Journal of Advances in Mathematics and Computer Science

25(4): 1-10, 2017; Article no.JAMCS.37301
ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Triple-Protocol – A New Direction of Elliptic-Curve Cryptography

Vladyslav Baytalskyy[1*]

[1]Independent Researcher, Engineer in Organization of Information Protection with Restricted Access, Odessa, Ukraine.

Author's contribution

The sole author designed, analyzed and interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/JAMCS/2017/37301
Editor(s):
(1) Francisco Welington de Sousa Lima, Professor, Dietrich Stauffer Laboratory for Computational Physics, Departamento de Física, Universidade Federal do Piaui, Teresina, Brazil.
Reviewers:
(1) G. Y. Sheu, Chang-Jung Christian University, Taiwan.
(2) Anand Nayyar, KCL Institute of Management and Technology, India.
(3) S. AKhila, BMS College of Engineering, India.
Complete Peer review History: http://www.sciencedomain.org/review-history/21974

Original Research Article

Received: 10th October 2017
Accepted: 14th November 2017
Published: 18th November 2017

Abstract

The original triple-protocol is proposed, which is a modification of the well-known Massey - Omura protocol, but greatly improves it from the standpoint of efficiency and enables data to be validated for integrity and authenticity without the use of hash functions. This is ensured by forwarding through open channels to an external ("cloud") information carrier (and also directly to a participant in the information exchange) of some data obtained from mathematical and logical transformations above the original message. The combined use of modulo exponentiation and the XOR logical operation ("exclusive OR") ensures that unauthorized access for an intruder is impossible, bypassing the difficult mathematical problem of discrete logarithm (ECDLP), on which all the algorithms for data transfer on elliptical curves are based.

Keywords: ECDLP; key; cryptographic; system; triple-protocol; ECC.

1 Introduction

For a successful understanding of the material presented in the article, it is necessary to say a few words about the properties of the unidirectional function and the device of asymmetric encryption protocols.

**Corresponding author: E-mail: a905954@gmail.com;*

A unidirectional function is a mathematical function whose calculation in the forward direction ($f(x) \rightarrow Y$) does not present any particular problems, while the inverse procedure ($Y \rightarrow f(x)$) is extremely difficult or impossible at all. In its pure form, unidirectional functions are not applied due to the lack of a mechanism for decrypting the encrypted message, so in practice one-way functions with a secret move are used, providing easy retrieval of the inverse function only for the owner of the secret.

Such a formula was proposed in 1976 by the researchers Diffie and Hellman [1] to use an asymmetric key distribution protocol: two keys are used - one, freely available, for encrypting the message; and another - private, for decryption, and these two keys should not coincide and there is no way to select one key from another.

The advantage of using encryption on the mathematical properties of elliptic curves is that for today, this is how you can get the highest value of "complexity / bit" (for more details, see [2]). Information on the requirements for curves and the like can be obtained from [3-7], and a detailed explanation of the Massey - Omura protocol, which acts as the basis for the proposed triple protocol, is in [8].

But the main drawback of traditional asymmetric encryption systems (as well as their modifications [8]) is the absence of their own independent mechanisms for verifying the integrity and authenticity of users. In case of a real attack (using advanced hardware and software), the attacker will look for the weakest point in the data transfer protocol, and the use of standard hash functions can simplify the task of decoding the message.

Now let's look briefly at how the triple-protocol is arranged.

Subscriber *A* wants to send a certain confidential message *M* to the subscriber, using *unprotected* channels.

To achieve the result, three separate data groups are used, each of which performs a certain function:

- The first flow is the transfer of the coding mask (Massey – Omura protocol is used);
- The second is the encrypted message *C*;
- Third – some verification number *M_A*, serving to confirm the integrity of the original message *M* and confirm the claimed data sender.

