Gulf Journal of Mathematics
Vol 6, Issue 4 (2018) 37-43

A DIGITAL SIGNATURE SCHEME BASED SIMULTANEOUSLY ON THE DSA AND RSA PROTOCOLS

L. ZAHHAFI* AND O. KHADIR

Abstract. In this paper, we propose a new digital signature scheme based on hard mathematical problems. We reenforce the DSA protocol by using the RSA cryptosystem. The security and complexity of the signature are analyzed.

1. Introduction

The digital signature (See [9] pp. 475, [18] pp. 271 and [18] pp. 178) is a mechanism that allows to verify the identity of the signer and the integrity of a document. There are three steps to execute a digital signature process namely:
The creation of keys: This step is to generate the private and public keys that will be needed to continue the signature process. To find the secret key chosen by the signer, it will be necessary to solve one of the mathematical problems often used in the public key cryptography. The size of selected parameters is also an important point in the creation of the keys. So, any signer must takes parameters with lengths that protect his secret keys.
The signature generation: It depends on the method chosen. The signer uses his private key to calculate the signature of his document. No one should be able to sign in his place since he's the only one holding the values of private keys.
The signature verification: To verify the signature received, the verifier uses the signer's public keys. Then, he accepts or rejects the signature.
To ensure the security of any confidential action, the cryptographic protocols are based on mathematical problems that are very difficult to solve. Among these problems we cite the discrete logarithm and factorization of large numbers [16].
In 1978, Rivest, Shamir and Adeleman [14] created a strong system to encrypt secret messages as well to sign documents. This method is based on the difficulty of solving a modular polynomial equation of this form: $x^a \equiv b\,[n]$ with: a is an integer, n is a large composite number and x the unknown variable. Then, Rabin [13] proposed a new method of signing using the equation: $x^2 \equiv b\,[n]$. Other signature protocols based on the problem of factorization [11] were proposed as the work of Paillier [12].
The discrete logarithm [1] is also widely used in the signature schemes. In 1985,

Date: Received: Dec 2, 2018

*Corresponding author.

2010 *Mathematics Subject Classification.* 11T71, 94A60.

Key words and phrases. DSA protocol, RSA scheme, Discrete Logarithm Problem, Factorization.

after the work of Diffie and Hellman [5], ElGamal [6] proposed an efficient signature protocol based on this equation: $a^x \equiv b\,[p]$ such that: p is a large prime, a a primitive root of the finite multiplicative group Z/pZ and x is the unknown variable. Several variants of this scheme have been created as in [8, 7]. In 1991, Schnorr [15] created a signature protocol using the discrete logarithm. Then, the DSA method [10] shared in 1994 has been widely used in the digital signature. It has also been improved in some works. In 2008, Chen-Yu Lee and Wei-Shen Lai [4] presented an extended version of the DSA in which they developed the sizes of parameters used in the scheme.

This article, describes a new version of the DSA protocol. We have integrated the RSA algorithm into the DSA method. So, the protocol that we propose is protected by using two hard mathematical problems.

The paper is organized as follows: In the next section, we recall the DSA signature. In section 3, we present the RSA cryptosystem. Then, we show the steps of our signature protocol in section 4. We end by a conclusion in section 5.

We denote by $\mathbb{Z}/n\mathbb{Z}$ the finite ring of modular integers for every positive integer n. We write $x \equiv y[n]$ if n divides the difference $x - y$, and $x = y \, mod \, n$ if x is the remainder in the division of y by n, with: x,y and n are three integers.

We begin by describing the DSA signature system.

2. The DSA protocol [10]

In this section, we review the basic DSA protocol in three steps, followed by an example.

2.1. Keys generation.

Alice chooses two primes P and q with the property that: q divides $P - 1$, $2^{t-1} < q < 2^t$ where: $t \in \{160, 256, 384, 512\}$ and $2^{L-1} < P < 2^L$ with: $768 < L < 1024$ and L is a multiple of 64.

Then, she selects a primitive root g mod P and computes $\alpha = g^{\frac{P-1}{q}} \, mod \, P$. Alice chooses also an integer a such that $1 \leq a \leq q - 1$ and computes $y = \alpha^a \, mod \, P$. Finally, she publishes (P, q, α, y) and keeps the parameter a secret as her private key.

2.2. The signature generation.

To sign a message m, Alice starts by selecting a random secret integer k, with: $1 \leq k \leq q$. Then, She computes $r = (\alpha^k \, mod \, P) \, mod \, q$. She computes $s = \frac{h(m)+a.r}{k} \, mod \, q$, with $h(m)$ is the hash ([17] pp. 119) of the message m. Finally, Alice's signature is the pair: (r, s).

2.3. The signature verification.

To verify Alice's signature, Bob calculates: $U_1 = \frac{h(m)}{s} \, mod \, q$ and $U_2 = \frac{r}{s} \, mod \, q$.

