Public key exchange using extensions by endomorphisms and matrices over a Galois field

Delaram Kahrobaei[1], Ha T. Lam[2], and Vladimir Shpilrain[3]

[1] CUNY Graduate Center and City Tech, City University of New York
dkahrobaei@gc.cuny.edu
[2] CUNY Graduate Center
hlam@gc.cuny.edu
[3] The City College of New York and CUNY Graduate Center
shpil@groups.sci.ccny.cuny.edu

Abstract. In this paper, we describe a public key exchange protocol based on an extension of a semigroup by automorphisms (more generally, by endomorphisms). One of its special cases is the standard Diffie-Hellman protocol, which is based on a cyclic group. However, when our protocol is used with a non-commutative (semi)group, it acquires several useful features that make it compare favorably to the Diffie-Hellman protocol. Here we suggest a couple of instantiations of our general protocol, with a non-commutative semigroup of matrices over a Galois field as the platform and show that security of the relevant protocols is based on quite different assumptions compared to that of the standard Diffie-Hellman protocol. Our key exchange protocols with this platform are quite efficient, too: with private keys of size 127 bits and public keys of size 1016 bits, the run time is 0.03 s on a typical desktop computer.

1 Introduction

The simplest, and original, implementation of the classical Diffie-Hellman key exchange protocol [3] uses the multiplicative group of integers modulo p, where p is prime and g is primitive $\mod p$. A more general description of the protocol uses an arbitrary finite cyclic group.

1. Alice and Bob agree on a finite cyclic semigroup G and a generating element g in G. We will write the group G multiplicatively.
2. Alice picks a random natural number a and sends g^a to Bob.
3. Bob picks a random natural number b and sends g^b to Alice.
4. Alice computes $K_A = (g^b)^a = g^{ba}$.
5. Bob computes $K_B = (g^a)^b = g^{ab}$.

Since $ab = ba$, both Alice and Bob are now in possession of the same group element $K = K_A = K_B$ which can serve as the shared secret key.

The protocol is considered secure against eavesdroppers if G and g are chosen properly. The eavesdropper must solve the *Diffie-Hellman problem* (recover g^{ab} from g, g^a and g^b) to obtain the shared secret key. This is currently considered difficult for a "good" choice of parameters (see e.g. [6] for details).

In [4], a new key exchange protocol was suggested, which has some similarity to the Diffie-Hellman protocol, but also has some important distinctive features that give the new protocol some important advantages. To explain the main idea of that protocol, assume for a moment that someone would like to modify the standard Diffie-Hellman protocol as follows:

1. Alice and Bob agree on a finite cyclic semigroup G and a generating element g in G.
2. Alice picks a random natural number a and sends g^a to Bob.
3. Bob picks a random natural number b and sends g^b to Alice.
4. Alice computes $K_A = g^b \cdot g^a = g^{b+a}$.
5. Bob computes $K_B = g^a \cdot g^b = g^{a+b} = K_A$.

It works just fine for Alice and Bob, but the problem is: anybody can compute the shared key the same way, so this protocol is completely insecure. However, if Alice and Bob do not transmit the whole g^a (respectively, g^b) but only part of it, and the final shared key is not the whole g^{a+b} but only part of it, then the protocol may become secure (or at least as secure as the standard Diffie-Hellman protocol). A particular implementation of this general idea was given in [4], where the platform semigroup G was the semigroup of matrices over the group ring $\mathbb{Z}_7[A_5]$, where A_5 is the alternating group on 5 elements. The automorphism used for an extension was an inner automorphism, i.e. conjugation by an invertible matrix H.

In this paper, we use the semigroup of matrices over a Galois field of characteristic 2; more specifically, over the field $\mathbb{GF}(2^{127})$. This allows us to reduce key size and to speed up computation quite a bit by utilizing known methods of fast computation in a field of characteristic 2. Specifically, we used the RELIC package [2] for computations in $\mathbb{GF}(2^{127})$. See our Sections 7 and 8 for more details on parameters and computational cost. Also, the endomorphism that we use for extension in our Section 4 is not inner, but a composition of an inner automorphism with the endomorphism that raises each entry of a given matrix to the power of 4. This yields new security assumptions, see our Section 6. If one uses just an inner automorphism and matrices over a field, then an extra "tweak" of the protocol is needed to avoid a linear algebra attack, see Section 5.

