MATHEMATICS OF COMPUTATION
Volume 67, Number 221, January 1998, Pages 343–352
S 0025-5718(98)00935-1

GAUSS PERIODS: ORDERS AND CRYPTOGRAPHICAL APPLICATIONS

SHUHONG GAO, JOACHIM VON ZUR GATHEN, AND DANIEL PANARIO

Abstract. Experimental results on the multiplicative orders of Gauss periods in finite fields are presented. These results indicate that Gauss periods have high order and are often primitive (self-dual) normal elements in finite fields. It is shown that Gauss periods can be exponentiated in quadratic time. An application is an efficient pseudorandom bit generator.

1. Introduction

$\mathbb{F}_q$ denotes a finite field with q elements. Let n and k be positive integers such that $r = nk + 1$ is a prime, not dividing q, and $\mathcal{K}$ the unique subgroup of order k of the multiplicative group of $\mathbb{Z}_r = \mathbb{Z}/r\mathbb{Z}$. For any primitive rth root β of unity in $\mathbb{F}_{q^{nk}}$, the element

$$\alpha = \sum_{a \in \mathcal{K}} \beta^a$$

is a *Gauss period* of type (n, k) over $\mathbb{F}_q$. It is easy to see that $\alpha \in \mathbb{F}_{q^n}$.

Adleman and Lenstra [1] and Mullin et al. [23] used Gauss periods to construct field extensions and normal bases with special properties over finite fields. A *normal basis* for $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ is a basis of the form $\alpha, \alpha^q, \ldots, \alpha^{q^{n-1}}$ generated by some $\alpha \in \mathbb{F}_{q^n}$. Any such α is called a *normal element*.

A Gauss period of type (n, k) over $\mathbb{F}_q$ generates a normal basis for $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ if and only if $\gcd(e, n) = 1$, where e denotes the index of q modulo $r = nk + 1$ (Wassermann [29, 30], Gao et al. [10]). Gao et al. [10] present a method for fast multiplication and division under the normal bases generated by Gauss periods; thus exponentiation in finite fields can be sped up. We refer to that paper and the books by Jungnickel [17] and Menezes et al. [21] for a discussion of the literature.

Gauss periods of type $(n, 2)$ over $\mathbb{F}_2$ also have other remarkable properties. Gao and Vanstone [12] proved that they can be exponentiated in $O(n^2)$ bit operations. This is faster than any known algorithm for exponentiation of an arbitrary element in $\mathbb{F}_{2^n}$ by a factor of $\operatorname{loglog} n$. The orders of Gauss periods of type $(n, 2)$ over $\mathbb{F}_2$ were also computed for $n \leq 1200$. The experimental results in their paper show that Gauss periods have high multiplicative order, and in fact are often primitive

Received by the editor February 16, 1996.
1991 *Mathematics Subject Classification.* Primary 11T30, 94A60; Secondary 11Y16, 12Y05, 68Q25.
Key words and phrases. Finite fields, primitive elements, normal bases, cryptography, pseudorandom bit generators.
This paper is in final form, no version of it will be submitted for publication elsewhere.

elements over $\mathbb{F}_2$. This is useful in cryptosystems where a fixed element needs to be raised to many large powers.

Naturally, one can ask if the above properties hold for Gauss periods of type (n,k) over $\mathbb{F}_2$ with $k>2$. In the next section, we prove that, for any fixed k and q, a Gauss period of type (n,k) over $\mathbb{F}_q$ can indeed be exponentiated in $O(n^2)$ operations in $\mathbb{F}_q$. We computed the multiplicative orders of all Gauss periods of type (n,k) over $\mathbb{F}_2$ for $n\le 1200$ and $3\le k\le 20$ that generate normal bases for $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ as far as the known factorizations of 2^n-1 permit. Our experiments show that Gauss periods of type (n,k) for $k\ge 3$ also have high orders and are often primitive. This means that Gauss periods are often primitive normal elements. When k is even, the normal bases generated by Gauss periods of type (n,k) over $\mathbb{F}_2$ are self-dual. Gauss periods thus are often primitive self-dual normal elements as well. In Section 3, we summarize our experimental results, state some conjectures about primitive normal elements, and show how to construct a primitive element from an element with high order. The experimental data appears in the microfiche supplement at the end of this issue. Finally, we mention in Section 4 some cryptographical applications. In particular, we describe a pseudorandom bit generator based on exponentiation in $\mathbb{F}_{2^n}$, and discuss its security and efficiency.

Our work also contributes to the construction of primitive polynomials and primitive normal polynomials, since their irreducible polynomials are normal and primitive when Gauss periods are primitive. The related literature is mentioned at the end of Section 3.

2. Fast exponentiation of Gauss periods

In this section we show that, for fixed k and q, a Gauss period of type (n,k) can be exponentiated in $O(n^2)$ operations in $\mathbb{F}_q$; see also Gao et al. [10].

A pair (n,k) is a *Gauss pair* over $\mathbb{F}_q$ if $r=nk+1$ is a prime not dividing q and $\gcd(e,n)=1$, where e is the index of q modulo r, i.e., $e=nk/\operatorname{ord}_r(q)$. When q is understood, we simply say that (n,k) is a Gauss pair. It is always assumed that (n,k) is a Gauss pair in the sequel.

In the notation of the introduction, $\mathcal{K}$ is the unique subgroup of order k of $\mathbb{Z}_r^\times$, and

$$\alpha=\sum_{a\in\mathcal{K}}\beta^a,$$

where β is a primitive rth root of unity. Let

$$\mathcal{K}_i=q^i\mathcal{K}=\{aq^i \bmod r : a\in\mathcal{K}\} \text{ for } 0\le i<n.$$

Then $\mathbb{Z}_r^\times$ is the disjoint union of $\mathcal{K}_0,\mathcal{K}_1,\cdots,\mathcal{K}_{n-1}$. We write $\alpha_i=\alpha^{q^i}$ for $0\le i<n$. Then $(\alpha_0,\alpha_1,\cdots,\alpha_{n-1})$ is a normal basis for $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$. It is shown in Gao et al. [10] that

$$\alpha\cdot\alpha_i=\delta_i k+\sum_{0\le j<n}t_{ij}\alpha_j$$

where $t_{ij}=|(1+\mathcal{K}_i)\cap\mathcal{K}_j|$, $\delta_i=1$ if $i=i_0$ and 0 otherwise, and i_0 is such that $-1\in\mathcal{K}_{i_0}$. Note that each $\alpha\cdot\alpha_i$ has at most k nonzero terms, and thus there are at most nk nonzero terms in total. We store all the nonzero t_{ij} in a table, called the multiplication table of Gauss periods of type (n,k).

Theorem 2.1. *Let α be a Gauss period of type (n,k) over $\mathbb{F}_q$ and $0\le e<q^n$. Then α^e can be computed in $O(n^2qk)$ operations in $\mathbb{F}_q$.*

Proof. We want to compute α^e expressed in the normal basis $(\alpha_0, \alpha_1, \cdots, \alpha_{n-1})$. We use a redundant representation of $\gamma \in \mathbb{F}_{q^n}$, writing

$$\gamma = \left(\sum_{0 \le i < n} a_i \alpha_i \right) + a_n,$$

where $a_0, \cdots, a_{n-1}, a_n \in \mathbb{F}_q$. Thus γ is represented by an $(n+1)$-tuple $(a_0, \cdots, a_{n-1}, a_n)$. Of course, this representation is not unique; we just want to compute any one of the representations. For example, the unit 1 can either be represented as $(0, \cdots, 0, 1)$ or $(-1, \cdots, -1, 0)$, since $-1 = \sum_{0 \le i < n} \alpha_i$. Our algorithm benefits from this flexibility.

Thus γ^q is the $(n+1)$-tuple obtained from that of γ by shifting cyclically the first n coordinates to the right by one position (the last coordinate remains fixed). So the cost for computing a qth power is negligible.

For any $\gamma = (\sum_{0 \le i < n} a_i \alpha_i) + a_n \in \mathbb{F}_{q^n}$ and $0 \le j < n$,

$$\alpha_j \gamma = \left(\sum_{0 \le i < n} a_i (\alpha \alpha_{i-j})^{q^j} \right) + a_n \alpha_j.$$

Since $\alpha\alpha_i$ is a sum of at most k terms and can be looked up from the multiplication table, $\alpha_j \gamma$ can be computed in $O(nk)$ operations in $\mathbb{F}_q$.