Then, he computes: $V = ((\alpha^{U_1} y^{U_2}) \, mod \, P) \, mod \, q$. Bob checks if $V = r$, then he accepts or rejects Alice's signature.

Example 2.1. Let's take $P = 4608587$ and $q = 1319$ two primes such that: q divides $P - 1$, $g = 2$ and $\alpha = g^{\frac{(P-1)}{q}} \, mod \, P = 357285$. $a = 75$ so $y = \alpha^a \, mod \, P = 3461023$.

Assume that Alice wants to sign the message m and suppose that $h(m) = 421$.

So, she selects $k = 15$ and computes $r = (\alpha^k\ mod\ P)\ mod\ q = 29$. Then, she calculates: $s = \frac{h(m)+a.r}{k}\ mod\ q = 261$.
Alice's signature for the message m is the pair (29,261).
To verify this signature, Bob finds: $U_1 = \frac{h(m)}{s}\ mod\ q = 886$, $U_2 = \frac{r}{s}\ mod\ q = 1026$ and $V = ((\alpha^{U_1} y^{U_2})\ mod\ P)\ mod\ q = 29$. Then he accepts the signature since $V = r$.

2.4. **Security of the method.** The DSA system is an efficient digital signature protocol. It is based on a hard mathematical problem namely the discrete logarithm problem. So, discovering the secrete key x of the signer that verifies the equation: $a^x = b\ mod\ P$ implies breaking the system. However, the algorithm is based on the use of a random parameter k kept secret . We mention that signing two documents using the same k reveals the system private key. So, it's recommended to change at each signature the k value.

Next section presents the RSA scheme.

3. The RSA system [14]

In this section, we recall how the RSA method works. We start by the production of keys.

3.1. **Keys generation.** The RSA algorithm is based on the hardness of factoring integers. To generate RSA keys, Alice starts by computing n as the product of two very large primes p and q: $n = pq$. Then, she finds $\varphi(n) = (p-1)(q-1)$ and selects an integer e that verifies $gcd\,(e, \varphi(n)) = 1$. Alice calculates her decryption key d such that: $d \equiv \frac{1}{e}\,[\varphi(n)]$. She keeps d secret and publishes (e, n). Parameters p and q will be destroyed for obvious security reasons.

Remark 3.1. The RSA method is also valid for three prime numbers $p, p\prime$ and q that verify: $n = pp\prime q$. The Euler function becomes: $\varphi(n)) = (p-1)(p\prime-1)(q-1)$. In our protocol, We will use the RSA with three prime numbers.

3.2. **The signature generation.** To sign a known message m according to the RSA system, Alice have to find the unknown X that verifies the fallowing equation:

$$h(m) \equiv X^e\,[n] \tag{3.1}$$

Such that $h(m)$ is the hash of the message m.
Using her private key d, Alice obtains: $X \equiv h(m)^d\,[n]$. Then, she sends X to Bob.

3.3. **The signature verification.** The verifier Bob checks Alice's signature by replacing the value X in the equation: $h(m) \equiv X^e\,[n]$. Then, he accepts or rejects her signature.
This is a result of the application of the Euler theorem. Indeed:

$$X^e \equiv (h(m)^d)^e \equiv h(m)^{e.d} \equiv h(m)^{k.\varphi(n)+1} \equiv h(m)\,[n] \text{ (for some integer } k).$$

3.4. **Security of the protocol.** As mentioned in the original article of the RSA system, breaking the protocol can be done in different ways. The method is based on one of hard mathematical problems namely factorization of large integers. Once an attacker gets to factor the composite number n, he can destroy the system completely [2]. Authors of the paper added that computing the value of $\varphi(n)$ without factoring or even finding the secret key d from or without computing the Euler function are also possibilities to attack the scheme. However, calculating d or $\varphi(n)$ is as difficult as factoring [16].

Now we move to our main contribution.

4. Our signature scheme

In this section we present a new digital signature protocol based simultaneously on the DSA and RSA schemes.

4.1. **Keys production.** Alice starts by choosing four primes P, p, $p\prime$ and q that verify: $P = 2pp\prime q + 1$, $2^{t-1} < q < 2^t$ such that: $t \in \{160, 256, 384, 512\}$, $2^{L-1} < P < 2^L$ where: $768 < L < 1024$ and L is a multiple of 64.
Then, she selects a primitive root g mod P and computes: $\alpha = g^{\frac{P-1}{q}} \, mod \, P$. Alice fixes also an integer a such that $1 \leq a \leq q-1$ and computes: $y = \alpha^a \, mod \, P$. She chooses an RSA exponent e prime with $\varphi(P-1) = (p-1)(p\prime - 1)(q-1)$ and calculates $d = \frac{1}{e} \, mod \, \varphi(P-1)$. Finally, Alice publishes (P, q, α, y, e) as his public keys and keeps(a, d) as his private keys. Parameters p and $p\prime$ will be destroyed for obvious security reasons.

