Int. J. Communications, Network and System Sciences, **2011, 4, 77-81**
doi:10.4236/ijcns.2011.42009 Published Online February 2011 (http://www.SciRP.org/journal/ijcns)

Scheme for Secure Communication via Information Hiding Based on Key Exchange and Decomposition Protocols

Boris S. Verkhovsky
Computer Science Department, New Jersey Institute of Technology, University Heights, Newark, USA
E-mail: verb@njit.edu
Received January 26, 2011; *revised February* 8, 2011; *accepted February* 9, 2011

Abstract

This paper considers a decomposition framework as a mechanism for information hiding for secure communication via open network channels. Two varieties of this framework are provided: one is based on Gaussian arithmetic with complex modulus and another on an elliptic curve modular equation. The proposed algorithm is illustrated in a numerical example.

Keywords: Complex Modulus, Cryptanalytic Protection, Decomposition, Gaussian Modular Arithmetic, Information Hiding, Key Exchange, Modular Elliptic Curve, Secure Communication

1. Introduction and Problem Definition

In this paper it is demonstrated how to use various Diffie-Hellman key establishment (DHKE) protocols in order to design a computationally efficient cryptographic schemes for secure communication between two parties {called Alice and Bob}. One of these key establishment protocols is based on modular elliptic curves (ECDHKE) [1]. Another DHKE protocol is based on arithmetic of complex integers with complex modulus [2].

***DHKE protocol based on complex integers*:** In this scheme both parties agree to select a Gaussian integer $L = (g, h) := g + ih$ called generator and a complex modulus $(l, p) := l + ip$ with real integer components l and p. Alice and Bob independently select secret real integers *alpha* and *beta* respectively. Alice and Bob respectively compute their public keys:

$$E := L^{alpha} \bmod (l, p); \tag{1}$$

and

$$F := L^{beta} \bmod (l, p). \tag{2}$$

Then Alice and Bob compute respectively

$$H_1 := F^{alpha} \bmod (l, p) \tag{3}$$

and

$$H_2 := E^{beta} \bmod (l, p). \tag{4}$$

As a result,

$$M = H_1 = H_2 = (x_0, y_0) := x_0 + iy_0. \tag{5}$$

Therefore a pair of real integers x_0 and y_0 can be used by Alice and Bob to design a cryptographic protocol.

***DHKE based on elliptic curves*:** Consider a modular elliptic curve (EC)

$$y^2 + ey = x^3 + ax + b \pmod p \tag{6}$$

where *a*, *b*, *e* and *p* are integer parameters of the EC, and modulus *p* is an odd real prime [3]. If (6) is used for ECDHKE between Alice and Bob, then both parties create a mutual secret key (x_0, y_0) that is a point on (6). The scheme is analogous to (1)-(5): Alice and Bob select a point *Q* with high order on (6) and real integers *alpha* (Alice's secret key) and *beta* (Bob's secret key). Then they respectively compute their public keys:

$$E := alpha \times Q \bmod p; \tag{7}$$

and

$$F := beta \times Q \bmod p. \tag{8}$$

Here both *E* and *F* are points on (6).

Then Alice and Bob compute respectively

$$J_1 := alpha \times F \bmod p; \tag{9}$$

and

$$J_2 := beta \times E \bmod p. \tag{10}$$

As a result,

$$M := J_1 = J_2 := (x_0, y_0). \tag{11}$$

2. Decomposition

Consider randomly selected non-zero integers $A \neq 1$; $B \neq 1; C \neq 1$ that are co-prime with *p*. Consider positive integers *k*, *q* and *r* that satisfy

$$k + q + r = 6; 1 \leq k, q, r \leq 4 . \quad (12)$$

Compute $R := y_0^k A \bmod p$;

$$S := x_0^q B \bmod p; \ T := y_0^r C \bmod p \quad (13)$$

or $R := x_0^{k_i} A$; $S := y_0^{q_i} B$; $T := x_0^{3-k_i-q_i} C(\bmod p)$;
{for details see **Step5** below};
Select integers *u*, *v* and *w* that satisfy

