Discrete logarithm hash function that is collision free and one way

J.K. Gibson

Indexing terms: Cryptography, Number theory, Hash functions, Discrete logarithm

Abstract: For suitable composite modulus n and suitable base a, the discrete logarithm hash function $x \to a^x \bmod n$ is collision free and one way if factoring n is hard. Further results on the relation between the discrete logarithm problem and factoring are given. Some complexity theory issues are considered.

1 Introduction

A hash function is a map $f: \{0, 1\}^s \to \{0, 1\}^t$ where $s > t$. A collision for f is a pair of unequal $x, y \in \{0, 1\}^s$ with $f(x) = f(y)$. f is collision free if finding a collision for f is hard, and f is one way if f is easy to compute but hard to invert. Hash functions that are one way and collision free are used in cryptography for the construction of digital signature schemes [4].

The discrete logarithm (DL) problem with modulus n and base a is that of solving $w = a^x \bmod n$ for the integer x when the integers a, n, w are given, and in general is a hard problem. (Integers will always be nonnegative).

The main purpose of this paper is to examine the conditions under which the DL problem with a composite modulus can be used to obtain a hash function that is collision free and one way. In doing so we give results that deepen our understanding of the relation between the DL problem and factoring. These results have a number theoretic interest in their own right, and given the increasing use of the DL problem in cryptography, References 2 and 12, they may well turn out to have a cryptographic value. A second purpose is to discuss some complexity theory issues, to clarify what hard should mean in the definitions of collision free and one way when these terms are used in the context of cryptographic hash functions. These issues are important, since some definitions of one way given in the literature, e.g. Reference 14, are not appropriate for hash functions used in signature schemes.

Accordingly we define a DL hash function with modulus n and base a to be a function $f: S \to T$ given by $f(x) = a^x \bmod n$, where n is a t-bit integer, a is an integer coprime to n, S is the set of integers $< 2^s$ for some $s > t$, and T is the set of integers $< 2^t$. Binary strings may be used as inputs to f by viewing them as integers and avoiding trivial collisions by prefixing leading zeros with a '1'.

Computing a DL hash function is slow compared to block cipher hashing [4], taking $O(s)$ t-bit modular multiplications for an s-bit input and t-bit output, but this is just about practical, and no other practical hash function yet proposed can be proven collision free and one way.

Paper 8363E (C2), first received 3rd September 1990 and in revised form 29th July 1991

The author is with the Department of Computer Science, Birbeck College, Malet Street, London WC1E 7HX, United Kingdom

We use the following notation. Z_n denotes the integers modulo n, and Z_n^* the multiplicative group of those $a \in Z_n$ which are coprime to n. Thus, the order of $a \in Z_n^*$ is the least positive r with $a^r = 1 \bmod n$.

Our results are summarised as follows. We will shortly define a DL-strong integer n, and a DL-strong base a for n. Accepting this terminology for the moment, let f be a DL hash function with modulus n and base a. Then

(*a*) If n is a DL-strong integer then almost all $a \in Z_n^*$ are DL-strong bases for n, and the remaining ones either permit the easy factorisation of n or have very small order in Z_n^*, when f is easy to invert.

(*b*) If n is a DL-strong integer, and a is a DL-strong base for n, then knowledge of a collision for f permits the easy factorisation of n.

(*c*) If F is a hash function family whose instances are DL hash functions with DL-strong modulus, then provided the set of moduli of F is hard to factor, F is collision free, and one way in two different senses.

(*d*) Almost all n for which a factor cannot be found easily by the Pollard $p - 1$ method [10] have the property that for almost all $a \in Z_n^*$, collisions for f reveal a factor of n. This implies that (*c*) remains true even if the moduli of F are not DL-strong.

Results (*a*) and (*b*) give conditions for f to be collision free, and are nonasymptotic in nature, in spite of the 'almost all' in (*a*), because (*b*) throws the complexity theory considerations onto the factorisation problem.

Proving that a function is one way cannot be done without a complexity theory setting, so this is introduced before proving result (*c*). We define a hash function family F and say what it means for F to be collision free and one way. We show that under suitable conditions collision free implies one way, and in fact give a lemma relating several conditions that can be imposed on hash function families which proves rather more. We define a hard to factor set of integers, and use our lemma to prove result (*c*).

Result (*d*) fills a gap in previous knowledge, by showing effectively that if n is hard to factor, then the DL problem with modulus n is hard for almost all bases.

The following work is related to ours. Bach [1] shows that for any integer n, and for at least half the members $a \in Z_n^*$, finding a collision for f permits the easy factorisation of n. McCurley [8] and Shmuely [11] consider the Diffie-Hellman (DH) problem [5] with modulus n and base a, which is certainly easy if the DL problem with the same modulus and base is easy. They show that if n is suitably chosen and a has odd order in Z_n^* then the DH problem is hard if factoring such n is hard. However, at most half the members of Z_n^* have this property.

