrix} \xrightarrow{x} \\ \vdots \\ \xrightarrow{x} \\ \\ \end{matrix} \begin{matrix} A_1 \\ \\ A_n \\ \\ \end{matrix}$$

$$\begin{matrix} \downarrow y & \cdots & \downarrow y \\ B_1 & \cdots & B_n \end{matrix}$$

The public key is an M-MSPE (M, A), where $M = \{a_{i,j}\}_{i,j \in [n]} \leftarrow D_a^{n \times n}$ is the base matrix, and $A = (A_1, \ldots, A_n) \in (\mathbb{Z}_q^\times)^n$ is the M-MSPE product sequence. The private key is the corresponding secret $(x, e) \in \{0,1\}^n \times L^n$.

To encode an n-bit message $m \in \{0,1\}^n$, we sample a random vector $y = (y_1, \ldots, y_n) \leftarrow \{0,1\}^n$ and compute an M-MSPE product sequence $B = (B_1, \ldots, B_n)$ with the error term f_i of B_i sampled from $D_e(m_i)$. Compute a composite MSPE $C = A_1^{y_1} \cdots A_n^{y_n} \cdot f_{n+1} \pmod q$. The output is $c = (B, C)$.

To decode, use x to compute a composite product $D = B_1^{x_1} \cdots B_n^{x_n} \pmod q$. Then compute $E = C/D \pmod q$, which contains only error primes. Then use the private errors $e_1, \ldots, e_n$ to recover y by testing if $e_i | E$. Then recover the error terms f_i which tell the plaintext m.

4. Scheme

Public parameters are (n, q, L), where L is represented by the 2^n-th prime ℓ_{2^n}. Plaintext is $m \in \{0,1\}^n$.

KeyGen(n, q, L):

Sample a base matrix $M = \{a_{i,j}\}_{i,j\in[n]} \leftarrow D_a^{n\times n}$. Sample $(x, e) \leftarrow \{0,1\}^n \times D_e^n$. Compute $A = (A_1, \ldots, A_n)$, where $A_i = a_{i,1}^{x_1} \cdots a_{i,n}^{x_n} \cdot e_i \pmod q$ for $i \in [n]$. Public key is $pk := (M, A)$; private key is $sk := (x, e)$.

Encrypt(n, q, L, pk, m):

Sample $(y, f) \leftarrow \{0,1\}^n \times D_e^n(m)$. Compute $B = (B_1, \ldots, B_n)$, where $B_i = a_{1,i}^{y_1} \cdots a_{n,i}^{y_n} \cdot 1/f_i \pmod q$ for $i \in [n]$. Sample $f_{n+1} \leftarrow D_e$. Compute $C = A_1^{y_1} \cdots A_n^{y_n} \cdot f_{n+1} \pmod q$. Output cyphertext $c = (B, C)$.

Decrypt(n, q, M, x, e, c):

Compute $D = B_1^{x_1} \cdots B_n^{x_n} \pmod q$. Compute $E = C/D \pmod q$. Compute $y' \in \{0,1\}^n$ such that $y'_i = 1$ if and only if $e_i | E$, for $i \in [n]$. Compute $f'_i = a_{1,i}^{y'_1} \cdots a_{n,i}^{y'_n} / B_i \pmod q$ and set $m'_i = \mathrm{MSB}(f'_i)$ [5], for $i \in [n]$. Output $m' = (m'_1, \ldots, m'_n)$.

5. Correctness

Theorem 1. $m' = m$ with overwhelming probability.

Proof. Note that L is exponentially large and there are $4n + 2$ (i.e. linearly many) error primes (they are the e_i's and f_i's in the scheme) sampled either from (roughly) the first half of L or the second half of L. Hence the error primes are all different with overwhelming probability p.

Again recall that q is greater than the product of any $2n + 1$ primes in L. Hence

$$\begin{aligned} E &= \left(e_1^{y_1} \cdots e_n^{y_n}\right) \cdot \left({f_1}^{x_1} \cdots {f_n}^{x_n}\right) \cdot f_{n+1} \pmod q \\ &= \left(e_1^{y_1} \cdots e_n^{y_n}\right) \cdot \left({f_1}^{x_1} \cdots {f_n}^{x_n}\right) \cdot f_{n+1}. \end{aligned}$$

Suppose all error primes in the scheme are different. Then $y_i = 1$ if and only if $e_i | E$. Then $y' = y$ and thus $f'_i = f_i$ for all $i \in [n]$. Then $m' = m$. Therefore $m' = m$ with overwhelming probability p. □

6. Efficiency

Due to the underlining operation being multiplication instead of addition, the modulus q and thus the size of the encoding is inevitably large. However the time complexity is actually not very high due to the logarithmic complexity of modular multiplication.

