MAPPING THE DISCRETE LOGARITHM: ABSTRACT

DANIEL R. CLOUTIER AND JOSHUA HOLDEN

The discrete logarithm is a problem that surfaces frequently in the field of cryptography as a result of using the transformation $g^a \mod n$. This paper focuses on a prime modulus, p, for which it is shown that the basic structure of the functional graph is largely dependent on an interaction between g and $p-1$. In fact, there are precisely as many different functional graph structures as there are divisors of $p-1$. This paper extracts two of these 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 suggests that both the permutations and binary functional graphs correspond well to the theoretical data which provides motivation to extend this to larger divisors of $p-1$ and study the impact this forced structure has on the many cryptographic algorithms that rely on the discrete logarithm for their security. This is especially applicable to those algorithms that require a "safe" prime ($p = 2q+1$, where q is prime) modulus since all non-trivial functional graphs generated using a safe prime modulus can be analyzed by the framework presented here.

In this work, we will restrict the values of n to primes and examine mappings of the form $x \mapsto g^x \mod p$, where p is a prime modulus. 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. An m-ary functional graph to be a graph where each node has in-degree of exactly zero or m. There are a number of statistics of interest derived from functional graphs. For permutations these are fairly well known. We begin our work by proving some results on the asymptotic form of these statistics for random binary functional graphs.

We then generate experimental data for the parameters described by these theoretical predictions. The method of data collection was straightforward. A prime was chosen as the modulus and then for each $g \in \{1, 2, 3, ..., p-1\}$, the corresponding map or permutation was generated. The results were then computed as averages 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.

The statistics derived from the binary functional graphs and the error when compared to the theoretical results can be found in Table 1. The observations seem to support the claim that binary functional graphs induced by exponentiation modulo a prime behave in the same fashion as random binary functional graphs.

	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 1. The observed results for the three primes over all binary functional graphs generated and the corresponding percent errors.