An Efficient Signature System Using Optimized RSA Algorithm

RANIA SALAH EL-SAYED *,Prof. MOUSTAFA ABD EL-AZIEM, Dr . MOHAMMAD ALI GOMAA *****
* Academy of Scientific Research & Technology- Cairo. Egypt

**Arab Academy For Science& Technology And Maritime Transport- Cairo. Egypt

*** Department of computer science- Faculty Of Science- Al-Azhar University- Cairo. Egypt

Summary
Digital signature are used in message transmission to verify the identity of the sender and ensure that a message has not been modified after signing. RSA algorithm is extensively used in the popular implementations of Public Key Infrastructures. In this paper, we have develop a new algorithm for generating signature that overcomes the shortcomings of the RSA system (longer processing time & computational overheads).In addition, the new algorithm achieve high security for digital signature. The performance of the two public key cryptosystem (RSA and DSS) and the new algorithm has been implemented and compared. The results obtained show that signing and verification operations are faster in the case of using new algorithm than in the case of RSA and DSS . Also it can forbid any one from reaching the sender's message because with the new algorithm an intruder cannot pose the message sent since the sender's private key is unknown for him. Accordingly, the sender can not be impersonated. On the receiver part, the message is verified by using sender's public key and his private key to decrypt the message successfully.

Key words:
Cryptography, RSA, DSS, PKC and DSA.

1. Introduction

Data communication is an important aspect of our living. So, protection of data from misuse is essential. A cryptosystem defines a pair of data transformations called encryption and decryption. Encryption is applied to the plain text i.e. the data to be communicated to produce cipher text (encrypted data) using encryption key. Decryption uses the decryption key to convert cipher text to plain text (the original data). Now, if the encryption key and the decryption key is the same or onecan be derived from the other then it is said to be **symmetric cryptography**. This type of cryptosystem can be easily broken if the key used to encrypt or decrypt can be found. To improve the protection mechanism **Public Key Cryptosystem** was introduced in 1976 by Whitfield Diffe and Martin Hellman of Stanford University [12]. It uses a pair of related keys one for encryption and other for decryption. One key, which is called the private key, is kept secret and other one known as public key is disclosed [10].

A digital signature, as shown in fig 1, takes the concept of traditional paper-based signing and turns it into a digital "fingerprint". Digital signature software enables you to easily migrate from cumbersome paper-based processes to a secure and efficient paper-free environment [5].

Fig.1 Signature on the document

This digital "fingerprint", or coded message, is unique to both the signer and the document. This ensures that the person who signed is indeed the originator of the message. This fingerprint cannot be reused or reassigned to anyone else at any time.

If any changes were made to the document after it was signed, they would automatically invalidate the signature, thereby protecting against forgery [2].

Digital signature software helps organizations sustain signer authenticity, accountability, data integrity, and non-repudiation of documents and transactions as shown fig.2.

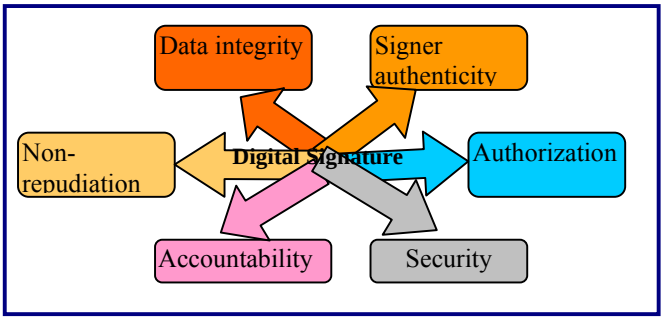
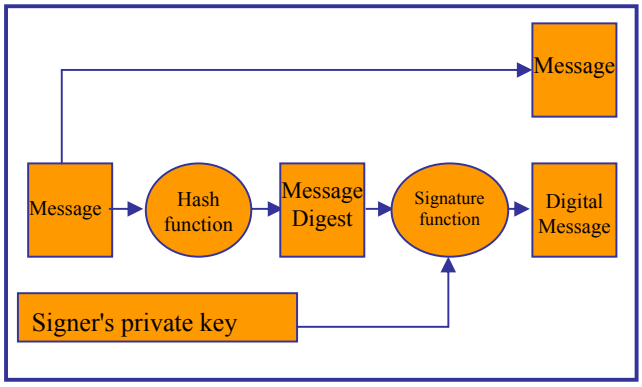


