

A Non-Commutative Generalization of ElGamal Key Exchange using Polycyclic Groups

Delaram Kahrobaei[†]

Bilal Khan^{*}

Abstract—In this paper, we propose a non-commutative key-exchange scheme which generalizes the classical ElGamal Cipher to polycyclic groups. We describe the criteria for groups which would provide good candidates for such cryptosystems, we also examine the complexity of the decision problems related to these key exchange.

I. INTRODUCTION

The ElGamal algorithm [7] is an asymmetric encryption algorithm for public key cryptography, based on Diffie-Hellman [5] key agreement. ElGamal is semantically secure [10] under reasonable assumptions, and is probabilistic [11] in the sense that a single plaintext can encrypt to many possible ciphertexts. The ElGamal algorithm is widely used in the free GNU Privacy Guard software, recent versions of PGP, and several other cryptosystems¹.

A. Classical ElGamal

The original ElGamal encryption scheme operates as follows. Suppose Alice and Bob wish to communicate over a network in a manner secure from malicious eavesdroppers. First, Bob fixes a large prime p (say $p > 10^{150}$), a primitive root $b \bmod p$ (meaning that any $y \in \mathbb{Z}_p$ may be expressed as $y = b^l \bmod p$ for some l), and an integer c in the range $1 < c < p$. The primitivity of b implies that there is some ℓ for which $b^\ell = c \bmod p$. Bob's private key is taken to be ℓ , while his public key is the tuple (b, c, p) .

Alice performs *encryption* by segmenting the plaintext x and encoding it as a sequence of integers in the range $0 < x < p$. Since each integer is treated independently, and so we assume (without loss of generality) that x consists of precisely one integer in the interval $(0, p)$. Alice chooses a

temporary secret in the form of an auxiliary random integer r and encrypts a plaintext as

$$X = (x \cdot c^r) \bmod p.$$

Along with this encrypted message X , Alice includes the header b^r . Note that for encryption Alice needs to know only (b, c, p) i.e. Bob's public key. Alice chooses the random temporary secret r , but does not require the discrete logarithm ℓ , which remains Bob's secret.

Bob performs *decryption* by first manipulating the header

$$(b^r)^\ell = b^{r \cdot \ell} = (b^\ell)^r = c^r \bmod p.$$

Since p is prime, c^r has a computable multiplicative inverse in \mathbb{Z}_p . It follows that the original message can be recovered by noting that

$$X \cdot (c^r)^{-1} = x \cdot c^r \cdot (c^r)^{-1} = x \bmod p.$$

Note that decryption requires knowledge of the discrete logarithm ℓ but not the random temporary secret r .

The ElGamal cipher leverages the purported difficulty of computing the *discrete logarithm*, that is given b , c and p in

$$b^x \equiv c \bmod p,$$

it is computationally infeasible to determine x . There is little connection between discrete logarithms and logarithms in \mathbb{Z} . The discrete logarithm can be attacked in one of two ways, one can take the naïve approach using trial and error, but if p is very large this method is highly inefficient. To avoid specialized logarithm computation attacks that are effective in certain cases, Bob must choose p such that $p-1$ does not have "too many" small prime factors [8]. To date no efficient way of computing discrete logarithms has been found. The best fully proved algorithm for solving this problem is the Index-calculus algorithm [9], which has time complexity $O(e^{\sqrt{n \log n}})$ where n is the bit-size of the modulus p . If the discrete logarithm problem could be solved efficiently, then ElGamal would be broken.

[†] University of St Andrews, St Andrews, Fife, KY 16 9SS, Scotland, UK.

^{*} ITT Industries, Advanced Engineering & Sciences, at the Center for Computational Sciences of the U.S. Naval Research Laboratory, Washington DC. John Jay College of Criminal Justice, City University of New York, NY.

¹The Digital Signature Algorithm is a variant of the ElGamal *signature* scheme, and should not be confused with the ElGamal algorithm.

B. Key Exchange with ElGamal

Unlike the original Diffie-Hellman protocol, ElGamal is intended to encrypt messages, not merely to communicate session keys. However, because of the computational expense of asymmetric encryption, hybrid encryption strategies are often used. ElGamal can be used for key exchange, simply by encrypting a short key to be used in a symmetric-key cipher. The (much longer) intended messages are then encrypted more efficiently using the symmetric-key cipher.

