Completely Regular Semigroups and the Discrete Log Problem

James Renshaw

January 2018

Abstract

We consider an application to the discrete log problem using completely regular semigroups which may provide a more secure symmetric cryptosystem than the classic system based on groups. In particular we describe a scheme that would appear to offer protection to a standard trial multiplication attack.

Keywords Semigroup, completely regular, discrete logarithm, cryptography

Mathematics Subject Classification (2000) 11T71, 94A60, 20M30

1 Introduction and Preliminaries

We refer the reader to [2] for basic results and terminology in semigroups and in particular for the necessary background in completely regular semigroups. See also [3] for the some background in applications of semigroup actions to the discrete log problem.

Let $G = U_{p-1}$, the group of units of the ring $\mathbb{Z}_{p-1}$ and let $X = U_p$ the group of units of $\mathbb{Z}_p$ where p is a prime. An algebraic description of the classic discrete log cipher involves defining a free action of G on X as $G \times X \to X$ by $(n, x) \mapsto x^n$. By Fermat's little theorem, since x is a unit modulo p, then $x^{p-1} \equiv 1 \mod p$ and since n is coprime to $p-1$ then there is a positive integer m such that $mn \equiv 1 \mod p-1$. Hence $x^{mn} \equiv x \mod p$ and so $x^{mn} = x$ in X. Consequently m is a 'decrypt' key for the 'encrypt' key n. In practice, of course we can use $\mathbb{Z}_p$ instead of X as only $0 \in \mathbb{Z}_p \setminus X$.

More generally, we can let X be a finite group of order r and let $G = U_r$, the group of units of the ring $\mathbb{Z}_r$. Then the action $G \times X \to X$ given by $(n, x) \mapsto x^n$ is the basis of a cryptosystem, in which the inverse of any key $n \in G$ can easily be computed using the extended Euclidean algorithm.

James Renshaw
Mathematical Sciences
University of Southampton
Southampton, SO17 1BJ, England
E-mail: j.h.renshaw@maths.soton.ac.uk
Tel: +44(0)2380593673
ORCID: 0000-0002-5571-8007

We shall consider the problem of replacing the group X by a semigroup, on the grounds that a semigroup should in principle be more complicated and potentially offer more security over a group. We note however, that in [1] the authors show that the discrete log problem over a semigroup can be reduced, in polynomial time, to the discrete log problem over a subgroup of the semigroup. Not withstanding this, we describe a scheme involving a semigroup which, by hiding part of the information relating to the semigroup multiplication, seems to exclude the possibility of computing this polynomial reduction. In addition, the scheme seems to offer some protection against a standard trial multiplication attack.

2 Completely Regular Semigroups

In the classic discrete log cipher, we can view the cryptosystem as a group acting freely on a group by exponentiation. We now briefly consider a group acting freely on a semigroup by exponentiation. It is clear that the semigroup needs to be periodic as every element will need to have finite order.

A semigroup S is called *completely regular* if every element of S belongs to a subgroup of S. A particular example of such a semigroup is a *completely simple* semigroup, which by Rees' Theorem ([2, Theorem 3.2.3]), can be shown to be isomorphic to what is commonly referred to as a Rees Matrix Semigroup. Indeed a semigroup is completely regular if and only if it is isomorphic to a semilattice of completely simple semigoups ([2, Theorem 4.1.3]). A semigroup $S = \mathcal{M}[G; I, \Lambda; P]$ is called a *Rees Matrix Semigroup over the group* G if for sets I and Λ,

$$S = I \times G \times \Lambda$$

and $P = (p_{\lambda i})$ is a $\Lambda \times I$ matrix, referred to as the *sandwich matrix*, with entries in the group G, and where multiplication is given by

$$(i, g, \lambda)(j, h, \mu) = (i, gp_{\lambda j}h, \mu).$$

If follows that for $n \in \mathbb{N}, (i, g, \lambda)^n = (i, (gp_{\lambda i})^{n-1}g, \lambda)$. Notice that S is not in general commutative, even if G is abelian.

It is worth noting that a group G is an example of a completely simple semigroup in which $|I| = |\Lambda| = 1$ and $P = (1_G)_{1\times 1}$.

