ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

An Efficient Approach to Enhance Security for Android Applications

Dr.A.Gnanabaskaran[1], M. Dinesh Kumar[2], M.Dinesh[3], K.Latha[4], S.Nithya[5], C.Prakash[6]

K.S.Rangasamy College of Technology, Tamilnadu, India[1, 2, 3, 4, 5, 6]

ABSTRACT— Android gives a world-class platform for creating apps and games, as well as open marketplaces for distributing them instantly. The hackers can easily hack such open source application codes, edit their signatures, publish them over the internet and take control of devices which install those applications, to gain access to user's personal information. To prevent this, an efficient digital signature scheme is needed. Thus a new system in which the mechanism of digital signature using elliptic curve cryptography has been proposed that gives the combined efficiency of Two Key Signature Scheme and Multi-signature Scheme. Multi-signature scheme is used to provide certifiable digital signatures that are signed many times rather than signing once. Each signature is signed using two key signature scheme of elliptic curve cryptography to increase the complexity of brute force attacks. Thus, it would be efficient digital signature approach that suits current mobile platforms such as Android.

KEYWORDS—Android Applications, Digital Signatures, Elliptic Curve Cryptography, Multi-signature Scheme, Two Key Signature Scheme

I. INTRODUCTION

Digital signatures play a vital role in the security of information and communication networks by providing message integrity, authentication and non-repudiation during transmission over any insecure or hostile network based on the public key cryptography. A digital signature or digital signature scheme is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. The property of message integrity guarantees that the receiver detects any alteration of the message during transmission, and the authentication property ensures the message generation by an expected sender. Compared with these two properties, the non-repudiation property is equally important, which assures that after creating a signature, the signer cannot deny the signature generation at a later time. However, this scheme becomes less secure in cases where only one person is involved in signing an application. An approach that allows more than one person to sign the application is needed in the digital signature certification environment.Also current technological trends have improved much towards the attacking the older signature schemes. Thus improved signature schemes that make an attacker difficult to hack the signature are needed. Here the combined features of Certificate less Multisignature scheme (SK Hafizul Islam *, G.P. Biswas, 2013) and Two key Signature Scheme (N. Anil Kumar ,ChakravarthyBhagvati, 2012) are used to solve above issues

II. PRELIMINARIES

Simplified Weierstrass equation for elliptic curve is $y^2 = x^3 + ax + b$
where a, b, x and y belongs to some field with $4a^2 + 27b^2$ =0. Let P and Q be two points on the elliptic curve. The line joining the two points will intersect the curve at the third point say R. The addition of points is defined as$P + Q + R = identity$

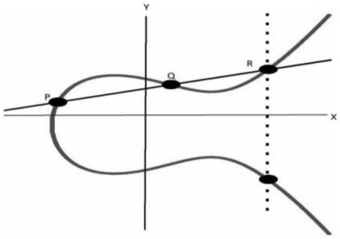

Fig. 1 Point addition in Elliptic curve cryptography

To make elliptic curve addition a group, identity is defined to be a point at infinity denoted as *O* or ∞. A line is said to pass through point at infinity when it is exactly vertical.

Let *P*, *Q* and *R* be the points on the elliptic curve then the following holds.

- *P*+*Q* will be point on the curve (Closure property)
- *P*+*Q*=*Q*+*P* (Commutative property)
- (*P*+*Q*) +*R*=*P*+(*Q*+*R*) (Associativity property)
- *P*+O=O+*P*=*P* (Existence of an identity element)
- There exist *(-P)* such that (−*P*) + *P* = *P* + (−*P*)= (*O*) (Existence of inverse)

Scalar multiplication is defined a repeated addition.

Let *n* be integer,*nP*= *P* + *P* +L*n*……… times

The points on the elliptic curve form an abelian group.

Let P(x1,y1) Q(x2,y2)=R (x3,y3)then
X3=m2-x1-x2 and y2=m(x1-x3)-y1
Where M= (y2-y1) / (x2-x1) for P≠Q
=($3x1^2$+a) / 2y1 for P=Q

The elliptic curve arithmetic which has been defined over real numbers can also be defined over finite fields. Most of the definitions over real numbers can be carried over to finite fields.

