Theoretical and Applied Informatics
Vol. 28 (2016), no. 4, pp. 1–14
DOI: 10.20904/284001

Post-Quantum Cryptography: generalized ElGamal cipher over GF(251^8)

P. Hecht[1*]

[1]Maestría en Seguridad Informática
Facultad de Ciencias Económicas, Facultad de Ciencias Exactas y Naturales y Facultad de Ingeniería
Universidad de Buenos Aires
Av. Córdoba 2122, (C1120AAQ) Ciudad de Buenos Aires, República Argentina

Abstract Post-Quantum Cryptography (PQC) attempts to find cryptographic protocols resistant to attacks by means of for instance Shor's polynomial time algorithm for numerical field problems like integer factorization (IFP) or the discrete logarithm (DLP). Other aspects are the backdoors discovered in deterministic random generators or recent advances in solving some instances of DLP. The use of alternative algebraic structures like non-commutative or non-associative partial groupoids, magmas, monoids, semigroups, quasigroups or groups, are valid choices for these new kinds of protocols. In this paper, we focus in an asymmetric cipher based on a generalized ElGamal non-arbitrated protocol using a non-commutative general linear group. The developed protocol forces a hard subgroup membership search problem into a non-commutative structure. The protocol involves at first a generalized Diffie-Hellman key interchange and further on the private and public parameters are recursively updated each time a new cipher session is launched. Security is based on a hard variation of the Generalized Symmetric Decomposition Problem (GSDP). Working with $GF(251^8)$ a 64-bits security is achieved, and if $GF(251^{16})$ is chosen, the security rises to 127-bits. An appealing feature is that there is no need for big number libraries as all arithmetic if performed in $\mathbb{Z}_{251}$ and therefore the new protocol is particularly useful for computational platforms with very limited capabilities like smartphones or smartcards.

Keywords Post-Quantum Cryptography; Non-Commutative Cryptography; Finite Fields

Received 22 FEB 2017 **Revised** 26 MAY 2017 **Accepted** 29 MAY 2017

*E-mail: phecht@dc.uba.ar

1 Introduction

Post-Quantum Cryptography is a relatively new cryptologic trend that recently acquired an official NIST status [1, 2] and which aims to be resistant to quantum computers attacks (like Shor algorithm). But PQC not only cover against that menace, it works also as a response against side-channel attacks [3], the increasing concern about pseudo-prime generator backdoor attacks (i.e. Dual_EC_DRBG NSA [4]) or the development of quasi-polynomial discrete logarithm attacks [5] which impact severely against current de facto standards [6] of asymmetric cryptography whose security rest on integer-factorization (IFP) and discrete-logarithm (DLP) over numeric fields. And more, sub-exponential time complexity attacks on many instances appear [5, 6]. Shor algorithm [7] opened a quantum computing way to break current asymmetric protocols. As a response, there rise an increasing interest in some simple solutions like Lattice-based, Pairing-based, Multi Quadratic, Code-based, Hash-based, Non-Commutative and Non-Associative algebraic cryptography [1, 2, 8–13].

A whole branch of new protocols was developed which do not rely on extended arithmetic's precision and instead exploit internal asymmetry of abstract algebraic structures like partial grupoids, categories, magmas, monoids, quasigroups, groups, rings, loops or neofields [9–24]. The new developed one-way trapdoor functions (OWTF) include conjugator search (CSP), decomposition (DP), commutative subgroup search (CSSP), symmetric decomposition (SDP) and generalized symmetric decomposition (GSDP) [9, 15, 17, 25, 26].

This paper focus a simple solution using the general linear multiplicative subgroup over prime field F_{251}, represented as $GL(d, F_{251})$, d is the square matrix order. All arithmetic operations are into Z_{251}. The prime characteristic 251 is the biggest one fitting into a byte. As advantage, no big number libraries are involved, memory requirement is reduced and fast computation is expected.

As a necessary condition for asymmetric cryptography, a hidden commutative subgroup is developed inside. PQC studies were purposely followed by the author over his past research [27–32]

2 Algebraic concepts

Let p be a prime, d any integer greater than one, $q = p^d$ and $F_p[x]$ the polynomial extension of the prime field F_p. The number of square matrices of order d and values in F_p is p^{d^2}, and of those p^{d^2-d} are nilpotent [33–36]. The number of elements in the general linear group of d-order non-singular square matrices is

$$|GL(d, F_p)| = \prod_{i=0}^{d-1} (p^d - p^i). \tag{1}$$

A non-singular matrix of d-order whose monic characteristic polynomial is irreducible in F_p, generates a cyclic (thus commutative) subgroup P_d of $M_d = \mathrm{GL}(d, F_p)$. Each d-degree irreducible polynomial $f(x)$ in $F_p[x]$ field has a square companion matrix of d-order who acts as a generator of the multiplicative cyclic subgroup P_d, and each member of this subgroup corresponds to a unique monic characteristic polynomial of at most $d-1$ degree [33]. The N_{tot} number of non-trivial (null

or unitary) monic d-degree $f(x)$ over F_{251} field is

$$N_{\text{tot}} = p^d - 2. \tag{2}$$

Using Möbius μ function, the $N_p(d)$ number of monic irreducible d-degree polynomials over $F_p[x]$ field is

$$N_p(d) = \frac{1}{d}\sum_{r|d}\mu(d)p^{d/r} = \frac{p^d-2}{d} = \frac{N_{\text{tot}}}{d}. \tag{3}$$

To generate a random d-order monic irreducible polynomial over $F_p[x]$, we use the probabilistic Algorithm 4.70 [6] whose complexity is $O(m^3 \lg(m) \lg(p))$ and requires approximately d-attempts. Once found, it is translated into the companion matrix [33]. Upon, it is of interest to find its order, because that would be the number of elements of the commutative subgroup P_d of the M_d matrix group. Whatever this value is, it must be a divisor of the multiplicative subfield order $p^d - 1$ and if it were maximal, the irreducible polynomial would be a primitive one. To calculate polynomial orders, a modified version of Algorithm 4.77 [6] can be used.

