Int. J. Communications, Network and System Sciences, **2013, 6, 77-81**
http://dx.doi.org/10.4236/ijcns.2013.62009 Published Online February 2013 (http://www.scirp.org/journal/ijcns)

Faster Method for Secure Transmission of Information with Sender Identification

Boris S. Verkhovsky
Computer Science Department, New Jersey Institute of Technology, Newark, USA
Email: verb73@gmail.com

Received December 3, 2012; revised January 8, 2013; accepted January 24, 2013

ABSTRACT

This paper describes an algorithm for secure transmission of information via open communication channels based on the discrete logarithm problem. The proposed algorithm also provides sender identification (digital signature). It is twice as fast as the RSA algorithm and requires fifty percent fewer exponentiations than the ElGamal cryptosystems. In addition, the algorithm requires twice less bandwidth than the ElGamal algorithm. Numerical examples illustrate all steps of the proposed algorithm: system design (selection of private and public keys), encryption, transmission of information, decryption and information recovery.

Keywords: Digital Signature; RSA Algorithm; Diffie-Hellman Key Exchange; ElGamal Cryptosystem; Encryptor; Discrete Logarithm; Sender Identification; Multiplicative Inverse

1. Introduction

This paper describes a protocol for secure transmission of information that resembles the RSA algorithm [1]. However, the crypto-immunity of the proposed protocol is not based on computational complexity of integer factorization. Hardness of its cryptanalysis is based on the computational complexity of a discrete logarithm problem (DLP) [2,3] if the base g is a generator in modular arithmetic with prime modulus p. **Definition1.1:** A prime integer p is called a *safe* prime if

$$q := (p-1)/2 \tag{1.1}$$

is also a prime; and for every $p \geq 7$ q is *odd*.

Here are examples of safe primes: 44618543, 64542503, 171534179, 1111127819, 2176078679, 2382062063.

As it is demonstrated in [4], if p is a safe prime, then the computation of a generator g is a computationally fast procedure.

2. Private and Public Keys

The proposed protocol is based on parallel establishment of a secret encryptor [5] by a sender and receiver.

Proposition2.1: If p is a safe prime greater than or equal 7, then

$$g = p - 2^2 \tag{2.1}$$

is a generator for every p.

Indeed, the Fermat Little Theorem [2] and (1.1) imply that

$$g^q = \left(p-2^2\right)^q = (-1)^q\, 2^{p-1} = -1 \neq 1 \pmod p; \tag{2.2}$$

and

$$g^2 = \left(p-2^2\right)^2 = 2^4 \neq 1 \pmod p, \tag{2.3}$$

if $p \geq 7$.

*Remark*2.1: Observe that for every integer n

$$g^n = p - 2^{2n} \pmod p. \tag{2.4}$$

Integer parameters p and g are used by all participating users.

Alice selects her *private* key a and computes her *public* key

$$u = g^a = p - 2^{2a} \pmod p. \tag{2.5}$$

Analogously and independently, Bob selects his private key b and computes his public key

$$w = g^b = p - 2^{2b} \pmod p. \tag{2.6}$$

*Remark*2.2: both private keys must satisfy the inequality

$$\log_4 p < a, b < p-2; \tag{2.7}$$

otherwise the intruder will be able to deduce a from (2.5) and/or b from (2.6) without confronting the complexity of the DLP; in addition, the private keys a and b must be distinct from q.

Suppose that Bob sends a plaintext m {represented in a numeric form}, where $2 \leq m \leq p-2$.

3. Encryption via Exponentiation

***System design*:**

a) Each user computes his/her *common secret encryptor*

$$e := u^b = w^a \pmod{p}; \tag{3.1}$$

b) If e is distinct from 2, q and $2q$, *i.e.*, if

$$\gcd(e,d)=1; \tag{3.2}$$

then the users compute an integer d that satisfies the equation

$$ed \bmod q = 1; \tag{3.3}$$

*Remark*3.1: Although the users can find d (decryptor) from (3.4)

$$d := \mathrm{e}^{q-2} \bmod q; \tag{3.4}$$

there is a more efficient algorithm for modular multiplicative inverse (MMI) proposed by the author of this paper in [6] and analyzed in [7]; see Example 2 and **Table 1** below.

***Encryption*:**

c) The sender of message m computes the ciphertext

$$c := m^e \bmod p; \tag{3.5}$$

d) The ciphertext c is sent to a receiver via an open communication channel;

***Decryption*:**

e) The receiver computes

$$f := c^d \bmod p\,. \tag{3.6}$$

*Remark*3.2: Although (3.5) and (3.6) resemble the RSA protocol [1], there are two distinct features: the encryptor e is a *secret* (not public!) key and modulo reduction is done by the prime q which is a public key rather than by a product $n_l = p_l q_l$ of two large primes that are the private keys of the l-th user.

