

involve

a journal of mathematics

Mapping the discrete logarithm

Daniel Cloutier and Joshua Holden

 mathematical sciences publishers

2010

vol. 3, no. 2

Mapping the discrete logarithm

Daniel Cloutier and Joshua Holden

(Communicated by Carl Pomerance)

The discrete logarithm is a problem that surfaces frequently in the field of cryptography as a result of using the transformation $x \mapsto g^x \bmod n$. Analysis of the security of many cryptographic algorithms depends on the assumption that it is statistically impossible to distinguish the use of this map from the use of a randomly chosen map with similar characteristics. This paper focuses on a prime modulus, p , for which it is shown that the basic structure of the functional graph produced by this map is largely dependent on an interaction between g and $p - 1$. We deal with two of the possible structures, permutations and binary functional graphs. Estimates exist for the shape of a random permutation, but similar estimates must be created for the binary functional graphs. Experimental data suggest that both the permutations and binary functional graphs correspond well to the theoretical predictions.

1. Introduction

Just a few decades ago, cryptography was considered a domain exclusive to national governments and militaries. However, the computer explosion has changed that. Every day, millions of people trust that their privacy will be protected as they make online purchases or communicate privately with a friend. Many of the cryptographic algorithms they will use are built upon a common transformation, namely

$$x \mapsto g^x \bmod n \tag{1}$$

where $\gcd(g, n) = 1$ and the transformation is considered as a function from $\{1, \dots, n - 1\}$ to itself. (We will call functions of this form discrete exponentiation maps.) For instance, Diffie–Hellman key exchange [Diffie and Hellman 1976], RSA [Rivest et al. 1978], and the Blum–Micali pseudorandom bit generator [Blum and Micali 1984] all use discrete exponentiation maps. In particular, if n is a prime and g is a primitive root modulo that prime, then a discrete exponentiation map has

MSC2000: primary 11Y16; secondary 11-04, 94A60, 05A15.

Keywords: discrete logarithm problem, random map, functional graph.

Holden was supported in part by a Rose–Hulman Summer Professional Development Grant during the summer of 2009.

an inverse which is known as the *discrete logarithm*. The security of the Diffie–Hellman protocol and the Blum–Micali generator both rely on the idea that the discrete logarithm is difficult to calculate.

Furthermore, the analyses of the security of many algorithms rely on the idea that not only is calculating the inverse of a discrete exponentiation map difficult, but in fact that certain properties of discrete exponentiation maps and/or discrete logarithms cannot be predicted better than a random guess. (It is not known to the authors who first suggested this general idea; it may be folklore.) For example, in [Blum and Micali 1984] the cryptographic security of a particular pseudorandom bit generator relies on the hypothesis that a certain property of discrete exponentiation cannot be predicted better than a random guess. Similarly, [Boneh 1998] shows that if certain statistical properties of the Diffie–Hellman problem cannot be guessed better than randomly then the Diffie–Hellman protocol can be made much more efficient than otherwise. This paper will consider some statistics of maps on $\{1, \dots, n - 1\}$ such that the expected values of these statistics for a randomly chosen map in a class containing the discrete exponentiation maps can be calculated theoretically. We conjecture that the particular values of these statistics for discrete exponentiation maps will resemble the expected values for the random maps. Furthermore, we will collect experimental data on discrete exponentiation maps for various values of g and n and compare them to our expected values to give evidence for this conjecture.

Some readers might be familiar with other papers that look at the discrete exponentiation map from a statistical point of view, such as [Canetti et al. 2000]. In both cases n is fixed and “measurements” are taken from a (nonrandom) sample which is derived from discrete exponentiation maps. The distribution of the measurements on the sample is then compared with the distribution of measurements taken from random samples of a certain population. In [Canetti et al. 2000], g is fixed, the measurements are a specified set of bits from triples of numbers, the sample is triples of the form (g^x, g^y, g^{xy}) (with varying x and y), and the population is all strings of bits. In this paper, the measurements are various graph-theoretic properties of functional graphs (as defined below). The sample is maps of the form (1) (with varying g) which have a certain property on their in-degrees and the population is all functional graphs with that same property.

2. Terminology and background

Throughout this paper, ϕ denotes the Euler phi function. The letter n will stand for an odd prime. We will examine mappings

$$f : S = \{1, 2, \dots, p - 1\} \rightarrow S$$

of the form $x \mapsto g^x \bmod p$, where $p \geq 3$ is a prime modulus and $\gcd(g, p) = 1$. In some instances, it will prove to be useful to interpret the mappings as functional graphs. A functional graph is a directed graph such that each vertex must have exactly one edge directed out from it. The relationship between the mappings which interest us and functional graphs is straightforward. Each element in S can be interpreted as a vertex. The edges are defined such that an edge $\langle a, b \rangle$ is in the graph if and only if $f(a) = b$.

There are a number of statistics of interest derived from functional graphs; in particular, Flajolet and Odlyzko — henceforth abbreviated FO — have treated random mappings in detail. Following the conventions in [FO 1990b], let $f : S \rightarrow S$ be the transition function so that the edges in the functional graph can be expressed as the ordered pair $\langle x, f(x) \rangle$ for $x, f(x) \in S$. By applying the pigeonhole principle and noting that the cardinality of S is $p - 1$ we can say that by starting at any random point u_0 and following the sequence $u_1 = f(u_0)$, $u_2 = f(u_1)$, \dots , there must be a $u_i = u_j$ after at most p iterations. Suppose u_i occurs before u_j in the sequence of nodes. In this case, the tail length is the number of iterations of the function from u_0 to u_i . The cycle length is the number of iterations from u_i to u_j . In more natural graphical terms, the tail length is the number of edges involved in the directed path from u_0 to u_i , and the cycle length is the number of edges (or equivalently nodes) involved in the directed path from u_i to itself. Additionally, a terminal node is one with no preimage, or more formally, x is a terminal node if $f^{-1}(x) = \emptyset$. A node is an image node if it is not a terminal node. Since each node has an out-degree of exactly one, each cycle with the trees grafted onto its nodes will form a connected component.

