An Approach for SMS Security using Authentication Functions

Neetesh Saxena
Department of Computer Sc. & Engineering
Indian Institute of Technology, Indore, India

Narendra S. Chaudhari
Department of Computer Sc. & Engineering
Indian Institute of Technology, Indore, India

ABSTRACT
Asymmetric algorithm like Diffie-Hellman can be used to encrypt the SMS message in M-commerce or mobile banking system. Password key exchange protocol based on Diffie-Hellman key exchange algorithm allows users to exchange a secret key that can be used in message encryption. The security of this protocol can be increased by using the MAC (message authentication code) or hash function with the encryption. These functions act as an error detecting code or checksum. This paper throws a light on the comparative analysis of both the authentication functions separately in password key exchange protocol. By analyzing some of the security issues viz. (i) brute force attach and (ii) cryptanalysis, it can be very well shown that the MAC function is more secure than hash.

General Terms
SMS Security, Algorithms et. al.

Keywords
GSM, SMS security, authentication function, public key cryptography

1. INTRODUCTION
Principles of Public-Key cryptosystems:

Public key cryptography technique is also known as asymmetric technique. The basic aim of public-key cryptography is to provide a helping hand to minimize, an attempt to attack on the two functionalities of symmetric technique: (i) Key distribution and (ii) Digital signature (both authentication and confidentiality). Diffie and Hellman achieved a solution that addressed both problems.

2. PRILIMINARIES
Before we move into the broader concept let us have a brief review of some basic facts.

2.1 Diffie-Hellman Key Exchange
Diffie and Hellman developed a public key cryptography algorithm known as Diffie-Hellman key exchange. This technique allows two users to exchange a key securely that can be used for the encryption of message. A brief description of this algorithm is given here:

Global Elements

p = Prime number

n = A primitive root of p and $n < p$

'User A' Key Generation

Select private key a where ($a < p$)

Calculate public key x where ($x = n^a \bmod p$)

'User B' Key Generation

Select private key b where ($b < p$)

Calculate public key 'y' where ($y = n^b \bmod p$)

Generation of Secret Key by User A

$$k = y^a \bmod p$$

Generation of Secret Key by User B

$$k = x^b \bmod p$$

2.2 Authentication Functions
An authentication and digital signature mechanism has two parts: first part provides some functions for authentication (that produces an authenticator) by the sender and the second part enables receiver to verify the authenticity of the message. These functions are:

1. Message encryption
2. Message authentication code (MAC)
3. Hash function

    1. **Message encryption:** The encrypted form of message i.e. the ciphertext works as its authenticator.
    2. **Message authentication code (MAC):** Message is encrypted by a public function (a MAC function) and a secret key that generates a fix length code.
    3. **Hash function:** Message is encrypted by a public function (form of a hash function) that converts a message of any length into the fixed length code.

Assume that:
User A
Private key: a
Public key: x

User B
Private key: b
Public key : y

(i) Encryption: confidentiality, authentication and signature

$$A \rightarrow B : E_y[E_a(M)]$$

(ii) MAC Encryption: confidentiality and authentication

$$A \rightarrow B : E_{k_2}[M] \| C_{k_1}[E_{k_2}(M)]$$

Where k1 and k2 are the two shared secret keys.

(iii) Hash Function Encryption: confidentiality and authentication

$$A \rightarrow B : E_k [M \| H(M) \| N]$$

Where k is the shared secret key and N is the shared secret value between A and B.

3. PASSWORD KEY EXCHANGE PROTOCOL BASED ON PUBLIC ENCRYPTION

Let us assume that there are two parties A and B respectively, who want to communicate with each other. They can share only the passwords of each other i.e. a calculated public key. Various steps of communication are as follows:

1. Both User A and B know their private keys (a random number) 'a' and 'b' respectively.
2. User A calculates its password (public key) 'x' as [$x = n^a \bmod p$].
3. User B calculates its password (public key) 'y' as [$y = n^b \bmod p$].
4. User A sends its IDA and password 'x' to User B.
5. User B sends its IDB and password 'y' to User A.
6. User A generates a shared secret key 'k' by password of User B 'y' and its private key 'a'.

   $k = y^a \bmod p$ = $[n^b \bmod p]^a \bmod p$ = $n^{ab} \bmod p$
7. User B generates a shared secret key 'k' by password of User A 'x' and its private key 'b'.

$k = x^b \bmod p$ = $[n^a \bmod p]^b \bmod p$ = $n^{ab} \bmod p$

It is difficult for attacker to guess the passwords of User A and User B. This protocol depends upon the effectiveness of computing discrete logarithms. Thus the attacker is forced to calculate the discrete logarithm to determine the key. The security of this protocol depends on the fact, that, it is relatively easy to calculate exponential modulo of a prime number while it is very difficult to calculate the discrete logarithms and for a large number it is almost infeasible to calculate discrete logarithm.

