ISSN: 2278-3075, Volume-2, Issue-4, March 2013

Secure Digital Signature Schemes Based on Hash Functions

Erfaneh Noroozi, Salwani Mohd Daud, Ali Sabouhi

Abstract— This paper provides a literature review and analysis of the security systems and the emphasis is on digital signature, hashed message algorithm. The proposed algorithm introduces a novel technique for producing small-sized output of digital signature as a result; the new scheme is potentially practical: signing and verifying signatures are reasonably fast, and both speed and time are improved.

Index Terms — Digital signature, Hashed message algorithm, Public key.

I. INTRODUCTION

Along with the thriving improvement of the technologies communication and information, systems of paper-based workflow is quickly substituted by the electronic-based medium in which all information and forms are digitally procedure such as e-government and e-commerce. In these systems, it is very significant to protect the sensitivity and security of digital object from malicious. Thus, how can this message are passed on so that only included or authentic parties obtain the comprehension of the message completely as it was transferred?

II. DIGITAL SIGNATURE

Diffie and Hellman put forth a solution to problem of authentication in their seminal paper entitled "New Direction in Cryptography" [1]. They first introduced the significant notion of public-key cryptography. The main idea of public key cryptosystem is to use two different keys; a public key for encryption and a private key for decryption, which are mathematically related. The two keys are such that computing the private key is infeasible from the public key.

Therefore, the main advantage of a public key cryptosystem is to allow users to communicate securely without exchanging secret keys. Diffie and Hellman did not come out with a concrete construction on how their concept can be implemented in practice. It was not until the work of Rivest, Shamir and Adlemen (RSA) [2].

That the first algorithm of a public-key cryptosystem was reac vnlized based on the difficulty of factoring of two large primes. Then, Tahar ElGamal (1985). Proposed his public key cryptosystem based on the difficulty of finding discrete logarithms.

Manuscript received on March, 2013.

Erfaneh Noroozi is a PhD candidate at Advanced Informatics School (AIS), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia.

Salwani Bt Mohd Daud is a lecture at Advanced Informatics School (AIS), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia.

Ali Sabouhi, Computer Science Kuala Lumpur, Malaysia.

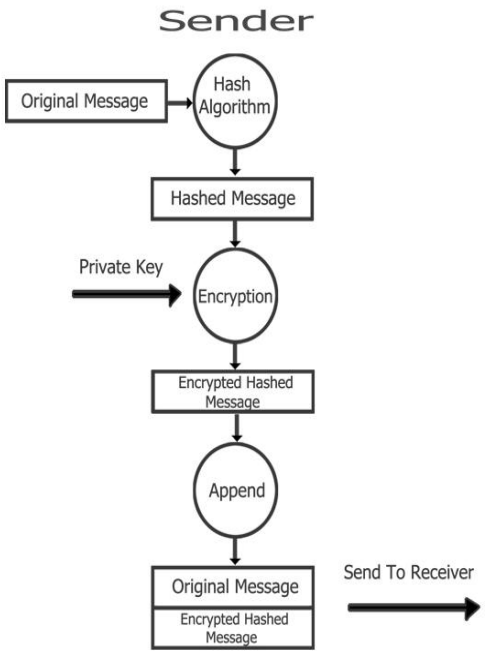

Figure 1: Process of digital signature for authentication

The public-key cryptosystem gives rise to a new and remarkable idea, which is the concept of digital signature. The digital signature is the electronic analog of the handwritten signature. A signer can digitally sign a document with his/her secret key (Private Key), and generates a signature on that document as shown in Figure 1.

Then, he/she sends the generated signature, a document and his\her public key to any verifier. Therefore, a verifier can check the validity of the signature with the corresponding public key (Figure 2). Note that, any involved party must register his public key with a central authority, which is known as the Certificate Authority (CA).

Therefore, this cryptosystem is known as a certificate-based public key cryptosystem (CB-PKC) [2]. The purpose of a digital signature is to achieve the same role of a handwritten signature in the electronic media. Handwritten signatures have the following properties: they must simple to create, easy to validate, and complex to fake [3]. Digital signatures can offer these properties but there are some differences between digital and handwritten signatures.