C. Prior Commutative Generalizations of ElGamal

The ElGamal cipher is typically described in the setting of the multiplicative group \mathbb{Z}_p , for prime integer p . However, it can be readily generalized to work in any finite cyclic group G . The following is a list of cyclic groups, of which the first three have received the greatest attention in ElGamal schemes:

- 1) \mathbb{Z}_p^* , the multiplicative group of integers modulo a prime.
- 2) $\mathbb{F}_{2^m}^*$, the multiplicative group of finite field \mathbb{F}_{2^m} of characteristic 2.
- 3) \mathbb{F}_q^* , the multiplicative group of the finite field \mathbb{F}_q where $q = p^m$, p a prime.
- 4) \mathbb{Z}_n^* , the group of units—where n is a composite integer.
- 5) The group of points on an elliptic curve over a finite field.
- 6) The Jacobian of a hyperelliptic curve over a finite field.
- 7) Class groups of imaginary quadratic number fields [12].

In each of the above generalizations—as in the classical setting (1)—the security of the encryption scheme rests on the (unproven) difference in the complexity of multiplication and discrete logarithm, and more precisely, the so-called Decisional Diffie-Hellman (DDH) assumption [4].

In this paper, we extend ElGamal from cyclic groups to the more general setting of non-commutative groups for which DDH is believed to hold. Although the proposed schemes can be used for general encryption, we will present them in the context of their application to key-exchange. As we shall see, security in our proposed schemes will be based on the disparity between the various group-theoretic decision problems.

D. Prior Non-Commutative Key-Exchange Schemes

In 1999, the Arithmetica or “commutator” key exchange [1] was introduced by Anshel, Anshel and Goldfeld. In contrast

with Diffie-Hellman and ElGamal, it uses non-commutative groups such as braid groups [2].

The following terminology is required to describe the Arithmetica scheme. Let G be a (not necessarily finite) group with generators g_1, g_2, \dots, g_l . Given $a, b \in G$, we define the *conjugate* of a by b to be $b^{-1}ab$ and write it as a^b ; likewise, we define the *commutator* of a and b to be $a^{-1}b^{-1}ab$, and write it as $[a, b]$. A subset $X \subset G$ is called a *subgroup* if it is closed under multiplication and inverses, and the relationship between X and G is then denoted $X < G$. The group G is said to have a *solvable word problem* if there is a uniform algorithm which, given any element from G (represented as a product of its generators g_i) determines whether the element is equal to the identity element in G .

The Arithmetica scheme is based on the following observation: Let S and T be two finitely generated subgroups of G with generators $\{s_1, \dots, s_n\}$ and $\{t_1, \dots, t_m\}$ respectively, and let $a \in S$ and $b \in T$. Note that given $s_1^a, \dots, s_m^a, t_1^b, \dots, t_m^b$ and either a or b , one can compute $[a, b]$. This is verified since, e.g. if we know $b = t_{i_1}^{e_{i_1}} \dots t_{i_k}^{e_{i_k}}$ then

$$[a, b] = (t_{i_1}^{e_{i_1}})^a \dots (t_{i_k}^{e_{i_k}})^a b.$$

Arithmetica leverages the above observation to enable key exchange as follows. Suppose that Alice and Bob want to agree on a key. The group G and two finitely generated subgroups S , and T in G are the public information. The secret information is $a \in S$ chosen by Alice, and $b \in T$ chosen by Bob. The public keys are s_1^a, \dots, s_m^a for Alice, and t_1^b, \dots, t_m^b for Bob. The shared secret is then taken to be $[a, b]$.

In this paper, we propose new paradigms for non-commutative key-exchange, based on the ElGamal and various group-theoretic decision problems.

II. NEW PARADIGMS

We will present two new group-theoretic paradigms for non-commutative key-exchange. The following exposition is common to both:

Let G be a finitely presented non-abelian group having solvable word problem. Let $S, T < G$ be finitely generated proper subgroups of G , for which the subgroup $[S, T]$ (i.e. the subgroup generated by $\{[s, t] \mid s \in S, t \in T\}$) is the trivial subgroup consisting of just the identity element of G . Now suppose two parties, Alice and Bob, wish to establish a session key over an unsecured network.

A. Non-Commutative Key Exchange using Conjugacy

Bob takes $s \in S, b \in G$ and publishes b and $c = b^s$ as his public keys, keeping s as his private key. If Alice wishes to send $x \in G$ as a session key to Bob, she first chooses a random $t \in T$ and sends

$$E = x^{(c^t)}$$

to Bob, along with the header

$$h = b^t.$$

Bob then calculates $(b^t)^s = (b^s)^t = c^t$ with the header. He can now compute

$$E' = (c^t)^{-1}$$

which allows him to decrypt the session key,

$$(x^{(c^t)})^{E'} = (x^{(c^t)})(c^t)^{-1} = x.$$

The element $x \in G$ can now be used as a session key.

