Multidimentional ModDiv public key exchange protocol

Samir Bouftass

E-mail : crypticator@gmail.com

July 20, 2021

Abstract

This paper presents Multidimentional ModDiv public key exchange protocol which security is based on the hardness of an LWR problem instance consisting on finding a secret vector **X** in $\mathbb{Z}_q^n$ knowing vectors **A** and **B** respectively in $\mathbb{Z}_p^m$ and $\mathbb{Z}_{p-q}^{m-n}$, where elements of vector **B** are defined as follows : $B(i) = (\sum_{j=1}^{j=n} A(i+n-j) \times X(j))\ Mod(P) Div(Q)$.
Mod is integer modulo, Div is integer division, P and Q are known positive integers which sizes in bits are respectively p and q which satisfy $p > 2 \times q$. m and n satisfy $m > 2 \times n$.

Keywords : Diffie Hellman key exchange protocol, Post Quantum cryptography, Lattice based cryptography, Closest vector problem, Learn with rounding problem.

1 Introduction :

Since its invention by Withfield Diffie and Martin Hellman [1], public key cryptography has imposed itself as the necessary and indispensable building block of every IT security architecture.
In the last decades, it has been proven that public key cryptosystems based on number theory problems are not immune againt quantum computing attacks [2], urging the necessity of inventing new algorithms not based on classical problems namely factoring, discret log over multiplicative groups or elliptic curves.

In [3] is presented a one dimentional ModDiv public key exchange protocol which security have been shown in Barcau et all [4] to be related to CVP problem.
Y Zang [5] have proven that one dimentional ModDiv security problem can be reduced to a CVP problem in 2 dimentional lattice.

Present paper proposes Multidimentional ModDiv public key exchange protocol which security is based on an instance of learn with rounding problem [6], we defined as ModDiv learn with rounding problem.

In section 2, we describe one dimentional ModDiv public key exchange protocol.
In section 3, we describe Multidimentional ModDiv public key exchange protocol.

2 One dimentional ModDiv public key exchange protocol :

2.1 Notations:

Div : Integer division, Mod : Integer modulo.

$Imdv_{(P,Q)}(A) = A\ Mod(P)Div(Q)$.

polyDiv : Polynomial division, polyMod : Polynomial modulo.

$Pmdv_{(m,n)}(\mathbf{P}) = \mathbf{P}\ polyMod(x^m)polyDiv(x^n)$.

$\parallel A \parallel$: size in bits of A.

P(i) : ith coefficient of polynomial **P**.

2.1.1 Public parameters :

Integer A, positive integers p, q, P and Q where $p > 2 \times q$.

A is pseudorandom and $\parallel A \parallel = \parallel P \parallel = p$.

2.1.2 Private Computations :

- Alice generates pseudorandomly a number X where $\parallel X \parallel = q$.

- Calculates $U = Imdv_{(P,Q)}(A \times X)$.

- Bob generates pseudorandomly a number Y where $\parallel Y \parallel = q$.

- Calculates $V = Imdv_{(P,Q)}(A \times Y)$.

2.1.3 Publicly exchanged values :

- Alice sends U to Bob.

- Bob sends V to Alice.

2.1.4 Further Private Computations :

- Alice calculates $Wa = Imdv_{(P,2^{2\times q})}(X \times Q \times V)$.

- Bob calculates $Wb = Imdv_{(P,2^{2\times q})}(Y \times Q \times U)$.

Bob and Alice know that :

$Wa = Wb$ or $\mid Wa - Wb \mid = 1$.

2.2 Proof of correctness :

Protocol described above is correct according to the following theorem :

Theorem 1. *Let A, X, Y, P and Q be integers and p, q positive integers where* $p > 2 \times q$.

$\parallel A \parallel = \parallel P \parallel = p$, $\parallel X \parallel = \parallel Y \parallel = \parallel Q \parallel = q$.

$Wa = Imdv_{(P,2^{2\times q})}(X \times Q \times Imdv_{(P,Q)}(A \times Y))$.

$Wb = Imdv_{(P,2^{2\times q})}(Y \times Q \times Imdv_{(P,Q)}(A \times X))$.

There is two ways that Wa *and* Wb *could be related :*

1 - $Wa = Wb$.

2 - $\mid Wa - Wb \mid = 1$.

Proof :

We know that for every positive integers A and Q where $A > Q$, $A\ Div(Q) \times Q = A - A\ Mod(Q)$.

$X \times Q \times Imdv_{(P,Q)}(A \times Y) = X \times (A \times Y)Mod(P)Div(Q) \times Q$.