Let q = $p^k$where p is prime. q F is the G also is field of order q. Let E be the elliptic curve. The set of points on the elliptic curve E (Fq)is

E(Fq)={0} U {(x,y)□Fq x Fq | y^2= x^3 + ax+ b

III. ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the Digital Signature Algorithm (DSA). We refer to this as one key ECDSA. It is the most widely standardized elliptic curve-based signature scheme, appearing in the ANSI X9.62, FIPS 186-2, IEEE 1363-2000 and ISO/IEC 15946-2 standards as well as several draft standards [4, 5].

The Domain parameters D = (q, FR, S, a, b, P, n, h) are comprised of

1. The field order q.
2. An indication FR (Field Representation) of the representation used for the elements of*Fq*.
3. A seed S if the elliptic curve was randomly generated.
4. Two coefficients *a*, *b* □*Fq*that define the equation of the elliptic curve E over q F (i.e. y^2=x^3 + ax+ b in the case of a prime field and y^2+ xy= x^3+ ax^2+ b in case of binary field).
5. Two field elements p x and p y in q F that define a finite point P=(xp,yp) □E(Fq)in affine coordinates. P is called the base point.
6. The order of P is n.
7. The cofactor *h* = #E (Fq) /n .

In the following, H denotes a cryptographic hash function whose outputs have bit length no more than that of *n* (if this condition is not satisfied, then the outputs of H can be truncated).

A. ECDSA Signature Generation
Input: Domain parameters D = (q, FR, S, a, b, P, n, h), private key d, message m.
Output: Signature (r, s).

B. ECDSA Signature Verification
Input: Domain parameters D = (q, FR, S, a, b, P, n, h), public key Q, message m, signature (r, s).
Output: Acceptance or rejection of the signature.
1. Verify that r and s are integers in the interval [1,n-1]. If any verification fails then return ("Reject the signature").
2. Compute e = H(m).
3. Compute $w = S^{-1}$mod n.
4. Compute u1=ewmod n and u2= rwmod n.
5. Compute $X = U1\ P+ U2\ Q$.
6. If $X = \infty$ then return ("Reject the signature").
7. Convert the x-coordinate x1 of X to an integer x1'. Compute v = x1'mod n.
8. If v = r then return ("Accept the signature"); Else return ("Reject the signature").

C. Proof That Signature Verification Works
If a signature (r, s) on a message m was indeed generated by the legitimate signer, then S ≡ k^{-1}(e + dr) mod n Rearranging gives
K ≡ S^{-1}(e + dr)
≡ S^{-1}e + S^{-1}rd (mod n)
≡ we + wrd
≡ (u1 + u2) d (mod n)
Thus X = u1P + u2Q = (u1 + u2)d =kp.
Sov = r as required.

IV. TWO KEY ECDSA

The domain parameters D = (q, FR, S, a, b, P, P1, R n, h) are same as the original ECDSA except P1 and R. P1 is another base point.
Let d and $d1$ be the private keys, d1P + d P1 = R1 anddP= Qwhere$P$, $P1$ and R are the public parameters.
H defines the hash function as in ECDSA.

A. Two Key ECDSA Signature Generation
Input: Domain parameters
$D = (q, FR, S, a, b, P, P1, R, n, h)$ private key d and $d1$, message m.
Output: Signature (x1,s1,s2).
1. Select k1 and k2□R [1, n-1]
2. Compute $(kP+ kP) = Sxy$If $S = \infty$ goto step 1.
3. Compute e = H(m). Convert the field element 1 x to an integer x (usually the first coordinate in the vector representation)
4. Let $s = ek+ x$ and $s = ek+ xd$.
5. Return (x1,s1,s2).