Fig.2 Security Advantages of Digital Signatures

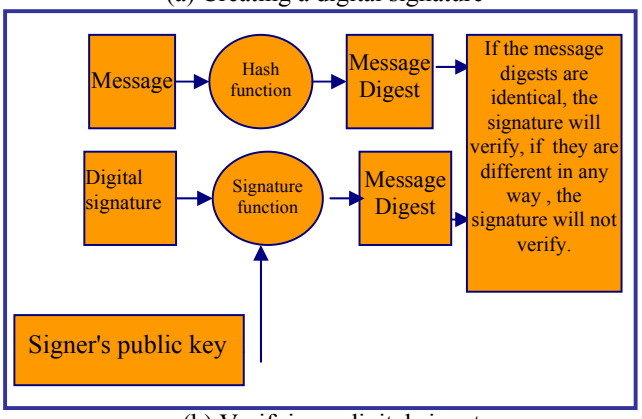
Manuscript received December 5, 2008
Manuscript revised December 20, 2008

Digital signature is most secure and widely used category of signatures. It relies on public key cryptography (PKC). Many PKC schemes are used for digital signatures.

Basic idea of digital signatures is each signer has a unique key called private key. There is also other part of key called public key. Whenever singer has to authenticate a document it creates a bit string called signature by applying his private key on the message or some hashed image of message as shown in fig.3a. User who receives this message then applies his public key on the signature and checks the validity of the bit-string as shown in fig.3b. If receiver is convinced that document is signed by legitimate signer, it accepts the document. Later if there is some dispute between sender and receiver regarding the validity of document, a third party inspects the signature and using the public key of signer verifies the signature[13,2].



(a) Creating a digital signature



(b) Verifying a digital signature

Fig.3 Generalized signatures Generation and Verification

Digital signatures are mainly divided into two categories [2]:

1. Digital signatures with message recovery: where a redundancy function (hash function) is applied to the message and then singed. The message can be recovered back from the signature itself by applying inverse of redundancy function [7]. e. g. **RSA** (Rivest Shamir Adelman).

In cryptology, RSA is an algorithm for public-key encryption Based on integer factorization system. It was the first known algorithm suitable for signing as well as encryption, and is considered a one of the first great advances in public key cryptography. RSA is still widely used in electronic commerce protocols, and is believed to be secure if and only if long keys strategy is kept [8,1] .

2. Digital signatures with appendix: this scheme require the original message as input to verification algorithm.. e. g. **DSS** (Digital Signature Standard).

DSS based on discrete logarithm system [8, 11,9] is quite different from the RSA approach. In effect, it uses the sender's private and public key much like RSA does, but it also generates another private and public key per message so that each message is also signed with a second private key that changes with each message [6].

DSS defines the Digital Signature Algorithm (DSA), which functions in a manner similar to RSA. Although similar to RSA, DSA does not encrypt message digests with the private key or decrypt the message digest with the public key. Instead, DSA uses special mathematical functions to generate a digital signature composed of two 160-bit numbers that are derived from the message digest and the private key. DSA uses the public key to verify the signature, but the verification process is more complex than RSA.

The digital signature processes for DSA and RSA are generally considered to be of equal strength. However, DSA requires the use of the SHA-1 message digest function to ensure strong digital signatures. RSA can be used with other message digest functions (besides SHA-1) that might produce weaker digital signatures. Because the DSA signature verification process increases computer processor load significantly, relative to the verification process for RSA (all other conditions being equal), the RSA digital signature process generally provides better overall performance.

2. Problem Statement

RSA is a highly secure algorithm .The only known way to attack it is to perform a "brute-force" attack on the modulus. This attack can be easily defeated by simply increasing the key size [10]. However, this approach can lead to a number of problems:

- Increased processing time – as a rough guide , decryption time increases 8-fold as key sizes double .
- Computational Overheads – the computation required to perform the public key and private key transformations.
- Increased key storage requirement – RSA key storage (private keys and public key) requires significant amounts of memory for storage.

