COMPARISON STUDY ON NP-HARD PROBLEM BASED DIGITAL SIGNATURE SCHEMES

Mohammad Ahmad Alia
dr.m.alia@zuj.edu.jo
Computer Information Systems Department, Faculty of Sciences and IT, Al-Zaytoonah University of Jordan, P.O.BOX 130, Amman 11733– Jordan

***Abstract*—**This paper shows the development in public-key digital signature schemes which are actually based on non deterministic polynomial mathematical hard problems (NP-Hard). In general, most of the currently used digital signature cryptosystems are computationally expensive with relatively lengthy key requirement due to the dependency on the number theory. Therefore, it's important to study the performance of the most used digital signature schemes which are based on different mathematical hard problems that are, in some sense, difficult to solve. In the surveyed schemes, we present the powerful and practical of some public-key schemes depending on its security level and execution time.

***Keywords*—**Cryptography, Digital Signature, Hard Problem, and Public-key.

I. INTRODUCTION

Digital signature is a verification mechanism based on the public-key scheme (refer to Figures 1 and 2) that is focused on message authenticity. The output of the signature process is called the digital signature [1] [2]. Digital signatures are then used to provide authentication of the associated input, which is called a message [3] (refer to Figure 2). In digital signature public-key algorithms, the private key is used to sign a message, while the public key is used to verify the authenticity of the message. Moreover, digital signatures scheme used to provide the following [4]:

- Data integrity (the assurance that data has not been changed by an unauthorized party).
- Message authentication (the assurance that the source of data is as claimed).
- Non-repudiation (the assurance that an entity cannot deny commitments).

Generally, every public-key digital signature schemes is based on a mathematical problem. This problem is known as NP (Non-deterministic polynomial) hard problem. The problem is considered to be an NP hard mathematical problem if the validity of a proposed solution can be checked only in polynomial time [5].

Basically, public-key digital signature schemes are successfully classified into many major types depending on the NP mathematical hard problem. These problems are the integer factorization problem (IFP), the discrete logarithm problem (DLP), the Elliptic Curve discrete logarithm problem (ECDLP), the chaotic hard problem, etc. This study will help us to identify the strength of the used public-key digital signature schemes according to their mathematical hard problem.

Furthermore, cryptography algorithms can be classified into two board categories, secret key (symmetric) algorithms and public key (asymmetric) algorithms (refer to Figure 1). In general, digital signature scheme is categorized under public-key cryptosystem as shown by Figure 1 [1] [6]. Public-key digital signature on the other hand, works in a very different way, since there are two keys; both belong to one party, either the recipient or the sender. One key is used to accomplish half of the task (e.g. signing) while the other key will be used to complete the rest of the task (e.g. verifying).

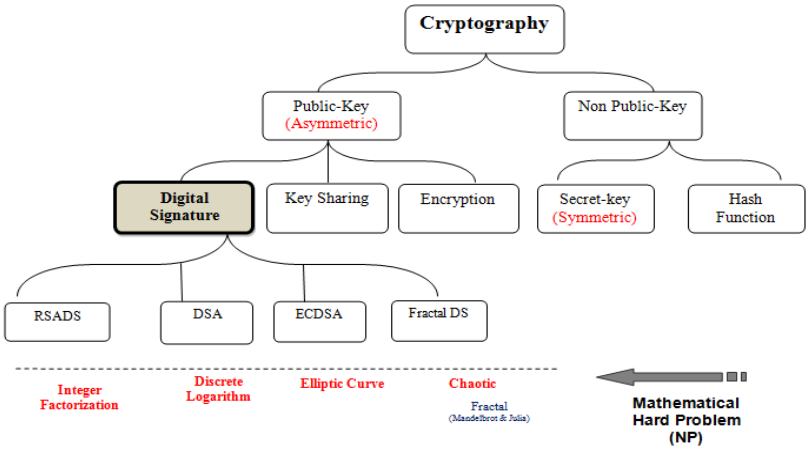

Fig. 1: Main branches of public-key scheme

This paper summarizes the development in public-key digital signature schemes that based on different mathematical NP hard problems. The paper shows a new development on public-key digital signature schemes which include RSADS [7], DSA [8], ECDSA [9], and Fractal digital signature [10].

