5. Public Key Cryptosystems

5.1. Public Key Distribution Systems

The ***public key distribution algorithm*** proposed by Deffie and Hellman appears to be difficult for computing logarithms over *GF(q)* with *q* elements *{0,1,2,...,q-1}*. Consider a pair of inverse functions

$$C=\alpha^M \ (\text{mod } q);$$

$$M=\log_\alpha C \ (\text{mod } q),$$

where $0<M,C<q$, respectively, q is prime, and α is primitive element of *GF(q)*. Calculation of C from M is easy, but computation of M from C is more difficult, because computing logarithms over *GF(q)* is believed to be much harder. The possible solution to a public-key distribution is

1. Each user A and B generates an independent random number M_i and M_j chosen uniformly from the set of integers *{1,2,...,q-1}*, and keep this numbers secret.

2. Both users make the following calculations A calculate $C_i=\alpha^{M_i} \ (\text{mod } q)$, and B calculate $C_j=\alpha^{M_J} \ (\text{mod } q)$ and place the values of C_i, C_j in a public file.

3. User A computes K_{ij} by obtaining C_j from the public file and letting

$$K_{ij}=C_j^{M_i} (mod\ q)=(\alpha^{M_J})^{M_i} (mod\ q)=\alpha^{M_J M_i} (mod\ q)$$

user B obtains the common key K_{ij} in a similar fashion as

$$K_{ji}=C_i^{M_j} (mod\ q)=(\alpha^{M_i})^{M_j} (mod\ q)=\alpha^{M_i M_j} (mod\ q)$$

We can see that $K_{ij}=K_{ji}$, that is why both users have the same key.

The other user must compute the $K_{ij}=K_{ji}$ from C_j and C_i by computing

$$K_{ij}=K_{ji}=C_i^{(\log_\alpha C_j)} (mod\ q)$$

5. Public Key Cryptosystems

5.1. Public Key Distribution Systems

Example 5.1. Consider a prime field *GF(q)* with a prime number *q*. Let α be a fixed primitive element of *GF(q)* such that the powers of α produce all nonzero elements *1,2,...,q-1* of *GF(q)*. Let us pick $\alpha = 2$ and $q = 11$. The powers of $\alpha = 2$ (mod *11)* produce the following table.

λ	0	1	2	3	4	5	6	7	8	9
α^{λ}	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9
α^{λ} (mod 11)	1	2	4	8	5	10	9	7	3	6

To initiate communication, user *A* chooses a random number $M_i = 5$ from the integer set 2^{λ} (mod 11)={1,2,3,...,10} in the above table and keeps it secret. Then he calculates $C_i = \alpha^{M_i}$ (mod q) $= 2^5$ (mod *11)=10* , and *B* generates secret $M_j = 7$ to calculate $C_j = \alpha^{M_j}$ (mod q) $= 2^7$ (mod *11)=7*. Finally both users A and B send their results C_i=10 and C_j=7 of calculation to partner to obtain the common key

$$K_{ij} = C_j^{M_i} (mod\ q) = (\alpha^{M_j})^{M_i} (mod\ q) = 7^5 (mod\ 11) = 10$$

user *B* obtains the common key K_{ij} in a similar fashion as

$$K_{ji} = C_i^{M_j} (mod\ q) = (\alpha^{M_i})^{M_j} (mod\ q) = 10^7 (mod\ 11) = 10$$

Thus, each user completes computation of the common key 10.

5. Public Key Cryptosystems

5.1. Public Key Distribution Systems

Example 5.2. Consider the key exchange problem in $GF(2^m)$ for $m=3$. A primitive polynomial $p(x)$ of degree $m=3$ over $GF(2)$ is $p(x)=1+x+x^3$. If a be α root of $p(x)$ over $GF(2)$, then the field elements of $GF(2^3)$ generated by $p(\alpha)=1+\alpha+\alpha^3=0$ are shown below

Power	Polynomial	Vector
α^0	1	1 0 0
α^1	α	0 1 0
α^2	α	0 0 1
α^3	$1+\alpha$	1 1 0
α^4	$\alpha+\alpha^2$	0 1 1
α^5	$1+\alpha+\alpha^2$	1 1 1
α^6	$1+\alpha^2$	1 0 1
α^7	1	1 0 0

Suppose users A and B select $M_i=2$ and $M_j=5$, respectively. Both $M_i=2$ and $M_j=5$ are kept secret, but $C_i=\alpha^{M_i}$ (mod $p(x)$)$=\alpha^2$(mod $(1+x+x^3)$)$=001$ and $C_j=\alpha^{M_j}$ (mod $p(x)$)$=\alpha^5$(mod $(1+x+x^3)$)$=111$ are placed in the public file. User A can communicate with user B by taking C_j=111 from the public file and computing their common key K_{ij} as follows

$$K_{ij}=(C_j)^{M_i}\ (\text{mod } p(x))=(\alpha^5)^2(\text{mod } p(x))=\alpha^{10}(\text{mod } p(x))=\alpha^3=110$$

5. Public Key Cryptosystems

5.2. Knapsack Cryptosystems

The knapsack problem is a mathematically attractive proportion for cryptography. The Merkle-Hellman scheme based on the trapdoor knapsack problem is a public-key asymmetric cryptosystem.

