Combining Public-Key Encryption with Digital Signature Scheme

Mohammad Ahmad Alia(✉)

Computer Information Systems Department, Faculty of Sciences and IT,
Al-Zaytoonah University of Jordan, Amman, Jordan
dr.m.alia@zuj.edu.jo

Abstract. This paper presents the possibilities of combing public-key encryption and digital signature algorithms which are actually based on different mathematical hard problems. Since the output of the combination produces an Encrypted signed message. In general, most of the currently used public-key algorithms are computationally expensive with relatively lengthy key requirement due to the dependency on the number theory. Therefore, it's important to show a combinational protocols which are based on different mathematical hard problem. In some sense, difficult to solve. In the combined scheme, we present the powerful and practical encryption digital signature scheme and its security level and execution time.

Keywords: Cryptography · NP-Hard problem · Digital signature · Public key encryption

1 Introduction

Cryptography algorithms can be classified into two board categories, secret key (one key, single key, symmetric) algorithms and public key (two key, asymmetric) algorithms (refer to Fig. 1). In general, Cryptography protocol employs public key cryptosystem to exchange the secret key and then uses faster secret key algorithms to ensure confidentiality, integrity, non-repudiation, authentication, and accessibility of the data stream [4, 5]. In 1976, Diffie-Hellman [2, 6] developed the concept of the public-key cryptosystem (refer to Figs. 1 and 2), and the first asymmetric encryption protocol is the RSA [7] algorithm which was published in 1978. As well as, there are many others asymmetric encryption algorithms issued since the RSA. Among them are Rabin [8], ElGamal [9], and Elliptic Curve [10].

In public-key Cryptography, Every public-key algorithm is actually based on a mathematical problem. These problems are called "mathematical hard problems" and are classified into two major groups according to the Cryptography standards. These groups are P (Polynomial) and NP (Non-deterministic polynomial). The P hard problem is defined when the problem is solved in polynomial time. Whereby, if the validity of a proposed solution can be checked only in polynomial time then the problem is considered as an NP hard mathematical problem [1–3]. Basically, public-key algorithms are classified into many major types depending on the mathematical hard problem. These problems are the discrete logarithm problem (DLP), the

A.E. Hassanien et al. (eds.), *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*, Advances in Intelligent Systems and Computing 533, DOI 10.1007/978-3-319-48308-5_83

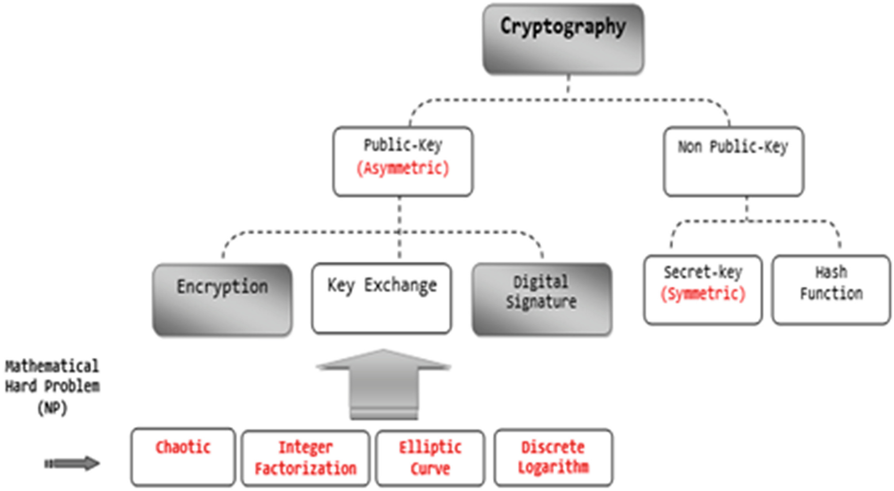


Fig. 1. Main branches of public-key scheme

integer factorization problem (IFP), the Elliptic Curve discrete logarithm problem (ECDLP), the chaotic hard problem, etc. This following survey study will help us to identify the strength of the used public-key algorithms according to their mathematical hard problem (refer to Fig. 1).

In asymmetric encryption protocols, there is a pair of keys, one of which is known to encrypt the plaintext and called as the public key, while the other key is known as the private key and is used to decrypt the encrypted plaintext.