Streams 2 and 3 are duplicated by sending to an independent network resource to reduce the probability of change and incorrect (including random) transmission of the original message. Protocol is called triple ("triple"), since there is a mixing of the three transmitted numbers (the first stream), which complicates the decryption. It is recommended to send a composite message (more than six elements), where by using already transmitted, but unprocessed information, it is possible to reduce the number of auxiliary bits and, in addition, to level out the impact of collisions (when trying to completely forge all traffic). The transmitted secret message can either be directly used or interact with traditional symmetric systems in the form of a key.

Example. Suppose it is necessary to encode an important information channel that does not allow long operations for encryption-decryption. Then you can use this algorithm:

a) Using the triple-protocol, the first master key of large dimension K_{512} is transmitted (for example, the value (5 * 512 * 512 * 512 = 671088640) bit – based on the 512-bit key);
b) Every (8 * 64 = 512) bits of the transmitted encoded message is a change of the generating point (XOR-addition), which makes it hard to decrypt;
c) There is a transition to 64-bit encryption of K_{64} (real-time encoding - audio and video information and the like) according to the requirements of the triple-protocol, but the openly transferred values of R_{64} and C_{64} are a serial XOR-addition with sequential values from the master key (its elements are 64 bits long), which significantly complicates the decryption (it requires hacking both 64-bit key and 512-bit key, which is technically difficult);
d) After exhausting the possibilities for encoding the first master key, a transition to the second master takes place (it is transmitted via the second sub-channel of information during the execution of point a)), and so on.

2 Basic Provisions of the Triple-protocol

The data exchange uses the same elliptical curve, the parameters of which are known to a wide range of possible participants in the information process.

Subscriber *A* before the beginning of an information exchange performs the following operations:

a) Selects from a pre-ordered list or accidentally generates an outgoing message *mM_A*. In the case of arbitrary generation, the code number is chosen by the interval [$\sqrt{p}$ ... *p*], where p is the module of the equation of the elliptic curve of the form

$y^2 = x^3 + ax + b \pmod p$,

where $a, b, \in GF(p)$, $4a^3 + 27b^2 (\mathrm{mod}\ p) \neq 0$, $p > 3$ – prime number;

b) Randomly selects the numbers r_1, r_2, r_3, where r_i belongs to [$\sqrt{p}$ ... p], and calculates the scalar product

$R_i = r_i[G]$,

where *G* is a generating point with parameters $G = (g\_x, g\_y)$;

c) Generates (the operations of generating and finding the returning elements fills up and subscriber b) three (if it is necessary to send a composite message, then more than three) of arbitrary numbers A_1, A_2 and A_3 within [3 ... *n*], where n is the order *n* elliptic curve (i.e., $nG = 0$) and calculates the reciprocal elements *A*_1_1, *A*_1_2, *A*_1_3 modulo *p*; for *p* - we have a simple number

$A\_1\_i = (A_i)^{p-2} \bmod p$;

d) Calculates the scalar product

$R_i = A_i[G]$,

and divides it into three streams (see below for details on specific protocols)

$R_1 = (d\_1\_x, d\_1\_y)$,
$R_2 = (d\_2\_x, d\_2\_y)$,
$R_3 = (d\_3\_x, d\_3\_y)$;

e) Finds intermediate results:

$d\_x = d\_1\_x + d\_3\_x \pmod p$,
$d\_y = d\_2\_y + d\_3\_y \pmod p$,

$C\_m = mM\_A \wedge d\_x \wedge d\_y$,

$mH = (d\_3\_x \wedge d\_2\_y)^{(d\_1\_x \wedge d\_3\_y)} \pmod p$,

$mH\_M = (mM\_A \wedge d\_1\_x)^{(d\_3\_x \wedge d\_3\_y \wedge d\_2\_y)} \pmod p$,

where the sign ^ *denotes* the logical operation of the *HOR*;

f) Publish in the wide access (say, in any cloud storage) at a certain time (one hour, two, or a day, depending on the secrecy requirements) the numbers *mH* and *mH_M*, which perform two functions:

1) Confirm the veracity of the addressee;
2) Greatly complicate the falsification of messages.