Clearly using an irreducible polynomial in an extension field is a method of generating a P_d commutative subgroup of the non-singular modular square matrices, but there exists another way to achieve the same goal. For matrices, the necessary and sufficient condition for two symmetric (diagonalizable) matrices to commute, is that they share the same orthonormal basis, that means the same eigenvectors P matrix [34, 35]. If we start from two different diagonal matrices D_1, D_2, then the transformed $A = PD_1P^{-1}$ and $B = PD_2P^{-1}$ commute, i.e. $AB = BA$. The later approach is computational faster than the first one, so it will be followed in our protocol.

3 Cryptographic aspects

Security of an asymmetric cipher protocol always relies on a hard OWTF [6]. Here we propose a generalized ElGamal cipher selecting GSDP as the one-way trapdoor function. If the algebraic structure and OWTF are well selected, a provably secure protocol could be developed [9,15]. This sounds good, but it is not easy to prove such a claim [37]; so caution at use is strongly advised. In our case, the GSDP could be stated as follows

> Let G be a non commutative group and S a commutative subgroup, given $(x, y) \in G^2$ and $(m, n) \in \mathbb{Z}^2$, find $z \in S | y = z^m x z^n$. (4)

This structure resembles a generalized discrete logarithm (GDLP) or a conjugation search problem (CSP). GSDP is more difficult as the first one, as no numeric field is directly involved and because the vector structure of elements is involved. GSDP is clearly a generalization of CSP, so a more sophisticated solution must be expected. GSDP is supposed to be one of the hardest challenges in group theory [9, 14–17]. As no cryptanalytic quantum algorithm is on sight and probably does not exist, the present protocol belongs to the PQC set. Of course, this statement should be proven, a question beyond the purpose of this paper.

In our protocol, we use a harder variety of GSDP, with less known information. We call it blind general symmetric decomposition problem (BGSDP), and it states as:

> Let G be a non commutative group and S a commutative subgroup, given $y \in G$ but unknown $x \in G, (m, n) \in \mathbb{Z}^2$, find $z \in S | y = z^m x z^n$. (5)

Not only this kind of generalized discrete logarithm problem is at least difficult as GSDP, in our case we change all hidden parameters each time a new cipher session is started. We accomplish this with an iterated update of those parameters.

4 Cipher Protocol

In our version, we work with two entities (Alice and Bob), but this could be easily generalized for any number of participants. All arithmetic operations should be assumed belonging to field F_{251}. At following box Tab. 1, common symbols are explained as used along this protocol. The setup steps Tab. 2 involve a generalized Diffie-Hellman key exchange.

Suppose that this protocol is intended to be used among an n-entities community then some caution should be held. The key point would be that each pair of interacting entities should store last interchanged session key until next opened session. That is not a big inconvenience and the protocol remains non-arbitrated.

Another feature could be the incorporation of authentication to block man-in-the-middle attacks. That could be made in a chained mode if each entity begins session exchanging HMAC codes [6] involving the last public key, a timestamp and eventually the last HMAC exchanged.

Symbol	Definition
$\in$	belongs to
$\in_R$	randomly selected element in
$\forall \neq$	all different elements
$M_8 \equiv GL(8, F_{251})$	non-commutative group
$P_8 \in M_8$	commutative subgroup
D_A, D_B	diagonal matrices
$K^{\nearrow}_{8,1}$	left upward first non-zero term of the secondary diagonal
$K^{\swarrow}_{1,8}$	right downward first non-zero term of the secondary diagonal
$K^{\searrow}_{1,1}$	left downward first non-zero term of the principal diagonal
$K^{\nwarrow}_{8,8}$	right upward first non-zero term of the secondary diagonal
$\Longrightarrow$	send publicly to the other entity
validation	greyed consistency proof

Table 1 Symbols and definitions. To clarify the four matrices arrow symbols, note that every square matrix has two diagonal rows (the main or principal diagonal and the secondary or counter diagonal). Each matrix arrow symbol point to the first upward or downward non-zero term appearing on the referred diagonal. At last reference, the double lined arrow mean that the last obtained value is publicly sent to the opposite entity.

