The Discrete Logarithm Problem in $GL(n,q)$

Alfred J. Menezes and Yi-Hong Wu
Dept. of Discrete and Statistical Sciences
120 Math Annex
Auburn University, Auburn, AL 36849

Abstract

This paper presents a probabilistic polynomial-time reduction of the discrete logarithm problem in the general linear group $GL(n,q)$ to the discrete logarithm problem in some small extension fields of $\mathbb{F}_q$.

1 Introduction

The Diffie-Hellman key exchange [6] is a protocol whereby two entities A and B can, by a sequence of transmissions over a public channel, agree upon a secret cryptographic key. The method is as follows. A and B first choose a (multiplicatively written) finite abelian group G and some element $\alpha \in G$. A then selects a random integer a and transmits α^a to B. B in turn selects a random integer b and transmits α^b to A. Both A and B can then determine α^{ab}, which is their shared secret key.

An eavesdropper C monitoring the transmission between A and B would know G, α, α^a, and α^b. The parameters G and α should be chosen so that it is computationally infeasible for C to then determine α^{ab}. Certainly, if C could compute either a or b, then C could determine α^{ab}. The problem of determining a given α and $\beta = \alpha^a$ is called the *discrete logarithm problem* in G. The integer a, which is unique if restricted to the range $[0, \text{order}(\alpha) - 1]$, is called the *discrete logarithm* of β to the base α. It is an open problem to decide whether or not determining α^{ab} is equivalent to computing discrete logarithms in G.

The best algorithms that are known for solving the discrete logarithm problem in an arbitrary group G are the exponential square root attacks (see [10]) that have a running time that is roughly proportional to the square root of the largest prime factor of l, where l is the order of α. Consequently, if G and α are chosen so that l has a large prime factor, then these attacks can be avoided.

Let $\mathbb{F}_q$ denote the finite field of order q, and let $q = p^m$ where p is the characteristic of $\mathbb{F}_q$. In [6], $G = \mathbb{F}_q^*$, the multiplicative group of $\mathbb{F}_q$, was proposed as a candidate for implementing the Diffie-Hellman key exchange. There are probabilistic subexponential-time algorithms known for computing logarithms in $\mathbb{F}_q$ (see [5] for the case q a prime, [14] for the case where $p = 2$, and [1] for the general situation). A *subexponential-time algorithm* is an algorithm whose running time is

$$O\left(e^{(c+o(1))\; z^d (\log z)^{1-d}} \right),$$

where z is the input size, c is a constant, and $0 < d < 1$. These algorithms are an asymptotic improvement on the general algorithms mentioned in the previous paragraph. For cryptographic purposes we are interested in groups for which subexponential algorithms for the corresponding discrete logarithm are not known. Additionally, for efficient and practical implementation, the group operation should be relatively easy to apply.

It was for these reasons that the group of non-singular matrices over a finite field [15], the group of points on an elliptic curve ([8] and [12]), the jacobian of a hyperelliptic curve defined over a finite field [9], and the class group of an imaginary quadratic field [3] have been proposed for cryptographic use. In [11] it was shown how the discrete logarithm problem in the special class of matrices considered in [15] can be reduced to the discrete logarithm problem in some extensions of the underlying field. This paper extends these results to show how the discrete logarithm problem in $GL(n, q)$ can be reduced in probabilistic polynomial time to the logarithm problem in small extensions of $\mathbb{F}_q$. This demonstrates that the group $GL(n, q)$ offers no significant advantage over finite fields for the implementation of cryptographic protocols whose security is based on the difficulty of computing discrete logarithms in a group.

The remainder of the paper is organized as follows. Some basic results from linear algebra are first reviewed in Section 2. Section 3 describes the orders of elements in $GL(n, q)$. In Section 4 we describe a polynomial-time algorithm for computing the Jordan canonical form of a matrix. Section 5 presents the reduction. Finally, Section 6 makes some concluding remarks.

2 Background

We review some basic concepts from linear algebra. For more details the reader is referred to Horn and Johnson [7].

Let $q = p^m$ be a prime power. $\mathbb{F}_q$ will denote the finite field of order q. The set of all $n \times n$ matrices with entries from $\mathbb{F}_q$ is denoted $M_n(q)$. The *general linear group*, denoted $GL(n,q)$ or $GL(n,\mathbb{F}_q)$, is the set of all non-singular $n \times n$ matrices over $\mathbb{F}_q$ under matrix multiplication. The order of $GL(n,q)$ is $\prod_{i=0}^{n-1}(q^n - q^i)$.

