JP Jour. Algebra, Number Theory & Appl. **7**(2) (2007), 285-296

ELGAMAL ENCRYPTION IN PAIGE LOOPS

JUHA PARTALA and TAPIO SEPPÄNEN

Department of Electrical and Information Engineering
University of Oulu
P. O. Box 4500, Fin-90014, Finland
e-mail: juha.partala@ee.oulu.fi

Abstract

ElGamal encryption is one of the best known public key encryption methods in use. Usually the encryption is carried out in an associative algebraic structure, such as a group. However, the ElGamal method can be formulated also in a more general structure without the associativity property. In this paper we study the discrete logarithm problem, exponentiation and ElGamal encryption in a Paige loop. We discuss the selection of the subgroup for the discrete logarithm problem and it is shown that the exponentiation can be completely carried out in the corresponding finite field. We also discuss the benefits, drawbacks and feasibility of this method.

1. Introduction

Traditionally public key encryption is carried out in a finite group or field. These have proved out to be useful algebraic structures, because they are well known. But are they optimal? A group, for instance, contains an axiom for the associativity of the operation. In many cases this is a useful property, but in some cases it is not needed. For example, the ElGamal encryption scheme can be formulated in a non-associative structure provided that the exponentiation can be meaningfully carried out. In a cryptographical point of view it is useful to study algebraic

---

2000 Mathematics Subject Classification: 20N05, 94A60.

Keywords and phrases: non-associative, cryptrography, discrete logarithm problem.

Received December 1, 2006

structures with minimum amount of properties and structure, and this way cut down tools used in cryptanalysis.

In this paper we study ElGamal encryption [1] in one of the most well-known generalization of groups - the Moufang loops. We have restricted our investigation to the non-associative, finite and simple case. These kind of Moufang loops are widely known as Paige loops, crediting L. Paige who studied them in 1956 [7]. In the following sections we briefly describe ElGamal encryption and the discrete logarithm problem. We also define Moufang and Paige loops and describe a method to generate them. The method is due to M. Zorn. In Section 2 we define the discrete logarithm problem in Paige loops and give some results regarding its security. We extend the results of Maze in [3]. We also study the selection of the generating element and asses the complexity of exponentiation. In Section 3 we study the ElGamal encryption in Paige loops and discuss the feasibility of the encryption method.

1.1. ElGamal encryption

The basis of ElGamal encryption is the discrete logarithm problem, which can be described as follows. Let G be a cyclic group of order n and let g be a generator of G. Suppose now that $a \in G$. The *discrete logarithm* of a to the base g is an integer x such that $g^x = a$ and $0 \leq x < n$. As usual, it is denoted as $x = \log_g a$. The problem is to find $\log_g a$ given G, g and a. It is known that in certain groups this is a very hard problem [4]. The discrete logarithm problem is generally abbreviated as DLP, and in the following we adopt the same convention.

A closely related problem to the DLP is the so-called Diffie-Hellman problem. The problem is to find g^{ab} given g^a and g^b. As can be seen, this is easily solved by computing the discrete logarithm of one of the elements g^a or g^b. This means that the DLP is at least as hard as the Diffie-Hellman problem. It is not known whether the Diffie-Hellman problem is computationally equivalent to the discrete logarithm problem.

Generalized ElGamal encryption utilizes the Diffie-Hellman problem in the following way. Let $G = \langle g \rangle$ be a given finite cyclic group. Suppose that Alice wants to send a secret message to her friend Bob. We assume that the message has been coded in some way as an element $m \in G$. Alice and Bob pick random integers a and b, respectively. These are their secret keys. Their public keys are g^a and g^b, respectively. To encrypt the message m, Alice computes $(g^b)^a = g^{ab}$ and multiplies it with m to get the encrypted message mg^{ab}. Bob can decrypt the message by multiplying the encrypted message with the inverse of the element $(g^a)^b = g^{ab}$. As can be seen, if the adversary can solve the Diffie-Hellman problem, he can solve the secret message m by computing the element g^{ab} from g^a and g^b.

It should be pointed out, that the ElGamal scheme given here can only be used as a primitive of the cryptosystem. It is not secure against several imaginable attacks in itself. A practical cryptosystem utilizing the ElGamal primitive can be found, for example, in [9].

1.2. Paige loops

Quasigroups are defined as follows.

Definition 1. Let Q be a non-empty set and $\cdot$ be a binary operation on Q. Then $(Q, \cdot)$ is a quasigroup if and only if for every ordered pair $(a, b) \in Q^2$ equations

$$x \cdot a = b, \quad a \cdot y = b \tag{1}$$

have unique solutions for every $x, y \in Q$.

