ISSN: 2321-7782 (Online)
ISSN: 2347-1778 (Print)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Online Document Repository System

G Manoj[1]
Dept. of CSE
BMSCE
Bengaluru, India

Ishan Deep[2]
Dept. of CSE
BMSCE
Bengaluru, India

Kalyani V[3]
Dept. of CSE
BMSCE
Bengaluru, India

Sahana K.C[4]
Dept. of CSE
BMSCE
Bengaluru, India

Madhavi R.P[5]
Associate Professor,Dept. of CSE
BMSCE
Benagluru, India

Abstract: ***The main idea here is to create a central document repository so that it can be used to store documents in a digital format after being digitally signed by the underlying appropriate digital signature algorithm. Once authenticated, the digital copy of that document should be brought to the repository and stored securely with access control limited to particular authorities. The end user here will have the power to restrict his/her documents access controls.***

Keywords: Document Management System, DMS, DSA, RSA, ECDSA, Digital Signatures.

I. INTRODUCTION

An online document repository system aims to eliminate the existing conventional methodology of paperwork based document management which wastes both time and energy apart from being highly disorganized. Although the existing document management system (DMS) products try to achieve that, they still lack some basic security features which make them unsecure and non-trustworthy. The Online document repository system explained below, aims to incorporate a few security enhancements so as to achieve advantages such as non-repudiation, confidentiality and data integrity. Given these advantages, a product providing these, when implemented on a larger scale, could bring significant changes in the present scenario of document handling. The required research to understand the underlying basics is provided in this document, with the advantages and limitations of the existing work.

II. SURVEY

The presented paper here brings out the idea of integration of a document management system (DMS) and digital signatures to achieve the best of the features. Hence, this paper is divided into two sections. The first discusses the various document management system (DMS) approaches and the second focuses on the digital signature and different algorithms such as DSA, RSA and ECDSA.

A. Document Management System (DMS)

People many times lose their documents and they will have to spend lot of time in researching the "lost" documents and so on. In order to overcome this problem the solution they can make use is DMS. DMS is like a traffic cop that brings order to these documents [1].

The primary functionality of a document management system is to record, manage and retrieve electronic data content. The further classification or sub-types of a DMS arise from the methodology they use in order to provide the services and the way in which they enhance the functionality of it.

According to [2], which describe a document management system wherein a DMS is implemented over a computer network system, a document can be attested by the owner of that document by the means of using his signer ID provided to him by the network when he registers. The unique signer ID creates a signature image of the owner which is attached to the file concerned and sent over the network to be stored onto the repository which he calls the attest data storage system. The receiver who is in need of this file access it and verifies the authenticity of the document by using necessary credentials. Though this method provides a way to authenticate documents, it does not concern over the data associated with the file hence providing no data-integrity.

The method of DMS described according to [3] is a slight variation of the method of DMS in [2]. This method defines the system as a relationship between a DMS and one or more clients which he calls a multi-function device (MFD). An MFD is incharge of creating an electronic instance of a physical document and sending over to the repository after attaching a signature image of the owner using parameters such as USER_ID and MFD_ID etc. Once on to the repository, the signature is matched with an already existing copy of signature image to verify the authenticity of it. He also provides a way of authenticating the user by using login commands and notifying the owner of the document whenever a user accesses it. This type of system though provides authenticity and access control, but is vulnerable to many attacks since it stores a copy of the signature image in the database.

An implementation example is presented in [4], according to which, they try to ease the inter-department communication and also public distribution of documents to the university members. Their work, termed as SUNIDOC, is based on customized software created by a company named SIVECO Romania, which specializes in this domain. The system was accessible using a web interface over the network in the university and was built using technologies like J2EE, XML, XSL and others. Every document in the repository had a unique registration number which was used for easy retrieval of the document. The product also looked onto features such as archiving of documents and backup and recovery options. A full pledged implementation was later done with a centralized architecture and core being at University Rectorate from which all the research centers and other departments could have access to the documents shared.

B. Digital Signatures

Before the digital signatures, the world was dominated by the traditional paper signatures. In case of paper signatures, if anyone wants to send a signed copy of a document, they would have to take the printout of the document, sign it, scan and then send it to the receiver. Also, paper signatures were easy to forge. Additional information or support along with paper signatures like organization seals or notaries could be used to provide security and assurance in case of disputes later. The whole process was hence time consuming and less secure.

Digital signatures were conceived in 1976. PKI based digital signatures are very popular now. Digital signatures are far more secure and easy to use in the electronic form. Digital signature provides authentication and non-repudiation with the help of private and public keys. Computer generated signature for the signing party provides appropriate authentication to the device. The device stores the private key in secure storage with no unauthorized access. This key is used in the rare case of verification failure to find the party. The digital signature changes for each transaction and hence is dynamic and secure. A variation in single word or a bit, can be easily detected by the digital signature.