When a functional graph is produced from a discrete exponentiation function, we will call it a discrete exponentiation functional graph. The value of g plays a major role in determining the basic structure of discrete exponentiation functional graphs. In fact, as Theorem 1 formalizes, the interaction between g and $p - 1$ will effectively fix the in-degrees of the nodes in the graph. First, though, define an m -ary functional graph to be a graph where each node has in-degree of exactly zero or m . The proof of the following theorem is then straightforward.

Theorem 1. *Let p be fixed and let m be any positive integer that divides $p - 1$. Then as g ranges from 1 to $p - 1$, there are $\phi((p - 1)/m)$ different functional graphs which are m -ary produced by maps of the form $f : x \mapsto g^x \bmod p$. Furthermore, if r is any primitive root modulo p , and $g \equiv r^a \bmod p$, then the values of g that produce an m -ary graph are precisely those for which $\gcd(a, p - 1) = m$.*

Theorem 1 gives a strong indication that the graphs generated by (1) have to be considered separately for different values of m . It should be noted, though, that there are some values of m which lead to completely predictable graphs. For

instance, there is one $(p - 1)$ -ary graph that corresponds to $g \equiv 1 \pmod{p}$. There is also one $((p - 1)/2)$ -ary graph that corresponds to $g \equiv -1 \pmod{p}$. In general, however, an m -ary discrete exponentiation functional graph is not trivially predictable. This paper will restrict its focus to unary functional graphs (which will be referred to as permutations since they simply permute the numbers $1, \dots, p - 1$) and binary functional graphs. As a consequence of Theorem 1, we can observe that the values of g which produce a permutation are precisely those which are primitive roots modulo p , and the values of g which produce a binary functional graph are precisely those which are the squares of primitive roots modulo p .

In cryptography, it is common to look for primes where $p - 1$ has at least one large prime factor. For instance, the pseudorandom bit generator described in [Genaro 2005], which is a modification of the Blum–Micali generator mentioned in Section 1, specifically requires the modulus to be of the form $p = 2q + 1$ where q is also prime. A prime of this form is known as a *safe prime* (q is also known as a *Sophie Germain prime*). These primes are of interest here not only because of their extensive use in cryptography, but also because $p - 1$ has only four divisors, namely $1, 2, q = (p - 1)/2$ and $2q = p - 1$. In addition to the one $(p - 1)$ -ary and one $((p - 1)/2)$ -ary graph mentioned above, there are $\phi(q)$ permutations and $\phi(q)$ binary functional graphs which represent the remaining values of g (since $\phi(q)$ is $q - 1$). Thus, not only do safe primes provide large numbers of permutations and binary functional graphs, but every graph generated by a safe prime is either trivial (the graphs where g is either 1 or -1) or fits into the theoretical framework presented in Section 3.

We can now present the central conjecture of this paper, which as far as we know has not been previously considered in this form:¹

Conjecture 2. *The average values of the following statistics are asymptotically the same for m -ary discrete exponential functional graphs on $n = p - 1$ nodes and for random m -ary functional graphs on n nodes as n goes to infinity:*

Number of components

Number of tail nodes

Number of image nodes

Average cycle length

Maximum tail length

(as seen from a random node)

Number of cyclic nodes

Number of terminal nodes

Maximum cycle length

Average tail length

(as seen from a random node)

We are a long way from proving this conjecture but we will give some supporting evidence for it in the cases of $m = 1$ and $m = 2$.

¹Pollard [1978] considers functional graphs corresponding to a similar map when analyzing his kangaroo method. That map takes $x \mapsto xg^{f(x)}$ for some pseudorandom function f , however.

3. Theoretical results

In Theorem 1, it is shown that the in-degree of each node is dependent on the value of both g and p . This is clearly imposing a structure on any functional graphs generated using (1). While most of the parameters that are of interest depend on the exact graph generated, the number of image nodes can be computed directly from the values of g and p . The proof is again straightforward.

Theorem 3. *The number of image nodes in any m -ary graph is $(p - 1)/m$.*

This fact helps quantify the repercussions of Theorem 1 and the restrictions on in-degree in m -ary graphs. The number of image nodes is a direct function of m which can greatly limit the shapes each graph can take on. None of the other parameters appear to be strictly controlled by m in this fashion.

3.1. Permutations. Predicting the behavior of the permutations is, in many ways, much easier than other m -ary graphs. The most important reason for this is that there are no terminal nodes or tail nodes. This follows quickly from the definition of a permutation as a unary functional graph and the fact that the sum of the in-degrees must be the same as the sum of the out-degrees. Each node has an out-degree of exactly one, and if any node were to have an in-degree of zero, then, by the pigeon-hole principle, at least one node must have an in-degree of more than one. This is not allowed so each node must have in-degree of exactly one. Furthermore, since every tail must contain at least one terminal node, this also implies that every node is cyclic. The parameters that can then be determined from the definition of a permutation are:

Number of cyclic nodes	n	Number of tail nodes	0
Number of terminal nodes	0	Number of image nodes	n
Average tail length	0	Maximum tail length	0

Theorem 4. *The expected values for the number of components, the average cycle length as seen from a random node and the maximum cycle length in a random permutation of size n have the following asymptotic forms:*

$$\text{Number of components} = \sum_{i=1}^n \frac{1}{i} + o(\log n), \tag{i}$$

$$\text{Average cycle length} = \frac{n + 1}{2} + o(1), \tag{ii}$$

$$\begin{aligned} \text{Maximum cycle length} &= n \int_0^\infty \left[1 - \exp\left(-\int_v^\infty e^{-u} \frac{du}{u}\right) \right] dv + o(n) \tag{iii} \\ &\approx 0.62432965n + o(n). \end{aligned}$$

Parts (i) and part (ii) are fairly well known. Part (iii) seems to have first been solved in [Shepp and Lloyd 1966]. An alternative solution and proof more similar to the methods used here is offered in [FO 1990a].

