International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Survey on Digital Signatures

Rabeya Sultana, Tashrifa Shahid

Prime University, Dhaka, Bangladesh.

ABSTRACT

The significance of authentication is expanding thanks to the growth of online transactions over the web. There is a have to initiate a framework for the authentication of computer-based information. A Digital Signature is one of all the authentication mechanisms. A legitimate digital signature gives a recipient ground to trust that the message was created by a familiar sender, which it absolutely was not changed in transit. Digital signatures are frequently used for e-commerce such as e-cash, e-voting, and e-auction where it's essential to detect forgery or tampering. Several asymmetric cryptosystems generate and verify digital signatures using different algorithms and procedures. This paper performs security analysis of Some common algorithms for key generation and finds out the advantages and disadvantages of that algorithm. Experiments results are specified to analyze the efficacy of every algorithm..

Keywords: Cryptography, Digital Signature, Authentication, Algorithms, Security

1. Introduction

In recent times, security plays a serious role in communication. the problem of information or message security arises thanks to the tremendous increase within the use of the net and various communication channels. it's necessary to safeguard our valuable information over differing types of networks. A digital signature is crucial for these reasons [1].

A Digital Signature could be a checksum that depends on the fundamental measure during which it had been produced. It depends on all the bits of a transmitted message, and also on a secret key, but which might be checked without knowledge of the secret key. a serious difference between handwritten and digital signatures is that a digital signature can't be a constant; it must be a function of the document that it signs. If this weren't the case then a signature may well be attached to any document. Furthermore, a signature must be a function of the whole document; changing even one bit should produce a special signature. A digital signature algorithm authenticates the integrity of the signed data and therefore the identity of the signatory [2]. A digital signature algorithm might also be utilized in proving to a 3rd party that data was actually signed by the generator of the signature. is meant to be used in pieces of email, electronic data interchange, software distribution, and other applications that need data integrity assurance and data origin authentication. The wireless protocols, like Hyper LAN and WAP, havespecified security layers and therefore the digital signature algorithm has been applied for authentication purposes.

2. Digital Signature

The term digital signature encompasses an excellent many types of “signatures”. Electronic signatures are simply an electronic confirmation of identity. This definition is knowingly broad enough to surround all kinds of electronic identification, from biometric signatures like iris scans and fingerprints to non-biometric signatures, like digitalsignatures.

Electronic signatures are often further subdivided into the highly secure and therefore the insecure. The digital signature must serve the uniform essential functions that we expect of documents signed by handwritten signatures, namely integrity, non-repudiation, authentication, and confidentiality. within

* *Corresponding author*
E-mail address: *roma110204018@gmail.com*

the digital realm, integrity means ensuring that communication has not been altered within the course of transmission. it's concerned with the accuracy and completeness of the communication. The recipient of transmission must be confident of a communication’s integrity before he/she will be able to depend on and act on the communication. Integrity is ticklish to eCommerce transactions, specifically where contracts are formed electronically. The process of digitally signing starts by taking a mathematical summary (called a hash code) of the check. This hash code could be a uniquely-identifying digital fingerprint of the check. If even a single little bit of the check changes, the hash code will dramatically change. the subsequent step in creating a digital signature is to sign the hash code together with your private key. This signed hash code is then appended to the check.

How is that this a signature? Well, the recipient of your check can verify the hash code sent by you, using your public key [3],[4]. At the identical time, a brand-new hash code is created from the received check and compared with the first signed hash code. If the hash codes resemble, then the recipient has verified that the check has not been changed. The recipient also knows that only you'll have sent the check because only you've got the private key that signed the first hash code.

2.1 Input to a Digital Signature

The message:

Since a digital signature has to offer data origin authentication (and non-repudiation) it's clear that the digital signature itself must be a bit of information that depends on the message, and can't be a totally separate identifier. It may be sent as a separate piece of information to the message, but its computation must involve the message. A secret parameter is thought only by the signer.

Since a digital signature must offer non-repudiation, its calculation must involve a secret parameter that's known only by the signer. the sole possible exception to the present rule is that if the opposite entity is completely trusted by all parties involved within the signing and verifying of digital signatures.

