

Cryptography Survey of DSS and DSA Algorithms

Mohammed Abdulhakim Al-Absi*, Azamjon Abdullaev*, Ahmed Abdulhakim Al-Absi**, Mangal Sain***, Hoon Jae Lee***

*Division of Information and Communication Engineering, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 47011, Republic of Korea

**Department of Smart Computing, Kyungdong University 46 4-gil, Bongpo, Gosung, Gangwon-do, 24764, Republic of Korea

***Division of Information and Communication Engineering, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 47011, Republic of Korea

Mohammed.a.absi@gmail.com, Azamjon.a.sobirovich@gmail.com, absiahmed@kduniv.ac.kr, mangalsain1@gmail.com, hjee@dongseo.ac.kr

Abstract—Digital signatures are fundamental in today's modern world to verify the sender of a document's identity. Digital signature is a method to bring integrity and authenticity to the digital data signature. Signature is computer utilizes a set of principles and parameters such that identification and authenticity of the data can be confirm. There are three algorithms that are suitable for digital signature generation under the DSS standard. First one is RSA algorithm second is the ElGamal algorithm, and the third is Elliptic Curve Digital Signature Algorithm (ECDSA). This paper discusses the cryptography including different types of digital signatures based on the kind of key and a few algorithms of Digital Signature RSA, ElGamal, and ECDSA.

Keywords—Digital Signature, Digital Signature Algorithm, DSA, ECDSA, RSA, DSS.

I. INTRODUCTION

The signature is always created by the use of a private key where only the user is known the private key. The signature is also verified makes use of a public key which match to the private key. Every user has one public key and one private key pair. Public keys are known by everyone, and can be used to verify the user signature. However private key is not shared, private key is used in generation signature, which is done only by the user [1].

Kumar et al. [17] proposed an efficient two-factor user authentication framework for wireless sensor networks, which is based on password and smart card and it uses one-way hash function. This scheme provides mutual authentication and gives user facility to change password at need. But the problem here is, it does not provide privileged insider attack as the password is sent to the base station in plain text. It also suffers from the synchronization problem as it uses the time stamp for avoiding replay attack.

Digital signatures used to reveal unauthorized amendment to data. However, the recipient of a digitally signed document in proves to a third party that the document was indeed signed by the human who it is assumed to be signed. This is recognized as nonrepudiation, because the human who signed the document cannot refuse the signature at a later time. We can use Digital signature algorithm in electronic funds transfer,

e-mails, data storage, electronic data interchange, software distribution, and just about any application that would need to assure the integrity and originality of data [2].

Digital Signature Algorithm (DSA): is set in Digital Signature Standard (DSS). The specification contains standard for the generation domain parameters, generation of private key and public key pairs, verification and generation of digital signatures.

RSA digital signature algorithm: Is set in Public Key Cryptography Standard and American National Standard (ANS). FIPS 186-4 notarize the use of enforcement of either or together of these standards and assign extra requirements.

Elliptic Curve Digital Signature Algorithm (ECDSA): Is set in ANS X9.62.

The public key is applied in the process of signature verification. However, the public key does not required to be preserved secret, where its integrity should be maintained. Using the public key, anybody can prove a correctly signed message [3],[15].

In signature verification and generation processes, the message is transform to a fixed-length personification of the message by means of a confirmed hash function. The digital signature and the original message are both made ready to a verifier (Fig. 1).

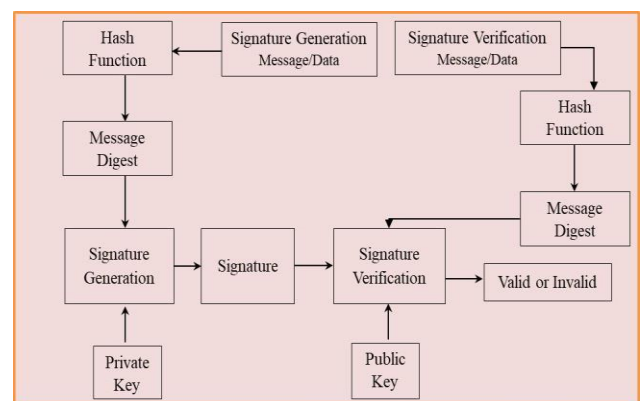


Fig. 1 Digital Signature Processes

II. APPLICATIONS OF DIGITAL SIGNATURES

Digital signatures can supply assertion of the evidence to exporter, identity, and status of an electronic document and acknowledging informed consent of the site by a signatory [4] and [6].

