5th International Conference on Recent Developments in Science, Humanities & Management

The International Centre Goa, Panjim, Goa (India)

Conference World

18th August 2019 www.conferenceworld.in ISBN: 978-81-941721-3-0

Comparative Analysis of ECDSA variant

Hiral Nagda[1], Kajal Mehta[2], Prasad Awate[3],

Rashmi Malvankar[4], Dhanashree Toradmalle[5]

[1,2,3,4,5] *Department of IT, Shah And Anchor Kutchhi Engineering College, India*

Abstract—*ECDSA is the principal effective calculation that in view of ECC. It was acknowledged by ANSI, IEEE, NIST and ISO as the standard signature computation. In this paper, ECDSA was examined and a comparative analysis of its variants is presented. The improved calculations lessen the computational expense while keeping a similar security as unique ECDSA. They are reasonable for the clients who have constrained register limit in various cases. The exhibition information of every calculation was exhibited too.*

Keywords—Cryptography, Digital Signature,ECC, ECDLP,ECDSA.

1. INTRODUCTION

While composing a cheque, the creator of the cheque signs on a same cheque to give its credibility to beneficiary. Yet, that is not the situation in computerized signature (digital signature), the sender needs to give separate archive as signature (digital signature) alongside the message to give its realness to the recipient.

For each message to be sent a new digital signature is created and is sent alongside the message. That implies for each bit of the message to be sent, a digital signature is generated and sent along side with the message.

Digital Signature[1] is an electronic signature that proves the authenticity of the sender.The signer uses a private key in the signing algorithm to sign the document and verifier uses the public key of the signer to verify the document. Digital Signature is a public key cryptosystem and they provide the following

1. Message Authentication
2. Message Integrity
3. Non-repudiation
4. Confidentiality

Digital signature consists of three phases

1. Key-pair generation
2. Signature generation(signing phase)
3. Signature verification(verifying phase)

Digital signatures are broadly of two types RSA based and ECC (Elliptic Curve Cryptography) based digital signatures. ECDSA is an ECC based DS.

2. ECC

2.1 Introduction of ECC

ECC (elliptic curve cryptography) is another public key cryptosystem. It is a way to deal with open key cryptography in view of the logarithmic structure of elliptic bends over limited fields. The utilization of elliptic bends in cryptography was proposed freely by Neal Koblitz and Victor S. Mill operator in 1985. Contrasted and other cryptosystem, ECC has the most astounding security capacity per bit.

2.2 Use of ECC in real world

These days, numerous openly perceptible Internet conventions contain elliptic bend based cryptographic calculations. Today, ECC is progressively used to actualize open key cryptography conventions, for example, advanced marks and key understanding conventions. Bitcoin , Secure SHell (SSH) and Transport Layer Security (TLS) are a portion of the application conventions which utilize ECC in reality. Numerous E-business applications[8] additionally utilize elliptic bend cryptography, because of its security highlights.

2.3 Why ECC based digital signatures are better?

With smaller size of key size ECC based digital signatures[8] give same or preferred dimension of security over RSA based digital signatures ECC based digital signatures are progressively secure in light of its modular

math behind it, as it's perplexing and hard to comprehend and break.

ECC is a method of public-key encryption based on the algebraic function and structure of curve over a finite graph. One myth about ECC curve is that it might be elliptical in shape but the truth is ECC has its name because of the Weierstrass equation:

$$y^2 = ax^3 + x + b \quad (1)$$

where a and b are constants and necessary condition :

$$4a^3 + 27b^2 \neq 0$$

It uses trapdoor function predicted on infeasibility of determining the discrete logarithm of a random elliptic curve element that has a publicly known as base point.

TRAP-DOOR FUNCTION[1]

$$X = f(Y) \rightarrow \text{EASY}$$

$$f^{-1}(X) = Y \rightarrow \text{DIFFICULT}$$

2.4 ECC ARITHMETIC[1]

Point Addition:

Let P=(x1, y1) and Q=(x2,y2) be two distinct points on an elliptic curve E. The sum R, of P and Q, is defined as follows: Draw a line connecting P and Q extend it to intersect the elliptic curve at a third point. Then the sum R, is the negative of the third point ,reflection of third point about X – axis.