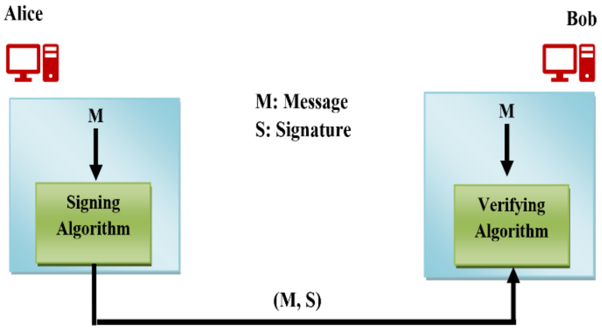

Figure 1: Digital Signature process

Digital signatures allow people to sign digital documents by providing the properties of a handwritten signature. They have to complete the five compelling attributes of handwritten signatures as listed by (Schneier, 1996) [5]. He stated that the handwritten signatures are authentic, unforgivable, not reusable, unalterable, and can't be repudiated. within the case of handwritten signatures, both the signature and therefore the document are physical things, which makes it difficult for the signer to say the signature isn't their own. so as to produce a secure electronic signature scheme, these attributes must be satisfied.

Electronic signature technologies contain PINs, user identifications and passwords, digital signatures, digitized signatures, and hardware and biometric tokens. Therefore, it's necessary to differentiate between electronic and digital signatures. Digital signatures are a subset of electronic signature technologies that make use of keys and cryptographic algorithms for signing documents.

Figure 1 illustrates the employment of PKI for generating digital signatures. the subsequent is an example of a digital signature scenario. Bob (sender) wants to send Alice (receiver) a text message with a digital signature. First, Bob reads the text message to be signed and generates a hashed message employing a message digest function (e.g., MD5, SHA1, etc.). A message-digest function may be a function that generates a 162-bit hash of the initial message; this hash can't be wont to regenerate the first message. Therefore, the hashed message is secure and unique. Once Bob has the hashed message, he uses the general public key digital signature algorithm and his private key to sign the hash to get a digital signature for the particular document.

Once Alice receives the digital signature, and therefore the corresponding text message, she's going to have to calculate two separate values. First, the hashed message of the received text is calculated using the identical hashing algorithm. Then, once she has the hash value, she will now use the decryption algorithm with Bob’s public key and digital signature to retrieve the signed hash. If she will be able to decrypt the digital signature, this suggests that Bob's private key was accustomed encrypt the hashed message. the ultimate step for Alice is to check the hash she calculated with the hash she retrieved from the decryption process. If these two hashed messages match, this suggests that she received the first message Bob signed (thus preserving message integrity). Key generation and distribution are the largest challenges in deploying PKI. the answer is to use a trusted central authority called a Certification Authority (CA) in PKI. CA could be a trusted entity that accepts certificate applications from entities, authenticates applications, issues certificates to users and devices in an exceedingly PKI, and maintains and provides status information about the certificates. If a CA is managing an outsized, geographically dispersed population, it's going to use Local Registration Authorities (LRAs), who provide direct physical contacts with subjects [6].

2.2 Properties of Digital Signature

To be valid, digital signatures require properties [7]:

Authenticity: A sound signature implies that the signer deliberately signed the associated message

Unforgeability: Only the signer can provide a valid signature for the associated message

Non-re-usability: The signature of a document can't be used on another document

Non-repudiation: The signer cannot deny having signed a document that incorporates a valid signature

Integrity: Make sure the contents haven't been modified

In this way, the digital signature replicates the desirable features of a handwritten signature and offers even stronger types of authentication if the correct procedures are followed within the handling of secret information. for instance, handwritten signatures may be verified only by experts practicing what can only be described as an inexact science. in contrast, the mathematical procedure for verifying digital signatures is verified by any number of independent agents, and there's no room for disagreement among these agents. The numbers are either correct or they're not.

There is now a minimum of two common methods of generating digital signatures. One may be a proprietary technology of RSA Data Security, Inc., and is named the RSA digital signature after its inventors Rivest, Shamir, and Adleman. This technology is being deployed in a very sizable amount of applications and has been licensed by nearly every big telecommunications and Computer Company within the U.S, including AT&T, Apple, IBM, Microsoft, Novell, and Sun Microsystems. the sole serious competitor to RSA signatures is that of the Digital Signature Algorithm (DSA), which was proposed by NIST as a Federal information science Standard called the Digital Signature Standard (DSS) [7]. there's some controversy surrounding the licensing of DSA since the holders of the RSA and other patents claim that DSA is roofed by their patents. NIST has declared that ``The Department of Commerce isn't conscious of any patents that might be infringed by this standard" [7]. within the next few years, now will become moot anyway, as many if not all of the relevant patents will expire by the year 2000 anyway.