B. Two key ECDSA Signature Verification
Input: Domain parameters

$D = (q, FR, S, a, b, P, P1, R, n, h)$
public key P,Q, $P1$, R message m, signature (x1,s1,s2)
Output: Acceptance or rejection of the signature.
1. Compute e = H(m).
2. Compute $u1 = s1P + s2P1$.
3. Compute$y1$ and $y1'$from$x1$ using curve equation.
LetT=(x1, $y1$) and T^1= (x1,$y1'$)
4. Compute $u2 =eT +x1R$.
5. If$u1 = u2$ accept the signature, else
If u2= eT + x1R accept the signature.
6. Else reject the signature.

C. Proof That the Signature Verification Works

If a signature (x1,s1,s2) on a message m was indeed generated by the legitimate signer, then

$$s1P + s2P1= (ek1 + x1d1)P1 + (ek2 + x1d1)P$$
$$= e(dP+ d1P1) + x1(k1P + k_2P2)$$
$$=eT+ x1\ R.$$

V. MULTI SIGNATURE APPROACH

Multi-signature approach is the process of signing an application or document by more than one signer. Here the signature by each signer is represented by □□and it is generated by the two key ECDSA method proposed above. This makes the signature more self-secured than other ordinary signature methods.

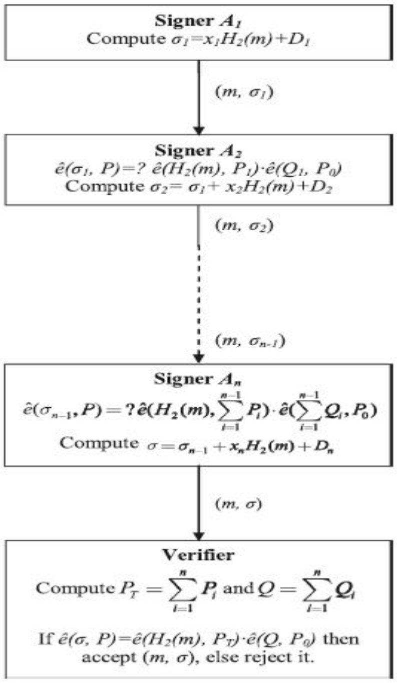
Signer A_1
Compute $\sigma_1 = x_1H_2(m)+D_1$
(m, σ_1)
Signer A_2
$\hat{e}(\sigma_1, P) =? \hat{e}(H_2(m), P_1)\cdot\hat{e}(Q_1, P_0)$
Compute $\sigma_2 = \sigma_1 + x_2H_2(m)+D_2$
(m, σ_2)
(m, σ_{n-1})
Signer A_n
$\hat{e}(\sigma_{n-1}, P) = ?\hat{e}(H_2(m), \sum_{i=1}^{n-1} P_i)\cdot\hat{e}(\sum_{i=1}^{n-1} Q_i, P_0)$
Compute $\sigma = \sigma_{n-1} + x_nH_2(m) + D_n$
(m, σ)
Verifier
Compute $P_T = \sum_{i=1}^{n} P_i$ and $Q = \sum_{i=1}^{n} Q_i$
If $\hat{e}(\sigma, P)=\hat{e}(H_2(m), P_T)\cdot\hat{e}(Q, P_0)$ then accept (m, σ), else reject it.

Fig. 2 Multisignature method of A1,A2,..An signers

A. Signing Algorithm

In order to generate a sequential short multisignature for a given message m□□{0, 1}*, each signer *Ai(1≤i≤n)*performs the following operations:

Step 1: The signer A1

(a) Computes □□□=x1H2(*m*) + D1.

(b)Sends the message-signature pair(*m*,□□)to the next signer*A2.*

Step 2: The signerA2

(a) Verifies (m,□□□) by determining whether the equation ê(□□,P)=ê(H2(m),P1)ê(Q1,P0) holds.

(b) If it holds, *A2* computes □□ = □□+ *x2*H2(*m*) + *D2* i.e.□□□=*x1*H2(m) + *x2*H2(m) + D1 +D2 and then sends (m,□□□) to the signer A3. Similarly, the signer *A3* signs and sends to *A4* and so on up to *An-2* to *An-1*. All sequentially compute their signatures and complete the multisignature process.