Now to compute α^e, we use the q-ary representation $e = \sum_{0 \le j \le \ell} e_j q^j$, with $0 \le e_j < q$ for all j and $e_\ell \ne 0$. Then $\ell < n$, and

$$\alpha^e = \prod_{0 \le j \le \ell} \left(\alpha^{q^j} \right)^{e_j} = \prod_{0 \le j \le \ell} \alpha_j^{e_j}.$$

This suggests that we compute α^e iteratively. Initially, set $\gamma := 1$. For j from 0 to ℓ set $\gamma := \alpha_j^{e_j} \gamma$. Then, at the end, we have $\gamma = \alpha^e$. We compute $\alpha_j^{e_j} \gamma$ iteratively as $\alpha_j^i \gamma$ for i from 1 to e_j. This algorithm computes α^e in

$$O\left(\left(\sum_{0 \le j < n} e_j \right) nk \right) = O(\sigma_q(e) nk)$$

operations in $\mathbb{F}_q$, where $\sigma_q(e)$ is the sum of digits of e in q-ary representation. Now $\sigma_q(e) \le (q-1)n < qn$ implies the claim. □

Thus α^e can be computed in $O(n^2)$ operations in $\mathbb{F}_q$, when the values of q and k are fixed. This is faster than any known algorithm for exponentiation of an arbitrary element. Furthermore, one needs only to store e, γ and the multiplication table, a total of $O(nk)$ elements of $\mathbb{F}_q$.

Exponentiation of an arbitrary element in $\mathbb{F}_{q^n}$ (with q bounded) can be performed with $O(n^2 \operatorname{loglog} n)$ operations in $\mathbb{F}_q$ by the currently fastest algorithm, with storage for $O(n/\log_q^2 n)$ elements in $\mathbb{F}_{q^n}$ (Shoup [28], Gao et al. [10]).

Von zur Gathen and Pappalardi [13] proved under ERH that, for any fixed k and q, there are infinitely many values of n such that (n, k) is a Gauss pair over $\mathbb{F}_q$. In fact, they determine a positive density for the primes $nk+1$, where (n, k) is a Gauss pair, in the set of all primes. Thus, there are infinitely many fields $\mathbb{F}_{q^n}$ in which Gauss periods can be exponentiated easily.

3. Experimental results

In this section, we present experimental results which indicate that Gauss periods almost always have high order. When (n, k) is a Gauss pair over $\mathbb{F}_q$, Gauss periods of type (n, k) are algebraic conjugates of each other, hence they have the same multiplicative order. That is, the order of $\alpha = \sum_{a \in \mathcal{K}} \beta^a$ does not depend on the choice of the primitive r-th root β of unity. By the algorithm in the previous section, α can be exponentiated under the normal basis generated by α itself without knowing β, one just needs to precompute the multiplication table for Gauss periods of type (n, k). We computed the multiplicative orders of α for all Gauss pairs (n, k) over $\mathbb{F}_2$ with $3 \leq k \leq 20$ and $2 \leq n < 569$, and also did the corresponding calculations for $569 \leq n \leq 1200$ as far as current knowledge of the factors of $2^n - 1$ permits. The results are tabulated in the table of the microfiche supplement, where "Ind" denotes index, which equals $2^n - 1$ divided by the corresponding multiplicative order. An entry with a question mark "i?" in the "Ind" column means that the corresponding index was computed from the partial factorization of $2^n - 1$ known to the authors at the time of writing. Thus the true index is i times some of the unknown prime factors of $2^n - 1$; we believe that these extra factors are unlikely to occur.

Our experiments show that Gauss periods have the expected multiplicative properties: they almost always have high multiplicative orders and are frequently primitive. More precisely, in the range $2 \leq n < 569$ and $2 \leq k \leq 20$ there are 1267 Gauss pairs (n, k), all the corresponding Gauss periods have order $\geq (2^n - 1)/n$ except for 8 pairs, and 977 of them are primitive. In the range $569 \leq n \leq 1200$ and $2 \leq k \leq 20$, there are 1151 Gauss pairs (n, k), and the corresponding Gauss periods have order $\geq (2^n - 1)/n$ except for 5 pairs, and 894 of them are primitive, provided that the corresponding index entries i? are the true indices.

All the Gauss periods in the table generate normal bases over $\mathbb{F}_2$. When the index is 1, the corresponding basis is a primitive normal basis. Thus Gauss periods yield many primitive normal bases over $\mathbb{F}_2$. Also, when k is even, the normal basis generated by a Gauss period of type (n, k) over $\mathbb{F}_2$ is self-dual (Gao et al. [10]). We see that Gauss periods generate many primitive self-dual normal bases as well.

Gao and Vanstone [12] observe that if n and $2n + 1$ are both primes, then Gauss periods of type $(n, 2)$ are primitive elements in $\mathbb{F}_{2^n}$ for $n \leq 1200$. Our experimental data show that their observation still holds for general Gauss periods. It is formulated as follows.

Conjecture 3.1. If n and $nk+1$ are both primes and $k < \log_2(n+1)$, then Gauss periods of type (n, k) form a primitive normal basis for $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$.

We note that normality is not a problem here. Since the order m of 2 modulo $nk+1$ is at least $\log_2(nk+1) \geq \log_2(n+1) > k$, it follows that n divides $m = nk/e$ and thus $\gcd(e, n) = 1$.

Wassermann [30] proves that for a given n there exists a Gauss pair (n, k) over $\mathbb{F}_2$ if and only if $8 \nmid n$. There are 61 values of $n \leq 1200$ with $8 \nmid n$ for which there is no Gauss pair (n, k) with $k \leq 20$. For each of these n, we list in the table the smallest Gauss pair (n, k) and the corresponding index (or index?).

Our computations lead us to believe that for every n not divisible by 8 there is a Gauss pair (n, k) yielding a primitive Gauss period $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$. As an experiment, we verified this for all $n < 569$. If there is no primitive Gauss period for $k \leq 20$,

then the last entry of k is the smallest k whose Gauss period is primitive and normal in $\mathbb{F}_{2^n}$. The largest such k occurs in (490, 69).

Conjecture 3.2. For any positive integer n not divisible by 8, there exists an integer $k \geq 1$ such that the Gauss period of type (n, k) is primitive normal in $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$.

Motivated by this work and the previous work of Gao and Vanstone [12], von zur Gathen and Shparlinski [14] prove that Gauss periods of type $(n, 2)$ have order at least $2^{\sqrt{2n}-2}$.

Next, we show how to construct primitive elements from elements of high order. The constructed primitive elements will still be essentially as easy to exponentiate as Gauss periods.

Theorem 3.3. *Let $\alpha \in \mathbb{F}_{2^n}$ with index e. A primitive element can be constructed from α deterministically in time polynomial in e and n.*

Proof. The order of α is $m = (2^n - 1)/e$. Write $e = e_1 e_2$, where $\gcd(e_2, m) = 1$ and every prime divisor of e_1 divides m. Let $\beta_1 \in \mathbb{F}_{2^n}$ satisfy

$$\beta_1^{e_1} = \alpha,$$

and let β_2 be a primitive e_2th root of unity in $\mathbb{F}_{2^n}$. Then β_1 has order me_1, and $\beta_1\beta_2$ has order $me_1e_2 = 2^n - 1$, as $\gcd(me_1, e_2) = 1$. This means that $\beta_1\beta_2$ is primitive in $\mathbb{F}_{2^n}$. Also, β_1 and β_2 can be constructed in time polynomial in e and n. □

Thus if the order of α is at least $(2^n - 1)/n^c$ for a constant c, then a primitive element in $\mathbb{F}_{2^n}$ can be constructed in time polynomial $n^{O(1)}$. In the special case $e = 2^k - 1$, the equation $x^e - \alpha$ can be written as $x^{2^k} = \alpha x$, which corresponds to a system of linear equations over $\mathbb{F}_2$, and can be solved by any efficient algorithm for linear equations. Our experimental data shows that many Gauss periods have indices $3, 7, 15$, etc, which are of the form $2^k - 1$. Thus from our table of Gauss periods, it is easy to construct primitive elements if one really needs primitive elements instead of elements of high orders. In the table, we give for each $n < 569$ and $8 \nmid n$, the smallest k such that a Gauss period of type (n, k) has index at most n. One can see that for these n it is possible to find a reasonably small such k.