II. PUBLIC-KEY DIGITAL SIGNATURE SCHEMES

In 1976, the first notion of a digital signature scheme was given by Whitfield Diffie and Martin Hellman, although at that time they only conjectured the existence of such scheme [11]. Soon after that, in 1978, Rivest, Shamir, and Adleman invented the first digital signature scheme which is called RSA digital signature algorithm [7]. Subsequently, there were a few more proposed digital signature algorithms such as ElGamal signature scheme [12], Undeniable signature [13] and others.

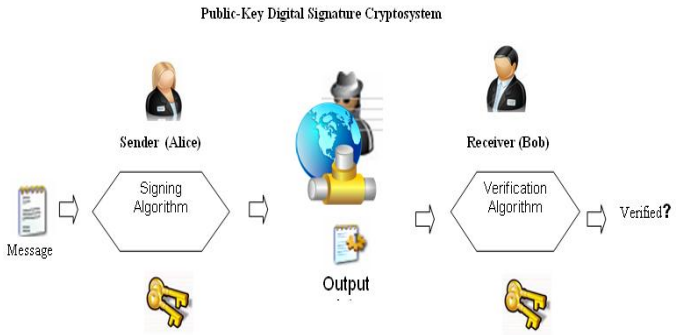

Fig. 2: Digital signature scheme

III. DIGITAL SIGNATURE BASED ON INTEGER FACTORIZATION: RSA DIGITAL SIGNATURE SCHEME

In the RSA digital signature algorithm (refer to Figure 3), the private key is used to sign the message. The signed message will be sent to the receiver with the sender's electronic signature. Figure 3 shows the steps of the RSA digital signature algorithm. To verify the contents of digitally signed data, the recipient generates a new verification key from the signed message that was received, by using his public key, and compares the verified value with the original message value. If the two values match, then the message is verified and authenticated.

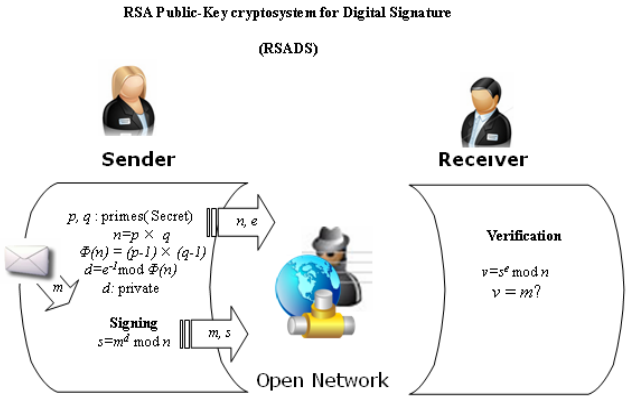

Fig. 3: RSA digital signature scheme

III-1- The RSA Digital Signature Algorithms:

Key generation algorithm (generated by receiver, Bob)

Alice must do the following (Figure 3):

1. Choose two prime numbers (p, q) randomly, secretly, and roughly of the same size.
2. Compute the modulus n as follows: $n = p \times q$.
3. Compute the $\Phi(n)$, as follows: $\Phi(n) = (p-1) \times (q-1)$.
4. Choose the key e, such that $1 < e < \Phi(n)$, and $GCD\ (e, \Phi(n)) = 1$.
5. Compute the private key d, such as $d = e^{-1} \bmod \Phi(n)$.

- Send the public (n, e) to Bob.

Signature and verification algorithms

Signature (sender - Alice)

Alice must do the following (Figure 3):

6. Determine the message m to be signed such that $0 < m < n$.

7. Sign the message as follows: $s = m^d \bmod n$.
8. Send the signature s with the message m to Bob (receiver).

Verification (receiver - Bob)

Bob must do the following (Figure 3):

9. Obtain the keys (d, n).
10. Obtain s, m from Alice.
11. Compute u as follows: $u = s^e \bmod n$.
12. Verify the message m as follows: is $m = u^{-1}$?.

II-2- The Security of RSA Digital Signature Scheme:

Similar to RSA encryption scheme [7], the security of RSA digital signature is based on the integer factorization problem and the large key size which is typically 1024-2048 bit is used [14].

IV. Digital Signature Based on Discrete Logarithm: The Digital Signature Algorithm (DSA)

In 1991, the U.S. National Institute of Standards and Technology (NIST) proposed the digital signature algorithm (DSA) and was specified in a U.S. Government Federal Information Processing Standard [8]. The algorithm is called Digital Signature Standard (DSS). Figure 4 illustrates the steps in the DSA digital signature algorithm. The DSA can be viewed as a variant of the ElGamal signature scheme[12]. Both signature schemes are based on the same mathematical problem - discrete logarithm problem. DSA bases its security on the complexity of the discrete logarithm problem in the field of Zp, where p is a prime [9].