Theorem 2. The time complexities of key generation, encryption and decryption are $O(n^5)$, $O(n^4)$ and $O(n^3)$ respectively.

Proof. The complexities mainly come from modular multiplications. Note that $q \gtrapprox \ell_{2^n}^{2n+1} \gtrsim (n2^n)^{2n+1} = 2^{O(n^2)}$. Hence the complexity of a single modular multiplication is $O(\log_2 q) = O(n^2)$. There are $O(n^3)$ modular multiplications in key generation ($O(n^3)$ to create the base

[5] That is, the most significant bit of f'_i in its length-$\lceil \log(\ell_{2^n}) \rceil$ binary representation.

matrix M and $O(n^2)$ to compute the M-MSPE product sequence A), $O(n^2)$ modular multiplications in encryption, and $O(n)$ modular multiplications in decryption. Hence the time complexities of the three algorithms are $O(n^5)$, $O(n^4)$ and $O(n^3)$. □

7. Security

The differences between the problem that we use to construct our PKC and the M-MSPE in Section 2 are: (1) instead of giving unlimited access to the oracle O_x, the PKC only gives $n+1$ MSPE instances; (2) one of the instances is a special one whose bases $A_1,\ldots,A_n$ are themselves MSPE products rather than regular bases sampled from D_a; and (3) the base matrix (M,A) (i.e. the public key) is reused in different implementations of encryption. We denote this M-MSPE as M-MSPE$_{PKC}$.

Decision M-MSPE$_{PKC}$ is defined similar to Decision M-MSPE in Section 2 with the above changes from M-MSPE to M-MSPE$_{PKC}$; and it asks to distinguish the MSPE$_{PKC}$ product sequence $(X_1,\ldots,X_{n+1})$ from uniform, with the bases (M,A) fixed.

Assume the hardness of Decision MSPE$_{PKC}$, we prove that our PKC is semantically secure against chosen plaintext attack. The key point of the following theorem is the hardness of Decision M-MSPE$_{PKC}$ with an *arbitrary* vector $v \in \{0,1\}^n$ in $D_e^n(v)$, where the vector v in the problem is actually the message m in the PKC. In other words, letting the error primes in each MSPE instance be chosen freely from the first half of L or the second half of L does make the resulting product sequence of the M-MSPE$_{PKC}$ less random.

Theorem 3. If Decision M-MSPE$_{PKC}$ is hard, then the PKC is semantically secure against chosen plaintext attack.

Proof. Let (B,C) and (B',C') be two ciphertexts of two messages m and m' respectively. Let d_1 be the distinguishable distance between (B,C) and uniform; d_2 be the distinguishable distance between (B',C') and uniform; and d_3 be the distinguishable distance between (B,C) and (B',C'). Now suppose for contradiction that there is a probabilistic polynomial time adversary $\mathcal{A}$ who is given the public key (M,A) and distinguishes (B,C) and (B',C') with noticeable probability. Then d_3 is noticeable. By the triangle inequality we have that $d_1 + d_2 > d_3$. Hence d_1+d_2 is noticeable. This means that at least one of d_1 and d_2 is noticeable. This contradicts the hardness of Decision M-MSPE$_{PKC}$, which says that both (B,C) and (B',C') are indistinguishable from uniform for any vectors m and m'. □

8. Acknowledgement

The author would like to thank Libin Wang for several discussions about efficiency.

References

[Li22a] Trey Li. "Subset Product with Errors over Unique Factorization Domains and Ideal Class Groups of Dedekind Domains". 1st paper of the series. 2022, October 1.

[Li22b] Trey Li. "Jacobi Symbol Parity Checking Algorithm for Subset Product". 2nd paper of the series. 2022, October 2.

[Li22c] Trey Li. "Power Residue Symbol Order Detecting Algorithm for Subset Product over Algebraic Integers". 3rd paper of the series. 2022, October 3.

[Li22d] Trey Li. “Multiple Modular Unique Factorization Domain Subset Product with Errors”. 4th paper of the series. 2022, October 4.
[Li22e] Trey Li. “Post-Quantum Key Exchange from Subset Product with Errors”. 5th paper of the series. 2022, October 5.