

# Multidimensional ModDiv public key exchange protocol

Samir Bouftass

E-mail : crypticator@gmail.com

June 1, 2021

## Abstract

This paper presents Multidimensional 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  $\mathbb{Z}_r^n$  knowing vectors  $A$  and  $B$  respectively in  $\mathbb{Z}_s^m$  and  $\mathbb{Z}_t^l$ , where elements of vector  $B$  are defined as follows :  $B(i) = (\sum_{j=1}^{j=n} A(i+j) \times X(j)) \text{ Mod}(2^p) \text{ Div}(2^q)$ . Mod is integer modulo operation, Div is integer division operation,  $p$  and  $q$  are known positive integers satisfying  $p > 2 \times q$ . Size in bits of  $s$  equals  $p$ , size in bits of  $r$  equals  $q$ , and size in bits of  $t$  equals  $p - q$ ,  $m > 2 \times n$  and  $l = m - n$ .

**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 against quantum computing attacks [2], urging the necessity of inventing new algorithms not based on classical problems namely factoring, discrete log over multiplicative groups or elliptic curves.

In [3] is presented a one dimensional ModDiv public key exchange protocol which security have been shown in Barcau et al [4] to be based on CVP problem.

Y Zang [5] have proven that one dimensional ModDiv security problem can be reduced to a CVP problem in 2 dimensional lattice.

Present paper proposes Multidimensional ModDiv public key exchange protocol which security is based on an instance of learn with rounding problem first proposed by Banerjee et al [6] .

In section 2, we describe one dimensional ModDiv public key exchange protocol and provide a proof of its correctness .

In section 3, we describe Multidimensional ModDiv public key exchange protocol.

## 2 One dimensional ModDiv public key exchange protocol :

### 2.1 Notations:

$mdv2_{(p,q)}(A) = A \text{ Mod}(2^p) \text{ Div}(2^q)$ . (  $A$  being an integer, Mod modulo operation, and Div integer division ).

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

#### 2.1.1 Public parameters :

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

$A$  is pseudorandom and  $\| A \| = p$ .

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

#### 2.1.2 Private Computations :

- Alice generates pseudorandomly a  $q$  bit number  $X$ , and calculates  $U = mdv2_{(p,q)}(A \times X)$ .

- Bob generates pseudorandomly a  $q$  bit number  $Y$ , and calculates  $V = mdv2_{(p,q)}(A \times Y)$ .

#### 2.1.3 Publicly exchanged values :

- Alice sends  $U$  to Bob.

- Bob sends  $V$  to Alice.

#### 2.1.4 Further Private Computations :

- Alice calculates  $Wa = mdv2_{(p-q,q)}(X \times V)$ .

- Bob calculates  $Wb = mdv2_{(p-q,q)}(Y \times U)$ .

Bob and Alice know that :

$$Wa = Wb \text{ or } | Wa - Wb | = 1.$$

## 2.2 Proof of correctness :

**Lemma 1.** *Let  $A$  be an integer,  $p$  and  $q$  positive integers.*

*If  $(p > q)$ , then  $mdv2_{(p,q)}(A \times 2^q) = A \text{ mod}(2^{p-q})$ .*

Proof :

$$mdv2_{(p,q)}(A \times 2^q) = (A \times 2^q) \text{ Mod}(2^p) \text{ Div}(2^q).$$

Observe least significant  $q$  bits of  $N = (A \times 2^q) \text{ Mod}(2^p)$  are zeros whereas its most significant  $p - q$  bits are the least significant  $p - q$  bits of  $A$ , dividing then  $N$  by  $2^q$  implies :

$$(A \times 2^q) \text{ Mod}(2^p) \text{ Div}(2^q) = A \text{ Mod}(2^{p-q}) = mdv2_{(p,q)}(A \times 2^q) .$$

**Theorem 1.** *Let  $A, X, Y, p$  and  $q$  be integers where  $p > q$ ,  $\| A \| = p$ ,  $\| X \| = \| Y \| = q$ .*

$$Wa = mdv2_{(p-q,q)}(X \times mdv_{(p,q)}(A \times Y)).$$

$$Wb = mdv2_{(p-q,q)}(Y \times mdv_{(p,q)}(A \times X)).$$

*There is two possibilities :*

$$1 - Wa = Wb.$$

$$2 - |Wa - Wb| = 1.$$

Proof :

Let  $H1$  and  $H2$  be integers such as :

$$U_1 = mdv2_{(p,q)}(A \times X) \times 2^q = (A \times X - H1) \text{ Mod}(2^p).$$

$$V_1 = mdv2_{(p,q)}(A \times Y) \times 2^q = (A \times Y - H2) \text{ Mod}(2^p).$$

The fact that the least significant  $q$  bits of  $U_1$  and  $V_1$  are zeroes, implies  $\| H_1 \| = \| H_2 \| = q$ .

