On a new fast public key cryptosystem

Samir Bouftass

E-mail : crypticator@gmail.com.

December 15, 2014

Abstract

This paper presents a new fast public key cryptosystem namely : a key exchange algorithm, a public key encryption algorithm and a digital signature algorithm, based on a the difficulty to invert the following function : $F(x) = (a \times x) Mod(2^p) Div(2^q)$.
Mod is modulo operation , Div is integer division operation , a , p and q are known natural numbers while $(p > q)$.
In this paper it is also proven that this problem is equivalent to SAT problem which is NP complete .

Keywords : key exchange, public key encryption, digital signature, boolean satisfability problem, NP complete .

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. But in the last decades it has been proven that public key cryptosystems based on number theory problems are not immune againt quantum computing attacks [3]. The advent of low computing ressources mobile devices such wirless rfid sensors, smart cellphones, ect has also put demands on very fast and lightweight public key algorithms .
Public key cryptosystem presented in this paper is not based on number theory problems and is very fast compared to Diffie-Hellman [1] and RSA algorithms [2]. It is based on the difficulty to invert the following function : $F(x) = (a \times x) Mod(2^p) Div(2^q)$.
Mod is modulo operation , Div is Integer division operation , a , p and q are known natural numbers while $(p > q)$. In this paper we construct three public key algorithms based on this problem namely a key exchange algorithm, a public key encryption algorithm and a digital signature algorithm.
We prove its efficiency compared to Diffie-Hellman and RSA, and that the problem which it is based on is equivalent to SAT which is a NP complet problem [4] .

2 Secret key exchange algorithm :

Before exchanging a secret key, Alice and Bob shared a knowledge of :

Natural numbers [l, m, p, q, M, D, Z] satisfying following conditions :

q = l + m - p , $p > m + q$, $M = 2^p$,$D = 2^{(m+q)}$, Z is l bits long.

To exchange a secret key :

- 1 Bob chooses randomly natural numbers [X , r1 , r2]. X is m bits long wheras r1 and r2 are q bits long. These numbers are a private knowledge of Bob.

- 2 Computes numbers : $Rx = r1 \times 2^p + r2$ and $U = X \times Z + Rx$, then sends U to Alice.

- 3 Alice chooses randomly natural numbers [Y , r3 , r4]. Y is m bits long whereas r3 et r4 q bits long. These numbers are a private knowledge of Alice.

- 4 Computes numbers : $Ry = r3 \times 2^p + r4$ and $V = Y \times Z + Ry$, then sends V to Bob.

- 5 Bob computes number $W = (X \times V)Mod(M)Div(D)$.

- 6 Alice computes number $W = (Y \times U)Mod(M)Div(D)$.

The secrete key exchanged by Bob and Alice is the number :

$W = (X \times V)Mod(M)Div(D) = (Y \times U)Mod(M)Div(D)$

The size of W in bits is egal to p - (m + q).

3 Public key encryption algorithm :

3.1 Encryption :

In order to send a encrypted message to Bob, Alice performs the following steps :

-1 She gots his public key composed by natural numbers [l, m, p, q, M, D, Z, U] , satisfying conditions : q = l + m - p , $p > m + q$, $M = 2^p$,$D = 2^{(m+q)}$, $Rx = r1 \times 2^p + r2$ and $U = X \times Z + Rx$, X is m bits long whereas r1 and r2 are q bits long.

- 2 She chooses randomly natural numbers [Y, r3, r4] Y is m bits long whereas r3 and r4 are q bits long.

- 3 She computes numbers $Ry = r3 \times 2^p + r4$ and $V = Y \times Z + Ry$, then the secret key $W = (Y \times U)Mod(M)Div(D)$.

- 4 She encrypts with secret key W her plaintext and sends corresponding ciphertext and number V to Bob.

3.2 Decryption :

In order to decrypt the ciphertext recieved from Alice, Bob performs the following steps :

- 1 From element X of his private key and number V recieved from Alice, he computes secret key $W = (X \times V)Mod(M)Div(D)$.