According to [5], digital signatures help the businesses and more specifically e-commerce to evolve by moving the paper signatures online. Legislation related to digital signatures such as the Electronic Signatures in Global and National Commerce

Act (E-Sign) and the Uniform Electronic Transactions Act (UETA) are important as they remove uncertainties regarding legal interpretations of online transaction systems. Below are the few most popular digital signature algorithms.

C. Signature Algorithms

Currently, the signature algorithms that are used in order to provide data integrity and authenticity are the well-known Digital Signature Algorithm (DSA), the RSA signature algorithm and the advanced version of DSA, the Elliptic Curve Digital Signature Algorithm (ECDSA). These three algorithms and their comparisons are given in this section.

» ***Digital Signature Algorithm (DSA):***

According to the US patent paper [6], US5231668, this method is used to provide the digital signature of say, a message m. In this method, we require a pair of public and secret keys (y and x) for every signer and a pair of public and secret values (r and k) that is generated by each message. From these keys and values, a value r is calculated according to the equation $r= (g^k \bmod p) \bmod q$, which is public. Once we get the value of r, we need to calculate s, which is the signature, $s= k^{-1}(H(m)+xr) \bmod q$ where H is the hashing function. The message m, along with the signature (r, s) calculated is accordingly transmitted. On the receiver side, the message is subjected to a verification process where the hash of the message is calculated and compared with the signature. The received values of r and s are tested whether they are congruent to 0 mod g. Along with this, the value of r is tested to determine whether it's equal to v mod q, where v is computed from m, s, y, and r. If the signature is legitimate then $v=g^k \bmod p$.

In order to provide security, the DSA algorithm is used along with Diffie-Hellman key exchange algorithm. According to [7], this has 3 types of protocols.

- One-Way Protocol: where it's used for single sided transmission.
- Two-Way Protocol: for interactive transmission.
- Three-Way Protocol: for key confirmation.

The e-voting system, according to [8], use the normal RSA algorithm and Kerberos concept for the generation of secret key and the tickets which is susceptible to attacks like factorization, implementation, short-message attack, cycling attack etc. The common practice of maintaining the data was signing and encryption, but a method called signcryption was developed according to [9] in China, where only single method (algorithm) was used (ROM- Random Oracle Model).

In order to apply the signing process, first we need to get the keys. According to the recent study on Information Centric Networking [10], ICN- a new architecture, the integrated Diffie-Hellman key exchange protocol has been redesigned for using it in the key distribution mechanism. The new protocol that was designed – z Formation approach where the link/node identifiers (LID's) are changeable and the secret key is shared upon that. In this procedure, the secret key is transmitted in a public channel.

Though DSA algorithm is a secured method to transmit the messages, it is subjected to two types of attacks. The first type of attack includes attacks that are designed to recover the secret key x. The second type of attack includes a set of attacks designed to forge the signatures without recovering x. And the other drawback is it uses too many variables which take lot of time for processing and also cumbersome. Hence the advanced version of DSA is ECDSA, which is explained later.

» ***RSA:***

RSA was given by Ron Rivest, Adi Shamir, and Leonard Adleman in the year 1977 when it was first publicized in the *Scientific American*. Though being around thirty years old, it is still one of the most successful algorithms ever used. According to [11], there have been numerous attacks on RSA, but none devastating. Later, these attacks were proved out to be because of the poor implementation of the algorithm rather than the core architecture and design of RSA itself. Currently, the most popular

signature algorithm is RSA with SHA-1 hashing algorithm, using keys that are 1024 or 2048 bits long. The signing methodology, according to [12], is as follows

hash = SHA1 (data)

signature = (01 | FF* | 00 | prefix | hash) ** e (mod n)

where, e is the private key of the signer and n is the modulus of signers public key.

An improvement over the traditional RSA encryption system was provided according to [13]. This method was a minor modification of the basic RSA algorithm and used Short Range Natural Numbers (SRNN). Hence termed SRNN algorithm. These short range numbers also improved some of the security features over the traditional RSA algorithm.

Another flavor of RSA was given according to [14], where, instead of using a pair of keys between the sender and receiver, they used multiple public keys to enhance the security, but, at the cost of speed.