3.2. Binary functional graphs. While estimates for the parameters investigated here exist in the literature for the random functional graphs and permutations, it does not appear that estimates for binary functional graphs have ever been all collected in one place. However, the methods in [FO 1990b] can be extended to develop these estimates, and some of the following results have appeared in various places already. Imitating those methods, we first need to convert our ideas of a binary functional graph into corresponding generating functions. We first note that a binary functional graph is a set of components. Each component is a cycle of nodes with each node having an attached binary tree to bring its in-degree to two. A binary tree is either a node (terminal node) or a node with two binary trees attached. Finally, a node is simply an atomic unit. A moment's reflection should indicate that this natural specification does, in fact, specify a binary functional graph.²

Imitating the transformations in [FO 1990b, Section 2.1], the generating functions of interest are

$$f(z) = e^{c(z)} = \frac{1}{1-zb(z)}, \quad (2)$$

$$c(z) = \ln \frac{1}{1-zb(z)}, \quad (3)$$

$$b(z) = z + \frac{1}{2}zb^2(z). \quad (4)$$

Here f generates the number of binary functional graphs, c generates the number of components, and b generates the number of binary trees of a given size. Solving the quadratic formula for (4), we can produce the following formulas for f and c which simplify some of the cases:

$$f(z) = \frac{1}{\sqrt{1-2z^2}}, \quad c(z) = \ln \frac{1}{\sqrt{1-2z^2}} \quad (5)$$

See also [FO 1990b, (70)] and [Flajolet et al. 1991, Theorem 11].

To compute asymptotic forms of any of the statistics of interest, we must first compute an asymptotic form for f to normalize results. The following derivations

²In the notation of [FO 1990b]:

BinFunGraph	= set(Components),
Component	= cycle(Node*BinaryTree),
BinaryTree	= Node + Node*set(BinaryTree, cardinality = 2),
Node	= Atomic Unit.

give only a highlight of the methods used by Flajolet and Odlyzko. The interested reader is encouraged to see [FO 1990a; 1990b] for detailed proofs.

From the formula for $f(z)$ in (5) it is clear that there is a singularity at $z = 1/\sqrt{2}$. Performing a singularity analysis³ as in [FO 1990b, Section 2], the asymptotic form for f falls out quickly as

$$f(z) \sim \frac{2^{n/2}}{\sqrt{\pi n/2}}. \tag{6}$$

In at least one case, there are some important second-order interactions between the error terms of the number of graphs and the appropriate statistic. In these cases, a more exact form of (6) must be used. Expanding one more term in the expansion of f gives

$$f(z) \sim \frac{2^{n/2}}{\sqrt{\pi n/2}} - \frac{2^{n/2}}{4n\sqrt{\pi n/2}} = \frac{2^{n/2}(4n-1)}{4n\sqrt{\pi n/2}}. \tag{7}$$

In most cases, using this more precise expansion of f is not necessary and does not change the results. Therefore, in all but the necessary cases, (6) will be used.

We begin by deriving the results for the simplest parameters.

Theorem 5. *The expected values for the number of components, number of cyclic nodes, number of tail nodes, number of terminal nodes and number of image nodes in a random binary functional graph of size n , as $n \rightarrow \infty$ have the following asymptotic forms:*

$$\text{Number of components} = \frac{\ln(2n) + \gamma}{2} + o(1), \tag{i}$$

$$\text{Number of cyclic nodes} = \sqrt{\pi n/2} - 1 + o(1), \tag{ii}$$

$$\text{Number of tail nodes} = n - \sqrt{\pi n/2} + 1 + o(1), \tag{iii}$$

$$\text{Number of terminal nodes} = n/2, \tag{iv}$$

$$\text{Number of image nodes} = n/2. \tag{v}$$

In part (i), γ represents the Euler constant which is approximately 0.57721566. The results for parts (iv) and (v) can in fact be shown to be exact and not merely asymptotic. The highlights of the proofs as they differ from those in [FO 1990b] follow.

Proof. As in [FO 1990b], the following bivariate generating functions need to be defined with parameter u marking the elements of interest. The generating

³The analyses in this paper have been performed using the computer algebra program Maple and the packages created as part of the Algorithms Project at INRIA, Rocquencourt, France. The packages can be found online at <http://algo.inria.fr/libraries/#down>.

functions for the number of components, number of cyclic nodes and number of terminal nodes are respectively:

$$\xi_1(u, z) = \exp\left(u \ln \frac{1}{1-zb(z)}\right), \quad (8)$$

$$\xi_2(u, z) = \frac{1}{1-uzb(z)}, \quad (9)$$

$$\xi_3(u, z) = \frac{1}{\sqrt{1-2uz^2}}. \quad (10)$$

(Equation (9) may also be found in [Flajolet et al. 1991, Theorem 11].) Imitating the methods in [FO 1990b], the mean value generating function, $\Xi(z)$, is found by taking the partial derivative of $\xi(u, z)$ with respect to u and evaluating at $u = 1$. This yields

$$\Xi_1(z) = \frac{1}{1-zb(z)} \ln \frac{1}{1-zb(z)}, \quad (11)$$

$$\Xi_2(z) = \frac{zb(z)}{(1-zb(z))^2}, \quad (12)$$

$$\Xi_3(z) = \frac{z^2}{(1-2z^2)^{3/2}}. \quad (13)$$

The forms in the statement of the theorem follow by expanding around the singularity $z = 1/\sqrt{2}$, applying singularity analysis as in [FO 1990b], and normalizing parts (i) and (ii) by (6) and (iv) by (7). Parts (iii) and (v) follow from parts (ii) and (iv) respectively since the respective pairs must sum to n . Also note that part (iv) can also be derived in an elementary fashion from the definition of the binary functional graph. \square

The asymptotic values for the average length of cycles and tails as seen from a random point in the graph are also interesting. The asymptotic forms of these values are given in Theorem 6.

