ISSN No. 0976-5697

Volume 5, No. 3, March-April 2014

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

The Techniques behind the Electronic Signature based upon Cryptographic Algorithms

Avik Dey
Department of Computer Science and Engineering,
Adamas Institute of Technology,
Barasat, West Bengal, India

Subhasree Dey
Department of Computer Science and Engineering,
Adamas Institute of Technology,
Barasat, West Bengal, India

Rajib Ghosh*
Assistant Professor, Department of Computer Science and Engineering,
Adamas Institute of Technology,
Barasat, West Bengal, India

Abstract: Cryptography mainlydeals with keeping the information by converting it into anindecipherable formatwhich is called cipher text. The user, who retains a secret key, cangenerate the cipher text into plain text. Cryptography algorithms have many important parts; one of them is Electronic signature. The electronic signature suite is an assembly of various components i.e., algorithm of hash functions key generation, signing algorithms, verification algorithms and hash functions techniques.It mostly deals with the maintenance events for applying cryptographic hash functions like Sha-1, RIPEMD-160, WHIRLPOOL etc. and the signature algorithms/techniques like RSA, DSA,EC-GDSA etc.

Keywords: Cryptography, Electronic Signature Suite, Hash functions, DSA.

I. INTRODUCTION

In the current times, data loss is an important issue along with illegal data access; so, the data security is a major issue in today's world i.e. in public, private and defense organizations. To secure these information data or to protect this valuable information, we need a strong cryptographic technique along with digital signature concept for authentication.

There are two types of cryptographic techniques [1] [2]: symmetric and asymmetric. In symmetric cryptography, similar key is used for encryption and decryption; while in the other technique i.e. asymmetric cryptographic method, two different keys are used, one for encryption called public key and another for decryption called private key.

II. CRYPTOGRAPHY

Cryptography is basically one set of technique which provides information security. The way of hiding a message in such a way that no third party can get the original message, this is called encryption. Through encryption the original message or plain text transformed into cipher text and send to the receiver. The receiver receives the cipher text and using the technique the cipher text transformed into original message is called decryption. This total process as a whole called cryptography. To perform the encryption/decryption process it needs encryption/decryption keys [3]. The main goals of cryptography is, Confidentiality: Confidentiality is a service used to keep the content of information from all but those authorized to have it. Data Integrity: Data integrity is a service which addresses the unauthorized alteration of data.

Authentication: Authentication is a service related to identification. This function applies to both entities and information itself.

Non-Repudiation: Non-repudiation is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary [4]. A fundamental goal of cryptography is to adequately address these four areas in both theory and practice. Cryptography is about the prevention and detection of cheating and other malicious activities. The above picture gives a brief idea about the total family of cryptographic primitives.

These primitives should be evaluated with respect to various criteria such as:

Level of security: This is usually difficult to quantify. Typically the level of security is defined by an upper bound on the amount of work necessary to defeat the objective.

This is sometimes called the work factor.

Functionality: Functionality Primitives will need to be combined to meet various information security objectives.

Methods of operation: Methods of operation Primitives, when applied in various ways and with various inputs, will typically exhibit different characteristics; thus, one primitive could provide very different functionality depending on its mode of operation or usage.

Performance: This refers to the efficiency of a primitive in a particular mode of operation. Ease of implementation: This refers to the difficulty of realizing the primitive in a practical instantiation. This might include the complexity of implementing the primitive in either a software or hardware environment [4].

The modern field of cryptography can be divided into several areas of study. The chief ones are discussed here, Symmetric-key cryptography: Symmetric-key cryptography refers to encryption methods in which both the sender and

receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976 [5].

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. Block ciphers take as input a block of plaintext and a key, and output a block of cipher text of the same size. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards [3]. Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad. In a stream cipher, the output stream is created based on a hidden internal state which changes as the cipher operates. That internal state is initially set up using the secret key material. RC4 is a widely used stream cipher [4].

Public-key cryptosystems: In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. The public key is typically used for encryption, while the private or secret key is used for decryption. Diffie and Hellman showed that public-key cryptography was possible by presenting the Diffie-Hellman key exchange protocol [5]. In 1978, Ronald Rivest, Adi Shamir, and Len Adleman invented RSA, another public-key system.