$X \times Q \times Imdv_{(P,Q)}(A \times Y) = X \times ((A \times Y)Mod(P) - (A \times Y)Mod(P)Mod(Q))$.

$(X \times Q \times Imdv_{(P,Q)}(A \times Y))Mod(P) = ((X \times A \times Y) - (X \times ((A \times Y)Mod(P)Mod(Q)))Mod(P)$.

$(X \times Q \times Imdv_{(P,Q)}(A \times Y))Mod(P) = ((X \times A \times Y)Mod(P) - (X \times ((A \times Y)Mod(P)Mod(Q))Mod(P))Mod(P)$.

$(X \times Q \times Imdv_{(P,Q)}(A \times Y))Mod(P) = ((X \times A \times Y)Mod(P) - (X \times ((A \times Y)Mod(P)Mod(Q)))$ (1).

Similarily, we can prove :

$(Y \times Q \times Imdv_{(P,Q)}(A \times X))Mod(P) = ((Y \times A \times X)Mod(P) - (Y \times ((A \times X)Mod(P)Mod(Q)))$ (2).

Note $\parallel (X \times ((A \times Y)Mod(P)Mod(Q)) \parallel = \parallel (Y \times ((A \times X)Mod(P)Mod(Q)) \parallel = 2 \times q$.

They mask at least $2 \times q$ least significant bits of $((X \times A \times Y)Mod(P)$.

If $2 \times q$ th borrow of substraction (1) equals 0, we have :

$Wa = (X \times Q \times Imdv_{(P,Q)}(A \times Y))Mod(P)Div(2^{2 \times q}) = ((X \times A \times Y)Mod(P)Div(2^{2 \times q})$.

If $2 \times q$ th borrow of substraction (1) equals 1, we have :

$Wa = (X \times Q \times Imdv_{(P,Q)}(A \times Y))Mod(P)Div(2^{2 \times q}) = ((X \times A \times Y)Mod(P)Div(2^{2 \times q}) - 1$.

Wa can have then two values :

$((X \times A \times Y)Mod(P)Div(2^{2 \times q})$ or $((X \times A \times Y)Mod(P)Div(2^{2 \times q}) - 1$.

Likewise we can prove that Wb can have the same values :

$((Y \times A \times X)Mod(P)Div(2^{2 \times q})$ or $((Y \times A \times X)Mod(P)Div(2^{2 \times q}) - 1$.

In other words Wa could be equal to Wb , $Wb - 1$ or $Wb + 1$.

Concluding thus the proof of theorem 1.

Observe, exchanged key size maximum is equal to $p - (2 \times q)$.

To get the same number, Alice or Bob have to divide their obtained values Wa and Wb by 2^{r+1} where r is the number of trailing zeros or ones of Wa or Wb (See Figure 1).
Indeed if $\mid Wa - Wb \mid = 1$, and Wa have r trailing ones, Wb would have r trailing zeros, the converse is also true.

r can be calculated by the following algorithm :

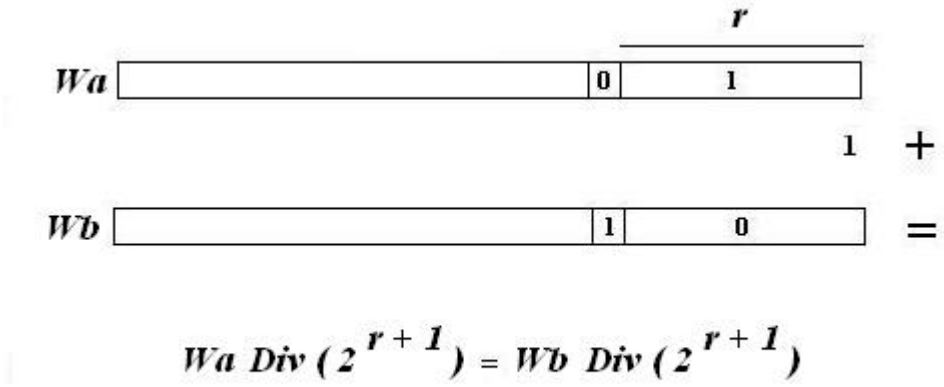
Algorithm 1.

Inputs : W is Wa or Wb
Output : r

---

1 : r := 0
2 : v := W Mod(2)
3 : W := W Div(2)
4 : **While** (v = W Mod(2))
5 : r := r + 1
6 : W := W Div(2)
7 : **End While**

Figure 1:



Alice and Bob can't know beforehand exact exchanged key size, but if Wa and Wb are uniformly distributed, Pr [exchanged key size = $p - (2 \times q) - i$] = $1 - 2^{-(i+1)}$.