Let $A \in M_n(\mathbb{F}_q)$. The rank of A is denoted $r(A)$ and the null space of A is denoted $N(A)$. The *characteristic polynomial* of A is $p_A(x) = \det(A - Ix)$; $p_A(x)$ is a polynomial of degree n in $\mathbb{F}_q[x]$. Let E denote the splitting field of $p_A(x)$ over $\mathbb{F}_q$. The roots $\lambda_1, \lambda_2, \ldots, \lambda_h$ of $p_A(x)$ in E are the *eigenvalues* of A. The (algebraic) *multiplicity* of an eigenvalue λ is the multiplicity of λ as a root of $p_A(x)$. A *Jordan block of order d corresponding to λ* is a $d \times d$ upper-triangular matrix of the form

$$J_d(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda & 1 & \cdots & 0 & 0 \\ 0 & 0 & \lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda \end{bmatrix}$$

A *Jordan matrix* is a direct sum of Jordan blocks. It is well known that for every matrix $A \in M_n(\mathbb{F}_q)$ there exists a matrix $Q \in GL(n,E)$ such that $Q^{-1}AQ = J_A$, where

$$J_A = J_{n_1}(\lambda_1) \oplus J_{n_2}(\lambda_2) \oplus \cdots \oplus J_{n_s}(\lambda_s)$$

is a Jordan matrix, $\lambda_1, \lambda_2, \ldots, \lambda_s$ are the eigenvalues of A (not necessarily distinct), and $\sum_{i=1}^{s} n_i = n$. The Jordan matrix J_A is unique up to rearrangement of the component Jordan blocks and is called the *Jordan canonical form* of A.

Let λ be an eigenvalue of $A \in GL(n,q)$ of multiplicity m. Let c be the smallest positive integer for which $r(A - \lambda I)^c = r(A - \lambda I)^{c+1}$; it is certainly the case that $c \leq n$. Then the number of Jordan blocks in J_A corresponding to λ is $n - r(A - \lambda I)$, and c is the size of the largest such block. More completely, the number of Jordan blocks of size at least k in J_A corresponding to λ is $r(A - \lambda I)^{k-1} - r(A - \lambda I)^k$ (where $(A - \lambda I)^0$ is defined to be I). It follows that the number of Jordan blocks of size exactly k in J_A corresponding to λ is

$$r(A - \lambda I)^{k+1} - 2r(A - \lambda I)^k + r(A - \lambda I)^{k-1}.$$

If ν is a conjugate of λ, i.e., $\nu = \lambda^{q^j}$ for some j, then ν is also an eigenvalue of A and there are an equal number of Jordan blocks of each size k corresponding to λ and ν in J_A.

Let λ be an eigenvalue of A. A non-zero vector u is called a *generalized eigenvector* of rank t corresponding to λ if $(A-\lambda I)^t u = 0$ and $(A-\lambda I)^{t-1} u \neq 0$. Let u be such a vector, and define the set of vectors $S = \{u_1, u_2, \ldots, u_t\}$ by

$$u_t = u \text{ and } u_i = (A-\lambda I)u_{i+1} \text{ for } i = t-1, \ldots, 2, 1. \tag{1}$$

Then u_i is a generalized eigenvector of rank i corresponding to λ, and the set S is linearly independent.

3 Orders of matrices in $GL(n,q)$

This section describes the orders of elements in $GL(n,q)$. The results are not new, and were proven by Niven [13] in 1948. The alternate proof using the Jordan canonical form that is presented here will facilitate the discussion for the remainder of the paper.

Lemma 1 *The order of the Jordan block $J = J_d(\lambda)$ is $\mathrm{ord}(\lambda)p\{d\}$, where $p\{d\}$ denotes the smallest power of p greater than or equal to d.*