Finally, we make some comments on the related work in the literature. As mentioned in the introduction, our work also contributes to the construction of primitive polynomials and primitive normal polynomials, since their irreducible polynomials are normal and primitive when Gauss periods are primitive. Hansen and Mullen [16] and Morgan and Mullen [22] give tables of primitive polynomials and primitive normal polynomials of degree m over $\mathbb{F}_p$ for all prime powers $p^m \leq 10^{50}$ with $p \leq 97$. Živković [35, 36] gives a more extensive table of primitive polynomials of degree $m \leq 1200$ (and a few values of m between 1200 and 5000) over $\mathbb{F}_2$ when the factorization of $2^m - 1$ is known. In their work, they search for sparse polynomials, i.e., those with the smallest number of nonzero terms. Such polynomials are useful in efficient implementation of feedback shift registers. Gao and Panario [11] provide a construction of infinite families of sparse irreducible polynomials. In the extreme case, there is much interest in constructing irreducible trinomials over $\mathbb{F}_2$. Zierler and Brillhart [33, 34] give a table of irreducible trinomials of degree ≤ 1000. Blake et al. [5] extend this list to all irreducible trinomials of

degree ≤ 2000 over $\mathbb{F}_2$ (and a table for degree ≤ 5000 is available from those authors).

It is, however, not clear how primitive elements from sparse polynomials can be exponentiated faster than an arbitrary primitive element. The primitive polynomials from Gauss periods may not in general be sparse, but they do provide computational advantage in fast exponentiation as shown by Theorem 2.1.

4. Cryptographical applications

Let $\alpha \in \mathbb{F}_{q^n}$ be a primitive element (or an element of high order, say at least $(q^n - 1)/n^c$ for some constant c). Computing the exponentiation function that maps $x \in \{0, \ldots, q^n - 1\}$ to α^x is easy, but computing its inverse function, i.e., computing x given α^x, called the discrete logarithm problem, is believed to be hard in general. This one-wayness of exponentiation has found many applications in public-key cryptography: Diffie-Hellman key exchange (Diffie and Hellman [8]), password schemes (Lamport [18]), ElGamal cryptosystem (ElGamal [9]), cryptosystems over $\mathbb{F}_2$ (Agnew et al. [2], Agnew et al. [3]), smart cards (Beth [4], Schnorr [26, 27]), US Digital Signature Algorithm (NIST [24]), pseudorandom bit generators (Blum and Micali [6], Long and Widgerson [20]). Pseudorandom bit generators based on discrete logarithms in finite fields are used by Zheng and Seberry [31, 32] and Lim and Lee [19] to construct cryptosystems that leak no partial information and are secure against adaptively chosen ciphertext attacks. In these applications, one needs a fixed element of high order and computes α^t for many random large integers t. For example, in a signature scheme, for each signature one needs to generate a random integer t and compute α^t. Sometimes the computing power of the signature generating device is limited, e.g., in a smart card. So α has to be chosen such that exponentiation of α is easy. In practice, the currently popular choice for α is from elements in $\mathbb{F}_p$. If the prime p has n bits, then computing α^t needs $O(n^3)$ bit operations using repeated square and multiply method, or $O(n^2 \log n \operatorname{loglog} n)$ bit operations using FFT-based fast multiplication algorithms. However, if we choose α to be a Gauss period of type (n, k) in $\mathbb{F}_{2^n}$ for a small k, then the cost of exponentiating α is reduced to $O(n^2)$ bit operations, which is just the cost of one multiplication by the "classical" method. Our experimental results show that α almost always has high order and is often primitive. Our exponentiation algorithm is also easy to implement. Gauss periods are therefore highly attractive in these applications.

In the following, we describe Blum and Micali's pseudorandom bit generator based on exponentiation in $\mathbb{F}_p$, then we adapt it to the fields $\mathbb{F}_{2^n}$ and comment on its security and efficiency.

A *pseudorandom bit generator* produces sequences of bits (0 or 1) that cannot be distinguished from truly random sequences of bits of equal length by any (probabilistic) polynomial time algorithm. Blum and Micali [6] presented the following pseudorandom bit generator. Let $m \geq 1$ be a fixed integer. Given $n \geq 2$, select an odd prime p of n bits, and a primitive root α modulo p. Pick a random integer a_0 (the seed) in the range $1 < a_0 < p - 1$. Set

$$a_{k+1} \equiv \alpha^{a_k} \bmod p \text{ for } k \geq 0,$$

and

$$b_{k+1} = 1 \text{ if } a_{k+1} > (p-1)/2, \text{ and } b_{k+1} = 0 \text{ otherwise.}$$

Then $\{b_k : 1 \le k \le n^m + m\}$ is a sequence of $n^m + m$ bits generated from the n-bit seed a_0. Blum and Micali [6] proved that if the discrete logarithm problem in $\mathbb{F}_p$ is hard, then this much longer sequence of bits is pseudorandom. Blum and Micali's generator outputs only one bit at each iteration, i.e., each bit costs one exponentiation mod p. Long and Wigderson [20] extended this to output about $\log n$ bits at each iteration.

We now adapt the above generator to the fields $\mathbb{F}_{2^n}$. Under a fixed basis for $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$, an element $x \in \mathbb{F}_{2^n}$ is represented as a sequence of n bits (0 and 1). We use $\overline{x}$ to denote the integer whose binary representation is the same as x. Let $\alpha \in \mathbb{F}_{2^n}$ be a primitive element (or an element of high order). Pick a random element $x_0 \in \mathbb{F}_{2^n}$. Set

$$x_{k+1} = \alpha^{\overline{x}_k} \text{ for } k \ge 0,$$

and let z_{k+1} be the least significant bit

$$z_{k+1} = \overline{x}_{k+1} \bmod 2. \tag{1}$$

Then $\{z_k : 1 \le k \le n^m + m\}$ is a sequence of bits generated from the seed x_0. We want to show that this sequence is pseudorandom.

Let f be a one-way function, i.e, it is easy to compute but hard to invert. A Boolean predicate B (i.e., $B(x) = 0$ or 1) is said to be *hard* for f if an oracle for $B(f(x))$ allows one to invert f easily. Blum and Micali [6, Theorem 2] proved that if B is a hard predicate for a one-way function f, then the following sequence is pseudorandom:

$$B(f(a_0)), B(f^2(a_0)), B(f^3(a_0)), \cdots, B(f^k(a_0)), \cdots$$

where a_0 is randomly chosen. To show that our sequence is pseudorandom, it is enough to prove Theorem 4.1 below. For any $\beta \in \mathbb{F}_{2^n}$, the smallest positive integer x such that $\beta = \alpha^x$ is called the discrete logarithm of β with respect to α, denoted by $\log_\alpha \beta$. If no such x exists, we set (arbitrarily) $\log_\alpha \beta = 0$.

Theorem 4.1. *Every bit of the discrete logarithm in $\mathbb{F}_{2^n}$ is a hard predicate (for the exponentiation function).*

Proof. Let $B(x)$ be the least significant bit of an integer x. We show how to compute discrete logarithms via an oracle for $B(\log_\alpha \beta)$, which returns the least significant bit of the discrete logarithm of any $\beta \in \mathbb{F}_{2^n}$. Note that for any integer i, $\log_\alpha \beta^{2^i} \equiv 2^i \log_\alpha \beta \bmod 2^n - 1$. Suppose that

$$\log_\alpha \beta = \sum_{k=0}^{n-1} a_k 2^k = (a_0, a_1, \cdots, a_{n-1})_2.$$

Then for $i \in \mathbb{N}$

$$2^i \log_\alpha \beta \equiv \sum_{k=0}^{n-1} a_k 2^{k+i} \equiv \sum_{k=0}^{n-1} a_{k-i} 2^k \bmod 2^n - 1,$$

where the subscripts of a are computed modulo n, and

$$\log_\alpha \beta^{2^i} = (a_{n-i}, \cdots, a_{n-1}, a_0, \cdots, a_{n-i-1})_2.$$

Therefore

$$B(\log_\alpha \beta^{2^i}) = a_{n-i} \text{ for } 0 \le i \le n-1,$$

and

$$\log_\alpha \beta = (B(\log_\alpha \beta), B(\log_\alpha \beta^{2^{n-1}}), \cdots, B(\log_\alpha \beta^2))_2.$$

This means that $\log_\alpha \beta$ can be computed by n calls to the oracle. Since the bits can be shifted cyclically, $\log_\alpha \beta$ can also be computed by n calls to an oracle for any bit. Therefore every bit is a hard predicate. □

Corollary 4.2. *If the discrete logarithm problem in $\mathbb{F}_{2^n}$ is hard, then the bit sequence $z_1, z_2, \ldots$ defined in* (1) *is pseudorandom.*

We have assumed that the oracle is perfect, that is, it always gives correct answers. Blum and Micali [6] dealt with the more general case of nonperfect oracles (an oracle that may give wrong answers, but it gives correct answers sufficiently more frequently than incorrect ones). It seems likely that Theorem 4.1 still holds for nonperfect oracles.

