International Research Journal of Computer Science (IRJCS)
Volume 1 Issue 3 (November 2014)
ISSN: 2393-9842
www.irjcs.com

Key Management based Multilevel Security Using Digital Signature and Encryption Techniques

Ritu Makani
Department of Computer Science &Engg.
GJUS&T Hisar (Haryana)

Yogesh Chaba
Department of Computer Science &Engg.
GJUS&T Hisar (Haryana)

Abstract: ***The challenges for implementing security to network is to address the major issues related to it like privacy, data security, confidentiality and authentication. A variety of encryption algorithms and techniques in combination have been tried to achieve this target. Managing keys for implementation of the security in network systems in an effective and efficient manner is one of them. But computation and distribution of cryptographic keys was a problem for a long. Diffie-Hellman gave the practical solution to compute and exchange cryptographic keys. It is designed to provide users to share a secret key that can be used for encryption of messages between them securely. In this paper a three way mechanism to ensure all three protection schemes of authentication, data security and verification to provide multilevel security has been proposed. It is proposed to make use of digital signature and Diffie Hellman key exchange blended with RSA encryption algorithm to provide data confidentiality. If the key is hacked during transmission then key exchange algorithm will make it useless because key in transit will be of no use without user's private key which is only with the authorised user. This combination provides a tough multilevel security model which is very effective and hard to crack.***

Key Words: Cryptography, Key Management, Digital Signature, RSA, Diffie Hellman.

I.INTRODUCTION:

During information and data transfer there can be attacks in the form of interruption, interception, modification or fabrication. These attacks are due to some action which compromises the information security. Security services enhance the security of the data processing and transferring. Security mechanisms are for detecting, preventing and recovering from a security attack. Important features of security are confidentiality, authentication, integrity, availability and non-repudiation etc[1]

Cryptography is the study of secret writing which is related to

i) Developing algorithms to conceal the context of some message from all except the sender and the recipient to invoke privacy or secrecy
ii) Verify the correctness of the message to the recipient which is called authentication.

These two form the basis of many technological solutions to network security problems. So simply speaking it is the art of transforming an intelligible message into the one which is unintelligible and vice versa. Cryptography or encryption and decryption methods fall into two main Categories: symmetric and public key. In symmetric cryptography, also called classical cryptography, parties share the same encryption/decryption key. Therefore, before using a symmetric cryptography system, the users must somehow come to an agreement on a key to use. An obvious problem arises when the parties are separated by large distances which are commonplace in today's worldwide digital communications. If the parties did not meet prior to their separation, how do they agree on the common key to use in their cryptosystem without a secure channel? They could send a trusted courier to exchange keys, but that is not feasible, if time is a critical factor in their communication [5].

The distribution of keys used in symmetric ciphers was a problem for cryptographers for a long time. If an unauthorized user gain access to the key, the cryptographic communication can be considered broken. Whitfield Diffie and Hellman , presented a key exchange protocol that provided the first practical solution to this dilemma. This protocol was named the Diffie-Hellman key exchange protocol. By virtue of this protocol two parties agree to derive a common secret key by communications over an unsecured channel, while sharing no secret keying material at prior[5] Before conducting the key exchange using the Diffie-Hellman protocol, the parties must agree on a prime number that defines the mathematical environment in which the key exchange will take place. If the prime number is large enough, a brute force attack to find the secret key becomes infeasible. However, if the two parties agree on certain prime numbers, an active adversary can compromise their communication [2][3]. A lot of research work has been done in the area of network security using digital signature and encryption techniques. Research paper Rewagad et.al discussed the Use of Digital Signature with Diffie-Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing[1]. Omura discussed the alternatives to RSA using Diffie-Hellman with digital signature standard [2]. Oorschot et al. explained Diffie-Hellman Key Agreement with short exponents [3]. Yogesh Chaba et al. has analysed the Performance of disable IP broadcast Technique for prevention of flooding based DDoS Attack in Manet[6]. Yudhvir Singh et al. performed Information theory tests based performance evaluation to cryptographic techniques [7]. Work in the area of network security can also be done by implementing key management based multilevel security using digital signature and encryption techniques.

