POST-QUANTUM KEY EXCHANGE FROM SUBSET PRODUCT WITH ERRORS

TREY LI

ABSTRACT. We introduce a new direction for post-quantum key exchange based on the multiple modular subset product with errors problem.

1. INTRODUCTION

Existing ideas for post-quantum key exchange are mostly from lattice problems (e.g., [ADPS16; BCDMNNRS16; DXL12]) and supersingular isogeny problems (e.g., [CLMPR18][1]).

We propose a Diffie-Hellman analogy (one-round and non-interactive) that does not seem to belong to the above two branches.

Our scheme is based on the hardness of the multiple modular subset product with errors problem (M-MSPE), which is a concrete case of the multiple modular unique factorization domain subset product with errors problem (M-MUSPE) proposed in [Li22d] with the underlining unique factorization domain (UFD) the concrete ring of integers $\mathbb{Z}$.

2. HARD PROBLEM

The abstract M-MUSPE is defined in [Li22d]. Our key exchange uses the following concrete settings.

Setup

Let $\ell_1,\ldots,\ell_{2^n}$ be the first 2^n primes[2]; and $p_1,\ldots,p_n$ be the next n primes. Denote $L = \{\ell_1,\ldots,\ell_{2^n}\}$ and $P = \{p_1,\ldots,p_n\}$.[3]

Choose a safe prime q in $[\ell_{2^n}^{2n+1}, p_1^{n^2/8}]$ [4] (e.g., the smallest safe prime greater than $\ell_{2^n}^{2n+1}$).[5]

This is the 5th paper of the series. Previously: [Li22a; Li22b; Li22c; Li22d].

Date: October 5, 2022.

Email: treyquantum@gmail.com

[1]The other line of works based on [JF11] are recently broken by [CD22; MM22; Rob22].

[2]Potential improvement of efficiency of the key exchange scheme 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 key exchange.

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

[4]We expect that such a safe prime exists for $n \geq 17$. This is because by the counting function of Sophie Germain primes $\pi_{SG}(m) \sim 2cm/(\log m)^2$ [Sho09], where $2c \approx 1.32032$ is a constant, we expect that a safe prime exists in every $(\log m)^2/2c$ integers. And the width of the interval $[\ell_{2^n}^{2n+1}, p_1^{n^2/8}] \sim [(n2^n)^{2n+1}, (n2^n)^{n^2/8}] = [m^{2n+1}, m^{n^2/8}]$ is $m^{n^2/8} - m^{2n+1} > m^2 > (\log m)^2/2c$, where $m := n2^n$ and $n \geq 17$. Note that 17 is the lower bound for n to be such that $n^2/8-(2n+1) > 0$; and whenever $n^2/8-(2n+1) > 0$ we have $n^2/8-(2n+1) > 1$ for integer n.

[5]In other words, q is greater than the product of any $2n+1$ primes in L; and smaller than the product of any $n/4$ integers a_j. The first condition is for success decoding of the key exchange scheme; and the second condition is to avoid the modulus q to be too large that can be ignored — Specifically, a uniform vector $x \leftarrow \{0,1\}^n$ is expected to have about $n/2$ entries to be 1 and thus if q is larger than the product of $n/2$ a_j's then the product $a_1^{x_1}\cdots a_n^{x_n} \pmod q = a_1^{x_1}\cdots a_n^{x_n}$ is not reduced at all modulo q; now we use $n/4$ rather than $n/2$ for a

Let D_a be the distribution that samples a vector $v \leftarrow \{0,1\}^n$ 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 $v = (v_0, \ldots, v_{n-1}) \leftarrow \{0,1\}^{\lceil \log(\ell_{2^n}) \rceil}$ uniformly at random until finding one such that the integer $e := \sum_{i=0}^{\lceil \log(\ell_{2^n}) \rceil - 1} (v_i \cdot 2^i)$ is a prime in L and outputs e.

Let D_x with respect to some $x \in \{0,1\}^n$ be the distribution that samples $a_1, \ldots, a_n \leftarrow D_a$ and $e \leftarrow D_e$, 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.

Problem

M-MSPE is given access to O_x, find x.[6]

3. Idea

The high level story of our key exchange is the following.

Before Key Exchange

- Alice and Bob: We use the same public matrix of base numbers $M := \{a_{i,j}\}_{n \times n}$; and the same public set of error primes $L = \{\ell_1, \ldots, \ell_{2^n}\}$.
- Alice: My static public key is an M-MSPE product sequence $S = (S_1, \ldots, S_n)$ with the base vectors the rows of M. My static private key is the corresponding M-MSPE secret $(s, u) \in \{0,1\}^n \times L^n$.
- Bob: My static public key is an M-MSPE product sequence $T = (T_1, \ldots, T_n)$ with the base vectors the columns of M. My static private key is the corresponding M-MSPE secret $(t, v) \in \{0,1\}^n \times L^n$.