Let's calculate :

$$Wa1 = (X \times V_1) \text{ Mod}(2^p) = ((X \times Y \times A) - (X \times H_2)) \text{ Mod}(2^p) \quad (1)$$

$$Wb1 = (Y \times U_1) \text{ Mod}(2^p) = ((Y \times X \times A) - (Y \times H_1)) \text{ Mod}(2^p) \quad (2)$$

$\| X \| = \| Y \| = \| H_1 \| = \| H_2 \| = q$  implies  $\| X \times H_2 \| = \| Y \times H_1 \| = 2 \times q$ , we have then :

$$Wa1 \text{ Div}(2^{2 \times q}) = (X \times Y \times A) \text{ Mod}(2^p) \text{ Div}(2^{2 \times q}) - E_a$$

$$Wb1 \text{ Div}(2^{2 \times q}) = (Y \times X \times A) \text{ Mod}(2^p) \text{ Div}(2^{2 \times q}) - E_b$$

where  $E_a$  and  $E_b$  are respectively the  $2 \times q$ 'th borrows of binary subtractions (1) and (2) .

$E_a$  and  $E_b$  being bits, they can have then for values 0 or 1, implying :

$$\text{if } E_a = E_b \text{ we have } Wa1 \text{ Div}(2^{2 \times q}) = Wb1 \text{ Div}(2^{2 \times q}).$$

$$\text{if } | E_a - E_b | = 1 \text{ we have } | Wa1 \text{ Div}(2^{2 \times q}) - Wb1 \text{ Div}(2^{2 \times q}) | = 1.$$

Now we'll show that :

$$Wa = Wa1 \text{ Div}(2^{2 \times q}) \text{ and } Wb = Wb1 \text{ Div}(2^{2 \times q}).$$

ending thus theorem's proof .

$$Wa1 = (X \times V_1) \text{ Mod}(2^p) = (X \times m dv_{(p,q)}(A \times Y) \times 2^q) \text{ Mod}(2^p).$$

$$Wa1 \text{ Div}(2^q) = (X \times V_1) \text{ Mod}(2^p) \text{ Div}(2^q) = (X \times m dv_{(p,q)}(A \times Y) \times 2^q) \text{ Mod}(2^p) \text{ Div}(2^q).$$

Applying Lemma 1, we get :

$$Wa1 \text{ div}(2^q) = (X \times V_1) \text{ Mod}(2^p) \text{ Div}(2^q) = (X \times m dv_{(p,q)}(A \times Y)) \text{ Mod}(2^{p-q}).$$

$$Wa1 \text{ Div}(2^{2 \times q}) = (X \times m dv_{(p,q)}(A \times Y)) \text{ Mod}(2^{p-q}) \text{ Div}(2^q).$$

$$Wa1 \text{ Div}(2^{2 \times q}) = m dv_{(p-q,q)}(X \times m dv_{(p,q)}(A \times Y)).$$

$$Wa1 \text{ Div}(2^{2 \times q}) = Wa.$$

Likewise we can prove :

$$Wb1 \text{ Div}(2^{2 \times q}) = Wb.$$

Observe,  $\text{Max}(S) = \| Wa \| = \| Wb \| = p - (2 \times q)$ .



### 3 Multidimensional ModDiv public key exchange protocol :

#### 3.1 Notations:

polyDiv : Polynomial division operation, polyMod : Polynomial modulo operation.

$$pmdv_{(m,n)}(\mathbf{A}) = (\mathbf{A}) \text{ polyMod}(x^m) \text{ polyDiv}(x^n).$$

##### 3.1.1 Public parameters :

Integers  $m, n, p, q$  where  $m > (2 \times n)$  and  $p > (2 \times q) + \log_2(n)$ .

An  $m$  degree polynomial  $\mathbf{A}$ , which coefficients sizes in bits equals  $\| 2^p \|$ .

$\mathbf{A}$  coefficients are pseudorandomly generated.