Furthermore , Key generation is complex and time consuming (times increase significantly as key sizes increase). In each of the systems (RSA – DSS) considerable computational savings can be made. In RSA, a short public exponent can be employed (although this does incur some security risks) to speed up signature verification and encryption. In DSS a large proportion of the signature generation and encrypting transformations can be pre-computed. Memory constrained devices cannot easily generate RSA keys and so may need to have keys generated by another system. However, this means that the non-repudiation service may not be achievable [3].

So, we need to make digital signature more secure with small bits (1024 bits) for keys generation .

3. Proposed Algorithm

The proposed digital signature algorithm is an adaptation of the RSA algorithm that overcomes the shortcomings of the RSA system (processing time & computational overheads). The new algorithm can solve the problem of processing time by not increased the key size but using key with small bit (1024 bit) so the problem of Increased processing time can be solved. And also it can solve the problem of computation overheads by making modified in RSA algorithm. It essentially involves the RSA system with DSA so that it can forbid any one from reaching the sender's message. The basic aim from this algorithm is to make RSA algorithm more efficient.

In this algorithm the message to be signed is input to a one-way hash algorithm producing an unrecoverable digest of the message. The digest is encrypted with receiver's public key then sender's private key to produce signature, appended to the original message and transmitted. In fig.4, the proposed algorithm is shown. It is clear that the transmitted message is not encrypted because the signature function is sufficient for securing the transmission process.

At the receiver, the message and signed digest (the signature) are separated. The original message is passed through the same hash function used by the originator and the signature is decrypted using sender's public key then receiver's private key to produce another copy of the original digest. The two digests are presented to a comparator. If they are equal, the message is accepted as genuine. If they do not match, the message is rejected, as shown in fig. 4.

At the receiver, the message and signed digest (the signature) are separated. The original message is passed through the same hash function
used by the originator and the signature is decrypted using sender's public key then receiver's private key to produce another copy of the original digest. The two digests are presented to a comparator. If they are equal, the message is accepted as genuine. If they do not match, the message is rejected, as shown in fig. 4.

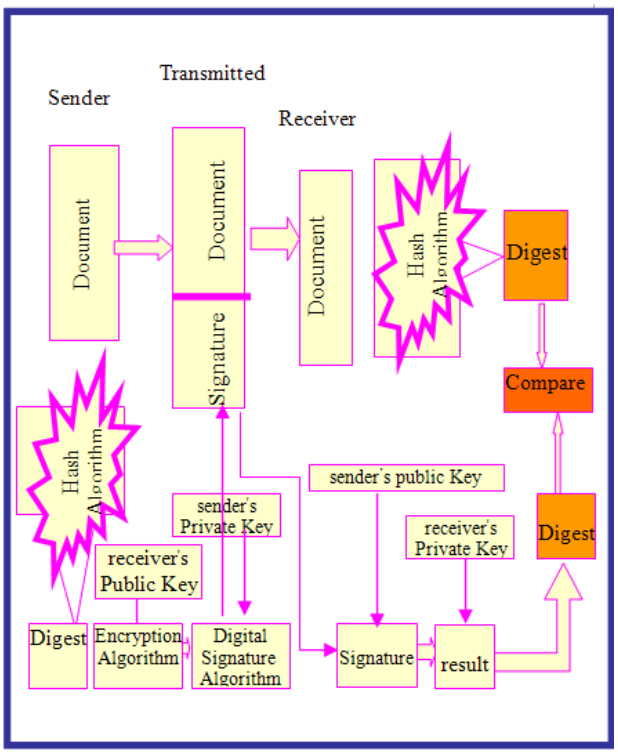


Fig.4 Proposed signature

3.1 Key Generation

Suppose a user A wishes to allow B sends a private message over an insecure transmission medium. A & B take the following algorithm to generate a public key and a private key.

INPUT : Bit length of modulus, k.
OUTPUT: Public key (E; N), and private key (D; N).

