Spring 4-29-2022

ElGamal Encryption in the General Linear Groups

Lynn Sherman

ElGamal Encryption in the General Linear Groups

Lynn Y. Sherman

submitted in partial fulfillment of the requirements for Honors in Mathematics at the University of Mary Washington

Fredericksburg, Virginia

April 2022

This thesis by Lynn Y. Sherman is accepted in its present form as satisfying the thesis requirement for Honors in Mathematics.

Date	Approved
_____	_____
	Randall Helmstutler, Ph.D. thesis advisor
_____	_____
	Janusz Konieczny, Ph.D. committee member
_____	_____
	Larry Lehman, Ph.D. committee member

Acknowledgement

I would like to thank to Dr. Randall Helmstutler for his great guidance and his detailed direction of this project. Furthermore I greatly appreciate the work of my committee members, Dr. Janusz Konieczy and Dr. Larry Lehman. Lastly, I would like to thank Dr. Brooks Kuykendall and Caitlin Holt who gave me a random numbers that helped me create the examples.

Contents

Abstract

We first modify ElGamal encryption system in $\mathbb{Z}_p^*$ to $\mathrm{GL_n}(\mathbb{F}_q)$. This allows us to change our message $\mathbf{m}$ to be n-component column vector which allows us to have some computation advantages and can encrypt the messages in a vector form. We check a validity of our modified system with examples. Then, we proceed to check some security aspects of our scheme. Ultimately, we suggest some directions to keep optimal security of our scheme.

1 ElGamal in $\mathbb{Z}_p^*$

1.1 Introduction

In this section, we will indicate some of the basic ideas of public key encryption system specifically ElGamal encryption system. ElGalmal encryption system is a public key encryption system that was first published by Egyptian cryptographer Taher ElGamal in 1985 [3]. ElGamal encryption system uses Diffie-Hellman Key Exchange (DHKE) as part of the scheme. DHKE cannot be used as an encryption system. However, DHKE is great way for two parties to create a common secret key. Because of this ElGamal does not require users to exchange secret keys. In other words, only a recipient needs to have a secret decryption key. Both ElGamal and DHKE operate in group $\mathbb{Z}_p^*$ where p is a prime.

Definition 1.1. Let G be a group and suppose $a \in G$. If there is a positive power $n \neq 1$ with $a^n = e$, we say $a \in G$ has *finite order*. In this case, the smallest such power is called the *order* of a.

Definition 1.2. Let G be a group. We say that G is *cyclic* if there exists and element $g \in G$ with $G =< g >$. In this case, g is called a *generator* for G.

We have a following definition for element g where g is a generator of G.

Definition 1.3. Let p be a prime. An element $g \in \mathbb{Z}_p^*$ is *primitive* if $< g >= \mathbb{Z}_p^*$, that is, if g is a generator for the cyclic group $\mathbb{Z}_p^*$.

An encryption key is made public and anyone can send a coded message to the recipient by using the possessive key. For this reason, management of secret keys is simple and one can send the information safely to the recipient. The security of ElGamal is based on the Discrete Log Problem, which is currently thought to be hard.

We introduce few individuals: Alice, Bob, and Eve. Alice is a receiver who receives messages from other people. Bob is a sender who sends the messages to Alice. Eve is a adversary who tries to break our encryption system.

In this chapter, we will go over how the classical ElGamal system works in $\mathbb{Z}_p^*$ and will check some security aspects of the classical scheme. Ultimately, we will modify our scheme with a different group.

1.2 How ElGamal Encryption System Works

In this section, we will discuss how the classical ElGamal encryption system works. In ElGamal encryption system, Alice first selects a prime p. In the encryption scheme, all calculations will then occur in the group $\mathbb{Z}_p^*$. Next, Alice selects an element $g \in \mathbb{Z}_p^*$ for a security reasons g should chosen to be primitive, and Alice chooses her secret number $x \in \mathbb{Z}$ which will end up being her private

decryption key. Alice calculates $A = g^x \bmod p$ using her secret number x. Finally, Alice publishes (p, g, A) as her public key, which Bob will use for encrypting messages to Alice. The integer x is her secret decryption key.

Now, we are moving on to Bob's side. Suppose Bob wants to send some secret message $m \in \mathbb{Z}_p^*$ to Alice. Bob selects a secret integer y, $1 \leq y \leq p-1$. Bob then computes $B = g^y$ and $\alpha = A^y \bmod p$. Bob can compute these since both g and A are public. Next, Bob calculates $c = \alpha m \bmod p$. After Bob computes B and c, he sends the ciphertext message (B, c) to Alice. Alice receives (B, c) from Bob, then to find α, she computes $B^x \bmod p$. Indeed,

$$B^x = (g^y)^x = g^{yx} = g^{xy} = (g^x)^y = A^y = \alpha \bmod p.$$

Next, Alice computes α^{-1}, which allows her to decrypt Bob's message:

$$\alpha^{-1}c = \alpha^{-1}\alpha m = m.$$

Thus, we see that Alice can decrypt Bob's plaintext message.

