A Reinforced ElGamal Scheme Proposal
Against a Pohlig-Hellman Attack

R. Flores-Carapia
Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional, México
rfcarapia@yahoo.com

V. M. Silva-García
Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional, México
vsilvag@ipn.mx

M. D. González-Ramírez
Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional, México
dgonzalezr@ipn.mx

C. Rentería-Márquez
Escuela Superior de Física y Matemáticas
Instituto Politécnico Nacional, México
crenteriam@gmail.com

Copyright © 2013 R. Flores-Carapia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Multiplicative group \((G, \cdot)\) where \(G = Z_p^* = Z_p - \{0\}\), the primitive element calculation, say \(\alpha\), is usually constructed from the following theorem: given a prime \(p > 2\) and \(\alpha \in Z_p\), which is a primitive or generator element of \(Z_p^*\) if and only if \(\alpha^{(p-1)/q} \neq 1 \mod p, \forall \text{ prime } q\) such that \(q \mid (p - 1)\). Furthermore, the Dirichlet theorem shows that:
given two integers $a, b$ with $\gcd(a, b) = 1$, that is, $a, b$ are relatively prime to each other, then, an infinite number of primes can be constructed in accordance with the formula $n(a) + b$, where $n = 1, 2, \ldots$. In this work high primality has the following meaning: given a number $p > 2$, it is said to be prime with an error at most $1/2^{100}$. Now, using the two theorems cited above and taking as particular values $b = 1$ and $a = p_1p_2$ where $p_1, p_2$ are two high primality numbers; a high primality number $p$ can be built which has the form $p = (m)p_1p_2 + 1$. with $m = 2n$ for $n = 1, 2, \ldots$, eliminating those that fulfill $p \equiv 0 \mod 5$. To construct positive integers $p$-form with high-primality intents that $p - 1$ can be factorized in a simple way. This research produced 700000 positive $p$-form integers. The result is that 88.5% of them have a $m$ less or equal to 2000; it is pointed out that $p_1, p_2$ are $10^{200}$ approximately. Then, one can obtain a primitive $\alpha$ easily, and also a double lock is provided with regard to a Pohlig-Hellman form attack. The first one is the $p - 1$ factorization and the second, solving the discrete logarithm problem when the $p - 1$ factors are known.

**Keywords:** Dirichlet Theorem, Pohlig-Hellman Algorithm, RSA Cryptosystem Factoring, Discrete Logarithm Problem, Primality Testing, Index Calculus Algorithm.

1 Introduction

In this work the ElGamal cryptosystem scheme is built on a multiplicative group $(G, \cdot)$ with $G = \mathbb{Z}_p^*$, in fact, the set $\mathbb{Z}_p^* = \{\alpha^i \mid 0 \leq i \leq p - 2\}$ where $\alpha$ is a $p - 1$ order element mod $p$ [7], the $\alpha$ element is also called a generator. On the other hand, there are general-type attacks to the discrete logarithm in finite fields, including: the Shank Algorithm [9], Pollard Rho [11] and the Pohlig-Hellman Algorithm [12]. However, the Pohlig-Hellman Algorithm attack requires factoring $p - 1$ [9], where $p$ prime and $p > 2$. Clearly, it is simpler to construct a $p$ prime such that, factoring $p - 1$ could be easier than choosing a random $p$ prime and then trying to factor $p - 1$. In the latter scenario, there is a risk of facing a factorization computationally intractable problem [3]. This is the reason why in this paper we propose the construction of high primality numbers of the form $p = (m)p_1p_2 + 1$. As a small illustration, the prime $p = 585798114457$ can be written in terms of the product $p_1 = 88241$ and $p_2 = 92201$ primes and the even number 72, that is, $p = (2^3)(3^2)(88241)(92201) + 1$. Then, if two numbers are chosen large enough with high primality, say $p_1, p_2 \approx 10^{200}$ a $p$ prime is constructed from them, also with high primality as mentioned above. With regard to a Pohlig-Hellman Algorithm attack it is necessary to solve the problems of factorization and the discrete logarithm, $\log_{\alpha} \beta = \alpha$ [9]. As discussed in the course of this investiga-
tion both problems have no solution from a computational viewpoint, at least so far. Thus, when producing the $p$-form numbers, the probability is that the even number $m \leq 2000$ is 0.885 approximately. The foregoing, suggests that if 4 type numbers are built as $p$, independently of one another, the probability is that at least one of them fulfills $m \leq 2000$ is 0.9999 approximately.