4.2. **The signature generation.** To sign a message m, Alice selects a secrete random integer k, with: $1 \leq k \leq q$. Then, she computes $r\prime = (\alpha^k \, mod \, P) \, mod \, q$. And, she calculates $s\prime = \frac{(h(m)+a.r\prime)}{k} \, mod \, q$.
Now, using her private key d, she obtains: $r = r\prime^d mod(P-1)$ and $s = s\prime^d mod(P-1)$. Alice's signature is the pair (r, s).

4.3. **The signature verification.** To verify Alice's signature, Bob calculates: $U_1 = \frac{h(m)}{s^e \, mod \, (P-1)} \, mod \, q$ and $U_2 = \frac{r^e \, mod \, (P-1)}{s^e \, mod \, (P-1)} \, mod \, q$. Then, he finds: $V = ((\alpha^{U_1} y^{U_2}) \, mod \, P) \, mod \, q$.
Bob computes: $U = (r^e \, mod \, (P-1)) \, mod \, q$. He accepts Alice's signature if and only if $V = U$.

Example 4.1. Let's take $P = 47378687$, $p = 347$, $p\prime = 293$ and $q = 233$ four primes such that: $P = 2pp\prime q + 1$, $g = 5$ and $\alpha = g^{\frac{P-1}{q}} \, mod \, P = 16199304$. $a = 715$ so $y = \alpha^a \, mod \, P = 26663129$. We fix $e = 7$ prime with $\varphi(P-1)$ so $d = 20090935$. Assume that Alice wants to sign the message m that verifies: $h(m) = 421$. She selects $k = 150$ and computes $r\prime = (\alpha^k \, mod \, P) \, mod \, q = 18$. Then, she computes: $s\prime = \frac{h(m)+a.r\prime}{k} \, mod \, q = 202$.
Using the private key d, Alice calculates: $r = r\prime^d \, mod \, (P-1) = 1126454$ and

$s = s\prime^d \, mod\,(P-1) = 12249846$. Alice's signature for the message m is the pair (1126454,12249846).
Bob finds: $U_1 = \frac{h(m)}{s^e \, mod\,(P-1)} \, mod \;\; q = 24$, $U_2 = \frac{r^e \, mod\,(P-1)}{s^e \, mod\,(P-1)} \, mod \;\; q = 37$, $V = ((\alpha^{U_1} y^{U_2}) \, mod\, P) \, mod\, q = 18$ and $U = (r^e \, mod\,(P-1)) \;\, mod\, q = 18$. He accepts the signature since: $V = U$.

4.4. **Security analysis.** Before detailing possible attacks of our protocol, we start with the following theorem:

Theorem 4.2. *Breaking our work allows to destroy the DSA protocol.*

Proof. To generate a signature according to our method, the signer calculates parameters $r\prime$ and $s\prime$. Then, he finds r and s that satisfy the verification equation. However, if an attacker finds a way to sign in the place of Alice using our protocol, then he obtains parameters: r and s. In this case, the attacker can calculate $r\prime$ and $s\prime$ using equations: $r\prime = r^e \, mod\,(P-1)$ and $s\prime = s^e \, mod\,(P-1)$. Then, the DSA protocol is necessarily broken. So, our method is stronger then that of DSA. □

Corollary 4.3. *Our method is an alternative solution if the DSA scheme is broken.*

Proof. The protocol is reinforced by including the RSA algorithm. The DSA method is based on the difficulty of solving a discrete logarithm to find the secret key a that satisfies: $y = \alpha^a \, mod\, P$. The protocol we propose is also based on the problem of factoring large numbers that allows to solve the modular polynomial equations: $r\prime = r^e mod(P-1)$ and $s\prime = s^e mod(P-1)$ with: r and s are unknown and $P-1$ a large composite number. □

We discuss in the fallowing some possible attacks. So, assume that Oscar is an attacker.

Attack 1: Knowing Alice's public key y, Oscar can not find her private key a. Indeed, he is confronted to the hard discrete logarithm problem to solve the equation: $y = \alpha^a \, mod\, P$.

Attack 2: Using the public RSA exponent e of Alice, Oscar is not able to compute the decryption key d. He has to solve the equation: $d = \frac{1}{e} \, mod\, \varphi(P-1)$, such that: $P - 1 = pp\prime q$. As p and $p\prime$ are secrets, Oscar can not factor the number $P-1$. He is not able to compute $\varphi(P-1)$ to solve the above equation.

