International Journal of IT & Knowledge Management (ISSN: 0973-4414)
July-Dec 2008, Vol 1, Number-2, pp 578-588

NEW VARIANT PUBLIC-KEY CRYPTOSYSTEMS BASED ON JORDAN'S TOTIENT FUNCTION

S.Makbul Hussian[1], Dr.S.Thajoddin[2], Dr.SAM Gazni[3]

[1,2]Lecturer in Mathematics, [3]Lecturer in Physics, Osmania College, Kurnool AP, India

Abstract: The public-key cryptography provides answers to all the problems of key managements and digital signatures. Its algorithms are based on mathematical functions rather than on substitution and permutation. The mathematical trick of this scheme is that it is relatively easy to compute exponents compared to computing discrete logarithms. The purpose of the algorithm is to enable two users to exchange a key securely that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of the keys.

Keywords: public-key, answers, algorithms, easy to compute, exchange of the keys.

INTRODUCTION:

The development of public-key cryptography is the greatest and perhaps the only true revolution in the entire history of cryptography. The public-key cryptography provides answers to all the problems of key managements and digital signatures. Its algorithms are based on mathematical functions rather than on substitution and permutation. More important public-key cryptography is asymmetric involving the use of two separate keys, in contrast to symmetric conventional encryption which uses only one key. The use of two keys has profound consequences in the area of confidentiality, key distribution and authentication.

We describe well known public-key cryptosystems namely Diffie-Hellman Key Exchange Scheme, El Gamal, Messey - Omura Cryptosystem and Paillier Cryptosystems. Before that we define informally the primitive root of a prime number.

Definition: g is a primitive root of a prime number p if the numbers g mod p, g^2 mod p……….g^{p-1} mod p are distinct and consists of the integers from 1 through p -1 in some permutation.

Diffie - Heilman Key Exchange:

The first published public-key algorithm appeared in the seminar paper by Diffie and Heilman that defined public-key cryptography and is generally referred to as Diffie - Heilman Key exchange. The mathematical trick of this scheme is that it is relatively easy to compute exponents compared to computing discrete logarithms. The purpose of the algorithm is to enable two users to exchange a key securely that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of the keys.

The Diffie - Heilman key exchange scheme works as follows. Alice and Bob wish to agree on a common, secret key, and then can communicate only over an insecure channel. First they agree on a large prime number p and an integer g which is a primitive root of p. The prime p and a primitive root g can be publicly known. Hence, Bob and Alice can use their insecure communication channel for this agreement.

Now Alice chooses a random integer a < p. She computes $A = g^a \bmod p$ and sends the result A to Bob, but she keeps the exponent a secret.

Bob chooses independently integer b < p randomly. He computes $B = g^b \bmod p$ and sends the result to Alice. He also keeps his exponent b secret. To obtain the common secret key, Alice computes $B^a \bmod p = g^{ab} \bmod p$ and Bob computes

$A^b \bmod p = g^{ab} \bmod p$

Therefore their common key is $k = g^{ab} \bmod p$.

Diffie-Hellman Key exchange Scheme on other groups.

A secure and efficient Diffie - Heilman key exchange scheme can be implemented in all cyclic groups in which the Diffie - Heilman problem is difficult to solve and for which the group operations can be efficiently implemented. Here we only describe how the implementation of the Diffie - Heilman protocol in such groups works in principle.

Alice and Bob agree on a finite cyclic group G and a generator g of G. Let n be the order of G. Alice chooses randomly an integer $a \in \{1,2,\ldots\ldots\ldots n-1\}$. She computes $A = g^a$ and sends the result A to Bob. Bob chooses randomly an integer $b \in \{1, 2, n-1\}$. He computes $B = g^b$ and sends the result B to Alice.

Alice determines $B^a = g^{ab}$ and Bob determines $A^b = g^{ab}$

The common secret key is $K = g^{ab}$

El Gamal Cryptosystem:

The El Gamal Cryptosystem was introduced by El Gamal [19] in 1985 and is based on the hardness of finding the discrete logarithm. The message space is defined by a cyclic group, for which the discrete logarithm problem is hard. Typical choices are Z_p (P is large safe prime) or Z_n (n is a RSA modulus). The algorithm for key generation, encryption and decryption is as follows.

Key Generation: Choose a group G of order p (e.g. Z_p), with p a safe prime and a generator g for this group.