Integrity: There is status where is a need for dependability where the message has not been changed during transmission. However, if a message is digitally signed, any modification in the message after signature will abrogate the Digital signature.

Confidentiality: it is used to protect the information and only authorized users can got it.

Authentication: to authenticate the origin of message, we can use digital signatures. For instance, Alice will sends Bob the encrypted message digest. In order for Bob to authenticate the signature he should apply the same hash function as Alice to the message she sent to him. He will also decrypt the encrypted message using Alice is public key and compares the two. If the two are the same then he has successfully authenticated the signature. If the two are not match then either an error happen through transmission or someone was trying to impersonate Alice. However, valid signature display that the message was sent by the ownership user, when the digital signature of the ownership is constrained to a particular user.

Non-repudiation: here the information that we may have received can be definitely attributed to somebody and there is no way that they can take it back. They definitely said what is that we have in this message right here.

III. COMPARISON OF RSA WITH ECDSA

ECDSA and RSA are applied to protect the data packet inside the network. For security, the RSA and ECDSA equip the same security level with the short and large keys. To use the cryptography for data transmissions, verification, less power consumption, faster signature generation, the minimal size key of ECDSA probably allow for minimum estimation fit light-weight wireless systems and devices.

Table 1: Comparison of RSA and ECDSA

	Domain	Key Generation	Execution Time	Verification Time	Signature Time	Complexity	Security
ECDSA	Lightweight devices	Faster	Fast	Slow	Slow	Discrete logarithm	High
RSA	Super computers Laptops, PC	Fast	Slow	Fast	Fast	Integer Factorization	High

Key Generation: The key generation is faster than the RSA because of the key lengths differences [12]. RSA create and generate utilizing two pig prime numbers so it takes so much time than the minimum ECDSA keys which is generated.

Verification time: RSA verification For RSA the verification time is so fast because it is simpler in terms of cryptographic computations as only a lower of modular multiplications is needful.

Execution Time: Considering the RSA keys are significantly bigger than the ECDSA keys, RSA takes longer time to generate the key compared to ECDSA.

Signature Time: The time needed for ECDSA operations to generate signatures is so mash slower because of the convergence among the key sizes. The time needed for RSA

operations to generate signatures so fast increases because of its big keys which is not reasonable and keep delay packets from being transferred.

IV. COMPARISON OF RSA WITH ECC

The ECC have full exponential time where the RSA takes sub-exponential time. For instance, the ECC for 160 bit key size take 9.6×10^{11} MIP years [8] where the RSA of 1024 bits for key size takes 3×10^{11} MIP years.

With the smaller key size the ECC offers same security level. The size of data in the RSA is smaller than the ECC and the power computational is smaller for ECC compared to the RSA.

V. DIFFERENT TYPES OF DIGITAL SIGNATURES

Digital Signatures with appendix: need the main message as input to the verification. For instance, DSS, ElGamal, DSA.

Message recovery with Digital signatures: A priori knowledge of the main message is not required for the verification where the main message can be recovered via the signature itself. For instance, Rabin, RSA, Rueppel.

Digital signatures based on RSA:Includes the features of both Digital Signatures with appendix and Digital signatures with message recovery

Blind signature schemes [14]: Allows sender A to remain anonymous so that spending patterns cannot be monitored

VI. SIMPLE DIGITAL SIGNATURE ALGORITHM

The public key is used in the signature verification process. The public key need not be kept secret, but its integrity must be maintained. Anyone can verify a correctly signed message using the public key.

Signing Process by Sender: Message digest (MD) is generated. A message digest is a “summary of the message that is going to be transmitted,” that generated by a set of hashing that were agreed to by both users. The hashing algorithm guarantees the integrity of a message by making a fully different hash value when a single part of the message changes. The encrypted message digest is created when the Message digest encrypted with a sender’s private key that is called a digital signature (DS). A DS is enclosed [16] with the message and sent to the receiver (Fig. 1).