##### 3.1.2 Private Computations :

- Alice chooses an  $n$  degree polynomial  $\mathbf{X}$ , which coefficient sizes in bits equals  $\| 2^q \|$ .  $\mathbf{X}$  coefficients are pseudorandomly generated.

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

- For each  $\mathbf{U}$  coefficient  $U(i)$ , she calculates  $U1(i) = mdv2_{(p,q)}(U(i))$ , getting thus polynomial  $\mathbf{U1}$ .

- Bob chooses an  $n$  degree polynomial  $\mathbf{Y}$ , which coefficient sizes in bits equal  $\| 2^q \|$ .  $\mathbf{Y}$  coefficients are pseudorandomly generated.

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

- For each  $\mathbf{V}$  coefficient  $V(i)$ , he calculates  $V1(i) = mdv2_{(p,q)}(V(i))$ , getting thus polynomial  $\mathbf{V1}$ .

##### 3.1.3 Publicly exchanged values :

- Alice sends  $\mathbf{U1}$  coefficients to Bob.

- Bob sends  $\mathbf{V1}$  coefficients to Alice.

### 3.1.4 Further Private Computations :

- Alice calculates polynomial  $\mathbf{Wa} = pmdv_{(m-n,n)}(\mathbf{X} \times \mathbf{V1})$ .
- For each  $\mathbf{Wa}$  coefficient  $Wa(i)$  :
  - She calculates  $r(i)$ ,  $Wa(i)$  consecutive least significant bits with the same value, number + 1.
  - She calculates  $Wa1(i) = mdv2_{(p-q,q)}(Wa(i)) Div(2^{r(i)+\log_2(n)})$
- Bob calculates polynomial  $\mathbf{Wb} = pmdv_{(m-n,n)}(\mathbf{X} \times \mathbf{U1})$ .
- For each  $\mathbf{Wb}$  coefficient  $Wb(i)$  :
  - He calculates  $r(i)$ ,  $Wb(i)$  consecutive least significant bits with the same value, number + 1.
  - He calculates  $Wb1(i) = mdv2_{(p-q,q)}(Wb(i)) Div(2^{r(i)+\log_2(n)})$

Bob and Alice know that for each coefficients  $Wa1(i)$  and  $Wb1(i)$  :

Shared key size in bits they would get equals :

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

### 3.2 Proof of Correction :

Proof of correction straitly follows from :

- Polynomial multiplication commutativity.
- Theorem 1.
- Size in bits of the result of adding n, s bits numbers is  $s \times \log_2(n)$ .

### 3.3 Generalized Multidimensional Moddiv public key exchange protocol:

#### 3.3.1 Public parameters :

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

An  $m$  degree polynomial  $\mathbf{A}$ , which coefficients sizes in bits equal  $\|2^p\|$ .

$P, Q$  and  $\mathbf{A}$  coefficients are pseudorandomly generated.

#### 3.3.2 Private Computations :

- Alice chooses an  $n$  degree polynomial  $\mathbf{X}$ , which coefficient sizes in bits equal  $\|2^q\|$ .  $\mathbf{X}$  coefficients are pseudorandomly generated.

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

- For each  $\mathbf{U}$  coefficient  $U(i)$  :

– she calculates  $U1(i) = U(i) \text{ Mod}(P) \text{ Div}(Q)$ , getting thus polynomial  $\mathbf{U1}$ .

- Bob chooses an  $n$  degree polynomial  $\mathbf{Y}$ , which coefficient sizes in bits equal  $\|2^q\|$ .  $\mathbf{Y}$  coefficients are pseudorandomly generated.

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

- For each  $\mathbf{V}$  coefficient  $V(i)$  :

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

#### 3.3.3 Publicly exchanged values :

- Alice sends  $\mathbf{U1}$  coefficients to Bob.

- Bob sends  $\mathbf{V1}$  coefficients to Alice.

#### 3.3.4 Further Private Computations :

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

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

- She calculates  $r(i)$ ,  $Wa(i)$  consecutive least significant bits with the same value, number + 1.
- She calculates  $Wa1(i) = Wa(i) \text{ Mod}(P \text{ Div}(Q)) \text{ Div}(2^{q+r(i)+\log_2(n)})$ .

- Bob calculates polynomial  $\mathbf{Wb} = \text{pmdv}_{(m-n,n)}(\mathbf{X} \times \mathbf{U1})$ .