III. ELECTRONIC SIGNATURE

A digital signature (standard electronic signature) takes the concept of traditional paper-based signing and turns it into an electronic “fingerprint”. This “fingerprint” or coded message is unique to both the document and the signer and binds both of them together. Digital signatures ensure the authenticity of the signer. Any changes made to the document after it has been signed invalidate the signature, thereby protecting against signature forgery and information tampering. As such, digital signatures help organizations sustain signer authenticity, accountability, data integrity and the non-repudiation of signed electronic documents and forms.

An electronic signature can be as basic as a typed name or a digitized image of a handwritten signature. Consequently, e-signatures are very problematic with regards to maintaining integrity and security, as nothing prevents one individual from typing another individual's name. Due to this reality, an electronic signature that does not incorporate additional measures of security (the way digital signatures do, as described above) is considered an insecure way of signing documentation.[8]
Main Working functionalities,

a. Identifies and authenticates a particular person as the source of the electronic message; and
b. Indicates such person's approval of the information contained in the electronic message.

Figure 1: Electronic Signature

IV. ELECTRONIC SIGNING PROCESS

Electronic Signature transpose in the electronic world, the rich semantics of handwritten signatures. As a signature is a symbolic representation of an individual, there is a strong intertwining between electronic signatures, authentication and identification. An e-signature as data in electronic form logically associated with other electronic data and which serve as a method of authentication. The intertwining is reinforced by the reliance of e-signature and e-identification on the same technology. There is a necessity of trust building instruments equally suitable for identification. Therefore, both are best addressed simultaneously. The identification systems are needed to be designed in such a way that the embedded signatures should also work across borders reaping the benefits of the cross border legal recognition of e-signatures [9]. The following figure describes the Electronic Signature process.

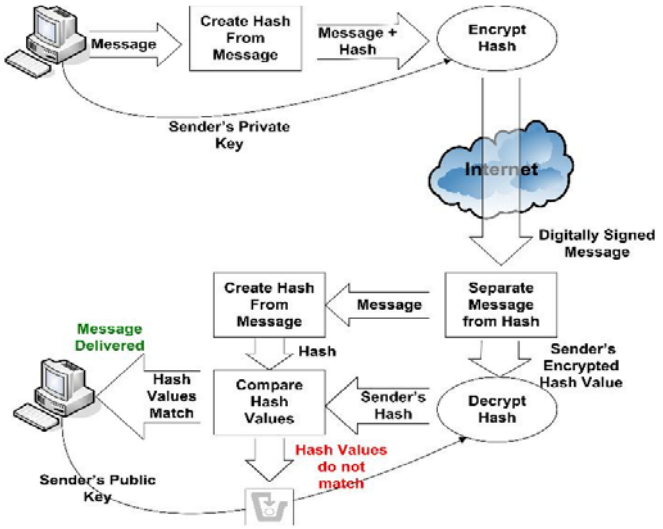

Figure 2: Electronics signing process

Electronic Signature suite is here defined as consisting of the following components (H, *K*, *S*, *V*):
H= Hash Function
K = Key generation algorithm
S = Digital Signing Algorithm with parameters and padding method
V = Verification algorithm

A. Hash Function:

A Hash Function takes a variable-length message as input and produces a fixed-length hash value as an output.Hash Functions may be used in a variety of circumstances, such as:

a. Advanced Electronic Signatures
b. Time- Stamp tokens

Public key certificates include the identifier of a signature suite which describes the hash function used to compute the digital signature. For the purpose of generating signatures the following (informally defined) three properties are required from the hash function h:

a) Pre-image resistance: Given y = h(m), it is practically infeasible to find m; without this stuff, a signature scheme may otherwise be vulnerable to an attack based on generating the signature "backwards", applying the verification function to a randomly chosen signature value.
b) 2nd pre-image resistance: Given h(m) and m, it is practically infeasible to find another m≠ m such that h(m) = h(m'). For signatures, this property protects from re-using an already existing signature for another message.
c) Collision resistance: It is practically infeasible to find any pair of distinct values m, m' such that h(m) = h(m').

This property is obviously needed to protect signature against chosen message attacks.

B. SHA-1:

SHA-1 may be used to hash a message, *M*, having a length of up to 264-1 bits. The main drawback is, several attacks against SHA-1 have been discovered. All known collision attacks on SHA-1 require full control of certain substrings within the data to be hashed and knowledge of the data bits prior to these strings. This is being considered as a realistic attack scenario for documents signed by signers (in particular, when a kind of "active" program element may be hidden in the document). On the other hand for X.509 certificates such attacks can be prevented by the CA by including a reasonable amount of entropy (i.e. data bits neither known to nor predictable by the attacker) in the certificate string prior to any data bits controllable by the attacker. This method leads to a considerably higher resistance of certificates against collision attacks.

