CRYPTANALYSIS OF 'MAKE'

DANIEL BROWN

Email: danibrown@blackberry.com

NEAL KOBLITZ

Department of Mathematics, Box 354350,
University of Washington, Seattle, WA 98195 U.S.A.
Email: koblitz@uw.edu

JASON LEGROW

Department of Mathematics,
University of Auckland, Auckland, New Zealand
Email: jason.legrow@auckland.ac.nz

Abstract. In [5] Rahman and Shpilrain proposed a Diffie-Hellman style key exchange based on a semidirect product of $n \times n$-matrices over a finite field. We show that, using public information, an adversary can recover the agreed upon secret key by solving a system of n^2 linear equations.

1. Introduction

Ever since the invention in 1976 of the Diffie-Hellman key exchange [1] based on the multiplicative group of a finite field, researchers have investigated other groups and algebraic structures that can be used for similarly constructed key exchanges. A natural candidate was the general linear groups over the finite field $\mathbb{F}_q$ of q elements. However, in 1997 Menezes and Wu [3] proved that the discrete log problem in the group $\mathrm{GL}(n, q)$ of invertible $n \times n$ matrices is no more difficult than the discrete log problem in $\mathbb{F}_{q^n}$; therefore, a Diffie-Hellman key exchange in $\mathrm{GL}(n, q)$ has no advantage over the original Diffie-Hellman construction.

Despite this result of Menezes-Wu, researchers have continued to look for ways to use matrix groups and semigroups for Diffie-Hellman style key

Date: 9 April 2021.

exchange. Many of the specific constructions using such ideas have been broken, basically by exploiting an underlying linear structure. For example, Stickel's nonabelian key exchange [8] was cryptanalyzed by Shpilrain [7] three years later; and the instantiation of a key exchange based on semidirect products in [2] was cryptanalyzed shortly after in [4, 6].

The most recent construction of this type is the MAKE key exchange of Rahman and Shpilrain [5].[1] We show that MAKE also succumbs to a linear algebra attack — an adversary can recover the shared secret key by solving a system of n^2 linear equations. After describing the MAKE key exchange, we explain how the adversary can obtain such a linear system. We then give an alternative attack that leads to a system of n^4 linear equations that can be solved to give the entries in an $(n^2 \times n^2)$-matrix from which the shared key can immediately be found.

2. MAKE

The MAKE key exchange is based on the semidirect product of the additive group M_n of $n \times n$ matrices and the product of two multiplicative semigroups consisting of powers of fixed $H_1, H_2 \in M_n$. More concretely, the analog of the k-th power of a matrix $M \in M_n$ (here M plays the role of a generator of $\mathbb{F}_q^\times$ in the classical Diffie-Hellman protocol) is the sum

$$M + H_1 M H_2 + H_1^2 M H_2^2 + \cdots + H_1^{k-1} M H_2^{k-1}.$$

In the protocol, Alice chooses a secret positive integer x and Bob likewise chooses y. Alice can efficiently compute

$$A = M + H_1 M H_2 + H_1^2 M H_2^2 + \cdots + H_1^{x-1} M H_2^{x-1}, \tag{1}$$

and Bob computes the analogous sum B with x replaced by y. The shared key is then

$$\begin{aligned} z &= M + H_1 M H_2 + H_1^2 M H_2^2 + \cdots + H_1^{x+y-1} M H_2^{x+y-1} \\ &= A + H_1^x B H_2^x \\ &= B + H_1^y A H_2^y, \end{aligned} \tag{2}$$

which Alice and Bob can each compute using their secret key. Here H_1, H_2, and M are fixed parameters.

3. Telescoping

Note that although $H_1^x B H_2^x$ is not publicly known, by Equation (1) we have

$$M + H_1 A H_2 - A = H_1^x M H_2^x, \tag{3}$$

and so the cryptanalyst can immediately compute $H_1^x M H_2^x$ from public information.

[1] We refer to the latest posted version of MAKE as of this writing in early April 2021.

4. Attack using Cayley-Hamilton

If we can find the entries in the matrix $H_1^x B H_2^x$, we're done by Equation (2), because the shared secret key is obtained simply by adding A.

For a matrix $H \in M_n$ let H_{ij} denote its ij-entry, $0 \le i, j \le n-1$. Let $\mathrm{vec}(H)$ denote the column vector of height n^2 whose $(jn+i)$-th entry is H_{ij}; thus, $\mathrm{vec}(H)$ is obtained by simply stringing the second column of H under the first column, the third column under the second column, and so on.