Attack 3: The DSA method is deduced from the ElGamal scheme. This confirms that signing two different messages m_1 and m_2 using the same signature key k is a great risk to destroy the system [10]. Indeed:
To sign messages m_1 and m_2, Alice computes simultaneously: $s_1 = \frac{h(m_1)+a.r}{k} modq$ and $s_2 = \frac{h(m2)+a.r}{k} \, mod\, q$. This implies that: $k = \frac{h(m_1)+a.r}{s_1} \, mod\, q$ and $k = \frac{h(m_2)+a.r}{s_2} \, mod\, q$. So, $a = \frac{s_1 h(m_2) - s_2 h(m_1)}{r(s_1 - s_2)} \, mod\, q$.
Our improvement to the DSA method ensures the security of the system also

when Alice chooses the same parameter k to sign two different message. As it's not enough for Oscar to find r and s that support conditions for verifying the validity of his signature but he's required to compute the e^{th} root of r and s.

Attack 4: We proved above that our signature method is stronger than the DSA scheme. Now, we show that if Oscar is able to break our protocol, then he destroys the RSA system. Indeed, To sign a message m, Alice determines $r\prime$ and $s\prime$. Then, she finds parameters r and s that verify: $r = r\prime^d \, mod \, (P-1)$ and $s = s\prime^d \, mod \, (P-1)$. Which requires destroying the RSA system.

4.5. **Complexity.** To generate her signature keys, Alice executes an exponentiation and five multiplications. In the signature step, she performs two multiplications and three exponentiations. And Bob calculates three multiplications and four exponentiations to verify Alice's signature.
Let T_{exp} and T_{mult} the times necessary to compute respectively an exponentiation and a multiplication. The total time to execute all operations using our approach is as follows:

$$T_{tot} = 8T_{exp} + 10T_{mult}$$

As: $T_{exp} = O((\log n)^3)$ and $T_{mult} = O((\log n)^2)$, (see [9] pp. 72). The final complexity of our signature protocol is:

$$T_{tot} = O((\log n)^2 + (\log n)^3) \tag{4.1}$$

This confirms that the method works in a polylogarithmic time.

5. Conclusion

In this work we presented a digital signature scheme. The method is based on the DSA protocol developed by using the RSA algorithm. The purpose of this combination is to improve the security of the DSA system.

Acknowledgements

This work is supported by the MMS e-orientation project.

References

[1] Adleman, L., (1979), A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography, *Proc. 20th IEEE Foundations of Computer Science Symposium*, pp 55−50.

[2] Boneh, D., Venkatesan, R. (1998). Breaking RSA may not be equivalent to factoring. *Advances in Cryptology — EUROCRYPT'98. Lecture Notes in Computer Science.* 1403. Springer, pp 59−71.

[3] J A. Buchmann (2001), Introduction to Cryptography. New York: Springer-Verlag.

[4] Chen-Yu, L. and Wei-Shen, L., (2008), Extended DSA, *Journal of Discrete Mathematical Sciences and Cryptography* 11:5, pp 545-550, DOI: 10.1080/09720529.2008.10698206.

[5] Diffie, W. and Hellman, M., (1976), New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22, 472−492.

[6] ElGamal, T. (1985), A public key cryptosystem and a signature scheme based on discrete logarithm problem, *IEEE Trans. Info. Theory , IT-31.*

[7] Horster, P., Michels, M., Petersen H., (1994) Generalized ElGamal signature schemes for one message block, *Technical Report*, TR-94-3.

[8] Khadir, O., (2010), New variant of ElGamal signature scheme, *Int. J. Contemp. Math. Sciences*Vol. 5, no. 34.

[9] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996) *Handbook of applied cryptography.*

[10] National institute of standard and technology NIST, (1994), FIPS Publication 186, DSA, Department of commerce.

[11] Pollard, J.M, (1974), Theorems on factorization and primality testing, *Proc. Camb. Phil. Soc. 76*, pp 521−528.

[12] Paillier, P., (1999), Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *Eurocrypt*, pp 223−238.

[13] Rabin, M.O., (1978), Digital signatures and public-key functions as intractable as factorization, *Technical Report MIT/LCS/TR−212*.

[14] Rivest, R., Shamir, A., & Adeleman, L. (1978), A method for obtaining digital signatures and public key cryptosystems, *Communication of the ACM*,Vol. no 21.

[15] Schnorr, C.P., (1991), Efficient Signature Generation by Smart Cards, *Journal of Cryptology* , pp 161−174.

[16] Shor & Peter (1997), Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal on Computing*, pp 1484−1509.

[17] Stinson D. R., (2006), Cryptography: Theory and practice, Third Edition, Discrete mathematics and its applications.

[18] Trappe W. and Washington L. (2005), Introduction to Cryptography with Coding Theory. Prentice Hall, 2nd edition.

Laboratory of Mathematics, Cryptography, Mechanics and Numerical Analysis (LMCMNA), university Hassan II of Casablanca, Morocco.

Email address: leila.zahhafi@gmail.com

Email address: khadir@hotmail.com