These solutions are often expressed as $x = b/a$ and $y = a\backslash b$. It is important to notice that quasigroups do not necessarily have a neutral element. If such an element exists, then the quasigroup is called a *loop*. In some sense it can be said that *loops* are groups without the associative property. This is because most of the concepts defined for groups can also be defined for loops. For example, the notion of a subloop can be directly adopted from group theory.

Probably the best known type of loops is the Moufang loop. These loops were studied by Ruth Moufang in 1935 [5]. A loop M is a Moufang loop, if the operation satisfies the Moufang identities

$$xy \cdot zx = x(yz \cdot x), \tag{2}$$

$$x(y \cdot xz) = (xy \cdot x)z, \tag{3}$$

$$x(y \cdot zy) = (x \cdot yz)y \tag{4}$$

for every $x, y, z \in M$. The following evaluation rules are employed: juxtaposition xy is evaluated first, followed by $\cdot$, and finally by parentheses. It can be shown that any of these equations implies the other two [8]. A Moufang loop is also known to be *power associative*. That is, every element generates a group.

A normal subloop P of Q is defined to satisfy the following conditions for every $x, y \in Q$:

$$xP = Px, \quad (xP)y = x(Py), \quad x(yP) = (xy)P. \tag{5}$$

If the loop is a group, then these properties can be seen to reduce to the well-known definition of a normal subgroup. If a loop has only the trivial normal subloops, then it is *simple*. If a Moufang loop M is non-associative, finite and simple, then it is called a *Paige loop*.

Paige loops can be constructed using Zorn's algebra in the following way. Let $\mathrm{F}_q = GF(q)$ be the Galois field of q elements, and $\alpha, \beta \in \mathrm{F}_q^3$. Zorn's algebra $Z(q)$ consists of every 2×2 matrix

$$\begin{pmatrix} a & \alpha \\ \beta & b \end{pmatrix}, \text{ where } a, b \in \mathrm{F}_q \text{ and } \alpha, \beta \in \mathrm{F}_q^3. \tag{6}$$

We define multiplication as

$$\begin{pmatrix} a & \alpha \\ \beta & b \end{pmatrix} \cdot \begin{pmatrix} c & \gamma \\ \delta & d \end{pmatrix} = \begin{pmatrix} ac + \alpha \cdot \delta & a\gamma + d\alpha - \beta \times \delta \\ c\beta + b\delta + \alpha \times \gamma & \beta \cdot \gamma + bd \end{pmatrix}, \tag{7}$$

where $\cdot$ is the normal inner product and $\times$ is the cross product of vectors. If the determinant

$$\det \begin{pmatrix} a & \alpha \\ \beta & b \end{pmatrix} = ab - a \cdot \beta \tag{8}$$

of an element is non-zero, then it has an inverse element

$$\begin{pmatrix} a & \alpha \\ \beta & b \end{pmatrix}^{-1} = \frac{1}{ab - \alpha \cdot \beta}\begin{pmatrix} b & -\alpha \\ -\beta & a \end{pmatrix}. \tag{9}$$

It can be shown [7] that the set of elements with a determinant of 1 forms a Moufang loop in respect to the multiplication defined in (7). This loop is denoted as $M(q)$ and its neutral element is

$$e = \begin{pmatrix} 1 & (0, 0, 0) \\ (0, 0, 0) & 1 \end{pmatrix}. \tag{10}$$

Clearly the set $E = \{e, -e\}$ is a normal subloop of $M(q)$ and it induces a congruence relation ~ on $M(q)$. Paige loop $M^*(q)$ is the quotient loop $M(q)/\sim$. A concise study of Moufang and Paige loops can be found, for example, in [10].

2. DLP in Paige Loops

The discrete logarithm problem in Paige loops was studied by Gérard Maze in [3]. In the case of Paige loop $M^*(q)$ we can work in the corresponding Moufang loop $M(q)$, if we keep in mind that every operation is considered modulo ~.

In [3] Maze shows that the problem completely reduces to the DLP in the projective special linear group

$$\mathrm{PSL}_2(\mathrm{F}_q) = \mathrm{SL}_2(\mathrm{F}_q)/Z(\mathrm{SL}_2(\mathrm{F}_q)) = \mathrm{SL}_2(\mathrm{F}_q)/\{\pm I\}. \tag{11}$$

In this case, we can work in $\mathrm{SL}_2(\mathrm{F}_q)$ if we keep in mind that the operations are considered modulo $Z(\mathrm{SL}_2(\mathrm{F}_q))$. Proof of the reduction is based on an injective group homomorphism $\omega : \langle x \rangle \to \mathrm{SL}_2(q)/\{\pm I\}$, where $x \in M^*(q)$.