» ***Elliptic Curve Digital Signature Algorithm (ECDSA):***

According to [15], ECDSA is the elliptic curve analogue to the DSA. It does not use the complete group of Z_p^*. This algorithm make use of the subgroup of Z_p^* where some of the members are replaced by a group of points on an elliptic curve over a finite field. The security of this curve is mathematically calculated by the intractability of the elliptic curve discrete logarithm problem (ECDLP). ECDLP appears to be harder than normal DLP though it uses minimum number of variables. The strength of each bit of the key is comparatively stronger than the key bit in DSA. Since the numbers of variables used is less than the conventional DSA, and also small parameters, this type of cryptosystems replaces the DL systems. Though it has small parameters, it provides equivalent level of security as DSA. The advantages because of this algorithm, according to [16], is mainly, speed (faster computations), certificates and smaller keys. These advantages are important where the bandwidth, processing power, storage (memory space) and power consumption parameters are constrained.

According to [17], till date ECDSA cryptosystems have been implemented in JAVA (JDK 7). The scheme implemented uses 256 bit key size and employs 256 bit SHA-1 hash algorithm. Time estimates are as follows: signing of 100 messages takes 163ms and verifying the same takes less than 300ms where the size of the message is 0.5kB. ECDSA uses inexpensive modular arithmetic operations like modular multiplicative inverses, addition etc. for signing and verification processes. Hence, using this is preferred more than group and normal signature process.

According to [18], an attack was shown on OpenSSL ECDSA called as FLUSH-RELOAD attack, where the nonces were recovered. OpenSSL implementation of ECDSA uses the Montgomery ladder algorithm. This algorithm behaves in a regular manner. This algorithm consists of 2 processes, multiplication and addition depending on the value of the bit. If the bit is 0, then an extra step of addition process comes into picture. Due to this, an extra amount of time is spent, and hence the attacker will realize that the bit as 0 and could relate it with the original message. From this, the scalar secret key is also found.

ECDSA can be applied for two types of messages, the first one is fixed length message and the latter is variable length. According to [19], ECDSA on variable length message is comparatively better than on fixed length and text based encryption. The reason for this is the speed which is mainly dependent on the scalar multiplication, which in turn constitutes for the efficiency of the whole system.

According to [20], many Standards and Drafts specify ECDSA. Among those, the ones that are officially approved by their respective accredited organizations are ANSIX9.62, FIPS 186-2, IEEE 1363-2000, and ISO 14888-3. It has also been standardized by the consortium of companies- Standards for Efficient Cryptography Group (SECG) which mainly dealt with the problems of interoperability.

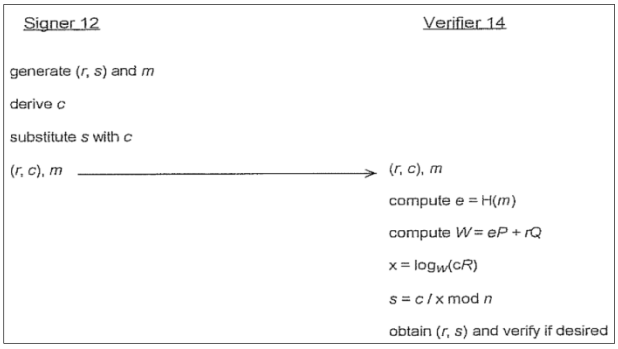

Fig. 1 Compressed ECDSA procedure according to [21].

According to Fig.1, H(m) is the hash of message m. c and s are integers, P,Q,R,n are large prime numbers.

D. Comparisons

1. DSA and RSA:

The performance and security measures of RSA are very much comparable to DSA defined before. The key generation and signature generation are faster in DSA whereas the signature verification is faster in RSA.

2. ECDSA and DSA:

- According to [16], both algorithms are based on the ElGamal signature scheme and use the same signing equation.
- According to [16], in both ECDSA and DSA, the values that are relatively difficult to generate are the system parameters which are public and their generation can be independently checked and audited.
- According to [18], in their current version, both DSA and ECDSA use the SHA-1 as the sole cryptographic hash function.
- According to [16], the private key d and the per-signature value k in ECDSA are defined to be statistically unique and unpredictable whereas they are random in case of DSA.
- According to [16], the analogous attack on weak system parameters does not apply in ECDSA.

3. ECDSA and RSA:

- According to [20], breaking an RSA key requires you to factor a large number. We are pretty good at factoring large numbers and getting better all the time. Breaking an ECDSA key requires you to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP). The mathematical community has not made any major progress in improving algorithms to solve this problem.
- According to [22], sub-exponential algorithms for solving elliptic curve discrete logarithm problem are not known whereas there are algorithms which can solve RSA in sub-exponential time.
- According to [22] and [23], the level of security offered by ECDSA is same as RSA even though the size of key used in ECDSA is much smaller than RSA. RSA with key size of 1024 bits takes 3x10^11 MIP years with best known attack whereas ECDSA with 160 bit key size takes 9.6x 10^11 MIP years.
- According to [24], data size for RSA is smaller than ECDSA.
- According to [23], as encrypted message is a function of key size and data size for both RSA and ECDSA and ECDSA key size is relatively smaller than RSA key size, thus encrypted message in ECDSA is smaller.