RSA and DSA signatures use similar types of calculations, and these may be performed by hardware and/or software solutions. They rely on the notion of a ``Key Certification Authority" (CA) that's liable for issuing and/or certifying keys. the first role of a key certification authority is to assure that a user's public key's accurate. The CA needn't be online to answer questions about the legitimacy of keys. Instead, when a user is issued (or chooses and registers) their public/private key pair, the CA simply issues a digital signature of those keys to certify that these should be recognized as having been issued to the actual user. These credentials generally have expiration dates and should convey other information like a task that the user is certified. When a digital signature is generated by a user, the credentials for the keys accustomed create the digital signature could also be included with the digital signature (but this needn't be the case). A user may obtain multiple credentials for his or her public key, and there's no need for the user to get multiple public/private key pairs for multiple applications. particularly, there should be no need that the public/private key pair of a user can't be used for signing health documents yet as their email or their charge card transactions. For a user that accesses a system held by a given hospital, they'll get credentials for his or her public key from the hospital itself. To access the system within the role of a nurse, they'll want to present credentials for the identical keys that were issued by an accreditation agency for nurses [17], [18], [19], [20].

In addition to basic key certification through digital signatures under the certification authority's own public key, a key certification authority can provide:

- Construction and issuance of personal and public keys for users.
- Recertification of keys whose certificates have expired.
- Issuing key revocation lists for keysthat are compromised.

There is no reason to own secret user keys stored on the certification authority machine, and there are good reasons for them to not be stored there.

2.3 Public Key Infrastructure

A public key infrastructure (PKI) supports the distribution and identification of public encryption keys, enabling users and computers to both securely exchange data over networks just like the web and verify the identity of the alternative party [5].

Without PKI, sensitive information can still be encrypted (ensuring confidentiality) and exchanged, but there would be no assurance of the identity (authentication) of the alternative party. Any type of sensitive data exchanged online is reliant on PKI for security.

A typical PKI consists of hardware, software, policies, and standards to manage the creation, administration, distribution, and revocation of keys and digital certificates. Digital certificates are in the middle of PKI as they validate the identity of the certificate subject and bind that identity to the overall public key contained within the certificate.

A typical PKI includes the following key elements:

- A trusted party, called a certificate authority (CA), acts because the root of trust and provides services that authenticate the identity of people, computers, and other entities
- A registration authority, often called a subordinate CA, certified by a root CA to issue certificates for specific uses permitted by the basis
- A certificate database, which stores certificate requests and issues and revokes certificates
- A certificate store, which resides on a neighborhood computer as an area to store issued certificates and personal keys

A CA issues digital certificates to entities and individuals after verifying their identity. It signs these certificates using its private key; its public keys made available to any or all interested parties during a self-signed CA certificate. CAs use this trusted root certificate to make a "chain of trust" -- many root certificates are embedded in Web browsers so that they have the built-in trust of these CAs. Web servers, email clients, smartphones, and plenty of other forms of hardware and software also support PKI and contain trusted root certificates from the key CAs.

Including an entity's or individual's public key, digital certificates involve information about the algorithm accustomed create the signature, the person or entity identified, the digital signature of the CA that verified the subject data and issued the certificate, the aim of the final public key encryption, signature and certificate signing, similarly as a date range during which the certificate are going to be considered valid.Certifying Authority must be widely known and trusted and must have well-defined identification process before issuing the certificate. CA provides online access to all the certificates issued and provides online access to the list of certificates revoked. CA displays online the license issued by the Controller and displays online approved Certification Practice Statement (CPS). It must adhere to IT Act/Rules/Regulations and Guidelines

2.4 Public-Key Certification

The private key of CA or CCA requires the highest level of security. Here hardware Security Module (HSM) is used for storing the Private Key. More than one person is required for signing. HSM is housed in a strong room with video surveillance on a 24x7 basis [8].

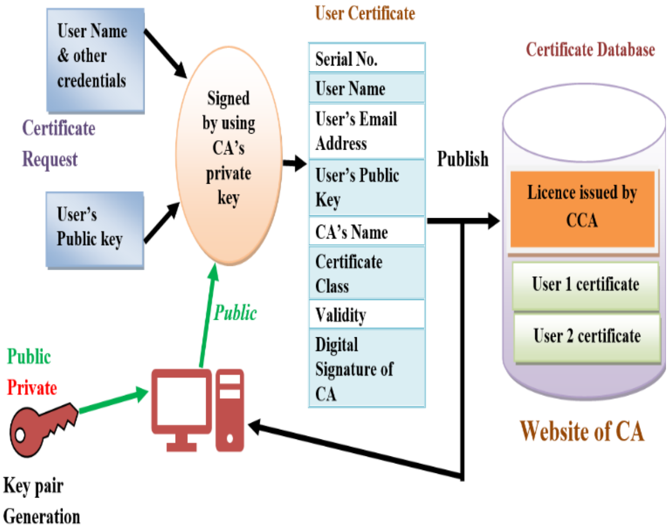

Figure 2: Certifying Authority

3. Challenges and Opportunities

Institutional overhead: The cost of establishing and utilizing certification authorities, repositories, and other important services, as well as assuring quality in the performance of their functions.