Our theorem says that every bit of the discrete logarithms is a hard predicate, that is, every bit is individually secure, provided the discrete logarithm is hard to compute. This is different from the discrete logarithms modulo an odd prime p, where its least significant bit is not secure while the most significant bit is indeed secure (Peralta [25]). It seems also possible to modify the proof in Long and Wigderson [20] to show that any $O(\log n)$ consecutive bits of the discrete logarithms in $\mathbb{F}_{2^n}$ are simultaneously secure. In this case, the modified pseudorandom bit generator could output $O(\log n)$ bits per iteration. The discrete logarithm problem in $\mathbb{F}_{2^n}$ seems easier than that in $\mathbb{F}_p$, and one may need to pick a bigger field $\mathbb{F}_{2^n}$ than for $\mathbb{F}_p$ to maintain the same level of security.

In implementing our pseudorandom bit generator in $\mathbb{F}_{2^n}$, one can choose α to be a Gauss period of type (n, k) for a small k. The cost of exponentiating α at each iteration is only $O(n^2)$ bit operations. This is advantageous compared to the generator in $\mathbb{F}_p$, where exponentiation needs $O(n^2 \log n \operatorname{loglog} n)$ bit operations by using fast multiplication algorithms.

Acknowledgement

We thank S.S. Wagstaff for making the updated factorization tables of the Cunningham project (Brillhart et al. [7]) available. We used Maple in our computations. Part of this work was done at the University of Toronto, where the first author was supported by an NSERC Postdoctoral Fellowship and the others by the second author's NSERC operating grant.

References

[1] L.M. Adleman and H.W. Lenstra, Jr., *Finding irreducible polynomials over finite fields*, Proc. 18th Annual ACM Symp. on Theory of Computing (1986), 350–355.

[2] G.B. Agnew, R.C. Mullin, I.M. Onyszchuk and S.A. Vanstone, *An implementation for a fast public key cryptosystem*, J. of Cryptology, **3** (1991), 63–79. MR **92b:**94034

[3] G.B. Agnew, R.C. Mullin and S.A. Vanstone, *An implementation of elliptic curve cryptosystems over $F_{2^{155}}$*, IEEE J. on Selected Areas in Communications, **11** (1993), 804–813.

[4] T. Beth, *Efficient zero-knowledge identification scheme for smart cards*, Advances in Cryptology (Proc. Eurocrypt '88), Springer LNCS **330** (1988), 77–84. CMP 21:11

[5] I.F. Blake, S. Gao and R. Lambert, *Constructive problems for irreducible polynomials over finite fields*, in Information Theory and Applications (A. Gulliver and N. Secord, eds), Springer LNCS **793** (1994), 1–23. MR **95c:**94007

[6] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudorandom bits*, SIAM J. Computing, **13** (1984), 850–864. MR **86a:**68021

[7] J. Brillhart, D.H. Lehmer, J.L. Selfridge, B. Tuckerman and S.S. Wagstaff, Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ Up to High Powers, Contemporary Mathematics, **22**, 2nd ed., AMS, 1988. MR **90d:**11009

[8] W. Diffie and M.E. Hellman, *New directions in cryptography*, IEEE Trans. Info. Th., **22** (1976), 644–654. MR **55:**10141

[9] T. ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Trans. Info. Th., **31** (1985), 469–472. MR **86j:**94045

[10] S. Gao, J. von zur Gathen and D. Panario, *Gauss periods, primitive normal bases, and fast exponentiation in finite fields*, preliminary version in Proc. Latin'95, Valparaíso, Chile, Springer LNCS **911** (1995), 311–322; full version in Technical Report 296/95, Department of Computer Science, University of Toronto, 1995.

[11] S. Gao and D. Panario, *Tests and constructions of irreducible polynomials over finite fields*, in Foundations of Computational Mathematics, F. Cucker and M. Shub (Eds.), Springer Verlag, 1997, 346–361.

[12] S. Gao and S.A. Vanstone, *On orders of optimal normal basis generators*, Math. Comp., **64** (1995), 1227–1233. MR **95j:**11117

[13] J. von zur Gathen and F. Pappalardi, *Density estimates related to Gauss periods*, 1996, preprint.

[14] J. von zur Gathen and I.E. Shparlinski, *Orders of Gauss periods in finite fields*, in Proc ISAAC '95, Springer LNCS **1004** (1995), 208–215. MR **97b:**11153

[15] C.F. Gauss, Disquisitiones Arithmeticae, Braunschweig, 1801. English Edition, Springer-Verlag, 1986. MR **87f:**01105

[16] T. Hansen and G.L. Mullen, *Primitive polynomials over finite fields*, Math. Comp., **59** (1992), 639–643. MR **93a:**11101

[17] D. Jungnickel, Finite Fields: Structure and Arithmetics, Bibliographisches Institut, Mannheim, 1993. MR **94g:**11109

[18] L. Lamport, *Password authentication with insecure communication*, Comm. ACM, **24** (1981), 770–772.

[19] C.H. Lim and P.J. Lee, *Another method for attaining security against adaptively chosen ciphertext attacks*, Advances in Cryptology (Proc. Crypto '93), Springer LNCS **773** (1994), 420–434.

[20] D.L. Long and A. Wigderson, *The discrete logarithm hides $O(\log n)$ bits*, SIAM J. Computing, **17** (1988), 363–372. MR **89e:**68026

[21] A.J. Menezes, I.F. Blake, X.H. Gao, R.C. Mullin, S.A. Vanstone, and T. Yaghoobian, Applications of Finite Fields, Kluwer Academic Publishers, 1993.

[22] I.H. Morgan and G.L. Mullen, *Primitive normal polynomials over finite fields*, Math. Comp., **63** (1994), 759–765. MR **95a:**11106

[23] R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone and R.M. Wilson, *Optimal normal bases in $GF(p^n)$*, Discrete Applied Math., **22** (1988/1989), 149–161. MR **90c:**11092

[24] National Institute for Standard and Technology, Federal Information Processing Standard for Digital Signature Standard (DSS), FIPS 186, May 1994.

[25] R. Peralta, *Simultaneous security of bits in the discrete log*, Advances in Cryptology (Proc. Eurocrypt '85), Springer LNCS **219** (1986), 62–72. CMP 18:16

[26] C.P. Schnorr, *Efficient identification and signatures for smart cards*, Advances in Cryptology (Proc. Crypto '89), Springer LNCS **435** (1990), 239–252. MR **92e:**94024

[27] C.P. Schnorr, *Efficient signature generation by smart cards*, J. of Cryptology, **4** (1991), 161–174.

[28] V. Shoup, *Exponentiation in $GF(2^n)$ using fewer polynomial multiplications*, preprint, 1994.

[29] A. Wassermann, *Konstruktion von Normalbasen*, Bayreuther Mathematische Schriften, **31** (1990), 155–164. MR **91h:**11145

[30] A. Wassermann, *Zur Arithmetik in endlichen Körpern*, Bayreuther Mathematische Schriften, **44** (1993), 147–251. MR **94g:**11114

[31] Y. Zheng and J. Seberry, *Practical approaches for attaining security against adaptively chosen ciphertext attacks*, Advances in Cryptology (Proc. Crypto '92), Springer LNCS **740** (1993), 292–304.

[32] Y. Zheng and J. Seberry, *Immunizing public key cryptosystems against chosen ciphertext attacks*, IEEE J. on Selected Areas in Communications, **11** (1993), 715–724.