Key Exchange

- Alice: I want to share a fresh M-MSPE secret $(x, e) \leftarrow \{0,1\}^n \times L^n$ with Bob. I first use this secret to compute an M-MSPE product sequence $(A_1, \ldots, A_n)$. I then use $x \in \{0,1\}^n$ to compute a composite MSPE product B by treating Bob's public M-MSPE product sequence T as the base vector. I send the M-MSPE product sequence $(A_1, \ldots, A_n, B)$ to Bob. If Bob has the secret (t, f) of the public key T he can recover (x, e).

---

more confident claim that this does not happen unless a random x has less than $n/4$ 1's, which happens with very low probability. For example, the probability that a 128-bit string has less than 32-bits of 1's is about 0.00000001.

However the safe prime that we want is quite large and not easy to find. We suggest to use a Mersenne prime instead. When $n = 256$, the 31[st] Mersenne prime $2^{216091} - 1$ is a proper choice for q.

[6] The differences between M-MSPE and M-MUSPE [Li22d] are: (1) M-MSPE is M-MUSPE over the concrete quotient ring $\mathbb{Z}_q^\times$; (2) the bases $a_1, \ldots, a_n$ in M-MSPE are square-free rather than uniform; (3) we use a single prime $e \in L$ for the error term $e^{\pm 1}$ of each MSPE instance, while MUSPE instances are allowed to use $t \geq 1$ many; and (4) we allow an arbitrary exponent in $\{-1, 1\}$ for the error e of each MSPE instance, while in the specific definition of MUSPE in [Li22d] the exponents of the error primes are -1 or 1 with equal probability.

Also, a further change is to use $x \in \mathbb{Z}_{q-1}^n$ instead of $x \in \{0,1\}^n$. In that case, the choice of q still works because when $x \in \mathbb{Z}_{q-1}^n$ the required upper bound of q is expected to be even greater than the previous required upper bound $p_1^{n^2/8}$ when $x \in \{0,1\}^n$.

- Bob (simultaneously): I want to share a fresh M-MSPE secret $(y,f) \in \{0,1\}^n \times L^n$ with Alice. I first use this secret to compute an M-MSPE product sequence $(C_1,\ldots,C_n)$. I then use $y \in \{0,1\}^n$ to compute a composite MSPE product D by treating Alice's public M-MSPE product vector S as the base vector. I send the M-MSPE product sequence $(C_1,\ldots,C_n,D)$ to Alice. If Alice has the secret (s,u) of the public key S she can recover (y,f).

Key Share

- Alice: I use my secret (s,u) to recover (y,f) and set $K_A = (x,y,e,f)$ as the shared secret.
- Bob: I use my secret (t,v) to recover (x,e) and set $K_B = (x,y,e,f)$ as the shared secret.

The key idea of the above story is that Alice and Bob respectively encode rows and columns of the same base matrix M into two M-MSPEs, called the "row M-MSPE" and the "column M-MSPE"; then the intersecting "block" of base numbers of the row M-MSPE and the column M-MSPE can be precisely cut off using the correct static private keys, and that the error terms will expose.

4. Scheme

The public parameters of the key exchange scheme are (n,q,M,L), where $M = \{a_{i,j}\}_{n\times n} \leftarrow D_a^{n\times n}$. The scheme is as follows.

Key Generation:

$$\begin{array}{cc}
\begin{bmatrix}
a_{1,1} & \cdots & a_{1,n} & u_1 \\
\vdots & & \vdots & \vdots \\
a_{n,1} & \ldots & a_{n,n} & u_n \\
v_1 & \ldots & v_n & 1
\end{bmatrix} & \begin{array}{c} \xrightarrow{s} \; S_1 \\ \vdots \\ \xrightarrow{s} \; S_n \\ \end{array} \\
\begin{array}{ccc} \downarrow t & \cdots & \downarrow t \\ T_1 & \cdots & T_n \end{array} &
\end{array}$$

- Alice: Sample static private key $(s,u) \leftarrow \{0,1\}^n \times D_e^n$. Compute an M-MSPE product sequence $S = (S_1,\ldots,S_n)$, where $S_i = a_{i,1}^{s_1} \cdots a_{i,n}^{s_n} \cdot u_i \pmod q$ for $i \in [n]$. Publish S as the public key.
- Bob: Sample static private key $(t,v) \leftarrow \{0,1\}^n \times D_e^n$. Compute an M-MSPE product sequence $T = (T_1,\ldots,T_n)$, where $T_j = a_{1,j}^{t_1} \cdots a_{n,j}^{t_n} \cdot (1/v_j) \pmod q$ for $j \in [n]$. Publish T as the public key.