$$u + v + w = R; \quad u + v - w = S; \quad u - v + w = T . \quad (14)$$

Then (14) implies that

$$u = (S + T)(p + 1)/2 \bmod p; \quad (15)$$

$$v = (R - T)(p + 1)/2 \bmod p; \quad (16)$$

$$w = (R - u - v) \bmod p. \quad (17)$$

Here *k* and *q* are secret keys that satisfy

$$1 \leq k \leq 4; \ 1 \leq q \leq 5 - k; \ r = 6 - k - q; \quad (18)$$

where *k* and *q* are periodically updated.

There are ten combinations of positive integers satisfying (12); these combinations are listed in lexicographically increasing order in **Table 1**.

3. Numeric Illustration

Let *p*=99991; consider an elliptic curve

$$y^2 + 1001y = x^3 + 217(\bmod 99991) . \quad (19)$$

Suppose Alice and Bob have established a secret key for communication as point $M = (x_0, y_0)$ = (86275, 81549); it is easy to verify that *P* indeed satisfies (19). Juxtapose (x_0, y_0) and let *G* := 8627581549.

4. Information Hiding Protocol—{*k*, *q*}

Step1: Communicating parties (Alice and Bob) establish a *key M* = (x_0, y_0) using one of schemes listed in section 1;

Table 1. All combinations of exponents.

States	0	1	2	3	4	5	6	7	8	9
k	1	1	1	1	2	2	2	3	3	4
q	1	2	3	4	1	2	3	1	2	1
r	4	3	2	1	3	2	1	2	1	1

Step2: Juxtapose coordinates (x_0, y_0); let

$$G = d_1 d_2 \cdots d_{t-1} d_t , \quad (20)$$

where d_i are its decimal digits.

Here

$$t \leq 2 \times \lceil \log_{10} p \rceil ;$$

Step3: Suppose Alice wants to transmit a plaintext array

$$m = \left\{ m_1^{(1)}, \cdots, m_5^{(1)}; \cdots; m_1^{(i)}, \cdots, m_5^{(i)}; \cdots; m_1^{(s)}, \cdots, m_5^{(s)} \right\}$$

where

$$s = \lceil t/5 \rceil . \quad (21)$$

Encryption of $m^{(i)} = \left\{ m_1^{(i)}, m_2^{(i)}, \cdots, m_5^{(i)} \right\}$:

Step4: Using d_i select corresponding $\{k_i, q_i, r_i\}$ from **Table 1**;

Step5: if d_i is even, then compute $R := x_0^{k_i} A$; $S := y_0^{q_i} B; T := x_0^{6-k_i-q_i} C(\bmod p)$;

else

$$R := y_0^{k_i} A;$$

$$S := x_0^{q_i} B; T := y_0^{6-k_i-q_i} C(\bmod p); \quad (22)$$

Step6: Compute the information hiding keys (15)-(17): {*u*, *v*, *w*};

Step7 {enhancement of crypto-immunity}:

for $z \in \{u, v, w, x_0, y_0\}$ **do**

if $z < \sqrt{p}$, then $z := p - 2z$ **else** *z* := 2*z*;

Step8: compute

$$c_1^{(i)} := m_1^{(i)} u; c_2^{(i)} := m_2^{(i)} v; c_3^{(i)} := m_3^{(i)} w(\bmod p); \quad (23)$$

$$c_4^{(i)} := m_4^{(i)} x_0; c_5^{(i)} := m_5^{(i)} y_0 (\bmod p). \quad (24)$$

Decryption is performed in reverse.

*Remark*2: After *t* cycles Alice and Bob must use a DHKE to establish a new mutual secret key *G*.

