Security Enhancement of Forward Digital Signatures Using ECC

[1]M.Rajasekhar, [2]I.M.V.Krishna, [3]M.Samuel John

Department of Computer Science Engineering
PVP Siddhartha Institute of Technology
Vijayawada, India

***Abstract*— In this paper we propose a technique to enhance the security of forward digital signature.In this enhancement scheme the private key and the public key changes at random intervals of time ,if there is a communication between two users. If there is no communication then the keys will not be changed. This technique is mainly based on elliptic curve cryptography and Digital signature algorithm. By using this enhancement scheme the attacker cannot get the older or future signatures even if the private key is compromised.The attacker can only know the signature at that session only.This scheme is more secure and it is proved practically.**

Keywords:security,signature,elliptical curve cryptography

I. INTRODUCTION

In modern day scenario, Security is the most important aspect when we want to transfer secret information from one place to another place. The digital signature plays an important role to guarantee the secured communication. There is less security when we want to transfer data from one place to another place when there is one private key and one public key. Because if at particular moment of time if any private key is leaked out then the attacker can fake the signature and can send irrelevant messages to others. To avoid this problem we are using an elliptic curve digital signature algorithm to provide more security. By using this concept the first step to do is to establish a tcp connection between two users. When ever once the tcp connection is established then the private key changes at random intervals of time and the public key is updated after the private key updation.After completion of communication then the tcp connection must be closed, so that the private key and public key will not change. By using this concept if an attacker find the private key at some particular session he can know only that digital signature, he cannot know the future and past digital signatures. The main theme of elliptic curve cryptography lies in finite fields of the form GF (2n)[1].

II. CONCEPT OF FORWARD DIGITAL SIGNATURE

An enhancement forward digital signature scheme is, first of all, a key-evolving digital signature scheme [2]. A key-evolving signature scheme contains a key generation algorithm, a signing algorithm, and a verification algorithm. The public key is also changed with the private key throughout the lifetime of the scheme, making the verification algorithm very similar to that of a standard signature scheme. Unlike a standard signature scheme, a key-evolving signature scheme has its operation divided into time periods, each of which uses a different secret key to sign a message. The way these keys are updated is given by a public update algorithm, which computes the secret key for the new time period based on that for the previous one. The forward security comes, in part, from the fact that this update function is one-way and, given the secret key for the current period, it is hard to compute any of the previously used secret keys. It is important, of course, for the signer to delete the old secret key as soon as the new one is generated, since otherwise an adversary breaking the system could easily get hold of these undeleted keys and forge messages for time periods preceding that of the break-in.

III. FINITE FIELDS OF THE FORM GF (2n)

Here GF stands for galois field.A field is a set that obeys some axioms.There are finite fields and infinite fields .The fininte fields plays an important role in cryptogarphic algorithms.The fininte fields must be of the form p^n ,Where p=2 and n is a prime number.The follwing below table shows addition by using finite fields of the form where n=3.This is main thing in elliptic curve cryptography.By using ECC we have following advantages they are small key size, greater speed and less storage. So for that purpose we are using mainly using ECC.

Table I. Addition in GF(2^3)

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

IV. ESTABLISHING TCP CONNECTION

The "three-way handshake" is the procedure used to establish a connection [3]. This procedure normally is initiated by one TCP and responded to by another TCP. The procedure also works if two TCP simultaneously initiate the procedure. When simultaneous attempt occurs, each TCP receives a "SYN" segment which carries no acknowledgment after it has sent a "SYN". The simplest three-way handshake is shown in figure 2 below. The figures should be interpreted in the following way. Each line is numbered for reference purposes. Right arrows (-->) indicate departure of a TCP segment from TCP A to TCP B, or arrival of a segment at B from A. Left arrows, indicate the reverse. Ellipsis (...) indicates a segment which is still in the network (delayed). An "XXX" indicates a segment which is lost or rejected. Comments appear in parentheses. TCP states represent the state AFTER the departure or arrival of the segment (whose contents are shown in the center of each line).

TCP A TCP B

1. CLOSED LISTEN
2. SYN-SENT --> <SEQ=100><CTL=SYN> --> SYN-RECEIVED
3. ESTABLISHED <SEQ=300><ACK=101><CTL=SYN, ACK> <-- SYN-RECEIVED
4. ESTABLISHED --> <SEQ=101><ACK=301><CTL=ACK> --> ESTABLISHED
5. ESTABLISHED --> <SEQ=101><ACK=301><CTL=ACK><DATA> --> ESTABLISHED

Figure 1. Basic 3-Way Handshake for Connection Synchronization

In line 2 of figure 1, TCP A begins by sending a SYN segment indicating that it will use sequence numbers starting with sequence number 100. In line 3, TCP B sends a SYN and acknowledges the SYN it received from TCP A. Note that the acknowledgment field indicates TCP B is now expecting to hear sequence 101, acknowledging the SYN which occupied sequence 100. At line 4, TCP A responds with an empty segment containing an ACK for TCP B's SYN; and in line 5, TCP A sends some data.