Choose a random x from Zp and compute $h = g^x \bmod p$.

Public – Key PK = (G, g, p, h)
Private Key SK = x

Encryption: Convert M into element of G, M_G chooses an r random from Z_p as the blinding factor and compute the cipher text pair.

$C_1 = g^r \bmod p$ and $C_2 = M_G \cdot h^r \bmod p$

Decryption: Compute $C2\,(C_1^{\,x})^{-1} \bmod p$

$$= MG\, h^r\, g^{-xr} \bmod p$$

$$= M_G\, g^{xr}\, g^{-xr} \bmod p$$

$$= M_G \bmod p$$

$$= M_G$$

Using the extended Euclidean algorithm to find the multiplicative inverse and convert back to the original message M.

Messey-Omura Cryptosystem:

This cryptosystem is also based on the difficulty of finding discrete logarithms. All the users have agreed upon a public prime p. Now each user A chooses two positives integers e_A and d_A such that $e_A\, d_A \bmod p = 1$

In contrast with RSA cryptosystem, in this system the users keep both the numbers secret, publishing neither of them. Now consider the situation in which the user A sends the message m to the user B. We assume that a number represents the message, which is less than p.

The algorithms work as follows.

- The user A computes m^{eA} mod p and sends to the user B.
- The user B computes the e_B^{th} power of the number he has received and return the result m^{eAeB} mod p to the user A.
- Now the user A applies his number d_A to what he received and gets m^{eAeBdA} mod p. This number turns out to be m^{eB} mod p. The user A sends this result to the user B.
- The user B applies dB to the received number and obtains the message m.

Paillier Cryptosystem:

The Paillier cryptosystem was introduced by Paillier [20] in 1999, based on the nth Residuosity class problem. The algorithm for key generation, encryption and decryption is as follows.

Define the function L (x) = x-1 / N

Key Generation: Select two large primes p and q and compute N = pq and λ = lcm (p-1, q-1).

Select a base g $\in Z_{N^2}{}^*$ and check

$\mathrm{Gcd}\left(L\left(g^{\lambda} \bmod N^2\right), N\right) = 1$ in order to make sure that N divides the order of g.

Public-Key PK = (g, N)

Private Key SK = λ

Encryption: Select a random r $\in Z_N$

$$C = g^m r^N \bmod N^2$$

Decryption:

$$m = \frac{L\left(C^{\lambda} \bmod N^2\right)}{L\left(g^{\lambda} \bmod N^2\right)} \bmod N$$

In this paper we present our new variants of Diffie-Hellman key exchange scheme, El Gamal cryptosystem, Messey-Omura cryptosystem and Paillier Schemes based on Jordan Totient function and explained these algorithms with simple examples. We analyze the significance and complexity of the above schemes.

Variant of Diffie-Hellman Key Exchange Scheme based on J_k (N).

This scheme is described as follows.

Alice and Bob wish to agree on a common secret key. They can communicate only over an insecure channel.

First, they agree on two positive integers N and K such that $1 \leq K \leq N$ and

Compute $J_K(N) = N^K \underset{P/N}{\pi} \left(1 - 1/P^K\right)$

Consider $\left(Z_{J_K(N)}, +_{J_K(N)}, x_{J_k(N)}\right)$a

commutative ring with unity as a message space.

They select a generator $g \in Z_{J_{K(N)}}$ and their

Public-Key $PK = \left(Z_{J_{K(N)}}, J_k(N), g\right)$

Alice	Bob
Alice choose an integer x randomly and keep it secret. **Encryption :** $C_A = g^x \bmod J_k(N)$ $= xg \bmod J_k(N)$ Sends Sends C_A to Bob **Message Verification :** $m_A = {C_B}^x \bmod J_k(N)$ $= x\, C_B \bmod J_k(N)$ $= x\, yg \bmod J_k(N)$	Bob choose an integer y randomly and keep it secret. **Encryption :** $C_B = g^y \bmod J_k(N)$ $= yg \bmod Jk(N)$ Sends C_B to Alice **Message Verification :** $m_B = {C_A}^y \bmod J_k(N)$ $= y\, C_A \bmod J_k(N)$ $= y\, x\, g \bmod J_k(N)$ $= x\, y\, g \bmod J_k(N)$
∴ ∴ $m_A = m_B$	

∴ The common secret Key $S_k = g^{xy} \bmod J_k(N)$

$= xyg \bmod J_k(N)$

Example:. Let N =7, K=2 :

$J_K(N) = J_2(7) = 7^2-1 = 79-1 = 48$

∴ $(Z_{48}, +_{48}, X_{48})$ is a commutative ring with unity of order 48 is a message space.

