KNAPSACK CRYPTOSYSTEM

&

NASH CRYPTOSYSTEM

Submitted by,

Sreekala S.

Christina Jacob

KNAPSACK CRYPTOSYSTEM

The Knapsack Cryptosystems are Public Key Cryptosystems that were first implemented in 1978 by R. Merkle and Hellmann and at once became quite popular because of its high speed and elegance. However, several attacks were made to break almost all Knapsack-Based Cryptosystems.

The purpose of this section is to introduce about:

* Knapsack Problem
* Construction of Public Key Knapsack Cryptosystem
* Analysis of the algorithm

**Knapsack Problem**

The Knapsack Problem is one of the classical combinatorial problems. The problem that is used in constructing Cryptosystem is closely related to it but still is exactly called Subset - Sum problem. The problem is , given an input set A and a sum S, to determine the Boolean integers x1,x2…xn such that the following holds:

x1a1 + x2a2 + …. + xnan = S

That is, to find out the subset of A which given the sum S. The general knapsack problem is known to be a NP-Complete and so it is believed to be hard to solve. But if the set A satisfies the super increasing ( ai >= a1+…ai-1) property then the problem gets reduced to a class P problem.

**Implementation**

In order to implement this problem in cryptosystem consider the set X as Plaintext, set A as Key and S as Ciphertext. Then from knapsack problem it is difficult to decrypt the plaintext (X) from the ciphertext (S) even though the key A is known. In order to decrypt the message, a trapdoor should be created. Those who know the trapdoor can decrypt it, others can’t do it. The implementation of the system has 3 steps. They are:

* Key Generation
* Encryption
* Decryption

Key Generation

* Create a super increasing k-tuple ‘ b ’
* Choose n and r , such that *n > b1+b2+…bk and GCD(n,r) = 1 by modulus n.*
* Create a temporary k-tuple *t=[t1,t2,…,tk]* in which

*ti = r bi mod n*

* Select a permutation of k objects and find a new tuple

*a = permute(t)*

* Public Key : a
* Private Key : n, r and b

Encryption

If the sender wishes to send a message, first he should converts his message into a binary format (set X). Then encrypt the message using the equation

S = x1a1 + x2a2 + …. + xnan

Decryption

The receiver gets the ciphertext S, which is the sum in terms of the set A. Since A is not a super increasing sequence there is no polynomial time algorithm to decrypt the message directly. So he should convert that sum into some other form. The equation is:

S‘ = S r-1 mod n = X A r-1 mod n { S = XA } = X p(b) r r-1 mod n { A = p(b) r mod n) } = X p(b) mod n

Thus S’ is the sum in terms of the permuted super increasing sequence and so it can be decrypted directly and thus the receiver gets the permuted plaintext. Then doing the inverse permutation gives the actual plaintext back.

**Analysis**

The main advantage of the Knapsack Cryptosystem is its speed. It is almost 100 times faster than RSA Cryptosystem. But it has several disadvantages. They are:

Set A is not purely random : We assume that the public key A is a random sequence and so there is no polynomial time algorithm to decrypt the message directly. But in the cryptosystem, since the set A is generated from a super increasing sequence B , it is not purely random. It contains some hidden information about the super increasing sequence. So in average case, it is easy to decrypt the text without knowing the private key. It is NP – Complete only in the worst case.

Low Density Problem : The density of the sequence is defined by the equation

Density, d = k/max log2 (ai)

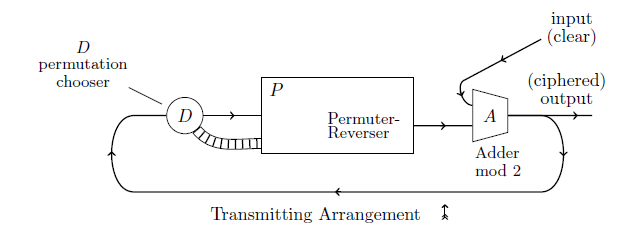
In this equation, ‘ a ‘ represents the public key. Since a = b r mod n , the value of ‘ a ‘ will be always less than ‘ n ‘ since there is a modulo operation. When the ‘ n ’ value increases, then the public key sequence is spread in a wide range. That is, the difference between two adjacent numbers is high and so the density is low. The significance of density is, it has been shown that if d<=0.9408, then the knapsack problem can be reduced to closest vector problem (CVP) which has a polynomial time approximation algorithm. Also it is very difficult to find a public key whose density is greater than the threshold value.