If

$$x = \begin{pmatrix} a & \alpha \\ \beta & b \end{pmatrix}, \tag{12}$$

then every element $y \in \langle x \rangle$ can be written in form

$$y = \begin{pmatrix} c & s\alpha \\ s\beta & d \end{pmatrix} \tag{13}$$

for some $c, d, s \in \mathrm{F}_q$. The injective group homomorphism can then be expressed as

$$\omega(y) = \omega\left(\begin{pmatrix} c & s\alpha \\ s\beta & d \end{pmatrix}\right) = \begin{pmatrix} c & s\alpha \cdot \beta \\ s & d \end{pmatrix} \tag{14}$$

for every $y \in \langle x \rangle$.

In [3] Maze also gives an algorithm for a polynomial-time reduction from the DLP in $M^*(q)$ to the DLP in F_q. This reduction means that the traditional DLP in a finite field can be considered at least as hard as the DLP in a Paige loop. For the sake of completeness we prove that the DLP in $M^*(q)$ is at least as hard as the DLP in F_q.

Proposition 1. *The discrete logarithm problem in* F_q *reduces to the discrete logarithm problem in* $M^*(q)$ *in polynomial time.*

Proof. Assume that there exists a polynomial time algorithm to solve the DLP in $M^*(q)$. Let

$$X = \begin{pmatrix} g & 0 \\ 0 & g^{-1} \end{pmatrix} \in \mathrm{SL}_2(\mathrm{F}_q). \tag{15}$$

Now

$$X^n = \begin{pmatrix} g & 0 \\ 0 & g^{-1} \end{pmatrix}^n = \begin{pmatrix} g^n & 0 \\ 0 & g^{-n} \end{pmatrix}, \tag{16}$$

and we can solve n from $\omega^{-1}(X^n) = (\omega^{-1}(X))^n$ using the assumed algorithm. Since we now know n from $g^n \in \mathrm{F}_q$, and this is a discrete logarithm problem in F_q, the algorithm can also solve the DLP in F_q.

The proposition above completes the proof that these two problems can be considered computationally equivalent. This is a crucial point, because the DLP in F_q is well known and can thus be considered a safe primitive. On the other hand, exponentiating in a Paige loop does not yield any additional security to the encryption process. As the binary operation of a Paige loop is obviously slower than that of the corresponding finite field, this means that we should completely carry out the exponentiation in F_q.

As can be seen from Proposition 1, it is possible to choose

$$x = \omega^{-1}(X) = \omega^{-1}\left(\begin{pmatrix} g & 0 \\ 0 & g^{-1} \end{pmatrix}\right) = \begin{pmatrix} g & (0, 0, 0) \\ (0, 0, 0) & g^{-1} \end{pmatrix} \tag{17}$$

as the base of the discrete logarithm. Clearly $\langle X \rangle$ is isomorphic to $\langle g \rangle$ and we can operate completely in F_q. The other possibility is to choose an element $X_2 \in SL_2(F_q)$,

$$X_2 = \begin{pmatrix} h & i \\ j & k \end{pmatrix}, \tag{18}$$

with $i, j \neq 0$ and $\det X_2 = 1$. The element is then exponentiated in $SL_2(F_q)$ and the result is mapped to $M^*(q)$ using ω^{-1}. One has to be careful in the selection of the generating element, because the DLP can be feasibly solved in certain subgroups of $SL_2(F_q)$. The following example can be found in [6]. Suppose that we are working in $SL_2(Z_p)$, and we choose an element G of order divisible by p. Then the element is a conjugate of $I + c\delta_{12}$ or $I + c\delta_{21}$, where $c \in Z_p$ and δ_{ij} is a matrix whose entries are all zero except the (i, j)-th entry, which is 1. Let us assume that $g = A^{-1}(I + \delta_{12})A$ for some $A \in SL_2(Z_p)$. Now

$$G^m = (A^{-1}(I + \delta_{12})A)^m = A^{-1}(I + m\delta_{12})A \tag{19}$$

and m can be easily deduced.