Subscriber and Relying Party Costs: A digital signer will require software, and will probably have to pay a certification authority some price to issue a certificate. Hardware to secure the subscriber's private key may also be advisable.

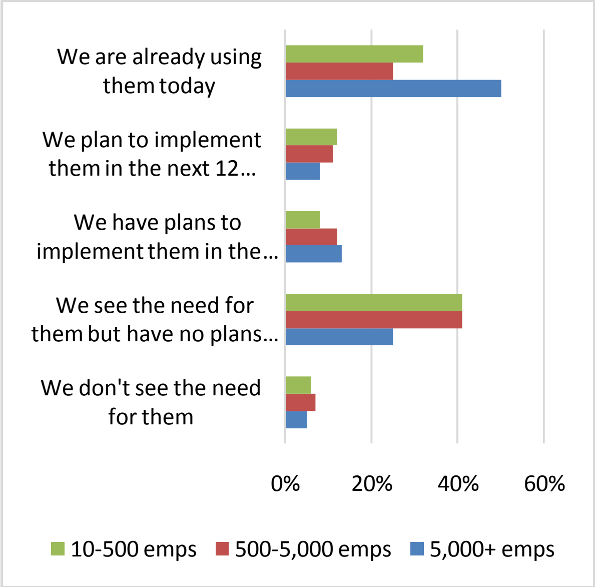

Figure 3: Comparison of adoption of digital signature

Adoption

- 35% of organizations have already automated their signature-dependent processes
- 35% of organizations who have answered the survey are already using digital/electronic signatures, up from 24% in the 2010 survey. A further 11% have plans in the next 12 months.
- Self-managed in-house digital signature (PKI) solutions are the most popular with existing users, but planned users are evenly split between this and server or appliance-based PKI systems. 30% are using non-PKI solutions.

4. Literature Survey on Digital Signature

This section discusses the research works conducted so far by various researchers to enforce E-Governance security using several types of digital signature schemes.

[1]. Paper title: Public Key Infrastructures and Digital Certificates for the Internet of Things [9].

Authors: Michael Schukat, Pablo Cortijo.

Description: Authors mainly analyze the benefits, limitations and suitability of both concepts for IoT deployments in combination with secure communication protocols. Based on this assessment it proposes an adopted PKI architecture that provides and manages customized X.509 digital certificates. Today's secure internet communication is provided by 3 principal components: (i) Network protocols (on data link, network or application layer) that provide secure and authenticated peer-to-peer communication, (ii) a verifiable digital identity for each peer and (iii) an infrastructure that allows for the generation, management and revocation of the latter.

They investigate how the above three concepts can be mapped onto IoT networks, considering that such networks will be predominantly based on IPv6 (as already seen in the 6LoWPAN wireless sensor network standard), the TCP/IP protocol stack and secure application layer protocols like TLS.

[2]. Paper title: A New Digital Signature Algorithm [10].

Authors: ErfanehNoorouzi, Amir Reza Estakhrian, Farzad Peyravi, Ahmad Khademzadeh.

Description: Propose a new algorithm that can be used in applications which have low file size for sending and want simple and fast algorithms for generating digital signature. A digital signature is a checksum which depends on the time period during which it was produced. It depends on all bits of a transmitted message and also on a secret key but which can be checked without knowledge of the secret key. A digital signature is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. Cryptographers have been studying electronic signature technologies for decades since the discovery of one-way functions. Several electronic signature schemes are (mathematically) proved to be secure under some complexity theoretical assumptions. In this paper author proposed a digital signature algorithm in which the new hash function generates dynamic and smaller size of bytes and a simple and fast generating digital signature which will append at the end of message.

[3]. Paper title: Comparison and Evaluation of Digital Signature Scheme Employed in NDN Network [11].

Authors: Al Imem Ali.

Description: There are numerous digital signatures algorithms used in NDN such as RSA and ECDSA characterized by their high level of security and their speed to encrypt and decrypt data according to, moreover their efficiency to generate signatures and verify the data integrity with reduced key sizes. Besides RSA and ECDSA, there is also MSS (Merkle signature scheme) defined by which is an interesting alternative for well-established signature schemes such as RSA, and ECDSA proved their eligibility against cyber-attacks e.g., timing attacks.