Assume a key $K=\{k_1, k_2,...,k_n\}$, where k_i are integers for $i=1,2,...,n$ and the n-bit plaintext $M=\{x_1,x_2,...,x_n\}$, where $x_i=\{0,1\}$. Then the knapsack cryptosystem enciphers the n-bit plaintext into the n-bit ciphertext according to the following formula:

$$C=KM= k_1x_1+k_2x_2+...+k_nx_n.$$

Calculation of C is as simple as seen from above equation, but recovery of M from C and K involves solving a knapsack problem and is generally difficult when n is large and K is randomly chosen.

If the key K is chosen such that each element of K is larger than the sum of the preceding elements, the corresponding knapsack problem becomes very simple. That is $k_i>k_1+k_2+k_3+...+k_{i-1}$. Letting $c_1=k_1x_1$, $c_2=k_1x_1+k_2x_2$, $c_n=k_1x_1+k_2x_2+...+k_nx_n$. where c_n represents the ciphertext C i.e. $C= y_n$. Now, the plaintext M can recovered from y_n for $i=1,2,...,n$ and the key K according to the following procedure. If $y_n<k_n$, then set x_n =0 and $y_{n-1}=y_n$. If $y_n>k_n$, then set x_n =1 and $y_{n-1}=y_n-k_n$. Using the computed value y_{n-1} we can find x_{n-1} and y_{n-2} in a similar fashion. The recovery procedure continues until $M=\{x_1,x_2,...,x_n\}$, has been completely recreated.

5. Public Key Cryptosystems

5.2. Knapsack Cryptosystems

Example 5.3. Suppose the plaintext $M=\{11001\}$ and the key $K=\{151, 187, 426, 1091, 2412\}$, the ciphertext C becomes

$$C=K*M=1*151+1*187+0*426+0*1091+1*2412=2750$$

Since $C=y_5=2750$ is the ciphertext, the recovery of x_5 from y_5 and k_5 is $x_5=1$ because $y_5=2750>k_5=2412$. Hence the procedure for recovery of x_i for $i=1,2,...,5$ will continue as shown below.

$y_5=2750>k_5=2412,\ x_5=1;$

$y_4=y_5-k_5=338<k_4=1091,\ x_4=0;$

$y_3=338<k_3=426,\ x_3=0;$

$y_2=338>k_2=187,\ x_2=1;$

$y_1=y_2-k_2=151=k_1,\ x_1=1.$

Thus the recovered plaintext is $M=\{11001\}$ as expected. However, this simple knapsack vector K cannot be used as a public enciphering key because someone can easily recover M from C.

5. Public Key Cryptosystems

5.2. Knapsack Cryptosystems

The public key is a vector $K_p=wK_p^*$ which is also published by the user as his public key, while the parameters $v=w^{-1}$ and m kept secret as his private keys. Kp is generated by multiplying each component of the knapsack secret vector $K_p^*=\{k_1^*, k_2^*,...,k_n^*\}$ by w modulo m such that

$$k_p=wk_p^* \ (mod\ m),\ i=1,2,...,n$$

Recovering the original plaintext M from the ciphertext C sometimes requires the transformation of C into C^* via v which is the multiplicative inverse of w modulo m. That is,

$$C^*=vC \ \ (mod\ m),$$

where a secret quantity $v=w^{-1}$ is calculated from m and w, where $m>w$ and $(w,m)=1$. Moreover, m has to be larger than $k_1+k_2+...+k_n$ according to

$$wv=1 \ \ (mod\ m) \ \text{ or } \ ww^{-1}=1\ (mod\ m)$$

5. Public Key Cryptosystems

5.3. Exponentiation Ciphers

In 1978, Pohling and Hellman published an ecryption scheme based on computing exponentials over a finite field. At about the same time, Riverst, Shamir and Adleman published a similar scheme, but with a slight twist - a twist that gave the MIT group a method for realizing public-key encryption as put forth by Deffie and Hellman.

The Pohling-Hellman and RSA schemes both encipher a message block $M=0,1,\ldots,n-1$ by computing the exponential

$$C=M^e \bmod n,$$

where e and n are the key to the enciphering transformation. M is restored by the same operation, but using a different exponent d for the key:

$$M=C^d \bmod n.$$

Enciphering and deciphering can be implemented using the fast exponentiation algorithm shown below, so $C=fastexp(M,e,n)$ and $M=fastexp(C,d,n)$.

