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 cryptosystems[1].

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 $O(e^{\sqrt{nlogn}})$ 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.

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

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 exponentionaly relitive 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.