II. DIFFIE-HELLMAN AND DSA PROTOCOL

The parameters of the protocol are: *p*, a large prime and *g*, a primitive element of Z_n. This means that all numbers *n*=1, ... , *p*-1 can be represented as $n = g^i$. These two numbers do not need to be kept secret. For example, Alice could send them to Bob in the open. The protocol runs as follows:

1. Alice choses a large random integer *x* and sends Bob
   $X=g^x \bmod p$
2. Bob choses a large random integer *y* and sends Alice
   $Y=g^y \bmod p$
3. Alice computes
   $k=Y^x \bmod p$
4. Bob computes
   $k=X^y \bmod p$

k is the key. *k* is equal to $g^{xy} \bmod p$.

Both Alice and Bob have arrived at the same value, because $(g^a)^b$ and $(g^b)^a$ are equal mod *p*. Note that only *a*, *b*, and $(g^{ab} \bmod p = g^{ba} \bmod p)$ are kept secret. All the other values – *p*, *g*, $g^a \bmod p$, and $g^b \bmod p$ – are sent in the clear. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel.[3]

There are two types of problems with Diffie-Hellman key exchange algorithm, the computational and the decisional. *The **Computational Diffie-Hellman Problem*** is defined as follows: Let p be a prime and let α be a primitive root mod p . Given $\alpha^x \pmod p$ and $\alpha^y \pmod p$, find $\alpha^{xy} \equiv \beta \pmod p$. Recall that Eve has access to both α^x and α^y as they are both made public during the exchange. It is not currently known whether or not this problem is easier than computing discrete logs A related problem, known as the ***Decisional Diffie-Hellman Problem***, is defined as follows: Let p be a prime and let *α*be a primitive root mod p . Given $\alpha^x \pmod p$ and $\alpha^y \pmod p$, and $\beta \neq 0 \pmod p$, decide whether or not $K \equiv \alpha^{xy} \pmod p$. In other words, if someone offers a number to Eve and claims it is K, can Eve decide whether or not that person is telling the truth with the information captured in the open channel? Solving these problems Eve can attack the Diffie-Hellman Key Exchange protocol. It may either pretend to be sender or it may alter the messages between the two clients also, it may simply hear to the conversation and compromise the privacy of the communication[4][5]

The most famous of the public key cryptosystem is RSA which is named after its three developers Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA cryptosystem is a public-key cryptosystem, widely used for secure communication and e-commerce applications. It is often used to encrypt messages sent between two communicating parties so that an eavesdropper who overhears the conversation cannot decode them easily. It also enables a party to append an unforgeable signature to the end of a message. This signature cannot be easily forged and can be checked by anyone.

The basic RSA cryptosystem is completely specified by the following sequence of steps.

1. Alice selects at random two large primes p and q.
2. Alice computes n = pq.
3. Alice selects a small odd integer e that is relatively prime to $(p - 1)(q - 1)$.
4. Alice sets d so that $de \bmod (p - 1)(q - 1)$ equals 1.
5. Alice publish the pair (e, n) as the public key, with $P_A(M) = M^e \bmod n$.
6. Alice stores the pair (d, n) as the secret key, with $S_A(E) = E^d \bmod n$.

In order to send message M in $\{0, 1, \ldots, n-1\}$, Bob sends $P_A(M) = M^e \bmod n$. On receiving the encrypted message Alice computes $S_A(P_A(M)) = M^{de} \bmod n$. Our choices of d, e, and n ensure that $M^{de} \bmod n$ equals M.