2.3 Multidimentional ModDiv public key exchange protocol:

2.3.1 Public parameters :

Integers m, n, p, q, P, Q where $m > (2 \times n) + 1$, $p > (2 \times q) + log_2(n)$, $\| P \|$ = p and $\| Q \|$ = q.

An m degree polynomial **A**, which coefficients size in bits equals p.

P, Q and **A** coefficients are pseudorandomly generated.

2.3.2 Private Computations :

- Alice chooses an n degree polynomial **X**, which coefficient size in bits equals q. **X** coefficients are pseudorandomly generated.

- Alice calculates $\mathbf{U} = Pmdv_{(m,n)}(\mathbf{A} \times \mathbf{X})$.

- For each **U** coefficient $U(i)$:

  – She calculates $U1(i) = U(i)\ Mod(P)Div(Q)$, getting thus polynomial **U1**.

- Bob chooses an n degree polynomial **Y**, which coefficient size in bits equals q. **Y** coefficients are pseudorandomly generated.

- Bob calculates $\mathbf{V} = Pmdv_{(m,n)}(\mathbf{A} \times \mathbf{Y})$.

- For each **V** coefficient $V(i)$:

  – He calculates $V1(i) = V(i)\ Mod(P)Div(Q)$, getting thus polynomial **V1**.

2.3.3 Publicly exchanged values :

- Alice sends **U1** coefficients to Bob.

- Bob sends **V1** coefficients to Alice.

2.3.4 Further Private Computations :

- Alice calculates polynomial $\mathbf{Wa} = Pmdv_{(m,2\times n)}(x^n \times \mathbf{X} \times \mathbf{V1})$.

- For each **Wa** coefficient $Wa(i)$:

  – Using algorithm 1, she calculates r(i), the number of $Wa(i)$ $Div(2^{log_2(n)})$'s trailing bits with same value.

  – She calculates $Wa1(i) = (Q \times Wa(i))\ Mod(P)\ Div(2^{(2\times q)+r(i)+1+log_2(n)})$.

- Bob calculates polynomial $\mathbf{Wb} = Pmdv_{(m,2\times n)}(x^n \times \mathbf{X} \times \mathbf{U1})$.

- For each **Wb** coefficient $Wb(i)$:

  – Using algorithm 1, he calculates r(i), the number of $Wb(i)$ $Div(2^{log_2(n)})$'s trailing bits with same value.

  – He calculates $Wb1(i) =(Q \times Wb(i))\ Mod(P)\ Div(2^{(2\times q)+r(i)+1+log_2(n)})$.

Shared key sise in bits they could get equals :

$\sum_{i=1}^{i=m-(2\times n)}(p - ((2 \times q) + r(i) + 1 + log_2(n))$.

2.4 Proof of Correction :

Proof of correction straitly follows from :

- Polynomial multiplication commutativity.

- Theorem 1.

- $Pmdv_{(m,2\times n)}(\mathbf{Y} \times x^n \times Pmdv_{(m,n)}(\mathbf{A} \times \mathbf{X})) = Pmdv_{(m,2\times n)}(\mathbf{X} \times x^n \times Pmdv_{(m,n)}(\mathbf{A} \times \mathbf{Y}))$. because basically polynomial multiplication is integer mutiplication without carry propagation, this equality can be proved similarily as theorem 1.

- Size in bits of the result of adding n, s bits numbers, is $s \times log_2(n)$.

2.5 Security :

To attack proposed public key exchange protocol, adversary Eve knows polynomials **A**, **U1**, **V1** and parameters p, q, P, Q, m, n satistifying the following conditions :

- **A** is of degree m.

- **A** coefficients size in bits, equals p.

- **U1** and **V1** are of degree m-n-1.

- **U1** and **V1** coefficients size in bits, equals $p - q$.

- $m > (2 \times n) + 1$ and $p > (2 \times q) + log_2(n)$.

She equaly knows that there exists polynomials **U**, **V**, **X** and **Y** Satisfying :

- **X** and **Y** are of degree n.

- **X** and **Y** coefficients size in bits, equals q.

- **U** and **V** are of degree m-n-1.

- $\mathbf{U} = (\mathbf{X} \times \mathbf{A})\ polyMod(x^m) polyDiv(x^n)$.

- $\mathbf{V} = (\mathbf{Y} \times \mathbf{A})\ polyMod(x^m) polyDiv(x^n)$.

Coefficients of **U** and **V** are respectively related to those of **U1** and **V1** by following equations :

- $U1(i) = U(i)\ Mod(P) Div(Q)$.