Theorem 6. *The expected values for the cycle size and tail length as seen from a random node in a random binary functional graph of size n have the following asymptotic forms as $n \rightarrow \infty$:*

$$\text{Average cycle length} = \sqrt{\pi n/8} + o(\sqrt{n}), \quad (i)$$

$$\text{Average tail length} = \sqrt{\pi n/8} + o(\sqrt{n}). \quad (ii)$$

Proof. In order to calculate the average cycle length and average tail length, the generating functions must be manipulated to account for each node in the cycle or tail. This can be done by using the same methods as in the previous proof, but marking only one component or tail at a time. This is essentially the same as the

strategy which is used to prove the result for average cycle size in [FO 1990b]. More background on the method can be found there.

Let $\xi_1(z)$ be the exponential generating function for the total cycle length over all binary functional graphs and $\xi_2(z)$ be the exponential generating function for the total tail length. Then, $\xi_1(z)$ can be defined as

$$\xi_1(z) = \frac{\partial^2}{\partial w \partial u} \left[\frac{1}{1 - \sqrt{1 - 2z^2}} \ln \left(\frac{1}{1 - u(1 - \sqrt{1 - 2(zw)^2})} \right) \right]_{u=1, w=1}. \quad (14)$$

In (14), u marks the cyclic nodes in the component we are considering and w marks all nodes in that component, so that each node in the component is weighted with the number of nodes in the cycle. (In [Salvy 1997], the method of “decorated” graphs is used to develop a generating function for a variation of this problem.)

In order to compute total tail length we need a version of the generating function for binary trees which marks the edges along one tail. We can write that as

$$\beta(z, u) = z + \frac{1}{2}zb^2(z) + uzb(z)\beta(z, u). \quad (15)$$

Then solving (15) and plugging it in appropriately gives us

$$\xi_2(z) = \frac{\partial}{\partial u} \left[\frac{1}{\sqrt{1 - 2z^2}} \frac{1}{\sqrt{1 - 2z^2}} \frac{u(1 - \sqrt{1 - 2z^2})}{(1 - u(1 - \sqrt{1 - 2z^2}))} \right]_{u=1}. \quad (16)$$

Note that the first factor in (16) is for the unmarked components and the second is for the unmarked trees in the marked components. (In [Salvy 1997] and [Flajolet et al. 1989; Mishna 2004],⁴ the methods of “decorated” graphs and attribute grammars, respectively, are used to develop the same generating function.)

Performing a singularity analysis of the two generating functions and normalizing by $2^{n/2}/(n\sqrt{\pi n/2})$, as done in the previous theorems, leads to the statement of the theorem. The additional factor of n in the denominator is needed to compensate for the fact that the parameters were estimated across all nodes in the graph and the goal is to determine them from any single random node in the graph. \square

The final parameters that needs to be calculated are the average maximum cycle length and the average maximum tail length.

Theorem 7. *The expected sizes of the largest cycle and the largest tail in a random binary functional graph of size n have the following asymptotic forms as $n \rightarrow \infty$:*

⁴According to [Flajolet et al. 1989], results from this analysis were first obtained by hand in [Flajolet 1979].

$$\begin{aligned} \text{Largest cycle} &= \sqrt{\frac{\pi n}{2}} \int_0^\infty \left[1 - \exp\left(-\int_v^\infty e^{-u} \frac{du}{u}\right) \right] dv + o(\sqrt{n}) \quad (\text{i}) \\ &\approx 0.78248\sqrt{n} + o(\sqrt{n}); \end{aligned}$$

$$\begin{aligned} \text{Largest tail} &= \sqrt{2\pi n} \ln 2 - 3 + 2 \ln 2 + o(1) \quad (\text{ii}) \\ &\approx 1.73746\sqrt{n} - 1.61371 + o(1). \end{aligned}$$

Proof. The proof for part (i) result follows precisely the methods of [FO 1990b] with substitution of the proper generating function f , and is therefore omitted.

The proof for part (ii) follows a combination of [FO 1990b, Theorem 6] and [FO 1982, Sections 3–5]. Let $b^{[h]}(z)$ be the exponential generating function for the number of binary trees with height at most h and $f^{[h]}(z)$ be the exponential generating function for the number of binary functional graphs with maximum tail length less than or equal to h , so that (as in [FO 1990b, Equations 41 and 42])

$$f^{[h]}(z) = \frac{1}{1 - zb^{[h]}(z)}$$

and

$$b^{[h+1]}(z) = z + \frac{1}{2}z(b^{[h]}(z))^2, \quad b^{[0]}(z) = z.$$

Now, as in [FO 1982, Proposition 2], note that

$$b(z) - b^{[h+1]}(z) = \frac{1}{2}z(b(z) - b^{[h]}(z))(b(z) + b^{[h]}(z)),$$

so if we let

$$e_h(z) = \frac{b(z) - b^{[h]}(z)}{2b(z)},$$

then

$$e_{h+1}(z) = (1 - \sqrt{1 - 2z^2})e_h(z)(1 - e_h(z)).$$

Now we want to approximate $e_h(z)$ with a function of h and some $\epsilon(z)$. If we let $\epsilon = \sqrt{1 - 2z^2}$ then we have

$$e_{j+1} = (1 - \epsilon)e_j(1 - e_j); \quad e_{-1} = 2.$$

This is essentially the same recursion as in [FO 1982]. As in Lemma 5 there, we can then “normalize” and “take inverses” to get the approximation

$$e_h \approx \frac{(1 - \epsilon)^{h+1} \epsilon}{1 - (1 - \epsilon)^{h+1}}. \quad (17)$$

The details of the error bounds proceed as in [FO 1982]; we omit them here.