[33] N. Zierler and J. Brillhart, *On primitive trinomials (mod 2)*, Information and Control, **13** (1968), 541–554. MR **38:**5750

[34] N. Zierler and J. Brillhart, *On primitive trinomials (mod 2), II*, Information and Control, **14** (1969), 566–569. MR **39:**5521

[35] M. Živković, *A table of primitive binary polynomials*, Math. Comp., **62** (1994a), 385–386. MR **94d:**11099
[36] M. Živković, *Table of primitive binary polynomials, II*, Math. Comp., **63** (1994b), 301–306. MR **94i:**11099

Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-1907, USA
E-mail address: sgao@math.clemson.edu

Fachbereich Mathematik-Informatik, Universität-GH Paderborn, D-33095 Paderborn, Germany
E-mail address: gathen@uni-paderborn.de

Department of Computer Science, University of Toronto, Toronto, Ontario M5S 1A4, Canada
E-mail address: daniel@cs.toronto.edu

TABLE 1. Appendix: Index of Gauss periods

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
2	2	1		10	1		18	1	46	3	3		8	1
	5	1		12	1	30	2	1		6	1		18	1
	6	1	14	2	1		6	1		10	3		20	1
	9	1		3	3		7	1		15	3	66	10	1
	14	1		14	1		11	1		18	1		13	1
	18	1		15	1		14	3	47	6	1		17	1
3	2	1	15	4	1		18	1		14	1	67	4	1
	4	1		12	1	31	10	1		20	1	68	9	1
	6	1		14	1		12	1	49	4	1		15	5
	12	1		18	31	33	2	1		10	1	69	2	1
	20	1	17	6	1		6	7		18	1		12	1
4	3	1		8	1		14	1	50	2	3		14	1
	7	1		14	1		20	161		14	1		20	1
	9	3		18	1	34	9	1	51	2	1	70	3	3
	13	3	18	2	3		13	1		8	1		6	1
	15	1		9	1		18	3		12	1		7	1
5	2	1		10	3	35	2	1	52	3	1		10	11
	6	1	19	10	1		6	1		13	5		15	3
	8	1		12	1		8	1	53	2	1	71	8	1
	12	1	20	3	1		12	1		14	1		12	1
	14	1		5	1		14	1		20	1		18	1
	20	1		9	5		20	1	54	3	9	73	4	1
6	2	1	21	10	1	36	5	5		7	1		6	1
	3	1		16	1		15	1		10	3		12	1
	6	3		18	1		17	1		14	19	74	2	1
	10	3		20	1	37	4	1	55	12	1		17	1
	11	1	22	3	1		6	1		16	1	75	10	1
7	4	1		18	1		16	1		18	1		16	1
	6	1		19	1	38	6	3	57	10	1	76	3	1
	10	1	23	2	1		11	1	58	6	3	77	6	1
	16	1		6	1		15	1		9	3		8	1
	18	1		12	1	39	2	1		22	1	78	7	9
9	2	1		20	1		8	1	59	12	1		11	7
	4	1	25	4	1		14	1		14	1		15	1
	8	7		16	1	41	2	1		18	1		19	1
	18	1	26	2	1		18	1		20	1	79	4	1
	20	1		5	1		20	1	60	3	1		18	1
10	6	1		6	1	42	5	387		7	1	81	2	1
	10	1		17	3		9	9		9	3		6	1
	13	1	27	6	1		10	3		11	3		16	1
	18	1		10	1		13	1	61	6	1		20	1
11	2	1		14	7		18	1		12	1	82	9	3
	6	1		18	1	43	4	1		16	1		10	1
	8	1		20	1		10	1	62	6	3	83	2	1
	18	1	28	7	3	44	9	5		18	1		6	1
12	3	1		15	3		15	1	63	6	1		12	1
	5	35		25	1	45	4	1		12	1		14	1
	15	3	29	2	1		6	1		14	1	84	5	1
13	4	1		8	1		12	1		16	1		9	39
	6	1		12	1		14	1	65	2	1		17	35

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
85	12	1		18	7		20	1	140	3	1	163	4	1
	16	1		24	1	121	6	1		5	3		16	1
	18	1	100	7	3		12	23	141	8	7	164	5	1
86	2	1		13	1		18	1		12	7		17	1
	11	3		19	375	122	6	3		18	1	165	4	1
	15	1	101	6	1		14	1	142	6	1		6	1
	18	3		8	1	123	10	1		15	1		14	1
87	4	1		12	1	124	3	1		18	1	166	3	1
	6	1	102	6	1		9	3		19	1		6	3
	18	1		11	21		15	3	143	6	1		7	1
	20	1		15	7		19	1		14	1	167	14	1
89	2	1	103	6	1	125	6	1		20	1	169	4	1
	12	1		10	1		18	1	145	10	1		12	1
90	2	1		12	1	126	3	1		12	1	170	6	1
	6	7	105	2	1		6	3	146	2	1		9	1
	13	1		4	1		7	21		6	3	171	12	1
	17	1	106	10	1		18	73	147	6	1		18	1
	18	1	107	6	1	127	4	1		18	1	172	9	1
91	6	1		8	1		18	1	148	15	3		13	1
	10	1		14	1	129	8	7		25	1	173	2	1
	12	1		20	1		10	1	149	8	1		6	1
92	3	3	108	5	1		12	1		12	1		14	1
	5	1		7	13	130	9	3		14	1		20	1
	9	1		15	9		10	1		18	1	174	2	3
	11	3		19	1		18	1	150	19	1		3	9
	15	3	109	10	1	131	2	1	151	6	1		10	1
93	4	1	110	6	1		6	1		10	1		14	1
	12	1	111	20	1		8	1		16	1		19	1
	14	1	113	2	1		20	1		18	1	175	4	1
	16	1		14	1	132	5	23	153	4	1		6	31
	20	7	114	5	1		11	1		10	1		16	1
94	3	1		13	7	133	12	1	154	25	3	177	4	1
	7	1		18	1		16	1		42	1		6	1
	10	1	115	4	1	134	2	3	155	2	1		18	1
	18	1		6	1		14	1		12	1	178	6	1
	19	1		10	1		15	3		18	1	179	2	1
95	2	1		12	1	135	2	1	156	13	5		8	1
	8	31	116	3	1		4	7		75	1		20	1
	20	1		11	1		12	1	157	10	1	180	3	1
97	4	1		15	1		16	1	158	2	1		9	11
	10	1		17	1	137	6	1		14	3	181	6	1
	16	1	117	8	1		8	1		15	1		10	1
	18	1		10	1		18	1	159	22	1		16	1
98	2	3		16	1		20	1	161	6	1		18	1
	5	3	118	6	3	138	6	3		8	1	182	3	3
	9	1		7	1		10	3		12	1		6	1
	14	3		10	1		22	1		20	1		14	3
	17	1		19	3	139	4	1	162	10	19	183	2	1
99	2	7	119	2	1		12	1		14	7	185	8	1
	10	7		14	1		18	1		30	1		12	1

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
	14	223		18	1	230	2	1	251	2	1		10	1
	20	1	206	3	1		3	1		18	1		12	1
186	2	3		6	1		6	1		20	1		16	1
	6	1		11	3		11	1	252	3	15	273	2	1
	10	3		18	3	231	2	1		9	27	274	9	1
187	6	1	207	4	1		10	1		19	21		10	3
	10	1		16	1		16	1		43	1		18	1
188	5	5		18	1		18	1	253	10	1	275	14	1
	9	3	209	2	1		20	1		12	1		18	1
	11	1		14	1	233	2	1		16	23		20	1
189	2	1	210	2	1		6	1	254	2	1	276	3	3
	4	1		17	3		12	1		14	3		5	7
	12	1	211	10	1	234	5	1	255	6	1		11	45
	14	7	212	5	15	235	4	1		10	7		41	1
190	10	93		11	1		18	1		14	7	277	4	1
	15	1		15	1	236	3	5		18	1		6	1
191	2	1	213	4	1		5	1	257	6	1		18	1
	12	1		6	1		15	1	258	5	3	278	2	3
	20	1		10	1		17	3		6	1		6	1
193	4	1		12	1	237	10	1		14	1	279	4	1
	10	1		20	1		14	1	259	10	223		14	1
	16	1	214	3	3		16	1		18	1		18	1
194	2	3		7	3	238	7	3	260	5	1	281	2	1
	5	1		10	3		10	1		9	1		12	1
	9	1		18	3		15	3		11	5		18	1
	17	1		27	1	239	2	1	261	2	1	282	6	1
195	6	1	215	6	1		8	1		6	7		9	1
	10	1		8	1		14	1	262	3	1		18	7
	18	1		14	1	241	6	1		10	1	283	6	1
196	7	15	217	6	1		10	1		15	1	284	3	3
	13	1		18	1		18	1	263	6	1		9	1
197	18	1	218	5	3	242	6	1		20	1		15	1
198	22	67		29	1		9	3	265	4	1	285	10	1
	34	1	219	4	1		14	3		12	1		16	7
199	4	1		14	1		18	1		16	1		20	1
	12	1	220	3	1	243	2	1	266	6	43	286	3	1
	18	1		13	1		20	1		17	1		7	3
201	8	1	221	2	1	244	3	5		18	1		10	1
	10	1		6	1		7	15	267	8	1	287	6	1
	16	1	222	10	3		19	1		14	1		8	1
	20	1		14	1	245	2	1		16	7		14	1
202	6	3		19	3		6	1	268	7	1		18	1
	18	1	223	12	1	246	11	1		15	5		20	1
203	12	1	225	22	1		15	1	269	8	1	289	12	1
	14	1	226	13	1	247	6	1		12	1	290	5	3
204	3	1	227	24	1		18	1		14	1		6	1
	17	1	228	9	9	249	8	1		20	1	291	6	1
	19	195		17	3		18	1	270	2	7	292	3	5
205	4	1		21	1	250	9	3		6	1		13	5
	6	1	229	12	1		22	1	271	6	1		21	1

