A Note on Discrete Logarithms with Special Structure

Rafi Heiman
Bellcore, 445 South Street, Morristown NJ 07962, USA

Abstract

Many cryptographic systems assume the computational difficulty of the discrete logarithm (DL) problem. In order to accelerate such practical systems, it was proposed to use logarithms of special structure, such as small Hamming weight. How difficult is the underlying restricted DL problem? By rephrasing Shanks' method we provide a close to square-root algorithm for such problems.

1 Introduction

Many cryptographic systems assume the computational difficulty of the discrete logarithm (DL) problem which is defined for a prime modulo as follows. Let p be a prime number, let Z_p be the additive group modulo p of all integers between 0 and $p-1$, let Z_p^* be the multiplicative group of order $p-1$ of the integers between 1 and $p-1$ modulo p, and let g be a generator of Z_p^*. The discrete logarithm (DL) problem (modulo p) is to compute $x \in Z_{p-1}$ for a given $y \in Z_p^*$, such that $g^x = y \pmod p$.

In practical cryptographic systems that are based on the difficulty of DL, logarithms of special structure are sometimes used. The idea is to choose a subset $X \subseteq Z_{p-1}$ of some special structure, which makes the use of the system (namely, the exponentiation) more efficient. Examples include the suggestion of Agnew et. al. [AMOV91] in which the special structure is small Hamming weight, and the suggestion of Yacobi [Yac90], in which the special structure is Lempel-Ziv compressibility. This however defines a restricted DL problem: it is guaranteed that the solution x of $g^x = y \pmod p$ satisfies $x \in X$. While for the general DL problem sub-exponential algorithms are known (see [LL90]), it is not clear whether one can compute restricted discrete logarithms faster than exhaustive search in the special structured set, X.

In this note we show a close to square-root algorithm for several such restricted DL problems, including that of [AMOV91]. This algorithm was independently noticed by Odlyzko [Odl]. We do not know of a similar result regarding the special structure suggested in [Yac90].

R.A. Rueppel (Ed.): Advances in Cryptology - EUROCRYPT '92, LNCS 658, pp. 454-457, 1993.

2 Rephrasing Shanks' Method

Shanks' method for computing DL (see [Knu73], pp. 9, 575-576) can be rephrased in the following slightly more general form. Assume, as above, that a subset $X \subseteq Z_{p-1}$ is known such that the required solution of $g^x = y \pmod p$ satisfies $x \in X$. Choose 'small' sets $A, B \subset Z_{p-1}$ such that $X \subseteq A+B$ where the sum of the sets is defined by $A+B = \{a+b \pmod{p-1} : a \in A,\ b \in B\}$. Rewrite $g^x = y \pmod p$ as $g^{a+b} = y \pmod p$ or as $g^a = yg^{-b} \pmod p$. Create the lists $\{g^a \pmod p\}_{a \in A}$ and $\{yg^{-b} \pmod p\}_{b \in B}$, sort (or hash) them, and find a common member, $g^a = yg^{-b} \pmod p$. The corresponding a and b define the required solution, $x = a + b \pmod{p-1}$.

The method has time complexity $O(s \log(s))$ (or $O(s)$ if hashing is used) and space complexity $O(s)$ where $s = \max\{|A|, |B|\}$. Clearly, $s \geq \sqrt{|X|}$ for any choice of A and B that satisfies $X \subseteq A + B$.

3 Applications

Many sets X of special structured logarithms can be decomposed as above into sets A and B of sizes not much greater than $\sqrt{|X|}$. Some examples follow. In these examples $n = \lceil \log p \rceil$ denotes the number of bits in p, t is some number between 0 and n, $[n]$ denotes the set $\{0, 1, ..., n-1\}$, and $||x||$ denotes the Hamming weight of a number x, that is, the number of 1's in the binary representation of X. Also, $X_{=t} = \{x \in Z_{p-1} : ||x|| = t\}$ denotes the set of logarithms with Hamming weight exactly t, and $X_{\leq t} = \{x \in Z_{p-1} : ||x|| \leq t\}$ denotes the set of logarithms with Hamming weight at most t.

Examples:

1. For $X = X_{=t}$ with $t < \frac{n}{2}$ choose $A = B = X_{=\frac{t}{2}}$.

2. For $X_{\leq t}$ with $t < \frac{n}{2}$ guess the Hamming weight of x and solve as above. Alternatively, choose $A = B = X_{\leq \frac{t}{2}}$.

3. For $X = X_{=t}$ with $t > \frac{n}{2}$ a cosmetic change in the method is convenient: choose $A = X_{=\frac{n+t}{2}}$ and $B = X_{=\frac{n-t}{2}}$, note that $X \subseteq A \setminus B$, and solve $g^{a-b} = y$, that is $g^a = yg^b \pmod p$.

