Secure Signcryption on Hyper Elliptic Curve with Sensor-Based Random Number

J.PREMALATHA[1], K.SATHYA[2] AND VANI RAJASEKAR[3]
[1]Professor, Department of Information Technology, Kongu Engineering College, INDIA
[2]PG Scholar, Department of Information Technology, Kongu Engineering College, INDIA
[3]PG Scholar, Department of Information Technology, Kongu Engineering College, INDIA
[1]jprem@kongu.ac.in; [2]pearlhoods@gmail.com; [3] vanikecit @gmail.com

Abstract:- The emerging trend now in network security is efficient signcryption which is due to the growth of wireless technology. Signcryption scheme is proposed with security features such as confidentiality, integrity(originality of sender and receiver), message and user authentication, non-repudiation, forward secrecy and public verifiability. Signcryption fulfils the functions of signature and encryption in one logical. Strength of security and privacy of any cryptographic mechanisms that use random numbers require that the random numbers generated have two important properties namely 1.Uniform distribution and 2.Independence [9]. One idea proposed is to use sensor data as seed for Random Number Generator (RNG) to generate the random numbers that is used for signcryption algorithm in wireless networks [8]. These sensor data also pose weaknesses where sensors may be under adversarial control that may lead to generating expected random sequence which breaks the security and privacy. This paper proposes an approach to process the raw sensor data that increases randomness in the seed value and points out the limitations of existing signcryption scheme such as lack of forward secrecy and public verifiability, computation and communication overhead, larger memory requirements [1]. The proposed scheme based on hyper elliptic curve (HEC) fulfils all the gaps of existing system [2].

Keywords:- Signcryption, hyper elliptic curve, random number, sensor-based, forward secrecy.

1 Introduction

CIA(confidentiality, integrity, authentication) are the main security requirements in any environment. Signcryption scheme is mainly used to provide security and efficiency in terms of computational cost. Till today, many signcryption schemes based on Elliptic curve cryptography(ECC), RSA, El-Gamal had been proposed but unfortunately these schemes are not suitable in today's advanced wireless and mobile networks due to lack of required security features. Similarly, signature-then-encryption is not well suited for resource(power) constrained environment. Hence a single logic which combines both signature and encryption is needed.

In recent years, the proposed scheme based on the technique of signcryptionin HEC has attracted many researchers because of its short key size, lower computational cost, lower communication cost when compared to other cryptosystems. Moreover, the proposed scheme provides higher security primitives such as public verifiability and forward secrecy which have its major contribution in the field of E-Commerce, M-Commerce and Banking sectors.

Signcryption scheme uses random number in its algorithm. Random numbers that satisfy Randomness and Independence are required and difficult to generate. Random Number Generators use seed value to generate long stream of random numbers.

The random number generators can be classified as True Random Number Generator (TRNG) and Pseudo Random Number Generator (PRNG). TRNG processes physical phenomena measured that are truly random in nature. PRNG that implement deterministic algorithm with some seed value produces random numbers that are not truly random as they can be determined in some way.

In PRNG seed values may be from any deterministic source like clock pulse, user activity, interrupts, time, etc. One possible solution is to use sensor data as seeds for PRNG. Sensors are devices that measure any physical parameters like pressure, temperature, motion, etc. Since these physical parameters exhibit randomness the values recorded by these sensors provide a great source of random seeds for PRNG.

ISBN: 978-1-61804-336-8

2 Hyperelliptic curve overview

Koblitz in 1989 first coined the use of Hyper elliptic curve cryptography(HECC) in public key cryptosystem. HEC is defined as a curve C of finite field F over the genus represented as g which is greater than one(g=1 means elliptic curve). It is denoted by the equation,

$$C: y^2 + h1(x) = f1(x) \quad (1)$$

where h1(x) is a polynomial of degree atmost g and f1(x) is a polynomial of degree 2g+1.

There are q number of points in the HECC defined over Fq. It is given by using Hasse-Weil representation as $(q+1-2g) \leq C \leq (q+1+2g)$.

3 Problem with sensor data

At first it may seem like more data from more sensors could produce more random number, however installing multiple sensors on same device introduces the problem of correlation that makes it not possible to use for RNG. Sensors are vulnerable to attacker who controls the sensor in generating vulnerable seeds. When these vulnerable seeds are fed to RNG the attacker produces the expected random number sequence that breaks the security. In this approach the sensor raw data are leveraged with 3 step approach of processing those sensor data before feeding them into RNG.

Using sensor data impose two problems 1.Adversarial control where attacker knows the predictable pattern there by learning the random sequence and 2.Collinearity among multiple sensor data. Also the sensor data are accessible to the operating systems which an attacker can gain by using specialized software. Thus the sensor data must be made secured from attackers.