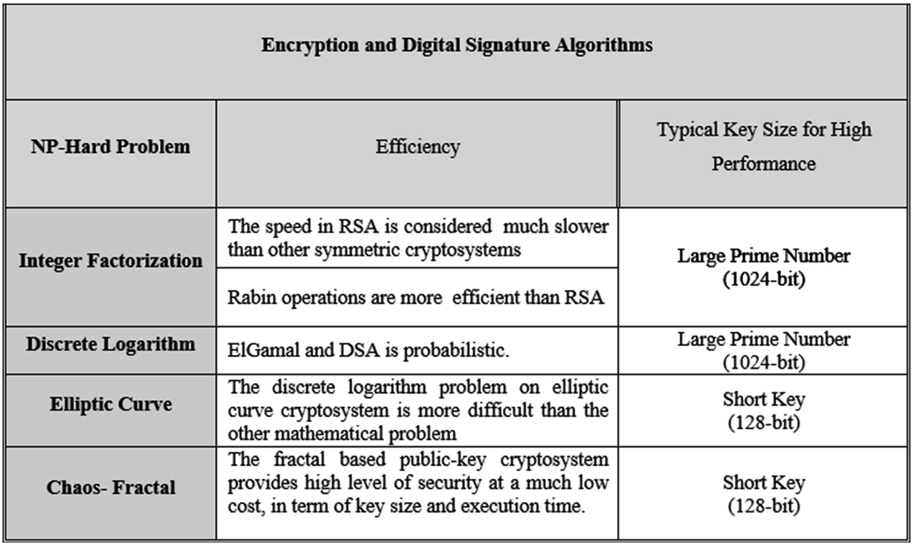
As discussed earlier, every public-key encryption algorithm is based on a NP mathematical problem that is in some sense difficult to be solved, therefore the public-key algorithms are classified according to their hard problems in this subsection. Table 1 show the key size for prime based algorithms (RSA, DSA, etc.) and integer based algorithm (ECC, Chaotic) algorithms, regarding to the resistance to brute force attacks. The keys space for RSA and DSA were calculated based on the number of primes existed for particular key sizes [14].

This paper developed a public-key encryption with digital signature scheme which are based on mathematical hard problem. The paper discusses the possibility of creating a combinational public key encryption and digital signature protocol.

2 The Proposed Encryption and Digital Signature Protocol

The combination between encryption algorithm and digital signature algorithm provides confidentiality, Integrity, authentication, and non-reputation services for messages. In this article, both algorithms are defined as any of the different mathematical problems; since it's possible to combine encryption based IFP with digital signature based DLP in some cases. However, in the proposed solution the sender and the receiver must generate their own private and public keys. The sender must compute

Table 1. Public-Key Encryption and Digital Signature protocols (Efficiency and Key Size) [14]



Encryption and Digital Signature Algorithms		
NP-Hard Problem	Efficiency	Typical Key Size for High Performance
Integer Factorization	The speed in RSA is considered much slower than other symmetric cryptosystems	**Large Prime Number (1024-bit)**
	Rabin operations are more efficient than RSA	
Discrete Logarithm	ElGamal and DSA is probabilistic.	Large Prime Number (1024-bit)
Elliptic Curve	The discrete logarithm problem on elliptic curve cryptosystem is more difficult than the other mathematical problem	Short Key (128-bit)
Chaos- Fractal	The fractal based public-key cryptosystem provides high level of security at a much low cost, in term of key size and execution time.	Short Key (128-bit)

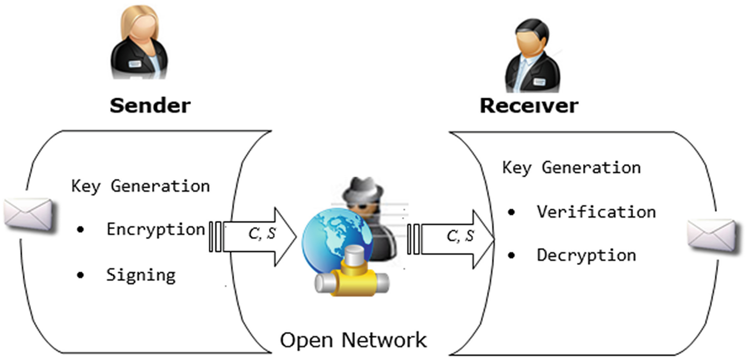


Fig. 2. Encryption and signature combinational protocol

his/her keys for digital signature purpose while the receiver will compute the encryption and decryption keys, in parallel with the sender.