If the selection of the generating element is made according to (18), exponentiation is significantly slower compared to the case of (17), because matrix multiplication has to be used. This can be countered, if we make a right selection for X_2. In fact, we can show that, by making a good selection, the complexity of exponentiating the element in (18) is virtually as good as exponentiating the element in (17). Let

$$X = \begin{pmatrix} g & 0 \\ 0 & g^{-1} \end{pmatrix}. \tag{20}$$

Now choose P with $\det P = 1$, and let

$$X_2 = P^{-1}XP. \tag{21}$$

Exponentiating X_2 gives

$$X_2^n = (P^{-1}XP)^n = P^{-1}X^nP. \tag{22}$$

Proposition 2. *$\langle X_2 \rangle$ is isomorphic to $\langle g \rangle$.*

Proof. Let $f : \langle X \rangle \to \langle P^{-1}XP \rangle$ and

$$f(A) = P^{-1}AP. \tag{23}$$

Now f is a group isomorphism and $\langle X_2 \rangle \cong \langle X \rangle \cong \langle g \rangle$.

It should be noted that since X_2 is a conjugation of X with P in $\mathrm{SL}_2(\mathrm{F}_q)$, the procedure above works, if X_2 is in the same conjugacy class with X. Proposition 2 and eq. (22) allow us to exponentiate in F_q, if X_2 is chosen accordingly. If $P \neq I$, then the mapping ω^{-1} becomes a bit more complicated. Let

$$X_2^n = \begin{pmatrix} h & i \\ j & k \end{pmatrix}. \tag{24}$$

Using (14), we find that

$$\omega^{-1}(X_2^n) = \omega^{-1}\begin{pmatrix} h & i \\ j & k \end{pmatrix} = \begin{pmatrix} h & j\alpha \\ j\beta & k \end{pmatrix}, \tag{25}$$

where $j\alpha \cdot \beta = i$, that is, $\alpha \cdot \beta = j^{-1}i$. We only need to find suitable vectors α, β. One possible choice is $\alpha = (j^{-1}, 0, 0)$ and $\beta = (i, 0, 0)$. Of course these vectors have to be fixed to fix the mapping ω. Regardless of the choice of α and β, the subgroup of $M^*(q)$ generated by x satisfies the following proposition.

Proposition 3. *$\langle x \rangle$ is isomorphic to $\langle X_2 \rangle / \{\pm I\}$.*

Proof. Now

$$\omega(\langle x \rangle) = \langle X_2 \rangle / \{\pm I\}. \tag{26}$$

Since ω is an injective group homomorphism from $\langle x \rangle$ to $\mathrm{SL}_2(q)/\{\pm I\}$ and thus an isomorphism from $\langle x \rangle$ to $\langle X_2 \rangle / \{\pm I\}$,

$$\langle x \rangle \cong \langle X_2 \rangle / \{\pm I\}. \tag{27}$$

3. Encryption and Decryption

As was seen in the previous section, the exponentiation can be completely carried out in the finite field F_q. For the encryption process we need a generator $g \in \mathrm{F}_q$ and $P \in \mathrm{SL}_2(\mathrm{F}_q)$. As in Subsection 1.1 Alice and Bob then choose their secret keys a and b and publish their public keys g^a and g^b.

If Alice wants to send a message to Bob, then she computes

$$x = \omega^{-1}(X^{ab}) = \omega^{-1}\left(P^{-1}\begin{pmatrix} g^{ab} & 0 \\ 0 & g^{-ab} \end{pmatrix} P\right) \tag{28}$$

using the method in Section 2. The vectors α and β have to be fixed. It is completely possible to make the selection at this point. It only has to be known in which way the selection is made. In particular, it is better to fix ω at this point, because the adversary cannot deduce it without solving the Diffie-Hellman problem to get g^{ab}.

The message has to be coded in some way as an element of $M^*(q)$. A very simple way to do this could be

$$m = \begin{pmatrix} m_0 & (0, 0, 0) \\ (0, 0, 0) & m_0^{-1} \end{pmatrix}, \tag{29}$$

where $m_0 \in \mathrm{F}_q$. The selection is up to the protocol, and more research is needed to find a secure method. Once the message has been coded as an element of $M^*(q)$, the encryption process is then straightforward: the elements m and $x = \omega^{-1}(X^{ab})$ are multiplied using the Zorn multiplication formula in (7). The secret message is therefore

$$c = mx = m\omega^{-1}(X^{ab}), \tag{30}$$

which, if m is coded in a way described in (29) and

$$x = \begin{pmatrix} h & j\alpha \\ j\beta & k \end{pmatrix}, \tag{31}$$

comes to

$$c = \begin{pmatrix} m_0 h & m_0 j\alpha \\ m_0^{-1} j\beta & m_0^{-1} k \end{pmatrix}. \tag{32}$$