Finally, we mention another, very different, proposal [1] of a cryptosystem based on the semidirect product of two monoids.

2 Extensions by automorphisms or endomorphisms

A special case of the general semidirect product construction is *extension by automorphisms* (or endomorphisms, if we do not require the result to be a group, but just a semigroup). Since in this paper, we are working with semigroups (of matrices), let us assume that G is a semigroup and ψ an endomorphism of G. Then the extension of G by ψ is the semidirect product of G with the cyclic semigroup generated by ψ, i.e. the set of all pairs $(g,\ \phi)$, where $g \in G$ and ϕ is a power of ψ, with the multiplication given by

$$(g,\ \phi) \cdot (g',\ \phi') = (\phi'(g) \cdot g',\ \phi \cdot \phi').$$

Note that when we write $\phi \cdot \phi'$, this means that ϕ is applied first.

If G is a group and ψ is an automorphism of G, then the extension of G by ψ is a group.

3 General key exchange protocol

In this section, we give a general (i.e., not platform-specific) description of our key exchange protocol.

Let G be a (semi)group. An element $g \in G$ is chosen and made public as well as an arbitrary automorphism $\phi \in Aut(G)$ (or an arbitrary endomorphism $\phi \in End(G)$). Bob chooses a private $n \in \mathbb{N}$, while Alice chooses a private $m \in \mathbb{N}$. Both Alice and Bob are going to work with elements of the form (g, ϕ^r), where $g \in G,\ r \in \mathbb{N}$. Note that two elements of this form are multiplied as follows: $(g, \phi^r) \cdot (h, \phi^s) = (\phi^s(g) \cdot h,\ \phi^{r+s})$.

1. Alice computes $(g, \phi)^m = (\phi^{m-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g,\ \phi^m)$ and sends **only the first component** of this pair to Bob. Thus, she sends to Bob **only** the element $a = \phi^{m-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g$ of the (semi)group G.
2. Bob computes $(g, \phi)^n = (\phi^{n-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g,\ \phi^n)$ and sends **only the first component** of this pair to Alice. Thus, he sends to Alice **only** the element $b = \phi^{n-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g$ of the (semi)group G.
3. Alice computes $(b, x) \cdot (a,\ \phi^m) = (\phi^m(b) \cdot a,\ x \cdot \phi^m)$. Her key is now $K_A = \phi^m(b) \cdot a$. Note that she does not actually "compute" $x \cdot \phi^m$ because she does not know the automorphism $x = \phi^n$; recall that it was not transmitted to her. But she does not need it to compute K_A.
4. Bob computes $(a, y) \cdot (b,\ \phi^n) = (\phi^n(a) \cdot b,\ y \cdot \phi^n)$. His key is now $K_B = \phi^n(a) \cdot b$. Again, Bob does not actually "compute" $y \cdot \phi^n$ because he does not know the automorphism $y = \phi^m$.
5. Since $(b, x) \cdot (a,\ \phi^m) = (a,\ y) \cdot (b,\ \phi^n) = (g,\ \phi)^{m+n}$, we should have $K_A = K_B = K$, the shared secret key.

Remark 1. Note that, in contrast with the "standard" Diffie-Hellman key exchange, correctness here is based on the equality $h^m \cdot h^n = h^n \cdot h^m = h^{m+n}$ rather than on the equality $(h^m)^n = (h^n)^m = h^{mn}$. In the "standard" Diffie-Hellman set up, our trick would not work because, if the shared key K was just the product of two openly transmitted elements, then anybody, including the eavesdropper, could compute K.

We note, on the other hand, that the standard Diffie-Hellman protocol is, in fact, one of the simplest instantiations of our protocol, if the multiplicative group $\mathbb{Z}_p^*$ is used as the platform group G, and the (public) endomorphism ϕ of $\mathbb{Z}_p^*$ is given by $\phi(h) = h^k$ for every $h \in \mathbb{Z}_p^*$ and a fixed integer k. See [4] for more details.

In the next sections, we describe more interesting instantiations, where the (semi)group G is non-commutative.