Proof: Let $s = \mathrm{ord}(\lambda)$ and $u = p\{d\}$ and note that $\gcd(s,u) = 1$. It can be shown that J^l is an upper triangular matrix with (i,j)-entry equal to $\lambda^{l-j+i}\binom{l}{j-i}$ for $1 \leq i \leq j \leq d$. Thus $J^l = I$ if and only if $\lambda^l = 1$ and $\binom{l}{k} \equiv 0 \pmod{p}$ for each $1 \leq k \leq d-1$.
Now, $(1+x)^{su} = (1+x^u)^s$ in $\mathbb{Z}_p[x]$. Comparing coefficients of x^k yields $\binom{su}{k} \equiv 0 \pmod{p}$, for each $1 \leq k \leq u-1$. Since $\lambda^{su} = 1$, it follows that $J^{su} = I$ and so $\mathrm{ord}(J) | su$.
Suppose now that $\mathrm{ord}(J) = sw$, where w is a divisor of u, $w < u$. Since $J^{sw} = I$, we have $\binom{sw}{k} \equiv 0 \pmod{p}$ for each $1 \leq k \leq d-1$. In particular, $\binom{sw}{w} \equiv 0 \pmod{p}$ since $w \leq d-1$. But equating coefficients of x^w in $(1+x)^{sw} = (1+x^w)^s$ yields $\binom{sw}{w} \equiv s \pmod{p}$ where $s \not\equiv 0 \pmod{p}$, thus contradicting the previous statement. We conclude that $\mathrm{ord}(J) = su$, as required. $\square$

Theorem 2 *Let $A \in GL(n,q)$. Let the distinct eigenvalues of A in E be $\lambda_1, \lambda_2, \ldots, \lambda_h$. Then the order of A is*

$$\mathrm{ord}(A) = \mathrm{lcm}(\mathrm{ord}(\lambda_1), \mathrm{ord}(\lambda_2), \ldots, \mathrm{ord}(\lambda_h))p\{t\},$$

where t is the size of the largest Jordan block in J_A.

Proof: Let the Jordan canonical form of A be $J_A = J_1 \oplus J_2 \oplus \cdots \oplus J_s$ and let $Q \in GL(n, E)$ be a matrix such that $Q^{-1}AQ = J_A$. Then

$$\mathrm{ord}(A) = \mathrm{ord}(J_A) = \mathrm{lcm}(\mathrm{ord}(J_1), \mathrm{ord}(J_2), \ldots, \mathrm{ord}(J_s)).$$

The result now follows from Lemma 1. □

4 Computing the Jordan canonical form

An algorithm involving elements of $GL(n, q)$ is a polynomial-time algorithm if its running time is bounded by a polynomial in n and $\log q$. The classical techniques for matrix addition, multiplication and Gaussian elimination take $O(n^2)$, $O(n^3)$ and $O(n^3)$ $\mathbb{F}_q$-operations respectively, where an $\mathbb{F}_q$-operation takes $O((\log q)^2)$ bit operations.

Let $A \in M_n(\mathbb{F}_q)$. Let $p_A(x)$ be the characteristic polynomial of A, and suppose that its factorization over $\mathbb{F}_q$ is $p_A(x) = f_1^{e_1} f_2^{e_2} \cdots f_s^{e_s}$, where the f_i are distinct irreducible polynomials of degree m_i in $\mathbb{F}_q[x]$. Then the smallest extension field containing all the eigenvalues of A is $E = \mathbb{F}_{q^k}$, where $k = \mathrm{lcm}(m_1, m_2, \ldots, m_s)$.

The algorithms usually described in textbooks (for example [7]) for computing the Jordan canonical form J_A work in the field E. However, as the following argument shows, the field E is very big in general, and hence the algorithms are not polynomial-time algorithms. Suppose that m_i is the i^{th} prime number, $1 \le i \le s$, and each $e_i = 1$. Let $d = m_s + 1$. Then by Corollary 1 of [16], we have

$$s \; < \; (1.3d)/(\log d).$$

Hence

$$n \; = \; \sum_{i=1}^{s} m_i \; < \; (1.3d^2)/(\log d) \; < \; 1.3d^2,$$

and so $d > 0.87\sqrt{n}$. And, by Theorem 10 of [16], we have that for $d \ge 101$,

$$k \; = \; \prod_{i=1}^{s} m_i \; > \; e^{0.84d} \; > \; e^{0.7\sqrt{n}}.$$

To overcome the problem of k being too big, we do the computations in the smaller extension fields $\mathbb{F}_{q^{m_1}}$, $\mathbb{F}_{q^{m_2}}$, $\ldots$, $\mathbb{F}_{q^{m_s}}$ in turn, instead of working in the field $\mathbb{F}_{q^k}$.

Algorithm 1 (Computing the Jordan canonical form)
Input: A matrix $A \in GL(n, q)$
Output: The Jordan canonical form J_A of A.