Example 1.4. Suppose, Alice chooses the prime $p = 569$, $g = 34 \in \mathbb{Z}_{569}^*$, and $x = 368$. Alice computes $A = 34^{368} = 69 \bmod 569$. After Alice computes A, she publishes her public key $(p, g, A) = (569, 34, 69)$. Bob wants to send Alice the message $m = 215 \in \mathbb{Z}_{569}^*$. First, Bob chooses his $y = 21$. Then, Bob computes $B = 34^{21} = 26 \bmod 569$ and $\alpha = 69^{21} = 25 \bmod 569$. Bob encrypts the message $m = 215$ using α that he computed from previous step,

$$c = (25)215 = 254 \bmod 569.$$

Bob sends his ciphertext message as the ordered pair $(B, c) = (26, 254)$ to Alice. Alice computes $\alpha = B^x = 26^{368} = 25 \bmod 569$. Notice that she got same α as Bob. Alice finds $\alpha^{-1} = (B^x)^{-1} = B^{-1} = 478$ in mod 569. Now, she can recover m by $(478)(254) = 215 \bmod 569$.

Example 1.5. Now let's consider the situation where Denise wants to send Alice the same message $m = 215$ as Bob. Suppose Denise's secret number is $w = 25$. Denise calculates $D = 34^{25} = 458 \bmod 569$ and $\alpha = 69^{25} = 114$. Denise computes $c = (114)(215) = 43 \bmod 569$. Denise sends the ordered pair $(458, 43)$ to Alice. Note that is different from what Bob sent to Alice. Alice receives the ordered pair ciphertext message from Denise. Let's check if Alice will decode the message. Alice computes $\alpha = 458^{368} = 114 \bmod 569$. Notice that Alice got the same α as Denise. Next Alice finds $\alpha^{-1} = 5 \bmod 569$, which will allow Alice to recover the message by $m = (5)43 = 215 \bmod 569$.

1.3 Security of Classical ElGamal

In this section, we are going to look some security aspects of classical ElGamal scheme.

1.3.1 Chosen Ciphertext Attack

From the our examples above, we have seen two different individuals: Bob and Denise try to send the same message $m = 215$ to Alice. When Bob was sending message to Alice he sent $(26, 254)$; when Denise was sending message to Alice she sent $(458, 43)$. Alice receives different pairs from different individuals. This is great advantage. Alice will receive different ciphertext messages from different individuals.

Example 1.6. Suppose Eve saw Bob's message which looks like the following $(26, 254)$. Eve chooses fake message $f = 278$ and requests Alice to decrypt $(B, f) = (26, 278)$ back. Alice decrypts the message like she did above. Alice finds $\alpha = (26)^{368} = 25$, and $\alpha^{-1} = 478$. Alice now decrypts the fake message m by $(478)(278) = 307$. Then Alice sends the $u = 307$ to Eve. This allows Eve decodes the following: $m = u \cdot f^{-1} \cdot 254 = 215 \bmod 569$ which is a plaintext message! This is really dangerous. However, this is avoidable if Alice was careful and not response to Eve's message. If Alice receives $(26, 278)$ from Eve and Eve requests to decrypt message and give her back. Then, Alice could ignore Eve. Because Alice knew that (1) she already receives same B from Bob, (2) no one will ask her to decrypt and give them back. If Alice will decrypt back, then this will allow Eve to have chosen ciphertext message attack.

1.3.2 Brute Force

In a brute force attack Eve will simply try every single exponent x to find Alice's secret x. Eve knows that all the possible secret exponents x are $1 \leq x \leq p - 1$. However, Alice could avoid this problem by: (a) selecting a large prime number p and (b) choosing a primitive base g. We want to spend some time to explain why a primitive g is so important. When we have a large prime p, our maximum order of element $g \in \mathbb{Z}_p^*$ is $p - 1$. However, do we always guarantee an element $g \in \mathbb{Z}_p^*$ will have maximum order? Let's consider the following question. We know A is public information that published by Alice and this could lead Eve to create the following equation: $A = g^x \bmod p$ which g and p are public. What if g is not primitive Eve could find a z that makes $A = g^z = g^x$? When we have large prime p and primitive element g, this will lead us to the Discrete Log Problem.

The Discrete Log Problem: Find an algorithm that solves a generic equation $A = g^x \bmod n$ for x (given A, g and n) in a way that is provably more efficient than brute force.

The DLP is considered to be hard to solve when g has maximum order.

Example 1.7. Now let's go back to our Example 1.4 Alice chooses her $g = 34$ when we check the order of 34 in mathematica we can find $|g|$ is 142. Eve knows the public information so she can create the following formula $A = 34^x = 69 \in \mathbb{Z}_{569}^*$. Consider $z = 84$, then this will create the following situation $34^{84} = 34^x = 69 \in \mathbb{Z}_{569}^*$. Eve also observed Bob sends $(26, 254)$ to Alice. This allows her to compute $26^{84} = 25$ which is α. In other word, Eve figured it out Alice and Bob's secret key! Eve can decrypt message by finding $\alpha^{-1} = 478$, $m = 478(254) = 215$. This is dangerous. We want to avoid this situation, so we want to use primitive base g.

Example 1.8. Here is the example with adjusted g. Suppose Alice selects new base $g = 31$ which is primitive; the rest settings are same.