Proposition3.1: If m is a quadratic residue modulo p, then $f = m$ otherwise $f = p - m$.

Proof: Let us consider two outcomes:

- both e and d are *odd*;
- either e or d or both are *even*.

*Outcome*1: (3.3) and the FLT imply that there exists an *even* integer k such that

$$ed = 1 + qk; \tag{3.7}$$

then

Table 1. MMI of *e*=92 mod *p*=22309271.

***p*=22309271**	***e*=92**	7	1
Stack	242492	13	**
***d*=3152397**	13	**1**	**0**

$$c^d = m^{ed} = m \times \left[m^{(p-1)/2} \right]^k \pmod{p} = m\,. \tag{3.8}$$

*Outcome*2: in this case (3.3), the FLT and Euler criterion of quadratic residuosity imply that there exists an *odd* integer k such that $ed = 1 + qk$; then

$$c^d = m^{ed} = m \times \left(m^q \right)^k \pmod{p} = \pm m\,. \tag{3.9}$$

*Remark*3.3: If m is a quadratic residue modulo p, then $f = m$ for each outcome, otherwise in (3.9) $c^d = p - m$.

However, the verification of quadratic residuosity of every plaintext block m is a time-consuming process. There are two options to overcome this hurdle:

*Option*1: together with the ciphertext c the sender transmits a binary indicator R, *i.e.*, 0 or 1: if m is even, then he/she sends 0 else the sender transmits 1.

The receiver action: If $\mathrm{parity}(f) = R$, then $m := f$; else

$$m := p - f. \tag{3.10}$$

All cases of *Option*1 are summarized in **Table 2**:

*Option*2: The sender pre-conditions m and assigns $v := 2m$; and computes $c := v^e \bmod p$.

If $f = c^d \bmod p$ is even, then $m := f/2$ else

$$m := (p - f)/2\,. \tag{3.11}$$

4. Numeric Illustrations

***Example*1**: Let $p = 107; m = 46$; and suppose that the private keys a and b are selected in such a way that $e = 48$.

Let us find the decryptor d using the **MMI algorithm**:

assign $a_0 := q; a_1 := e;$

repeat

$$q_k := \lfloor a_{k-1}/a_k \rfloor;$$

{store all quotients q_k in a stack};

$$a_{k+1} := a_{k-1} - q_k a_k;$$

until $a_n = 0$; or $a_n = 1$;
if $a_n = 0$, **then** the MMI does not exist; **stop**;
if $a_n = 1$, **then** assign $b_n := 0; b_{n-1} := 1;$
for k **from** $n-1$ **down to** 1
iterate $b_{k-1} := q_k b_k + b_{k+1}$;
if n is *odd*, **then** $d := b_0$; **else** $d := q - b_0$ [8].

Therefore, from the MMI algorithm d=21.

Table 2. Cases for information recovery.

e, *d*	Information recovery
e and *d odd*	$f = m$
e or *d even*	$f = \pm m$; **if** $par(f) = R$ **then** $m := f$ **else** $m := p - f$

Indeed: $48 \times 21 \bmod 53 = 1$.

Encryption/decryption via Option1:

*Encryption*1:

$$c = m^e \bmod p = 46^{48} \bmod 107 = 99;$$
$$R := par(46) = 0;$$

the sender transmits $(c, R) = (99, 0)$ to the receiver

*Decryption*1:

$$f := c^d \bmod p = 99^{21} \bmod 107 = 61;$$

Since $par(f) \neq R$, then $m = p - f = 46$.

Encryption/decryption via Option2:

*Encryption*2: $v := 2\text{m} = 92; c = 92^{48} \bmod 107 = 27;$

*Decryption*2: $f := 27^{21} \bmod 107 = 92$;

Since f is even, then $m := f/2 = 46$.

***Example*2:** p=44618543; then q=22309271.

If a plaintext is divided into blocks of five characters each, and the size of an alphabet is 26, then

$$26^5 - 1 = 11881376 < p.$$

Suppose that the private keys a and b are selected in such a way that e=92. Therefore, from the MMI algorithm d=3152397 (see **Table 1**). Indeed:

$$ed = 92 \times 3152397 \bmod 22309271 = 1.$$

***Example*3:** Let $p = 9839; a = 1777, b = 1913;$ (private keys); therefore, the public keys are

$$u := p - 4^a = 2892; w := p - 4^b = 1649;$$

and the mutual secret encryptor for Alice and Bob:

$$e := u^b = w^a \pmod p = 1057.$$

Then both Alice and Bob solve independently $1057d \bmod 4919 = 1$ (3.3).