Key Exchange:

$$\begin{array}{cc}
\begin{bmatrix}
a_{1,1} & \cdots & a_{1,n} & e_1 \\
\vdots & & \vdots & \vdots \\
a_{n,1} & \ldots & a_{n,n} & e_n \\
f_1 & \ldots & f_n & 1
\end{bmatrix} & \begin{array}{c} \xrightarrow{x} \; A_1 \\ \vdots \\ \xrightarrow{x} \; A_n \\ \end{array} \\
\begin{array}{ccc} \downarrow y & \cdots & \downarrow y \\ C_1 & \cdots & C_n \end{array} &
\end{array}$$

- Alice: Sample ephemeral key $(x,e) \leftarrow \{0,1\}^n \times D_L^{n+1}$. Compute an M-MSPE product sequence $A=(A_1,\ldots,A_n)$, where $A_j = a_{i,1}^{x_1}\cdots a_{i,n}^{x_n}\cdot e_i \pmod q$ for $i\in[n]$. Compute an MSPE product $B = T_1^{x_1}\cdots T_n^{x_n}\cdot(1/e_{n+1}) \pmod q$. Send (A,B) to Bob.
- Bob (simultaneously): Sample ephemeral key $(y,f) \leftarrow \{0,1\}^n \times D_L^{n+1}$. Compute an M-MSPE product sequence $C=(C_1,\ldots,C_n)$, where $C_j = a_{1,j}^{y_1}\cdots a_{n,j}^{y_n}\cdot(1/f_j)$ $\pmod q$, for $j\in[n]$. Compute an MSPE product $D = S_1^{y_1}\cdots S_n^{y_n}\cdot f_{n+1} \pmod q$. Send (C,D) to Alice.

Key Share:

- Alice: Compute $E = D/C_1^{s_1}\cdots C_n^{s_n} \pmod q$. Compute $y'\in\{0,1\}^n$ such that $y'_j=1$ if and only if $u_j|E$, for $j\in[n]$. Compute $f'\in L^{n+1}$ such that $f'_j = a_{1,j}^{y'_1}\cdots a_{n,j}^{y'_n}/C_j$ $\pmod q$, for $j\in[n]$; and $f'_{n+1} = D/S_1^{y'_1}\cdots S_n^{y'_n}$. Set the shared secret to be $K_A = (x,y',e,f')$.
- Bob: Compute $F := A_1^{t_1}\cdots A_n^{t_n}/B \pmod q$. Compute $x'\in\{0,1\}^n$ such that $x'_i=1$ if and only if $v_i|F$, for $i\in[n]$. Compute $e'\in L^{n+1}$ such that $e'_i = A_j/a_{i,1}^{x'_1}\cdots a_{i,n}^{x'_n}$ $\pmod q$ for $i\in[n]$; and $e'_{n+1} = T_1^{x'_1}\cdots T_n^{x'_n}/B$. Set the shared secret to be $K_B = (x',y,e',f)$.

5. Correctness

Theorem 1. $K_A = K_B$ with overwhelming probability.

Proof. Note that L is exponentially large and we only sample $4n+2$ (i.e. linearly many) error primes from L (they are the u_i's, v_i's, e_i's and the f_i's in the scheme). 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

$$\begin{aligned} E &= \left(u_1^{y_1}\cdots u_n^{y_n}\right)\cdot\left({f_1}^{s_1}\cdots {f_n}^{s_n}\right)\cdot f_{n+1} \pmod q \\ &= \left(u_1^{y_1}\cdots u_n^{y_n}\right)\cdot\left({f_1}^{s_1}\cdots {f_n}^{s_n}\right)\cdot f_{n+1} \end{aligned}$$

and

$$\begin{aligned} F &= \left({e_1}^{t_1}\cdots {e_n}^{t_n}\right)\cdot\left(v_1^{x_1}\cdots v_n^{x_n}\right)\cdot e_{n+1} \pmod q \\ &= \left({e_1}^{t_1}\cdots {e_n}^{t_n}\right)\cdot\left(v_1^{x_1}\cdots v_n^{x_n}\right)\cdot e_{n+1}. \end{aligned}$$

Therefore if all error primes in the scheme are different, then $y_j=1$ if and only if $u_j|E$; and $x_i=1$ if and only if $v_i|F$. Then $y'=y$ and $x'=x$ (and thus $f'_j=f_j$ and $e'_i=e_i$ for all $i,j\in[n+1]$). Then $K_A=K_B$. Hence $K_A=K_B$ with overwhelming probability p. □

6. Efficiency

Theorem 2. The time complexities of key generation and key exchange are both $O(n^4)$.