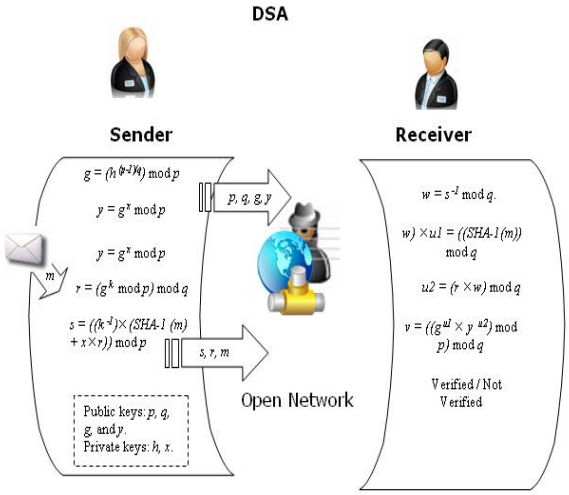

Fig. 4: DSA scheme

IV-1- The DSA Algorithms:

Key generation algorithm (generated by receiver, Alice)

Alice must do the following (Figure 4):

1. Choose a prime number (p), where $2^{L-1} < p < 2^L$ for $512 \le L \le 1024$ and L a multiple of 64.
2. Choose a prime numbers (q), where q divisor of $(p - 1)$, and $2^{159} < q < 2^{160}$.
3. Compute g as follows: $g = (h^{(p-1)/q}) \bmod p$, where $1<h<(p - 1)$, and $g > 1$.
4. Choose a random integer x, with $0 < x < q$.
5. Compute y as follows: $y = g^x \bmod p$.
6. Send (p, q, g, and y) to Bob (verifier).

Signing and verifying algorithms

Signing (sender - Alice)

Alice must do the following (Figure 4):

7. Determine the message m to be signed such that: $0 < m < p$.
8. Choose a random integer k, with $0 < k < q$.
9. Compute r as follows $r = (g^k \bmod p) \bmod q$.
10. Compute s as follows: $s = ((k^{-1}) \times (SHA\text{-}1(m) + x \times r)) \bmod q$.
11. The signature is (r, s).

- Send the signature(r, s) and the message to the receiver.
- k^{-1} is a multiplicative inverse of k in Z_q.

Verifying (receiver - Bob)

Bob must do the following (Figure 4):

12. Obtain the keys (p, q, g, and y).
13. $w = s^{-1} \bmod q$.

14. $u1 = ((SHA\text{-}1(m)) \times w) \bmod q$.
15. $u2 = (r \times w) \bmod q$.
16. $v = ((g^{u1} \times y^{u2}) \bmod p) \bmod q$.

- Verify the message m as follows: is $v = r$?.

IV-2- The Security of DSA:

The discrete logarithm problem ensures the security of the digital signature algorithm (DSA) (Figure 1). The security of a digital signature system relies on maintaining the confidentiality of the user private key [8]. However, the key space in DSA depends on the key large prime numbers (refer to Figure 4) [14]. The 128-bit DSA key space is limited the number of primes existing in the finite field of *Zp*, where *p* is the largest prime that can be represented by a 128-bit value. Hence, DSA keys size space is typically 1024 bit long [3].

V. DIGITAL SIGNATURE BASED ON ELLIPTIC CURVE: ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM (ECDSA)

In 1992, Elliptic Curve DSA (ECDSA) was proposed by Scott Vanstone requesting for the National Institute of Standards and Technology comments on their first proposal for DSS. ECDSA is the Elliptic Curve analogue of the DSA (refer to Figure 5). Elliptic Curve digital signature was accepted in three stages: ISO (International Standards Organization), ANSI (American National Standards Institute), and IEEE (Institute of Electrical and Electronics Engineers) in 1998, 1999 and 2000, respectively.