In a classic encryption system, when two parties (A is a sender and B is a recipient), they must agree on a particular private key so as to be used in encryption and decryption processes. 'A' can encrypt his message using the secret key and sends the corresponding cipher text to 'B'. The cipher text should be produced in such a way that any unintended receiver could not determine the original message. Then, 'B' can retrieve the original message by performing an inverse transformation of the cipher text using the same key. The triple algorithms (a key generation, encryption, and

decryption) are called a private key encryption system, which should be efficient and secure.

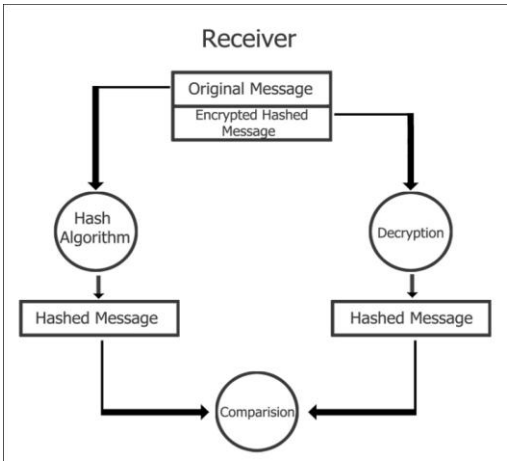

Figure 2: Verification of Digital Signature on the Receiver Side

III. COMPARISON OF HASH ALGORITHMS IN DIGITAL SIGNATURE

Cryptographic hash functions are used extensively due to its cheap construction. The function is further used for digital signatures [4]. In verification of the authentication of the data, the sender and the receiver compare the hash code and checks if it is genuine. The message is authentic when the message retrieved by the receiver is similar to the messages originally signed. Any changes to the data will affect the hash code which is sent with the data.

A. MD5 Algorithm

MD5 algorithm was introduced by Ray West in 1991 for calculation of hashed message. This algorithm receives the message with any length and divides it into a 512 piece. Except for the last piece, which must be 448 bits because of 64 bits added to the end of this algorithm, all pieces are in 512 bits [5]. If the length of message is not equal to this number, some useless elements will be added at the end of the last block. One bit of '1' and then '0' are added as much as possible till the total length reaches the 448 bits. In MD5, four variables of 32 bits are used to combine block of data and each time one ring of repetition of a 64 step disorder the data and a simple operation such as and, OR, NOT, XOR, is used so that the implementation of software with its hardware will be effective.

MD5 has been used in various security applications, [6] and it is also commonly used to check data integrity. In 2004, more serious flaws were found out in MD5. A group of researchers described how to create a pair of files that share the same MD5 checksum.

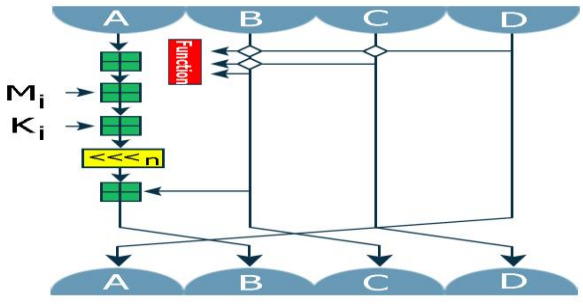

Figure 3: One Operation of MD5 Algorithm (Ronald Rivest, 1992)

The main algorithm then operates on each 512-bit message block in turn, each block modifying the state. The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. Figure 3 illustrates one operation within one round [7]. There are four possible functions F; a different one is used in each round:

$$F[X, Y, Z] = [X \wedge Y] \vee [\neg X \wedge Z]$$
$$G[X, Y, Z] = [X \wedge Z] \vee [Y \wedge \neg Z]$$
$$H[X, Y, Z] = X \oplus Y \oplus Z$$
$$I[X, Y, Z] = Y \oplus [X \vee \neg Z]$$

Denote the XOR, AND, OR and NOT operations respectively.

B. SHA-1 Algorithm (Secure Hash Standard)

SHA1, this algorithm was introduced by the standards and modern technology in the USA for public use.