The generating function associated to the average maximum tail length is, as in [FO 1990b, Equation 43],

$$\Xi(z) = \sum_{h \geq 0} \left[\frac{1}{1-zb(z)} - \frac{1}{1-zb^{[h]}(z)} \right],$$

and we proceed as in [FO 1990b, Equation 51] to write

$$\Xi(z) = \frac{2zb(z)}{1-zb(z)} \sum_{h \geq 0} \frac{e_h(z)}{1-zb(z)+2e_h(z)zb(z)}.$$

Putting this entirely in terms of ϵ and h , and shifting the index of summation for convenience, we can write

$$\Xi(z) \approx \frac{2(1-\epsilon)}{\epsilon} \sum_{h \geq 1} \frac{(1-\epsilon)^h}{1+(1-2\epsilon)(1-\epsilon)^h}. \tag{18}$$

We approximate the sum with an integral, using Euler–Maclaurin summation. Taking the integral and noting that $\ln(1-\epsilon) \sim -\epsilon$ as $\epsilon \rightarrow 0$, we finally get

$$\Xi(z) \approx \frac{2(1-\epsilon)}{\epsilon^2(1-2\epsilon)} \ln(2-3\epsilon+2\epsilon^2). \tag{19}$$

The next step is to substitute $\epsilon = \sqrt{1-2z^2}$ into (19) and do the singularity analysis, which gives us the statement of the theorem. \square

4. Observed results

In [Holden 2002; Holden and Moree 2004; 2006], heuristics and observed values for the number of small cycles (fixed points and two-cycles) in discrete exponentiation graphs are given. Our methods build on this to generate experimental data for the parameters described by the theoretical predictions in Section 3. The method of data collection was straightforward. A prime was chosen as the modulus and then for each $g \in \{1, 2, 3, \dots, p-1\}$, the corresponding discrete exponentiation binary functional graph or permutation was generated. The results were then computed as average statistics over all $p-1$ graphs observed. The permutations and binary functional graphs were noted and their results were also tabulated separately. In this manner, the data can be examined in its complete form over all graphs and individually over the permutations and binary functional graphs. The generation and analysis of each of the graphs was handled by C++ code written by the first author.

	100043	100057	106261
Permutations	50020	30240	21120
Binary functional graphs	50020	15120	10560
Total functional graphs	100042	100056	106260

Table 1. The number of permutations, binary functional graphs and total discrete exponentiation functional graphs associated with $p = 100043$, $p = 100057$, and $p = 106260$.

The primes chosen for these calculations were

$$100043 = 2 \cdot 50021 + 1,$$

$$100057 = 2^3 \cdot 3 \cdot 11 \cdot 379 + 1,$$

$$106261 = 2^2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 23 + 1.$$

The total number of graphs, permutations and binary functional graphs can be computed using Theorem 1 and are shown in Table 1.

In Section 4.1, the observed results for the discrete exponentiation permutations will be compared to the theoretical results for random permutations given in Theorem 4. Finally, the observed results for the discrete exponentiation binary functional graphs will be examined in Section 4.2. Theorems 5 through 7 will provide the theoretical predictions for these values on random binary functional graphs. Since the terminal nodes and tail nodes can be directly computed from the image nodes and cyclic nodes, including them in the collected data does not add any insight. For this reason, they have both been excluded from the analysis conducted in the following sections. The Appendix gives some of the interesting extremal data such as the longest cycle observed for each prime. More information on the data and how they were computed may be found in [Cloutier 2005].

4.1. *Permutation results.* The results of looking at only the values of g that were a primitive root modulo p (and thus produced permutation discrete exponentiation graphs) can be found in Table 2.

The percent error here is nearly zero in every instance. This seems to indicate that there are no obvious structural differences between a random permutation and a permutation generated by the process used here.

4.2. *Binary functional graph results.* The binary functional graphs should prove more interesting than the permutations examined in the previous section. Unlike permutations, binary functional graphs do not appear to have been previously studied in detail. The statistics derived from the binary discrete exponentiation

functional graphs and the error when compared to the results derived for random binary functional graphs in Section 3.2 can be found in Table 3.

The number of image nodes came out exactly as expected and predicted by Theorem 3. However, in many other cases the results were nearly as good. The relative size of the error decreases as the number of binary discrete exponentiation functional graphs increases over the different primes. This is especially worth noting for $p = 100043$ which has over fifty thousand binary functional graphs while 100057 and 106261 have approximately fifteen thousand and ten thousand respectively. Since having more graphs appears to push the results closer to those derived in Section 3.2, this seems to further support the claim that any binary functional graph produced by our mapping does in fact resemble a randomly chosen binary functional graph.

5. Conclusions and future work

The transformation used here to generate functional graphs is an exceedingly important transformation in cryptography. If the output of the function were to fall into a predictable pattern, it could be an exploitable flaw in many algorithms considered secure today. For instance, the average cycle length seems particularly important for pseudorandom bit generators since, in many cases, it relates directly to the predictability of the pseudorandom bit generator. As Theorem 1 demonstrates, the use of (1) repeatedly forces a nontrivial structure onto the graphs generated. This is certainly worth investigating as any imposed structure may be open to an exploit.

The advantage of using a safe prime is that every nontrivial graph can be analyzed by the theoretical framework laid out in this paper. Their use is also very prevalent in cryptographic applications. As mentioned above, the pseudorandom bit generator specified in [Gennaro 2005] requires the use of a safe prime to defend against other attacks. However, the methods used for binary functional graphs in Section 3.2 can and should be extended to larger values of m . (This is currently underway in the case $m = 3$ and some results may be found in [Brugger and Frederick

	100043		100057		106261	
	Observed	Error	Observed	Error	Observed	Error
Components	12.081	0.083%	12.054	0.306%	12.126	0.205%
Avg cycle	49980.551	0.082%	50191.352	0.326%	53105.104	0.048%
Max cycle	62395.488	0.102%	62627.745	0.256%	66245.807	0.144%

Table 2. Observed results for the three primes over the permutation discrete exponentiation graphs, with corresponding errors.