As shown by Fig. 2, the message is selected by the sender and then encrypted and signed to be sent over the insecure network to the receiver. After receiving the encrypted and signed message, the receiver should verify the signature authenticity before decrypting the ciphertext.

2.1 Combination of RSA and DSA Protocol

Figure 3 shows the combinational protocol between RSA and DSA algorithms, since RSA is based on integer factorization problem while DSA is based on discrete logarithm problem.

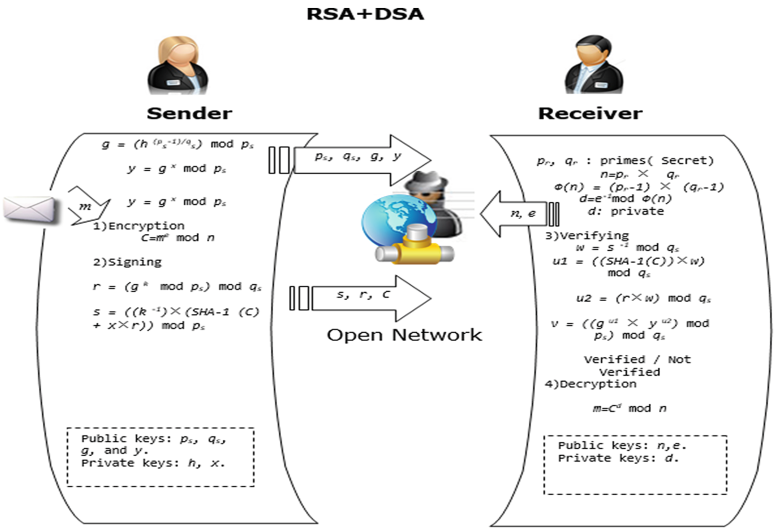

RSA+DSA
Sender
Receiver
$g = (h^{(p_s-1)/q_s}) \bmod p_s$
$y = g^x \bmod p_s$
$y = g^x \bmod p_s$
m
1)Encryption
$C=m^e \bmod n$
2)Signing
$r = (g^k \bmod p_s) \bmod q_s$
$s = ((k^{-1}) \times (\text{SHA-1 }(C) + x \times r)) \bmod p_s$
Public keys: p_s, q_s, g, and y.
Private keys: h, x.
p_s, q_s, g, y
s, r, C
n, e
Open Network
p_r, q_r : primes(Secret)
$n=p_r \times q_r$
$\Phi(n) = (p_r-1) \times (q_r-1)$
$d=e^{-1} \bmod \Phi(n)$
d: private
3)Verifying
$w = s^{-1} \bmod q_s$
$u1 = ((\text{SHA-1}(C)) \times w) \bmod q_s$
$u2 = (r \times w) \bmod q_s$
$v = ((g^{u1} \times y^{u2}) \bmod p_s) \bmod q_s$
Verified / Not Verified
4)Decryption
$m=C^d \bmod n$
Public keys: n,e.
Private keys: d.

Fig. 3. RSA + DSA protocol

Key generation algorithm (generated by sender, Alice) - Alice must do:

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

Key generation algorithm (generated by receiver, Bob) - Bob must do:

6. Choose two prime numbers (p_r, q_r) randomly, secretly, and roughly of the same size.
7. Compute the modulus n as follows: $n = p_r \times q_r$.
8. Compute the $\Phi(n)$, as follows: $\Phi(n) = (p_r-1) \times (q_r-1)$.
9. Choose the public key e, such that $1 < e < \Phi(n)$, and GCD (e, $\Phi(n)$) = 1.
10. Compute the decryption key d, where $d = e^{-1} \bmod \Phi(n)$.
    Determine the public keys (e, n) and determine the private keys ($\Phi(n)$, d).

Encryption and Signing (sender - Alice) - Alice must do the following:

11. Obtain the public keys (e, n).
12. Determine the message m to be encrypt such that $0 < m < n$.
13. Encrypt the message as follows, $c = m^e \bmod n$.
14. Choose a random integer k, with $0 < k < q_s$.
15. Compute r as follows $r = (g^k \bmod p_s) \bmod q_s$.
16. Compute s as follows: $s = ((k^{-1}) \times (SHA\text{-}1(C) + x \times r)) \bmod q_s$.
    The signature is (r, s).
    Send the signature and the ciphertext (c, r, s) to the receiver.
    k^{-1} is a multiplicative inverse of k in Z_q.