The feasibility of this scheme rests on the assumption that products and inverses of elements of G can be computed efficiently. To deduce Bob's private key from public information would require solving the equation $c = b^s$ for s , given the public values b and c . This is called the *conjugacy search problem* for G . Thus the security of this scheme rests on the assumption that there is no fast algorithm for solving the conjugacy search problem for the group G .

B. Non-Commutative Key Exchange using Power Conjugacy

What if the conjugacy search problem is tractable? The next paradigm embellishes conjugacy-based key exchange to address this possibility. Bob takes $s \in S, g \in G$ and $n \in \mathbb{N}$ and publishes $v = g^n$ and $w = s^{-1}gs$ as his public keys. Bob keeps $n \in \mathbb{N}$ and $s \in S$ as his private keys. Note that v and w satisfy $w^n = s^{-1}vs$. If Alice wishes to send $x \in G$ to Bob, she first chooses a random $m \in \mathbb{N}$ and $t \in T$. To encrypt x , Alice computes

$$E = x^{-1}t^{-1}(v)^m tx = x^{-1}t^{-1}g^{mn}tx$$

and sends it to Bob along with the header

$$h = t^{-1}w^m t = t^{-1}s^{-1}g^m st.$$

Bob receives E and h , and computes

$$E' = sh^n s^{-1} = t^{-1}g^{mn}t.$$

Note that $E = x^{-1}E'x$, so if Bob can solve the conjugacy search problem, he can obtain $x \in G$, which can then serve as the common secret that can be used as a symmetric session key for secure communication.

The feasibility of this scheme rests on the assumption that products and inverses of elements of G can be computed efficiently, and that the conjugacy problem is solvable. To deduce Bob's private key from public information would require solving the equation $w^n = s^{-1}g^n s$ for n and s , given the public values g^n and w . This is called the *power conjugacy search problem* for G . Thus the security of this scheme rests on the assumption that there is no fast algorithm for solving the power conjugacy search problem for the group G .

III. ANALYSIS OF PARADIGM REQUIREMENTS

In this section we describe the requirements for a group G to be a good candidate for the previously described paradigms.

A. Normal Forms

Prior to the transmission of the encrypted message it is necessary to disguise the form of this message. The importance of this is obvious since if Bob were to send Alice the word $x^{-1}ax$, the eavesdropper could just examine the word in its present form and read out the middle section, determining a , the "hidden" message. This situation is unacceptable and a method must be devised to obliterate information about potential factorizations of a group element when it is transmitted. A finitely presented group G is said to have a *normal form* if every element $g \in G$ may be expressed uniquely in the form

$$g = \prod_{i=1}^n a_i^{\epsilon_i} : \epsilon_i \in \mathbb{Z}.$$

Where a_i is either a generator of G or a commutator of generators. The existence of a unique or *canonical* normal form in certain classes of groups provides an invaluable, efficient method for masking information about potential factorizations of a group element. The existence of canonical normal forms also imply the solvability of the word problem, since testing whether an element is trivial simply requires writing it in normal form and then checking to see if the expression is trivial, i.e. $n = 0$.

B. Exponential Growth

Let G be a finitely generated group. The growth function $\gamma : \mathbb{N} \rightarrow \mathbb{R}$ is defined by $\gamma(n) = \#\{w \in G : l(w) \leq n\}$ where $l(w)$ is the length of w as a word in the generators of G . If we use normal forms to represent group elements, then each element has a unique representation, and there is an obvious relation between the growth function of a group

and the key space that the group provides. A large growth rate would imply a large key space for the set of all possible keys, thus making the brute force search of this space intractable. Ideally, we would like to use groups which exhibit provably exponential growth.

C. Complexity Considerations

The key exchange scheme A based on *conjugacy* relies on the hypothesis that the complexity of conjugacy is exponentially higher than the complexity of the word problem. While this is not known to be true in general, it is clear that the complexity of conjugacy is not less than the complexity of the word problem. To see this assume that the conjugacy problem for G is solvable. Hence there exists an algorithm such that given $v, w \in G$ that lie in the same conjugacy class, we can obtain a $k \in G$ that satisfies $w = k^{-1}vk$. Taking the special case $k = 1$ yields $w = v \Rightarrow wv^{-1} = 1$, hence given any word,

$$g = g_1^{\epsilon_1} g_2^{\epsilon_2} \cdots g_n^{\epsilon_n} \in G$$

$$g = 1 \Leftrightarrow g_1^{\epsilon_1} \cdots g_i^{\epsilon_i} \sim (g_{i+1}^{\epsilon_{i+1}} \cdots g_n^{\epsilon_n})^{-1} : \forall 1 \leq i < n.$$

where \sim represents the equivalence relation “is conjugate to”.