- According to [23], computational power is smaller for ECDSA when compared to RSA.
- According to [25], key size of ECDSA is much smaller than that of RSA.
- According to [25], the strength per key bit is substantially greater when compare with conventional discrete logarithm systems.

III. Conclusion

As explained in the previous sections DMS come across a number of challenges and limitations that are faced in order to maintain documents in a secured manner. The repository that we are trying to maintain provides data integrity, authentication and non-repudiation by means of digital signatures and data confidentiality by means of encryption, which the previously explained types of DMS failed to do. And also, since we are not maintaining a copy of the signature, this makes the system less vulnerable to attacks. The signature algorithm, which is an important factor that decides the security of the system, has to be carefully chosen. As a DSA and RSA key will work everywhere, in practice, whereas ECDSA support is newer, hence, some old client or server may have trouble with ECDSA keys. We would consider all these criteria regarding RSA, DSA and ECDSA and proceed further.

Acknowledgement

The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India.

References

1. J.Heckman,"Why document management-a white paper", Heckman Consulting (201) 792-0022, November, 2008.
2. T.Hirose, "Document management system", US Patent No. US6907529 B1, 2005.
3. R.Cowburn, "Document Management System", US Patent No. US 20070115497 A1,2007
4. M. Costoiu, V. Plesu, R. Isopescu, S. Soriga, G. Alesincu and I. Arsene,"Electronic document management information system for universities" 2012.
5. H.Zhu and D.Li,"Research on Digital Signature in Electronic Commerce", IMECS Hong Kong, 2008.
6. D.W. Kravitz,"Digital signature algorithm", US Patent No. US 5231668 A, 1993
7. L.Harn, M.Metha and W.J.Hsin,"Integrating Diffie-Hellman Key Exchange into the Digital Signature Algorithm", IEEE Communications Letters,March,2004.
8. T.Afrin and K.J.Satao,"Detailed Implementation of E Voting System for on Duty Persons using RSA Algorithm with Kerberos Concept", IJIRCCE , September, 2013.
9. C.Zhou, "A Multi-Receiver ID-Based Generalized Signcryption Scheme", JiuJiang, 2005.
10. B.A.Alzahrani, V.G.Vassilakis and M.J.Reed,"Key Management In Information Centric Networking", IJCNC , November, 2013.
11. D.Boneh, "Twenty Years of Attacks on the RSA Cryptosystem", Survey Paper, Volume 46, Number 2, American Mathematical Society (AMS).
12. D.E.Eastlake,"RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS)", RFC- 3110b, IETF, 2001.
13. S.Sharma, J.S.Yadav and P.Sharma, "Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm", IJARCSSE, August, 2012.
14. A.A.Ayele and Dr. V.Sreenivasarao, "A Modified RSA Encryption Technique Based on Multiple public keys", International Journal of Innovative Research in Computer and Communication Engineering, June, 2013
15. P.Hoffman and W.C.A. Wijngaards, "Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC", RFC-6605, IETF, 2012.
16. D.B. Johnson and A.J.Menezes,"Elliptic Curve DSA (ECDSA): An Enhanced DSA" August,1999.
17. L.Malina,"Privacy preserving Cryptographic protocols for secure Heterogeneous Network", BRNO, 2014.
18. Y.Yarom and N.Benger, "Recovering OpenSSL ECDSA Nonces Using the FLUSH+RELOAD Cache Side-channel Attack", IACR, February 24, 2014.
19. R.B.Mulinti and Dr.G.A.Ramachandra, "Implementation of Elliptic Curve Digital Signature Algorithm Using Variable Text Based Message Encryption", International Journal Of Computational Engineering Research Vol. 4 Issue. 12, December, 2014.
20. D.Johnson, A.Menezes and S.Vanstone,"The Elliptic Curve Digital Signature Algorithm (ECDSA)", University Of California-Santa Barbara, 2012.
21. S.A.Vanstone,"Compressed ECDSA signatures", Patent No US8631240 B2, January 14, 2014.
22. A.Khalique, K.Singh and S.Sood, "Implementation of Elliptic Curve Digital Signature Algorithm", International Journal of Computer Applications, May, 2010.

23. K.Jarvinen, "Cryptoprocessor for Elliptic Curve Digital Signature Algorithm", August 7, 2007.

24. S.R.Patel, Prof. K.Shah and G.R.Patel, "Study on Improvements in RSA Algorithm" IJEDR Vol.1, Issue 3, pp.142 - 145, December, 2014.

25. J.Muthukuru, B. Sathyanarayana,"A Secure Elliptic Curve Digital Signature Approach without Inversion", International Journal of Engineering and Advanced Technology (IJEAT), December, 2013.