Point Doubling:

Draw a tangent from a point on elliptic curve and will intersect at a point. It computes Pk where k is an integer and P is a point on elliptic curve. It is done by repeated addition (P + P + P + P + P +... for k times). This operation dominates the execution time of elliptic curve cryptographic schemes.

3 LITERATURE SURVEY

In original ECDSA[1], if Eve intends to crack the elliptic curve he needs to know d, which is difficult task if appropriate, secure pseudorandom generator is used. For eg: a key of size 192 bits,if Eve uses a supercomputer with a speed of 1 billion operations per second ,then also Eve will take approximately 2 ½ trillion years to crack the secret d.ECDSA is able to provide this level of security because of it's math that implies ECDLP problem. Secret key k used for signing two separate messages should be different otherwise it is possible to recover it. Calculating inverse function in signing phase is expensive also curve parameters to be chosen wisely otherwise prone to well-known Pollard's rho algorithm. Variant 1 of ECDSA[6] is compatible for a signer with bounded/confined computing resources, therefore signing phase is less complex. Another positive is, no need to calculate inverse of d for every single message, as it remains stable for a period of time and can be computed and stored during key-generation phase. Variant 2 of ECDSA[4] is compatible for a verifier with bounded/confined computing resources, therefore in verifying phase no need to compute inverse function. One negative is that common between both Variant 1 and 2 is that with same k an attack can be successfully imposed. This is negative tried to overcome in Variant 3 of ECDSA[3,4] as it uses two secret keys k1, k2 but a big positive is d cannot be determined even if same secret key is used. Definitely this variant is more secure than the original ECDSA and expensive in terms of time and cost. Variant 4 of ECDSA[7] also called as Elliptic Curve German Digital Signature Algorithm (ECGDSA) here the inverse function computation in key-generation phase and key will remain constant for a specific period of time (but still inverse function needs to be calculated). This will not only save cost but also time taken by each operation performed. A big plus in this variant is, k cannot be determined even though the same secret key is used to sign 2 separate messages that implies not vulnerable to attack on the same key. Variant 5 of ECDSA[7], there is no need to compute inverse function neither in signing nor in verifying phase. The entire information regarding the signature is embodied into a point on elliptic curve. But a big negative is that the signature can be forged.

4 Comparative analysis of ECDSA Variants

Algorithm	*No. of secret keys used*	*Attack when same secret key used*	*Use of inverse in key generation phase*	*Use of inverse in signing phase*	*Use of inverse in verifying phase*
ECDSA	1	Vulnerable	✓	✓	✗
Variant 1	1	Vulnerable	✓	✗	✓
Variant 2	1	Vulnerable	✗	✓	✗
Variant 3	2	Not Vulnerable	✓	✓	✗
Variant 4	1	Not Vulnerable	✓	✗	✓
Variant 5	1	Not Vulnerable	✓	✗	✗
ECKCDSA	1	Not Vulnerable	✗	✗	✓

Graphs of time required in ms by each phase:

The graphical representations of computation time for each variant to generate the keys, signature and to verify that signature when performing ECDSA are shown in Fig. 1, Fig. 2 and Fig. 3 respectively.