Damgard [3] has considered the question of when collision free implies one way.

2 Number theory preliminaries

For an integer n, the Euler function $\phi(n)$ is defined to be the order of the group Z_n^*, and the Carmichael function $\lambda(n)$ is defined to be the largest order that any member of Z_n^* can have. For odd n they can be calculated from the prime factorisation of n as follows. For an odd prime p, $\phi(p^e) = \lambda(p^e) = p^{e-1}(p-1)$. In particular, if $a \in Z_p^*$ then $a^{p-1} = 1 \bmod p$. If P and Q are coprime and odd, then $\phi(PQ) = \phi(P)\phi(Q)$, and $\lambda(PQ) = \mathrm{lcm}(\phi(P), \phi(Q))$. (We use lcm and gcd to denote least common multiple and greatest common divisor.)

The following result from Miller [9] and Bach [1] shows that if n is a t-bit odd integer with at least two distinct prime factors, then an s-bit multiple of $\lambda(n)$, or of $\phi(n)$, can be used to find a factor of n with probability at least 0.5 in at most $2s$ modular multiplications and one gcd computation of t-bit integers.

Lemma 1. (Miller–Bach)
Let the odd integer n have at least two distinct prime factors, and let $x \neq 0$ be a multiple of $\lambda(n)$. Pick $a \in Z_n^*$. Write $x = 2^h z$, z odd. Define $z_0 = a^z \bmod n$, and $z_i = z_{i-1}^2 \bmod n$, $i = 1, 2, \ldots$. Let r be minimal with $z_r = 1$. For at least half the choices of a, $a^{\lambda(n)/2} \neq \pm 1 \bmod n$, and for these choices of a, $r \neq 0$ and $\gcd(z_{r-1} - 1, n)$ is a non-trivial factor of n.

3 DL-strong integers and bases

Definitions: A DL-strong t-bit integer n is a product of odd primes p, q for which $p - 1 = 2up_1$, $q - 1 = 2vq_1$, where p_1, q_1 are odd primes, p, q, p_1, q_1, are large and distinct, and u, v are small. A DL-strong base for n is an $a \in Z_n^*$ whose order is a multiple of $p_1 q_1$. A strong DL hash function is one with DL-strong modulus and base. We call n DL-superstrong if p and q are congruent to 3 mod 4, and $p + 1$ and $q + 1$ both have a large prime factor.

For theorems 1 and 2 small and large can be left undefined, but the significance of these theorems can be appreciated by thinking of small as < 1000 and large as $> 2^{64}$. For theorem 3 small/large mean polynomially/nonpolynomially bounded in t.

DL-superstrong integers are widely believed to be hard to factor. The condition on $p + 1$ and $q + 1$ is to avoid factorisation by the Williams $p + 1$ method [13].

DL-strong bases for a DL-strong integer n are those that have very large order in Z_n^*. The following theorem shows that almost all $a \in Z_n^*$ are DL-strong bases for n, and those that are not either reveal the factors of n, or have very small order in Z_n^*, when the DL hash function with modulus n and base a is easy to invert.

Theorem 1: Let n be a product of odd primes p, q for which $p - 1 = 2up_1$, $q - 1 = 2vq_1$, where p_1, q_1 are odd primes, and p, q, p_1, q_1 are distinct and coprime to u, v. Let $a \in Z_n^*$, and let $d = \gcd(u, v)$. Then

(*a*) The proportion of members of Z_n^* whose order is not a multiple of $p_1 q_1$ is $1/p_1 + 1/q_1 - 1/p_1 q_1$.

(*b*) If the order of a is not a multiple of $p_1 q_1$ then either one of $\gcd(a^{2u} - 1, n)$, $\gcd(a^{2v} - 1, n)$, $\gcd(a^d - 1, n)$ is a nontrivial factor of n, or $a^d = \pm 1 \bmod n$.

Proof:

(*a*) This follows from the primary decomposition theorem for abelian groups [7], noting that Z_n^* is the direct product of a group of order $4uv$ and two cyclic groups of orders p_1 and q_1.

(*b*) First, $a^{\mathrm{lcm}(p-1, q-1)} = a^{2uvp_1q_1/d} = 1 \bmod n$. Suppose the order of a is not a multiple of p_1. Then $a^{2uvq_1/d} = 1 \bmod n$, and since vq_1/d is coprime to $p - 1$ this means $a^{2u} = 1 \bmod p$. On the other hand if the order of a is a multiple of q_1 then $a^{2u} \neq 1 \bmod n$. Thus, if the order of a is a multiple of q_1 but not of p_1 then $\gcd(a^{2u} - 1, n) = p$, and likewise if the order of a is a multiple of p_1 but not of q_1 then $\gcd(a^{2v} - 1, n) = q$. If the order of a is not a multiple of either p_1 or q_1 then either one of $\gcd(a^{2u} - 1, n)$, $\gcd(a^{2v} - 1, n)$ is a factor of n, or else $a^{2u} = a^{2v} = 1 \bmod n$. In the latter case $a^{2d} = 1 \bmod n$, and unless $a^d = \pm 1 \bmod n$, this means $\gcd(a^d - 1, n)$ is a factor of n.