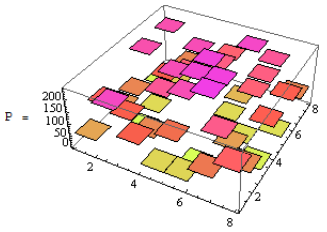
	Alice	Bob
Any entity begins	$P \in_R M_8$ $G \in_R M_8$	
Generating private elements	$k_1, k_2 \in_R \mathbb{Z}^*_{251}$ $\forall \neq \lambda_1, \ldots, \lambda_8 \in_R \mathbb{Z}^*_{251}$ $D_A = (\lambda_1, \ldots, \lambda_8)$ $A = PD_AP^{-1} \in P_8$	$r_1, r_2 \in_R \mathbb{Z}^*_{251}$ $\forall \neq \mu_1, \ldots, \mu_8 \in_R \mathbb{Z}^*_{251}$ $D_B = (\mu_1, \ldots, \mu_8)$ $B = PD_BP^{-1} \in P_8$
Interchange tokens	$A' = A^{k_1}GA^{k_2} \implies$	$B' = B^{r_1}GB^{r_2} \implies$
First common key K is obtained	$K = A^{k_1}B'A^{k_2}$ $m = K^{\nearrow}_{8,1}.K^{\swarrow}_{1,8}$ $n = K^{\searrow}_{1,1}.K^{\nwarrow}_{8,8}$	$K = B^{r_1}A'B^{r_2}$ $m = K^{\nearrow}_{8,1}.K^{\swarrow}_{1,8}$ $n = K^{\searrow}_{1,1}.K^{\nwarrow}_{8,8}$
	$K = A^{k_1}T_BA^{k_2} = A^{k_1}(B^{r_1}G_0B^{r_2})A^{k_2}$ $= B^{r_1}(A^{k_1}G_0A^{k_2})B^{r_2} = B^{r_1}T_AB^{r_2} = K$	
Alice start a new cipher session updating recursively parameters	$K = K^{m.n}$ $m = K^{\nearrow}_{8,1}.K^{\swarrow}_{1,8}$ $n = K^{\searrow}_{1,1}.K^{\nwarrow}_{8,8}$ $P = K^mPK^n$ $G = K^mGK^n$ $A = PD_AP^{-1}$ $A' = A^mGA^n \implies$	
Bob acknowledges and update parameters		$K = K^{m.n}$ $m = K^{\nearrow}_{8,1}.K^{\swarrow}_{1,8}$ $n = K^{\searrow}_{1,1}.K^{\nwarrow}_{8,8}$ $P = K^mPK^n$ $G = K^mGK^n$ $B = PD_BP^{-1}$ $B' = B^mGB^n \implies$
Alice ciphers an H message to Bob	$H \in M_8$ $J \in_R P_8$ $C = (y_1, y_2) \implies$ $y_1 = J^mGJ^n$ $y_2 = H(J^mB'J^n)$	
Bob deciphers H		$H = y_2(B^my_1B^n)^{-1}$
	$H = y_2(B^my_1B^n)^{-1}$ $= H(J^mB'J^n)(B^my_1B^n)^{-1}$ $= H(J^mB^m)G_1(B^nJ^n)(B^my_1B^n)^{-1}$ $= H(B^m(J^mG_1J^n)B^n)(B^my_1B^n)^{-1}$ $= H(B^my_1B^n)(B^my_1B^n)^{-1}$ $= H$	

Table 2 Setup steps, new session start, consequent parameters updating and message encryption and decryption cycle"

5 Step-By-Step Sample

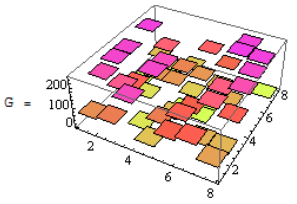
All symbols used here refers to the previous section. Any interested reader should be able to reconstruct this sequence, as no hidden values are included into this description.

1. Cipher setup. Any entity defines P and send to the other.



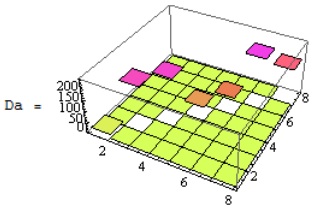
$$\begin{pmatrix} 59 & 214 & 111 & 33 & 34 & 86 & 147 & 53 \\ 140 & 115 & 75 & 94 & 6 & 175 & 75 & 175 \\ 67 & 0 & 183 & 204 & 238 & 62 & 18 & 75 \\ 76 & 31 & 206 & 224 & 232 & 28 & 131 & 166 \\ 199 & 118 & 160 & 229 & 31 & 161 & 28 & 50 \\ 14 & 41 & 205 & 170 & 50 & 169 & 123 & 14 \\ 200 & 76 & 54 & 157 & 217 & 184 & 76 & 201 \\ 11 & 237 & 87 & 244 & 219 & 226 & 52 & 51 \end{pmatrix}$$

2. Cipher setup. Any entity defines G and send to the other, it would also be possible that one defines P and the other answers G.

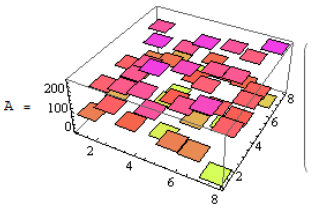


$$\begin{pmatrix} 74 & 86 & 221 & 45 & 115 & 133 & 64 & 80 \\ 210 & 185 & 25 & 86 & 139 & 56 & 52 & 102 \\ 231 & 140 & 228 & 48 & 111 & 171 & 138 & 85 \\ 230 & 29 & 217 & 6 & 1 & 178 & 210 & 248 \\ 7 & 69 & 70 & 39 & 2 & 57 & 229 & 43 \\ 242 & 167 & 81 & 69 & 48 & 249 & 242 & 33 \\ 49 & 0 & 41 & 156 & 102 & 239 & 11 & 80 \\ 214 & 127 & 113 & 199 & 76 & 63 & 241 & 161 \end{pmatrix}$$