4 Matrices over a Galois field and extensions by special endomorphisms

To begin with, we note that our general protocol in Section 3 can be used with *any* non-commutative group G if ϕ is selected to be a non-trivial inner automorphism, i.e., conjugation by an element which is not in the center of G. Furthermore, it can be used with any non-commutative *semigroup* G as well, as long as G has some invertible elements; these can be used to produce inner automorphisms. A typical example of such a semigroup would be a semigroup of matrices over some ring.

Now let G be the semigroup of 2×2 matrices over the Galois field $\mathbb{GF}(2^{127})$. Here we use an extension of the semigroup G by an endomorphism φ, which is a composition of a conjugation by a matrix $H \in GL_2(\mathbb{GF}(2^{127})$ with the endomorphism ψ that raises each entry of a given matrix to the power of 4. The composition is such that ψ is applied first, followed by conjugation.

Thus, for any matrix $M \in G$ and for any integer $k \geq 1$, we have

$$\varphi(M) = H^{-1}\psi(M)H.$$

$$\varphi^k(M) = H^{-1}\psi(H^{-1}) \cdots \psi^{k-1}(H^{-1})\psi^k(M)\psi^{k-1}(H) \cdots \psi(H)H.$$

Now our general protocol from Section 3 is specialized in this case as follows.

1. Alice and Bob agree on public matrices $M \in G$ and $H \in GL_2(\mathbb{GF}(2^{127})$. Alice selects a private positive integer m, and Bob selects a private positive integer n.

2. Alice computes $(M, \varphi)^m$ and sends **only the first component** of this pair to Bob. Thus, she sends to Bob **only** the matrix

$$A = H^{-1}\psi(H^{-1}) \cdots \psi^{m-1}(H^{-1})\psi^m(M)\psi^{m-1}(H) \cdots \psi(H)H.$$

3. Bob computes $(M, \varphi_H)^n$ and sends **only the first component** of this pair to Alice. Thus, he sends to Alice **only** the matrix

$$B = H^{-1}\psi(H^{-1}) \cdots \psi^{n-1}(H^{-1})\psi^n(M)\psi^{n-1}(H) \cdots \psi(H)H.$$

4. Alice computes $(B, x) \cdot (A,\ \varphi^m) = (\varphi^m(B) \cdot A,\ x \cdot \varphi^m)$. Her key is now $K_{Alice} = \varphi^m(B) \cdot A$, which is the first component of $(M, \varphi)^{m+n}$. Note that she does not actually "compute" $x \cdot \varphi^m$ because she does not know the automorphism $x = \varphi^n$; recall that it was not transmitted to her. But she does not need it to compute K_{Alice}.

5. Bob computes $(A, y) \cdot (B,\ \varphi^n) = (\varphi^n(A) \cdot B,\ y \cdot \varphi^n)$. His key is now $K_{Bob} = \varphi^n(A) \cdot B$. Again, Bob does not actually "compute" $y \cdot \varphi^n$ because he does not know the automorphism $y = \varphi^m$.

6. Since $(B, x) \cdot (A,\ \varphi^m) = (A,\ y) \cdot (B,\ \varphi^n) = (M,\ \varphi)^{m+n}$, we have $K_{Alice} = K_{Bob} = K$, the shared secret key.

5 What if the automorphism is just conjugation?

In this section, we consider the situation where the (public) automorphism ϕ is just conjugation by a (public) matrix H. In this case, transmitted matrices simplify to $H^{-m}(HM)^m$ (from Alice to Bob) and $H^{-n}(HM)^n$ (from Bob to Alice). Thus, the situation becomes similar to that in Stickel's protocol [11], and that protocol is vulnerable to a linear algebra attack if matrices involved in the protocol are over a field. The attack is as follows (see [12], [8] for more details). The attacker, Eve, is looking for matrices X and Y such that $XH = HX$, $Y(HM) = (HM)Y$, and $XY = H^{-m}(HM)^m$. Note that the first two matrix equations translate into a system of linear equations in the entries of X and Y over the ground field, whereas the last one does not. However, if X is invertible, then the last matrix equation can be re-written as $Y = X^{-1}H^{-m}(HM)^m$, and this does translate into a system of linear equations in the entries of X^{-1} and Y. Thus, upon replacing the first matrix equation $XH = HX$ by the equivalent $X^{-1}H = HX^{-1}$, Eve ends up with a system of linear equations in the entries of X^{-1} and Y over the ground field. After solving this system and finding X and Y, Eve can recover the shared secret key K from the public transmissions as follows: $X(H^{-n}(HM)^n)Y = H^{-n}(XY)(HM)^n = H^{-n}H^{-m}(HM)^m(HM)^n = H^{-(m+n)}(HM)^{m+n} = K$.