Verifying and Decryption (receiver - Bob) - Bob must do the following:

17. Obtain the keys (p_s, q_s, g, and y).
18. $w = s^{-1} \bmod q_s$.
19. $u1 = ((SHA\text{-}1(C)) \times w) \bmod q_s$.
20. $u2 = (r \times w) \bmod q_s$.
21. $v = ((g^{u1} \times y^{u2}) \bmod p_s) \bmod q_s$.
    Verify the message m as follows: is $v = r?$.
22. Obtain the ciphertext c from Alice.
23. Recover the message as follows, $m = c^d \bmod n$.

2.2 Combination of RSA and RSADS

Figure 4 shows the combinational protocol between RSA and RSADS algorithms, since both RSA algorithms are based on integer factorization problem.

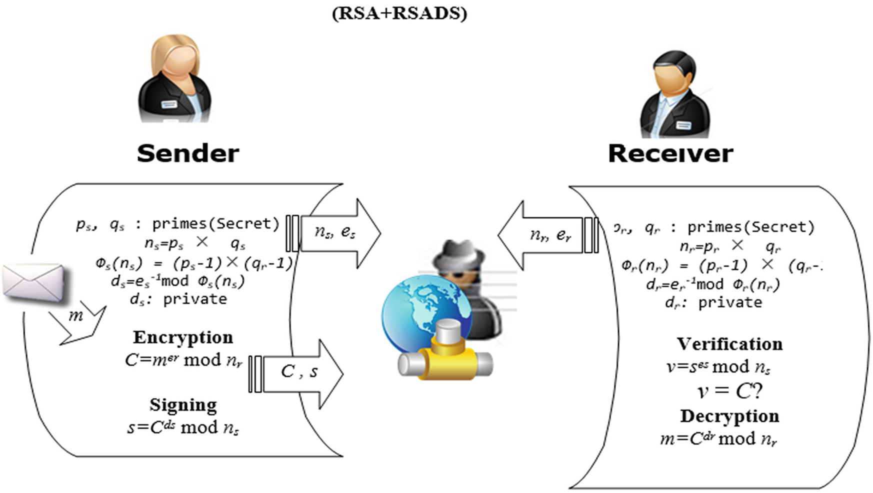


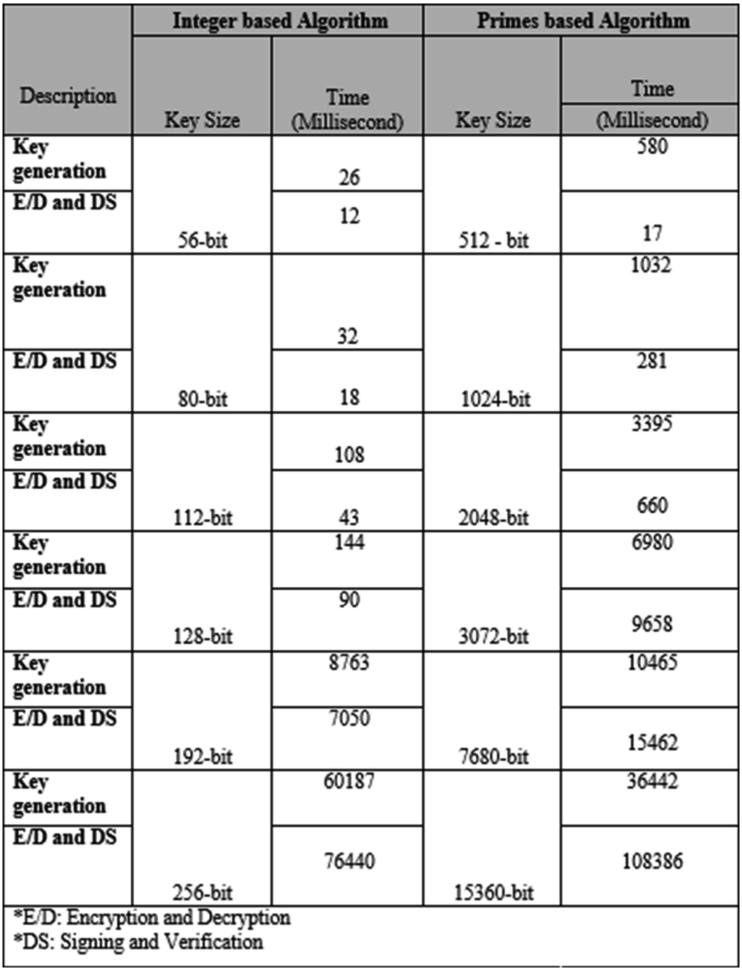
Fig. 4. RSA + RSADS protocol

Key generation algorithm (generated by sender, Alice) - Alice must do:

1. Choose two prime numbers (p_s, q_s) randomly, secretly, and roughly of the same size.
2. Compute the modulus n as follows: $n_s = p_s \times q_r$.
3. Compute the $\Phi(n)$, as follows: $\Phi_s(n_s) = (p_s - 1) \times (q_s - 1)$.
4. Choose the public key e, such that $1 < e_s < \Phi_s(n_s)$, and GCD (e_s, $\Phi_s(n_s)$) = 1.
5. Compute the decryption key d_s, where $d_s = e_s^{-1} \bmod \Phi_s(n_s)$.

Determine the public keys (e_s, n_s) and determine the private keys ($\Phi_s(n_s)$, d_s).

Table 2. Performance evaluation between integers and primes based public key algorithms



Description	Integer based Algorithm		Primes based Algorithm	
	Key Size	Time (Millisecond)	Key Size	Time (Millisecond)
Key generation	56-bit	26	512 - bit	580
E/D and DS		12		17
Key generation	80-bit	32	1024-bit	1032
E/D and DS		18		281
Key generation	112-bit	108	2048-bit	3395
E/D and DS		43		660
Key generation	128-bit	144	3072-bit	6980
E/D and DS		90		9658
Key generation	192-bit	8763	7680-bit	10465
E/D and DS		7050		15462
Key generation	256-bit	60187	15360-bit	36442
E/D and DS		76440		108386

*E/D: Encryption and Decryption
*DS: Signing and Verification

Key generation algorithm (generated by receiver, Bob)- Bob must do:

6. Choose two prime numbers (p_r, q_r) randomly, secretly, and roughly of the same size.
7. Compute the modulus n as follows: $n_r = p_r \times q_r$.
8. Compute the $\Phi(n)$, as follows: $\Phi_r(n_r) = (p_r - 1) \times (q_r - 1)$.
9. Choose the public key e, such that $1 < e_r < \Phi_r(n_r)$, and GCD $(er, \Phi_r(n_r)) = 1$.
10. Compute the decryption key d_r, where $d_r = er^{-1} \, mod \, \Phi_r(n_r)$.
    Determine the public keys (e_r, n_r) and determine the private keys ($\Phi_r(n_r)$, d_r).

Encryption and Signing (sender - Alice) - Alice must do the following:

11. Determine the message m to be encrypt such that $0 < m < n_r$.
12. Encrypt the message as follows, $c = m^{er}$ mod n_r.
13. Sign the ciphertext C as: $c = C^{ds}$ mod n_s
    Send the signature and the ciphertext (C, s) to the receiver.

Verifying and Decryption (receiver - Bob) - Bob must do the following:

14. $v = s^{es}$ mod n_s
    Verify the message m as follows: is $v = C?$
15. Obtain the ciphertext C from Alice.
16. Recover the message as follows, $m = C^{dr}$ mod n_r.