Consider $g = 5 \in Z_{48}$

Public – Key PK = $(Z_{48}, 48, 5)$

Alice	Bob
Secret Key x = 50 **Encryption :** $C_A = g^x \bmod J_k(N)$ $= xg \bmod J_k(N)$ $= 50 \text{ x } 5 \bmod 48$ $= 250 \bmod 48$ $= 10$ Sends $C_A = 10$ to Bob **Message Verification :** $m_A = {C_B}^x \bmod J_k(N)$ $= x\, C_B \bmod J_k(N)$	Secret Key y = 60 **Encryption :** $C_B = g^y \bmod J_k(N)$ $= yg \bmod J_k(N)$ $= 60 \text{ x } 5 \bmod 48$ $= 300 \bmod 48$ $= 12$ Sends $C_B = 12$ to Alice **Message Verification :** $m_B = {C_A}^y \bmod J_k(N)$ $= y\, C_A \bmod J_k(N)$

= 50 x 12 mod 48 = 600 mod 48 = 24	= 60 x 10 mod 48 = 600 mod 48 = 24
∴ Their common secret key is 24	

Variant of El Gamal Cryptosystem based on J_k (N) :

Choose two positive integers N and K such that $1 \leq K \leq N$ and

Compute $Jk(N) = N^k \underset{p/N}{\pi}\left(1-1/p^k\right)$

Consider $\left(Z_{J_K(N),}{}^{+}J_K(N),{}^{X}J_K(N)\right)$ a Commutative ring with unity as a message space. Choose a generator $g \in Z_{J_K(N)}$ and compute

$h = g^x \bmod J_k(N) = xg \bmod J_k(N)$

Public – Key PK = $\left(Z_{J_K(N),}J_K(N), g, h\right)$	
Private Key SK = x	

Encryption:

Choose a random r from $Z_{J_K(N)}$ as the blinding factor.

Cipher text pair $C_1 = g^r \bmod J_k(N) = rg \bmod J_k(N)$

$C_2 = (M+h^r) \bmod J_k(N) = (M+rh) \bmod J_k(N)$

Decryption:

$$\text{Compute } \left\{C_2 + \left(C_1^{X}\right)^{-1}\right\} \bmod J_k(N)$$

$$= \left\{\left(M + h^r\right) + g^{-xr}\right\} \bmod J_k(N)$$

$$= \left\{\left(M + g^{xr}\right) + g^{-xr}\right\} \bmod J_k(N)$$

$$= \left\{M + (xr)g + (-xr)g\right\} \bmod J_k(N)$$

$$= \left\{M + (xr)g + (-xr)g\right\} \bmod J_k(N)$$

$$= M \bmod J_k(N)$$

$$= M$$

Example

Choose N=7, K = 2

$J_k(N) = J_k(7) = 7^2 - 1 = 49-1=48$

Consider $\left(Z_{48}, +_{48}, X_{48}\right)$ a commutative ring with unit of order 48 as a message space

Choose a generator g = 5

Select a Select a secret key x = 11

Find h = g mod Jk (N)

= xg mod Jk (N)

= 11 x 5 mod 48

= 55 mod 48

= 7

Public – Key PK = (Z_{48}, 48, 5, 7)

Encryption:

Choose M = 10 ∈ Z_{48}

Choose a random r from Z_{48} let it be 40

i.e. r = 40

Cipher text pair $C_1 = g^r \bmod J_k (N)$

$= rg \bmod J_k (N)$

= 40 x 5 mod 48

= 200 mod 48

= 8

$C_1 = \{ M+h^r) \bmod J_k (N)$

= { 10 +rh } mod 48

= { 10+40x7 } mod 48

= 290 mod 48

= 2

Decryption:

Compute $\left\{C_2 + \left(C_1^{X}\right)^{-1}\right\} \operatorname{mod} J_k\left(N\right)$

$$= \left\{\left(2 + (8^{11}\right)^{-1}\right\} \operatorname{mod} 48$$

$$= \left\{2 + 8^{-11}\right\} \operatorname{mod} 48$$

$$= \left\{2 + \left(-11\right)8\right\} \operatorname{mod} 48$$

$$= -86 \operatorname{mod} 48$$

$$= 10$$

$$= M$$

Variant of Messey-Omura Cryptosystem Cryptosystem based on J_k (N) :

In this cryptosystem all the users have agreed upon a public-prime p computers J_k (p) = (p^k – 1) and consider $\left(Z_{J_{K}(P)},^{+} J_{K(P)},^{X} J_{K(P)}\right)$ a commutative ring with unity as a message space. Convert the message into the elements of $Z_{J_{K}(P)}$.