Notice that Eve can set same formula as example 1.4:

$$A = 31^x = 69 \mod 569.$$

However, since our base is primitive, x could be all the possible key $1 \leq x \leq 569$. There is no point, Eve would have to try every single key especially our prime p will be large which is even harder as p gets larger. Of course, Eve always can decide to spend time to brute force and wasting her time.

2 ElGamal in $\mathrm{GL_n}(\mathbb{F}_q)$

At this point, we have seen the classical ElGamal system work based in the abelian group $\mathbb{Z}_p^*$. This is a good encryption system with strong security but we hope to modify for adaptability to vectors, and some computation advantages. To this end, we will replace by group $\mathbb{Z}_p^*$ by $\mathrm{GL}_n(\mathbb{F}_q)$ which is the general linear group of invertible $n \times n$ matrices over the finite field $\mathbb{F}_q$. If $\mathbb{F}_q$ is any finite fields with q elements then $q = p^d$ where p is a prime, d is an integer. It turns out that any two finite field with the same order are isomorphic to each other [4]. You can find the proofs from Thm 8.2.1 in Roman's book.

2.1 Description of Modified Scheme

Here is our new procedure. Now we will change ElGamal encryption scheme by replacing the group $\mathbb{Z}_p^*$ with the group $\mathrm{GL}_n(\mathbb{F}_q)$. Let g be an element in $\mathrm{GL}_n(\mathbb{F}_q)$. For the classical ElGamal, we choose the x values such that $1 \leq x \leq p-1$. Now we can choose x that is any positive integer (at least temporarily).

We want to show how modified ElGamal in $\mathrm{GL_n}(\mathbb{F}_q)$ encryption scheme works in modified system. Alice first selects a $\mathrm{GL}_n(\mathbb{F}_q)$. Then, Alice selects an element $g \in \mathrm{GL}_n(\mathbb{F}_q)$, and Alice chooses her secret number x which is an integer. Alice calculates $A = g^x \in \mathrm{GL}_n(\mathbb{F}_q)$ using her secret number x. Finally, Alice publishes her public encryption key (n, q, A, g).

Our message $\mathbf{m}$ will be any n-component column vectors over $\mathbb{F}_q$. Suppose Bob wants to send this message to Alice. Bob first chooses his secret integer y. Then, Bob computes $B = g^y$ in $\mathrm{GL_n}(\mathbb{F}_q)$. Next, Bob computes $\alpha = A^y$ and $\mathbf{c} = \alpha\mathbf{m}$. Thus, Bob sends $(B, \mathbf{c})$ to Alice. Alice receives the message $(B, \mathbf{c})$ from Bob. She computes $\alpha = B^x$ using her secret key x. Indeed, $\alpha = B^x$ because we have the following:

$$B^x = (g^y)^x = g^{yx} = g^{xy} = (g^x)^y = A^y = \alpha.$$

Note that Alice discovers same α as Bob. Because α is guranteed to be in $\mathrm{GL}_n(\mathbb{F}_q)$ so α^{-1} exists. Then Alice decodes plaintext by computing $\alpha^{-1}\mathbf{c} = \alpha^{-1}\alpha\mathbf{m} = \mathbf{m}$. Thus, we can see this encryption system works.

Theorem 2.1. *Suppose that Alice chooses a matrix $g \in \mathrm{GL_n}(\mathbb{F}_q)$ and an integer x. Assume the matrix $A = g^x$ is made public, along with the group $\mathrm{GL_n}(\mathbb{F}_q)$ and the matrix g. Given an n-component column vector $\mathbf{m}$ with entries in $\mathbb{F}_q$, Bob may encrypt $\mathbf{m}$ into another vector $\mathbf{c}$ by the following process. First, Bob selects an integer y. Next, Bob computes $B = g^y$ and $\alpha = A^y$. Then, Bob computes $\mathbf{c} = \alpha\mathbf{m}$. Finally he sends ciphertext message as an ordered pair $(B, \mathbf{c})$ to Alice. Alice receives the ciphertext message from the Bob. First, she computes $\alpha = B^x$. Next, she finds α^{-1}. Then, she decrypts $\alpha^{-1}\mathbf{c} = m$.*

Example 2.2. Here is an example of our new scheme. First Alice chooses $n = 3$ and $q = 7$, $g = \begin{pmatrix} 2 & 4 & 6 \\ 5 & 5 & 2 \\ 1 & 1 & 0 \end{pmatrix} \in \mathrm{GL}_3(\mathbb{Z}_7)$, and $x = 6$. Alice computes $A = \begin{pmatrix} 2 & 4 & 6 \\ 5 & 5 & 2 \\ 1 & 1 & 0 \end{pmatrix}^6 = \begin{pmatrix} 0 & 0 & 1 \\ 3 & 4 & 4 \\ 4 & 1 & 4 \end{pmatrix}$. Then Alice publishes her information to public; as previous example she holds $x = 6$ for the secret. The following box shows Alice's public information.