2 The proposal and ElGamal Cryptosystem
attacks

One question to be answered is how will we know if $p_1, p_2$ and $(m)p_1p_2 + 1$ are high primality numbers? A Monte Carlo scheme for composite numbers will be used, which is called the Miller-Rabin Algorithm [8]. The above scheme performs a random type test to determine whether a given number is a positive integer composite or not. If the answer is yes, then this is undoubtedly composite, but if the answer is no, then the error in this case is at most $1/4$ [8].

However, since in each test a pseudo-random number is chosen thus each test is considered independent of each other, then the error can be reduced as much as desired. In this research the error will be at most $1/2^{100}$. In this vein, it can be considered that the numbers working here have a very high primality. Another question that may arise is how to make $p = (m)p_1p_2 + 1$ have a termination of 1, 3, 7 or 9? The answer is simple, since the product $p_1p_2$ is odd it follows $(m)p_1p_2$ is even, therefore $(m)p_1p_2 + 1$ is odd. The way to ensure the termination has to be 1, 3, 7 or 9 is to eliminate all the odd numbers fulfilling the expression $(m)p_1p_2 + 1 \equiv 0 \mod 5$. With regard to the question why the even number, $m = 2n$ for $n = 1, 2, \ldots$ does not take values beyond 2000, before finding a high primality $p$ in the majority of attempts performed? The answer will be given in the results section according to the 700000 runs of the $p$-form; and later a reporting percentage.

This paper divides the attacks that are conducted by the ElGamal cryptosystem into two types, namely: generic and specific. The generic class is applicable to any finite field, for example: Shank, Pollard Rho and Pohlig-Hellman Algorithms. The specific applies to a particular type of group, for example, the multiplicative group $Z_p^* = Z_p - \{0\}$ [2].

The Index Calculus Method applies in the above case [1]. This is a process consisting of two parts, namely: the first phase is called pre computation and asymptotic run time is $O(e^{(1+o(1))\sqrt{\ln\ln\ln p}})$ [9]. This clarifies that $o(1)$ is an amount that approaches zero when $p$ is getting bigger. In the second phase
the discrete logarithm for the particular number is searched and asymptotic run time is $O(e^{(1/2+o(1))\sqrt{\ln p\ln \ln p}})$ [9].

Practical case studies consider that if $p > 10^{300}$, the Elgamal scheme is safe when carrying out an attack on the Index Calculus Method type [9]. Here we propose positive integer’s $p \approx 10^{400}$ size; however, following the procedure proposed $p$ can be calculated even greater with higher primality.

The Pohlig-Hellman attacks start with the factorization of $p - 1$, so that an attacker has the problem of finding the factors $p_1p_2$ product. The $m$ factorization is very simple, since in the majority of times 47 will be the bound of its factors. The appropriate algorithm to solve this problem is: number field sieve [9], because $p_1 \approx p_2$. The asymptotic time used by this algorithm for $n = p_1p_2$ is $O(e^{(1.98+o(1))(\ln(n)^{2/3})(\ln\ln(n)^{2/3})})$, where $o(1)$ approach to zero as $n$ becomes increasingly large [9].