Signature Verification Process by Receiver: Using the sender’s public key, a receiver decrypts the digital signature to obtain the message digest generated by the sender. Using the same hashing algorithm, the receiver calculates the MD of the received message. The acquired MD value is compared with the sender’s MD value. If they are identical, then the message is not altered and the originality is assured. At this phase, if decrypting the message using the sender’s public key [5] results in a faulty message digests, then the message has been changed and cannot be trusted. However, it is clearly shown that the integrity of the message is maintained but not the privacy, since the message is sent plainly.

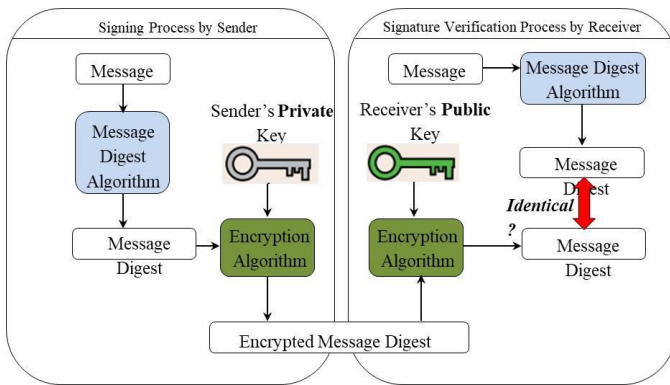


Fig. 1 Simple digital signature algorithm

This may be suited to a situation where confidentiality is not an issue. In order to ensure confidentiality in communication, the message should be encrypted. Basically, a digital signature scheme typically consists of three algorithms, namely key generation, signing and signature-verifying algorithms. The techniques are varied, and these systems are categorized based on their mathematical problems. Generally, public key systems are grouped into three main classes which are based on integer factorization (i.e. RSA), discrete logarithm (i.e. DSA), and elliptic curve discrete logarithm (i.e. ECDSA). The security degrees of all the techniques are based on the hardness of mathematical problems.

A. Digital Signature Algorithm (RSA)

RSA is a public-key cryptosystem that gets its name in 1977 from its inventors – Ronald Rivest, Adi Shamir and Leonard Adleman and was developed [4]. It has since withstood years of extensive cryptanalysis. It is used for electronic commerce and many other secure communications over the Internet. The RSA system utilizes one way functions of a additional complex nature [13]. RSA is a Block cipher in which the plain text and cipher text are integers between 0 and $e - 1$ for some integer e . RSA gets its security from the difficulty of factoring large numbers.

Working of RSA: Select 2 random large prime numbers r and q of almost equal length. Compute their product $e = rq$. The Euler's Totient function $\phi(e)$ is computed, i.e. $\phi(e) = (r - 1)(q - 1)$. We then choose two keys a and b such that, $a \cdot b \equiv 1 \pmod{\phi(e)}$. One of the keys say a is made public while the other key b is kept a secret. At this point, we no more require r , q and $\phi(e)$. We can discard these values. If we have a message H , encryption of H is $C = H^a \pmod{e}$, C is the resultant cipher text. Decryption of C is achieved by $H' = C^b \pmod{e}$.

Consider $H' = M^{ab} \pmod{e} = H^{k\phi(e) + 1} \pmod{e}$ (Since $a \cdot b \equiv 1 \pmod{\phi(e)} \Rightarrow H' = H \cdot H^{k\phi(e)} \pmod{e} = H \pmod{e}$
(It can be proved that $x\phi(e) \equiv 1 \pmod{e}$)

Hence we see that $H = H'$. Thus we have achieved efficient encryption and decryption using RSA.

B. Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA was first proposed in 1992 by Scott Vanstone [9]. Elliptic Curve Digital Signature Algorithm (ECDSA) Key

Generation [7].

User A has these steps where r is a large prime:

Select a random integer $d \in [1, e - 1]$.

Compute $Q = d \times R$.