The main task of this paper is to find the optimal algorithm that avoids the system's overhead and offers the best time during the signature scheme.

[4]. Paper title: Digital Signatures, Certification Authorities: Certainty in the Allocation of Liability [12].

Authors: Yee Fen Lim.

Description: Electronic signatures are simply an electronic confirmation of authenticity. This definition is deliberately broad enough to encompass all forms of electronic identification, from the very informal (and insecure), such as initials at the end of an email, to the very formal (and highly secure), such as iris scans. Digital signatures are a particular subset of electronic signatures. Author focused on digital signatures and argue that certainty with respect to the liability of certification authorities is crucial and holds the key to the success of digital signature take up.

[5]. Paper title: SSL/TLS: What's Under the Hood [13].

Authors: Sally Vandeven, sallyvdv@gmail.com.

Description: Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are both protocols used for the encryption of network data. In this paper they explain how to capture a TLS session along with its encryption keys on a Linux system. They use RSA and DSA algorithm for key exchange process and Diffie-Hellman (DH) algorithm is used for key generation method. This paper has outlined how to capture, decrypt and analyze TLS sessions using the built-in capabilities of Wireshark. Wireshark is a feature-rich, free tool that captures and dissects network traffic. Based on experiments done for this paper, the browsers that currently will support export to a key log file for this type of operation are Firefox versions 19 -22 and Chrome versions 24 - 30. This type of analysis is useful for those who wish to understand the TLS protocol in greater depth. Web application developers may find these tools useful when troubleshooting an application that uses TLS. Penetration testers may find value in examining all the data transferred back and forth during an encrypted session with targets of their test without using a proxy.

[6]. Paper title: Practical Issues with TLS Client Certificate Authentication [14].

Authors: ArnisParsovs.

Description: They analyze Apache’s mod_ssl implementation on the server side and the most popular browsers – Mozilla Firefox, Google Chrome and Microsoft Internet Explorer on the client side. They complement their paper with a measurement study performed in Estonia where TLS client certificate authentication is widely used. They present their recommendations to improve the security and usability of TLS client certificate authentication.

[7]. Paper title: Digital Signature [15].

Authors: Ravneet Kaur, Amandeep Kaur.

Description: In this paper they describe privacy, authentication, integrity and non-repudiation, introduce encryption and illustrates steps involved in Digital Signatures. They discuss about various algorithm used in digital signature. DSA and RSA’s minimum key size is same but DSA has no licenses and patents, RSA is patented digital signature. Digital signature based on elliptic curve key technology uses smaller keys than other public key technologies but may be encumbered by various.

[8]. Paper title: Hash Based Digital Signature Schemes [16].

Author: Anders Fog Bunzel, Aarhus University.

Description: In this paper they analyze the four one-time signature schemas, the FMTseq signature scheme, the Winternitz signature scheme, the Bleichenbacher Mauer-Tree signature scheme and the Bleichenbacher-Mauer-Graph signature schema. The limitation of these one-time signature schemes is key management, i.e., they can only sign one message per key pair. But they presented a solution to this problem with the Merkle tree which authenticates multiple keys, but compared to RSA, this solution is not sufficiently efficient.

5. Background of Digital Signature

There are some algorithms that are used for digital signature generation under the DSS standard.

[1]. El-Gamal
[2]. RSA Digital Signature Algorithm
[3]. Digital Signature Algorithm
[4]. Elliptic Curve Digital Signature Algorithm
[5]. Elliptic Curve ElGamal Digital Signature Schema

5.1 El-Gamal Digital Signature Scheme

ElGamal digital signature is the asymmetric approach of authentication mechanism based on discrete logarithm problem. This technique uses β as the universally known random number that serves as the generator, $u$as the universally known prime number that serves as the modulus, *H()* as the universally known hash function [21], [22], [23].

At initial phase:

i. Bob selects static secret key S_{Bob}.

ii. Bob then compute the static public key P_{Bob} using S_{Bob}. [i.e.,$P_{Bob} = \beta^{SBob}$ mod u]

iii. Bob selects an ephemeral secret key R_i

iv. Bob then computes the ephemeral public key V_i[i.e.,$V_i = \beta^{Ri}$ mod u]

To sign a message msg_i, Bob performs the following-

v. Bob uses *H()* to compute hash of msg_i using V_i . [i.e., h_i= $H(msg_i \| V_i)$ where $h_i$is the hash of message msg_i]

vi. Bob now creates the El Gamal digital signature -

[$sign_i = R_i + h_i S_{Bob}$ mod (u-1)]

Once the signature is created, Bob sends P_{Bob},V_i , *msg* and $sign_i$to Alice. Alice receives P_{Bob}, V_i, msg’ and $sign_i$ and computes the following to verify the signature-

vii. Alice computes h_i’ (i.e., hash’ of the message) [i.e., h_i' = H(msg_i' || V_i)]

viii. After computing the hash' of the message, Alice finallychecks verifies if -

[i.e.,$sign_i$ mod *u*= V_i P h_i' mod *u*]

If the match is found, Alice then confirms the authenticity and integrity of the message to Bob.