This kind of attack may also work if the platform semigroup consists of matrices not over a field, but over a ring that can itself be embedded in a ring of matrices over a field, see e.g. [5], [7], [9].

Below we show how a little "tweak" of our protocol allows one to avoid this kind of attack.

1. Alice and Bob agree on public matrices $M, H \in G$, where H is invertible and M is not. Let the automorphism φ be conjugation by H.

2. Alice selects a private positive integer m, and Bob selects a private positive integer n. Alice also selects a private nonzero matrix R such that $R \cdot (HM) = O$ (the zero matrix), and Bob selects a private nonzero matrix S such that $S \cdot (HM) = O$. Such matrices R, S exist because the matrix HM is not invertible.

3. Alice computes $(M, \varphi)^m$. The first component of this pair is $H^{-m}(HM)^m$. Alice then sends the matrix $A = H^{-m}(HM)^m + R$ to Bob.

4. Bob computes $(M, \varphi)^n$. The first component of this pair is $H^{-n}(HM)^n$. Bob then sends the matrix $B = H^{-n}(HM)^n + S$ to Alice.

5. Alice computes $(B, x)\cdot(H^{-m}(HM)^m,\ \varphi^m) = (\varphi^m(B)\cdot H^{-m}(HM)^m,\ x\cdot\varphi^m)$. She only needs the first component of this pair, which is $H^{-(m+n)}(HM)^{m+n} + (H^{-m}SH^m)\cdot(H^{-m}(HM)^m)$. Since $S\cdot(HM) = O$, the second summand vanishes, so Alice ends up with $K_{Alice} = H^{-(m+n)}(HM)^{m+n}$.

6. Bob computes $(A, y)\cdot(H^{-n}(HM)^n,\ \varphi^n) = (\varphi^n(A)\cdot H^{-n}(HM)^n,\ y\cdot\varphi^n)$. He only needs the first component of this pair, which is $H^{-(m+n)}(HM)^{m+n} + (H^{-n}RH^n)\cdot(H^{-n}(HM)^n)$. Since $R\cdot(HM) = O$, the second summand vanishes, so Bob ends up with $K_{Bob} = H^{-(m+n)}(HM)^{m+n}$.

7. We therefore have $K_{Alice} = K_{Bob} = K$, the shared secret key.

Now let us see why the linear algebra attack as above does not work against this protocol. Here Eve would be looking for matrices X, Y and Z such that $XH = HX$, $Y(HM) = (HM)Y$, $Z\cdot(HM) = O$, and $XY + Z = H^{-m}(HM)^m + R$. The first three matrix equations do translate into a system of linear equations in the entries of X, Y and Z. However, with the last equation there is now a problem: no matter how Eve re-arranges it, she cannot avoid having a product of two unknown matrices, and this cannot be translated into a system of linear equations in their entries. If Eve attempts to solve the first three matrix equations first (by translating them into a system of linear equations in the entries of X, Y and Z), then she will have to deal with the fact that the linear system has multiple solutions. More specifically, the equation $XH = HX$ should have multiple solutions because, for example, any polynomial in the matrix H commutes with H, so at the very least, there are as many solutions of $XH = HX$ as there are elements in the ground field. (In fact, there are many more solutions than that.) The same consideration applies to the equation $Y(HM) = (HM)Y$. As for the equation $Z\cdot(HM) = O$, it has at least as many solutions as there are elements in the ground field because the matrix HM is not invertible. Thus, if the number of elements in the ground field is sufficiently large, this approach (trying to solve one or more of the first three matrix equations first and then plug the solution in $XY + Z = H^{-m}(HM)^m + R$) is computationally infeasible.