Table 3 demonstrates step-by-step how the MMI algorithm operates.

Since the number of columns in **Table 1** is even, then d=1680.

Indeed, $1057 \times 1680 \bmod 4919 = 1$.

Table 4 provides an array of seven plaintext blocks, shows their encryption and information recovery by the receiver. In this case, the sender transmits with each ciphertext a corresponding binary indicator R=0 if m is even; and R=1 if m is odd.

Table 3. MMI of *e*=1057 modulo *q*=4919.

4919	**1057**	691	366	325	41	38	3	2	1
Stack	4	1	1	1	7	1	12	1	*
1680	361	236	125	111	14	13	1	***1***	***0***

Table 4. Encryption and information recovery: *p*=9839; *e*=1057; *d*=1680.

m	1272	7871	4123	6802	9546	8325	6531
c	8374	4842	9197	9527	5204	4193	7202
f	8567	1968	4123	6802	293	8325	3308
m=f	**	**	4123	6802	**	8325	**
m=p-f	1272	7871	**	**	9546	**	6531
R	*R*=0	*R*=1	*R*=1	*R*=0	*R*=0	*R*=1	*R*=1

5. Complexity Analysis of EvESE Cryptosystem

On the system design level, each user performs two exponentiations to compute their public key (2.5) and (2.6), and the secret encryptor (3.1).

For the encryption, it is necessary to perform only one exponentiation (3.5). Analogously, for decryption, every receiver performs only one exponentiation (3.6). Although for the purpose of maintaining the high security level we need to periodically select new private keys and recompute the encryptor and decryptor, we do not need to send the hints with every block of the transmitted message as it is done in the ElGamal algorithm [9] (see **Table 5**).

Since the proposed algorithms (3.1)-(3.6) are based on computational complexity of the DLP, it has certain advantages over the RSA algorithm based on factorization. It is also more efficient than the ElGamal algorithm. Indeed, it needs twice fewer exponentiations for the secure transmission of each block than in the RSA algorithm with digital signature, and 1.5 fewer exponentiations for the secure transmission of each block than in ElGamal. In addition, the ElGamal algorithm requires twice as much bandwidth since together with the ciphertext it is necessary to send an ephemeral public key {the hint}

$$h := g^x \bmod p;$$

with every encrypted block m.

An idea of "binary" shift is proposed in [8] if e is an even integer: $e := e \pm 1$. However, even if the encryptor e is an odd integer, there is an additional advantage to find the decryptor d from the Equation (3.3).

Proposition5.1: Suppose that

$$eD \bmod (p-1) = 1; \tag{5.1}$$

and e is odd; then

$$q \mid (D - d). \tag{5.2}$$

Proof: Let

$$ed = 1 + qk; \text{ and } eD = 1 + 2qK; \tag{5.3}$$

then (5.3) implies that

$$e(D - d) = q(2K - k). \tag{5.4}$$

Since e and q are relatively prime, then (5.4) implies

that q divides $D-d$.

Therefore,

$$\text{either } d=D \text{ or } d=D-q. \tag{5.5}$$

Now, suppose that $D=d+qz$, where z is either 0 or 1; { $z<2$ since from (5.1) $D<2q$ }.

Hence, if $eD=1+2qK$;

then $e(d+qz)=1+2qK$ implies that

$$ed=1+q(2K-z). \tag{5.6}$$

Finally, from analysis of parities in (5.6) we deduce that if d is odd, then z=0; and, if d is even, then z=1.

Table 5 provides several examples of corresponding decryptors D and d. Since in many cases $\boldsymbol{d \ll D/2}$, therefore recovery of information with decryptor d is faster rather than with D.

Therefore, for $p=9839; D \geq d; q \mid (D-d)$ and on average $D/d=3.82$.

In addition, the encryptor and decryptor provide a digital signature (sender identification) since they are computed for communication between the specific pair of users (Alice and Bob).

6. Novelty Elements and Conclusion

Notice that the ElGamal algorithm is just one of several constructive ways to dynamically apply the Diffie-Hellman key establishment scheme for hiding information in secret communication. Indeed, both parties are dynamically establishing a common secret key (encryptor $e(m)$) and then its inverse value $d(m)$ (decryptor). In the ElGamal algorithm the sender conceals message m by multiplying it on the encryptor $e(m)$.

Other options: instead of multiplying, the sender *adds* the encryptor $e(m)$ to m or he/she uses exponentiation $m^e \bmod p$.

Although it seems that addition of e or even multiplication by e is computationally simpler than the exponentiation, the analysis shows the opposite (see **Table 6** below).