We now regard $H_1, H_2 \in M_n$ and a positive integer x as fixed, and define a function $L(Y) = L_{H_1,H_2,x}(Y)$ from M_n to M_{n^2} by setting

$$(L(Y))_{jn+i,hn+g} = (H_1^g Y H_2^h)_{i,j}, \qquad 0 \le i, j, g, h \le n-1.$$

In other words, the $(hn+g)$-th column of $L(Y)$ is $\mathrm{vec}(H_1^g Y H_2^h)$. By the Cayley-Hamilton theorem, we can write

$$H_1^x = \sum_{g=0}^{n-1} p_g H_1^g \qquad \text{and} \qquad H_2^x = \sum_{h=0}^{n-1} q_h H_2^h.$$

Define $S \in M_n$ by $S_{ij} = p_i q_j$ and set $s = \mathrm{vec}(S)$.

Lemma 1. *For $Y \in M_n$,*

$$L(Y)s = \mathrm{vec}(H_1^x Y H_2^x).$$

Proof. The proof follows by an elementary computation. □

Remark 1. *Of course, the cryptanalyst does not know x, H_1^x, or H_2^x, and so cannot compute s. The purpose of Lemma 1 is to ensure existence of a solution. The characteristic polynomials of H_1^x and H_2^x are used only for existence. The cryptanalyst does not compute the characteristic polynomial of any matrix.*

Lemma 2. *If u is any vector such that*

$$L(Y)u = 0,$$

then for any positive integer ℓ we also have

$$L(H_1^\ell Y H_2^\ell)u = 0.$$

Proof. It follows from the definitions that

$$\begin{aligned} L(H_1^\ell Y H_2^\ell)u &= \mathrm{vec}\left(\sum_{g,h=0}^{n-1} u_{hn+g}(H_1^g(H_1^\ell Y H_2^\ell)H_2^h)\right) \\ &= \mathrm{vec}\left(H_1^\ell\left(\sum_{g,h=0}^{n-1} u_{hn+g}(H_1^g Y H_2^h)\right)H_2^\ell\right) \\ &= \mathrm{vec}\left(H_1^\ell \mathrm{vec}^{-1}(L(Y)u)H_2^\ell\right) \\ &= 0. \end{aligned}$$

□

The adversary first computes $H_1^x M H_2^x$ by Equation (3) and then solves the system of n^2 linear equations

$$L(M)t = \mathrm{vec}(H_1^x M H_2^x)$$

for t. By Lemma 1 with $Y = M$, this system has at least one solution s; and by Lemma 1 with $Y = B$, the same vector s also solves the system

$$L(B)s = \mathrm{vec}(H_1^x B H_2^x). \tag{4}$$

We claim that the adversary's vector t also satisfies

$$L(B)t = \mathrm{vec}(H_1^x B H_2^x).$$

To see this, we set $u = t - s$. We apply Lemma 2 with $Y = M$ for $\ell = 0, 1, \ldots, y-1$, and add. We find that

$$\begin{aligned} 0 &= L(M)u + L(H_1 M H_2)u + \cdots + L(H_1^{y-1} M H_2^{y-1})u \\ &= L(M + H_1 M H_2 + \cdots + H_1^{y-1} M H_2^{y-1})u \\ &= L(B)u. \end{aligned}$$

Hence $L(B)t = L(B)s + L(B)u = \mathrm{vec}(H_1^x B H_2^x)$ by Equation (4). From B and t the adversary can now recover $H_1^x B H_2^x$ and hence the shared key $z = A + H_1^x B H_2^x$.

5. Attack by simulating Bob

Recall that the tensor product of an $(m_1 \times n_1)$-matrix X and an $(m_2 \times n_2)$-matrix Y is the $(m_1 m_2 \times n_1 n_2)$-matrix $X \otimes Y$ given by

$$\begin{bmatrix} X_{1,1}Y & X_{1,2}Y & \cdots & X_{1,n_1}Y \\ X_{2,1}Y & X_{2,2}Y & \cdots & X_{2,n_1}Y \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots \\ X_{m_1,1}Y & X_{m_1,2}Y & \cdots & X_{m_1,n_1}Y \end{bmatrix}$$

We have the following identity for three matrices X, Y, Z whenever the product XYZ is defined:

$$\mathrm{vec}(XYZ) = (Z^T \otimes X)\mathrm{vec}(Y). \tag{5}$$

We also note that $(X \otimes Y)^\ell = X^\ell \otimes Y^\ell$. In particular,

$$\mathrm{vec}(H_1^\ell Y H_2^\ell) = (H_2^T \otimes H_1)^\ell \mathrm{vec}(Y).$$

From this and Equation (2) it follows that, if we can determine the unknown $(n^2 \times n^2)$-matrix $H = (H_2^T \otimes H_1)^x$, we just have to compute $H\mathrm{vec}(B) + \mathrm{vec}(A)$ to get the shared private key.