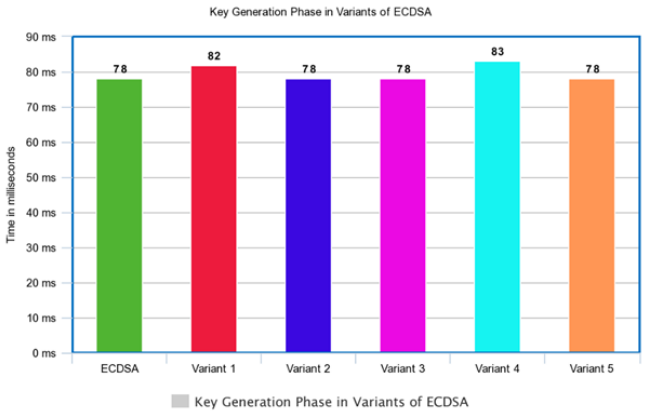

Fig 1: Key Generation Phase

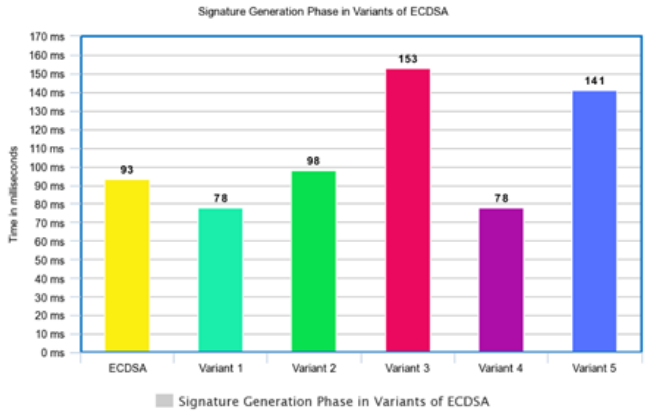

Fig 2: Signature Generation Phase

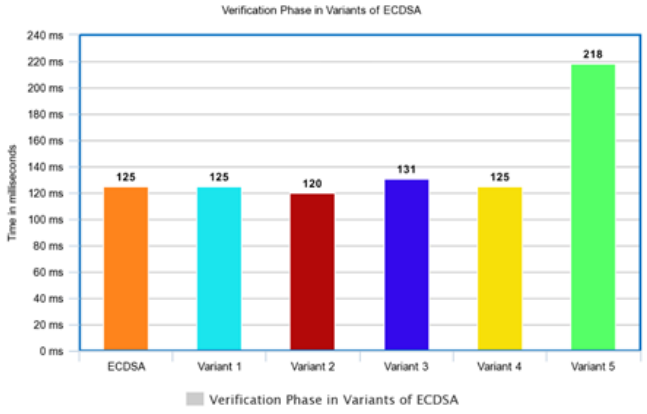

Fig 3: Verification Phase

5 CONCLUSION

The security of ECDSA relies on the complexity of the ECDLP of the system. If large number of keys are chosen in ECC, then a brute-force attacker cannot easily break the cryptographic algorithm. For ECDLP security, the four parameters need to be taken into count were first, the safety of the system against Pollard's rho algorithm; second, its security against multiplicative transfer; third, the rigidity of system; and last, the complex multiplication field discriminator. If the system fails to resist any one of the parameters listed above, the authors[8] considered that particular system to be unsafe for use in the ECC application. Again the choice of variant to be implemented in the system for security measures it relies on the preferred dimension of security to be implemented in the system. From a security point of view Variant 3 is most secure than its peers but time required to generate a signature is maximum than its peers as seen in fig 2.

6 ACKNOWLEDGMENT

The authors 1,2 and 3 would like to thank Mrs Dhanashree Toradmalle, Asst Professor and Mrs Rashmi Malvankar, Associate professors at Shah and Anchor Engg College, Mumbai for guiding them in surveying and analysing the topic of the paper.

REFERENCES

[1] William Stallings "Cryptography and Network Security", fourth edition

[2] John Malone-Lee , Nigel P. Smart, "Modifications of ECDSA", Springer- Verlag Berlin Heidelberg 2003.

[3] Hung-Zih Liao, Yuan-Yuan Shen, "On the Elliptic Curve Digital Signature Algorithm"Tunghai

Science Vol. 8: 109126 July, 2006

[4] M.Prabu,R.Shanmugalakshmi, " schemes in a new approach to variant on ECDSA A comparative Analysis of signature", 2009 IEEE.

[5] Don B. Johnson,Alfred J. Menezes,Elliptic Curve DSA (ECDSA): An Enhanced DSA

[6] Hu Junru , "The Improved Elliptic Curve Digital Signature Algorithm", 2011 IEEE.

[7] Qiuxia Zhang , Zhan Li , Chao Song , "The Improvement of digital signature algorithm Based on elliptic curve cryptography", 2011 IEEE.

[8] D. J. Bernstein and T. Lange, "Safe Curves: choosing safe curves for elliptic-curve cryptography", accessed 14 May 2017,safecurves.cr.yp.to/.