4 Signcryption scheme

4.1Key generation

Inorder to establish more secure communication, sender and receiver will generate their own private and public keys.

1. Private key of sender d_a
2. Public key of sender $P_a = d_a D$
3. Private key of receiverd_b
4. Public key of receiver $P_b = d_b D$

4.2 Signcryption

The eight steps of signcryption with the parameters (k, m, d_a, P_a, P_b) are given as follows

Step 1: Choose an integer k randomly specified in the range {1,2,…,n-1}

Step 2: Compute the secret key $k1= H(\varphi(kD))$

Step 3: Compute the secret key $k2= H(\varphi(kP_b))$

Step 4: Calculate the cipher text $c = E_{k2}(m)$

Step 5: Calculate the value $r = h_{k1}(m||$ bind-msg)

Step 6: Compute the value $s = (k/(r+ d_a)) \bmod n$

Step 7: calculate the value R = rD

Step 8: The signcrypted message to be transmitted to receiver is (c,R,s)

4.3 Un Signcryption

The receiver on receiving the signcrypted message(c,R,s), performs un signcryption. The five steps for getting the plain text are as follows:

Step 1: Compute the secret key $k1 = H(\varphi(s(P_a+R)))$

Step 2: Compute the secret key $k2 = H(\varphi(s(d_b (P_a+R))))$

Step 3: Compute the value $m = D_{k2}(c)$

Step 4: Calculate the value $r = h_{k1}(m||$ bind-msg)

Step 5: Check rD = R? if condition is satisfied accept the message else reject

5 Processing the sensor data

To make the sensor data usable for RNG a three step approach has been proposed. 1. WASH- Eliminating the true data 2.RINSE- Transforming to continuous random sequence 3.SPIN-Feeding the processed data to a RNG to generate the random sequence. To test this approach accelerometer data are collected that is embedded in smartphone. The accelerometer measures and records the daily usage with respect to movement, tapping of screen, orientation of screen. The acceleration magnitude in X-axis is considered.

5.1 Wash

Sensors produce data that are mostly predictable with little randomness, our first step is to remove these major predictable to completely eliminate the user behavior from sensor data. Since the data keep on drifting and moving within range the mean and variance sequence keep on changing over time. This is called nonstationarity. Wash step involves removing those predictable patterns by contaminating them. True data are removed by

ISBN: 978-1-61804-336-8

differentiating the raw data until the nonstationarity is removed.

First order derivative of sensor data will be sufficient to remove nonstationarity. If still nonstationarity remains differentiation can be repeated until threshold stationarity is obtained. Stationarity represents the data are not drifting and moving anymore.

5.2 Rinse

The washed data still contains little predictable data and bare spots that need to be rinsed off to improve unpredictability. This step does not directly use the washed data instead they are transformed to complex numbers and then rinsed.

The data sequences are converted to complex form by Fast Fourier Transform. The complex exponentials are given by

$$X_k=\sum_{n=0}^{N-1} x_n e^{-i2\pi k\frac{n}{N}}$$

The real number represents the magnitude of sine wave. To provide randomness to data, the imaginary number is replaced by random number. Then the complex numbers are changed from frequency domain to time domain by Inverse Fast Fourier Transform. The rinsed data look similar to the data sequence generated by Mersenne twister random number generator. This shows that rinse step has improved the randomness of data sequence. At the end of rinse 8-bit integer is now available that is fed as seed to a RNG.

5.3 Spin

The data sequence that is made random in the rinse step is now used as seed for a PRNG that is used for password creation, key exchange, encryption, connection establishment, etc. The data sequence is broken down into sizes depending on the PRNG used. For our testing Blum-Blum-Shub generator was used.

The Blum-Blum-Shub generator uses the equation of form

$$x_{n+1} = {x_n}^2 mod\ M$$

Two prime numbers p and q are chosen and M=pq. The seed value x_0 is any integer that is co-prime with M. The random bits are derived from the output's x_{n+1} parity bit or few least significant bits. As a result our processed seeds are spun into long stable sequence of random numbers.

6 NIST test suite results

Generator	Without sensor data (% of randomness)	Raw sensor data (% of randomness)	Processed data (% of randomness)
Blum-Blum-Shub	94.2	94.7	98.3
Linear congruential generator	90.5	92.4	97.6
Mersenne Twister	93.8	95.3	98.7

From the above table it has been identified that the proposed scheme produces better random bits when compared to existing schemes.

7 Security analysis of signcryption in HECC

The proposed scheme satisfies all the security strategies such as 1.Confidentiality 2.Authentication 3.Integrity 4.Un forgeability5.Forward secrecy 6.Public verifiability. Table 2 shows the security analysis of proposed scheme based on HECC with the other existing systems.