Alice
$\mathrm{GL}_3(\mathbb{Z}_7)$
$A=\begin{pmatrix}0&0&1\\3&4&4\\4&1&4\end{pmatrix}$
$g=\begin{pmatrix}2&4&6\\5&5&2\\1&1&0\end{pmatrix}$

Now we move on Bob's side, Bob wants to send the following message to Alice. Consider the following:

$$\mathbf{m}=\begin{pmatrix}1\\1\\2\end{pmatrix}.$$

Bob chooses his secret $y=27$, then he calculates $B=g^{27}$ and $\alpha=A^{27}$. Bob gets the following results.

$$B=\begin{pmatrix}5&2&3\\6&3&1\\4&4&4\end{pmatrix}\quad \alpha=\begin{pmatrix}5&1&3\\4&3&3\\1&0&0\end{pmatrix}.$$

Next, Bob computes c by the following step:

$$\mathbf{c}=\alpha\mathbf{m}=\begin{pmatrix}5&1&3\\4&3&3\\1&0&0\end{pmatrix}\begin{pmatrix}1\\1\\2\end{pmatrix}=\begin{pmatrix}5\\6\\1\end{pmatrix}.$$

Bob sends ciphertext message $(B,\mathbf{c})$ to Alice. Alice computes $\alpha=B^x=B^6$ and this allows her to find α^{-1}. The result is the following:

$$\alpha=B^6=\begin{pmatrix}5&1&3\\4&3&3\\1&0&0\end{pmatrix}\quad \alpha^{-1}=\begin{pmatrix}0&0&1\\3&4&4\\4&1&4\end{pmatrix}.$$

Notice that Alice gets the same α as Bob, which is just as like in our classical ElGamal scheme. Now, Alice computes m by the following:

$$\mathbf{m}=\alpha^{-1}\mathbf{c}=\begin{pmatrix}0&0&1\\3&4&4\\4&1&4\end{pmatrix}\begin{pmatrix}5\\6\\1\end{pmatrix}=\begin{pmatrix}1\\1\\2\end{pmatrix}.$$

We checked that our encryption system works in the same way as the classical ElGamal Scheme.

Example 2.3. Now we are moving to Denise. Suppose Denise wants to send the same message as Bob,

$$\mathbf{m}=\begin{pmatrix}1\\1\\2\end{pmatrix}.$$

Denise chooses her secret $y=14$, then she calculates $D=g^{14}$ and $\alpha=A^{14}$. Then, Denise gets following results.

$$D = \begin{pmatrix} 2 & 2 & 5 \\ 5 & 2 & 2 \\ 5 & 6 & 3 \end{pmatrix} \alpha = \begin{pmatrix} 4 & 1 & 4 \\ 0 & 6 & 0 \\ 5 & 1 & 3 \end{pmatrix}.$$

Next, Denise computes c by the following step.

$$\mathbf{c} = \alpha\mathbf{m} = \begin{pmatrix} 4 & 1 & 4 \\ 0 & 6 & 0 \\ 4 & 1 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 6 \\ 5 \end{pmatrix}.$$

Denise sends ciphertext message $(D, \mathbf{c})$ to Alice. First, Alice computes $\alpha = D^x = D^6$ and this allows her to find α^{-1}. The result is the following:

$$\alpha = D^6 = \begin{pmatrix} 4 & 1 & 4 \\ 0 & 6 & 0 \\ 5 & 1 & 3 \end{pmatrix} \alpha^{-1} = \begin{pmatrix} 4 & 1 & 4 \\ 0 & 6 & 0 \\ 5 & 1 & 3 \end{pmatrix}.$$

Notice that Alice gets the same α as Denise. This also shows that Alice will still get different ciphertext messages from different individuals sending the same message m. Let's check if Alice can recover the message m. Now, Alice computes m by the following:

$$\mathbf{m} = \alpha^{-1}\mathbf{c} = \begin{pmatrix} 4 & 1 & 4 \\ 0 & 6 & 0 \\ 5 & 1 & 3 \end{pmatrix} \begin{pmatrix} 6 \\ 6 \\ 5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}.$$

We have seen that our ElGamal in $\mathrm{GL}_n(\mathbb{F}_q)$ has the same advantage as the classical ElGamal encryption system. Two different individuals will send different ciphertext messages for the same message m. This will enable Alice to avoid Eve's chosen chiphertext attack.