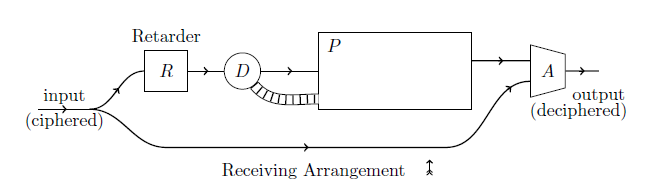
That is, it can be concluded that even though the actual knapsack problem is NP-Complete, when it is implemented as a cryptosystem it is NP-Complete only in worst case. In average case, it is only P-hard. Because of these reasons, Knapsack System got crashed.

NASH CRYPTOSYSTEM

The concept was introduced by the famous cryptographer John Nash back in 1955. In 1955 he wrote some letters to NSA (The National Security Agency of America). It is declassified recently in 2011. The cryptosystem got rejected due to lack of security. But its significance lies in the fact that in those letters he introduced the basics of the complexity theory which is the base for the modern cryptosystems.

Nash proposed an encryption decryption machine pair (both are physical systems), which is a symmetric key stream cipher.





You can see the encryption and decryption machine are the same except that there is an additional component called the retarder in the decryption machine .

The other components are

1. The Permuter--- Performs the core part of encryption/decryption
2. Adder
3. Decider

The purposes of this part of the lecture are

1. To introduce the complexity theory proposed by John Nash.
2. To study in detail about the encryption and decryption machines

* The functioning of permuter
* The significance of retarder

1. Analysis - The reason for rejecting the proposal

**Complexity Theory**

The cryptosystems should be classified based on the difficulty to decrypt it and the security should not depend on the secrecy of the algorithm or on the length of the key.

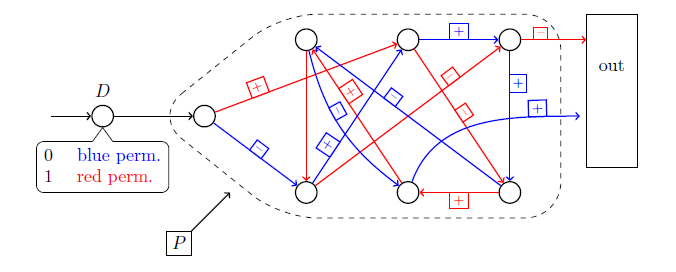
**Encryption Machine**

Adder : Adder takes in one bit of input plaintext, xor’s it with the output of the permuter and gives it as the output ciphertext bit. The same output bit is given as a feed back to the decider.

Ci = Pr(C i-1) XOR Pi

Decider : The decider takes in this input and decides what should happen inside the permuter.

Permuter : The permuter as the name implies, performs the permutation of the bits in its nodes. For the purpose it takes the help of two permutation paths( Both are Hamiltonian paths ie, passes through all the nodes exactly once).



How the permutation is performed? :

When a bit of input is comes from the decider to the permuter, the 1st node receives the input and every other bits in the nodes get shifted to the next node in the permutation path selected and the last node bit will come out as the output of the permuter. The selection of path is made according to the incoming bit from the decider. Ie the if the input bit from the decider is 0 the blue path will be selected otherwise the red path will be selected. While moving through the path according to the sign on the edge through which the bit is moving it will either get complemented (if sign is +) or it will be kept as it is ( if sign is -).

**Decryption Machine**

We performed the XOR operation to get the cipher text , so in order to get back the plaintext, we have to XOR the ciphertext with the same permuter output bit.

Ci = Pr(C i-1) XOR Pi

= Pr(C i-1) XOR Ci

= Pr(C i-1) XOR Pr(C i-1) XOR Pi

= P

From the decryptions it is clear that the same cipher text bit is required in two successive cycles of the decryption machine. The bit is required for the permuter in the second cycle and required for the adder in the 1st cycle. From the circuit we can see that it is available in the 1st cycle for both permuter and adder simultaneously on the same line. So we need to delay the input to the permuter by one clock cycle and it is accomplished by the retarder

**Analysis**

Unfortunately his ideas got rejected. The reason for the rejection of his complexity theory is simply the reluctance of the society to accept innovations. At the time when Nash proposed his complexity theory, the scientific community was completely unaware of the concept. So they simply rejected him.

The reason that his cryptosystem got rejected is a different case.We can see that his machines can be considered as a linear combination of a rotor machine (permuter) and a linear shift feedback register (adder). By the time when Nash proposed the system NSA had devised the cryptanalysis methods for rotor machines and the linear shift feedback register is a weak key generator. Combination of both will not give a strong cryptosystem

**Conclusion**

The fact that a problem is NP –hard is not enough to make it a good key generator.As in the knapsack system in the average cases it might not be NP-hard. Unfortunately we are not having a cryptosystem which is provably NP-hard. Presently we are depending on the factorization and elliptic curves which are believed to be NP-hard but there are chances that in future these system may also be proven non NP-hard and we will have to move on to new cryptosystems