The private and public keys of the users A are Q and d , respectively.

The other users can check if the public key is valid by;

Checking that $Q \neq 0$.

Checking that xQ and yQ are properly represented elements of F_q .

Checking that Q is on the elliptic curve defined by a and b .

Checking that $eQ = Q$.

If any of these checks fail the public key Q is invalid, otherwise Q is valid. The following procedure describes how to generate the signature.

ECDSA Signature Generation

Select a pseudorandom integer $k \in [1, e - 1]$.

Computing $k \times R = (x_1, y_1)$ and $r = x_1 \pmod{e}$.

If $x_1 \in GF(2k)$, it is assumed that x_1 is represented as a binary number.

If $r = 0$ then go to Step 1. Compute $k^{-1} \pmod{e}$.

Compute $s = k^{-1}(H(h) + d \cdot r) \pmod{e}$.

Here H is the secure hash algorithm SHA-1.

If $s = 0$ go to Step 1.

The signature for the message h is the pair of integers (r, s) .

ECDSA Signature Verification

Verify that r and s are integers in the interval $[1, e-1]$.

Compute $c = s^{-1} \pmod{e}$ and $H(h)$.

Compute $u_1 = H(h) \cdot c \pmod{e}$ and $u_2 = r \cdot c \pmod{e}$.

Compute $u_1 \times R + u_2 \times Q = (x_0, y_0)$ and $v = x_0 \pmod{e}$.

Accept the signature if $v = r$.

ElGamal Digital Signature

El Gamal Key Generation

The prime r , a generator g of field Z_r^* , A's private key d_A is a random integer from the interval $[1; r-1]$ and her public key is $y_A = g^{d_A} \pmod{r}$.

C. El Gamal Signature Generation

Select a random integer k from interval $[1; r-1]$, satisfying $\gcd(k; r-1) = 1$; [10][11].

Compute $k^{-1} \pmod{r-1}$; Compute $r = g^k \pmod{r}$;

Compute $s = k^{-1}\{h(h) - d_A r\} \pmod{r-1}$. h is the hash function: $\{0; 1\} \rightarrow Z_r$.

$(r; s)$ is A's signature of message h .

El Gamal Signature Verification

Verify that $1 \leq r \leq r - 1$;

Compute $v_1 = y_A^r g^k \pmod{r}$;

Compute $h(h)$ and $v_2 = gh(h)$;

Accept if and only if $v_1 = v_2$.

Schnorr Digital Signature; Schnorr Key Generation

Primes q, r , satisfying $q \mid (r-1)$, the generator α of the unique cyclic subgroup of Z_r^* (satisfying $\exists u \in Z_r^*, \alpha = u(r-1)/q \pmod{r}$, but $\alpha \neq 1$). A's private key d_A is a random integer from the interval $[1; q-1]$, and her public key is $y_A = \alpha^{d_A} \pmod{r}$.

Schnorr Signature Generation

Select a random integer k from interval $[1; q-1]$;

Compute $r = \alpha k \bmod r$, $e = h(h \parallel r)$ and $s = dAe + k \bmod q$. h is the hash function $\{0; 1\} \rightarrow Z_q$. $(s; e)$ is A's signature of message h .

Schnorr Signature Verification

Compute $v = \alpha s y_A^{-e} \bmod r$ and $e' = h(h \parallel v)$;

Accept if and only if $e = e'$;

VII. COMPARISON OF SECURITY LEVELS

An elliptic curve, defined modulo a prime r , is the set of solutions (x,y) to an equation of the form:

$$y^2 = x^3 + ax + b \pmod{r}$$

For two numbers a and b .

If (x,y) satisfies the above equation

Then $R=(x,y)$ is a point on the elliptic curve.

In fact, an elliptic curve can also be defined over the finite field consisting of 2^h elements. Such a representation offers extra efficiency in the operation of the ECC.

Using some particularly deep mathematics, it is possible to define the "addition" of two points on the elliptic curve.