Recent attacks show factorizations of $n = 10^{212}$ [3]. In this sense if a Pohlig-Hellman attack is performed, first it must solve the factorization of $n = p_1p_2$ that is an unsolved problem at this time, computationally speaking, and according to the conditions for $p_1, p_2$ as given above.

A Pohlig-Hellman Algorithm attack begins with the fact that $p - 1 = \prod_{i=1}^{k} p_i^{c_i}$ where the $p_i$ are the prime factors of $p - 1$. In the ElGamal scheme the exponent of $\alpha$, the generator element, is denoted as $a$ and what is sought, is calculating a mod $p_i^{c_i}$ for each $1 \leq i \leq k$. Then, we can use the Chinese Remainder Theorem [10], calculating a mod $p - 1$. Considering that the high primality number is constructed as $p = (m)p_1p_2 + 1$, where $p_1, p_2$ are also positive integers with high primality around of $10^{200}$ each; the discrete logarithm problem to be solved has size $O(\sum_{i=1}^{k} c_i(\sqrt{p_i}))$ [1]. Clearly, factorizing $m$ is simple since it is small and can therefore be disregarded. Then, the size of the problem to be solved is $O(\sum_{i=1}^{2} \sqrt{p_i})$. It follows then that for this particular case is $O(\sum_{i=1}^{2} \sqrt{p_i}) \approx O(10^{100})$. It can be seen that this problem is intractable at this time.

With regard to Shank and Pollard Rho Algorithm attacks, the first employs $O(\sqrt{p - 1})$ and the second $O(\sqrt{p - 1})$ so they are not recommended.

3 Presentation of results

This section provides answers to the question: is the even positive integer $m$ of $(m)p_1p_2 + 1$ less than 2000 most of the time before a number with high
primality can be found? As mentioned in the previous section the percentage of several runs will be shown.

In this research we proceed as follows: in a random way 700000 numbers pairs of high primality, \( \{p_1, p_2\}_j \) are chosen with \( 1 \leq j \leq 700000 \). To make this process a prime number multiplied by \( \pi \) is chosen [5], then constructs a number taking 200 digits after the decimal point and its termination in 1, 3, 7 or 9 is written. We denote this positive integer as \( p_p \). Then we look for the number closest to \( p_p \) with high primality and greater or equal to it.

With the 700000 pairs \( \{p_1, p_2\}_j \) where \( 1 \leq j \leq 700000 \) the following operation \((m)p_1p_2 + 1\), where \( m = 2n \) and \( n = 1, 2, \ldots \) is performed. It seeks the smaller \( m \) that makes \((m)p_1p_2 + 1\) a high primality number. It reports the percentage of these pairs where, \( m \) was less or equal to 2000, and in a simple manner the percentage where they were bigger than 2000.

Table 1 reports the results of the \( \{p_1, p_2\}_j \) pairs, also as illustrated it shows an example with particular values.

<table>
<thead>
<tr>
<th>Numbers with high Primality</th>
<th>( p_1, p_2 )</th>
<th>Percentage ( m \leq 2000 )</th>
<th>Percentage ( m &gt; 2000 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((m)p_1p_2 + 1)</td>
<td>700000</td>
<td>88.5%</td>
<td>11.5%</td>
</tr>
</tbody>
</table>

The example with specific data is provided below:

\[
p_1 = 917146757521065534521747732113012387489349379005185080619620 \\
314556549413528977677123865252804012512279322506987264676311 \\
631131335034832591300669899942951256595408853841606788753390 \\
087245165486387905487
\]

\[
p_2 = 618530878154473121184477822428917501751401792452699129576539 \\
919783566602236026498570136053369640209616440917626787222529 \\
885470128916341163767078499126313686181612602495751525169962 \\
17563310436718957547
\]

\[ m = 112 \]