- $V1(i) = V(i)\ Mod(P) Div(Q)$.

Meaning to know secret polynomials **X** and **Y**, Eve have to solve the following equations set :

$U1(i) = (\ \sum_{j=1}^{j=n+1} A(i+n+1-j) \times X(j))\ Mod(P) Div(Q)$.

$V1(i) = (\ \sum_{j=1}^{j=n+1} A(i+n+1-j) \times Y(j))\ Mod(P) Div(Q)$.

Where $1 \leq \mathrm{i} \leq m - n$.

2.5.1 ModDiv learn with rounding problem:

Basically underlying security problem of proposed public key exchange protocol is an instance of learn with rounding problem first proposed by Banerjee et all [6], which we would define as follows :

Definition 1. *ModDiv learn with rounding problem is an instance of learn with rounding problem caracterized by the following features :*

- Rounding is done from $\mathbb{Z}_P$ to $\mathbb{Z}_{PDiv(Q)}$.

*- Public vectors are not random, they reflect polynomial mutiplication algebraic structure : If vector **A1** [A(1), A(2) .... A(m+1)] is given as public, vector **A2** [R(1),A(1),.... A(m)]. is also given as public. Element R(1) is pseudorandomly generated.*

- Secret vector elements size is the same as derandomized error vector elements induced by rounding.

Learn with rounding problem is considered to be at least as hard as Learn with error problem, and was used recently to construct public key cryptosystems like Saber [7], one of NIST third round finalists. Corresponding lattices to ModDiv learn with rounding problem, are a generalization of cyclic lattices where recycled elements, if we may say, are pseudorandomly generated unlike in Ideal lattices where they are deterministically generated, which is a proof that CVP problem equivalent to ModDiv learn with rounding problem might be at least as hard as CVP problem in Ideal lattices.

2.5.2 Cyclic ModDiv learn with rounding problem:

For efficiency as regards computation and public key size, public polynomial **A** coefficients could be choosen such as $A(i) = A(i - n - 1)$ where $n + 2 \leq i \leq m + 1$.

In this case proposed key exchange protocol would be based on a instance of ModDiv learn with rounding problem, defined as follows :

Definition 2. *Cyclic ModDiv learn with rounding problem is an instance of ModDiv learn with rounding problem caracterized by :*

*- If vector **A1** [A(1), A(2) .... A(m+1)] is given as public, vector **A2** [A(m+1),A(1),.... A(m)] is also given as public.*

Observe the lattice corresponding to this instance of learn with rounding problem is cyclic meaning in order for it, to be hard as assumed [8], public vector elements or public polynomial coefficients should satisfiy this condition : $\sum_{j=1}^{j=m+1} A(i) = 0$.

3 Conclusion :

In this paper we have presented Multidimentional ModDiv public key exchange protocol, the multidimentional version of public key exchange protocol presented in [3].

We have shown that its security is based on an instance of learn with rounding problem, we defined as ModDiv learn with rounding problem. We also introduced a cyclic variant of this problem and determine a condition under which it is assumed to be hard [8], key exchange protocol based on this varient is efficient as regards computation and public key size.

One can construct public key encryption and digital signature ElGamal schemes based on presented Key exchange protocol, it is also quite possible to device hash functions and pseudo random numbers generators, based on introduced problems.

References

[1] Whitfield Diffie, Martin E.Hellman. *New Directions in cryptography, IEEE Trans. on Info. Theory, Vol. IT-22, Nov. 1976 (1976)*

[2] Daniel J Bernstein, Johannes Buchmann, Erik Dahman. *Post-Quantum Cryptography*, (2009), Springer Verlag , Berlin Heidelberg .

[3] A Azhari, S Bouftass : On a new fast public key cryptosystem. *IACR Cryptology eprint Archive 2014:946(2014).*

[4] Mugurel Barcau, Vicentiu Pasol, Cezar Plesca, and Mihai Togan : On a Key Exchange Protocol *SECITC 2017.*

[5] Y Zhang : A practical attack to Bouftasss crypto system *arXiv:1605.00987 [cs.CR].*

[6] Abhishek Banerjee, Chris Peikert, Alon Rosen : Pseudorandom Functions and Lattices *IACR Cryptology eprint Archive 2011:401(2011).*

[7] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren Saber: Module-LWR based key exchange,CPA-secure encryption and CCA-secure KEM *IACR Cryptology eprint Archive 2018:230(2018).*

[8] Daniele Micciancio : Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions *Springer computational complexity: 16, pages 365411(2007).*