	100043		100057		106261	
	Observed	Error %	Observed	Error %	Observed	Error %
Components	6.389	0.047	6.364	0.437	6.370	0.810
Cyclic nodes	395.303	0.029	395.858	0.105	408.433	0.217
Image nodes	50021	0	50028	0	53130	0
Avg cycle	198.319	0.056	197.766	0.230	202.651	0.795
Avg tail	197.961	0.125	197.550	0.339	202.422	0.907
Max cycle	247.261	0.094	247.302	0.082	256.986	0.754
Max tail	541.827	1.115	549.588	1.145	566.370	1.744

Table 3. The observed results for the three primes over all binary discrete exponentiation functional graphs generated and the corresponding percent errors.

2007; Brugger 2008].) In an ideal case, they should be extended to the general case of an m -ary graph, which can be specified as a set of components, each of which is a cycle of nodes with each node having an attached m -ary tree.⁵ The associated generating functions for these functional graphs would be

$$f(z) = e^{c(z)}, \quad c(z) = \ln\left(1 - \frac{z}{(m-1)!}t^{m-1}(z)\right)^{-1}, \quad t(z) = z + \frac{z}{m!}t^m(z),$$

where $f(z)$ is the exponential generating function associated to the functional graphs, $c(z)$ is the exponential generating function associated to the connected components and $t(z)$ is associated to the trees. The methods in Section 3.2 could also be extended to obtain values for additional parameters such as the average and maximum tree size.

This paper has focused on the graphs generated when the modulus is prime. In practice, though, this is not always the case. For this reason, it could be worthwhile to attempt to extend the type of analysis done here to a composite modulus. Some work in this direction may be found in [Mace 2009].

While the data generated for this project appears to confirm that the graphs do tend toward the shape and structure of a random graph of the appropriate type, no data were collected on the distribution of the different parameters. This data could help to give a clearer picture of how closely individual graphs may be expected to exhibit the characteristics of a random graph, especially given the observation that primes with a larger number of binary functional graphs seem to conform better to

⁵In the notation of [FO 1990b]:

FunctionalGraph = set(Components),
 Component = cycle(Node*Set(Tree, cardinality = $m - 1$)),
 Tree = Node + Node*set(Tree, cardinality = m),
 Node = Atomic Unit.

prediction on the average. The methods used in [Flajolet et al. 1993] would seem to be potentially helpful here. In addition, finding the standard deviation for the parameters of interest across all graphs of the appropriate type would allow us to do a more sophisticated analysis of the observed errors. Initial work along these lines has been done for permutations in [Hoffman 2009] and for binary functional graphs in [Lindle 2008].

Appendix: Extremal data

For $p = 100043$, the longest cycle observed was 100042 which occurred for two different values of g . They were $g = 20812$ and $g = 94034$. The longest tail had a length of 1448 and was observed when $g = 89339$. There were five instances where the graphs contained no cycles longer than one which occurred for $g = 1, 72116, 91980, 95997, \text{ and } 100042$.

The graphs generated by $p = 100057$ had an overall longest cycle of 100052 when $g = 58303$. The longest tail observed was 1589 when $g = 18115$. There were also 26 different values of g that produced a graph that did not have a cycle longer than one.

The largest cycle observed in graphs generated using $p = 106261$ was 106257 when $g = 102141$. The longest tail was 35822 when $g = 1480$. There were 92 different values of g that produced graphs with no cycles longer than a fixed point.

Acknowledgments

The authors thank the anonymous referee for advice on how to improve the clarity of the paper.

References

- [Blum and Micali 1984] M. Blum and S. Micali, “How to generate cryptographically strong sequences of pseudorandom bits”, *SIAM J. Comput.* **13**:4 (1984), 850–864. MR 86a:68021
- [Boneh 1998] D. Boneh, “The decision Diffie–Hellman problem”, pp. 48–63 in *Algorithmic number theory* (Portland, OR, 1998), edited by J. P. Buhler, Lecture Notes in Comput. Sci. **1423**, Springer, Berlin, 1998. MR 2000k:94024 Zbl 1067.94523
- [Brugger 2008] M. F. Brugger, *Exploring the discrete logarithm with random ternary graphs*, senior thesis, Oregon State University, 2008, available at <http://hdl.handle.net/1957/8777>.
- [Brugger and Frederick 2007] M. Brugger and C. Frederick, “The discrete logarithm problem and ternary functional graphs”, *Rose-Hulman Undergraduate Mathematics Journal* **8**:2 (2007).
- [Canetti et al. 2000] R. Canetti, J. Friedlander, S. Konyagin, M. Larsen, D. Lieman, and I. Shparlinski, “On the statistical properties of Diffie–Hellman distributions”, *Israel J. Math.* **120**:part A (2000), 23–46. MR 2001k:11258 Zbl 0997.11066
- [Cloutier 2005] D. R. Cloutier, *Mapping the discrete logarithm*, senior thesis, Rose-Hulman Institute of Technology, 2005, available at <http://www.csse.rose-hulman.edu/images/docs/theses/DanielCloutier2005.pdf>.