1. Use the Hessenberg algorithm [4, page 55] to find the characteristic polynomial $p_A(x)$ of A.

2. Find the factorization of $p_A(x)$ over $\mathbb{F}_q$ using, for example, Ben-Or's algorithm [2]: $p_A(x) = f_1^{e_1} f_2^{e_2} \cdots f_s^{e_s}$, where each f_i is an irreducible polynomial of degree m_i. Let the roots of f_i in $\mathbb{F}_{q^{m_i}}$ be α_{ij}, $1 \leq j \leq m_i$. Note that we may conveniently represent the field $\mathbb{F}_{q^{m_i}}$ as $\mathbb{F}_q[x]/(f_i(x))$. In this representation, we simply have $\alpha_{i1} = x$, and $\alpha_{ij} = x^{q^{j-1}} \bmod f_i(x)$ for $2 \leq j \leq m_i$.

3. For i from 1 to s, do the following:

   3.1 Set $r_0 \leftarrow n$.

   3.2 Compute $(A - \alpha_{i1} I)^l$ and $r_l = r(A - \alpha_{i1} I)^l$ for $l = 1, 2, \ldots, c, c+1$, where c is the smallest positive integer such that $r_c = r_{c+1}$.

   3.3 Let J_{i1} be the direct sum of $(r_{l+1} - 2r_l + r_{l-1})$ Jordan blocks of order l corresponding to α_{i1}, $1 \leq l \leq c$.

   3.4 Let J_{ij} be the same matrix as J_{i1} but with α_{i1} replaced by α_{ij}, $2 \leq j \leq m_i$.

   3.5 Set $J_i \leftarrow J_{i1} \oplus J_{i2} \oplus \cdots \oplus J_{im_i}$.

4. Set $J_A \leftarrow J_1 \oplus J_2 \oplus \cdots \oplus J_s$.

Theorem 3 *Algorithm 1 takes expected polynomial time.*

Proof: Hessenberg's algorithm takes polynomial time, while Ben-Or's algorithm takes expected polynomial time. In each iteration of step 3, the computations are performed in the field $\mathbb{F}_{q^{m_i}}$. Since $m_i \leq n$, we have $\log q^{m_i} \leq n \log q$, and so each iteration of step 3 takes polynomial time. Finally, since step 3 is iterated s times and $s \leq n$, we see that the expected running time of Algorithm 1 is bounded by a polynomial in n and $\log q$. $\Box$

5 The reduction

The discrete logarithm problem in $GL(n, q)$ is to find l, given matrices A and $B = A^l$ in $GL(n, q)$. We show how this problem can be reduced in expected polynomial time to the problem of computing logarithms in several small extension fields $\mathbb{F}_{q^{m_i}}$.

Let the factorization of the characteristic polynomial $p_A(x)$ of A over $\mathbb{F}_q$ be $p_A(x) = f_1^{e_1} f_2^{e_2} \cdots f_s^{e_s}$, where $\deg(f_i) = m_i$. Let λ_i be a root of f_i in $\mathbb{F}_{q^{m_i}}$, and let t be the order of the largest Jordan block in J_A. Then since all the roots of f_i have the same order and Jordan block structure, by Theorem 2 we have

$$\mathrm{ord}(A) = \mathrm{lcm}(\mathrm{ord}(\lambda_1), \mathrm{ord}(\lambda_2), \ldots, \mathrm{ord}(\lambda_s))p\{t\},$$

and l is uniquely determined modulo this number.

To determine $l \bmod \mathrm{ord}(\lambda_i)$ we find an eigenvector μ_i corresponding to λ_i. If Q_i is a non-singular matrix whose first column is μ_i, then the first column of the matrix $Q_i^{-1}AQ_i$ is $(\lambda_i, 0, \ldots, 0)^T$. Then

$$D_i \;=\; Q_i^{-1}BQ_i \;=\; Q_i^{-1}A^lQ_i \;=\; (Q_i^{-1}AQ_i)^l,$$

and so the $(1,1)$ entry of D_i is λ_i^l. The quantity $l \bmod \mathrm{ord}(\lambda_i)$ can thus be obtained by computing the logarithm of λ_i^l to the base λ_i.