Step n: The last signer An

(a) Verifies (m,□□□*n-1*) received from A_{n-1} by determiningwhether the equationê(□*n*□□,P) – ê$(H_2(\mathrm{m}), \sum_{i-1}^{n} P_i)$ ê $\sum_{i-1}^{n} Q_i, P_o)$ holds.

(b) If it holds,A_n Computes□□*n*□□□□*n*□□□□*xn*+$x_n H_2(\mathrm{m})$+ D_ni.e.,□□*n*−$\sum_{i-1}^{n}[x_i\, H_2(\mathrm{m}) + D_i]$ – □ (say) and then sends the final signature (m,□) to the verifier forverification.

C.Verification Algorithm

In order to verify (m, r), the following steps are to be executed by the verifier:

(a) Compute $P_r - \sum_{i-1}^{n} P_i$ and Q$-\sum_{i-1}^{n} Q_i$.

(b) Verify whether the equation ê(□,P) – ê(H_2(m),P_r)$\hat{e}$(Q,P_o) holds . If so, the verifier accepts (m,□); otherwise the verifier rejects it.

D.Correctness of theMulti-signature Scheme

The received message-signature pair (m, r) is accepted bythe verifier since the following holds:

ê(□, P) = ê($\sum_{i=1}^{n}$ □□*I, P)*

$$= \hat{\mathrm{e}}(\sum_{i=1}^{n}(x_i H_2(\mathrm{m}) + D_i), P)$$

$= \hat{\mathrm{e}}(\sum_{i=1}^{n}(x_i H_2(\mathrm{m}), p) \cdot \hat{e}(\sum_{i=1}^{n} D_i\, , P)$[due to bilinearity]

$= \hat{\mathrm{e}}(H_2(\mathrm{m}), \sum_{i=1}^{n} x_i p) \cdot \hat{e}(\sum_{i=1}^{n} sQ_i\, , P)[D_i$=s$Q_i]$

$= \hat{\mathrm{e}}(H_2(\mathrm{m}), \sum_{i=1}^{n} P_i) \cdot \hat{e}(\sum_{i=1}^{n} Q_i\, , sP)$ [P_i=x_iP]

$= \hat{\mathrm{e}}(H_2(\mathrm{m}), P_r) \cdot$ ê(Q ,P_O)[$P_r = \sum_{i=1}^{n} P_i$, Q= $\sum_{i=1}^{n} Q_i$,$P_o = sP$]

VI. PROPOSED APPROACH

To make the digital signature seem too hard to break and to simultaneously maintain the simplicity of such algorithm combined efficiency of above two methods has been proposed. This includes signing a single application by more than one authority. Each signer should use two key ECDSA methodology of digital signature to sign the application. This method thus removes complexities or issues arose when signed by only one signer.