3 Completely Simple Cryptosystems

Suppose now that S is a completely simple semigroup, considered as a Rees matrix semigroup $\mathcal{M}[G; I, \Lambda; P]$ and suppose also that G is finite, of order r so that $g^r = 1$ for all $g \in G$. Define an action of U_r, the group of units in $\mathbb{Z}_r$, on S by $n \cdot x = x^n$, so that if $x = (i, g, \lambda)$ then $n \cdot x = (i, (gp_{\lambda i})^{n-1}g, \lambda)$. Notice that $|U_r| = \phi(r)$.
Suppose now that $n \in U_r$ so that n is coprime to r, and hence there exists $m \in U_r$ such that $mn \equiv 1 \mod r$. Then

$$x^{mn} = (i, (gp_{\lambda i})^{mn-1}g, \lambda) = (i, (gp_{\lambda i})^{mn}p_{\lambda i}^{-1}, \lambda) = (i, (gp_{\lambda i})p_{\lambda i}^{-1}, \lambda) = (i, g, \lambda) = x.$$

Consequently if we know n, x^n and P, then we can compute x^{mn} and so recover x. We can in fact compute x^{mn} in an efficient manner, as we can deduce the values of i and λ from x^n and so we can deduce the value of $p_{\lambda i}$. Then

$$(gp_{\lambda i})^{mn-1}g = (gp_{\lambda i})^{mn}p_{\lambda i}^{-1} = \left(\left((gp_{\lambda i})^{n-1}g\right)p_{\lambda i}\right)^m p_{\lambda i}^{-1}.$$

Suppose now we know x, x^n and G. Can we compute n and therefore solve the discrete log problem over S? If we also know P then we know $p_{\lambda i}$ and so $(gp_{\lambda i})^n$. Consequently, the discrete log problem in this case is equivalent to that in the classic discrete log problem over the group G and we are no better off using the completely simple semigroup rather than a group. Suppose however that P is kept secret and that it is hard to deduce the value of $p_{\lambda i}$ from that of i and λ. We know $(gp_{\lambda i})^{n-1}g$ and we know g and hence we can compute $(gp_{\lambda i})^{n-1}$ but we don't know $p_{\lambda i}$ and so can't obviously recover the classic discrete log problem from this. According to [1], the discrete log problem over a semigroup, can be reduced, in polynomial time, to the classic discrete log problem in a subgroup of S, namely the kernel of the element x. However this assumes that we can compute with the semigroup S and in order to do that with a Rees Matrix Semigroup, we would require knowledge of the sandwich matrix P.

In this application of Rees matrix semigroups, the sets I and Λ are being used as index sets to point at the value $p_{\lambda i} \in P$, and as such we clearly don't require both of these indices. Let us therefore assume, without loss of generality, that $|\Lambda| = 1$ so that $S = I \times G, P = (p_i)_{i \in I}$ with multiplication given by $(i, g)(j, h) = (i, gp_j h)$ and so $(i, g)^n = (gp_i)^{n-1}g$. We will also assume from now on that G is abelian.

3.1 Chosen plaintext attack

Although we keep the values of P secret, if the size of I is small then we can consider the following chosen plaintext attack based on the existence of an oracle for solving the classic discrete log problem over the group G. Suppose that $|I| = m$ and let $g_1, \ldots, g_{m+1}$ be distinct elements of G. Suppose we encrypt the values (i, g_i) as $(i, g_i^n p_i^{n-1})$. By the pigeon hole principle there exists $i \neq j$ such that $p_i = p_j$ and hence

$$(g_i^n p_i^{n-1})(g_j^n p_j^{n-1})^{-1} = (g_i g_j^{-1})^n.$$

Consequently we can reduce the semigroup discrete log problem over S to the group discrete log problem over G. However, we do not know the values of i and j and so have to compute this quantity for each pair $1 \leq i, j \leq m + 1$, and there are $\binom{m+1}{2} = O(m^2)$ of these. If m is relatively small, then running m^2 versions of the group oracle in parallel is probably feasible and consequently we need to ensure that m is sufficiently large, say comparable to the size of the group G.

This clearly imposes some issues with storing the matrix P. If P is part of the secret key then a large value of m means that, in practical terms, we must compute the entries $p_i \in P$, dynamically.

3.2 Brute Force

At first sight, having P both secret and large would seem to indicate that S will be difficult to work with. However, the discrete log problem over S

seems to be effectively immune to a standard trial multiplication attack. To see this, suppose we are given (i, g) and $(i, g^n p_i^{n-1})$. Computing n using a trial multiplication attack would consists of computing $g^m q^{m-1}$ for $1 \leq m \leq \phi(|G|)$ and $q \in G$ in order to find the relevant pair with $(m, q) = (n, p_i)$. In principle there are a maximum of $\phi(|G|)|G|$ such computations. However, notice that if $\gcd(m-1, |G|) = 1$ then there exists k such that $k(m-1) \equiv 1 \mod |G|$ and so for any $q \in G, q^{k(m-1)} = q$. Consequently

$$g^n p_i^{n-1} = g^m \left(\left(g^{n-m} p_i^{n-1} \right)^k \right)^{m-1}$$

and so there is no unique pair $(m, q) = (n, p_i)$ that can be computed by a simple trial multiplication attack alone. In fact the number of such solutions is at least $\phi(|G|) - 1$.