This algorithm also organizes and processes the messages with any length in the format of 512 bits, the same as MD5 [8].

In SHA-1, the message expansion is defined as follows. The input is a 512-bit message, denoted by a row vector m. The message is also represented by 16 32-bit words, denoted by Figure 4 with t = 0, 1. . . 15.

In cryptography, SHA-1 is a cryptographic hash function which was designed by the United States National Security Agency and was published by the United States NIST (U.S) Federal Information Processing Standard [9]. SHA stands for "secure hash algorithm". The four SHA algorithms are structured in different ways and are distinguished as SHA-0, SHA-1, SHA-2, and SHA-3. SHA-1 is very similar to SHA-0, but it corrects an error in the original SHA hash specification that leads to significant weaknesses. The SHA-0 algorithm has not been adopted by many applications. On the other hand, SHA-2 differs significantly from the SHA-1 hash function. Among the existing SHA hash functions, SHA-1 is the most-widely used and is utilized in several commonly used applications and protocols. In 2005, security defects were identified in SHA-1. A mathematical weakness might exist which indicated that a stronger hash function would be more acceptable. Although no successful attacks on the SHA-2 variants have been reported yet, they are algorithmically similar to SHA-1 Figure 4 improved alternatives [10]. A new hash standard, SHA-3, was announced on October 2, 2012.

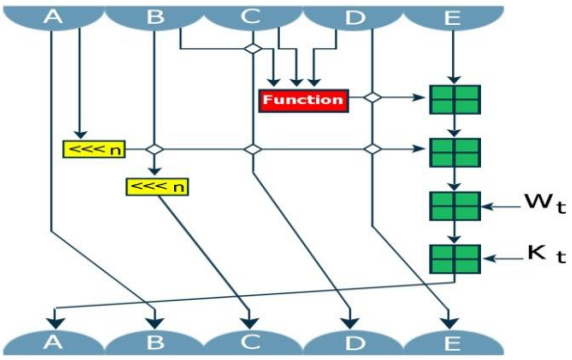

Figure 4: One Operation of SHA-1 Algorithm (National Security Agency, 1995)

C. SHA-2 Algorithm (Secure Hash Standard)

In cryptography, SHA-2 is a set of cryptographic hash functions (SHA-224, SHA-256, SHA-384, SHA-512) which was designed by the National Security Agency (NSA) and was published in 2001 by the NIST as a U.S. Federal Information Processing Standard [11]. SHA stands for Secure Hash Algorithm. SHA-2 includes an outstanding number of changes from its predecessor, SHA-1. SHA-2 consists of a set of four hash functions with digests that are 224, 256, 384 or 512 bits [12]. In 2005, security defects were identified in SHA-1 a mathematical weakness might exist which indicated that a stronger hash function would be more acceptable. Although SHA-2 is similar to the SHA-1 algorithm, these attacks have not been successfully extended to SHA-2. A new hash function, SHA-3, was selected by The NIST hash function competition in 2012. The SHA-3 algorithm is not derived from SHA-2.

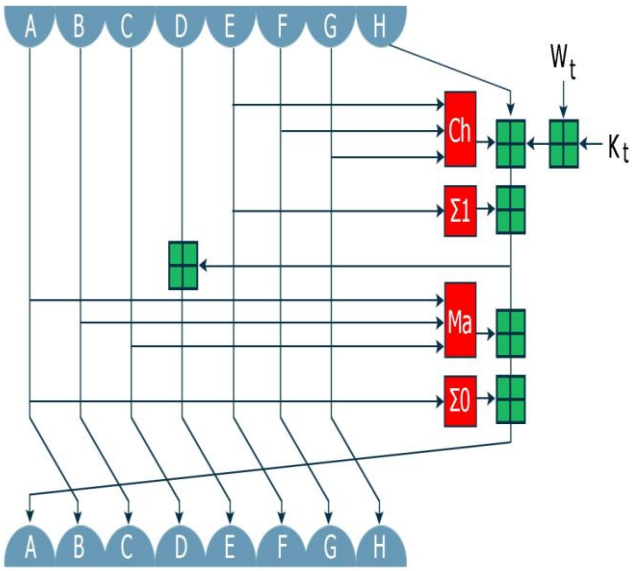

Figure 5: Iteration in a SHA-2 Family Function (National Security Agency, 2001)