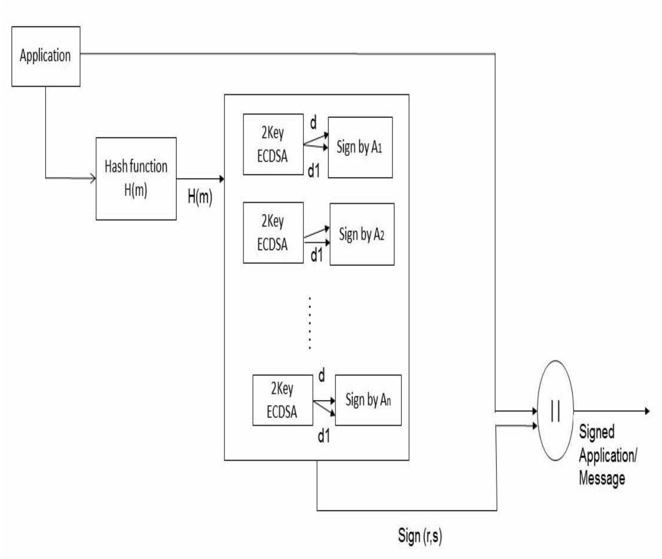

Fig. 3 Proposed method of Multi-signature combined with Two Key ECDSA

VII. COMPARISON OF ECDSA, TWO KEY ECDSA AND COMBINED APPROACH

The brute force method to attack the ECDSA is to find d (private key) from public key Q and domain parameter P which satisfy the relationship Q=d P. The order of P is n so to find Q we have to check n possibilities. So the time complexity is O(n). The brute force method to attack the Two key ECDSA is to find d and d1 (private keys) from P1, P and S which satisfy the relationship dP + d1*P1 = S. The order of P is n1 and order of P1 is n2. To find S we have to check n1×n2 possibilities. So the time complexity is O(n1 × n2).

The brute force attack for combined approach of multi-signature and two key ECDSA involves the finding of d & d1 private keys of Signer An, then same for An-1upto An and thus for each pair of keys to be found, the complexity is O(n1 xn2) thus accounting a total complexity of O(n x n1 x n2) where n is the number of signers. Thus usage of multiple signatures makes it practically impossible to crack the code in finite period of time within which the application is valid. Also, the signer has the option to choose between multiple signature and single signature.

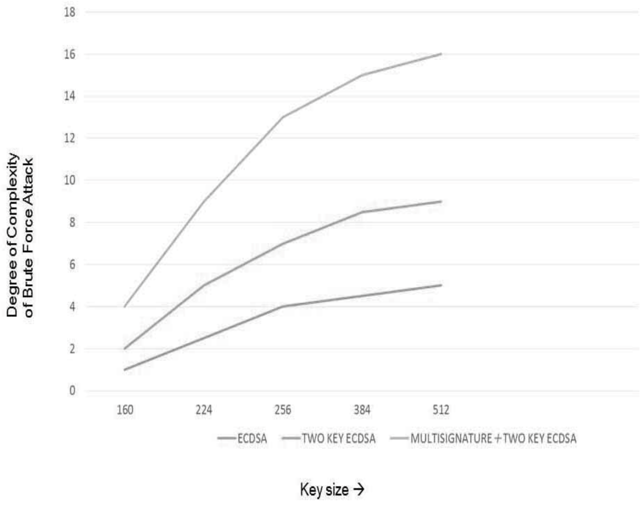

Fig. 4Comparison between standard ECDSA, two key ECDSA and proposed method i.e., combined approach of multi-signature and two key ECDSA

VIII. CONCLUSION

The method proposed above is thus an efficient method that can be used for signing any application which is more vulnerable. Thus it can be a best suit for android as well as other applications. It not only makes an application harder to crack but also brings out a means for signing an application by more than one authority still leaving the signing mechanism less complex than the other ones.The method can be applied to any type of application signing processes.

REFERENCES

[1] N. Anil Kumar, ChakravarthyBhagvati "Two Key Signature Scheme with Application to Digital Certificates" in 1st Int'l Conf. on Recent Advances in Information Technology | RAIT-2012.

[2] Islam, S.H., Biswas, G.P., "Certificateless short sequential and broadcast multisignature schemes using elliptic curve bilinear pairings" in Journal of King Saud University – Computer and Information Sciences (2013).

[3] Chu, H., Zhao, Y., 2008. Two Efficient Digital MultisignatureSchemes. In: Proceedings of the International Symposium onComputational Intelligence and Design (ISCISD'08), pp. 258–261.

[4] V. Miller, "Use of elliptic curves in cryptography," Lecture notesin computer sciences; 218 on Advances in cryptology—CRYPTO 85, pp. 417–426, 1986.

[5]Chen, T.-S., Huang, K.-H., Chung, Y.-F., 2004. Digital multisignaturescheme based on the elliptic curve cryptosystem. J.Comput. Sci. Technol. 19 (4), 570–573.

[6] N.Anil Kumar, R.Tandon and ChakravarthyBhagvati, "Modifiedelliptic curve digital signature algorithm," In Proceedings of theInternational
Conference on Mathematics and Computer Science2009, Vol: 1, pp: 304-306, 2009.