3. Alice defines her initial private keys, she also has randomly selected $k_1 = 77$ and $k_2 = 184$.



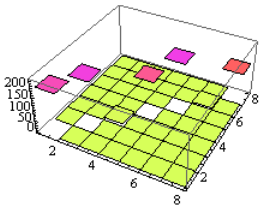
$$\begin{pmatrix} 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 211 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 221 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 65 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 82 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 238 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 165 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 142 \end{pmatrix}$$



$$\begin{pmatrix} 86 & 154 & 107 & 197 & 85 & 78 & 3 & 160 \\ 164 & 183 & 194 & 23 & 188 & 222 & 158 & 203 \\ 57 & 188 & 245 & 132 & 58 & 156 & 142 & 14 \\ 157 & 187 & 29 & 219 & 169 & 8 & 160 & 13 \\ 64 & 244 & 11 & 120 & 187 & 189 & 145 & 30 \\ 81 & 98 & 189 & 241 & 112 & 103 & 42 & 44 \\ 198 & 58 & 18 & 103 & 196 & 174 & 248 & 195 \\ 180 & 112 & 225 & 65 & 34 & 201 & 203 & 123 \end{pmatrix}$$

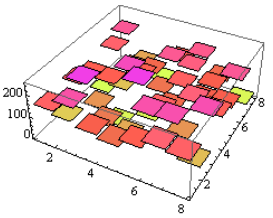
4. Bob defines his initial private keys, he also has randomly selected $r_1 = 42$ and $r_2 = 229$.

Db =



193	0	0	0	0	0	0	0
0	217	0	0	0	0	0	0
0	0	12	0	0	0	0	0
0	0	0	168	0	0	0	0
0	0	0	0	222	0	0	0
0	0	0	0	0	7	0	0
0	0	0	0	0	0	135	0
0	0	0	0	0	0	0	220

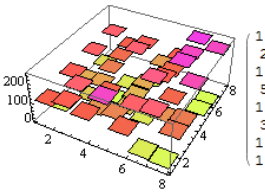
B =



158	206	122	105	137	169	159	165
48	250	94	66	209	211	56	177
163	78	26	250	158	78	213	199
139	139	221	27	107	174	149	64
9	53	141	99	155	117	127	208
164	94	109	192	179	125	194	116
194	69	12	140	188	106	24	46
188	138	117	24	75	54	126	158

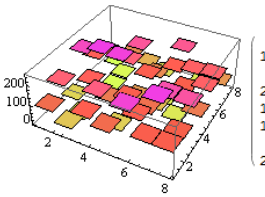
5. Alice and Bob define tokens and exchange them.

A' =



114	139	111	227	106	23	16	18
21	163	71	81	159	168	94	37
143	42	104	45	25	74	88	190
58	64	1	54	180	104	215	36
156	86	25	60	130	237	14	40
33	101	167	15	105	225	2	130
162	114	147	66	131	250	225	82
135	44	92	14	91	58	132	220

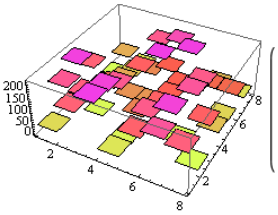
B' =



99	44	134	100	237	121	120	247
169	26	177	39	129	249	111	216
36	146	87	123	96	208	201	220
223	250	242	67	50	24	88	10
144	7	172	127	20	117	156	63
116	16	196	77	136	154	162	215
0	76	55	28	189	23	70	183
200	37	182	49	97	225	218	5

6. Both obtain the first common session key and the first power parameters using diagonal values ($m = 41$, $n = 178$, $m.n = 19$).

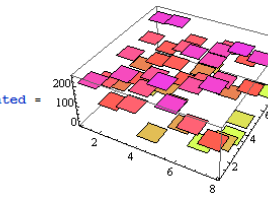
K =



46	162	231	40	155	185	166	98
180	167	222	33	175	222	25	162
226	30	178	88	150	29	149	219
162	100	80	78	136	185	58	35
24	70	148	230	131	174	48	97
19	53	85	125	228	107	106	89
67	39	176	200	18	61	42	121
213	138	111	161	39	96	73	113

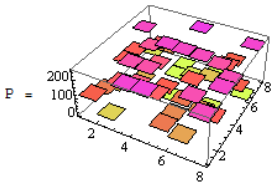
7. Alice starts a new cipher session. Both update the session key using the current power parameters and calculate new power parameters ($m = 139$, $n = 203$).

K updated =

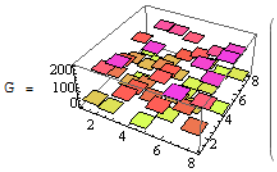


214	142	162	51	229	167	127	227
162	218	181	244	65	4	53	58
87	143	190	79	198	232	11	162
189	37	66	140	165	200	31	160
83	133	144	229	214	83	134	44
172	146	175	77	216	246	228	113
234	240	160	49	178	143	5	67
172	162	200	20	117	132	150	42