It seems clear therefore that some other information much be gained and used in order to execute a successful trial multiplication attack.

3.3 The Proposed Completely Simple Scheme

Technically the value of p_i is only dependant on i and not on g. This may cause a problem, as if we could encrypt the data (i, g) and (i, g^{-1}) then we would obtain the values $(i, (gp_i)^{n-1}g)$ and $(i, (g^{-1}p_i)^{n-1}g^{-1})$. If, as we are assuming, G is abelian then we can calculate $(p_i^{n-1})^2$ and hence possibly p_i^{n-1}. Consequently we can deduce the value of g^n and so again reduce the semigroup discrete log problem to the corresponding group discrete log problem. We could avert this problem if the value of i was chosen in a random fashion.

Alice wants to sent Bob a secret message. Let G be a finite (abelian) group and let $I = G$. Let $n \in U_{|G|}$, the group of units mod $|G|$, and $s \in I$ be two secret keys known only to Alice and Bob. Suppose also that $f : I \times I \to G$ is a function, perhaps based on a cryptographically secure hash. We encrypt $g \in G$ as follows: choose a random value $i \in I$ and let $p_i = f(i, s)$. Clearly f must have the property that it is difficult to compute $f(i, s)$ from the value of i alone. In addition it should be hard to calculate s given $f(i, s)$ and i. For example the function $f(i, j) = H(i \oplus j)$ where H is a suitable hash and where $i \oplus j$ is the bitwise xor of i and j might suffice. Alice computes $(i, (gp_i)^{n-1}g)$ as her encrypted value of g to send to Bob. Bob calculates $p_i = f(i, s)$ and $m \in U_{|G|}$ such that $mn \equiv 1 \mod |G|$ and then computes

$$g = \left(\left((gp_i)^{n-1} g \right) p_i \right)^m p_i^{-1}.$$

However, as we have seen an attacker can't easily compute (n, p_i) by trial multiplication attack alone and as long as p_i is hard to deduce from the value of i, and I is large then the two chosen plaintext attacks detailed above would appear to be infeasible.

In taking $I = G$ the ciphertext would be twice the length of the plaintext, but a smaller value of $|I|$, but still large enough to withstand the limitations set by the chosen plaintext attack above, could reduce this by a significant amount.

One other possible chosen plaintext attack comes to mind. Suppose we encrypt the value g twice. The first time we obtain the encrypted value $(i, (gp_i)^{n-1}g) = (i, g^n p_i^{n-1})$ and the second time the value $(j, (gp_j)^{n-1}g) = (j, g^n p_j^{n-1})$. We can

then deduce the value of $(p_i p_j^{-1})^{n-1}$, but as we know neither n nor $p_i p_j^{-1}$ then it is hard to see what advantage we have gained. In fact even if we could deduce the value of n, perhaps using a different attack or some oracle, we would still need to factorise $p_i p_j^{-1}$ to deduce that values of p_i and p_j. But in addition, this still wouldn't allow us to deduce the value of the secret key s unless the function f is cryptographically insecure.

As a possible variant of this scheme, let $1 \leq k < n$ be a value known only to Alice and Bob. Alice encrypts her value of g as the value $(i, g^n p_i^{n-k})$. Bob then calculates, as before, m such that $mn \equiv 1 \mod |G|$ and recovers g from

$$g = ((g^n p_i^{n-k}) p_i^k)^m p_i^{-1}.$$

For values of $k > 1$, this is however no longer a (free) group action on the completely simple semigroup and so it is not clear if this decrypt key is unique. It is also not clear whether any increase in security will actually be achieved by choosing $k > 1$.

The major drawback of this scheme is of course that the security would be dependant on the security of the key exchange system used to exchange the key (n, s).

References

[1] Matan Banin, Boaz Tsaban, A reduction of Semigroup DLP to Classic DLP, *Designs, Codes and Cryptography*, (2016), Volume 81, Issue 1, 75–82.

[2] J.M. Howie, *Fundamentals of Semigroup Theory*, London Mathematical Society Monographs, (OUP, 1995).

[3] J. Renshaw, *$E-$dense actions of semigroups and an application to the discrete log problem*, submitted (arXiv:1712.07426).