Finally, we point out that in the protocol in this section we used the fact that matrices over a field (or over any ring, for that matter) form not just a semigroup, but a ring where both multiplication and addition of matrices can be employed.

6 Security assumptions

In this section, we address the question of security of the particular instantiation of our protocol described in Section 4.

Recall that the shared secret key K in the protocol of Section 4 is the first component of $(M,\ \varphi)^{m+n}$, which is a product of the following matrices, with k running from 0 to $m+n-1$:

$$\varphi^k(M) = H^{-1}\psi(H^{-1})\cdots\psi^{k-1}(H^{-1})\psi^k(M)\psi^{k-1}(H)\cdots\psi(H)H.$$

The assumption is that it is computationally hard to recover K from the public information, i.e., from the matrices H, M, A, and B, where A and B are the first components of $(M,\ \varphi)^m$ and $(M,\ \varphi)^n$, respectively.

If we drop ψ and let φ be just the conjugation by H, then, due to cancellations in the above referenced product, the security assumption will look "nicer": it is computationally hard to retrieve the key $K = H^{-(m+n)}(HM)^{m+n}$ from the quadruple of matrices $(H,\ M,\ H^{-m}(HM)^m,\ H^{-n}(HM)^n)$, assuming that the matrices H and M do not commute, i.e., $HM \neq MH$.

However, "nicer" is not necessarily better in this context, as we have explained in Section 5, but in any case, we used a slightly more complex endomorphism φ in Section 4 to illustrate our point, which is: by varying an endomorphism used for a (semi)group extension, one can get a variety of new security assumptions.

To verify the robustness of our protocol in Section 4, we have experimentally addressed two questions. The first question is whether or not any information about the private exponent n is leaked from transmission. That is, for a random exponent n, how different is the first component of $(M,\varphi)^n$ from N, where N is a random matrix? The second point is to determine how different the final shared key is from a random matrix. More specifically, if Alice and Bob choose secret integers m and n respectively, how different is the first component of $(M,\varphi)^{n+m}$ from $(M,\varphi)^q$, where q is of the same bit size as $n+m$? To address these questions, we used the same strategy as in [4]; namely, we looked at the two distributions generated by the corresponding entries of the first component of $(M,\varphi)^n$ and N, where M and N are random matrices. (Note that each entry of a matrix is an element of the Galois field $\mathbb{GF}(2^{127})$ and is therefore representable by a bit string of length 127.) We repeated this process 500 times and generated a frequency distribution table for the two distributions. From the table, we produced $Q-Q$ (quantile) plots of the entries of the two matrices: the corresponding entries of the first component of $(M,\varphi)^n$ and a random matrix N. These plots essentially compare the cumulative distribution functions of two distributions. If the distributions are identical, the resulting graph will be a straight line, which is exactly what happened in our experiment.

The second experiment we carried out was similar to the first one, except in this case we were comparing the first components of $(M,\varphi)^q$ and $(M,\varphi)^{n+m}$, where n, m and q are random and all roughly of the same bit size. This experiment helps address the DDH (decisional Diffie-Hellman) assumption by comparing the shared secret key to a random key and ensuring that no information about the former is leaked. Again, our resulting $Q-Q$ plots suggest that the two distributions generated by these keys are in fact indistinguishable.

Finally, we point out that if the adversary works just with the determinants of the public matrices, then she can reduce the problem of recovering the private exponent n to the discrete logarithm problem for the pair $(\det M, (\det M)^n)$. To foil this kind of attack, it makes sense to select a matrix M with the determinant equal to 0 or 1, see Section 7 below.

7 Parameters and key generation

Private exponents m and n should be of the magnitude 2^t, where t is the security parameter, to make brute force search infeasible. Thus, m and n are roughly t bits long. In particular, for 127-bit security private keys should have size 127 bits, which is consistent with the fact that we are working with the Galois field $\mathbb{GF}(2^{127})$.

Our realization of the Galois field $\mathbb{GF}(2^{127})$ is the factor algebra $\mathbb{Z}_2[x]/\langle p(x)\rangle$, where $\langle p(x)\rangle$ is the ideal of the polynomial algebra $\mathbb{Z}_2[x]$ generated by the (irreducible) polynomial $p(x) = x^{127} + x^{63} + 1$. Elements of $\mathbb{GF}(2^{127})$ are therefore polynomials of degree at most 126 over $\mathbb{Z}_2$.