8. Now both independently update auxiliary matrices.

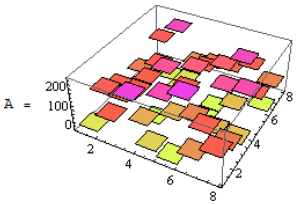


$$P = \begin{pmatrix} 116 & 43 & 233 & 237 & 93 & 68 & 219 & 63 \\ 89 & 159 & 180 & 159 & 82 & 211 & 146 & 231 \\ 50 & 207 & 205 & 132 & 96 & 78 & 227 & 173 \\ 144 & 59 & 177 & 16 & 146 & 25 & 214 & 112 \\ 145 & 195 & 229 & 224 & 219 & 203 & 120 & 120 \\ 223 & 157 & 9 & 32 & 51 & 149 & 10 & 142 \\ 34 & 77 & 8 & 247 & 17 & 118 & 231 & 106 \\ 150 & 19 & 216 & 21 & 110 & 29 & 132 & 123 \end{pmatrix}$$

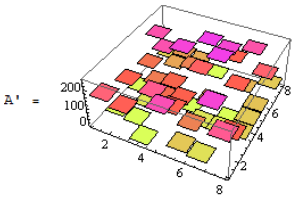


$$G = \begin{pmatrix} 43 & 30 & 139 & 13 & 140 & 243 & 91 & 107 \\ 165 & 231 & 114 & 94 & 20 & 101 & 166 & 223 \\ 89 & 56 & 80 & 147 & 56 & 40 & 229 & 129 \\ 106 & 14 & 233 & 129 & 22 & 228 & 3 & 109 \\ 65 & 51 & 76 & 50 & 18 & 28 & 240 & 38 \\ 141 & 16 & 51 & 112 & 148 & 188 & 225 & 243 \\ 3 & 195 & 191 & 171 & 41 & 10 & 26 & 40 \\ 121 & 159 & 108 & 203 & 120 & 110 & 221 & 222 \end{pmatrix}$$

9. Alice update her private and public session keys, note that for increased security reason, each new session use recursively updated keys.

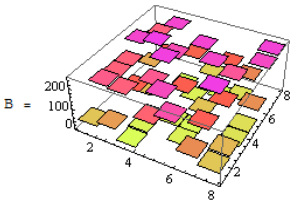


$$A = \begin{pmatrix} 25 & 108 & 238 & 37 & 13 & 72 & 125 & 132 \\ 143 & 141 & 58 & 210 & 247 & 127 & 88 & 176 \\ 106 & 111 & 1 & 54 & 80 & 54 & 44 & 206 \\ 0 & 114 & 193 & 127 & 9 & 195 & 208 & 21 \\ 83 & 135 & 26 & 172 & 6 & 48 & 137 & 74 \\ 14 & 24 & 28 & 126 & 160 & 218 & 15 & 45 \\ 142 & 218 & 70 & 89 & 97 & 138 & 79 & 208 \\ 66 & 219 & 80 & 181 & 245 & 163 & 146 & 44 \end{pmatrix}$$

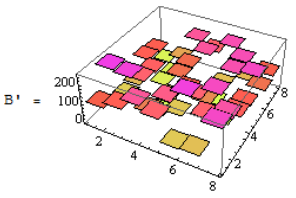


$$A' = \begin{pmatrix} 170 & 141 & 5 & 213 & 45 & 31 & 188 & 235 \\ 18 & 2 & 187 & 137 & 84 & 236 & 116 & 161 \\ 124 & 29 & 103 & 119 & 28 & 27 & 41 & 82 \\ 16 & 88 & 108 & 1 & 76 & 7 & 43 & 166 \\ 97 & 208 & 238 & 153 & 249 & 129 & 48 & 5 \\ 207 & 64 & 83 & 250 & 233 & 55 & 163 & 194 \\ 39 & 70 & 60 & 7 & 131 & 205 & 44 & 11 \\ 223 & 79 & 221 & 34 & 171 & 148 & 248 & 12 \end{pmatrix}$$

10. Bob updates his private and public session keys.

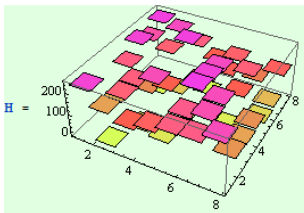


$$B = \begin{pmatrix} 47 & 69 & 14 & 165 & 208 & 79 & 46 & 92 \\ 184 & 174 & 21 & 221 & 23 & 144 & 54 & 171 \\ 173 & 130 & 189 & 22 & 18 & 249 & 30 & 66 \\ 173 & 51 & 245 & 127 & 48 & 124 & 33 & 227 \\ 232 & 250 & 214 & 23 & 229 & 194 & 77 & 115 \\ 62 & 135 & 3 & 235 & 5 & 31 & 226 & 215 \\ 174 & 233 & 109 & 40 & 123 & 82 & 239 & 181 \\ 37 & 117 & 211 & 246 & 128 & 175 & 234 & 138 \end{pmatrix}$$



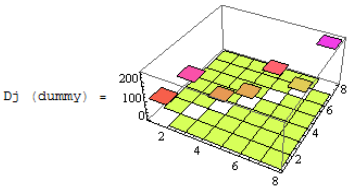
$$B' = \begin{pmatrix} 113 & 142 & 195 & 162 & 45 & 51 & 227 & 82 \\ 250 & 240 & 87 & 183 & 244 & 163 & 204 & 244 \\ 42 & 164 & 13 & 46 & 169 & 224 & 148 & 96 \\ 55 & 154 & 36 & 113 & 80 & 4 & 137 & 48 \\ 142 & 25 & 100 & 13 & 13 & 210 & 225 & 119 \\ 133 & 23 & 104 & 198 & 51 & 215 & 98 & 145 \\ 21 & 65 & 196 & 162 & 213 & 57 & 120 & 213 \\ 101 & 16 & 23 & 23 & 28 & 152 & 196 & 174 \end{pmatrix}$$