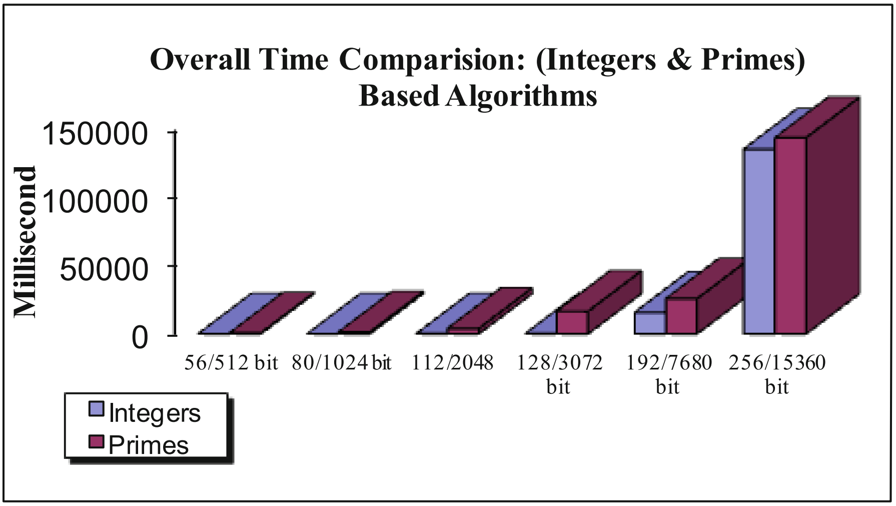


Fig. 5. Overall time comparison between integers and primes based public key algorithms' time

3 Performance Evaluation Based on Equivalent Key Sizes for Fractal and Public-Key Encryption Protocol

We compare the performance of the integer based public-key algorithm against the well-known prime's public-key algorithms (such as RSA and DSA) for the combined encryption and digital signature cryptosystems. Table 2 shows the performance for both approaches. Both protocols were coded in C++ with GMP library. Also, Miller-Rabin algorithm [18] is implemented for primarily test which was coded using C ++ and GMP as well. However, the comparison between integers and primes based public-key cryptosystems shows that integers based public key encryption and digital signature algorithms performs better than primes based public key algorithms in general. As Fig. 5 indicate, integers based public-key algorithm (ECC), provides higher level of security at a much lower cost, both in term of key size and execution time. Moreover,

4 Conclusions

This paper presents and implemented a scheme of combining public-key encryption and digital signature algorithms and proposed an encryption digital signature protocol. However, an overall running time that compare between integers and primes based public key algorithms' time were presented.

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

References

1. Alia, M.A.: Survey on mathematical hard problems based public-key cryptosystems. World Acad. Sci. Eng. Technol. **68**, 395–402 (2010)
2. Alia, M.A.: Cryptosystems based on chaos theory. In: International Symposium on Chaos, Complexity and Leadership, 17–19 December 2013 (2013)
3. RSA Laboratories, What is a Hard Problem. RSA the Security Division of EMC (2007)
4. Branovic, I., Giorgi, R., Martinelli, E.: Memory performance of public-key cryptography methods in mobile environments. In: ACM SIGARCH Workshop on Memory Performance: Dealing with Applications, Systems and Architecture (MEDEA 2003), New Orleans, LA, USA, pp. 24−31 (2003)
5. Menezes, A., Van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography, pp. 4–15, 516. CRC Press (1996)
6. Diffie, W., Hellman, M. E.: New directions in cryptography. IEEE Trans. Inf. Theory **IT-22**, 644–654 (1976)
7. Rivest, R.A., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)

8. Rabin, M.O.: Digitalized signatures and public-key functions as intractable as factorization: the ACM digital library. Technical report. UMI Order Number: TR-212, Massachusetts Institute of Technology (1979)
9. ElGamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **IT-31**(4), 469–472 (1985)
10. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **48**, 203–209 (1987)
11. Stallings, W.: Cryptography and Network Security Principles and Practices. Pearson Education, 3 edn. (2003)
12. Al-Tuwaijry, F.A., Barton, S.K.: A high speed RSA processor. In: IEEE Conference, 2–6 September, Loughborough, UK, pp. 210–214 (1991)
13. Burnett, S., Paine, S.: RSA Security's Official Guide to Cryptography. Osborne/McGraw-Hill, Berkeley (2001)
14. Elaine, B., Barker, W., Burr, W., Polk, W., Smid, M.: Recommendation for Key Management–Part 1: General NIST Special Publication 800-57 (2006)
15. Chaum, D., van Antwerpen, H.: Undeniable signatures. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 212–216. Springer, Heidelberg (1990)
16. Burrows, J.H.: Digital Signature Standard (DSS). In: Federal Information Processing Standards Publication 186, Computer Systems Laboratory, National Institute of Standards and Technology, Fips Pub, vol. 186, pp. 1–5 (1994)
17. Public Law, Electronic Signatures in Global and National Commerce Act. Weekly Compilation of Presidential Documents, Public Law, vol. 36, pp. 106–229 (2000)
18. MediaWiki: Literate Programs, Miller-Rabin (2006). http://en.literateprograms.org/Miller-Rabin_primality_test_(C,_GMP)