The enciphering and deciphering transformations are based on Euler's generalization of Fermat's Theorem, which states that for every M relatevily prime to n:

$$M^{\phi(n)} \bmod n = 1.$$

This property implies that if e and d satisfy the relation $ed \bmod \phi(n)=1$, then deciphering restores the original plaintext message. This result is proved in the following theorem:

5. Public Key Cryptosystems

5.4. Fermat's Theorem

Theorem 5.1. Given e and d satisfying $ed \bmod \phi(n)=1$ and a message $M=0,1,\ldots,n-1$ such that $gcd(M,n)=1$, then

$$(M^e \bmod n)^d \bmod n=M.$$

Proof: We have

$$(M^e \bmod n)^d \bmod n=M^{ed} \bmod n.$$

Now, $ed \bmod \phi(n)=1$ implies that $ed = t\phi(n)+1$ for some integer t.

Thus,

$$M^{ed} \bmod n=M^{t\phi(n)+1} \bmod n=MM^{t\phi(n)} \bmod n=M(M^{t\phi(n)} \bmod n)\bmod n=M(M^{\phi(n)} \bmod n)^t \bmod n= M1^t \bmod n = M.$$

By symmetry, enciphering and deciphering are commutative and mutual inverses; thus,

$$(M^d \bmod n)^e \bmod n=M^{de} \bmod n = M.$$

Given $\phi(n)$, it is easy to generate a pair (e,d) satisfying $ed \bmod \phi(n)=1$. This is done by first choosing d relatively prime $\phi(n)$, and then using the extended version of Euclid's algorithm to compute its inverse:

$$e=inv(d,\phi(n)).$$

Because e and d are symmetric, we could also pick e and compute $d=inv(e,\phi(n))$.

5. Public Key Cryptosystems

5.4. Pohling-Hellman Scheme

In the Pohlig-Hellman Sheme, the modulus is chosen to be large prime p. The enciphering and deciphering functions are thus given by

$$C=M^e \bmod p$$

$$M=C^d \bmod p,$$

where all arithmetic is done in the Galois field ***GF(p)***. Because p is prime, $\phi(n)= p-1$, which is trivially derived from p.Thus the scheme can be used for conventional encryption, where e and d are both kept secret.

Example 5.4. Let $p=11$, whence $\phi(n)=p-1=10$. Choose $d=7$ and compute $e=inv(7,10)=3$. Suppose, $M=5$. Then M is enciphered as:

$$C=M^e \bmod p=5^3 \bmod 11=4.$$

Similarly, C is deciphered as:

$$M=C^d \bmod p=4^7 \bmod 11=5.$$

The security of the scheme rests on the complexity of computing discrete logarithms in ***GF(p)***.

5. Public Key Cryptosystems

5.5. Riverst-Shamir-Adleman (RSA) Scheme

In the RSA scheme, the modulus n is the product of two large primes p and q:

$$n=pq.$$

Thus, $\phi(n)=(p-1)(q-1)$. The enciphering and deciphering functions are

$$C=M^e \bmod n$$

$$M=C^d \bmod n,$$

Example 5.5. Let $p=5$, $p=7$ whence $n=pq=35$ and $\phi(n)=(5-1)(7-1)=24$. Choose $d=11$. and compute $e=inv(11,24)=11$. Suppose, $M=2$. Then M is enciphered as:

$C=M^e \bmod p=2^{11} \bmod 35=2048 \bmod 35 = 18$.and $M=C^d \bmod p=18^{11} \bmod 35=2$.

Example 5.6. Let $p=53$, $p=61$ whence $n=pq=53\ 61=3233$and $\phi(n)=(53-1)(61-1)=3120$. Letting $d=791$, we get $e=71$. To encipher the message $M=RENAISSANCE$, we break it into blocks of 4 digits each, where A=00, B=01,...,Z=25, and blank =26. We thus get

$$M= \ RE \quad NA \quad IS \quad SA \quad NC \quad E$$

$$M=1704\ 1300\ 0818\ 1800\ 1302\ 0426$$

The first block is enciphered as $1704^{71}=3106$. The entire message is enciphered as

$$C=3106\ 0100\ 0931\ 2691\ 1984\ 2927.$$

5. Public Key Cryptosystems

5.5. Procedure RSA's Key Generation

1. Select two secret primes, p and q, where a) p and q differ in 11 by a few digits; b) both $p-1$ and $q-1$ should contain large prime factors; c) $gcd(p-1,q-1)$ should be small.

2. Compute the product $n=pq$, which is the public modulus, where p and q are randomly selected.

3. Calculate Euler's function, $\phi(n)=(p-1)(q-1)$, which gives the secret number.

4. Select the key e or d, which is relatively prime to $\phi(n)$.

5. Calculate the multiplicative inverse of either d or $ed \bmod \phi(n)=1$ modulo n according to the equation $ed \bmod \phi(n)=1$.