The public matrix M is selected as a random 2×2 matrix over the Galois field $\mathbb{GF}(2^{127})$, which means that each entry of M is a random element of $\mathbb{GF}(2^{127})$. A random element of $\mathbb{GF}(2^{127})$ is selected as a random bit string of length 127 corresponding to coefficients of a polynomial of degree at most 126 over $\mathbb{Z}_2$. For security reasons (see our Section 6) it is better to have the matrix M either non-invertible or have determinant 1. To select such a 2×2 matrix, we first select 3 entries randomly, and then select the remaining entry so that the determinant of the matrix is 0 or 1.

The bit complexity of the matrix M is $127 \cdot 4 = 508$ bits, and the procedure for sampling M is quite efficient. The whole public key consists of the matrix M and an invertible matrix H, so the total size of the public key is 1016 bits.

Then, we need to sample an *invertible* 2×2 matrix H over $\mathbb{GF}(2^{127})$. To do that, we sample a random 2×2 matrix as above, compute its determinant and check that it is not equal to 0 in $\mathbb{GF}(2^{127})$. If it is equal to 0 (this can happen with small probability), then we start over. Also, having computed the determinant of H, we then easily compute H^{-1}. We have to check that H does not commute with M, i.e., $HM \neq MH$. If it does, we select a different H.

Finally, in reference to the protocol in Section 5, we have to say how to sample a matrix R such that $R \cdot (HM) = O$ (the zero matrix) if the matrix HM is not invertible. This is done by using classical linear algebra; typically, the matrix equation $R \cdot (HM) = O$ will translate into a system of linear homogeneous equations in the entries of R, where the general solution is a one-parameter family. We then select the value of this parameter uniformly randomly among nonzero elements of the field $\mathbb{GF}(2^{127})$. If the general solution has more than one parameter (this might happen if participating matrices are larger than 2×2), then we randomly select the value of each parameter independently.

7.1 How to make sure the base element has a large order

We note that there is always a concern (also in the standard Diffie-Hellman protocol) about the order of a public element: if the order is too small, then a brute force attack may be feasible. In our situation, this concern is significantly alleviated by the fact that our transmissions are products of powers of different matrices rather than powers of a single matrix. Therefore, even if the order of one of the matrices happens to be small by accident, this does not mean that the whole product will go into loop of a small size.

However, if one wants a guarantee that the base element (M, φ) has a large order, this requires some extra effort. First we observe that the order of an element (g, φ) of a semidirect product tends to have the magnitude of the g.c.d. of the orders of the individual elements g and φ in their "native" (semi)groups. Of course, this statement and the statement in the previous paragraph are too informal, so now we will look at the second component of $(M, \varphi)^k$ specifically in our protocol from Section 4. This second component is φ^k, and here is how this endomorphism acts on an arbitrary 2×2 matrix S over $\mathbb{GF}(2^{127})$:

$$\varphi^k(S) = H^{-1}\psi(H^{-1}) \cdots \psi^{k-1}(H^{-1})\psi^k(S)\psi^{k-1}(H) \cdots \psi(H)H.$$

To bound the order k from below, we can look at the determinant of $\varphi^k(S)$. The determinant is obviously equal to $\det(\psi^k(S))$. Since we are working in characteristic 2, we have $\det(\psi^k(S)) = \det(\psi(S)^k) = (\det(\psi(S)))^k$ (recall that the endomorphism ψ acts by raising each entry of S to the power of 4). Now notice that if S is invertible, then $\det(S)$ is an element of the multiplicative group of the Galois field $\mathbb{GF}(2^{127})$; the order of this group is $2^{127} - 1$, which happens to be a prime number. Therefore, if $\det(S) \neq 1$, then $\det(S)$, as well as $\det(\psi(S))$, has order $2^{127} - 1$ in the multiplicative group of $\mathbb{GF}(2^{127})$. This gives a lower bound for the order of the second component of a base element (M, ϕ). The actual order should be much higher for "generic" matrices S and H, but if one wants a guaranteed lower bound, then $2^{127} - 1$ should be satisfactory.