1) Generate prime numbers (P_a)and (P_b)of bit length [k/2]

2) Generate prime numbers (Q_a) and (Q_b)of bit length k – [k/2]

3) Compute N ← P. Q for both a & b

4) Select E to be an integer, where GCD (e; (P - 1) . (Q - 1)) = 1

5) Compute D such that E . D ≡ 1 (mod (P - 1) . (Q - 1))

6) Return ((E_a; N_a); (D_a ;N_a) for A And ((E_b; N_b); (D_b ;N_b)) for B

3.2 Signature Generation

INPUT : Private Key (D_a ; N_a) for the sender, Public key (E_b ; N_b) for the receiver , and message to be signed, M.
OUTPUT: s, signature of M

1) A ← $h(M)^{E_b} \pmod{N_b}$

2) S ← $A^{D_a} \bmod N_a$.

3) Return (s)

A common hash algorithm used is SHA-1

3.3 Signature Verification

INPUT : Private Key (D_b; N_b) for the receiver, Public key (E_a ;N_a)for
the sender , message(M) and signature (S).
OUTPUT: VALID or INVALID.

1) B ← $S^{Ea} \bmod N_a$

2) Q ← $B^{Db} \bmod N_b$

3) If Q = h(M), Return VALID
Else, Return INVALID

4. Experimental Results

To test and compare the performance characteristics of the RSA, DSS, and proposed signature algorithms, we developed three programs using C++ language to implement them [4]. Then we test each of the three main components (key generation, signature generation and signature verification) in each program independently.

Three experiments has been done to test and compare the time required for achieving the implementation of RSA, DSS, and proposed algorithms in small range of key size. Tests are preformed on an Intel P4 3.06 GHz machine with 512MB of RAM. The experiment results are tabulated in second as shown in Table1.

Table1 results of creating signature

Algorithm	Key Generation (Second)	Signature (Second)	Verification (Second)
Experiment 1			
RSA	4.485000	0.016000	13.594000
DSS	34.328000	8.110000	4.531000
Proposed	10.719000	0.015000	0.015000
Experiment 2			
RSA	7.735000	0.016000	13.281000
DSS	72.922000	30.906000	8.312000
Proposed	10.750000	0.031000	0.047000
Experiment 3			
RSA	12.324000	0.016000	19.625000
DSS	50.640000	35.344000	10.632000
Proposed	11.45600	0.040000	0.0500000

4.1 Key Generation

The time required for key generation of RSA is smaller than DSS and the proposed algorithm. The time required for DSS is 8-fold that of RSA since in DSS key generation algorithm we need to generate key for user as well as key for each message the sender can sent it. But the time of proposed algorithm is double that of RSA because two keys for sender and receiver, are needed to be generated. Fig 5 shows the results obtained.

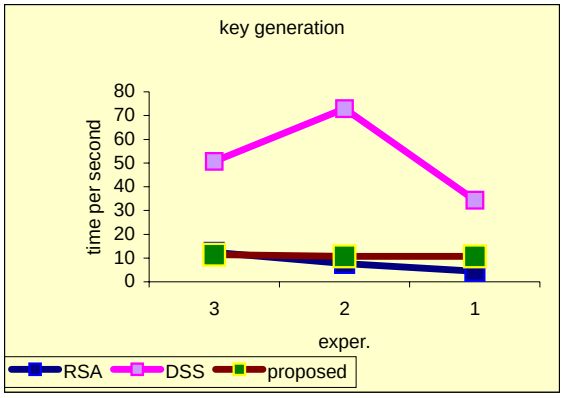


Fig.5 Comparison of key generation

4.2 Signature Generation

The time required for signature generation of RSA is smaller than DSS because hash function is used in RSA signature algorithm while secure hash function is used in DSS. Moreover, time required for the proposed algorithm is smaller than RSA since it develops the RSA signature algorithm by using it with DSS .The results are shown in fig 6.