4. PASSWORD KEY EXCHANGE PROTOCOL BASED ON MAC ENCRYPTION

Again let us assume that there are 2 shared keys k1 and k2 respectively that are used by both users.

Various steps of communication followed in MAC encryption are as follows:

1. User A and User B know their private keys (a random number) 'a' and 'b' respectively.
2. User A calculates its password (public key) 'x' as [$x = n^a \bmod p$] and $C_k(passowrd)$ using a MAC function/algorithm.
3. User B calculates its password (public key) 'y' as [$y = n^b \bmod p$] and $C_k(passowrd)$ using a MAC function/algorithm.
4. User A sends its IDA and $E_{k2}[(password) \| C_{k1}(passowrd)]$ to User B.
5. User B sends its IDB and $E_{k2}[(password) \| C_{k1}(passowrd)]$ to User A.
6. User A generates a shared secret key 'k' by the password of User B and its private key 'a'. {use $D_{k2}[(password) \| C_{k1}(passowrd)]$ to get the password and use k_1 shared key to calculate $C_{k1}(password)$ and match with the actual $C_{k1}(password)$ send by User B to detect the error}

   $k = y^a \bmod p$ = $[n^b \bmod p]^a \bmod p$ = $n^{ab} \bmod p$
7. User B generates a shared secret key 'k' by the password of User A and its private key 'b'. {[use $D_{k2}[(password) \| C_{k1}(passowrd)]$ to get the password and use k_1 shared key to calculate $C_{k1}(password)$ and match with the actual $C_{k1}(password)$ send by User A to detect the error}.

   $k = x^b \bmod p$ = $[n^a \bmod p]^b \bmod p$ = $n^{ab} \bmod p$

But MAC does not provide digital signature because it uses shared keys.

5. PASSWORD KEY EXCHANGE PROTOCOL BASED ON HASH FUNCTION ENCRYPTION

This protocol uses a hash function. The basic purpose of hash function is to provide authentication by generating a hash code value (like fingerprints). It is easy to calculate H(M) for a given message M. There are three main properties of hash function:
1. One way property: For a given value h, it is computationally infeasible to find M such that H(M) = h.
2. Weak collision resistance: For a given block M, it is computationally infeasible to find N ≠ M such that H(N) = H(M).
3. Strong collision resistance: It is computationally infeasible to find any pair (M, N) such that H(M) = H(N).

The difference of this protocol with password key exchange public encryption protocol is that here User A and User B send

H(password) instead of only password. Various steps of communication by hash function encryption are as follows:

1. User A and User B know their private keys (a random number) 'a' and 'b' respectively.
2. User A calculates its password (public key) 'x' as $[x = n^a \bmod p]$ and H(password) using a hash function.
3. User B calculates its password (public key) 'y' as $[y = n^b \bmod p]$ and H(password) using a hash function.
4. User A sends its ID_A, $E_k[(password) \| E_{kRa}[H(password)]]$ to User B.
5. User B sends its ID_B, $E_k[(password) \| E_{kRb}[H(password)]]$ to User A.
6. User A generates a secret key 'k' by the password of User B and its private key 'a'. {Use $D_k[(password) \| E_{kRa}[H(password)]]$ to get the password and use public key of User A 'kU_a' to decrypt the H(password). Now calculate H(password) and match with the decrypted H(password) send by User B to detect the error.}
   $k = y^a \bmod p$ = $[n^b \bmod p]^a \bmod p$ = $n^{ab} \bmod p$
7. User B generates a secret key 'k' by the password of User A and its private key 'b'. {Use $D_k[(password) \| E_{kRb}[H(password)]]$ to get the password and use public key of User B 'kU_b' to decrypt the H(password). Now calculate H(password) and match with the decrypted H(password) send by User A to detect the error}
   $k = x^b \bmod p$ = $[n^a \bmod p]^b \bmod p$ = $n^{ab} \bmod p$

6. COMPARISON

Both MAC and hash values are used as error detecting code or checksum. The difference between a one-way hash and a MAC (Message authentication code), is that the hash verifies the uniqueness of a message and the MAC is usually an encrypted hash, also used to verify the uniqueness of a message, but which only can be verified if you know the secret key.

We can analyze the security of both functions by two major cryptographic terms:

(i) Brute force attack and, (ii) cryptanalysis.

MAC uses a secret key to generate a MAC code but hash function doesn't use any key. In the brute force attack all the possible combination of the key are applied. Suppose if the key is of n-bit size then the brute force attack is 2^n. In MAC, even if an attacker gets the MAC value, he can't get the original message unless he knows the secret key. Thus we can say that, the MAC encryption is more secure than the hash encryption.