11. Alice choose H message to cipher, this modular matrix is a general one.

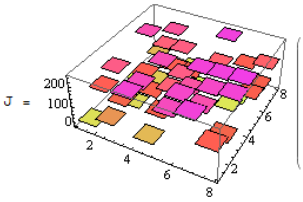


$$H = \begin{pmatrix} 237 & 13 & 132 & 156 & 181 & 185 & 220 & 39 \\ 74 & 174 & 57 & 37 & 188 & 213 & 81 & 59 \\ 66 & 178 & 217 & 15 & 250 & 219 & 73 & 228 \\ 181 & 87 & 128 & 22 & 239 & 32 & 83 & 14 \\ 237 & 27 & 55 & 123 & 225 & 135 & 44 & 249 \\ 206 & 171 & 243 & 23 & 149 & 176 & 61 & 11 \\ 245 & 224 & 181 & 131 & 165 & 95 & 154 & 35 \\ 70 & 97 & 190 & 95 & 38 & 91 & 89 & 128 \end{pmatrix}$$

12. Alice uses a random diagonal matrix to generate a session matrix J. It is mandatory to change J at each cipher session, the same as the k-parameter in a ElGamal numeric field cipher. Please watch out that the updated auxiliary matrix P are used to obtain J.

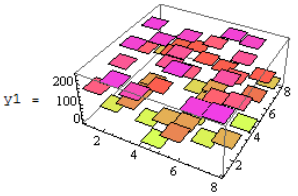


$$\begin{pmatrix} 112 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 202 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 68 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 59 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 159 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 245 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 25 \end{pmatrix}$$

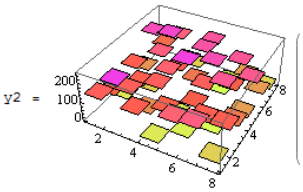


$$\begin{pmatrix} 29 & 71 & 234 & 51 & 219 & 239 & 134 & 39 \\ 140 & 104 & 219 & 112 & 227 & 207 & 104 & 32 \\ 180 & 32 & 120 & 168 & 200 & 226 & 235 & 29 \\ 5 & 130 & 214 & 179 & 238 & 241 & 248 & 198 \\ 19 & 40 & 136 & 137 & 76 & 25 & 121 & 108 \\ 210 & 170 & 176 & 6 & 161 & 0 & 112 & 128 \\ 52 & 196 & 26 & 40 & 242 & 130 & 158 & 162 \\ 12 & 69 & 226 & 186 & 246 & 152 & 9 & 244 \end{pmatrix}$$

13. Alice cipher H matrix.

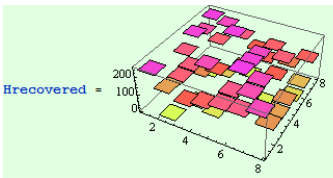


$$\begin{pmatrix} 175 & 239 & 212 & 11 & 80 & 239 & 245 & 173 \\ 25 & 74 & 18 & 57 & 66 & 20 & 56 & 170 \\ 188 & 69 & 126 & 239 & 33 & 162 & 176 & 121 \\ 231 & 65 & 187 & 175 & 55 & 199 & 2 & 214 \\ 163 & 229 & 190 & 35 & 64 & 151 & 92 & 76 \\ 238 & 117 & 225 & 133 & 119 & 190 & 131 & 96 \\ 46 & 106 & 121 & 169 & 234 & 222 & 44 & 19 \\ 138 & 88 & 1 & 47 & 195 & 147 & 61 & 133 \end{pmatrix}$$



$$\begin{pmatrix} 146 & 250 & 137 & 25 & 167 & 178 & 39 & 151 \\ 109 & 146 & 76 & 113 & 20 & 139 & 138 & 110 \\ 25 & 129 & 148 & 23 & 113 & 9 & 89 & 212 \\ 138 & 39 & 164 & 244 & 146 & 191 & 66 & 39 \\ 73 & 176 & 129 & 92 & 214 & 115 & 187 & 250 \\ 217 & 230 & 117 & 203 & 1 & 194 & 69 & 101 \\ 101 & 174 & 40 & 197 & 1 & 67 & 37 & 168 \\ 198 & 160 & 170 & 173 & 168 & 113 & 52 & 77 \end{pmatrix}$$

14. Bob recovers the H message.



$$\begin{pmatrix} 237 & 13 & 132 & 156 & 181 & 185 & 220 & 39 \\ 74 & 174 & 57 & 37 & 188 & 213 & 81 & 59 \\ 66 & 178 & 217 & 15 & 250 & 219 & 73 & 228 \\ 181 & 87 & 128 & 22 & 239 & 32 & 83 & 14 \\ 237 & 27 & 55 & 123 & 225 & 135 & 44 & 249 \\ 206 & 171 & 243 & 23 & 149 & 176 & 61 & 11 \\ 245 & 224 & 181 & 131 & 165 & 95 & 154 & 35 \\ 70 & 97 & 190 & 95 & 38 & 91 & 89 & 128 \end{pmatrix}$$

6 Benchmarking

To estimate the performance of the protocol, we used a simple textbook interpreted program written in *Mathematica 8+* language. This could be one of the worst scenario to test, but it also provides a kind of lower bound for the timing. The computational platform was an Intel® Core™ i5-5200U CPU @ 2.20GHz, 2 Cores, 4 Logical Processors 64-bit Windows 10 Home, version 10.0.14393, 8 GB physical RAM in a Dell XPS 13 9343.