Choice of A, B and C: one way to choose *A* and *B* is to assign *even* digits of *G* to *A* and *odd* digits of it to *B*. Then select *C* that is a multiplicative inverse of *AB* modulo *p*:

$$C := (AB)^{-1} \bmod p \quad [4]. \quad (25)$$

*Remark*3: The ideas of decomposition can be applied to any secret key; where splitting is completely independent of how this key is established.

5. Plaintext Pre-conditioning

If there is a pair $\{c_j^i, c_l^i\}$, where both components are smaller than p, then with high probability holds that $\gcd\left(c_j^i, c_l^i\right) > 1$. Therefore either

$$\gcd\left(c_j^i, c_l^i\right) = x_0 \quad \text{or} \quad \gcd\left(c_j^i, c_l^i\right) = y_0. \qquad (26)$$

To preclude this possibility consider the following protocol of plaintext pre-conditioning: subdivide plaintext m into arrays of blocks in such a way that for every block m_r holds $m_r \le (p-2)/2$; if $m_r \le \sqrt{p}$, then assign

$$m_r := p - 2m_r \qquad (27)$$

*Remark*4: Notice that the right-most binary digit of m_r equals 1.

6. Numeric Illustration Continued

Assign all *even* digits of G := 8627581549 to A and all *odd* digits of G to B.

Then A = 67859, {in **Table 2** they are shown in **bold** font}, B = 82514, and $C = (AB)^{-1} \bmod p$ = 87964 [3].

Indeed: (21508*87964) mod99991 = 1.

Since G does not have digits 0 and 3, then only *eight* of ten combinations of {k, q, r} that are listed in **Table 1** are used to compute the information hiders u, v, w:

Computation of encryptors u, v, w

For $d_1 = 8$ {see the 1st column in **Table 2**} compute

$$R := x_0^{k_i} A = 86275^3 \times 67859 \bmod p;$$

$$S := y_0^{q_i} B = 81549^2 \times 82514 \bmod p;$$

$$T := x_0^{3-k_i-q_i} C = x_0^1 C \bmod p;$$

and then compute encryptors u, v and w (15)-(17).

For $d_2 = 6$ {see the 2nd column in **Table 2**} compute

$$R := x_0^{k_i} A = 86275^2 \times 67859 \bmod p;$$

$$S := y_0^{q_i} B = 81549^3 \times 82514 \bmod p;$$

$$T := x_0^{3-k_i-q_i} C = x_0^1 \times 87964 \bmod p;$$

and then compute encryptors u, v and w.

See **Table 3** of all encryptors for $i = 1, 2, \cdots t$.

Plaintext pre-conditioning

Let $m^{(1)} = \left\{m_1^{(1)}, m_2^{(1)}, m_3^{(1)}, m_4^{(1)}, m_5^{(1)}\right\}$

= {266, 45769, 37585, 36488, 46572}.

*Remark*5: Notice that each component in m is smaller than $(p-1)/2$.

Because $m_1 < \sqrt{p}$, reassign $m_1 := p - 2m_1$; {*odd* integer}.

Since all other blocks in plaintext m are greater than $\sqrt{p}$, therefore reassign

$$m_2 := 2m_2;\ m_3 := 2m_3;$$
$$m_4 := 2m_4;\ m_5 := 2m_5$$

{all four are *even* integers}.

Encryption {see Step7}:

$c_1 := m_1 u \bmod p; c_2 := m_2 v \bmod p;$

$c_3 := m_3 w \bmod p;$

$c_4 := m_4 x_0 \bmod p;\ c_5 := m_5 y_0 \bmod p.$

Alice sends ciphertext $\{c_1, .., c_5\}$ to Bob via open communication channels. See **Table 3** with encryptors R, S, T, u, v *and* w; **Table 4** with plaintext arrays $m_j^{(i)}$ and **Table 5** of corresponding ciphertext arrays.

Table 2. Sequence of exponents *k*, *q* and *r* based on secret key (x_0, y_0).