C. Whirlpool:

WHIRLPOOL is a formed hash function which operates on messages less than 2256-1 bits in length, and produces a message digest of 512 bits. Whirlpool may be used to compute the imprint of a message placed in a time-stamp token. Whirlpool may only be used with a secure signature scheme supporting key sizes that match the Whirlpool output, i.e. 512 bits. DSA and ECDSA cannot be used with Whirlpool. However, it may be used with the RSA algorithm.

The Whirlpool output, i.e. 512 bits, is more than what may be needed, but there is currently no Whirlpool algorithm variant defined by an OID/URN with an output less than 512 bits, beside the general rule to take the leftmost bits of the output. Whirlpool has been included as an alternative to the SHA-2 family and can be used either to compute a hash value (for a time-stamp token) or with the RSA algorithm.

D. SHA-224& SHA-256:

SHA-224 may be used to hash a message, *M*, having a length of up to 264-1 bits and the output size is 224 bits. The function is defined in the exact same manner as SHA-256 except for the initial value and the truncation of the final hash value.

The specification for SHA-224 is identical to SHA-256, except that different initial values are used, and the final hash value is truncated to 224 bits. Therefore it is not recommended to use SHA-224, if SHA-256 can be used instead without truncation. The final result of SHA-256 is a 256-bit message digest.

V. SIGNATURE ALGORITHMS

A. RSA :

Ron Rivest, Adi Shamir and Leonard Adleman invented the RSA public key cryptography system. RSA has its security on factoring large numbers. In this system the private key and public are functions of a pair of large prime numbers. To generate the cipher text and to recover from plain text those large prime numbers factoring values are needed [6]. The following section gives a brief overview of the RSA algorithm for encrypting and decrypting messages.

Key generation:

The RSA algorithm,

Step 1: Randomly generate/choose two large prime numbers p and q of same size in bits.

Step 2: Compute the product n and Φ where n = pq and Φ = (p-1)(q-1).

Step 3: Randomly chooses an odd integer, e such that e < Φ and such that e and Φ are relatively prime(gcd (e, n) = 1).

Step 4: Using extended Euclidean algorithm the decryption key d has been generated. The formula of generating d is d = e-1 mod Φ. Now, the public key is the pair (e,n) and the private key is d. [7]

RSA Encryption: Sender wishes to send a message ('m') to receiver. To encrypt the message using the RSA encryption algorithm, sender must obtain receiver's public key pair (e,n). The message to send must now be encrypted using this pair (e,n). However, the message 'm' must be represented as an integer in the interval [0, (n-1)]. To encrypt it, Bob simply computes the number 'C' where Ci = mie mod n. Sender sends the cipher text C to receiver. [7]

RSA Decryption: To decrypt the cipher text C, receiver needs to use her own private key d and the modulus n. The decryption formula is, mi = Cid mod n which yields back the decrypted message (m). [7]

B. DSA:

The DSA algorithm's [10] security is based on the difficulty of computing the discrete logarithm in the multiplicative group of a prime field *Fp*. The public parameters *p*, *q* and *g* may be common to a group of users.

The bit length α of the prime modulus *p* shall be at least p MinLen bits long. The bit length β of *q*, which is a prime divisor of (*p*-1), shall be at least q MinLen bits long. Only the following choices for α and β are specified:

α = 1024, β = 160;
α = 2048, β = 224;

$\alpha = 2048, \beta = 256;$
$\alpha = 3072, \beta = 256.$

The value of β determines the defined in hash function to be used. This requires for β = 160 the function SHA-1, which should not be used for new applications.SHA-224 does not provide security advantages over SHA-256. If it is not required by a signature length restriction, since a signature with β =224 occupies 448 bits whereas a signature with β =256 needs 512 bits, it is recommended to use parameters with β =256.

The private key consists of:
(a). The public parameters *p*, *q* and *g*;
(b). A statistically unique and unpredictable integer *x*, 0<*x* <*q*, which is signatory-specific; and
(c). A statistically unique and unpredictable integer *k*, 0<*k* <*q*, which must be regenerated for each signature.