Digital Signature Algorithm (DSA) **:** DSA takes three parameters called Global Public Key Components :

p = prime number where $2^{L-1} < p < 2^L$ for $512 < L < 1024$ and L is a multiple of 64 : i,e between 512 and 1024 bits in increments of 64 bit

q = prime divisor of (p-1), where $2^{159} < q < 2^{260}$; i,e bit length of 160 bits .

g = $h^{(p-1)/q} \bmod p$, where h is any integer with $1 < h < (p-1)$ such that $h^{(p-1)} \bmod p > 1$

User's Private Key

x = Random or pseudo random iteger with $0 < x < q$

User's Public Key

$y = g^x \bmod p$

User's per message Secret Number

k = Random or pseudo random integer with $0 < k < q$

To create a signature a user calculates two quantities, 'r' and 's' , that are functions of the public key components (p,q,g), the users private key(x), the Hash code of the message H(M) and an additional integer K that should be generated randomly or pseudo randomly and be unique for each signing. At the receiving end verification is performed using the formulas as shown below:-

Signing :

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1} (H(M) + x^r)] \bmod q$

Signature = (r, s)

Verifying :

$w = (S')^{-1} \bmod q$

$u_1 = [(H(M') w)] \bmod q$

$u_2 = (r') w \bmod q$

$v = [(g^{u1} y^{u2}) \bmod p] \bmod q$

Test

M = Message to be signed

H(M) = Hash of M

M' , r', s' = Received versions of M,r,s

The receiver generates a quantity 'v' that is a function of the public key components, the senders public key , and the Hash code of the incoming message. If this quantity matches thr 'r' component of the signature, then the signature is validated. The test at the end is on the value 'r' , which does not depend on the message at all. Instead, 'r' is a function of K and the three global public ket components. The multiplicative inverse of k(mod q) is passed to the function that also has as input the message Hash code and user's private key.[6] The structure of this function is such that the receiver can recover 'r' using the incoming message and signature, the public key of the use and the global public key. Given the difficulty of taking discrete logarithms, it is infeasible for an opponent to recover k from 'r' or to recover x from 's'. The only computationally demanding task in signature generation is the exponential calculation g^k md p , because this value does not depend upon the message to be signed and it can be calculated ahead of time. A user could pre calculate a number of values of 'r' to be used to sign documents as needed. Similarly determination of the multiplicative inverse k^{-1} is another demanding task and number of these values can be pre calculated.

III. PROPOSED WORK

A. Applying Digital Signature with RSA encryption on Diffie-Hellman key exchange algorithm:

The idea of applying **Digital Signature** using **RSA** is that 'f' is function that is known to everyone , but only you know your decryption function. As shown in figure, in order for Alice to sign a message, m, sends $g_A(m)$ together with an indication that the message is from Alice. When Bob gets it he sees that the message is from Alice & applies his public encryption function, f_A to $g_A(m)$ and will get m purportedly from Alice to Bob, he has no way of being successful since he does not know g_A . That means that you can sign a message without encrypting it. In the scheme described in this section, anyone can intercept Alice's signed message and read it because her public key is known. Applying encryption in addition to signing a message is quite simple, If Alice wants to sign and encrypt a message, he can do it as shown in the following sequence :

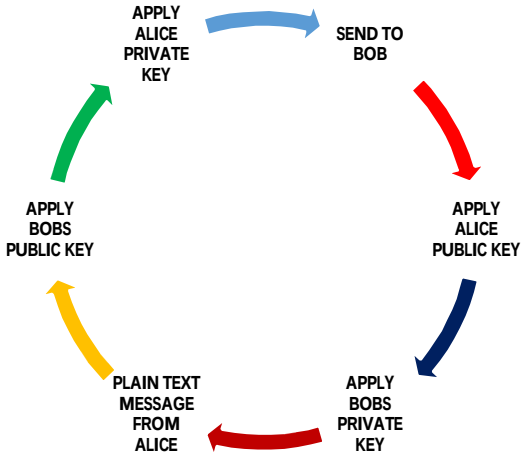


Fig.1 CORRECT ORDERING

This implies that when Bob applies Alice's public key to what is received, the result is

$$f_A (g_A (f_B (m))) = f_B (m)$$

then when Bob applies his private key he sees

$$g_B (f_B (m)) = m$$

The order of operations for the simultaneous encryption and signing is ::