\[
(m)p_1p_2 + 1 = 6353576200451561656755917944219113323212948943702393 \\
1187213805252859481317613549337260537722594641145947 \\
3922902123102132699420521652723509691549855223543842 \\
7791809441858024391649476690225015861852649645029200
\]
Moreover, for practical cases it is useful to know what is the minimum number of \( \{p_1, p_2\} \) pairs ensuring high probability, say, 0.9999 that at least one of the integers constructed on the form \((m)p_1p_2 + 1\) will have high primality, with the \(m\) less or equal to 2000. With this goal in mind we can start with some facts, namely, let the random variable \(X = \{0, 1\}\), according to the following rule: \(X = 1\) if the number \(m\) is less than or equal to 2000, otherwise \(X = 0\). The \(X\) is a Bernoulli random variable \([6]\). Also, the estimated value of the probability \(p_r\) is that the integer \(m\) is less or equal to 2000, is taken from table 1, i.e, \(p_r = 0.885\). Thus, if we have \(l\) Bernoulli events, with \(l = 1, 2, \ldots\) they are considered independent to each other \([6]\). Therefore, the random variable defined as: \(Y = \sum_{i=0}^{l} X_i\) adjusts to a binomial model, i.e, \(X \sim B(y; l, p_r)\) \([14]\). Then it follows that:

\[
P[Y = y] = \binom{l}{y} p_r^y (1 - p_r)^{l-y}
\]

Using the above expression it is simple to find the \(l\) value that satisfies the condition mentioned later, since for \(l = 4\) we have \(P[Y \geq 1] = 1 - \binom{4}{0} (0.885)^0 (0.115)^4 \approx 0.9999\).

4 Conclusions

This paper constructs an asymmetric cryptosystem, ElGamal, on the multiplicative group \(G = Z_p^* = Z_p - \{0\}\). This is based on the construction of a positive integer \(p = (m)p_1p_2 + 1\) form with high primality and \(m\) an even number, i.e, the possibility that \(p\) could be a composite number is at most \(1/2^{100}\). In turn \(p_1, p_2\) have high primality with the same possibility of error.
This research shows that the probability that \( m \) could be less or equal to 2000 is 88.5\%, approximately, of the 700000 runs that were carried out. Developing an integer \( p = (m)p_1p_2 + 1 \) form has a double advantage. The first one is the \((m)p_1p_2\) product factorization that is simple for the designer, since it is highly likely that \( m \) is less or equal to 2000 and \( p_1, p_2 \) are known. The second is that it provides a double lock to a Pohlig-Hellman type attack. On the other hand, when the attacker only knows the \((m)p_1p_2\) product it is not possible to factor \( p_1p_2 \), over all, if it is \( 10^{400} \) approximately. Furthermore, to solve the discrete logarithm problem when it has a \( O(10^{100}) \) size it is not possible, at least at this moment, remembering that \( p_1, p_2 \approx 10^{200} \).

Also, it was found that if we built four \((m)p_1p_2 + 1\) integers, the probability that one of them has a \( m \) less or equal to 2000, is 0.9999 approximately. Thus, if 4 computation programs are run at the same time to obtain \( p = (m)p_1p_2 + 1 \) and the process stops when a \( p \) number with high primality is found, surely the \( p \) number has a \( m \) less or equal to 2000. If we proceed in this way the calculating time has a high probability of being reduced.

With respect to the Index Calculus attack Method for the discrete logarithm, that is one of the most widely used at this moment, the construction of the \( p \) integer with high primality resists this attack, since \( p > 10^{300} \) is considered an intractable problem, at least until now.

Future work will be developing a Diffie-Hellman scheme [13] using the ElGamal cryptosystem. To reduce decryption time \((\alpha^{ka})^{-1} \equiv (\alpha^{-1})^{ka} \mod p\) property will be used keeping the characteristic that \( k \) is random and unknown to an attacker. To reach this objective the Hash Sha function will be used [4].

**ACKNOWLEDGEMENTS.** The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, CIDETEC and ESFM) and the CONACyT for their economical support to develop this work.

**References**


Received: March 21, 2013