The Mathematica notebooks here used are freely available upon request. In this simulation sample, instantaneous transfers between entities are assumed, so only computational steps are considered. At same time, no simultaneous or parallel computations are performed, Alice and Bob sum sequentially their timed calculations. All results informed refer to the mean run time of

1000 random iterations.

1. At setup, definition of P and G, took 0.12 ms,
2. from P, G already defined until first session key K and new power parameters obtained, took 29.56 ms,
3. new session updating took 52.94 ms,
4. Enciphering–deciphering cycle took 32.36 ms.

As observed, a full session of an approximate 64-bytes message (an H matrix) secured transmission took 85 ms in our environment. Of course, a lot of optimization should be accomplished before a real-life application is planned.

7 Protocol Security

The group of order 8 modular integer matrices $M(8, \mathbb{Z}_{251})$ has a cardinal $251^{64} \approx 10^{153.579}$. The invertible Hill matrices subgroup $M_8 = GL(8, \mathbb{Z}_{251})$ has a slightly lower order [36].

$$251^{64}\left(1-\frac{1}{251}\right)\left(1-\frac{1}{251^2}\right)\cdots\left(1-\frac{1}{251^8}\right) \approx 10^{153{,}177}. \tag{6}$$

Comparing both numbers, the probability of selecting a singular matrix in $M(8, \mathbb{Z}_2 51)$ is $p \approx 0.004$, a low but not negligible value. Each time a new random modular matrix is obtained, it must be controlled that his determinant is not null.

Supposing no other weakness are available, cracking a private key depends on an order eight diagonal matrix, so a brute force search of the commutative P_8 subgroup of M_8 involves the cardinal

$$|P_8| = 249.248.247.246.245.244.243.242 = 13{,}190{,}481{,}178{,}699{,}144{,}320 \approx 10^{19} \approx 2^{64}. \tag{7}$$

Currently it is impossible to make a systematic search of that space, and if a greater security is pursued, it would suffice to expand the commutative subgroup to P_{16}, who implies a 127-bit level. It is recommended to adopt a compromise solution between increased security and the concomitant use of more resources, which are always costly and limited.

A second way to attack the present protocol would be to find a polynomial time algorithm to solve the algebraic generalized symmetric decomposition. As some simpler OWTF based on algebraic conjugation were successfully cryptanalyzed [38, 39], it was mandatory to find very hard functions. We presented earlier (see Definition (5)) a stronger version, the blind general symmetric decomposition problem (BGSDP). As posted, it could be conjectured that this kind of algebraic challenge belongs to a NP time-complexity class and at same time resistant to quantum computers attacks. As said, this statement is currently unproved and it seems not easy to be solved.

As all matrices publicly involved in this protocol are indistinguishable from full random ones (same size, same numeric format), it could be conjectured that it complies with semantic security

levels IND-CPA, IND-CCA and IND-CCA2 [40]. These conditions are necessary to be qualified as a valid PQC solution. Perhaps there exists a completely different way to attack the present protocol; but at current time the author is unaware of it. As consequence, we assume a 64-bit security for the protocol as it is stated

8 Conclusions

We developed a non-arbitrated and compact algebraic post-quantum cipher protocol, which could easily be adapted to other purposes as key exchange, key transport and ZKP authentication [9,30]. By compact, we mean that no big number library is required as only $\mathbb{Z}_{251}$ field operations are involved. This feature would enable the use of it in low computational resources environments like smartphones, smartcards, etc.

Supplementary materials

- *PQC: ELGAMAL GF*(251^8) *CIPHER SAMPLE*, `DOI:10.20904/284001-supp1`
- *PQC: ELGAMAL GF*(251^8) *BENCHMARK*, `DOI:10.20904/284001-supp2`

References

[1] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. Report on Post-Quantum Cryptography. Technical report, 2016. DOI: 10.6028/nist.ir.8105.

[2] D. Moody. Update on the NIST Post-Quantum Cryptography Project, 2016. `http://csrc.nist.gov/groups/SMA/ispab/`, Accessed: 10.02.2017.

[3] YB. Zhou and DG. Feng. Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing. *State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences*, 2006.

[4] B. Schneier. Did NSA put a secret backdoor in new encryption standard? `https://www.wired.com/2007/11/securitymatters-1115/`, Accessed: 10.02.2017.

[5] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. *A Heuristic Quasi-Polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic*, pages 1–16. Springer Berlin Heidelberg, 2014. DOI: 10.1007/978-3-642-55220-5_1.

[6] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[7] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. DOI: 10.1137/S0036144598347011.

[8] P. S. L. M. Barreto, F. P. Biasi, R. Dahab, J. César, G. C. C. F. Pereira, and J. E. Ricardini. Introdução à criptografia pós-quântica. *Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais—SBSeg*, 2013.

[9] L. Gerritzen, D. Goldfeld, M. Kreuzer, G. Rosenberger, and Shpilrain V. *Algebraic Methods in Cryptography*, volume 518. American Mathematical Soc., 2006.

[10] B. Tsaban. Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography. *Journal of Cryptology*, 28(3):601–622, 2015. DOI: 10.1007/s00145-013-9170-9.

[11] A. Kalka. Non-associative public-key cryptography. *arXiv:1210.8270*, 2012.