Now each user choose two positive integers e and d such that ed ≡ 1 mod J_k (p)

In contrast with RSA cryptosystem, in the system the user keeps both the numbers secret, publishing neither of them.

Consider the situation in which the user Alice sends the message m belongs to $Z_{J_{K(P)}}$ to the user Bob. The algorithm works as follows.

- The user Alice computes $W = m^{eA} \bmod J_K(p)$

$$= e_A m \bmod J_k(p)$$

  And sends to the user Bob.

  The use Bob computers $X \quad = W^{eB} \bmod J_K(p)$

$$= e_B W \bmod J_k(p)$$

$$= e_B e_A m \bmod J_k(p) \text{ and}$$

  Returns to the user Alice.

  Now the user Alice applies his number d_A to that he received and gets.

$$\begin{aligned} Y &= X^{dA} \bmod J_K(p) \\ &= d_A m \bmod J_K(p) \\ &= d_A e_B e_A m \bmod J_K(p) \\ &= (e_A d_A) e_B m \bmod J_K(p) \\ &= e_B m \bmod J_K(p) \end{aligned}$$

. Now Alice sends this result to the user Bob.

- The user Bob applies d_B to the recovered number Y and computes $Y = y^{d_b} \bmod J_k(p)$

$$\begin{aligned} &= d_B y \bmod J_k(p) \\ &= e_B d_B m \bmod J_k(p) \\ &= m \bmod J_k(p) \;= m. \end{aligned}$$

Example :

Let P =7, k = 2

$J_K(p)=J_2(7) = 7^2-1 = 49-1=48$

Consider $\left(Z_{48,}\ +_{48}, X_{48}\right)$ a commutative ring with unity as a message space. Let m = 10 ∈ Z_{48}

Alice	Bob
Selects $e_{A,}\ d_A$ such that	Selects $e_{B,}\ d_B$ such that

$e_A\, d_A \equiv$ (mod 48) Suppose Alice selects $e_A = 5$, $d_A = 29$ Computes $W = m^{e}{}_{A}$ mod 48 $= e_A$ m mod 48 = 5.10 mod 48 =2 Alice sends W to Bob Computes $Y = X^{d}{}_{A}$ mod 48 $= d_A x$ mod 48 = 29 x14 mod 48 = 406 mod 48 = 22 Alice sends y to Bob	$e_B\, d_B \equiv 1$ (mod 48) Suppose Alice selects $E_B = 7$, $d_B = 7$ Computes $X = w^{eB}$ mod 48 $= e_{B\,W}$ mod 48 = 7.2 mod 48 = 14 Bob sends X to Alice Computers $Z = Y^{d}{}_{B}$ mod 48 $= d_B y$ mod 48 = 7 x22 mod 48 = 154 mod 48 = 10

Variant of Paillier Cryptosystem based on J_k (N)

Choose two positive integers N and K, where $1 \leq K \leq N$ Compute J_k (N)

Consider $\left(Z_{J_K(N)}, {}^{+} J_{K(N)}, {}^{X} J_{K(N)}\right)$ a

Commutative ring with unity as a message space.