The key exchange scheme B based on *power conjugacy* relies on the hypothesis that the complexity of power conjugacy is exponentially higher than the complexity of conjugacy. While this is not proven in general, it is clear that the complexity of power conjugacy is not less than the complexity of the conjugacy. To see this, assume that the power conjugacy problem is solvable. This means that we have a method that given $v, w \in G$, we can obtain a $n \in \mathbb{N}$ and $k \in G$ that satisfies $w^n = k^{-1}vk$. Therefore given $a, b \in G$ that lie in the same conjugacy class, we can find $k \in G$ such that $a = k^{-1}bk$. This follows since conjugacy is just a special case of power conjugacy problem, when $n = 1$.

IV. A SPECIFIC SCHEME BASED ON POLYCYCLIC GROUPS

A group G is said to be *polycyclic* if it has a series

$$G \triangleright G_n \triangleright G_{n-1} \triangleright \cdots \triangleright G_1 \triangleright G_0 = \{1\}$$

where for each $i = 1, \dots, n-1$:

- 1) G_i is a *normal subgroup* of G_{i+1} , i.e. $G_i < G_{i+1}$ and

$$\forall x \in G_{i+1}, y \in G_i : x^{-1}yx \in G_i$$

- 2) The quotient G_{i+1}/G_i is a cyclic group, i.e. $\exists z \in G_{i+1}$ such that $\forall y \in G_{i+1}$

$$\exists x \in G_i, n \in \mathbb{N} : y = xz^n.$$

A subnormal series with these properties is called a *polycyclic series*. Clearly, polycyclic groups are a non-commutative generalization of cyclic groups, since the latter have a polycyclic series with $n = 1$. The *Hirsch length* of a polycyclic group G is the number of infinite groups in its polycyclic series. The Hirsch length of a group is independent of the choice of polycyclic series, as a consequence of the Schreier Refinement Theorem.

A. Normal Forms

A polycyclic group can always be presented in with a finite set of generators a_1, \dots, a_n , which are related by a set of equations of the following forms: $a_j^{a_i} = w_{ij}$, $a_j^{a_i^{-1}} = v_{ij}$, $a_k^{r_k} = u_{kk}$, where $k \in \{1, \dots, n\} = I$, $r_i \in \mathbb{N}$ if $i \in I$, and the right hand sides w_{ij}, v_{ij}, u_{jj} of the relations are words in the generators a_{j+1}, \dots, a_n . Using induction, one may show that every element in the group defined by this presentation can be written in the form $a_1^{e_1} \cdots a_n^{e_n}$ with $e_i \in \mathbb{Z}$ and $0 \leq e_i < r_i$ if $i \in I$. A polycyclic presentation is called *consistent* if every element in the group defined by the presentation can be represented uniquely by a word of the form $a_1^{e_1} \cdots a_n^{e_n}$ with $e_i \in \mathbb{Z}$ and $0 \leq e_i < r_i$ if $i \in I$, and in this case these words are called normal words. It is well-known that every polycyclic group has a consistent polycyclic presentation and these presentations are frequently used as a basis for computations with polycyclic groups [14]. Every polycyclic group can also be described as a finitely generated subgroup of matrices with integer valued entries, i.e. the group $GL(d, \mathbb{Z})$ for some $d \in \mathbb{N}$.

B. Growth Rate

A large class of polycyclic groups are known to have an exponential growth rate (namely those which are not virtually nilpotent, see Wolf [16] and Milnor [13]).

C. Complexity Considerations

As was explained in the previous section polycyclic groups are linear groups, that is they can be embedded as a subgroup of $GL(n, \mathbb{Z})$. In this setting, both group multiplication and the word problem are efficiently solvable, since matrix multiplication for such groups is solvable in polynomial time. The conjugacy problem for polycyclic groups is decidable by results of Remeslenikov [15] and Blackburn [3]. To see directly, we can appeal to the fact that polycyclic group is a subgroup of $GL(n, \mathbb{Z})$. This leads to the following lemma:

Lemma 1: Let $G < GL(n, \mathbb{F}^*)$ then if $x, y \in G$ are conjugate then the Jordan normal form of x is also the Jordan normal form of y .

Proof: Let $J(a)$ be the Jordan normal form of $a \in G$, where $G < GL(n, \mathbb{F}^*)$. Let $G < GL(n, \mathbb{F}^*)$ and $x, y \in H$ such that $\exists k \in H : x = y^k$. Since $x, y \in GL(n, \mathbb{F}^*)$, then $\exists p \in GL(n, \mathbb{F}^*) : J(x) = x^p = y^{kp} = J(y)$. ■

Proposition 2: The search conjugacy problem in any subgroup of the General Linear group is solvable.