$$\mathrm{Ch}[E, F, G] = [E \wedge F] \oplus [\neg E \wedge G]$$
$$\mathrm{Ma}[A, B, C] = [A \wedge B] \oplus [A \wedge C] \oplus [B \wedge C]$$
$$\sum_0 (A) = [A \ggg 2] \oplus [A \ggg 13] \oplus [A \ggg 22]$$
$$\sum_1 (E) = [E \ggg 6] \oplus [E \ggg 11] \oplus [A \ggg 25]$$

The bitwise rotation uses different constants for SHA-512. The given numbers are for SHA-256. The green ⊞ is an addition modulo 232 in Figure 5, SHA-1 and SHA-2 are the secure hash algorithms which are required by law to be used in certain U.S. Government applications [13], including the usage within other cryptographic algorithms and protocols in order to protect sensitive unclassified information. FIPS PUB 180-1 also favored adoption and use of SHA-1 by private and commercial organizations. SHA-1 is not used for most government applications.

D. Comparison of Hashed Algorithms

The size of the output MD5 algorithm is 32 bytes, SHA-1 is 40 and SHA-2 is 64 bytes Table III. In the first step of processing, useless elements should be added to its length will be a ratio of 512. This is done by adding both '1' bit and adequate '0' bits to the end of the message. Then, the actual length of the message in the format of 64 bits should be ORed in the last 62 bits in order that the length of the message is involved in the calculation of hashed message, Because of the output, these algorithms are constant and their time of complex thus equals O (n). SHA-1 is the improved version SHA-0, which was collided by Biham in 1993. It was expired the same as MD5 when the news of exploring the collision was officially announced in 2006 Table I.

TABLE I: COMPARISON OF HASH ALGORITHM

Algorithm	Methodology	Output	Time Complex	Performance
MD5	Divide to 512b, 64 times loop	32B	O(n)	Collision After 2006
SHA-1	Divide to 512b, 80 times loop	40B	O(n)	Collision After 2006
SHA-2	Divide to 512b, 64 times loop	64B	O(n)	Without collision

Hashed message, the so-called man-in-the-middle attack is important in defying digital signature. A remarkable feature of digital signature is that it cannot be altered once it is signed. Digital signatures are not similar to handwritten signatures as their constancy depends on the signed document.

A presentation of 64 bits of the first bits message is added at the end of the message. Now, the earning height of the message is exactly a multiplication of 512. Thus, it can be asserted that a 16-word multiplication represent the message. M [0...N −1] are the earning words in which each M[i] has 16 bits.

The computation of the message takes place in a 16-word-block. At first, four functions have been defined; each of which adopts words consisting of a triple of 32 bits as input and one 32-bit word as the output.

Then we compared the hashed algorithms in terms of its logical operators and the complexity of the hardware involved as shown in Table II.

TABLE II: COMPARISON OF LOGICAL OPERATIONS, CURRENT STATUS AND HARDWARE COMPLEXITY

Algorithm	Logical operations	Current status	Hardware complexity
MD5 algorithm [10]	AND,OR,NOT,Rotating shifts	Collision	Medium
SHA1 algorithm [5]	AND,OR,NOT,Rotating shifts ,XOR	Collision	Large-scale
SHA2 algorithm [12]	AND,OR,NOT,Rotating shifts,XOR	Running	Large

From Table II, the logical operations required for proposed algorithm are OR and XOR compared to other algorithms which required more than four (4) logical operations. The hardware complexity requirement is also lower compared to other algorithms. Hardware complexity contains devices such as Logic Devices, Programmable and Gate Arrays and Application Specific Integrated Circuits.

Then Table III compares the file size during transmission for these algorithms.