If $t > 1$ then $l \bmod p\{t\}$ is obtained as follows. Let λ be an eigenvalue of A which has a corresponding Jordan block J of order t. Find a generalized eigenvector μ of rank t corresponding to λ by solving $(A - \lambda I)^t y = 0$, $(A - \lambda I)^{t-1} y \neq 0$. If Q is an invertible matrix whose first t columns are the vectors $u_1, u_2, \ldots, u_t$ defined by (1) then the first t columns of the matrix $Q^{-1}AQ$ has the form

$$\begin{bmatrix} J \\ 0 \end{bmatrix}.$$

Hence, the $t \times t$ submatrix in the upper left-hand corner of $D = Q^{-1}BQ$ is J^l. Now, if $p\{t\} = p$, then since the $(1,1)$ entry of D is λ^l and the $(1,2)$ entry of D is $l\lambda^{l-1}$, $l \bmod p$ can be easily obtained. If $p\{t\} \geq p^2$, then $p < t \leq n$ and $p\{t\} < n^2$. In this case, let $s = \mathrm{ord}(\lambda)$, $l' = l \bmod s$, and compute $J^{l'}$, $J^{l'+s}$, $J^{l'+2s}, \ldots$ until $J^{l'+js} = J^l$, in which case $l \bmod p\{t\} = j$.

A detailed description of the reduction is given below.

Algorithm 2 (Reduce the logarithm problem in $GL(n,q)$ to the logarithm problem in $\mathbb{F}_{q^{m_i}}$, $1 \leq i \leq s$.)
Input: Matrices A, $B \in GL(n,q)$ with $B = A^l$.
Output: The integer l.

1. Use the Hessenberg algorithm to find the characteristic polynomial $p_A(x)$ of A.

2. Find the factorization of $p_A(x)$ over $\mathbb{F}_q$: $p_A(x) = f_1^{e_1} f_2^{e_2} \cdots f_s^{e_s}$, where each f_i is an irreducible polynomial of degree m_i. Let the roots of f_i in $\mathbb{F}_{q^{m_i}}$ be α_{ij}, $1 \leq j \leq m_i$. Note that we may conveniently represent the field $\mathbb{F}_{q^{m_i}}$ as $\mathbb{F}_q[x]/(f_i(x))$. In this representation, we simply have $\alpha_{i1} = x$, and $\alpha_{ij} = x^{q^{j-1}} \bmod f_i(x)$ for $2 \leq j \leq m_i$.

3. For i from 1 to s do the following:

   3.1 Compute $(A-\alpha_{i1}I)^l$ and $r_l = r(A-\alpha_{i1}I)^l$ for $l = 1, 2, \ldots, c, c+1$, where c is the smallest positive integer such that $r_c = r_{c+1}$.

   3.2 Find an eigenvector μ_i corresponding to α_{i1} by solving $(A - \alpha_{i1}I)y = 0$.

   3.3 Construct a matrix $Q_i \in GL(n, q^{m_i})$ whose first column is μ_i.

   3.4 Compute $D_i \leftarrow Q_i^{-1}BQ_i$.

   3.5 The $(1,1)$ entry of D_i is α_{i1}^l, and so one can find l modulo $\mathrm{ord}(\alpha_{i1})$ by solving a discrete logarithm problem in $\mathbb{F}_{q^{m_i}}$.

4. Let t be the maximum of the c values found in step 3.1. If $t > 1$ then do the following.

   4.1 Let $\lambda \in \mathbb{F}_{q^m}$ be an eigenvalue which has a corresponding Jordan block of size t.

   4.2 Find a basis B_1 for $N((A-\lambda I)^{t-1})$.

   4.3 Find a basis B_2 for $N((A-\lambda I)^t)$.

   4.4 Hence find a vector u in B_2 which is not in the subspace spanned by B_1. (u is a generalized eigenvector of rank t.)

   4.5 Set $u_t \leftarrow u$, and $u_j \leftarrow (A-\lambda I)u_{j+1}$ for $j = t-1, \ldots, 2, 1$.

   4.6 Construct a matrix $Q \in GL(n, q^m)$ whose first t columns are $u_1, u_2, \ldots, u_t$.

   4.7 Compute $Q^{-1}AQ$ and $D \leftarrow Q^{-1}BQ$.

   4.8 The $(1,1)$ entry of D is λ^l and the $(1,2)$ entry of D is $l\lambda^{l-1}$. If $p\{t\} = p$ then first compute λ^{l-1} as λ^l/λ, and then divide $l\lambda^{l-1}$ by λ^{l-1} to obtain $l \bmod p$.