If the distribution of *k* is significantly different from uniform within the interval then there may be weaknesses. Bleichenbacher has presented an attack which can be sub-exhaustive depending on the size of the bias and the number of signatures produced using a single secret key.

The value of *k* must be kept secret as well as the private key, even if *k* is only partially known there exists an attack (Nguyen/Shparlinski).The public key consists of *p*, *q*, *g* and an integer *y* computed as $y = g^x \bmod p$. When computing a signature of a message M, no padding of the hash-code is necessary.

IV. ELLIPTIC CURVE ANALOGUE OF DSA BASED ON A GROUP E (FP)

This signature algorithm is referred to as ecdsa-Fp. The security of the ecdsa-Fp algorithm is based on the difficulty of computing the elliptic curve discrete logarithm [11].

The public parameters are as follows:
(a). *p* prime;
(b). *q* large prime at least qMinLen bits long, $p \neq q$;
(c). *E* elliptic curve over a finite field *Fp*whose order *n* is divisible by *q*; and
(d). *p* point on *E(Fp)* of order *q*.

The public parameters may be common to a group of users. The quotient *h* of the group order *n* divided by *q* may be considered as a public parameter too. The class number of the maximal order of the endomorphism ring of *E* shall be at least MinClass = 200.

The value *r*0: = min (*r*: *q* divides *pr*-1) shall be greater than r0Min=104.*h* = *n*/*q* must be less or equal 4.

The private key consists of:
(a). The public parameters *E*, *m*, *q* and *P*;
(b). A statistically unique and unpredictable integer *x*, 0<*x* <*q*, which is signatory-specific; and
(c). A statistically unique and unpredictable integer *k*, 0<*k* <*q*, which must be regenerated for each signature.

The public key consists of *E*, *q*, *P* and *Q*, a point of *E*, which is computed as *Q* = *xP*.

V. CONCLUSION

To encounter this security prerequisite and to permit signing of more or less random long messages, a signature suite requires a hash function, so that the signing and verification algorithms operate on a fixed-size hash of the message. Avital issue is to bond the hash function to the signature arrangement [10]; without this, the feeblest accessible hash function could express the complete security level.If any module of a set is modified, then the suite must be altered accordingly. A signature set involves the following modules:
i. A hash function;
ii. A signature technique
iii. A procedure to do the padding

VI. FUTURE SCOPE

In this paper the discussion revolves around the various hash functions and the signing algorithms. The procedures to generate the key can be studied. The main part is to develop the strategies for padding. The consequences of phase parameter and the message length can also be combined and verified for producingacompetent electronic signature set.

VII. REFERENCES

[1]. Douglas, R. Stinson, "Cryptography - Theory and Practice", CRC Press, 1995.

[2]. Menzes A. J., Paul, C., Van Dorschot, V., anstone, S. A., "Handbook of Applied Cryptography", CRS press 5th Printing; 2001.

[3]. Stallings W., "Cryptography and Network security", PHI, Third edition.

[4]. Menezes, A., van Oorschot, P., Vanstone, S., "Handbook of Applied Cryptography", CRC Press, 1996.

[5]. Diffie W., Hellman M., "New Directions in Cryptography", IEEE Transactions on Information Theory, vol. IT-22, Nov. 1976, pp: 644–654.

[6]. Bruice S., "*Applied Cryptography: protocols, Algorithms & source code in C*", Wiley-India Edition, 2007.

[7]. Cetin Kaya Koc, "*High speed RSA implementation*", RSA Laboratories, CA, November, 1994.

[8]. Applied Cryptography And Data Security, Prof. ChristofPaar (version 2.5) — January 2005

[9]. A Signature System Based on Trust Computing‖ by Chi Ya Ping , Li ZhiPeng ; Wei Zhan Zhen ; Fang Yong ,in International Conference on —Computational Intelligence and Software Engineering (CiSE), 10-12 Dec. 2010.

[10]. Integrating Diffie-Hellman key exchange into the digital signature algorithm (DSA)‖, by Harn, L.; Mehta, M.; Wen-Jung Hsin in Communications Letters, IEEE Volume:8, Issue:3 Digital Object Identifier: 10.1109/ LCOMM.2004. 825705 Publication Year: 2004, Page(s): 198 – 200.

[11]. "A Unified Approach to the Discrete Logarithm Problem for the Multiplicative Group and for Elliptic Curves over finite Fields". September 20, 2004.