According to the Massey - Omura protocol, subscriber *A* and subscriber *B* exchange information in the following way:

a) In rounds:

1) The first round $A \rightarrow B$: $Y_{ai} = r_i[G]$,
2) The second round $A \leftarrow B$: $Y_{bi} = b_i[Y_i]$,
3) The third round $A \rightarrow B$: C_m, $(C\_ab)_i = A\{A\_1\_i[Y_{bi}]\}$.

Subscriber *B* calculates $(rG\_B)_i = B\_1\_i\ [(C\_ab)_i]$.

b) The purpose of the information exchange are:

1) From the point d_1 (d_1_x, d_1_y) have the number d_1_x;
2) From point d_2 (d_2_x, d_2_y) takes the value d_2_y;
3) The point d_3 (d_3_x, d_3_y) is used as the basis for calculating:

$d\_x = d\_1\_x + d\_3\_x \pmod p$,
$d\_y = d\_2\_y + d\_3\_y \pmod p$.

Subscriber *B*, having calculated *mM_B* from the formula $mM\_B = C\_m \wedge d\_y \wedge d\_x$, verifies the truth of the relations:

$$mH\_B = (d\_3\_x \wedge d\_2\_y)^{\,d\_1\_x \wedge d\_3\_y} \pmod p,$$

$$mM\_B = C\_m \wedge d\_y \wedge d\_x,$$

$$mH\_M\_B = (mM\_B \wedge d\_1\_x)^{\,d\_3\_x \wedge d\_3\_y \wedge d\_2\_y} \pmod p,$$

and if there is a match

$$mH\_B = mH\_A;\ mH\_M\_B = mH\_M\_A,$$

it concludes that

$$mM\_B = mM\_A,$$

and the information was received without errors and precisely from subscriber *A*.

If a consecutive message of a compiled message occurs (suppose ASCII text), the penultimate and last value of d_2_y and d_3 are taken from the beginning of the message (see more details in the section on sending and receiving constituents messages). Calculated formulas for cryptographic triple-protocol ($\oplus$ = XOR):

$$d\_x = d\_1\_x + d\_3\_x \pmod p,$$

$$d\_y = d\_2\_y + d\_3\_y \pmod p,$$

$$R_1 = (d\_1\_x, d\_1\_y);\ R_2 = (d\_2\_x, d\_2\_y);\ R_3 = (d\_3\_x, d\_3\_y),$$

$$C\_m = f(mM\_A, d\_y, d\_x) = mM\_A \oplus d\_x \oplus d\_y,$$

$$mH\_A = f(R_1, R_2, R_3) = (d\_3\_x \oplus d\_2\_y)^{(d\_1\_x \oplus d\_3\_y)} \pmod p,$$

$$mH\_M\_A = f(mM\_A, R_1, R_2, R_3) = (mM\_A \oplus d\_1\_x)^{(d\_3\_x \oplus d\_3\_y \oplus d\_2\_y)} \pmod p,$$

$$mH\_B = f(R_1, R_2, R_3) = (d\_3\_x \oplus d\_2\_y)^{(d\_1\_x \oplus d\_3\_y)} \pmod p,$$

$$mM\_B = f(C\_m, d\_y, d\_x) = C\_m \oplus d\_x \oplus d\_y,$$

$$mH\_M\_B = (mM\_B \oplus d\_1\_x)^{(d\_3\_x \oplus d\_3\_y \oplus d\_2\_y)} \pmod p.$$

Successful transmission from subscriber *A* to subscriber *B* (naturally, conditions are checked by B):

$mH\_B = mH\_A$; $mH\_M\_B = mH\_M\_A$; $mM\_B = mM\_A$.