- 2 With secret key W, he decrypts the ciphertext recieved from Alice.

4 Digital signature Algorithm :

4.1 Signature :

In order to sign a Message Msg, Bob performs the following steps :

- 1 He chooses randomly natural numbers r3 and r4 wich are q bits long.

- 2 Computes $Ry = r3 \times 2^p + r4$.

- 3 Hashes Msg by a hash function HF and gets a digest H which length in bits is the same as elements Z of his public key . From element X of his private key, he computes a signature $S = X \times H + Ry$

- 4 Sends Message Msg and signature S to Alice.

4.2 Verification :

In order to verify that Message Msg is sent by Bob, Alice performs the following steps :

- 1 She gots his public key composed by natural numbers [l, m, p, q, M, D, Z , U] , satisfying conditions : q = l + m - p , $p > l + q$, $M = 2^p$,$D = 2^{(l+q)}$, $Rx = r1 \times 2^p + r2$ and $U = X \times Z + Rx$, X is m bits long whereas r1 and r2 are q bits long.

- 2 Hashes Msg by HF and gets a digest H which the length in bits is the same as Zs.

- 3 From digest H , signature S and the elements [U, Z, M, D] of Bob's public key, she computes two numbers $Wx = (H \times U) Mod(M) Div(D)$ and $Wy = (Z \times S) Mod(M) Div(D)$.

- 4 Compares Wx to Wy : Msg is sent by Bob if Wx = Wy

5 Efficiency :

In comparaison to the standardised key exchange algorithms such as Diffie-Hellman in the multiplicatif group and RSA which needed in average N multiplications modulo operations to exchange a secret key (N being private key's lenght).
The key exchange algorithm presented in this paper needed just 4 multiplications ,
Meaning that presented public key cryptosystem is very fast and efficient compared to Diffie-Hellman and RSA cryptosystems.

6 Security :

The Security of presented public key cryptosystem is based on the difficulty of finding X and Y while knowing Z, l, m, p, q, $U = X \times Z + Rx$,$V = Y \times Z + Ry$, $Rx = r1 \times 2^p + r2$, $Ry = r3 \times 2^p + r4$, $l + m = p + q$, $p > m + q$, Z is m bit long , X and Y are l bits long whereas r1, r2, r3, r4 are q bits long

Numbers Rx and Ry have a format such as by adding them to $Z \times X$ and $Z \times Y$, the rightmost and the leftmost q bits of these two products are masked.
To ge X from U , and Y from V a attacker schould :

1 - Brute force Rx and Ry , X and Y are solutions if they devide rescpectively U - Rx and V - Ry.

2 - Invert $F(X) = (Z \times X)Mod(2^p)Div(2^q) = (U)Mod(2^p)Div(2^q)$ and $F(Y) = (Z \times Y)Mod(2^p)Div(2^q) = (V)Mod(2^p)Div(2^q)$.

Puting it otherwise, presented public key cryptosystem is based on the difficulty to invert the following function :

$F(x) = y = (a \times x)Mod(2^p)Div(2^q)$.

a, x, p and q are known natural numbers, while a and x are respectively n and m bits long, $(n > m)$ and $(p > q)$.

At first glance we can notice that it is easy to verify a solution but it is difficult to find one, implying that this problem is in NP.

6.1 Proof of equivalence to SAT :

Let the binary representation of A be $a_{(n)}...a_{(i+1)}a_{(i)}...a_{(0)}$.

The binary representation of X be $x_{(m)}...x_{(i+1)}x_{(i)}...x_{(0)}$.

The binary representation of Y be $y_{(n+m)}...y_{(i+1)}y_{(i)}...y_{(0)}$.