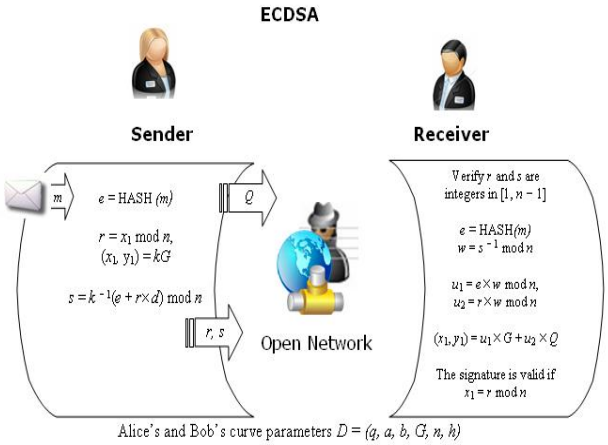

Fig. 5 ECDSA scheme

The operations in the ECDSA are different from the operations in DSA because the ECDSA operations are based on the Elliptic Curve's mathematical operations (refer to Figure 5). Supposedly the ECDSA uses smaller key size for similar security levels compared with DSA. For example, the DSA with 1024-bit *p* and 160-bit *q* and ECDSA with the 160-bit prime field both produce 320-bits signatures in less time [9].

V-1- The ECDSA Algorithms:

Alice and Bob must agree with the curve parameters $D = (q, a, b, G, n, h)$.

<u>Algorithm for Signature generation algorithm (Sender - Alice)</u>

Alice must do the following (Figure 5):

1. Choose the associated keys suitable for Elliptic Curve Cryptography (d, Q) (where d is a randomly selected integer in the interval $[1, n-1]$)) and Q is a public key (where $Q = d \times G$).
2. Calculate e = HASH(m), where HASH is a Cryptographic Hash function, such as SHA-1.
3. Choose a random integer k, with $1 < k < n$.
4. Calculate $r = x_1 \bmod n$, where $(x_1, y_1) = kG$. If $r = 0$, go back to step 2.
5. Calculate $s = k^{-1}(e + r \times d) \bmod n$. If $s = 0$, go back to step 2.
6. Note that the signature is the pair (r, s).

<u>Algorithms for Signature verification (Receiver – Bob)</u>

Bob must do the following (Figure 5):

7. Obtain Alice's public keys (Q).
8. Obtain Alice's signature (r, s).
9. Verify r and s are integers in $[1, n-1]$. If not, the signature is invalid.
10. Calculate e = HASH(m), where HASH is the same function used in the signature generation.
11. Calculate $w = s^{-1} \bmod n$.
12. Calculate $u_1 = e \times w \bmod n$ and $u_2 = r \times w \bmod n$.
13. Calculate $(x_1, y_1) = u_1 \times G + u_2 \times Q$.
14. The signature is valid if $x_1 = r \bmod n$.

V-2- The Security of Elliptic Curve Digital Signature Algorithm:

The security in the Elliptic Curve Digital Signature Algorithm is based on discrete logarithm problem. However, the Elliptic Curve discrete logarithm is considered more difficult than the discrete logarithm problem (refer to Fig. 1). In addition, the ECDSA can use short key size comparing with the key size used in the other traditional discrete logarithm. The ECDSA keys are typically 128 bit [9] [15].

VI. DIGITAL SIGNATURE BASED ON THE MANDELBROT AND JULIA FRACTAL SETS

Mandelbrot and Julia Fractal shapes (refer to Figure 6) consist of complex number points, computed by the recursive functions [10]. In this Section, with the aid of Figure 7, we are going to explain in brief the idea of the Fractal digital signature scheme based on Fractal set.

The Fractal based digital signature uses a specific Mandelbrot function, *Mandelfn* and similarly, a specific Julia function, *Juliafn* (refer to Equation 1 and 2). Figure 7 shows an image which was generated from the *Mandelfn* and Julia function, *Juliafn*. However, the value which is generated by *Mandelfn* must belong to the Mandelbrot set, and likewise, the value generated by *Juliafn* must belong to the Julia set [16]. This scheme sets *f( )* as shown by Equation 3 for *Mandelfn* function and Equation 4 for *Juliafn* function.