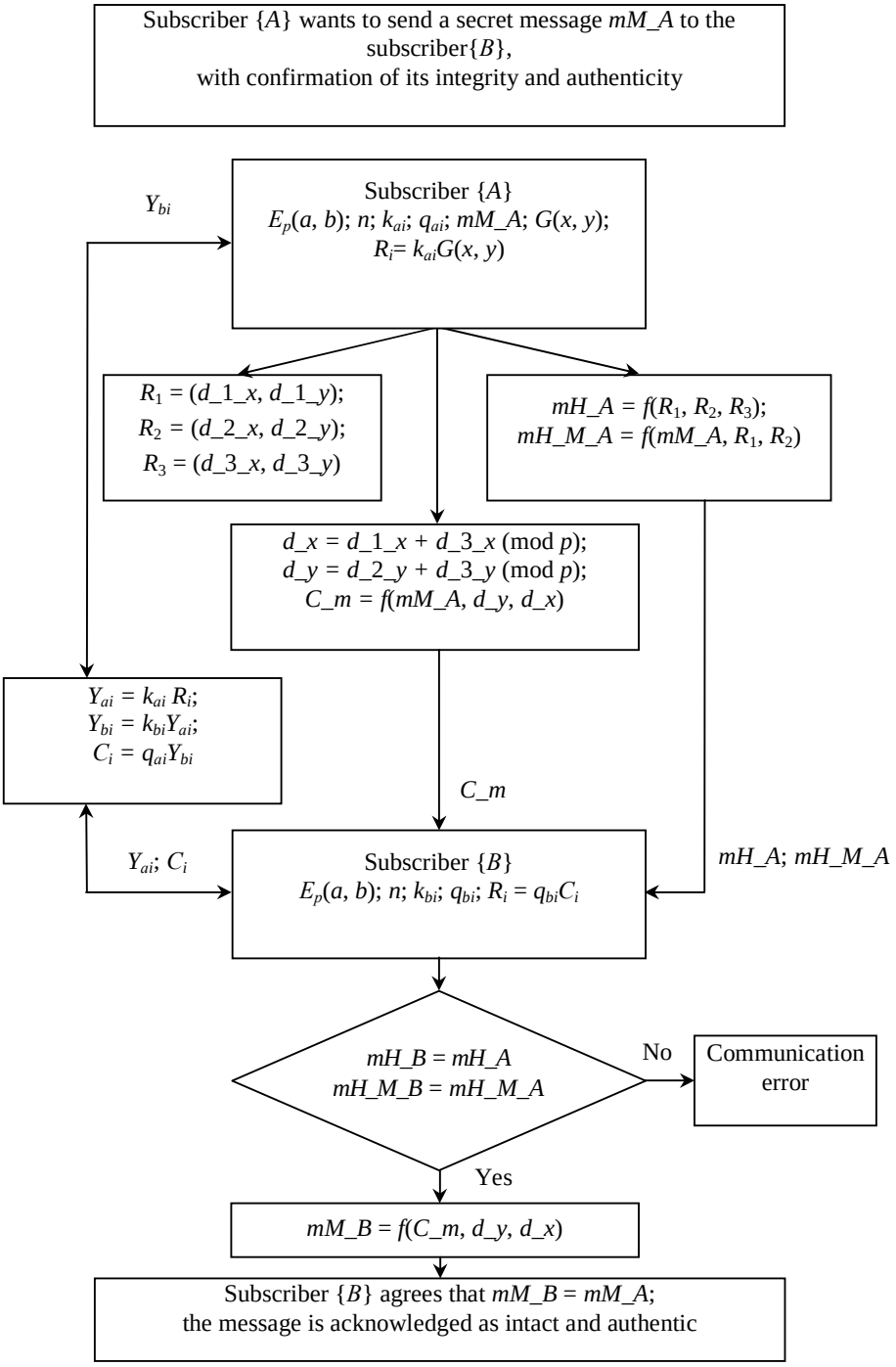

Fig. 1. Scheme of cryptographic triple-protocol

If we now use the distributed image of the Massey - Omura protocol, which uses pairs of numbers (k_a, q_a) and (k_b, q_b) instead of (A_i, A_1_i) and (B_i, B_1_i) we get the following Fig. 1, which shows the main moments of the triple-protocol.

3 Examples of Using the Triple-protocol

So, we will try to use a triple-protocol to pass the encrypted code as the basis of the secured channel.

