Synopsis of

Implementation of the Diffie-Hellman Key Exchange

A Project to be submitted by

Sourabh S Shenoy (4NM12CS161)

As a part of CN Task

Department of Computer Science, NMAM Institute of Technology, Nitte, India

March, 2015

1. Introduction

In cryptography it is often required to exchange keys between two parties. To decrypt the message communicate between two parties, both of them need sufficient information to decrypt the message. Either same copy of code is required to maintain at both end or they need appropriate keys. Key exchange protocols are designed so that cryptographic keys can be exchanged between two parties using an algorithm.

There are problems associated with symmetric key. For example,

- If n people want to communicate with one another, there is a need for n(n-1)/2 symmetric keys. This is usually referred as n2 problem. This is acceptable only if n is small.
- In a group of n people, each person must have and remember (n-1) keys. This means if one million people want to communicate with each other, each must remember or store almost one million keys in computer.
- Two parties must be securely acquiring the shared key, it cannot be done over phone or the internet, and these are not secure.

Considering the above problems, a symmetric key between two parties is useful if it is dynamic. The key is created for each session and destroyed when the session is over. It does not have to remember by the two parties. One protocol, the Diffie-Hellman key exchange protocol provides a one-time session key for two parties. The two parties use the session key to exchange data without having to remember or store it for future use. The agreement of a key for the session can be done through the Internet.

The Diffie-Hellman key agreement was discovered in 1976 during collaboration between Whitfield Diffie and Martin Hellman. It was the first practical method for establishing a shared secret over an unprotected communications channel. Ralph Merkle's work on public key distribution influenced this key exchange protocol. John ill suggested use of the discrete logarithm problem. It had first been invented by Malcolm Williamson of GCHQ in the UK some years previously, but GCHQ chose not to make it public until 1997, by which time it had no influence on research in academia. The method was followed shortly afterwards by RSA, another implementation of public key cryptography using asymmetric algorithms. USA patent de scribes the algorithm and credits Hellman, Diffie, and Merkle as inventors.

The original implementation of protocol uses the multiplicative group of integers N, where N is prime and is equal to 2, 4, a power of an odd prime, or twice a power of an odd prime and α is primitive root mod N. N and α are agreed between Party A and party B who are going to exchange keys between them.

2. Feasibility Study

- The system can be integrated with other applications to so that it can be used as a component. Currently the system works as standalone unit.
- The system is based on LINUX. In future it can be enhanced to make it platform independent.
- Digital signature or public key certificates can be utilized to authenticate parties involved in communication.
- The system can be enhanced using thread instead of message queue. Due to system limitation thread mechanism is not implemented.
- Currently server process implemented in the system, handles one request at a time. The process can be improved to handle multiple requests concurrently. Fail over mechanism can be implemented to handle the system in more user friendly way.

- The administrative application of the system solely depends on the authentication process implemented by LIN UX.
- An extra security layer can be implemented in time of execution of admin part.
- The system can be enhanced with storage feature so that administrator has a control over the number assigned to different parties.
- Diffie Hellman Key exchange is vulnerable to man-in-the-Middle attack. To prevent it the implemented system ask user to input IP of another party. But in real life it cannot be assured that user will not use dynamic IP. To get rid of this problem, authentication of parties involved in communication is required.

3. Methodology

Diffie-Hellman key exchange offers the best of both worlds -- it uses public key techniques to allow the exchange of a private encryption key. Let's take a look at how the protocol works, from the perspective of Alice and Bob, two users who wish to establish secure communications. We can assume that Alice and Bob know nothing about each other.

1. Communicating in the clear, Alice and Bob agree on two large positive integers, n and g, with the stipulation that n is a prime number and g is a generator of n.
2. Alice randomly chooses another large positive integer, X_A, which is smaller than n. X_A will serve as Alice's private key.
3. Bob similarly chooses his own private key, X_B.
4. Alice computes her public key, Y_A, using the formula $Y_A = (g \hat{} X_A) \bmod n$.
5. Bob similarly computes his public key, Y_B, using the formula $Y_B = (g \hat{} X_B) \bmod n$.
6. Alice and Bob exchange public keys over the insecure circuit.
7. Alice computes the shared secret key, k, using the formula $k = (Y_B \hat{} X_A) \bmod n$.
8. Bob computes the same shared secret key, k, using the formula $k = (Y_A \hat{} X_B) \bmod n$.
9. Alice and Bob communicate using the symmetric algorithm of their choice and the shared secret key, k, which was never transmitted over the insecure circuit.

4. Bibliography

1. http://www.cc.gatech.edu/classes/cs8113e_96_winter/ - Visited on 03/16/2015 – college website
2. http://www.cryptography.com/- Visited on 03/08/2015
3. http://www.cs.purdue.edu/homes/jiangx/02spring/- Visited on 02/18/2015
4. http://www.securitydocs.com - Visited on 02/14/2015 – Network Security White papers
5. http://www.sans.org/rr/whitepapers/vpns/751.php - Visited on 02/17/2015 – SANS Institute 2001, a paper on the Diffie – Hellman Algorithm and its use in secure Internet protocols.