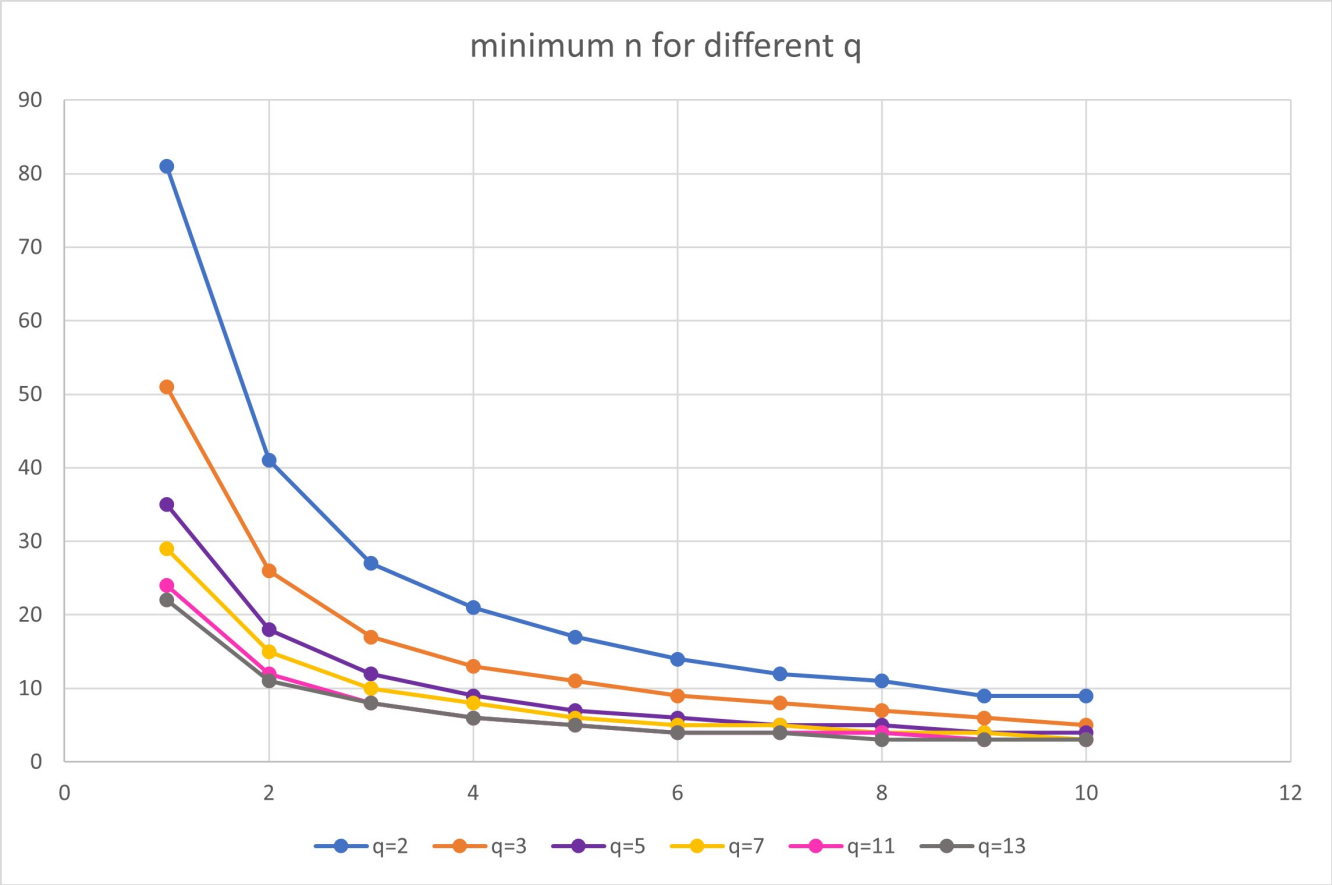
2.2 Security

From the previous section we were able to see that our modified scheme works, and it has same advantage as classical ElGamal scheme to resist chosen cipher text attack. Now, we want to check some security aspect of ElGamal in $\mathrm{GL}_n(\mathbb{F}_q)$. Does our system resist brute force? In other words, can we guarantee as sufficiently large key size? Also, one of great advantages of classical ElGamal was Discrete Log problem that is currently hard to solve.

2.2.1 Brute Force

To make sure our modified scheme is secure. We want to check key size. In other word, we want to find what is the maximum order element in $\mathrm{GL_n}(\mathbb{F}_q)$. Here is the well defined answer for how many potential keys in ElGamal in $\mathrm{GL}_n(\mathbb{F}_q)$. Consider the following theorem.

Proposition 2.4. *The maximum possible order of an element in* $\mathrm{GL}_n(\mathbb{F}_q)$ *is* $q^n - 1$ *and elements of this order always exist.*



Proof. We can found in example, in M.R Darafsheh's paper [2]. □

Now, we know how many keys we have available when we work with $\mathrm{GL}_n(\mathbb{F}_q)$.

We want to ask this question for security of new protocol. Suppose q is fixed, and we are willing to vary n. We want to find the minimum matrix size n such that $q^n - 1$ is greater or equal to given large number for example 2^{80}. We will use 2^{80} as a threshold of big key size. In other words, modified scheme starts to act same as classical ElGamal. Suppose, p is fixed, what is the minimum n to make the maximum order exceed 2^{80}?

We can look at our result with the graph. This graph in page 7 explains us that when we have a bigger q we require smaller matrix size. This is also reducing required n pretty fast. For example, when $q = 2$ our scheme requires 81×81 matrix to have larger key size, when $q = 8$ (Which is blue line, $d = 3$) it only requires 8×8 matrix. We will ask one more security question. This time we fixed $q = p^d$. If q is fixed, what is the minimum power d to exceed 2^{80}. We want to find that what is the smallest matrix size that will start act same as classical ElGamal. We can calculate this by the following formula:

$$\begin{aligned} p^d &\geq 2^{80} \\ d \log p &\geq 80 \log 2 \\ d &\geq 80 \frac{\log 2}{\log p}. \end{aligned}$$

Here is a chart for the required minimum power d for prime number smaller than 100.

p	minimum d
2	81
3	51
5	35
7	29
11	23
13	22
17	20
19	19
23	18
29	17
31	17
37	16
41	15
43	15
47	15
53	14
59	14
61	14
67	14
71	14
73	13
79	13
83	13
89	13
97	13

When we look our graph and chart, we can see that as p or q increases the matrix size is getting smaller. Let's look at some examples.