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
293	2	1		16	1	339	8	7	364	3	1	390	3	1
	6	1	314	5	3		20	7		7	43		15	9
	12	1		14	1		22	1		19	3		19	1
	20	1		18	1	340	3	1	365	24	1	391	6	1
294	3	3	315	8	7		13	1	366	22	9		10	1
	22	1		40	1		15	1		43	1		16	1
295	16	1	316	7	1	341	8	23	367	6	1		18	1
297	6	1	317	26	1		12	1		10	1	393	2	7
	8	7	318	11	1	342	6	7		18	1		10	1
	14	1		19	1		14	63	369	10	1	394	9	3
298	6	1	319	4	1		83	1	370	6	3		22	1
	9	1		18	1	343	4	1		10	1	395	6	1
299	2	1	321	12	1	345	4	151		18	1		14	1
	8	1		18	1		14	1	371	2	1		20	1
300	19	17173	322	6	1		16	1		8	1	396	11	135
	33	1		10	3		18	1	372	3	3		17	7
301	10	1	323	2	1	346	10	1		5	9		25	1
	12	1		12	1		18	1		11	1	397	6	1
	16	1	324	5	1	347	6	1		19	1		16	1
302	3	1		7	3		8	1	373	4	1	398	2	1
	11	1		15	5		18	1		6	1		6	1
	14	1	325	4	1	348	5	1	374	3	1		14	1
	18	1		6	1		7	7		14	1	399	12	1
303	2	1		18	1	349	10	1		18	1	401	8	1
	4	1	326	2	1	350	2	3	375	2	1		12	1
	12	1		18	1		3	3		8	1		18	1
	14	1	327	8	1		11	31	377	14	1	402	5	1
305	6	31		16	1		18	3		20	1		10	9
	8	1	329	2	1		35	1	378	2	3		13	1
	14	1		8	1	351	10	1		6	3		18	1
	20	1		18	1	353	14	1		46	1	403	16	1
306	2	1		20	1	354	2	3	379	12	1	404	3	3
	5	19	330	2	1		9	1		18	1		9	1
307	4	1		14	3		13	9	380	5	3		17	5
	18	1	331	6	1		14	9		15	1	405	4	1
308	15	5		16	1		18	3	381	8	1		16	31
	35	1	332	3	5	355	6	1		20	7	406	6	1
309	2	1		33	1		12	1	382	6	1		7	43
	4	1	333	24	7	356	3	1		10	1		15	1
	8	1		32	1		11	5	383	12	1	407	8	1
	12	1	334	7	1		17	1	385	6	1	409	4	1
	14	1		15	1	357	10	1		10	1		10	1
	18	7	335	12	1	358	10	3		12	71		12	1
310	6	3		14	1		19	1	386	2	1	410	2	11
	15	33		20	1	359	2	1		5	3		9	1
	43	1	337	10	1	361	30	1		17	3		14	1
311	6	1		16	1	362	5	1		18	1		17	1
	12	1		18	1		9	1	387	4	1	411	2	1
	20	1	338	2	3	363	4	1	388	15	1		6	1
313	6	1		6	1		12	7	389	24	1		10	1

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
	12	1		18	33	459	8	1		7	1		15	1
	16	1		19	3		10	1		9	1	509	2	1
	20	1	431	2	1		18	1	485	18	1		8	1
412	3	3		8	1		20	1	486	10	5103		14	1
	9	1		18	1	460	3	75		26	1		20	1
	19	1	433	4	1		13	5	487	4	1	510	3	3
413	2	1		12	1		19	5		10	1		7	1
	12	1	434	9	1		21	1		16	1		18	3
	14	1	435	4	1	461	6	1	489	12	7	511	6	1
414	2	3	436	13	1		20	1		18	1		12	1
	15	3	437	18	1	462	10	3	490	18	3		18	1
	42	1		20	47		14	1		69	1	513	4	7
415	28	1	438	2	3	463	12	1	491	2	1		6	1
417	4	1		7	9	465	4	1		8	1		16	1
	6	1		15	1	466	6	3		18	1	514	33	1
	14	7	439	10	1		18	1	492	13	5	515	2	1
	18	1	441	2	1	467	6	1		19	1		14	1
418	21	3		6	1		20	1	493	4	1		20	1
	30	1		8	1	468	21	1		10	1	516	3	1
419	2	1		20	1	469	4	1	494	3	1		7	3
	14	1	442	21	3		10	1		14	3		13	9
420	11	143		45	1		18	1		15	1	517	4	1
	25	1	443	2	1	470	2	1		18	1		10	1
421	10	1		6	1		11	3	495	2	1		16	1
	16	1		14	1	471	8	1		10	1	518	14	3
422	11	1		20	1		20	1		20	1		26	1
423	4	1	444	5	1	473	2	1	497	20	1	519	2	1
	6	1		7	1		20	1	498	9	7		12	1
	10	1		17	3	474	5	1		17	9		18	7
	12	1	445	6	1		14	7		21	1	521	32	1
	14	1		10	1	475	4	1	499	4	1	522	14	1
	20	7		16	1		6	1	500	11	3		18	7
425	6	1		18	1		12	1		39	1	523	10	1
	12	1	446	6	3	476	5	3	501	10	1		12	1
426	2	1		15	1		17	3		16	1		16	1
	5	1	447	6	1		35	1	502	10	3	524	5	5
	6	3	449	8	1	477	46	1		19	1		27	1
	10	1		14	1	478	7	1	503	6	1	525	8	1
	17	3	450	13	1	479	8	1		12	1		20	7
	18	9	451	6	1		12	1		14	1	526	3	1
427	16	1		12	1		18	1		20	1		10	3
	18	1	452	11	1	481	6	1	505	10	1	527	6	1
428	5	15		15	3	482	5	1		16	1		8	1
	9	1	453	2	1		9	1	506	5	3	529	24	1
	15	1		6	7		18	3		6	1	530	2	1
429	2	1		14	1	483	2	1		18	1		6	3
	8	23	454	19	1		10	1	507	4	7		17	1
430	3	1	455	26	1		14	1		18	1	531	2	1
	7	431	457	30	1		20	1		20	1		6	1
	15	1	458	6	1	484	3	1	508	7	1		12	1