We find the n^4 unknown entries of H by obtaining n^4 independent linear equations that they satisfy. We do this in two ways: (1) by using a general commutativity property, and (2) by simulating Bob with various choices of his secret y.

(1) The first method for finding equations uses only the parameters H_1, H_2 and not the values A, B of a particular exchange of keys. Let I_n denote the $n \times n$ identity matrix. The commutation relations

$$(I_n \otimes H_1)(H_2^T \otimes H_1)^x(I_{n^2}) = (I_{n^2})(H_2^T \otimes H_1)^x(I_n \otimes H_1)$$
$$(H_2^T \otimes I_n)(H_2^T \otimes H_1)^x(I_{n^2}) = (I_{n^2})(H_2^T \otimes H_1)^x(H_2^T \otimes I_n)$$

give us Equations (6) below, where we again let $H = (H_2^T \otimes H_1)^x$ denote our unknown $(n^2 \times n^2)$-matrix and apply Equation (5):

$$\begin{aligned} &(I_{n^2} \otimes (I_n \otimes H_1) - (I_n \otimes H_1^T) \otimes I_{n^2})\mathrm{vec}(H) = 0 \\ &(I_{n^2} \otimes (H_2^T \otimes I_n) - (H_2 \otimes I_n) \otimes I_{n^2})\mathrm{vec}(H) = 0 \end{aligned} \tag{6}$$

In numerical experiments with randomly chosen rank-$(n-1)$ matrices H_1 and H_2, these give $n^2(n^2-1)$ independent equations for the n^4 entries of H, that is, just n^2 fewer than we need.

(2) Perhaps the simplest identity satisfied by H is: $HM = H_1^x M H_2^x$, where the right side is publicly known by Equation (3). This gives n^2 linear equations for the entries of H. We can regard this as the case $y = 0$ of the key exchange, that is, $B = 0$, $z = A$. For any integer $y \geq 0$, we can write the equation

$$H(H_1^y M H_2^y) = H_1^{x+y} M H_2^{x+y},$$

where the adversary, simulating Bob, chooses arbitrary[2] y and then knows both sides except for the entries of H. If the value $y = 0$ does not give n^2 independent equations that are also independent of the $n^2(n^2-1)$ equations from the commutation relations, then the adversary continues with $y = 1, 2, 3, \ldots$ until they get the required number of independent equations. Numerical experiments indicate that a very few small values of y are sufficient.

Remark 2. *In §4 our first method was proved to give a system of linear equations any of whose solutions leads to the secret key. In §5 we have heuristics and numerical evidence, but no proof, to support the belief that the method quickly leads to the required number of independent equations.*

6. Conclusion

The MAKE key exchange is insecure; the shared key can be recovered by linear algebra in polynomial time. This shows once again that even a matrix-based protocol that seems much more complicated than a standard Diffie-Hellman key exchange may have an essential linearity that makes it vulnerable. Caution seems to be especially necessary when considering matrix-based cryptosystems.

[2] In the actual protocol, Bob chooses a very large integer y, but in the cryptanalysis algorithm y can be very small.

Acknowledgment

We thank Vladimir Shpilrain for helpful correspondence.

References

[1] Diffie, W., and M. Hellman, New Directions in Cryptography, *IEEE Transactions on Information Theory*, **IT-22** (1976), 644-654.

[2] Habeeb, M., D. Kahrobaei, C. Koupparis, and V. Shpilrain, Public key exchange using semidirect product of (semi)groups, *ACNS 2013*, LNCS **7954** (2013) , 475-486.

[3] Menezes, A. J., and Y.-H. Wu, The discrete logarithm problem in $\mathrm{GL}(n, q)$, *Ars Combinatoria*, **47** (1997), 23-32.

[4] Myasnikov, A. G., and V. Roman'kov, A linear decomposition attack, *Groups, Complexity, Cryptology*, **7** (2015), 81-94.

[5] Rahman, N., and V. Shpilrain, MAKE: a Matrix Action Key Exchange, https://eprint.iacr.org/2021/116.pdf.

[6] Roman'kov, V., Linear decomposition attack on public key exchange protocols using semidirect products of (semi)groups, http://arxiv.org/abs/1501.01152.

[7] Shpilrain, V., Cryptanalysis of Stickel's key exchange scheme, *Computer Science in Russia 2008*, LNCS **5010** (2008), 283-288.

[8] Stickel, E., A New Method for Exchanging Secret Keys, *Proc. Third Intern. Conf. on Information Technology and Applications (ICITA 05)*, *Contemp. Math.* **2** (2005), 426-430.