As the basis for the study, take an elliptic curve with a small value of the module.

Let $E_{241}(0, -4)$; $G = (2, 2)$; $n = 211$; $241G = 0$; $p = 241$ corresponding to the curve $y^2 = x^3 - 4$. Next we will refer to the protocol points defined in section 1:

a) Subscriber A generates an arbitrary number on the interval $[\sqrt{p} \dots p) - mM\_A = 82$.
b) Subscriber A finds some random numbers $r_1 = 116$; $r_2 = 98$; $r_3 = 196$ and calculates $R_i = r_i[G]$:

$R_1 = 116[2, 2] = (203, 180)$,
$R_2 = 98[2, 2] = (74, 201)$,
$R_3 = 196[2, 2] = (118, 155)$.

c) Subscribers A and B independently find their random elements A_i and B_i:

$A_1 = 62$; $A\_1\_1 = 35$; $A_2 = 81$; $A\_1\_2 = 122$; $A_3 = 182$; $A\_1\_3 = 49$,
$B_1 = 168$; $B\_1\_1 = 33$; $B_2 = 59$; $B\_1\_2 = 192$; $B_3 = 9$; $B\_1\_3 = 134$.

d) The subscriber A calculates d_x, d_y, C_m, mH_A and mH_M_A:

$d\_x = 203 + 118 \pmod{241} = 80$,
$d\_y = 201 + 155 \pmod{241} = 115$,
$C\_m = 82 \wedge 80 \wedge 115 = 113$,
$mH\_A = (118 \wedge 201)^{(203 \wedge 155)} \pmod{241} = 15$,
$mH\_M\_A = (82 \wedge 203)^{(118 \wedge 155 \wedge 201)} \pmod{241} = 177$.

e) The subscriber A publish in open access numbers $mH\_A = 15$ and $mH\_M\_A = 177$. After that, it is possible to use a triple-protocol for the exchange of classified information using open channels:

a) The first round $A \rightarrow B$: $Y_{ai} = r_i[G]$:

$Y_{a\_1} = 62[203; 180] = (130, 203)$,
$Y_{a\_2} = 81[74; 201] = (28, 209)$,
$Y_{a\_3} = 182[118; 155] = (159, 114)$;

b) The second round $A \leftarrow B$: $Y_{bi} = b_i[Y_i]$:

$Y_{b\_1} = 168[130, 203] = (104, 190)$,
$Y_{b\_2} = 59[28, 209] = (34, 73)$,
$Y_{b\_3} = 9[159, 114] = (96, 3)$;

c) The third round $A \rightarrow B$: C_m, $(C\_ab)_i = A\{A\_1\_i[Y_{bi}]\}$:

$(C\_ab)_1 = 35[104, 190] = (107, 124)$,
$(C\_ab)_2 = 122[34, 73] = (5, 11)$,
$(C\_ab)_3 = 49[96, 3] = (16, 111)$,
$C\_m = 82 \wedge 80 \wedge 115 = 113$.

Then subscriber *B* calculates $(rG\_B)_i = B\_1\_i(C\_ab)_i$:

$(rG\_B)_1 = 33[107, 124] = (203, 180)$,
$(rG\_B)_2 = 192[5, 11] = (74, 201)$,
$(rG\_B)_3 = 134[16, 111] = (118, 155)$.

d) The subscriber *B* then calculates *d_x* and *d_y* and checks for the identity *mH_B* = *mH_A* and *mH_M_B* = *mH_M_A*:

$d\_x = 203 + 118 \pmod{241} = 80$,
$d\_y = 201 + 155 \pmod{241} = 115$,
$mH\_B = (118 \wedge 201)^{203 \wedge 155} \pmod{241} = 15$,
$mM\_B = 113 \wedge 115 \wedge 80 = 82$,
$mH\_M\_B = (82 \wedge 203)^{(118 \wedge 155 \wedge 201)} \pmod{241} = 177$.