TABLE III: COMPARISON OF SIZE OF FILE IN BYTES

Size of original files (Byte)	MD5 algorithm (Byte)	SHA1 algorithm (Byte)	SHA2 algorithm (Byte)
14	32	40	64
18	32	40	64
72	32	40	64

(*SHA is Secure Hash Algorithm)([10], [5], [12])

It can be said that these algorithms have increased the file size from the original size, for original file of 18 bytes, MD5 increase to 32 bytes, SHA-1 to 40 bytes and SHA-2 to 64 bytes. File size capacity is any the drawbacks of these algorithms.

IV. HASHED MESSAGE V ENCRYPTED HASH MESSAGE

Hashed file is called as a sign file and stored as hashed file. During operation, the "signature version" emerges on the screen then the message of entering file name will be printed, and asks the user to enter the path of the file and also opens a file with Rb "only read binary". Then, hashing function is fetched followed by the encoder function which is then converted to hexadecimal, producing a unique code for each file. Whenever there will be a change in the first file, for example if one character will be omitted or the output hash of function, the output of sign function will be differed from the first file. The algorithm adopted is as follows:

```
int main (void)
{
int x,y; int mn; strnset (sign, passlen*2);
FILE *myfile;
char filename[80]; textmode(C80); clrscr();
printf("Signature Generator Version 1.00\n");
printf("Enter File Name : ");
scanf("%s",filename);
myfile = fopen(filename,"rb");
if (myfile==NULL) {
printf("\nCan not Open File %s !!!.\n",filename);
exit(0); } else {
hashing(myfile);
printf("\nHashing Was Compeleted ...\n"); encoder();
printf("Result Was : %s\n",sign); fclose(myfile);
getch(); getch(); return 0;};
```

To convert an array of character to hexadecimal function, and modify it to hexadecimal. Two variables lo, hi are required to implement the operation from the beginning till the end of encoded file.

hi = data [i] &240

lo = data [1] & 15

The algorithm for the conversion is shown as follows:

```
int converttohex(char*data,char*result)
  { int j=0; int i=0; int hi,lo;
    for  (;i<passlen;i++)
     { hi=data[i] & 240;
       lo=data[i] & 15;
       hi=hi>>4;
         if (hi>9)
              {hi=hi+'A"0'; };
         if (lo>9)
              {lo=lo+'A'-'9'-1; };
                   hi+='0';
                   lo+='0';
           result [j++]=hi;
           result [j++]=lo;
  }; return 0;};
```

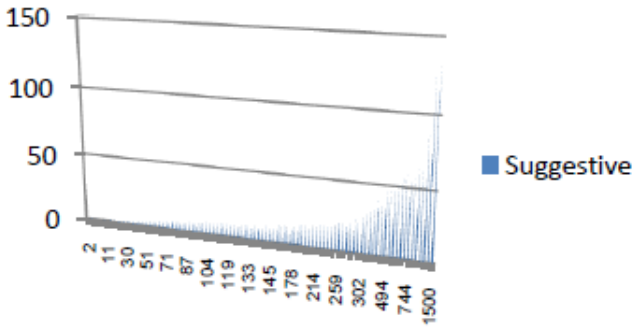

Figure 6: A Sample of 100 Hashed Files in Digital Signature

Figure 6 shows the size of original file versus size of hashed files. It illustrates the average of hashed size is 8.51% of the size of the original file.

V. CONCLUSION

Digital signatures are supposed to achieve some of the properties for hand signatures, e.g. (Validity and Verifiability). The bandwidth of a subliminal channel is defined as how many bits of covert message can be transmitted through such a channel in one session of protocol run. It measures the capacity of the subliminal channel in conveying hidden information. Testing new algorithms showed that its hashed file size is 4% reduction of the original file in messages with size lower than 1600 bytes.

In future work, proposed a method for the extension of the authentication system to image and extend digital signatures to sequences, which can also be used to enhance the robustness of signatures for still video.

REFERENCES

[1] W. Diffie, M. Hellman, New directions in cryptography." *Information Theory, IEEE Transactions on* 22.6, pp. 644-654, Jun 1976.

[2] R. Rivest, A. Shamir, L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21, no. 2, pp. 120-126, May 1978.