   4.9 If $p\{t\} \geq p^2$ then let J be the $t \times t$ Jordan block in the upper left-hand corner of $Q^{-1}AQ$. Set $s \leftarrow \mathrm{ord}(\lambda)$, $l' \leftarrow l \bmod s$ (which was computed in step 3), and compute $J^{l'}$, $J^{l'+s}$, $J^{l'+2s}, \ldots$ until $J^{l'+js}$ is equal to the $t \times t$ matrix in the upper left-hand corner of D. Then $l \bmod p\{t\} = j$.

5. Find $l \bmod \mathrm{ord}(A)$ by using the generalized Chinese remainder theorem.

Theorem 4 *Algorithm 2 is an expected polynomial-time reduction of the discrete logarithm problem in $GL(n,q)$ to the discrete logarithm problem in $\mathbb{F}_{q^{m_i}}$, $1 \leq i \leq s$.*

Proof: Hessenberg's algorithm takes polynomial time, while Ben-Or's algorithm takes expected polynomial time. Each iteration of step 3 involves linear algebra over $\mathbb{F}_{q^{m_i}}$, where $m_i \leq n$. Since $\log q^{m_i} \leq n \log q$ and $s \leq n$, step 3 is a polynomial-time reduction. Finally, step 4 involves linear algebra over $\mathbb{F}_{q^m}$, where $m \leq n$. If $p\{t\} \geq p^2$, then $p\{t\} < n^2$, and so the process of computing $J^{l'+js}$ in step 4.9 is iterated at most n^2 times. This proves the statement of the Theorem. $\square$

6 Conclusions

We have shown that the discrete logarithm problem in $GL(n, q)$ is no more difficult than the discrete logarithm problem in $\mathbb{F}_{q^n}$. Since the group operation in $GL(n, q)$ is computationally more expensive than the group operation in $\mathbb{F}_{q^n}$, the former group offers no advantage over finite fields for the implementation of cryptographic protocols whose security is based on the difficulty of computing logarithms in a group.

References

[1] L. Adleman and J. DeMarrais, "A subexponential algorithm for discrete logarithms over all finite fields", *Mathematics of Computation*, **61** (1993), 1-15.

[2] M. Ben-Or, "Probabilistic algorithms in finite fields", *22nd Annual Symposium on Foundations of Computer Science*, 394-398, 1981.

[3] J. Buchmann and H. Williams, "A key-exchange system based on imaginary quadratic fields" *Journal of Cryptology*, **1** (1988), 107-118.

[4] H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, Berlin, 1993.

[5] D. Coppersmith, A. Odlyzko and R. Schroeppel, "Discrete logarithms in $GF(p)$", *Algorithmica*, **1** (1986), 1-15.

[6] W. Diffie and M. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, **22** (1976), 644-654.

[7] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[8] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, **48** (1987), 203-209.

[9] N. Koblitz, "Hyperelliptic cryptosystems", *Journal of Cryptology*, **1** (1989), 139-150.

[10] K. McCurley, "The discrete logarithm problem", *Cryptology and Computational Number Theory*, Proceedings of Symposia in Applied Mathematics, **42** (1990), 49-74.

[11] A. Menezes and S. Vanstone, "A note on cyclic groups, finite fields, and the discrete logarithm problem", *Applicable Algebra in Engineering, Communication and Computing*, **3** (1992), 67-74.

[12] V. Miller, "Uses of elliptic curves in cryptography", *Advances in Cryptology – Proceedings of Crypto '85*, Lecture Notes in Computer Science, **218** (1986), Springer-Verlag, 417-426.

[13] I. Niven, "Fermat's theorem for matrices", *Duke Mathematical Journal*, **15** (1948), 823-826.

[14] A. Odlyzko, "Discrete logarithms in finite fields and their cryptographic significance", *Advances in Cryptology – Proceedings of Eurocrypt '84*, Lecture Notes in Computer Science, **209** (1985), Springer-Verlag, 224-314.

[15] R. Odoni, V. Varadharajan and R. Sanders, "Public key distribution in matrix rings", *Electronics Letters*, **20** (1984), 386-387.

[16] J. Rosser and L. Schoenfeld, "Approximate formulas for some functions of prime numbers", *Illinois Journal of Mathematics*, **6** (1962), 64-94.