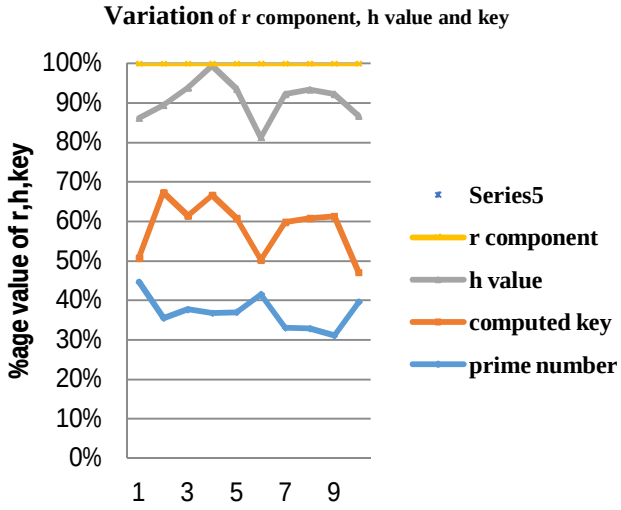
ENCRYPT → SIGN → SEND→ AUTHENTICATE→ DECRYPT

B. Results

Table 1 and Graph1 shows the computed key and r components of the digital signature for given prime numbers.RSA involves modular exponentiations working on and yield big integers of the same size as modulus therefore RSA based signature with RSA key will necessarily include 1024 bit integer. Following table and graph shows the computed values of the modular exponents of the keys with large integers.

TABLE I Computed key & "r" component of the Digital Signature

Prime Number	Computed key	h value	'r' Component
29	4	23	9
37	33	23	11
43	27	37	7
53	43	47	1
103	66	91	18
293	61	219	132
337	273	331	79
557	471	551	113
997	963	991	248
65537	12365	65531	21949



GRAPH I Percent values of the Computed Keys and "r" for given Prime Numbers

The graph shows that with the prime number increase in the range of 37% to 43% computed key shows an increase from 50% to 67%.

IV. **CONCLUSION**

RSA and Diffie-Hellman are both based on intractable problems, the difficulty of factoring large numbers and exponentiation and modular arithmetic respectively. The nature of the Diffie-Hellman key exchange does make it susceptible to man-in-the-middle attacks since it doesn't authenticate either party involved in the exchange. This is why Diffie-Hellman is used in combination with an additional authentication method, generally digital signatures. When using RSA, a 1,024-bit key is considered suitable both for generating digital signatures and for key exchange when used with bulk encryption, while a 2048-bit key is recommended when a digital signature must be kept secure for an extended period of time. such as a certificate authority's key. The graph shows that with the prime number increase in the range of 37% to 43% computed key shows an increase from 50% to 67%.

REFERENCES

[1] Rewagad, Parshant, Yogita *"Use of Digital Signature with Diffie-Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing"* IEEE April,62013,978-0-7695-4958/13
[2] Dr.Jim Omura *"Alternatives to RSA: Using Diffie-Hellman with DSS"* CYUNK Resource Library White Papers
[3] P. C. van Oorschot and M. J. Wiener, On *"Diffie-Hellman Key Agreement with Short Exponents. EUROCRYPT'96*, LNCS 1070, Springer-Verlag, 1996, pp. 332–343.
[4] A. J. Menezes, P. C. van Oorshot, and S. A Vanstone, *Handbook of Applied Cryptography*. CRC Press, New York, New York, 1997.
[5] Geary, Aaron C." Analysis of a man-in-the-middle attack on the Diffie-Hellman key exchange protocol", California. Naval Postgraduate School,Issue 2009-09
[6] Yogesh Chaba, Yudhvir Singh, P Aneja "Performance Analysis of disable IP broadcast Technique for prevention of flooding based DDoS Attack in Manet" *Journal of Networks,* Vol. 4(3), pp. 178-183, 2009
[7] Yudhvir Singh, Yogesh Chaba "Information theory tests based performance evaluation to cryptographic techniques" *International Journal of Information Technology, No.* 1(2), pp. 475-483, 2008