5.2 RSA Digital Signature Algorithm

This technique uses the modulo arithmetic to sign a message digitally. Let Bob (sender) sends the message to Alice (receiver). This technique considers the public key of Bob and hash function *H( )* is universally known [22], [24].

At initial stage, Bob performs the following-

i. Selects two prime numbers, *U* and *V*

ii. Computes $N_{Bob} = U.V$

iii. Selects P_{Bob} such that P_{Bob} has no division (factors) in common with
[(U-1) .(V-1)]

iv. Calculates the secret key S_{Bob}such that –
$S_{Bob}P_{Bob}$ = 1 mod [(U-1) .(V-1)]
The public key set of Bob contains *N* and P_{Bob} , using which Bob creates the signature of the message.

v. Bob hashes the msg i.e. message
[h= $H(msg)$i.e.h is the hash of the message *msg*]

vi. Bob creates the digital signature -
[$sign = h^{SBob}$ mod N_{Bob} where *sign* is the signature]
Once the signature is created, Bob sends (*msg*, *sign*) to Alice.

vii. Alice uses the *H()* to obtain the *h'* (i.e.*hash'*) [$h' = H(msg')$]

viii. Alice decrypts the signature to retrieve its hash (i.e.*h*)
[$h = sign^{PBob}$ mod N_{bob}]

ix. Alice finally checks if: $h = h'$

If the match is found in the hash value retrieved and the hash value calculated, then Alice confirms the authenticity and integrity of the message along with the signature, else it is rejected.

5.3 Digital Signature Algorithm (DSA)

Digital signature algorithm is generated using various domain parameters like the private key *x*, per message secret key number *k*, data to be signed, and the hash function. Similarly, it is verified using various parameters like the public key *y* which is mathematically calculated from *x*, the data to be verified and the same hash function used during signature generation [18], [19]. Thus, the parameters used are as follows -

p – a prime modulus

q – a prime divisor of (*p*-1)

g – a generator of the sub group of order *q* mod *p*.

x - the private key is a randomly selected integer within the range [1, *q*-1]

y – the public-key obtained through $y = g^x$ mod *p*.

k – the per message secret key (i.e. unique to each message) obtained randomly within the range [1, *q*-1]

Let *N* be the bit length of *q*. Let *min* (*N*, *outlen*) denote the minimum of the positive integers *N* and *outlen*, where *outlen* is the bit length of the hash function output block. The signature of message *M* contains pair of numbers *r* and *s* obtained using -

$r = (g^k \bmod p) \bmod q$.

z = the leftmost *min*(*N*, *outlen*) bits of *Hash(M)*.

$s = (k^{-1} (z + xr)) \bmod q$.

Once the signature (*r*, *s*) is generated, Alice may transmit message *M*, and (*r*, *s*) to Bob. Let *M'*, *r'* and *s'* be the transmitted version of *M*, *r* and *s*.

To verify the signature Bob will perform the following steps -

i. Bob shall check that $0 < r' < q$ and $0 < s' < q$; if any one of the conditions is violated, the signature is rejected.

ii. If both the conditions in step-I. are satisfied, Bob computes $w= (s')^{-1} \bmod q$, where $(s')^{-1}$ is the multiplicative inverse of *s'* mod *q*

z = the leftmost *min*(*N*, *outlen*) bits of *Hash(M')*.

$u1 = (zw) \bmod q$.

$u2 = ((r')w) \bmod q$.

$v = (((g)^{u1} (y)^{u2}) \bmod p) \bmod q$.

iii. If $v = r'$, then the signature is accepted else rejected.

5.4 Elliptic Curve Digital Algorithm [ECDSA]

This is the elliptic curve cryptographic version of Digital Signature Algorithm i.e. ECDSA. This algorithm operates based on combination of three algorithms, key generation, signature generation and signature verification [24], [25].