[12] Cz. Kościelny. Generating quasigroups for cryptographic applications. *International Journal of Applied Mathematics and Computer Science*, 12(4):559–569, 2002.

[13] S. Markovski. Design of crypto primitives based on quasigroups. *Quasigroups and Related Systems*, 23(1):41–90, 2015.

[14] D. Grigoriev and I. Ponomarenko. Constructions in public-key cryptography over matrix groups. In *International Workshop on Algebraic Methods in Cryptography*, volume 418 of *Contemporary Mathematics*, pages 103–119. American Mathematical Soc., 2005.

[15] Z. Cao, X. Dong, and L. Wang. New public key cryptosystems using polynomials over non-commutative rings. *IACR Cryptology ePrint Archive*, 2007.

[16] S.-H. Paeng, D. Kwon, K.-Ch. Ha, and J. H. Kim. Improved public key cryptosystem using finite non abelian groups. 2001.

[17] J.-C. Birget, S. S. Magliverasy, and M. Sramkay. On public-key cryptosystems based on combinatorial group theory. *Tatra Mt. Math. Publ*, 33(137):137–148, 2006.

[18] M. I. González Vasco, C. Martínez, and R. Steinwandt. Towards a uniform description of several group based cryptographic primitives. *Designs, Codes and Cryptography*, 33(3):215–226, 2004.

[19] V. Shpilrain and A. Ushakov. *Thompson's Group and Public Key Cryptography*, pages 151–163. Springer Berlin Heidelberg, 2005. DOI: 10.1007/11496137_11.

[20] K. Mahlburg. An overview of braid group cryptography. 2004. `http://www.math.wisc.edu/~boston/mahlburg.pdf`.

[21] E. K. Lee. Braid groups in cryptology. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 87(5):986–992, 2004.

[22] B. Eick and D. Kahrobaei. Polycyclic groups: A new platform for cryptology? *arXiv:math/0411077*, 2004.

[23] A. Mahalanobis. The Diffie-Hellman key exchange protocol and non-abelian nilpotent groups. *Israel Journal of Mathematics*, 165(1):161–187, 2008. DOI: 10.1007/s11856-008-1008-z.

[24] V. A. Shcherbacov. Quasigroups in cryptology. *Computer Science Journal of Moldova*, 17(2):50, 2009. `https://ibn.idsi.md/en/vizualizare_articol/2712`.

[25] S. S. Magliveras, D. R. Stinson, and T. van Trung. New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups. *Journal of Cryptology*, 15(4):285–297, 2002.

[26] V. Shpilrain and G. Zapata. Combinatorial group theory and public key cryptography. *Applicable Algebra in Engineering, Communication and Computing*, 17(3):291–302, 2006. DOI: 10.1007/s00200-006-0006-9.

[27] J. P. Hecht. Un modelo compacto de criptografía asimétrica empleando anillos no conmutativos. In *Actas del V Congreso Iberoamericano de Seguridad Informática CIBSI*, volume 9, pages 188–201, 2009.

[28] P. Hecht. A Zero-Knowledge authentication protocol using non commutative groups. In *Actas del VI Congreso Iberoamericano de Seguridad Informática CIBSI*, volume 11, pages 96–102, 2011.

[29] P. Hecht. Criptografía no conmutativa usando un grupo general lineal de orden primo de mersenne. In *Actas del VII Congreso Iberoamericano de Seguridad Informática CIBSI*, volume 13, pages 147–153, 2013.

[30] P. Hecht. A post-quantum set of compact asymmetric protocols using a general linear group. In *Actas del VIII Congreso Iberoamericano de Seguridad Informática CIBSI*, volume 15, pages 96–101, 2015.

[31] P. Hecht. Zero-knowledge proof authentication using Left Self Distributive Systems: a post-quantum approach. In *Actas del VIII Congreso Iberoamericano de Seguridad Informática CIBSI*, volume 15, pages 96–101, 2015.

[32] J. Kamlofsky, J. Hecht, S. Abdel Masih, and O. Hidalgo Izzi. A Diffie-Hellman compact model over non-commutative rings using quaternions. In *VIII Congreso Iberoamericano de Seguridad Informática CIBSI, Quito*, 2015.

[33] R. Lidl and H. Niederreiter. *Finite fields*, volume 20. Cambridge university press, 1997.

[34] T. Beth, D. Jungnickel, and H. Lenz. *Encyclopedia of Mathematics and Its Applications*, volume 69. Cambridge University Press, Cambridge, 1999.

[35] R. A. Horn and Ch. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[36] J. Overbey, W. Traves, and J. Wojdylo. On the key space of the Hill Cipher. *Cryptologia*, 29(1):59–72, 2005. DOI: 10.1080/0161-110591893771.

[37] A. W. Dent. Fundamental problems in provable security and cryptography. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 364(1849):3215–3230, 2006. DOI: 10.1098/rsta.2006.1895.

[38] A. D. Myasnikov and A. Ushakov. Cryptanalysis of matrix conjugation schemes. *Journal of Mathematical Cryptology*, 8(2), 2014. DOI: 10.1515/jmc-2012-0033.

[39] A. A. Kamal and A. M. Youssef. Cryptanalysis of Álvarez et al. key exchange scheme. *Information Sciences*, 223:317–321, 2013. DOI: 10.1016/j.ins.2012.10.010.

[40] J. Katz and Y. Lindell. Introduction to modern cryptography: principles and protocols. Cryptography and network security, 2008.