ON COMPUTING LOGARITHMS OVER $GF(2^p)$

TORE HERLESTAM and ROLF JOHANNESSON

Abstract.

In this paper we present a new, heuristic method for computing logarithms over $GF(2^p)$. When $2^p - 1$ is a Mersenne prime $\leq 2^{231} - 1$ it works in very short running times on a general purpose computer. It is based on the interdependent relations

\[ f_{rs}(t) = t^{-2^r} f(t)^{2^s} \]

and

\[ \log f_{rs}(t) = -2^r + 2^s \log f(t), \]

where $f$ and $f_{rs}$ are polynomials over $GF(2)$.

Its cryptographic significance is discussed and it can be considered as an attempt to swindle MITRE Corporation which reportedly is using a public key distribution system, based on the presumed difficulty in computing logarithms over $GF(2^{127})$.

I. Introduction.

In [1] Diffie and Hellman proposed a public key distribution system based on the exponential function $x \rightarrow b^x$ over $GF(p)$, where $p$ is a very large prime and $b$ generates the multiplicative group $GF(p)^*$. The idea was further developed by Pohlig and Hellman in [2] and by Pohlig in [3]. They investigated an algorithm for computing logarithms over $GF(p)$, originating from Roland Silver, and proved it to be efficient when $p - 1$ consists of small prime factors only, but computationally infeasible when $p - 1$ contains at least one large prime factor. Therefore it was concluded that exponentiation over $GF(p)$ may be considered one-way in this latter case.

Since the arithmetic in $GF(p)$ is not very well suited to most computers, a variant is proposed in [2], [3] and [4]. Let $2^p - 1$ be a large Mersenne prime, e.g. $2^{127} - 1$ or $2^{521} - 1$, and let $b$ be a generator of the group $GF(2^p)^*$. Then the exponentiation $x \rightarrow b^x$ over $GF(2^p)$ is suggested to be one-way for the same reason as above. The apparent advantages are that the arithmetic in $GF(2^p)$ is easy to implement on a computer and that $b$ can be chosen arbitrarily, since $GF(2^p)^*$ is a prime order group.

In order to test the security of this public key distribution system we have
started an investigation of heuristic methods for computing logarithms over $GF(2^p)$. Below we shall describe a first result from this investigation.

In Sections II and III some relevant algebraic preliminaries are given. After a description of the algorithm in Section IV, some numerical results are commented in Section V. Section VI concerns the complexity issue, and the concluding Section VII deals with applications and future possible transfers of the technique to similar problems.

II. The different faces of $GF(2^p)$.

By definition $GF(2^p)$ is a finite field with $q = 2^p$ elements. Its prime field is $GF(2)$, i.e. it has characteristic 2, which implies that

$$(a + b)^2 = a^2 + b^2$$

within the field.

In particular, $GF(q)$ is a vector space $V$ over $GF(2)$. From $(a+b)'=a^2+b^2$ it follows that

$$S: V \ni x \mapsto x^2 \in V$$

is a linear transformation on $V$. By iteration, $S^2, S^3, S^4, \ldots$ are also linear transformations on $V$.

The elements of $GF(q)$ are the roots of the equation $x^q=x$ over $GF(2)$. This implies that $S^p=I$ and that $I, S, S^2, \ldots, S^{p-1}$ are linearly independent, i.e. constitute a basis for a $p$-dimensional subspace of the vector space $L(V, V)$, consisting of all linear transformations on $V$.

A most convenient way of implementing $GF(q)$ is to choose a prime polynomial $p(t)$ of degree $p$ over $GF(2)$ and then form the residue class ring $A = GF(2)[t]/p(t)GF(2)[t]$, which is isomorphic to $GF(q)$. Preferably, $A$ is represented as the set of all polynomials of degree $<p$ over $GF(2)$, with addition modulo 2 componentwise and multiplication modulo $p(t)$.

Now

$$T: A \ni f(t) \mapsto tf(t) + a(t)p(t) \in A,$$

where $tf(t) + a(t)p(t)$ denotes the lowest degree residue of $tf(t)$ modulo $p(t)$, is a linear transformation on the vector space $A$. Since $p(0) \neq 0$, we have $p(t) = 1 + tq(t)$, i.e. $q(t) = t^{-1}$ modulo $p(t)$, and so $T$ is one-to-one, with the inverse

$$T^{-1}: A \ni f(t) \mapsto q(t)f(t) + b(t)p(t) \in A,$$

which can also conventionally be written as $T^{-1} = q(T)$. Since $p(t)$ divides $t^q-t$, we have $T^q = T$. Provided $p(t)$ is a maximum-length polynomial, all the powers $T^{-1}, T^{-2}, T^{-4}, \ldots, T^{-2^{r-1}}$ are different, and so are the transformations

$$T^{-2^r}S^r, 0 \leq r, s \leq p-1.$$