Y is the arithmetic product of A and X, our problem consists then on solving this system of equations :

If $(q \leq j \leq m)$:

$$y_j = ((\sum_{i=0}^{j} a_{(j-i)} \times x_i) + c_j)Mod(2) \quad \text{and} \quad c_j = ((\sum_{i=0}^{j-1} a_{(j-1-i)} \times x_i) + c_{j-1})Div(2)$$

If $(m \leq j \leq n)$:

$$y_j = ((\sum_{i=j-m}^{j} a_{(j-i)} \times x_i) + c_j)Mod(2) \quad \text{and} \quad c_j = ((\sum_{i=j-1-m}^{j-1} a_{(j-1-i)} \times x_i) + c_{j-1})Div(2)$$

If $(n \leq j \leq p)$:

$$y_j = ((\sum_{i=j-n}^{n} a_{(j-i)} \times x_i) + c_j)Mod(2) \quad \text{and} \quad c_j = ((\sum_{i=j-1-n}^{n} a_{(j-1-i)} \times x_i) + c_{j-1})Div(2)$$

c_j is the retenue bit of multiplication product $(Y = A \times X)$ at column j.

Solving this system of equations is equivalent to find boolean values $x_{i=0 \to m}$ satisfying the following logical functions :

If $(j \leq m)$ $c_j = F_j(x_{(j-1)}, ..., x_{(k+1)}, x_{(k)}, ..., x_0, c_{(j-1)})$

If $(m \leq j)$ $c_j = F_j(x_m, ..., x_{(k+1)}, x_{(k)}, ..., x_0, c_{(j-1)})$

$\wedge_{j=q}^{m}((\oplus_{i=0}^{j}(a_{(j-i)} \wedge x_i) \oplus c_j) = y_j) = true$

$\wedge_{j=m}^{n}((\oplus_{i=j-m}^{j}(a_{(j-i)} \wedge x_i) \oplus c_j) = y_j) = true$

$\wedge_{j=n}^{p}((\oplus_{i=j-n}^{n}(a_{(j-i)} \wedge x_i) \oplus c_j) = y_j) = true$

It's known that every logical function can be converted into an equivalent formula that is in CNF, proving then that our problem is equivalent to the boolean satisfability problem which is NP Complete.

7 Conclusion and open question :

In this paper we have presented a new fast public key cryptosystem based on the difficulty of inverting the following function : $F(x) = (a \times x)Mod(2^p)Div(2^q)$.
Mod is modulo operation , Div is integer division operation , A , r and s are known natural numbers while $(p > q)$.

We have proved its efficiency compared to Diffie Hellman and RSA cryptosystems. We have also proved that its security is based on a new problem equivalent to SAT.
The fact that its security is not based on number theory problems is also a proof of its resistance against quantum computing attacks.

The last decade have seen a enormous progress of SAT Solvers but they are still inefficient in solving logical statements containg a lot of xors which is the case of our problem [5].

SAT is NP complete, meaning that solving it can take exponential time. It is has been found that the hardest instances of a SAT problem depends on its constraindness which is defined as the ratio of clauses to variables [6].

This lead us to ask what forms should have the natural numbers composing public parameters of our PKCS in order to produce hard SAT instances even to a eventual SAT Solver that have not problems with xor clauses.

References

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

[2] R.L. Rivest , A. Shamir , L. Adleman *A method of obtaining digital signatures and public key cryptosystems , SL Graham, RL Rivest* Editors (1978)*

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

[4] Thomas J Schaefer *The complexity of satisfability problems* , (1978), Departement of Mathematics University of California, Berkeley .

[5] Tero Laitinen, Tommi Junttila, Ilkka Niemel*Conflict-Driven XOR-Clause Learning* , (2014), Logic in Computer Science .

[6] Eugene Nudelman, Kevin Leyton-Brown, Holger H. Hoos, Alex Devkar, Yoav Shoham *Understanding Random SAT: Beyond the Clauses-to-Variables Ratio* , (1978), Principles and Practice of Constraint Programming CP 2004 Lecture Notes in Computer Science Volume 3258, 2004 .