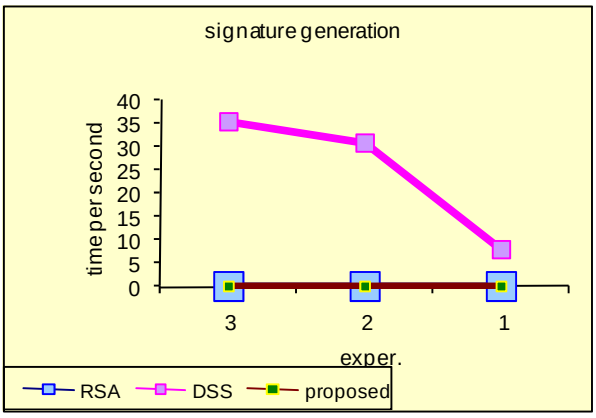


Fig.6 Comparison of signature generation

4.3 Signature Verification

In signature verification process, the proposed algorithm pulls ahead both RSA and DSS in performance. Because, in proposed algorithm, two keys are used, private for receiver and the public key for sender, while in RSA, only public key is used. One important consideration of the signature verification process is that part of each algorithm time is spent computing the SHA-1 hash of the massage, as shown in fig 7.

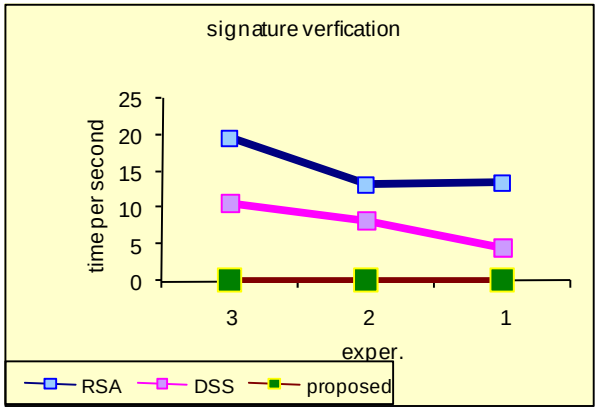


Fig.7 Comparison of signature verification

5. Conclusion

The results obtained show that RSA signature generation is significantly slower than the developed signature generation algorithm. The cost of signature generation can be considered as a factor in the choice of signature systems. Thus, the proposed signature cost is lower than RSA signature.

With the proposed algorithm achieve high security for digital signature in addition to decrease processing time and computational overheads. And an intruder cannot pose the message sent since the sender's private key is unknown for him. Accordingly, the sender can not be impersonated. On the receiver part, the message is verified by using sender's public key and his private key to decrypt the message successfully.

General speaking, in RSA, signature generation is faster than signature verification, and in DSS, signature verification is faster than signature generation. The proposed algorithm is overall faster than both RSA and DSS.

References

[1] A.Hosseinzadeh Namin," Elliptic Curve Cryptography " , university of Windsor, April 2005

[2] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1997

[3] Burt Kaliski , "The Mathematics of the RSA Public-Key Cryptosystem" , RSA Laboratories. April 9,2006

[4] Bruce Eckel,President & MindView Inc, " Thinking in C++", Volume 1&2, 2nd Edition, Completed January 13, 2000

[5] David Youd , "An introduction to Digital Signatures ",published1996

[6] [FIPS186] "Digital Signature Standard (DSS), " Federal Information Processing Standards Publication

186, " U. S. Department of Commerce, National Institute of Standards and Technology, 27 January 2000

[7] Nyberg K. and Rueppel R.A. (1994). A new signature scheme based on the DLP giving message recovery, Advances in Cryptology – Eurocrypt – 94,Springer and Verlag, p.p. 182 -193.

[8] Patrick J. Flinn and James M. Jordan , " Using the RSA Algorithm for Encryption and Digital Signatures", July 9, 1997. (http://www.cyberlaw.com/cylw_home.html).

[9] Robert E. Mahan," Digital Signatures and Authentication Protocols" , copyright 1998, 1999.

[10] R.L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, volume 21, pages 120-126, February1978.

[11] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31(4):469–472, July 1985.

[12] W.Diffie and M. Hellman." New Directions in Cryptography". IEEE transactions on Information Theory. IT-22(1978).472-492.

[13] William Stallings, "Cryptography and Network Security Principles and practices" Fourth Edition. November26,2005