For the decryption process Bob has to invert the element $x = \omega^{-1}(X^{ab})$. Since $\det x = 1$ in Moufang loops, (9) reduces to

$$\begin{pmatrix} a & \alpha \\ \beta & b \end{pmatrix}^{-1} = \begin{pmatrix} b & -\alpha \\ -\beta & a \end{pmatrix}. \tag{33}$$

Using this fact we get

$$x^{-1} = \begin{pmatrix} k & -j\alpha \\ -j\beta & h \end{pmatrix}. \tag{34}$$

Decryption is therefore

$$cx^{-1} = \begin{pmatrix} m_0 h & m_0 j\alpha \\ m_0^{-1} j\beta & m_0^{-1} k \end{pmatrix}\begin{pmatrix} k & -j\alpha \\ -j\beta & h \end{pmatrix} = \begin{pmatrix} m_0 & (0, 0, 0) \\ (0, 0, 0) & m_0^{-1} \end{pmatrix}. \tag{35}$$

4. Discussion

As is well known, in the tradition ElGamal encryption system, the secret keys a and b have to be changed for every new message m. Let m_1, m_2 be two consecutive messages and c_1, c_2 be the respective secret messages. The adversary now knows $c_1 = m_1 g^{ab}$ and $c_2 = m_2 g^{ab}$, and he can compute – using associativity – that

$$c_1 c_2^{-1} = (m_1 g^{ab}) \cdot (g^{-ab} m_2^{-1}) = m_1 g^{ab} g^{-ab} m_2^{-1} = m_1 m_2^{-1}. \tag{36}$$

It is not a desirable property that secret messages are in relation to the original messages in such a way. This property renders the method vulnerable, for example, to differential cryptanalysis, unless some kind of randomization is used. If non-associative structures are used instead of groups, this is no longer the case as pointed out by Keedwell in [2]. Unfortunately this is not the case in Paige loops due to the weak associative property in (2).

It is up to the protocol whether the loss of the associative law benefits the encryption process. It was seen in Section 2, that the actual computation intensive part of the encryption, the exponentiation, is virtually as fast in a Paige loop as in the corresponding Galois field F_q. This is an interesting property as the complexity of the encryption process is almost the same compared to finite field encryption. In addition, in the former case, the associative law is missing. Unfortunately we still have the weak associative laws that force us to change the encryption key for every new message. More research is needed whether it is possible to find a loop in which the exponentiation is fast, the DLP is secure and there is not even any weak forms associativity.

For the ElGamal encryption method to work, we obviously still need some additional structure in the loop. To have a unique interpretation to the expression x^n and thus a meaningful discrete logarithm, the loop has to be power associative. For the decryption we also need to be able to cancel out g^{ab}.

Another point is the discrete logarithm problem. Since there exist many fast algorithms for computing discrete logarithms in finite fields [11], the key size has to be very large. It would be better to find a loop whose DLP does not reduce to the DLP of F_q in polynomial time. If it is possible to show that the DLP in F_q reduces to the DLP in that particular loop, then we could consider the DLP of that loop safe. For this the loop would have to have a big cyclic subgroup but as little structure as possible. This is not the case with the nearly associative Moufang and Paige loops.

References

[1] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inform. Theory 31(4) (1985), 469-472.

[2] A. D. Keedwell, Construction, properties and applications of finite neofields, Comment. Math. Univ. Carolin. 41(2) (2000), 283-297.

[3] G. Maze, Algebraic methods for constructing one-way Trapdoor functions, Ph.D. Thesis, University of Notre Dame, 2003.

[4] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, Handbook of applied cryptography, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton, FL, 1997.

[5] R. Moufang, Zur Struktur von Alternativkörpern, Math. Ann. 110(1) (1935), 416-430.

[6] S. Paeng, K. Ha, J. Kim, S. Chee and C. Park, New public key cryptosystem using finite non abelian groups, Lecture Notes in Comput. Sci. 2139 (2001), 470-485.

[7] L. J. Paige, A class of simple Moufang loops, Proc. Amer. Math. Soc. 7 (1956), 471-482.

[8] H. O. Pflugfelder, Quasigroups and loops: introduction, Heldermann Verlag, Berlin, 1990.

[9] C. P. Schnorr and M. Jakobsson, Security of signed ElGamal encryption, Lecture Notes in Comput. Sci. 1976 (2000), 73-89.

[10] P. Vojtěchovský, Finite simple Moufang loops, Ph.D. Thesis, Iowa State University, 2001.

[11] S. S. Wagstaff, Jr., Cryptanalysis of number theoretic ciphers, Computational Mathematics Series, Chapman & Hall/CRC, Boca Raton, FL, 2003.

■