We get

$mH\_B = mH\_A = 15$,
$mH\_M\_B = mH\_M\_A = 177$,
$mM\_B = mM\_A = 82$,

hence the data transmission is acknowledged to be successful and error free.

As can be seen from this example, with a single transfer of the master code there is no saving of network traffic, only the process of a possible break in the secure connection at the expense of three-fold encoding increases.

Table 1. Summary table of step-by-step transfer of the master code and basic ratios in the triple-protocol

№	Name	Message: *mM_A* = 82			Note
		Element			
		0	**1**	**2**	
0	A_i	62	81	182	
1	*A_1_i*	35	122	49	*p* = 241
2	B_i	168	59	9	
3	*B_1_i*	33	192	134	mod 241
4	r_i	116	98	196	
5	R_i	(203, 180)	(74, 201)	(118, 155)	
6	Y_{ai}	(130, 203)	(28, 209)	(159, 114)	*n_G* = 211
7	Y_{bi}	(104, 190)	(34, 73)	(96, 3)	
8	*C_ab_i*	(107, 124)	(5, 11)	(16, 111)	mod 211
9	$(rG\_B)_i$	(203, 180)	(74, 201)	(118, 155)	
10	*d_1_x*	203	–	–	
11	*d_2_y*	201	–	–	
12	*d_3*	(118, 155)	–	–	
13	*d_x*	80	–	–	*p* = 241
14	*d_y*	115	–	–	
15	*mM_B*	82	–	–	mod 241
16	*C_m*	113	–	–	
17	*mH_A*	15	–	–	
18	*mH_M_A*	177	–	–	

Now consider the transmission of the composite message "Hello world!", which will specifically display with the error "Hello world !".

Table 2. Initial information for the transmission of a complex message

№	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
m.	“	H	e	l	l	o		w	o	r	l	d		!	“
ch.	34	72	101	108	108	111	32	119	111	114	108	100	32	33	34

Then the following events occur:

a) Subscriber *A* counts the total number of transmitted symbols in the message (Σ №$_i$ = 14);
b) Then the subscriber *A* forms the vector *v_A* from the *i*-th number of random elements; a similar operation is performed by subscriber *B*.

Let the following vectors be formed:

v_A = {29 89 5 17 36 21 8 156 132 14 27 38 47 98 49};
v_B = {13 180 106 101 30 224 103 109 9 107 234 117 98 179 225}.

immediately organized vectors *v_A*_1 and *v_B*_1, where each element is inverse to the corresponding modulo p:

$z\_1 = z^{p-2} \bmod p$.

Get the following values:

*v_A*_1 = {133 65 193 156 154 23 211 17 42 155 125 222 200 91 182},
*v_B*_1 = {204 79 216 105 233 85 117 199 134 232 172 103 91 206 15}.

In the same way, the vector *v_r*, consisting of r_i-elements, and *v_R* = {R_i} are organized:

v_r = {104 15 147 85 173 146 159 179 192 149 164 26 153 94 147},
v_R = {(163, 50) (28, 2) (37, 3) (30, 58) (55, 37) (17, 30) (20, 191) (52, 194)
(207, 115) (81, 136) (16, 100) (51, 136) (132, 203) (37, 208) (37, 3)}.

The further process of data transfer from {*A*} to {*B*} occurs similarly to the one discussed earlier when forwarding the master code.

Thus, when transmitting a message of more than five parts in length it is possible to achieve savings in the use of the information channel, while at the same time increasing the cryptostability of three or more times.

Table 3. A summary table for the transmission of a composite message

№	**Mes.**	**ASCII**	***d_1_x***	***d_2_y***	***d_3***	***d_x***	***d_y***	***C_m***	***mM_B***	***mH***	***mH_M***
0	“	34	163	2	37, 3	200	5	239	34	225	25
1	H	72	28	3	30, 58	58	61	79	72	151	215
2	e	101	37	58	55, 37	92	95	102	101	1	1
3	l	108	30	37	17, 30	47	67	0	108	1	194
4	l	108	55	30	20, 191	75	221	250	108	231	91
5	o	111	17	191	52, 194	69	144	186	111	168	115
6		32	20	194	207,115	227	68	135	32	112	61
7	w	119	52	115	81, 136	133	10	248	119	123	4
8	o	111	207	136	16, 100	223	236	92	111	44	87