V. CLOSING TCP CONNECTION

CLOSE is an operation meaning "I have no more data to send." The notion of closing a full-duplex connection is subject to ambiguous interpretation, of course, since it may not be obvious how to treat the receiving side of the connection. We have chosen to treat CLOSE in a simplex fashion. The user who CLOSE may continue to RECEIVE until he is told that the other side has CLOSED also. Thus, a program could initiate several SEND followed by a CLOSE, and then continue to RECEIVE until signaled that a RECEIVE failed because the other side has CLOSED. We assume that the TCP will signal a user, even if no RECEIVE are outstanding, that the other side has closed, so the user can terminate his side gracefully. A TCP will reliably deliver all buffers SENT before the connection was CLOSED so a user who expects no data in return need only wait to hear the connection was CLOSED successfully to know that all his data was received at the destination TCP. Users must keep reading connections they close for sending until the TCP says no more data. There are essentially three cases:

1) The user initiates by telling the TCP to CLOSE the Connection

2) The remote TCP initiates by sending a FIN control signal

3) Both users CLOSE simultaneously

TCP A TCP B

1. ESTABLISHED ESTABLISHED
2. (Close) FIN-WAIT-1 --> <SEQ=100><ACK=300><CTL=FIN, ACK> --> CLOSE-WAIT
3. FIN-WAIT-2 <-- <SEQ=300><ACK=101><CTL=ACK> <-- CLOSE-WAIT
4. (Close) TIME-WAIT <-- <SEQ=300><ACK=101><CTL=FIN, ACK> <-- LAST-ACK
5. TIME-WAIT --> <SEQ=101><ACK=301><CTL=ACK> --> CLOSED
6. (2 MSL)

CLOSED

Figure 2. Normal Close Sequence

VI. DIGITAL SIGNATURE ALGORITHM

Consider finite fields of the form $GF(2^n)$.Select a rational point T on $E(GF(2^n)$ and name that as a base point ,where n is a prime number.Select n in such a way that nT=0.Consider the private key for each for each system used is Pr,then public key PU=PRT.Select a secure hash function h(m)[4].If a user want to sign a message m then he should follow the below steps.

a. First user A selects an integer where $0<c<n$ then cT=(a,b), where y= a mod n, if r=0 return to a.

b. Calculate e = h(m),Now calculate the value of $w=c^{-1}$ (e+yPR) mod n, if w=0 return to a.

c. .Now consider the digital signature of the message m as (y,w) by A.

VII. HOW TO VERIFY THE DIGITAL SIGNATURE

a. Compute the value for $e_1=h(m1)$ and also compute the values for I and j where $i=w^{-1} e_1$ mod n , $j= w^{-1\ y}$ mod n.[5]

b. Compute the value for z where $z=iT+jPu_{0=}\ w^{-1}$ $(e_1T+yPrT)= w^{-1} (e_1+yPr)T=cT=(a_{1,} b_1)$.

c. If the result of z is equal to zero, then the signature is denied, other wise compute the value of y_1 where $y_1= a_1$ mod n, if $y= y_1$ then it will accept the signature.

VIII. ENHANCING SECURITY OF FORWARD DIGITAL SIGNATURE

In this enhancement it mainly consists of

a. The key generation algorithm is a probabilistic algorithm which takes as input a security parameter and the total number of periods T and returns a pair (PR, PK), the initial secret key and the public key.

b. The signing algorithm, takes as input the secret key PRj for the current time period and the message M to be signed and returns a pair (j , sign j) the signature of M for time period j.

c. Secret key update algorithm, takes as input the secret key for the current period PRj and returns the new secret key PRj+1 for the next period.

d. verification algorithm, takes as input the public key PU, a message M, and a candidate signature (j,sign j) and returns 1 if (j ,sign j) is a valid signature of M .

IX. UPDATING OF PRIVATE KEY AND PUBLIC KEY

$PR_i = PR^{\ j}{}_{i-1}$ mod n where j is some random number [6]

$W_1=EF\ PR^{\ j}{}_0$ mod n Where E, F are two random numbers

$W_2=E_2F^{-1}{}_1\ PR^{\ j-1}{}_1$ mod n = $E_2F^{-1}{}_1\ PR^{\ j(j-1)}{}_0$ mod n = $E_2F^{-1}{}_1 (E_2\ F^{-1}{}_1 W_1)^{\ j}$ mod n

$PU_1=E_1F_1T = E_1 E_1^{-1} E_0 F^{\ q}{}_0T= E_1 E_0^{q-1} E_0 F^{\ q-1}{}_0\ PU_0 = W_1PU_0$

$PU_2=E_2F_2T = E_2 E_1{}^{-1} E_1 F_1{}^{\ q} T= E_2 E_1^{-1} F_1{}^{\ q-1} E_1F_1T =E_2 E_1^{-1} F1^{\ q-1}\ PU_1 =W_2PU_{1=}\ W_2\ W_1\ PU_0$

CONCLUSION

In this technique we chose ECC because it has advantages like small key size, greater speed and less storage. This is very secure method and in performance point of view also it is good. Because by using this method the attacker could not fake out the older or future signature even if he gets the present private key.

REFERENCES

[1] William Stallings, Cryptography and network security : Principles and Practices Fourth Edition.

[2] Bellare M. Practice-oriented provable security.In: Damgard I,ed.Advances in cryptology Eurocrypt'99.LNCS 1561, Berlin:spinger – verlag,1999,pp.221-231

[3] W.Diffie , P.van Oorschot , " Authentication and authenticated key exchange ", 1998, pp. 271–350.

[4] S.Vanstone,Responses to NIST 's proposal.Communications of the ACM,35:50-52, July 2002

[5] Johnson D,Menzes A.The elliptic curve digital signature algorithm,CORR 99-31.

[6] S.Michali , " A secure and efficient digital signature algorithm", ACM,40:45-50, Aug 2005.