Key generation -

The key pair of a user (say Alice) is associated with a specific set of EC domain parameters D= (q, FR, a, b, G, n, h), where -

E is an elliptic curve defined over Fq; P is a point of prime order n in $E(Fq)$; q is a prime; FR is the Field Representation which is an indication for representation used for the elements of Fq; a and b are the two field elements in Fq which define the equation of the elliptic curve E over Fq' ; two field elements x_G and y_G in Fq which define a finite point G=(x_G, y_G) of prime order in $E(Fq)$; the cofactor $h= \#E(Fq)/n$

To generate the key, Alice does the following-

i. Select a random integer d in the interval [1, n- 1].

ii. Compute $Q = dP$.

iii. Alice’s public key is Q and private key is d.

Signature generation -

To sign a message m, using domain parameters D = (q, FR, a, b, G, n, h) Alice does the following-

[1]. Select a random integer k in the interval [1, n-1].

[2]. Compute $kP = x_1, y_1$ and $r= x_1$ mod n (where x_1 is an integer between 0, q-1). If r= 0 then go back to step 1.

[3]. Compute k^{-1} mod n.

[4]. Compute $s= k^{-1} \{h\ (m)+ dr\}$ mod n, where h is the Secure Hash Algorithm (SHA-1). If s = 0, then go back to step 1.

[5]. The signature for the message m is the pair of integers (r,s).

Signature Verification:

To verify Alice’s signature (r, s) on m, Bob obtains an authenticated copy of Alice’s domain parameters D = (q, FR, a, b, G, n, h) and public key Q and computes -

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

[2]. Compute $w = s^{-1}$ mod n and $h\ (m)$

[3]. Compute $u_1 = h(m)w$ mod n and $u_2 = rw$ mod n.

[4]. Compute $u_1P + u_2Q = (x_0, y_0)$ and $v= x_0$ mod n.

[5]. If and only if $v = r$, then the signature is considered as valid else declared invalid by Bob.

5.5 Elliptic Curve ElGamal (EC ElGamal) Digital Signature Scheme

Elliptic Curve Cryptography can be combined with ElGamal Digital signature algorithm to generate EC ElGamal Digital Signature Scheme [26]. Entity A (Alice) selects a random integer k_A from the interval (1, n-1) as the private key and computes the public key, $A = k_A G$. Signing scheme:

i. Select random integer k from the interval (1, n-1)

ii. Compute $R= kG = (xR, yR)$ where $r = xR$ mod n; if r = 0 go to step i.

iii. Compute $e = h(M)$, where h is the hash function

$\{0,1\}^* \longrightarrow F_n$

iv. Compute $s = k^{-1} (e + rkA)$ mod n; if then go to step i. (R,s) is the signature of message M. Alice sends the signature and the message to Bob for verification.

Bob performs the following to verify the signature:

Verify that s is an integer in (1, n-1) and $R = (xR, yR)$ ε $E(Fq)$

i. Compute $V_1 = sR$

ii. Compute $V_2 = h\ (M)G + rA$, where $r = xR$

iii. If $V_1 = V_2$, then the signature is accepted by Bob, else declared as invalid.

5.6 RSA and ECDSA

RSA: The RSA concept is predicated on the factorization of huge numbers which implies the larger sequence of numbers you have got, the more you're protected

ECDSA: “The Elliptic Curve Digital Signature Algorithm (ECDSA) is that the elliptic curve analog of the Digital Signature Algorithm (DSA).

Table: Equivalent key lengths for RSA and ECDSA

RSA key length (bits)	ECDSA key length (bits)
1024	192
2048	256

5.6.1 Comparison of ECDSA with RSA:

ECDSA offers the identical level of security with smaller key sizes. the information size for RSA is smaller than ECDSA. The encrypted message could be a function of key size and data size for both RSA and ECDSA ECDSA key size iscomparatively smaller than RSA key size, thus encrypted message in ECDSA is smaller. Computational power is smaller for ECDSA. ECDSA provides faster computations and fewer space for storing. ECDSA key sizes are such a lot shorter than comparable RSA keys. The length of the general public and personal keys is far shorter in ECDSA.

Some researchers have found that ECDSA is quicker than RSA for the signing and decryption process, however, ECDSA could be a bit slower for signature verification and encryption.

5.6.2Advantages of ECDSA

It provides greater security with smaller key sizes. It provides effective and compact implementations for cryptographic operations requiring smaller chips. thanks to smaller chips less heat generation and fewer power consumption. it's mostly suitable for machines having low bandwidth, low computing power, less memory. it's easier hardware implementations.

5.6.3 RSA and ECDSA limitations

Key generation is incredibly slow. The speed of encrypting data is slow. Message length should be but the bit length otherwise algorithm will fail. RSA may be a factorization-based algorithm so on every occasion RSA initialization takes two large prime p and q.