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
532	3	3		9	1	580	3	1		9	1	631	10	1?
	9	5		14	1		13	3		13	3		12	1?
	13	1		17	3	581	8	1?	605	6	1	633	34	1
533	12	1		18	1		18	1?		18	1	634	13	3
534	7	1	555	4	1		20	1?	606	2	9	635	8	1?
	14	1		8	1	582	3	1		6	1		12	1?
	18	9	556	3	3		15	9		7	1	636	13	3
535	4	1		13	1		18	3		15	3	637	4	1
	10	1	557	6	1	583	4	1		18	1		6	1
	12	1		8	1		6	1	607	6	1		16	1
	18	1	558	2	1		12	1	609	4	1	638	2	1
537	8	1		7	73	585	2	1		14	1		11	1
538	6	3	559	4	1		6	1	610	10	1		14	3
	10	1		10	1		10	1	611	2	1?	639	2	1
539	12	1		12	1	586	6	1		8	1?		4	1
	14	1	561	2	1		10	1		12	1?		12	7
	20	1		12	1	587	14	1	612	11	1		18	1
540	3	1		18	1		18	1		15	7		20	7
	17	27	562	6	1	588	11	147	613	10	1?	641	2	1?
541	18	1		9	3		15	1	614	2	3		6	1?
542	3	3		13	1	589	4	1?		14	3		20	1?
	6	1	563	14	1		12	1?	615	2	1	642	6	3
	14	1		20	1	590	11	1		6	1		9	1
543	2	1	564	3	1	591	6	1		18	1	643	12	1?
	10	1		9	1		18	1		20	7		16	1?
	14	1		15	13		20	1	617	8	1?	644	3	1
	16	1	565	10	1	593	2	1?	618	2	1		5	1
	20	1		12	1		6	1?		6	3		15	1
545	2	1	566	3	1	594	17	1		9	1		17	5
	6	1	567	4	1	595	6	1	619	4	1?	645	2	7
	18	31	569	12	1?	596	3	1	620	3	3		16	1
546	17	1		18	1?		15	1	621	6	1	646	6	3
547	10	1	570	5	1		17	3		8	1	647	14	1
	16	1		10	1	597	4	1		10	1		20	1
548	5	1		13	1		6	1	622	3	1	649	10	1
	9	1	571	10	1?		20	1		6	1		12	1
	11	1		16	1?	598	15	3		10	1	650	2	3
	15	1	572	5	1	599	8	1?		18	3		9	1
549	14	7	573	4	1		14	1?	623	12	1?	651	2	1
	18	1	574	3	1		20	1?	625	36	1		6	1
550	7	3		7	3	601	6	1?	626	21	1		18	1
	10	1		10	1		10	1?	627	20	1	652	9	3
	18	121		18	1		12	1?	628	7	3		15	3
551	6	1	575	2	1	602	5	1		9	1	653	2	1
	8	1	577	4	1		6	1		15	3		6	1
553	4	127		6	1		14	1		19	1	654	14	7
	6	1	578	6	1		18	1293	629	2	1?	655	4	31?
	12	1		14	1	603	12	1		12	1?		6	1?
	16	1	579	10	1		14	7		14	1?		10	1?
554	2	1		12	1	604	7	1	630	14	3	657	10	1

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
	14	7	683	2	1?		17	1365		12	1		12	1?
	16	1		6	1?	709	4	1	737	6	1?	762	10	3
	18	1	684	3	3		18	1		8	1?		13	1
658	9	1		7	15	710	3	33		18	1?	763	22	1?
	10	1	685	4	1		6	1		20	1?	764	3	3
659	2	1?		6	1		14	33	738	5	1		5	1
	8	1?		12	1		15	1		17	1	765	2	1
	14	1?	686	2	3		18	3	739	4	1?		6	1
	18	1?		11	3	711	8	1	740	3	3		8	1
660	7	335	687	10	1		20	1		5	1		12	1
	13	1		14	1	713	2	1?		9	1	766	6	3
	15	3		16	7	714	5	129	741	2	7		15	3
661	6	1?	689	12	1?		13	1		6	1		18	3
	12	1?		20	1?	715	4	1		10	1	767	6	1?
662	3	1	690	2	151	716	5	1		18	1		14	1?
	11	3		9	1		11	1		20	1		18	1?
	15	1		13	1	717	18	1	742	15	1	769	10	1
663	14	1?		14	1		20	1	743	2	1?	770	5	3
665	14	31	691	10	1?	718	15	3	745	10	1		6	1
	18	1		12	1?	719	2	1?		12	1	771	2	1
666	22	1		16	1?		12	1?		18	1		18	1
667	6	1?	692	5	1		14	1?	746	2	1	772	9	1
	18	1?		9	5	721	6	1	747	6	1		13	3
668	11	1	693	6	7		10	1		14	1		19	15
669	4	1	694	3	3		18	1	748	7	3	773	6	1?
	10	1		19	1	722	26	1		9	1		12	1?
	18	1	695	18	1?	723	2	1	749	2	1?		20	1?
	20	1		20	1?		12	1		14	1?	774	2	1
670	6	3	697	4	1		20	1	750	14	1		14	21
	7	1		10	1	724	13	1	751	6	1?		18	1
	10	1		18	1		15	1		12	1?	775	6	1
671	6	1?	698	5	3		19	1	753	16	1		16	1
	12	23?		17	3	725	2	1?		20	1	777	16	1
	20	23?	699	4	1		8	1?	754	10	3		20	7
673	4	1?		10	1		14	1?		13	1	778	21	1
674	5	1	700	9	1	726	2	1	755	2	1	779	2	1?
	9	3		15	5		6	9		18	1		12	1?
	14	1	701	18	1		15	1		20	1		20	1?
675	22	1	702	14	7	727	4	1?	756	3	1	780	13	3
676	3	3		18	9		6	1?	757	16	1?		19	1
	15	3		19	1		16	1?		18	1?	781	16	1?
677	8	1?	703	6	1?	729	24	7?	758	6	1	782	3	1
	14	1?	705	6	31	730	13	3		14	1		14	3
678	10	3		14	1	731	8	1?	759	4	1		15	3
	11	1589	706	21	1		18	1?		8	1	783	2	7?
679	10	1?	707	6	1?		20	1?		10	1		12	1?
681	22	1?		8	1?	732	11	45		14	1		20	1?
682	6	3	708	7	35	733	10	1?	761	2	1?	785	2	1?
	13	1		9	1	734	3	1		6	1?		12	1?
	18	23		11	13	735	8	1		8	1?	786	5	63

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
787	6	1?	813	4	1	836	15	1	862	31	3	891	2	1
788	11	1		14	1	837	6	1	863	6	1?		8	1
	15	5	814	15	1		14	1		12	1?	892	3	1
	17	3		18	1	838	7	1	865	4	1		13	5
789	14	1		19	3		19	1		16	1		15	1
790	3	1	815	8	1?	839	12	1?	866	2	1	893	2	1?
	7	31		14	1?	841	12	1?		6	3	894	3	3
	10	11		20	1?		16	1?		17	3		15	7
	18	11	817	6	1?		18	1?	867	4	1	895	4	1?
791	2	1?		10	1?	842	5	1		16	1		10	1?
	8	1?	818	2	1		14	1		18	1		16	1?
793	6	1?		5	1	843	6	1	868	19	43	897	8	1
	16	1?		6	1		10	1	869	12	1		10	1
794	14	3		17	1	844	13	1		18	1	898	21	3
	17	1	819	20	7	845	8	1?	870	2	1	899	8	1?
	18	1	820	7	451		12	1?		18	1		12	1?
795	10	1		15	3		14	1?	871	6	1?		20	1?
	14	7		19	3		20	1?		18	1?	900	11	111
796	3	1	821	8	1?	846	2	1	873	2	1	901	6	1
	7	1		18	1?		3	1		10	1		10	1
	15	5		20	1?		6	27		12	1	902	3	3
797	6	1?	822	3	1		7	3	874	9	1		6	1
	14	1?		6	63		10	3		10	1		11	3
	18	1?		10	7	847	30	1		18	3	903	4	1
798	6	1		18	1	849	8	1	875	12	1		12	1
799	22	1?	823	10	1?		14	1		14	1	905	6	1?
801	12	1	825	6	1		20	1	876	23	5		12	1?
	18	1		12	1	850	6	1	877	16	1?		14	1?
802	6	3		18	1	851	6	1		18	1?	906	13	1
	9	1	826	6	1		18	1	878	15	1	907	6	1
	13	1		18	1		20	1	879	2	1	908	21	5
	18	3	827	14	1?	852	3	9		4	1	909	4	1
803	2	23?		18	1?		5	1		18	1		10	1
	14	1?	828	21	1		9	1		20	7		12	1
804	5	1	829	10	1		15	3	881	18	1		18	7
	9	5		12	1		19	3	882	10	1		20	1
	13	1		18	1	853	4	1?	883	4	1	910	18	1
805	6	1?	830	14	1		6	1?		10	1	911	2	1
	12	1?		15	1		16	1?		12	1	913	6	1?
806	11	3	831	2	1	854	18	3	884	27	3		12	1?
807	14	1		6	1	855	8	1?	885	28	1	914	18	1
	20	1		10	1		16	7?	886	3	1	915	10	1?
809	2	1?	833	2	1?		18	1?		7	1	916	3	5
	8	1?		20	1?	857	8	1?		10	1		13	3
	18	1?	834	2	1		18	1?		15	3	917	6	1?
810	2	1		9	3	858	34	1	887	6	1?		20	1?
811	10	1?		14	1	859	22	1?	889	4	1	918	10	1
	12	1?		18	1	860	9	1	890	5	3		11	9
812	3	1	835	6	1		17	5		9	1		19	57
	11	1		18	1	861	28	1?		17	1	919	4	1?