Proof: Let $G < GL(n, \mathbb{F}^*)$ Assume that $v, w \in G$ are conjugate, that is $\exists k \in G : v = k^{-1}wk$. Then by Lemma 1 $J(v) = J(w)$, then $\exists p, q \in GL(n, \mathbb{F}^*)$ such that $J(v) = v^p = w^q = J(w)$ which implies that $v = w^{qp^{-1}}$, this solves the conjugacy search problem. ■

Implementing theorem 2 into an algorithm yields a solution to the search conjugacy problem. Although the precise complexity of conjugacy search is not known, it is widely conjectured to be exponential. The status of power conjugacy search for polycyclic groups remains an *open question*—no uniform algorithm is known.

V. EXPERIMENTAL EVALUATION

Recently Eick and Kahrobaei [6], ran a series of experiments on how the complexity of the conjugacy problem varied with the Hirsch length of a polycyclic group using a collection algorithm. Their findings were that the time complexity grew exponentially relative to the Hirsch length. For example with a Hirsch length of 2, the word problem on a randomly generated word took 0.00 secs and 9.96 secs for the conjugacy problem, however for a Hirsch length of 14, the time took to solve the word problem took 0.05 secs, however the conjugacy problem took in excess of 100 hours. These results demonstrated the suitability of polycyclic groups in cryptology.

r	h(G(w))	coll	conj
3	2	0.00 sec	9.96 sec
4	2	0.00 sec	9.37 sec
7	6	0.01 sec	10.16 sec
11	14	0.05 sec	> 100 hrs

For the prime 11, the result of a single conjugacy test could not be computed within one hundred hours using the current methods. For primes larger than 11, the run-times of experiments are expected to be dramatically longer.

VI. CONCLUSION

We have proposed two new paradigms for the construction of group-theoretic one-way functions for key exchange. Our paradigms are based on the complexity differences between various group-theoretic decision problems, specifically the complexity gap between *conjugacy* and *word* problems, and the complexity gap between *power conjugacy* and *conjugacy* problems. We have argued that *polycyclic groups* fulfill the three characteristics required in order for a group to provide security within these new paradigms. Our experimental trials confirm that these schemes will provide effective one way functions for public key exchange. Our future research and development efforts include implementing practical cryptographic tools based on the polycyclic groups schemes described in this paper.

REFERENCES

- [1] Iris Anshel, Michael Anshel, and Dorian Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett. (1999), 6:287–291.
- [2] Joan S. Birman, K. H. Ko, and J. S. Lee, *A new approach to the word and conjugacy problems in the braid groups.*, <http://xxx.lanl.gov/abs/math.GT/9712211> (1998), 1–31.
- [3] N. Blackburn, *Conjugacy in nilpotent groups*, Proc. Amer. Math. Soc. **16** (1965), 143–148.
- [4] D. Boneh, *The decision diffie-hellman problem*, Proceedings of the Third Algorithmic Number Theory Symposium, Lecture Notes in Computer Science **1423** (1998), 48–63.
- [5] W. Diffie and M. E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **IT-22** (1976), 644–654.
- [6] Bettina Eick and Delaram Kahrobaei, *Polycyclic groups: A new platform for cryptology?*, math.GR/0411077 (2004), 1–7.
- [7] Taher ElGamal, *A public-key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory **IT-31** (1985), no. 4, 469–472.
- [8] Paul Garrett, *Making, breaking codes: Introduction to cryptology*, Pearson Education, 2000.
- [9] S. Goldwasser and M. Bellare, *Lecture notes on cryptography*, 2001.
- [10] S. Goldwasser and S. Micali, *Probabilistic encryption & how to play mental poker keeping secret all partial information*, Proceedings of Annual ACM Symposium on Theory of Computing (1982).
- [11] ———, *Probabilistic encryption*, Journal of Computer and System Sciences **28** (1984), 270–299.
- [12] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of applied cryptography*, CRC Press, 2001.
- [13] J. Milnor, *Growth of finitely generated solvable groups*, J. Diff. Geom. **2** (1968), 447–449.
- [14] C. C. Sims, *Computation with finitely presented groups*, Encyclopedia of Mathematics and its Applications **48** (1994).
- [15] V.N. Remeslennikov, *Conjugacy in polycyclic groups*, Algebra i logika **8** (1969), no. 6, 712–725.
- [16] J. Wolf, *Growth of finitely generated solvable groups and curvature of riemannian manifolds*, J. Diff. Geom. **2** (1968), 421–446.