Proof. The complexities mainly come from modular multiplications. Note that $q \gtrsim \ell_{2^n}^{2n+1} \gtrsim (n2^n)^{2n+1} = 2^{O(n^2)}$. Hence the complexity of a single modular multiplication is $O(\log_2 q) = O(n^2)$. There are $O(n^2)$ modular multiplications in both key generation and key exchange (including key share). Hence the time complexities of key generation and key exchange are both $O(n^4)$. □

7. SECURITY

The differences between the problem that we use to construct our key exchange scheme and the M-MSPE in Section 2 are: (1) instead of giving unlimited access to the oracle O_x, the scheme only gives $n+1$ MSPE instances for each secret; (2) one of the instances is a special instance whose base numbers are themselves MSPE products rather than regular bases sampled from D_a; and (3) the scheme reuses the base matrix M for different uniformly sampled secrets x in different executions of the scheme[7]. We denote this M-MSPE as M-MSPE$_{KE}$.

We show securities against private key recovery and shared key recovery assuming the hardness of MSPE$_{KE}$.

THEOREM 3. If M-MSPE$_{KE}$ (with regular bases only, i.e., bases sampled from D_a) is hard, then there does not exist a probabilistic polynomial time adversary that finds the static private keys (s,u) or (t,v) from the transcripts S,T,A,B,C,D and the public parameters n,q,M,L.

Proof. Suppose for contradiction that such an adversary $\mathcal{A}$ exists. We use it to solve M-MSPE$_{KE}$. Given an M-MSPE$_{KE}$ (M,S), where $S=(S_1,\ldots,S_n)$ is the MSPE product sequence. Treat S as the public key of Alice in the scheme. Compute all the rest of the scheme to have T,A,B,C,D. Note that this can be done because T, A, B and C are independent of S; and D only relies on the public numbers $S_1,\ldots,S_n$. Then call $\mathcal{A}$ to find the secret (s,e), where s is the solution to the target M-MSPE$_{KE}$ (M,S). □

THEOREM 4. If M-MSPE$_{KE}$ is hard, then there does not exist a probabilistic polynomial time adversary that finds the shared key (x,y,e,f) from the transcripts S,T,A,B,C,D and the public parameters n,q,M,L.

Proof. Suppose for contradiction that such an adversary $\mathcal{A}$ exists. We use it to solve M-MSPE$_{KE}$. Given an M-MSPE$_{KE}$ $((M,T),(A,B))$, where $(A,B)=(A_1,\ldots,A_n,B)$ is the MSPE product sequence, and $T=(T_1,\ldots,T_n)$ is the base vector of B with the bases $T_1,\ldots,T_n \in \mathbb{Z}_q^\times$ themselves MSPE products. Now treat (A,B) as Alice's message in the key exchange scheme. We compute the rest of the scheme to have S,T,C,D. This can be done because S,C,D are independent of (A,B); and T is given. Then call $\mathcal{A}$ with (S,T,A,B,C,D) to solve for (x,y,e,f) and x is the solution to the target M-MSPE$_{KE}$. □

REFERENCES

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. "Post-quantum Key {ExchangeA} New Hope". In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 327–343.

[7] One way to avoid reusing the base matrix M is to change the scheme to be an interactive key exchange scheme by putting Alice and Bob's own ephemeral base matrices M_A and M_B into their key exchange messages respectively; and cancel the use of public keys. Then the security relies on a weaker assumption than M-MSPE$_{KE}$. The tradeoffs are lower efficiency, larger message sizes, and interactivity.

[BCDMNNRS16] Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. "Frodo: Take off the ring! practical, quantum-secure key exchange from LWE". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 1006–1018.

[CD22] Wouter Castryck and Thomas Decru. *An efficient key recovery attack on SIDH (preliminary version)*. Cryptology ePrint Archive, Paper 2022/975. https://eprint.iacr.org/2022/975. 2022. URL: https://eprint.iacr.org/2022/975.

[CLMPR18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. "CSIDH: an efficient post-quantum commutative group action". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2018, pp. 395–427.

[DXL12] Jintai Ding, Xiang Xie, and Xiaodong Lin. "A simple provably secure key exchange scheme based on the learning with errors problem". In: *Cryptology ePrint Archive* (2012).

[JF11] David Jao and Luca De Feo. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *International Workshop on Post-Quantum Cryptography*. Springer. 2011, pp. 19–34.

[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.

[MM22] Luciano Maino and Chloe Martindale. *An attack on SIDH with arbitrary starting curve*. Cryptology ePrint Archive, Paper 2022/1026. https://eprint.iacr.org/2022/1026. 2022. URL: https://eprint.iacr.org/2022/1026.

[Rob22] Damien Robert. *Breaking SIDH in polynomial time*. Cryptology ePrint Archive, Paper 2022/1038. https://eprint.iacr.org/2022/1038. 2022. URL: https://eprint.iacr.org/2022/1038.

[Sho09] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009.