Example 2.5. Here we will look at the case when $p = 5$.

When $p = 5$ then we have the following chart.

d	$q = 5^d$	minimum n
1	5	35
2	25	18
3	125	9
4	625	7
5	3125	6
6	15625	5
7	78125	5
8	390625	4
9	1953125	4
10	9765625	4

Here is how to interpret these charts. Suppose we want to use $\mathrm{GL}_n(\mathbb{F}_{5^d})$ and make sure our key size will be big enough. We can decide to use bigger field such as 3125 then our required matrix size is getting smaller, if we decide to smaller field then our suggested matrix size is getting bigger.

Overall, these chart will help you decide if you want to work on bigger matrix or bigger field. We can efficiently reduce the field size using by larger matrix, or if we willing we can reduce the order of our field by using larger matrix.

You can find other rest chart with different q values from the Appendix A.

We can check our modified scheme will be flexible to have smaller filed size if we have larger size of matrix n. If we can find an element g that has maximum order then we will be able to resisting brute force.

2.2.2 Determinant

Let's consider our example 2.2. Since we are in group $\mathrm{GL}_3(\mathbb{Z}_7)$, we will have pretty big size of the group, and key size is $7^3 - 1 = 342$. But our field size is $q = 7$. In other words, if Eve decide to take determinant which will be an element in $\mathbb{Z}_7^*$, isn't that possibly leak some of our key information? Since the size of group will be smaller.

Example 2.6. Eve knows the following information:

Alice
$\mathrm{GL}_3(\mathbb{Z}_7)$
$A=\begin{pmatrix}0&0&1\\3&4&4\\4&1&4\end{pmatrix}$
$g=\begin{pmatrix}2&4&6\\5&5&2\\1&1&0\end{pmatrix}$

Then Eve can creates following formula by those information like she did in classical ElGamal. Consider the following:

$$A = g^x \in \mathrm{GL_n}(\mathbb{F}_q) \begin{pmatrix}0&0&1\\3&4&4\\4&1&4\end{pmatrix} = \begin{pmatrix}2&4&6\\5&5&2\\1&1&0\end{pmatrix}^x$$

In our example, our g has order of 24. It was not an element g that has maximum order. When Eve creates first relationship, she needs to check 24 different keys. However, what will be happen if Eve decide to take a determinant?

$$\begin{aligned}\det(A) &= \det(g)^x \\ 1 &= 4^x.\end{aligned}$$

Eve knows that 4 is non-primitive element in $\mathbb{Z}_7$, and has order 3. So, Eve can find the x is multiple of 3. For our Example 2.1 and current example g has order of 24. So, Alice could choose her secret number x such that $1 \leq x \leq 24$ which you would never do. This allows Eve limiting her choice by $3x \leq 24$, the possible x will be 3, 6, 9, 12, 15, 18, 21, 24. The chart below shows what if Eve calculates all the possibilities that she figured it out (Eve could stop at x=6).

Table 1

x	g^x
3	$\begin{pmatrix} 5 & 2 & 3 \\ 6 & 3 & 1 \\ 4 & 4 & 4 \end{pmatrix}$
6	$\begin{pmatrix} 2 & 4 & 6 \\ 5 & 5 & 2 \\ 1 & 1 & 0 \end{pmatrix}$
9	$\begin{pmatrix} 4 & 4 & 4 \\ 6 & 6 & 1 \\ 0 & 6 & 1 \end{pmatrix}$
15	$\begin{pmatrix} 0 & 6 & 1 \\ 1 & 4 & 6 \\ 1 & 4 & 0 \end{pmatrix}$
18	$\begin{pmatrix} 5 & 1 & 3 \\ 4 & 3 & 3 \\ 1 & 0 & 0 \end{pmatrix}$
21	$\begin{pmatrix} 1 & 4 & 0 \\ 1 & 1 & 6 \\ 5 & 2 & 3 \end{pmatrix}$
24	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Also, we can quickly check that Bob was choosing wrong number, $y = 27$ is actually $y = 3$.

Our g has order 24 and determinant of 4, we have three different cases by selecting different x. Then can we be possibly predict how many cases we will loose if Alice is choosing wrong g? Before we move on the theorem to figure out how many cases Alice will lose by choosing a wrong g. We want to consider following Theorem.

Theorem 2.7. *Let G be a group with an element $a \in G$ that has finite order, let $n = \mathrm{ord}(a)$ Then, $a^k = e$ iff $n|k$.*

Proof. Suppose forst that $n|k$ then we can rewrite $k = n \cdot m$ for some integer m. Then,

$$a^k = a^{mn} = (a^n)^m = e^m = e.$$

Suppose conversely let $a^k = e$, we will long divide k/n to get $k = nq + r$ where $0 \geq r \geq n - 1$. Let's compute the following:

$$a^r = a^{k-nq} = a^k a^{-nq} = e(a^n)^{-q} = e.$$

As $n = \mathrm{ord}(a)$ and $0 \geq r \geq n - 1$, we have $r = 0$. Hence, $k = nq$. So, $n|k$. □

We will use this theorem to prove the following theorem.