4. An interesting structure is when some subset $I \subseteq [n]$ is known to project every $x \in X$ in the same way. Namely, there are some fixed values $c_i \in \{0, 1\}$ for $i \in I$, such that every $x = \sum_{i=0}^{n-1} x_i 2^i \in X$ satisfies $x_i = c_i$ for all $i \in I$. For this case, pick $I_A \subseteq [n]$ and $I_B \subseteq [n]$ of (roughly) the same size which are disjoint and satisfy $I_A \cup I_B = [n] \setminus I$. Then choose $A = X \cap \{x : \forall i \in I_A,\ x_i = 0\}$ and $B = \{x : \forall i \in I \cup I_B,\ x_i = 0\}$.

   We would like to mention that this example is a generalization of the set-up for Pollard's λ-method for catching kangaroos [Pol78], in which X is some segment

within Z_{p-1}, or an arithmetic sequence in Z_{p-1}. A segment that starts at some multiple of 2^{i_0} and is of length 2^{i_0} corresponds to this example with I consisting of all but the i_0 least significant bits, $I = \{i : i_0 \leq i < n\}$. An arithmetic sequence with jumps of size 2^{i_0}, that starts at some multiple of 2^{i_1}, and that is of length $2^{i_1 - i_0}$, corresponds to this example with I consisting of both the i_0 least significant bits and the $n - i_1$ most significant bits, $I = \{i : 0 \leq i < i_0\} \cup \{i : i_1 \leq i < n\}$.

5. If X has restricted Hamming weight and in addition is restricted by some subset $I \subset [n]$ with fixed values c_i for $i \in I$ as above, the decomposition is easily obtained by 'merging' the two corresponding decompositions above.

Complexity:
Note that the complexity of example 4 is exactly the square-root of the size of the structured set, X. The complexity of small Hamming weight DL is worse than this, say $|X|^\beta$ for some $\beta > 1/2$. We compute the value β for example 1. In this example β satisfies $\binom{n}{t}^\beta = \binom{n}{t/2}$, where as above, n is the number of bits in the logarithms and t is their Hamming weight. As an example, for the concrete values $n = 512$ and $t = 50$ we have $\beta = .60$. Asymptotically, we look at a fixed ratio $\alpha = t/n$. By Stirling formula, $n! = \Theta(\frac{1}{\sqrt{n}} \cdot (n/e)^n)$ we have

$$\binom{n}{\alpha n} = \Theta(\frac{\sqrt{n}}{\alpha^{\alpha n}(1-\alpha)^{(1-\alpha)n}}).$$

Thus,

$$\beta = \frac{\log\binom{n}{(\alpha/2)n}}{\log\binom{n}{\alpha n}} = \frac{H(\alpha/2)}{H(\alpha)} + o(1),$$

where H is the entropy function, $H(x) = -x\log(x) - (1-x)\log(1-x)$, and $o()$ is the 'little-oh' notation. For $\alpha = 1/4$, $1/10$ and $1/50$, β approaches .67, .61 and .57, respectively. Same asymptotical values of β are valid for examples 2, 3 and 5, where for the latter example, n should be replaced by $n - |I|$ and t should be replaced by the number of 1's that are allowed among the indices not belonging to I.

4 Concluding Remarks

This method can be clearly used over any finite group, e.g., Z_n^* for composite n, $GF(2^n)$ with the multiplication of polynomials as the group operator, or elliptic curve groups. The Agnew et. al. system [AMOV91] uses small Hamming weight of the secret exponent in $GF(2^n)$. This note suggests that not only $\binom{n}{t}$ should be large enough to prevent exhaustive search, but already $\binom{n}{t/2}$ should be 'large'. We would like to emphasize that the actual parameters chosen for the implementation of that system [Ros89], do seem to be so.

An interesting structured logarithm-set for accelerating exponentiations for which we do not know a better than exhaustive search algorithm is that of Lempel-Ziv compressibility, suggested by Yacobi [Yac90].

It would also be very interesting to obtain a faster than square-root attack for any of the structured DL problems mentioned above.

Acknowledgement

I thank Yacov Yacobi for asking the question, and I thank him as well as Stuart Haber and Arjen Lenstra for their comments.

References

[AMOV91] G.B. Agnew, R.C. Mullin, I.M. Onyszchuk, and S.A. Vanstone. An implementation for a fast public-key cryptosystem. *Journal of Cryptology*, 3(2):63–79, 1991.

[Knu73] D.E. Knuth. *The art of computer programming, vol. 3: sorting and searching.* Addison-Wesley, Reading, Mass., 1973.

[LL90] A.K. Lenstra and H.W. Lenstra. *Algorithms in Number Theory*, volume A, chapter 12, pages 673–716. MIT press, 1990.

[Odl] A. Odlyzko. Private communication.

[Pol78] J.M. Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of computation*, 32(143):918–924, July 1978.

[Ros89] T. Rosati. A high speed data encryption processor for public key cryptography. In *IEEE Custom Integrated Circuits Conference*, pages 12.3.1–12.3.5, May 1989.

[Yac90] Y. Yacobi. Discrete-log with compressible exponents. In *Advances in Cryptology, Crypto90*, pages 639–643, 1990.