States	8	**6**	2	**7**	5	**8**	1	**5**	4	**9**
k	3	2	1	3	2	3	1	2	2	4
q	2	3	3	1	2	2	2	2	1	1
r	1	1	2	2	2	1	3	2	3	1

Table 3. Encryption stage: information hiders *u, v, w* and ciphertexts.

d_i	$d_1 = 8$	$d_2 = 6$	$\cdots$	$d_t = 9$
R	02480	30939	$\cdots$	21751
S	86137	18463	$\cdots$	36105
T	77173	77173	$\cdots$	21896
u	81655	47818	$\cdots$	78996
v	12649	15594	$\cdots$	49923
w	08167	67518	$\cdots$	92814

Table 4. Plaintext arrays $m_j^{(i)}$; *i* = 1,2,$\cdots$,*t*.

i	1	2	$\cdots$	*t*
$m_1^{(i)}$	00266	08764	$\cdots$	38643
$m_2^{(i)}$	45769	43654	$\cdots$	00179
$m_3^{(i)}$	37585	34631	$\cdots$	07320
$m_4^{(i)}$	36488	45731	$\cdots$	34219
$m_5^{(i)}$	46572	00301	$\cdots$	04352

Table 5. Ciphertext arrays $c_j^{(i)}$; *i* = 1,2,$\cdots$,*t*.

i	1	2	$\cdots$	*t*
$c_1^{(i)}$	70985	29342	$\cdots$	34378
$c_2^{(i)}$	68373	03496	$\cdots$	25955
$c_3^{(i)}$	68641	52628	$\cdots$	19261
$c_4^{(i)}$	71085	94285	$\cdots$	19900
$c_5^{(i)}$	83732	03083	$\cdots$	66378

Decryption is performed in reverse: since Bob knows the mutual secret key $M = (x_0, y_0)$, he finds *A*, *B*, *k*, *q* and *r*; then computes *C*, *R*, *S*, *T;* and the multiplicative inverse values of *u*, *v* and *w***.**

7. Key Establishment Based on Gaussian Modulus

Consider (*l*, *p*) = (1000, 3001); and a generator *L* = (2269, –2204). All corresponding steps and actions by Alice and Bob are provided in **Table 6**.

Therefore, *M* = (–0502, 1853) is the mutual secret key established between Alice and Bob. Notice that components in *M* can be positive and negative. If a component is negative, post digit "2" in front of its left-most digit; if the component is positive, post digit "9" in front of its left-most digit. Therefore *M* := (20502, 91853). For large *l* and *p* in modulus (*l*, *p*), the probability is negligibly small that either $x_0 = 0$ or $y_0 = 0$.

8. Computational Complexity

For every digit in juxtaposed *G* it is possible to encrypt one plaintext array.

With high probability each component in (x_0, y_0) has the same number of digits *t* as modulus *p*. Therefore in *G* there are about 2*t* digits. For each digit we select an appropriate combination of keys {*k*, *q*, *r*} from **Table 1** and encrypt five blocks of the plaintext. Therefore for every *G* we can encrypt $N(p) = 5 \times 2t = 10t = 10\lceil \log_{10} p \rceil$ blocks.

In application, to assure strong crypto-immunity, $t = 2\lceil \log_{10} p \rceil \in [100, 200]$.

Thus, if *p* is a 50-digit long integer, then $N(p) = 10 \times 50 = 500$ blocks of plaintext.

9. Reduction of Complexity

To reduce computational complexity of encryption for every *G*, we pre-compute and store for every $f = 1, \cdots, 4$ $Dx_0^f \bmod p$ and $Dy_0^f \bmod p$, where $D \in \{A, B, C\}$.