Now we consider another security aspect of cryptography i.e. Cryptanalysis. We can analyze the structure of hash function and the computational complexity of different attacks applied on hash function by [2].

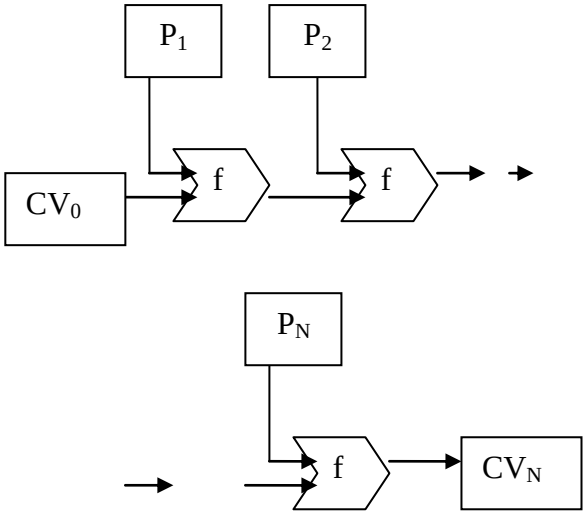

Figure 1: A simple hash function

Basically cryptanalysis of hash function is based on the internal structure of function 'f' that is used in hash algorithm.

Hash function can be summarized as follows:

$CV_0 = IV$ (initial n-bit value)

$CV_i = f(CV_{i-1}, P_{i-1})$

H(M) = CV_N

To study the MAC function and its structure we can refer [5]. A simple MAC function is shown in the figure 2. Here,

$C_1 = E_k(P_1)$

$C_2 = E_k(P_2 XOR C_1)$

……..

$C_N = E_k(P_N XOR C_{N-1})$

Here, in this MAC function, DES is used as the encryption algorithm and the encryption is done by a secret key 'K'. As there is a vast variety in the structure of MAC function, so it is very difficult to apply cryptanalysis on it.

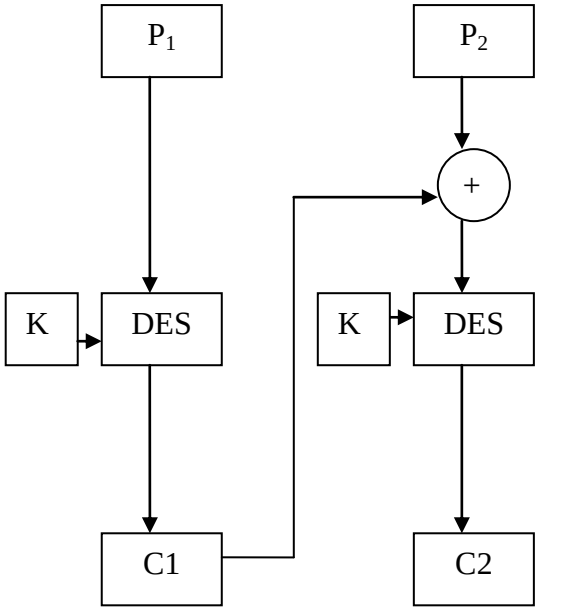

Figure 2: A simple MAC function

Further, far less work has been done on developing cryptanalysis of MAC.

7. CONCLUSION AND FUTURE WORK

At the end it can be concluded that out of these three authentication functions hash function and MAC function are more secure. The comparative analysis states that the MAC functions are more secure than hash functions because it is more difficult to apply brute force attack on MAC functions than hash functions and it is also difficult to generalize the cryptanalysis of MAC functions as it has large variety of its structure. So if we apply MAC over the password key exchange protocol then it'll be more secure.

A lot of work has to be done on MAC. So it is a research - oriented area related to the cryptanalysis of MAC function's internal structure and its various varieties.

8. REFERENCES

[1] Steven M. Bellovin, Michael Merritt "Augmented Encrypted Key Exchange: A Password-Based Protocol Secure against Dictionary Attacks and Password File Compromise" (1993).

[2] Xun Yi and Kwok Yan Lam "Hash function based on block cipher" IEE 1997 Electronics Letters Online No: I9971336.

[3] Luo Zhong Zhao Zhongining Zhu Chongguang "The Unfavourable Effects of Hash Coding on CMAC Convergence and Compensatory Measure" Instilutc of Rcmotc Scnsing Applications .CAS Dept Image Processing.P 0.Box 9718 Beijing china.

[4] H.E. Michail, A.P. Kakarountas, G. Selimis, C.E. Goutis "Throughput Optimization of the Cipher Message Authentication Code" VLSI Design Laboratory, Dpt. of Electrical & Computer Engineering, University of Patras, Greece.

[5] C.J. Mitchell "Truncation attacks on MACs" IEE 2003 Electronics Letters Online No: 20030921DOI: 10.1049/el: 20030921.