$$z_n = c \times f\ (z_{n-1});\ z(0) = c;\ c, z \in \mathbf{C}; n \in \mathbf{Z}. \quad (1)$$

$$z_n = c \times f\ (z_{n-1});\ z(0) = y;\ y, c, z \in \mathbf{C}; n \in \mathbf{Z}. \quad (2)$$

$$z_n d = z_{n-1} \times c^2 \times d; z, c, d \in \mathbf{C}; n \in \mathbf{Z}. \quad (3)$$

$$z_k e = z_{k-1} \times c^2 \times e; z, c, e \in \mathbf{C}; k \in \mathbf{Z}. \quad (4)$$

$$s = c^{k-x} \times (z_n d)_k e \times m; \quad s, c, e, d \in \mathbf{C};\ n, x, k \in \mathbf{Z};\ m \in \mathbf{R}. \quad (5)$$

$$v = c^{n-x} \times (z_k e)_n d \times m; \quad v, c, e, d \in \mathbf{C};\ n, x, k \in \mathbf{Z};\ m \in \mathbf{R}. \quad (6)$$

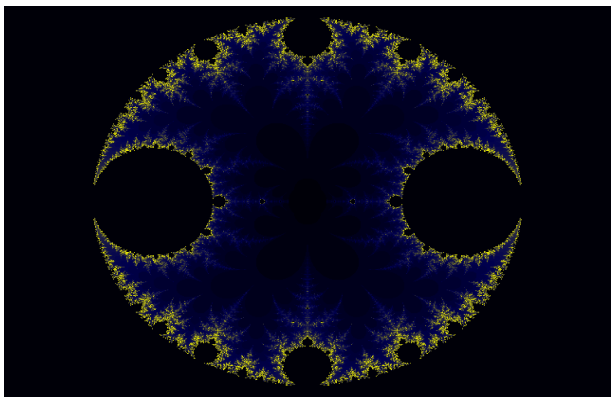

Fig. 6: Mandelfn image with the sine function (*sin( )*) [17]

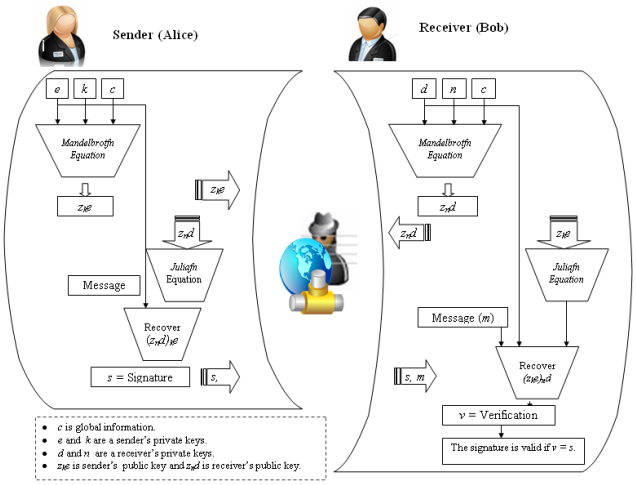

Fig. 7: Fractal digital signature algorithm

VI-1- Fractal DS Algorithms:

Algorithm for Signature generation algorithm:

Alice and Bob must do the following (Figure 7):

1. Sender and receiver must agree and use the public domain value, *c*.
2. The receiver, Bob, will generate *e* and *k* as the private keys.
3. The sender, Alice, generates *n* and *d* as her private keys.
4. Sender and receiver use their private values as well as the value *c* as inputs to the Mandelbrot function to produce the public keys $z_n d$ and $z_k e$.
5. Then Bob and Alice must exchange their public keys.

Algorithm for Signature generation algorithm (Sender - Alice)

Alice must do the following (Figure 7):

6. Alice will obtain Bob's public key, $z_n d$ and will use these values together with her private key and the plaintext, as inputs to the Julia function to produce the signature *s*, which will then be sent with the message to Bob (refer to Equation 5).

Algorithms for Signature verification (Receiver – Bob)

Bob must do the following (Figure 7):

7. Bob must obtain Alice's public key, $z_k e$, the signature *s* and the message *m* from Alice which will be used as input values together with his own private key to Julia function, to verify the message *v* (refer to Equation 6).

VI-2- The Security of Fractal Digital Signature Scheme:

The strength of the algorithm and the size of the key used, are key factors in the security of digital signature protocol. Nevertheless, Fractal digital signature algorithm is efficient since the algorithm uses small key size and executes faster than other public key digital signature schemes. It is computationally impossible to attack the Fractal digital signature protocol. Since the system is based on the Chaos NP hard problem [18].

The Fractal digital signature is based on The Chaos hard problem, whereby, the chaotic nature of the Fractal equations ensures the security of this algorithm. However, the crucial key size in Fractal digital signature algorithm is chosen to prevent a brute force attack. The key space in Fractal digital signature depends on the size of the key [14]. The fractal based digital signature scheme provides high level of security at a much low cost, in term of key size and execution time.