- [Diffie and Hellman 1976] W. Diffie and M. E. Hellman, “New directions in cryptography”, *IEEE Trans. Information Theory* **IT-22**:6 (1976), 644–654. MR 55 #10141
- [Flajolet 1979] P. Flajolet, *Analyse d’algorithmes de manipulation d’arbres et de fichiers*, Ph.D. thesis, Université de Paris-Sud, Orsay, 1979.
- [Flajolet et al. 1989] P. Flajolet, B. Salvy, and P. Zimmermann, “Lambda-epsilon-omega: The 1989 cookbook”, Technical Report RR1073, Institut National de Recherche en Informatique et en Automatique, 1989, available at <http://www.inria.fr/rrrt/rr-1073.html>.
- [Flajolet et al. 1991] P. Flajolet, B. Salvy, and P. Zimmermann, “Automatic average-case analysis of algorithms”, *Theoret. Comput. Sci.* **79**:1, (Part A) (1991), 37–109. MR 92k:68049 Zbl 0768.68041
- [Flajolet et al. 1993] P. Flajolet, Z. Gao, A. Odlyzko, and B. Richmond, “The distribution of heights of binary trees and other simple trees”, *Combin. Probab. Comput.* **2**:2 (1993), 145–156. MR 94k:05061 Zbl 0795.05042
- [FO 1982] P. Flajolet and A. Odlyzko, “The average height of binary trees and other simple trees”, *J. Comput. System Sci.* **25**:2 (1982), 171–213. MR 84a:68056 Zbl 0499.68027
- [FO 1990a] P. Flajolet and A. Odlyzko, “Singularity analysis of generating functions”, *SIAM J. Discrete Math.* **3**:2 (1990), 216–240. MR 90m:05012 Zbl 0712.05004
- [FO 1990b] P. Flajolet and A. M. Odlyzko, “Random mapping statistics”, pp. 329–354 in *Advances in cryptology* (Houthalen, Belgium, 1989), edited by A. J. Menezes and S. A. Vanstone, Lecture Notes in Comput. Sci. **434**, Springer, Berlin, 1990. MR 1083961 Zbl 0747.05006
- [Gennaro 2005] R. Gennaro, “An improved pseudo-random generator based on the discrete logarithm problem”, *J. Cryptology* **18**:2 (2005), 91–110. MR 2007c:94124 Zbl 1084.68046
- [Hoffman 2009] A. Hoffman, “Statistical investigation of structure in the discrete logarithm”, *Rose-Hulman Undergrad. Math. J.* **10**:2 (2009).
- [Holden 2002] J. Holden, “Fixed points and two-cycles of the discrete logarithm”, pp. 405–415 in *Algorithmic number theory* (Sydney, 2002), edited by C. Fieker and D. R. Kohel, Lecture Notes in Comput. Sci. **2369**, Springer, Berlin, 2002. “Addenda and corrigenda” at arXiv:math.NT/0208028. MR 2005h:11277 Zbl 1058.11073
- [Holden and Moree 2004] J. Holden and P. Moree, “New conjectures and results for small cycles of the discrete logarithm”, pp. 245–254 in *High primes and misdemeanours*, edited by A. van der Poorten and A. Stein, Fields Inst. Commun. **41**, Amer. Math. Soc., Providence, RI, 2004. MR 2005d:11005 Zbl 1100.11005
- [Holden and Moree 2006] J. Holden and P. Moree, “Some heuristics and results for small cycles of the discrete logarithm”, *Math. Comp.* **75**:253 (2006), 419–449. MR 2006i:11145 Zbl 1116.11004
- [Lindle 2008] N. Lindle, *A statistical look at maps of the discrete logarithm*, senior thesis, Rose-Hulman Institute of Technology, 2008, available at <http://www.csse.rose-hulman.edu/images/docs/theses/NathanLindle2008.pdf>.
- [Mace 2009] M. L. Mace, “Discrete logarithm over composite moduli”, REU technical report, Rose-Hulman Institute of Technology, 2009, available at <http://www.rose-hulman.edu/~holden/REU/Reports/mace.pdf>.
- [Mishna 2004] M. Mishna, “How to use attribute grammars with ease and pleasure”, 2004, available at <http://www.math.sfu.ca/~mmishna/Publications/cook2.ps>.
- [Pollard 1978] J. M. Pollard, “Monte Carlo methods for index computation (mod p)”, *Math. Comp.* **32**:143 (1978), 918–924. MR 58 #10684 Zbl 0382.10001
- [Rivest et al. 1978] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, *Comm. ACM* **21**:2 (1978), 120–126. MR 83m:94003 Zbl 0368.94005

[Salvy 1997] B. Salvy, “Pollard’s rho algorithm”, worksheet, 1997, available at <http://algo.inria.fr/libraries/autocomb/pollard-html/pollard1.html>.

[Shepp and Lloyd 1966] L. A. Shepp and S. P. Lloyd, “Ordered cycle lengths in a random permutation”, *Trans. Amer. Math. Soc.* **121** (1966), 340–357. MR 33 #3320 Zbl 0156.18705

Received: 2009-10-17 Revised: Accepted: 2010-06-19

Daniel.R.Cloutier@alumni.rose-hulman.edu

*Rose-Hulman Institute of Technology,
Terre Haute, IN 47803, United States*

holden@rose-hulman.edu

*Rose-Hulman Institute of Technology,
Department of Mathematics, CM #125, 5500 Wabash Ave.,
Terre Haute, IN 47803, United States
<http://www.rose-hulman.edu/~holden>*