Another way to reduce complexity is to avoid computation of multiplicative inverse *C* (24). Instead we can partition *G* onto about equal number of digits. For instance, if *G* = <u>27182</u>**818284**<u>59045</u>, we can either assign A_1 := <u>27182</u>; B_1 := **818284** and C_1 := 59045 or to substitute digits $1, 2, \cdots, 9, 1, 2, \cdots$ into *G*:

A_2 := **1**7**2**8**3**8**4**8**5**8**6**5**7**0**8**5;

B_2 := 2**1**1**2**2**3**1**4**2**5**4**6**9**7**4**8**;

C_2 := 27**1**8**2**8**3**8**4**8**5**5**6**0**7**5.

Table 6. Key establishment (1)-(5).

Keys	Alice's action	Bob's action
Secret	*Alpha* = 1913	*Beta* = 1999
Public	*E* = (–846,1022)	*F* = (439,2876)
Private key	$H_1 = F^{alpha} = (-0502, 1853)$	$H_2 = E^{beta} = (-0502, 1823)$

10. Conclusion

In the proposed cryptocol it is shown that for every pair of integers in secret key (x_0, y_0) there are numerous ways to compute integers {*u*, *v*, *w*} that hide information on the encryption stage.

11. Acknowledgements

I express my appreciation to NJIT students A. Koripella and M. Sikorski for their assistance and suggestions that improved style of this paper, to D. Kanevsky for his participation in discussion, and to C. Pomerance and H. Cohen for their advices about complex modulus reduction.

12. References

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

[2] S. G. Krantz, "Modulus of a Complex Number," *Handbook of Complex Variables*, Birkhäuser Publishing Ltd., Boston, 1999, pp. 2-3.

[3] J. Hoffstein, J. Pipher and J. Silverman, "An Introduction to Mathematical Cryptography," Springer, New York, 2008.

[4] B. Verkhovsky, "Enhanced Euclid Algorithm for Modular Multiplicative Inverse and Its Application in Cryptographic Protocols," *International Journal of Communications, Network and System Sciences*, Vol. 3, No.12, 2010, pp. 901-906. doi:10.4236/ijcns.2010.312123

Appendix

A1. Generalization

Step1A: Establish a secret key M between communicating parties and juxtapose it.

*Remark*6: Either Gaussian arithmetic with complex modulus or other mechanisms for DHKE can be used to establish M.

Step2A: Using M, the parties select $A_1, \cdots, A_s$, where s is an integer parameter of encryption protocol;

Step3A: for i=1, $\cdots$,t

for j=1, $\cdots$,s **do**

if $$d_i = j \pmod 2 \quad \text{(A1)}$$

then $$R_j := A_j x_0^{k_j} \bmod p, \quad \text{(A2)}$$

else $$R_j := A_j y_0^{k_j} \bmod p; \quad \text{(A3)}$$

Step4A: Compute for j=2, 3, $\cdots$,s

$$u_j := \left[\left(R_1 - R_j\right)/2\right] \bmod p; \quad \text{(A4)}$$

and $$u_1 = \left[R_1 - \left(u_2 + .. + u_s\right)\right] \bmod p; \quad \text{(A5)}$$

Step5A {encryption}: **for** i **from** 1 to s

$$c_i := m_i u_i \bmod p. \quad \text{(A6)}$$

A2. Selection of Table for $k_1, k_2, \cdots k_s$

If $s > 3$, the number of possible combinations for secret keys $k_1, k_2, \cdots k_s$ grows exponentially if s is increasing.

This is an additional potential for randomization. If a protocol designer of encryption/decryption scheme represents G in a numeric form with base n, then it is possible to select n combinations of secret exponents $k_1, k_2, \cdots k_s$, where each combination corresponds to every digit of G. Therefore the parties must exchange between themselves a $n \times n$ square matrix:

$$\begin{pmatrix} k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{n1} & \cdots & k_{nn} \end{pmatrix} \quad \text{(A7)}$$

For example, if s = 4 and n = 16, then we need to specify *sixteen* combinations of $k_1, k_2, ..k_s$ and k_4.

If $1 \le k_i \le 5$ and $k_1 + \cdots + k_4 = d$, then the number of possible combinations of k's is 35 for d = 8.