VII. CONCLUSION

This paper gives the reader basic concepts used throughout the rest of the study which are related to the concepts in digital signature cryptosystem. In addition, we studied some digital signature schemes (refer to Table 1) which are based on different mathematical hard problems as classified earlier. Those classifications help the reader to be familiar with the public-key digital signature cryptosystem. However, the security protection of the discussed digital signature schemes depend on the mathematical NP-hard problems and the randomness of the output generated.

TABLE 1
EXAMPLE OF DIGITAL SIGNATURE SCHEMES

Mathematical Hard Problem	Digital Signature Schemes	
	Efficiency	Typical Key Size for High Performance
Integer Factorization RSADS	The speed in RSADS is considered much slower than other symmetric cryptosystems	Large Prime Number (1024-bit)
Discrete Logarithm DSA	DSA is probabilistic , however, The security of a digital signature system relies on maintaining the confidentiality of the user private key	Large Prime Number (1024-bit)
Elliptic Curve ECDSA	The discrete logarithm problem on elliptic curve cryptosystem is more difficult than the other mathematical problem	Short Key (128-bit)
Chaos Fractal DS	The fractal based digital signature scheme provides high level of security at a much low cost, in term of key size and execution time.	Short Key (128-bit)

ACKNOWLEDGMENTS

The author would like to thank Al Zaytoonah University of Jordan for supporting this study.

REFERENCES

[1] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, pp.4-15, 516, 1996.
[2] S. Burnett, and S. Paine, "*RSA Security's Official Guide to Cryptography"*, Berkeley: Osborne/McGraw-Hill, 2001.
[3] Public Law, Electronic Signatures in Global And National Commerce Act. Weekly Compilation of Presidential Documents, 36, Public Law, pp.106-229, 2000.
[4] Digital Signature, Wikipedia. Available from World Wide Web: http://en.wikipedia.org/wiki/Digital_signature , 2007.
[5] RSA Laboratories, "What is a Hard Problem. RSA the Security Division of EMC", 2007.
[6] I. Branovic, R. Giorgi, E. Martinelli, "Memory Performance of Public-Key Cryptography Methods in Mobile Environments", *ACM SIGARCH Workshop on Memory performance*: *Dealing with Applications, systems and architecture (MEDEA-03)*, New Orleans, LA, USA, pp. 24-31, 2003.
[7] R. A. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, 21(2), pp.120-126, 1978.
[8] G. Locke, and P. Gallagher, "Digital Signature Standard (DSS)", *Federal Information Processing Standards Publication 186-3, Computer Systems Laboratory, National Institute of Standards and Technology, FIPS PUB 186-3*, 2009.
[9] D. Johnson, A. Menezes, S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", *Certicom Corporation*, 2001.
[10] M. Alia, and A. Samsudin, "A New Digital Signature Scheme Based on Mandelbrot and Julia Fractal Sets", *American Journal of Applied Sciences, AJAS*, 4(11), pp. 850-858, 2007.
[11] W. Diffie, and M. E. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, IT-22, pp. 644-654, 1976.
[12] T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *IEEE Transactions on Information Theory*, IT-31(4), pp. 469–472, 1985.
[13] C. David, and H. V. Antwerpen, "Undeniable Signatures", *Crypto'89, LNCS 435, Springer-Verlag*, Berlin, pp. 212-216, 1990.
[14] B. Elaine, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for Key Management – Part 1: General", *NIST Special Publication*, 800-57, 2006.
[15] G. V. S. Raju, and R. Akbani, "Elliptic Curve Cryptosystem and its Applications", *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics (IEEE-SMC)*, 2003.
[16] N. Giffin, "Fractint", TRIUMF Project at The University of British Columbia Campus in Vancouver B.C. Canada, Available from World Wide Web: http://spanky.triumf.ca/www/fractint/fractint.html , 2006.
[17] A. R. Johansen, "ARJ's Fractal Gallery", Available from World Wide Web: http://arj.nvg.org/pic/gallery/ , 2000.
[18] M. Ruhl, and H. Hartenstein "Optimal Fractal Coding is NP-Hard", *Proceedings DCC97 Data Compression Conference*, J. A. Storer, M. Cohn (eds.), IEEE Computer Society *Press*, 1997.