7.1 Hyperelliptic Curve Discrete Logarithmic Problem (HECDLP)

HECDLP is defined over a curve C of finite field F such that find two divisors D1 of known order n and D2 contained within D1. To find an integer z such that D1 = zD2 is hard.

7.2 Confidentiality

This efficient signcryption scheme based on HECC can withstand breach of confidentiality even though the attacker can find some public values of P_a and P_b since because finding private key from public key is infeasible.

7.3 Authentication

Proposed scheme ensures user authentication as well as message authentication. The signed message contains the cipher text, R,s. The value of r is calculated by applying hash algorithm SHA-1 on the message with secret key k1. But by the definition of HECDLP solving k1 is infeasible and also SHA-1 is more secure.

7.4 Integrity

The proposed scheme ensures the originality of sender as well as message. Sender calculates the cipher text c and send it to the receiver. If an attacker gets c, he modifies it to c' so the values of r and s change to r' and s'. In order to prove the legitimacy, attacker needs to calculate s and r. Obtaining the value of s needs the value d_a , it has been proved that solving d_a is infeasible.

7.5 Un forgeability

In the proposed scheme, an attacker can never forge the signcrypted message (c,R,s) because for forging he needs to calculate the sender's private key d_a but it is infeasible by the definition of HECDLP.

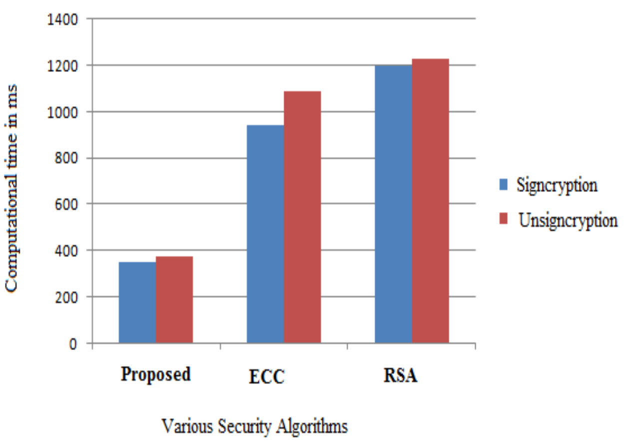
7.6 Forward secrecy

Forward secrecy is defined as protecting the session key even if the sender's private key is compromised by an attacker. Let us consider, an attacker to gain access to sender private key (d_a) he has to find the session key k2 and the value of r and s which have already been proved by the definition of HECDLP that they are infeasible.

7.7 Resist password guessing attack

As the proposed scheme uses signcryption with strong AES and SHA-1 it is impossible for an attacker to calculate the values of R and s even though he got public key values. As this scheme provides forward secrecy guessing password for user identity is infeasible.

8 Result Analysis



9 Conclusion

The proposed scheme based on Hyperelliptic curve provides all the necessary security features- confidentiality, authentication, forward secrecy since HECDLP is hard to solve. This scheme resists the smart card attack, offline password guessing attack, etc. which are having their major applications in today's world. The three step processed sensor data are more secure and strong in random number generation. Although second step is invertible, the first step of eliminating the true data is not invertible thus making the entire process not invertible. There is scope in future to implement this signcryption scheme for remote authentication, seed for random number can be improved by deploying other methods that can be replace FFT, complex numbers. Also it considers only raw data from single sensor where multiple sensor data can be considered.

References:

[1] N. Koblitz, Algebraic Aspects of Cryptography, *Springer series of Algorithms and Computation in Mathematics*, Vol. 3, 1998.

[2] D.He and D. Wang, Robust biometrics-based authentication scheme for multi-server environment, *IEEE Systems Journal*,Vol. 3, 2014, pp.816-823.

[3] J. Qu, X. Tan, Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem, *Journal of Electrical and Computer Engineering*, Vol.2014, 2014, pp.1–6.

[4] S. Ashraf Ch, M. Sher, A. Ghani, H. Naqvi and A. Irshad, An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography, *Springer journal on Multimed Tools Appl*, Vol. 74, 2015, pp.1711–1723.

[5] S.K. Park and K.W. Miller, Random number generators: Good ones are hard to find, *Communications of the ACM*, Vol. 31, 1988, pp. 1192-1201.

[6] L. Blum, M. Blum, and M. Shub, A simple unpredictable pseudo random number generator, *SIAM Journal on Computing*, Vol. 15, 1986, pp. 364-383.

[7] S.L. Hong and C. Liu, Sensor-Based Random Number Generator Seeding, *IEEE Access*, Vol.3, 2015, pp. 562-568.

[8] M. Babaei and M. Farhadi, Introduction to secure PRNGs, *IET journal on Communications Network and System Sciences*, Vol. 4, 2011, pp.616-621.