[3] E. Noroozi, D. Salwani, A. Sabouhi and M., SalehNamadi, "New Implementation of Hashing and Encoding in Digital Signature", *International Conference on Security Science and Technology – ICSST, Hong Kong*, March 2012.

[4] M. Bellare, and P. Rogaway ,"Entity authentication and key distribution". In *Advances in Cryptology — CRYPTO'93*, pp. 232-249, 1994.

[5] A. Sinha and K. Singh, "A technique for image encryption using digital signature." *Optics Communications* 218, no. 4, pp. 229-234, 2003.

[6] O. Mikle, "Practical attacks on digital signatures using MD5 message digest.", *Cryprology ePrint Archive, report* 356, 2004.

[7] Q. Sun and S. F. Chang, "A secure and robust digital signature scheme for JPEG2000 image authentication." *Multimedia, IEEE Transactions* on 7(3), pp. 480-494, 2005.

[8] R. Cramer and V. Shoup, "Signature schemes based on the strong RSA assumption." *ACM Transactions on Information and System Security (TISSEC)* 3(3), pp.161-185, 2000.

[9] A. Noore, "A secure conditional access system using digital signature and encryption". *Consumer Electronics, ICCE. 2003 IEEE International Conference on, IEEE*, 2003.

[10] J.Stern, D. Pointcheval, "Flaws in applying proof methodologies to signature schemes." *Advances in CRYPTO*, pp. 215-224,(2002)

[11] M. Aydos, T. Yantk, "A high-speed ECC-based wireless authentication on an ARM microprocessor". *Computer Security Applications, ACSAC'00. 16th Annual Conference, IEEE*. (2000).

[12] M. Shah, A. R. Swaminathan, "Privacy-preserving audit and extraction of digital contents." *Cryptology ePrintArchive, Report 186*, 2008.

[13] J. Ding, B. Y. Yang, "New differential-algebraic attacks and reparametrization of rainbow". *Applied Cryptography and Network Security, Springer*, (2008).

Erfaneh Noroozi: She completed her Computer Engineering first degree in University of Shiraz, Iran 2001 and attached to University of Najafabad, Iran-Esfahan 2005 got Bachelor of Computer. In 2007 she had obtained Master. Eng of Computer engineering−software in Iran South Tehran, She had started her career with Faculty member of Computer Engineering in 2006, and her teaching experience in Azad University of Sepidan branch. She starts her PhD. (Computer Eng.) in UTM International Campus 2011, Kuala Lumpur. Her interest in this profession motivates her to work hard in the teaching, research, publications. Her main expertise is in software engineering system, Information Security(cryptography, digital signature, steganography) as well as engineering education. Currently she is PhD student in Advanced Informatics School (AIS), UTM Kuala Lumpur.

Salwani Mohd Daud: She completed her Electronics Engineering first degree in University of Liverpool, United Kingdom1984 and attached to Universiti Teknologi Malaysia (UTM) on 1st November 1985. She had started her career with Faculty of Electrical Engineering (FKE) in 1985, and her teaching experience had exceeded 25 years in UTM. After FKE transferred to the main campus (Johor Bahru) in 1995, she was placed in Program of Diploma Studies that was renamed as College of Science and Technology in UTM International Campus, Kuala Lumpur. She had obtained her Master of Electrical Eng. and PhD.(Electrical Eng.) from UTM in 1989 and 2006 respectively. Her interest in this profession motivates her to work hard in the teaching, research, publications and consultancy. Her main expertise is in intelligent system, biomedical engineering system and computer system engineering. Her other interests include multimedia content development, protection on multimedia content (cryptography, watermarking, steganography) as well as engineering education. Currently she is an Associate Professor in Advanced Informatics School (AIS), UTM Kuala Lumpur.

Sabouhi. Ali: Date of birth: 12 August 1975 Place of birth: Tehran Iran. Major: Computer engineering−software Degree: M. Eng. from Kuala Lumpur, Malaysia. He Concrete design and analyze and implementation brothers manufacturing plant management for 14 years and also Construction firm manager In Tehran, Imam Ali highway project workshop. His main expertise is in software engineering system, design and implementation system and Information Security such as digital signature and video steganography as well as engineering education.