№	**Mes.**	**ASCII**	***d_1_x***	***d_2_y***	***d_3***	***d_x***	***d_y***	***C_m***	***mM_B***	***mH***	***mH_M***
9	r	114	81	100	51, 136	132	236	26	114	98	175
10	l	108	16	136	132, 203	148	98	154	108	162	103
11	d	100	51	203	37, 208	88	170	150	100	29	98
12		32	132	208	37, 3	169	211	90	32	64	216
13	!	33	37	3	163, 50	200	53	220	33	94	16
14	“	34	37	50	28, 2	65	52	87	34	168	97
Additional Segments			163 28	p = 241 mod 241							

4 Advantages and Disadvantages of the Triple-protocol

Advantages:

a) First of all, this protocol, unlike the usual unassembled protocols, can independently provide the transmission of a message confirming its affiliation to a particular sender, because the free transfer to the open network carrier numbers *mH* and *mH_M* makes it difficult to substitute the message and the sender.
b) Even if the correct internal log values are available to the attacker, it does not bring it closer to the disclosure because of the ambiguity of the inverse transformations associated with the features of the XOR changes over the relevant elements.
c) The advantage of using the *mH* and *mH_M* calculation from standard hashing is as follows:

   1) The performance of the software implementation (nothing superfluous and simple implementation in any programming language);
   2) Independent of external manufacturers, because it is impossible to be completely sure that in the program providing hashing, there are no hidden defects in the code or mathematical implementation.

d) The protocol has a standard advantage over symmetric protocols - the ability to secretly transmit data through unprotected channels.
e) The idea of an arbitrary sample from a pre-ordered array of random values at a given range is proposed to reduce the risk of the same numbers and predict a further number when disclosing the value of its predecessor. Consequently, the cracked protocol is resistant to attacks like the "person inside" and to the attacks on the weakness of the generation of arbitrary numbers, as well as to the attacks of the kind selection based on known texts, given the complexity of the discrete logarithm in the modular field formed on the elliptic curve.

Disadvantages:

a) There are no mechanisms for finding and correcting errors in the data transfer, which requires encapsulation of a lower from the point of view of the hierarchy of the OSI protocol model, but this is not very difficult either from the software or from the technical side;
b) High-performance computing power is required with software support of the so-called "long arithmetic". In a simplified project to test the concept, I used the Python programming language to organize the simplest messaging work, and problems with timely processing of information were not noticed.

5 Conclusions

In this short article, the author did not pursue the goal of explaining all the features of constructing a triple-protocol for practical applications, which would significantly increase the article to the size of a monograph,

but for fans of building secure networks for information exchange, the applicability and development of the triple protocol is demonstrated.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Diffie W, Hellman ME. New directions in cryptography. Information Theory, IEEE Transactions. 1976;22(6):644–654.

[2] Elliptic Curve Digital Signature Algoritm:
Available:https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algoritm

[3] ANSI X9.62–1999. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA); 1999.

[4] ANSI X9.63–1999. Public Key Cryptography for the Financial Services Industry: Elliptic Curve Key Agreement and Transport Protocols; 1999.

[5] IEEE Std 1363–2000. IEEE Standard Specifications for Public-Key Cryptography; 2000.

[6] National Institute of Standards and Technology. NIST FIPS PUB 186, Digital Signature Standard, U.S. Department of Commerce; 1994.

[7] Menezes A. Elliptic curve public key cryptosystems / Menezes A. – Kluwer Academic Publishers; 1993.

[8] Zakharchenko MV, Onatsky OV, Yona LG, Shikarchuk TM. Asymmetric methods of encryption in telecommunications. Module 2 - Cryptographic Methods of Information Protection in Telecommunication Systems and Networks. ONAT n.a. O. S. Popov; 2011.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/21974