Consider function $L(x) = \dfrac{x \bmod J_k(N)}{r}$

Where r is random belongs to $Z_{J_{K(N)}}$

Select a generator $g \in Z_{J_{K(N)}}$ such that gcd (g, J_k (N)) = 1

Select d such that gd ≡ 1 mod J_k (N)

Public –Key PK = $\left(Z_{J_{K(N)}}, J_k(N), g, r\right)$
Private Key SK = $\left(Z_{J_{K(N)}}, J_k(N), g, d\right)$

Encryption:

$C = g^m$ r mod J_k (N), where m is the message belongs to $Z_{J_{K(N)}}$

Decryption:

$$L(dc) = \frac{dc \bmod J_k(N)}{r} = \frac{dg^m r \bmod J_k(n)}{r}$$

= d m g mod J_k (N)

= d g m mod J_k (N)

= g d m mod J_k (N)

= 1 m mod J_k (N)

= m

Example :

Let P =7, k = 2

$J_K(p)=J_2(7) = 7^2-1 = 49-1=48$

Consider $\left(Z_{48,} +_{48}, X_{48}\right)$ a commutative ring with unity as a message space.

$$L(x) = \frac{x \bmod J_k(N)}{r}$$

Where r is a random belongs to Z_{48}

Let r = 4

Select g ∈ Z_{48} such that (g, Jk (N)) = (g, 48) = 1

∴ (5, 48) = 1 ∴ take g = 5

Select d such that gd ≡ mod J_k (N)

5d ≡ 1 mod 48

5 x 29 ≡ 1 mod 48

∴ Take d = 29

Public Key PK = (Z_{48}, 48, 5, 4)

Private Key SK = (Z_{48}, 48, 29)

Take a message m = 3 ∈ Z_{48}

Encryption:

C = g^m r mod J_k (N)

= m g r mod J_k (N)

= (3 x 5 x 4) mod 48

= 60 mod 48

= 12

Decryption:

$$L(dc) = \frac{dc \bmod J_k(N)}{r}$$

$$= \frac{29 \times 12 \bmod 48}{4} \qquad = \frac{12}{4} = 3 = m$$

Remarks: This cryptosystem works when N is a product of two primes. In this case we can easily calculate e and d such that ed ≡ 1 mod J_k (N).

SIGNIFICANCE AND COMPLEXITY OF OUR DEVELOPED SCHEMES:

Our Developed Schemes have the following significance features.

1) All our schemes provide strong security but are also practicable.

2) The encryption algorithms of T.E1 Gamal Scheme and Paillier Schemes are one way functions unless, some trap door function is given, we cannot decrypt the plaintext from the ciphertext. So the schemes are very much secure and intractable.

3) Since we have taken $\left(Z_{J_K(N)}, +_{J_K(N)}, x_{J_K(N)}\right)$a commutative ring with unity as a message space we can use both the operations $+_{J_K(N)}$ and $x_{J_K(N)}$ in these cryptosystems.

4) Since k is a positive integer such that $1 \leq k \leq N$, therefore k is our choice. By choosing appropriate k we can make the message space as large as possible. If we assign numerical equivalents to the alphabets randomly from this message space, certainly it is very difficult to recover the plaintext from the cipher text. So all the above systems are very much secure and complex.

REFERENCES

1) A.J. Menezes, P.C.Van Oorschot and S.A.Vanstone : Handbook of Applied Cryptography, CRC Press, 1996

2) D.R. Stinson : Cryptography Theory and Practice, CRC 1995

3) G.J. Simmons: Contemporary Cryptography: The science of information integrity, IEEE Press, 1992.

4) R.Rivest : A. Shamir and L.Adleman : A Method for obtaining Digital signatures and public –key cryptosystems communications of the ACM 21 (2), pages 120-126, 1978.

5) J.J.Quisquater and C.Couvreur. Fast Deciphernent Algorithm for RSA public – key cryptosystem. Electronic Lectures, Vol-18, 905-907, 1982.

6) T.Collins, D,Hopkins, S.Langform and M.Sabin public key cryptographic Apparatus and Method U.S.Parent #5,848, 159, January – 1997

7) D.Bone and H.Shacham. Fast variantsof RSA. RSA laboratus 2002.

8) Alison Monteiro Paixao : An efficient of vaciant of the RSA cryptosystem.

9) T.M.Apostal, introduction to analytic number theory, springer International Students Edition 1980.

10) W.Diffie andM.Helpman : New Directions in cryptography, IEEE transactions on Information theory Vol-10, pages74-84, IEEE, 1977.

11) T.El.Gamal : A public key cryptosystem and signature scheme based on discrete Logarithms, IEEE Transactions on Information Theory Vol-31 pages 469-472, IEEE,1985

12) P.Pailer : Public-key cryptosystems based on composite residuosity classes Advances in cryptology proceedings of EUROCRYPT'99, LNCS 1592, pages 223-238, Springer Verlag'1999.