Suppose R and Q are both points on the curve, then $R + Q$ will always be a different point on the curve. The elliptic curve discrete logarithm problem can be stated as follows. Fix a prime p and an elliptic curve. xR represents the point R added to itself x times. Suppose Q is a multiple of R , so that $Q = xR$ for some x . Then the elliptic curve discrete logarithm problem is to determine x given R and Q .

Fig. 2 compares the time required to break the ECC with the time required to break RSA or DSA for various modulus sizes using the best general algorithm known. The values are computed in MIPS years. A MIPS year represents a computing time of one year on a machine capable of performing one million instructions per second. As a benchmark, it is generally accepted that 1012 MIPS years represents reasonable security at this time, since this would require most of the computing power on the planet to work for a considerable amount of time.

COMPARISON OF SECURITY LEVELS
ECC AND RSA & DSA

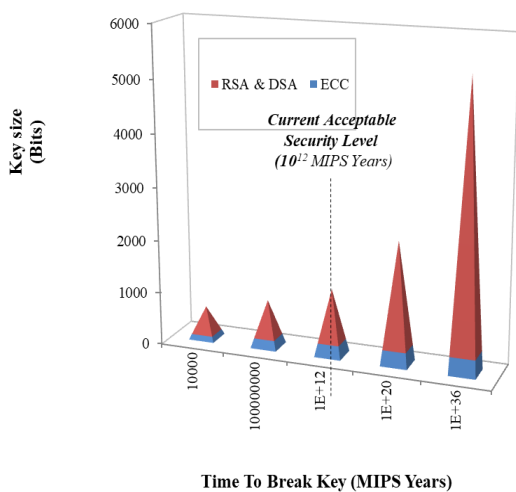


Fig. 2 Comparison of Security Levels

However, from this figure, we can see that to obtain sensible security, DSA and RSA must use a 1024-bit modulus, and a 160-bit modulus must be enough for the Elliptic-curve cryptography (ECC). The ECC and security gap required a smaller modulus more than the RSA or DSA, among the systems grows where the key size is increased. For instance, 300-bit ECC is a large deal extra secure than 2000 bit DSA or RSA [6].

I. CONCLUSION

Digital signatures are fundamental in today's modern world to verify the sender of a document's identity. In this paper we perused the concept of Cryptography including different types of digital signatures based on the kind of key and a few algorithms first one is RSA algorithm, second is the ElGamal algorithm, and the third is Elliptic Curve Digital Signature Algorithm (ECDSA). Electronic digital signature (EDS) has eliminated most of the problems inherent in the signature on a paper document; however the future work would consider the proof of identity of who sent the document and confirmation of the authenticity of the document.

ACKNOWLEDGMENT

This work was supported by Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00245), And it was also supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (grant number: NRF2016R1D1A1B01011908).

REFERENCES

- [1] C. Hai-peng, S. Xuan-jing, W. Wei "Digital Signature Algorithm Based on Hash Round Function and Self-certified Public Key System" 2009 First International Workshop on Education Technology and Computer Science.
- [2] G. Agnew, R. Mullin and S. Vanstone, "An implementation of elliptic curve cryptosystems over 2155 ", IEEE Journal on Selected Areas in Communications, 11 (1993), 804-813.
- [3] W. Romney "Digital signature signing engine to protect the integrity of digital assets by Gordon, 1-4244-0406- 1/06/\$20.00 © 2006 IEEE.
- [4] M, S.Q.Y A.A "New Efficient Digital Signature Scheme Algorithm based on Block cipher by Prakash Kuppuswamy" , IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661, ISBN: 2278-8727 Volume 7, Issue 1 (Nov. - Dec. 2012), PP 47-52
- [5] R.R.L, Shamir, A., and A.L., "A method of obtaining Digital Signatures and Public key cryptosystems", Comm.ACM, 21, 1978.
- [6] S. Baral "An Efficient Blind Digital Signature Protocol Based on Elliptic Curve" International Journal of Technology Enhancements and Emerging Engineering Research, vol 2, Issue 9 ISSN 2347-4289
- [7] A. Khalique, K. Singh, S. Sood., "Implementation of Elliptic Curve Digital Signature Algorithm" International Journal of Computer Applications (0975 – 8887) Volume 2 – No.2, May 2010
- [8] Raju, G.V.S., Akbani, R., 2003. Elliptic Curve Cryptosystem And Its Application. In Proceedings of the 2003 IEEE International Conference on Systems Man and Cybernetics (IEEE-SMC), 1540-1543.
- [9] Vanstone, S. A., 1992. Responses to NIST's Proposal Communications of the ACM, 35, 50-52.
- [10] Douglas Stinson, Cryptography Theory and Practice, by CRC Press, 1995.
- [11] T.El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. Advances in Cryptology: Proceedings of