8 Computational cost and run time

From the look of transmitted elements in our protocol in Section 4, it may seem that the parties have to compute a product of m (respectively, n) elements of the (semi)group G. However, since the parties actually compute powers of an element of a semigroup (which is an extension of G by an endomorphism), they can use the "square-and-multiply" method, as in the standard Diffie-Hellman protocol. Then there is a cost of applying an endomorphism φ to an element of G, and also of computing powers of φ applied to an element of G. In our situation in Section 4, applying conjugation by a matrix H amounts to just two multiplications of matrices in G (which boils down to 8 multiplications in $\mathbb{GF}(2^{127})$), and applying the endomorphism ψ does not, in fact, require any multiplications, just inserting "0" bits in a bit string representing an element of $\mathbb{GF}(2^{127})$, see [2].

Thus, the cost of computing $(M, \varphi)^n$ is $O(\log n)$, just as in the standard Diffie-Hellman protocol. However, there is no reduction modulo a large prime p; instead, all computations are done by utilizing known methods of fast computation in a field of characteristic 2. Specifically, we used the RELIC package [2] for computations in $\mathbb{GF}(2^{127})$.

With the parameters specified in our Section 7, the average run time of the whole key exchange protocol in Section 4 is 0.2 s on a typical desktop computer, whereas for the protocol in Section 5 the average run time is 0.03 s.

9 Conclusions

We have presented new key exchange protocols based on extension of a semigroup of matrices over $\mathbb{GF}(2^{127})$ by endomorphisms. It has some resemblance to the classical Diffie-Hellman protocol, but there are several distinctive features that, we believe, give our protocols important advantages:

• Even though the parties do compute a large power of a public element (as in the classical Diffie-Hellman protocol), they do not transmit the whole result, but rather just part of it.

• By varying automorphisms (or endomorphisms) used for extension, we get new security assumptions. We illustrate this point in the present paper by using a particular endomorphism which is a composition of an inner automorphism (i.e., conjugation by an invertible matrix) with the endomorphism that raises each entry of a given matrix to the power of 4.

• By working in a Galois field of characteristic 2, we make computation very efficient.

References

1. I. Anshel, M. Anshel, D. Goldfeld, and S. Lemieux, *Key agreement, the Algebraic Eraser, and lightweight cryptography*, Algebraic methods in cryptography, Contemp. Math. Amer. Math. Soc. **418** (2006), 1–34.
2. D. F. Aranha and C. P. L. Gouvêa, *RELIC is an Efficient Library for Cryptography*, http://code.google.com/p/relic-toolkit/
3. W. Diffie and M. E. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory **IT-22** (1976), 644–654.
4. M. Habeeb, D. Kahrobaei, C. Koupparis, V. Shpilrain, *Public key exchange using semidirect product of (semi)groups*, in: ACNS 2013, Lecture Notes Comp. Sc. **7954** (2013), 475–486.
5. M. Kreuzer, A. D. Myasnikov and A. Ushakov, *A linear algebra attack on group-ring-based key exchange protocols*, in: ACNS 2014, to appear.
6. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC-Press 1996.
7. C. Monico and M. Neusel, *Cryptanalysis of a system using matrices over group rings*, preprint.
8. C. Mullan, *Cryptanalysing variants of Stickel's key agreement protocol*, J. Math. Crypt. **4** (2011), 365-373.

9. A. D. Myasnikov and A. Ushakov, *Quantum algorithm for the discrete logarithm problem for matrices over finite group rings*, Groups, Complexity, Cryptology **6** (2014), 31–36.
10. A. G. Myasnikov, V. Shpilrain, and A. Ushakov, *Non-commutative cryptography and complexity of group-theoretic problems*, Amer. Math. Soc. Surveys and Monographs, 2011.
11. E. Stickel, *A New Method for Exchanging Secret Keys.* In: Proc. of the Third International Conference on Information Technology and Applications (ICITA05) 2 (2005), 426–430.
12. V. Shpilrain, *Cryptanalysis of Stickel's key exchange scheme*, in: Computer Science in Russia 2008, Lecture Notes Comp. Sc. **5010** (2008), 283-288.