6. Future Work and Conclusion

Supported this survey, there will be an attempt toimprove a replacement algorithm for digital signatureand a mechanism for adopting the digital signature for those organizations who haven't adopted digital or electronic signing solutions. Also, there will be aComparison the work with previous work. Hence, we can conclude that digital signatures minimize the risk of dealing with imposters and also minimizes the risk of undetected message tampering and forgery and retains a high degree of information security.

References

[1] Sur C, Roy A, Green ICT Culture and Corporate Social Responsibility, Proceedings of International Conference On Emerging Green Technologies (ICEGT 2011), July 27-30, 2011, pp-215-219, Organized by:PeriyarManiammai University, Tamil Nadu, INDIA.

[2] Sarkar S., Roy A., A Study on Biometric based Authentication, Proceedings of Second National Conference on Computing and Systems - 2012 (NaCCS - 2012) organized by the Department of Computer Science, The University of Burdwan, March 15 - 16, 2012, 1st Edition - 2012, Pp: 263-268, ISBN 978-93-80813-18-9.

[3] http://en.wikipedia.org/wiki/Key_(cryptography), Date of access – 24th March (2012).

[4] http://en.wikipedia.org/wiki/Key_generation, Date of access – 24th March, (2012).

[5] Tulu et al. Design and Implementation of a Digital Signature Solution, 10th Americas Conference on Information Systems, AMCIS 2004, New York, NY, USA, August 6-8, 2004

[6] https://www.csoonline.com/article/3400836/what-is-pki-and-how-it-secures-just-about-everything-online.html

[7] https://www.comparitech.com/blog/information-security/digital-signatures/

[8] https://www.sciencedirect.com/topics/computer-science/public-key-certificate

[9] M. Schukat and P. Cortijo, "Public key infrastructures and digital certificates for the Internet of things," *2015 26th Irish Signals and Systems Conference (ISSC)*, Carlow, 2015, pp. 1-5, doi: 10.1109/ISSC.2015.7163785.

[10] ErfanehNoorouzi, Amir Reza EstakhrianHaghighi, Farzad Peyravi, Ahmad Khademzadeh, A New Digital Signature Algorithm, 2009 International Conference on Machine Learning and Computing IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

[11] Al Imem Ali,Comparison and Evaluation of Digital Signature Schemes Employed in ndnNetwork, International Journal of Embedded systems and

Applications (IJESA) Vol.5, No.2, June 2015

[12] Yee Fen Lim, Digital Signatures, Certification Authorities: Certainty in the Allocation of Liability, Singapore Journal of International & Comparative Law (2003) 7 pp 183–200

[13] https://www.sans.org/readingroom/whitepapers/authentication/paper/34297

[14] ArnisParsovs, Practical Issues with TLS Client Certificate Authentication, January 2014, 10.14722/ndss.2014.23036, Network and Distributed System Security Symposium

[15] Ravneet Kaur, Amandeep Kaur, Digital Signature, :ICCS '12: Proceedings of the 2012 International Conference on Computing SciencesSeptember 2012 Pages 295–301

[16] Bunzel, Anders Fog. "Hash Based Digital Signature Schemes." (2015).

[17] http://en.wikipedia.org/wiki Digital_signature, Date of access – 24th March (2012).

[18] http://sajospsindia.com/back-issue.php?id=9, Date of access -24th March (2012).

[19] people.csail.mit.edu/rivest/pubs/GMR88.pdf, Date of access -24th March (2012).

[20] infoscience.epfl.ch/record/99523/files/Vau04b.pdf, Date of access -24th March, (2012).

[21] www.inf.ed.ac.uk/teaching/courses/cs/1112/lecs/signatures6up.pdf, Date of access -22nd May (2012).

[22] Cryptography and E-Commerce, A Wiley Tech Brief, Jon C. Graff, Wiley Computer Publishing, ISBN- 0471-40574-4.

[23] Karforma S., Mukhopadhyay S., Sen S.,AnObjectOrientedApproach of ElGamalDigital Signature Algorithm, ProceedingsofFirst International Conference onEmerging Applications of InformationTechnology (EAIT 2006), Science City,Kolkata, India, February 10-11, 2006, Pp-259-260 organized by Computer Societyof India Kolkata Chapter ISBN 10,81-312-0445-6.

[24] csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf, Date of access -24th March (2012).

[25] www.ijcaonline.org/volume2/number2/pxc387876.pdf, Date of access -22nd May, (2012).

[26] 198.170.104.138/itj/2005/299-306.pdf Date of access -24th March (201