Proposition 2.8. *Suppose $g \in \mathrm{GL}_n(\mathbb{F}_q)$, then $\frac{|g|}{|\det(g)|}$ is always an integer.*

Proof. Suppose $g \in \mathrm{GL}_n(\mathbb{F}_q)$ where $\mathbb{F}_q$ is a finite field and n is a positive integer. Let α be a function that refers α: take a determinant of g, $\alpha : \mathrm{GL}_n(\mathbb{F}_q) \to \mathbb{F}_q^*$. Consider the following:

$$\det(g_1 g_2) = \det(g_1) \det(g_2)$$

for all $g_1, g_2 \in \mathrm{GL_n}(\mathbb{F}_q)$. This works since there is no matrix $\det(g) = 0$ by definition of $\mathrm{GL_n}(F_q)$. Thus, α is a homomorphism. We know g has a finite order since $\mathbb{F}_q$ is a finite field. We can re-write $g^k = I_n$ where k is an integer. Then we have the following.

$$\begin{aligned}\det(g^k) &= (\det(g))^k \\ \det(\mathbb{I}_n) &= (\det(g))^k \\ 1 &= (\det(g))^k.\end{aligned}$$

Thus, $\alpha(g^k)$ has a finite order. By little Lagrange Theorem, $\frac{|g|}{|\det(g)|}$ is always an integer. □

For the summary, if we have $\det(g)$ is not primitive and $|g|$ is not maximum then Eve will have less case to brute force. Here is one question we can think of, is any element g that has maximum order and determinant of g is primitive?

2.3 Maximum Order Example

Suppose we are in $g = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 2 & 1 \end{pmatrix} \in \mathrm{GL}_3(\mathbb{Z}_3)$ has maximum order. From the previous equation the maximum order of $g \in \mathrm{GL}_3(\mathbb{Z}_3)$ is $3^3 - 1 = 26$.

x	g^x	x	g^x	x	g^x	x	g^x	x	g^x
2	$\begin{pmatrix} 2 & 2 & 2 \\ 1 & 2 & 2 \\ 1 & 0 & 2 \end{pmatrix}$	3	$\begin{pmatrix} 0 & 2 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix}$	4	$\begin{pmatrix} 2 & 2 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix}$	5	$\begin{pmatrix} 2 & 1 & 1 \\ 2 & 2 & 1 \\ 2 & 0 & 2 \end{pmatrix}$	6	$\begin{pmatrix} 0 & 2 & 2 \\ 1 & 0 & 2 \\ 1 & 0 & 0 \end{pmatrix}$
7	$\begin{pmatrix} 0 & 0 & 2 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$	8	$\begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 2 & 2 \end{pmatrix}$	9	$\begin{pmatrix} 2 & 0 & 1 \\ 2 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix}$	10	$\begin{pmatrix} 2 & 1 & 2 \\ 1 & 0 & 0 \\ 2 & 2 & 0 \end{pmatrix}$	11	$\begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \\ 2 & 1 & 1 \end{pmatrix}$
					...				
22	$\begin{pmatrix} 1 & 0 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	23	$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$	24	$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix}$	25	$\begin{pmatrix} 2 & 0 & 2 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$	26	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

There is an algorithmic way to find an element g that has both maximum order and primitive determinant, which we did not investigate in our thesis. Still, here is simple suggestion, we are changing group again to $\mathrm{SL}_n(\mathbb{F}_q)$. This will make Eve facing $1 = 1^x \in \mathbb{Z}_p^*$ which every x solves this relationship. Here is the definition for $\mathrm{SL_n}(\mathbb{F}_q)$.

Definition 2.9. Let $\mathrm{SL_n}(\mathbb{F}_q)$ denote set of all $n \times n$ matrices A with $\det(A) = 1$. NOTE: Also, $\mathrm{SL_n}(\mathbb{F}_q)$ is known as the *special linear group.*

Then, consider the following proposition.

Proposition 2.10. *Let $\mathbb{F}_q$ be a finite field then $\mathrm{SL_n}(\mathbb{F_q})$ is a normal subgroup of $\mathrm{GL_n}(\mathbb{F}_q)$.*

Proof. To prove $\mathrm{SL_n}(\mathbb{F_q})$ is a normal subgroup of $\mathrm{GL_n}(\mathbb{F}_q)$ we will prove the following: (1) det : $\mathrm{GL_n}(F_q) \to \mathbb{F}_q^*$ (2) $SL_n(\mathbb{F}_q)$ is kernel of determinant. We already proved (1) from prop 2.7. It is obvious that $\mathrm{SL_n}(\mathbb{F}_q)$ is kernel of determinant.

Therefore, $\mathrm{SL_n}(\mathbb{F}_q)$ is normal subgroup of $\mathrm{GL_n}(\mathbb{F_q})$. □

We now want to change our group into $\mathrm{SL_n}(\mathbb{F}_q)$, this will allow Eve faces the following situation:

$$\det(A) = \det(g)^x$$
$$1 = 1^x.$$

Where every $1 \leq x \leq |g|$ will be the possible key, and strengthen our scheme with harder discrete log problem.