involve

pjm.math.berkeley.edu/involve

EDITORS

MANAGING EDITOR

Kenneth S. Berenhaut, Wake Forest University, USA, berenhks@wfu.edu

BOARD OF EDITORS

John V. Baxley	Wake Forest University, NC, USA baxley@wfu.edu	Chi-Kwong Li	College of William and Mary, USA ckli@math.wm.edu
Arthur T. Benjamin	Harvey Mudd College, USA benjamin@hmc.edu	Robert B. Lund	Clemson University, USA lund@clemson.edu
Martin Bohner	Missouri U of Science and Technology, USA bohner@mst.edu	Gaven J. Martin	Massey University, New Zealand g.j.martin@massey.ac.nz
Nigel Boston	University of Wisconsin, USA boston@math.wisc.edu	Mary Meyer	Colorado State University, USA meyer@stat.colostate.edu
Amarjit S. Budhiraja	U of North Carolina, Chapel Hill, USA budhiraj@email.unc.edu	Emil Minchev	Ruse, Bulgaria eminchev@hotmail.com
Pietro Cerone	Victoria University, Australia pietro.cerone@vu.edu.au	Frank Morgan	Williams College, USA frank.morgan@williams.edu
Scott Chapman	Sam Houston State University, USA scott.chapman@shsu.edu	Mohammad Sal Moslehian	Ferdowsi University of Mashhad, Iran moslehian@ferdowsi.um.ac.ir
Jem N. Corcoran	University of Colorado, USA corcoran@colorado.edu	Zuhair Nashed	University of Central Florida, USA znashed@mail.ucf.edu
Michael Dorff	Brigham Young University, USA mdorff@math.byu.edu	Ken Ono	University of Wisconsin, USA ono@math.wisc.edu
Sever S. Dragomir	Victoria University, Australia sever@matilda.vu.edu.au	Joseph O'Rourke	Smith College, USA orourke@cs.smith.edu
Behrouz Emamizadeh	The Petroleum Institute, UAE bemamizadeh@pi.ac.ae	Yuval Peres	Microsoft Research, USA peres@microsoft.com
Errin W. Fulp	Wake Forest University, USA fulp@wfu.edu	Y.-F. S. Pétermann	Université de Genève, Switzerland petermann@math.unige.ch
Andrew Granville	Université Montréal, Canada andrew@dms.umontreal.ca	Robert J. Plemmons	Wake Forest University, USA plemmons@wfu.edu
Jerrold Griggs	University of South Carolina, USA griggs@math.sc.edu	Carl B. Pomerance	Dartmouth College, USA carl.pomerance@dartmouth.edu
Ron Gould	Emory University, USA rg@mathcs.emory.edu	Bjorn Poonen	UC Berkeley, USA poonen@math.berkeley.edu
Sat Gupta	U of North Carolina, Greensboro, USA sgupta@uncg.edu	James Propp	U Mass Lowell, USA jpropp@cs.uml.edu
Jim Haglund	University of Pennsylvania, USA jhaglund@math.upenn.edu	József H. Przytycki	George Washington University, USA przytyck@gwu.edu
Johnny Henderson	Baylor University, USA johnny_henderson@baylor.edu	Richard Rebarber	University of Nebraska, USA rrebarbe@math.unl.edu
Natalia Hritonenko	Prairie View A&M University, USA nahritonenko@pvamu.edu	Robert W. Robinson	University of Georgia, USA rwr@cs.uga.edu
Charles R. Johnson	College of William and Mary, USA crjohnso@math.wm.edu	Filip Saidak	U of North Carolina, Greensboro, USA f.saidak@uncg.edu
Karen Kafadar	University of Colorado, USA karen.kafadar@cudenver.edu	Andrew J. Sterge	Honorary Editor andy@ajsterge.com
K. B. Kulasekera	Clemson University, USA kk@ces.clemson.edu	Ann Trenk	Wellesley College, USA atrenk@wellesley.edu
Gerry Ladas	University of Rhode Island, USA gladas@math.uri.edu	Ravi Vakil	Stanford University, USA vakil@math.stanford.edu
David Larson	Texas A&M University, USA larson@math.tamu.edu	Ram U. Verma	University of Toledo, USA verma99@msn.com
Suzanne Lenhart	University of Tennessee, USA lenhart@math.utk.edu	John C. Wierman	Johns Hopkins University, USA wierman@jhu.edu

PRODUCTION

Silvio Levy, Scientific Editor

Sheila Newbery, Senior Production Editor

Cover design: ©2008 Alex Scorpan


See inside back cover or <http://pjm.math.berkeley.edu/involve> for submission instructions.

The subscription price for 2010 is US \$100/year for the electronic version, and \$120/year (+\$20 shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94704-3840, USA.

Involve (ISSN 1944-4184 electronic, 1944-4176 printed) at Mathematical Sciences Publishers, Department of Mathematics, University of California, Berkeley, CA 94720-3840 is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

Involve peer review and production are managed by EditFLOW™ from Mathematical Sciences Publishers.

PUBLISHED BY

 mathematical sciences publishers

<http://www.mathscipub.org>

A NON-PROFIT CORPORATION

Typeset in L^AT_EX

Copyright ©2010 by Mathematical Sciences Publishers

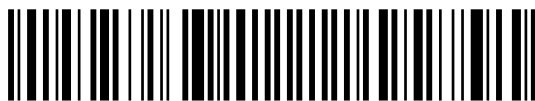
involve

2010

vol. 3

no. 2

Recursive sequences and polynomial congruences J. LARRY LEHMAN AND CHRISTOPHER TRIOLA	129
The Gram determinant for plane curves JÓZEF H. PRZYTYCKI AND XIAOQI ZHU	149
The cardinality of the value sets modulo n of $x^2 + x^{-2}$ and $x^2 + y^2$ SARA HANRAHAN AND MIZAN KHAN	171
Minimal k -rankings for prism graphs JUAN ORTIZ, ANDREW ZEMKE, HALA KING, DARREN NARAYAN AND MIRKO HORŇÁK	183
An unresolved analogue of the Littlewood Conjecture CLARICE FEROLITO	191
Mapping the discrete logarithm DANIEL CLOUTIER AND JOSHUA HOLDEN	197
Linear dependency for the difference in exponential regression INDIKA SATHISH AND DIAWARA NOROU	215
The probability of relatively prime polynomials in $\mathbb{Z}_{p^k}[x]$ THOMAS R. HAGEDORN AND JEFFREY HATLEY	223
\mathbb{G} -planar abelian groups ANDREA DEWITT, JILLIAN HAMILTON, ALYS RODRIGUEZ AND JENNIFER DANIEL	233



1944-4176(2010)3:2;1-F