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
921	6	1	942	10	1		18	1?	995	14	1
	16	7	943	6	1?		20	1?		18	1
922	10	1		10	1?	970	9	3	996	43	1
	13	3		12	1?		13	1	997	4	1?
923	2	1?	945	8	1?	971	6	1?	998	2	1
	14	1?		18	1?		20	1?		11	1
	20	1?	946	10	1	972	5	1	999	8	1?
924	5	3	947	6	1?		13	7	1001	6	1?
	7	1		8	1?	973	6	1?		8	1?
925	4	1		14	1?		12	1?		20	1?
	18	1		18	1?		16	1?	1002	5	9
926	6	3	948	7	7	974	2	3	1003	4	1?
927	4	1		15	9	975	2	1	1004	5	7545
	6	1		19	1		6	1		15	3
	8	1	949	4	1?		20	1	1005	4	7?
	14	7		10	1?	977	8	1?	1006	3	1
929	8	1?	950	2	3		14	1?		6	1
	12	1?		3	1		20	1?		7	1
	14	1?		6	1	978	6	1		10	1
930	2	3	951	16	1?		9	3		15	1
931	10	1	953	2	1?	979	4	1	1007	18	1?
	12	1		12	1?		10	1	1009	10	1?
	16	1	954	49	1		18	1		12	1?
	18	1	955	10	1	980	9	87	1010	5	3
932	3	1		18	1	981	32	7?		18	3
	9	1	956	15	3	982	15	3	1011	6	1
	11	1		17	3	983	14	1?		8	1
933	2	1?	957	6	1?		20	1?		10	7
	4	1?		16	623?	985	10	1?	1012	3	1
	12	1?		20	7?		12	1?		9	1
	16	7?	958	6	3		16	1?	1013	2	1?
934	3	3	959	8	1?	986	2	3		6	1?
	10	1		20	1?		5	1		12	1?
	15	1	961	16	1?		17	1		20	1?
	19	1		18	1?		18	3	1014	2	7
935	2	1?	962	14	1	987	6	1		10	1
	8	1?		18	3		10	1	1015	6	1
	20	1?	963	4	1?	988	7	1		10	1
937	6	1?		6	1?		9	3	1017	16	73?
	10	1?	964	9	1		15	3		20	1?
938	2	1		15	3		19	1	1018	10	3
	5	1	965	2	1	989	2	1?	1019	2	1?
	9	1		6	1	990	10	1	1020	9	7
939	2	1?		20	1		11	3	1021	10	1?
	10	1?	966	7	1		15	1		12	1?
	14	1?		10	1	991	18	1?		18	1?
	18	1?		18	9	993	2	1	1022	3	1
940	37	1	967	16	1?		14	1		6	3
941	6	1?	969	4	1?	994	10	3		15	1
	8	1?		14	1?		13	1		18	1

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
1023	4	1?		20	1	1076	3	1	1106	2	381
	12	1?	1050	10	9		5	3		6	3
1025	6	1?		17	1		15	1	1107	10	1
	12	1?	1051	12	1?	1077	18	1		16	1
1026	2	7		18	1?	1078	6	3		18	1
	13	189	1052	5	1		7	1	1108	7	1
	17	1	1053	12	1		10	1		9	5
1027	6	1?	1054	3	1	1079	14	1?	1109	12	1?
	10	1?		18	3		18	1?		14	1?
	16	1?	1055	2	1?	1081	12	1?		18	1?
1028	17	3		14	1?	1082	9	1	1110	2	9
1029	8	1?		20	1?		14	3		11	11
1030	7	3	1057	4	1		18	1		15	15993
	10	3		6	1	1083	10	1?	1111	22	1?
	18	11	1058	14	1		20	1?	1113	10	7?
	19	3		17	3	1084	3	3	1114	22	1
1031	2	1?	1059	14	1?		7	1	1115	6	1?
	12	1?	1060	3	825	1085	18	1		18	1?
1033	4	1?		15	3		20	1	1116	11	3
	6	1?	1061	6	1?	1086	7	9		17	3
	10	1?		20	1?		10	1	1117	6	1?
	16	1?	1062	3	7	1087	4	1?		10	1?
1034	2	3		6	3		16	1?		18	1?
	5	1		14	1	1089	4	1?	1118	2	1
	17	3	1063	4	1		10	1?		6	1
1035	6	1?		6	1		18	7?	1119	2	1
1036	7	3		10	1	1090	9	3		14	1
	13	3		12	1	1091	6	1?		20	1
	15	129	1065	2	1?		12	1?	1121	2	1
1037	8	1?		4	1?		20	1?		8	1
	14	1?		8	1?	1092	15	1	1122	6	1
1038	6	3		12	1?		19	15		9	1
	14	1	1066	6	1	1093	4	1?	1123	4	1?
1039	4	1?		13	1		16	1?		12	1?
	10	1?	1067	8	1?	1094	15	1	1124	3	1
1041	2	1		14	1?	1095	14	1?	1125	8	1?
	6	1		18	1?	1097	14	1?		20	1?
	8	1		20	1?	1098	9	1	1126	7	1
1042	18	1	1068	7	1		14	1		10	1
1043	2	1?		9	5	1099	4	1?		18	1
	12	1?	1069	10	1?	1100	5	1	1127	6	1?
1044	7	1		12	1?		9	3		18	1?
	9	3	1070	2	1		17	1		20	1?
	15	5		6	1	1101	6	7	1129	4	1?
	17	3	1071	10	1?	1102	3	1		18	1?
1045	6	89?		16	1?		19	1	1130	5	1
1046	6	1	1073	30	1?	1103	2	1		6	1
1047	36	1	1074	13	1		6	1		17	3
1049	2	1		18	1		14	1		18	3
	12	1	1075	6	1	1105	18	1?	1131	8	1?

n	k	Ind	n	k	Ind	n	k	Ind	n	k	Ind
	10	1?	1148	5	1	1166	2	1	1185	2	1
	16	1?		9	3		3	1		12	1
	20	1?	1149	14	7?		6	3	1186	21	3?
1132	13	3	1150	19	3		15	1	1187	8	1?
	15	1	1151	6	1?	1167	8	1?		14	1?
1133	2	1?		8	1?		14	1?		20	1?
	12	1?		18	1?	1169	2	1	1188	19	1
1134	2	3		20	1?		12	1	1189	24	1?
	15	9	1153	22	1?	1170	5	1	1190	3	1
1135	10	1?	1154	2	1		14	1		15	1
	18	1?		18	1		17	11	1191	28	7?
1137	6	7?	1155	2	1	1171	6	1?	1193	6	1?
	14	1?		4	1	1172	3	1		14	1?
1138	6	1?		8	1		5	1	1194	2	1
	9	1?		16	1		11	1		10	1
1139	24	1?	1156	3	15		15	1		18	1
1140	5	341		7	3	1173	6	1?	1195	12	31?
	13	3		15	5	1174	7	1?		16	1?
	19	7	1157	8	1?		19	1?	1196	17	3
1141	12	1?	1158	6	1	1175	24	1?	1197	4	1?
	16	1?		11	1	1177	18	1?		14	1?
1142	23	1?		19	1	1178	2	3?	1198	7	3?
1143	16	1?	1159	4	1?		6	1?		10	3?
	20	1?	1161	12	1?		14	1?		15	3?
1145	8	1?		18	1?	1179	8	1?	1199	2	1?
	18	1?	1162	9	3?	1180	21	25		12	1?
	20	1?		10	1?	1181	12	1?		14	1?
1146	2	1		13	1?	1182	3	1		20	1?
	17	1	1163	32	1?		10	3			
1147	6	1?	1164	9	7	1183	10	1?			
	16	1?	1165	6	1?		12	1?			