3 ElGamal in $\mathrm{SL_n}(\mathbb{F}_q)$

With our new modified system, we do not need to check validity of new modified system since $\mathrm{SL_n}(\mathbb{F}_q)$ is a subgroup of $\mathrm{GL}_n(\mathbb{F}_q)$. The reason is that it will work exactly as the modified scheme $\mathrm{GL_n}(\mathbb{F}_q)$.

3.1 Security

We checked our new modified scheme will work exactly same as first modified scheme. That tells us we will have quite a strong security based on the same advantage as before. Our new modified scheme will resist the chosen ciphertext attack well since our new modified scheme will have same advantage as classical ElGamal and modified ElGamal. Also, we figured it out problem which Eve can help disappears since

$$\det(A) = \det(g)^x$$
$$1 = 1^x.$$

Now our determinant will not leak any the information about x. We now want to check the key size like we did with $\mathrm{GL_n}(\mathbb{F}_q)$. Here is one proposition:

Theorem 3.1. *When $n > 2$, the maximum possible order of an element in $\mathrm{SL}_n(\mathbb{F}_q)$ is $\frac{q^n-1}{q-1}$ and elements of this order always exist.*

Proof. We can find the proof of this proposition from M.R Darafsheh's paper [2]. □

Previous Theorem showed that the maximum possible order of g is $\frac{q^n-1}{q-1}$. Our key size is reduced a bit, but only by factor of $q-1$.

3.2 Maximum Order Example

Suppose we are in $\mathrm{SL}_3(\mathbb{Z}_3)$ then $g = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ has maximum order. From the previous equation the maximum order of $g \in \mathrm{SL}_3(\mathbb{Z}_3)$ is $(3^3-1)/(3-1) = 13$.

x	g^x	x	g^x	x	g^x	x	g^x	x	g^x
1	$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	2	$\begin{pmatrix} 2 & 1 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}$	3	$\begin{pmatrix} 2 & 2 & 1 \\ 2 & 1 & 2 \\ 0 & 2 & 2 \end{pmatrix}$	4	$\begin{pmatrix} 2 & 1 & 0 \\ 2 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix}$	5	$\begin{pmatrix} 0 & 0 & 2 \\ 2 & 1 & 0 \\ 2 & 2 & 0 \end{pmatrix}$
6	$\begin{pmatrix} 2 & 2 & 2 \\ 0 & 0 & 2 \\ 1 & 0 & 2 \end{pmatrix}$	7	$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 2 & 2 \\ 0 & 2 & 0 \end{pmatrix}$	8	$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 2 & 1 \\ 2 & 0 & 0 \end{pmatrix}$	9	$\begin{pmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 2 & 0 & 2 \end{pmatrix}$	10	$\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix}$
11	$\begin{pmatrix} 2 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 2 \end{pmatrix}$	12	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix}$	13	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$				

This example illustrates the situation when our second modified scheme is most secure; our element g has the maximum order (and the determinant does not leak any information).

A Appendix A

The calculations of the minimum matrix size so that, for a give g, the maximum order of an element g in $\mathrm{SL_n}(\mathbb{F}_q)$ is at least 2^{80} (See chapter 2 for the analogous calculation for $\mathrm{GL_n}(\mathbb{F}_q)$). Suppose $q = p^d$ where d is positive integer and n is minimum matrix size such that $q^n - 1 \geq 2^{80}$.

When $p = 2$ then we have the following chart.

d	$q = 2^d$	minimum n
1	2	81
2	4	41
3	8	27
4	16	21
5	32	17
6	64	14
7	128	12
8	256	11
9	512	9
10	1024	9

When $p = 3$ then we have the following chart.

d	$q = 3^d$	minimum n
1	3	51
2	9	26
3	27	17
4	81	13
5	243	9
6	729	8
7	2187	7
8	6561	6
9	19638	6
10	59049	5

When $p = 7$ then we have the following chart.

d	$q = 7^d$	minimum n
1	7	29
2	49	15
3	343	10
4	2401	8
5	16807	6
6	117649	5
7	8235	5
8	5764801	4
9	40353607	4
10	282475249	3

When $p = 11$ then we have the following chart.

d	$q = 11^d$	minimum n
1	11	24
2	121	12
3	1331	8
4	14641	6
5	161051	5
6	1771561	4
7	19487171	4
8	21438881	4
9	2357947691	3
10	25937424601	3

When $p = 13$ then we have the following chart.

d	$q = 13^d$	minimum n
1	13	22
2	169	11
3	2197	8
4	28561	6
5	371293	5
6	4826809	4
7	62748517	4
8	815730721	3
9	10604499373	3
10	137858491849	3

References

[1] J. L. Alperin, *Book Review: Group theory*, Bull. Amer. Math. Soc. (N.S.) **17** (1987), no. 2, 339–340. MR 1567639

[2] M. R. Darafsheh, *Order of elements in the groups related to the general linear group*, Finite Fields Appl. **11** (2005), no. 4, 738–747. MR 2181417

[3] Taher ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Trans. Inform. Theory **31** (1985), no. 4, 469–472. MR 798552

[4] Steven Roman, *Field theory*, second ed., Graduate Texts in Mathematics, vol. 158, Springer, New York, 2006. MR 2178351