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

- He calculates  $r(i)$ ,  $Wb(i)$  consecutive least significant bits with the same value, number + 1.
- He calculates  $Wb1(i) = Wb(i) \text{ Mod}(P \text{ Div}(Q)) \text{ Div}(2^{q+r(i)+\log_2(n)})$ .

Shared key size in bits they could get equals :

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

We have experimentally observed that generalized variant of proposed public key exchange protocol holds.

### 3.4 Security :

To attack proposed public key exchange protocol, adversary Eve knows polynomials  $\mathbf{A}$ ,  $\mathbf{U1}$ ,  $\mathbf{V1}$  and parameters  $p$ ,  $q$ ,  $m$ ,  $n$  satisfying the following conditions :

- $\mathbf{A}$  is of degree  $m$ .
- $\mathbf{U1}$  and  $\mathbf{V1}$  are of degree  $m-n$ .
- $\mathbf{A}$  coefficients sizes in bits are  $p$ .
- $\mathbf{U1}$  and  $\mathbf{V1}$  coefficients sizes in bits are  $p-q$ .
- $m > 2 \times n$  and  $p > 2 \times q$ .

She equally knows that there exists polynomials  $\mathbf{U}$ ,  $\mathbf{V}$ ,  $\mathbf{X}$  and  $\mathbf{Y}$  Satisfying :

- $\mathbf{X}$  and  $\mathbf{Y}$  are of degree  $n$ .
- $\mathbf{U}$  and  $\mathbf{V}$  are of degree  $m-n$ .
- $\mathbf{X}$  and  $\mathbf{Y}$  coefficients sizes in bits are  $q$ .
- $\mathbf{U}$  and  $\mathbf{V}$  coefficients sizes in bits are  $p + q + \log_2(n)$ .
- $\mathbf{U} = (\mathbf{X} \times \mathbf{A}) \text{ polyMod}(x^m) \text{ polyDiv}(x^n)$ .
- $\mathbf{V} = (\mathbf{Y} \times \mathbf{A}) \text{ polyMod}(x^m) \text{ polyDiv}(x^n)$ .

Coefficients of  $\mathbf{U}$  and  $\mathbf{V}$  are respectively related to those of  $\mathbf{U1}$  and  $\mathbf{V1}$  by following equations :

- $U1(i) = U(i) \text{ Mod}(2^p) \text{ Div}(2^q)$ .
- $V1(i) = V(i) \text{ Mod}(2^p) \text{ Div}(2^q)$ .

Meaning to know secret polynomials  $\mathbf{X}$  and  $\mathbf{Y}$ , Eve have to solve the following equations set :

For  $1 \leq i \leq m - n$

$$U1(i) = ( \sum_{j=1}^{j=n} A(j+i) \times X(j) ) \text{ Mod}(2^p) \text{ Div}(2^q) .$$

$$V1(i) = ( \sum_{j=1}^{j=n} A(j+i) \times Y(j) ) \text{ Mod}(2^p) \text{ Div}(2^q) .$$

To Attack Generalized Multidimensional ModDiv public key, Eve had to solve the following equations set, knowing parameters P and Q satisfy  $\| P \| = \| 2^p \|$  and  $\| Q \| = \| 2^q \|$

For  $1 \leq i \leq m - n$

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

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

### 3.4.1 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 al [6], which we would define as ModDiv learn with rounding problem.

Said instance is characterized by the following features :

- Rounding is done from  $\mathbb{Z}_{2^p}$  to  $\mathbb{Z}_{2^{(p-q)}}$ .
- In Generalized Multidimensional ModDiv, rounding is done from  $\mathbb{Z}_P$  to  $\mathbb{Z}_{P/Q}$ .
- Public vectors are not random, they reflect polynomial multiplication algebraic structure : If vector  $\mathbf{A1}$  [ A(1), A(2) .... A(n) ] is given as public, vector  $\mathbf{A2}$  [ R(1),A(2),.... A(n-1) ]. 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, which is q bits.

Seemingly Multidimensional ModDiv learn with rounding problem is at most as hard as its generalized counterpart, but public key cryptosystem based on the first is easier to implement, the modulus are powers of two : bit shifting operations could be used instead of actual integer modulo and division operations.

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.

The open question is how hard ModDiv learn with rounding problem, actually is ?.

## 4 Conclusion :

In this paper we have presented Multidimensional ModDiv public key exchange protocol, the multidimensional 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.

One can construct public key encryption and digital signature ElGamal schemes based on presented Key exchange protocols, 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).*