CRYPTO' 84, Volume 196 of Lecture Notes in Computer Science, pages 10-18. Springer-Verlag, 1985

- [12] A. I. Ali., "Comparision and Evaluation of Digital Signature Schemes Employed in NDN Network" International Journal of Embedded systems and Applications(IJESA) Vol.5, No.2, June 2015
- [13]P. J, F and J. M. J, Using the RSA Algorithm for Encryption and Digital Signatures: Can You Encrypt, Decrypt, Sign and Verify without Infringing the RSA Patent? Alston & Bird LLP, July 9, 1997.
- [14] <http://www.iet.unipi.it/g.dini/Teaching/sanna/lecturenotes/applied-cryptography-digital-signature.pdf>
- [15] A. Abdullaev, Mohammed Abdulhakim Al-Absi, A. A. Al-absi, M. Sain, H. J.Lee., "Survey on Digital Signature algorithms", International Conference on Culture Technology[ICCT], Nov. 15~18 Zhijiang College of Zhejiang University of Technology Shaoxing, China, pp 408-411, 2018
- [16] A. Abdullaev, Mohammed Abdulhakim Al-Absi, A. A. Al-absi, M. Sain, H. J.Lee., "General Study of Digital Signature Schemes", International Conference on Culture Technology, Zhijiang College of Zhejiang University of Technology Shaoxing, China, ICCT Nov. 15~18 pp 412-415, 2018
- [17] Kumar, P., Sain, M. and Lee H. J. "An Efficient TwoFactor User Authentication Framework for Wireless Senor Networks" ICACT 2011



Mohammed Abdulhakim Alabsi received his BS degree in Computer Application from Bangalore University in India. He earned his (MS) degree at Dongseo University- South Korea in 2018. Currently, he is a Ph.D. student in the Department of Information and Communication Engineering at Dongseo University, Korea. His research interests include IoT, VANET, Cryptology, Network Security, Computer Networks and Digital Communications.

AZamjon Abdullaev received his BS degree in finance from Tashkent financial Institute in Uzbekistan 2011-2015. Currently, he is a Master student in the Department of Information and Communication Engineering at Dongseo University, Korea. His research interests include Mobile Banking, Wireless Sensor Networks, Cryptography, and Network Security.

Ahmed Abdulhakim Al-Absi is an assistant professor and head of smart computing department at Kyungdong University - Global Campus in South Korea. He earned his Ph.D. degree in ubiquitous computing at Dongseo University- South Korea in 2016. His research interests include Database Systems, Big Data, Hadoop, Cloud computing, Distributed systems, Parallel computing, High-performance computing, VANET, and bioinformatics. He received a Master of Science (MS) degree in information technology at University Utara Malaysia-Malaysia in 2011, and a Bachelor of Science (BS) degree in computer applications at Bangalore University- India in 2008

Mangal Sain received the M.Sc. degree in computer application from India in 2003 and the Ph.D. degree in computer science in 2011. Since 2012, he has been an Assistant Professor with the Department of Computer Engineering, Dongseo University, South Korea. His research interest includes wireless sensor network, cloud computing, Internet of Things, embedded systems, and middleware. He has authored over 50 international publications including journals and international conferences. He is a member of THIS and a TPC member of more than ten international conferences.

HoonJae Lee received his BS, MS, and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1985, 1987, and 1998, respectively. He is currently a professor in the Department of Information Communication Engineering at Dongseo University. From 1987 to 1998, he was a research associate at the Agency for Defense Development (ADD). His current research interests include Password Theory, Network Security, Side-Channel Attack, and Information Communication/Information Network.