In the proposed EvESE cryptographic algorithm we use the following novelties:

a) a *safe* prime p is considered as the modulus (1.1);

b) a computationally *simple* and deterministic method is proposed to select the generator (primitive element) g for all users (2.1);

c) the encryptor e for secure communication between the sender and receiver is *private* (3.1);

d) the plaintext block m is concealed via the *exponentiation* (3.5) rather than by multiplication or any other binary operation;

e) a *deterministic* procedure based on the equation $ed \bmod q = 1$ (3.3) finds a mutual decryptor d for the communicating parties [6];

f) one of two options is applied for the information recovery: we either transmit a *binary* indicator R (3.1) or every plaintext block m is *pre-conditioned* (3.11) prior to its encryption;

g) even if encryptor e is an *odd* integer, the decryption with d (3.3) in many cases is faster than with D (see **Table 5** and (5.1)-(5.5)).

I express my deep appreciation to Dr. Roberto Rubino

Table 5. Corresponding *D* and *d*; *p*=9839.

e	5	9	11	43	333	4307	4567
D	5903	8745	7155	5491	8479	8769	545
d	984	3826	2236	572	3560	3850	545
D/d	6.00	2.29	3.20	9.60	2.38	2.28	1.00

Table 6. Comparison of ElGamal, RSA and EvESE {Alice sends signed *m* to Bob}.

	Private keys	Public keys	Encryption	Trans-mission	Decryption	Information Recovery	Digital Signature
ElGamal	$a, b,$ $x, y,$ $e(m)$ $d(m)$	p, g u, w $h(m)$	$h(m)=g^x$ $e(m)=w^x$ $c=me(m) \bmod p$	$\{c, h(m)\}$	$d(m)=h^{p-1-b}$ $f=cd \bmod p$	$m=f$	Requires ***three*** exponentiations
RSA	$p_k, q_k, d_k,$ $k=1,2,\cdots$	$e_k,$ $n_k=p_k q_k$	$c=\left(m^{d_a}\right)^{e_b}(\#)$	c	$f=\left(c^{d_b}\right)^{e_a}(\$)$	$m=f$	Requires ***four*** exponentiations
EvESE	$a, b,$ e, d	p, g u, w	$c=m^e \bmod p$	$\{c, R\}$	$f=c^d \bmod p$	$m=f$ if $par(f)=\mathrm{R}$ else $m=p-f$	Requires ***two*** exponentiations

Legends: In (#) $c=\left(m^{d_a} \bmod n_a\right)^{e_b} \bmod n_b$; in ($) $f=\left(c^{d_b} \bmod n_b\right)^{e_a} \bmod n_a$. The RSA algorithm with digital signature works for *every m* only if $2 \leq m \leq n_a \leq n_b$; [10,11].

for his comments that improved the style of this paper.

REFERENCES

[1] R. L. Rivest, A. Shamir and L. M. Adleman, "Cryptographic Communications System and Method", US Patent #4405829, 1983.

[2] C. F. Gauss, "Disquisitiones Arithmeticae", 2nd Edition, Springer, New York, 1986.

[3] P. Garrett, "Making, Braking Codes: An Introduction to Cryptology", Prentice Hall, Upper Saddle River, 2001.

[4] B. Verkhovsky, "Deterministic Algorithm Computing All Generators: Application in Cryptographic Systems Design", *International Journal of Communications, Network and System Sciences*, Vol. 5, No. 11, 2012, pp. 715-719. doi:10.4236/ijcns.2012.511074

[5] W. Diffie and M. E. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 1976, pp. 644-654. doi:10.1109/TIT.1976.1055638

[6] B. Verkhovsky, "Multiplicative Inverse Algorithm and Its Space Complexity", *Annals of European Academy of Sciences*, 2004, pp. 110-124.

[7] B. Verkhovsky, "Space Complexity of Algorithm for Modular Multiplicative Inverse", *International Journal of Communications, Network and System Sciences*, Vol. 4, No. 6, 2011, pp. 357-363. doi:10.4236/ijcns.2011.46041

[8] B. Verkhovsky, "Public-Key Cryptosystems with Secret Encryptor and Digital Identification", *International Journal of Communications, Network and System Sciences*, Vol. 6, No. 1, 2013, pp. 1-6.

[9] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *Proceedings of CRYPTO* 84 *on Advances in Cryptology*, Springer-Verlag New York, Inc., New York, 1985, pp. 10-18.

[10] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography", CRC Press, Boca Raton, 1997.

[11] G. I. Davida, "Chosen Signature Cryptanalysis of the RSA (MIT) Public Key Cryptosystem", Technical Report TR-CS-82-2, University of Wisconsin, Milwaukee, 1982.