4 Strong DL hash function is collision free

To show that a collision for the DL hash function with modulus n and base a can be used to factor n, it is sufficient to show that knowledge of a nonzero x with $a^x = 1 \bmod n$ permits the easy factorisation of n. For DL-strong moduli and bases this is guaranteed by the following theorem.

Theorem 2: Let n be a t-bit product of distinct odd primes p, q for which $p - 1 = 2up_1$ and $q - 1 = 2vq_1$, where p_1, q_1 are distinct odd primes. Let $a \in Z_n^*$ and suppose the order of a is a multiple of $p_1 q_1$. Let $x \neq 0$ satisfy $a^x = 1 \bmod n$, and suppose $4uvx$ is s-bit. Then there is an algorithm with input a, n, x that outputs the factors of n with probability at least 0.5 in at most $2uvs$ modular multiplications and one gcd computation of t-bit integers.

Proof: Since the order of a is a multiple of $p_1 q_1$ it follows that $4uvx$ is a multiple of $(p - 1)(q - 1)$, which is $\phi(n)$. Thus a nonzero multiple y of $\phi(n)$ can be found by considering kx for at most uv values of k, y will have at most s bits, and the result follows from the Miller-Bach lemma.

5 Two definitions of one way

If Alice is a cryptographer wanting to use a DL hash function f to hash binary strings then she will want to know that if she chooses an input to f, an adversary given the resulting output will almost always find it hard to compute any corresponding input. However if Bob is a number theorist wanting to know whether the DL problem with the modulus and base of f is hard he will want to know that if he chooses an output from f then it will almost always be hard to compute any corresponding input. Alice's requirements lead to the standard definition of one way, but Bob's lead to a different concept which we call output one way. It turns out that for the DL hash function the two concepts of one way coincide, and in future applications Alice may be able to make use of this fact. Note that Alice's adversary should fail for almost all inputs to f. The definition of one way given by Yao [14] would require only that failure occurs for a significant proportion of inputs, and that is clearly unacceptable if f is used as part of a signature scheme [4].

We will assume that Alice and Bob make their choices using a uniform probability distribution, and we will accordingly use the term nonnegligible in the following way. If $\{S_m\}$, $\{T_m\}$, $m = 1, 2, \ldots$, are infinite families of finite sets, then when we say T_m is a nonnegligible subset of S_m we mean there is a polynomial P such that for each m, T_m consists of a fraction $> 1/P(m)$ of the members of S_m, and we will refer to this fraction as being nonnegligible. By 'almost all' we will mean all but a negligible fraction.

6 Hash function families

The following definition of a hash function family takes its cue from one given by Damgard [3], but differs from his in a number of respects. It is followed by definitions of six conditions (a)–(f) that can be imposed on such families, and a lemma relating these conditions. Algorithms may be probabalistic.

We will use the following notation. If $f: S \to T$ is a function, and J is a subset of T, then $f^{-1}(J)$ denotes the inverse image of J under f. For any finite set X, $|X|$ denotes the number of members of X.

Definition: A hash function family F is an infinite family $\{F_m\}$ of finite sets, $m = 1, 2, \ldots$, and two functions s, $t: N \to N$, polynomially bounded both above and below, with $s(m) > t(m)$, $m > m_0$. Here N denotes the natural numbers. A member of F_m is a function $f: S \to T$, where $S = \{0, 1\}^{s(m)}$, $T = \{0, 1\}^{t(m)}$. We refer to f as an instance of F of size m. We include in the definition that $|F_m|$ is not polynomially bounded in m, but that there are polynomial in m algorithms both to select polynomially in m many instances of size m, and to compute an instance of size m. We also impose the condition that for almost all instances $f: S \to T$ of size m, $|f(S)|$ is not polynomially bounded in m. ('Almost all' outputs of f would not make sense otherwise.)

(*a*) F is collision free if there is no polynomial in m algorithm to find collisions for F that succeeds for a nonnegligible proportion of instances of F of size m.

(*b*) F has many collisions if almost all instances $f: S \to T$ of F of size m have the property that for almost all $x \in S$ there is a $y \in S$, $y \neq x$, with $f(x) = f(y)$.

(*c*) F is one way if there is no polynomial in m algorithm to invert F which for a nonnegligible proportion of instances $f: S \to T$ of F of size m succeeds on the images under f of a nonnegligible subset of S.

(*d*) F is output one way if there is no polynomial in m algorithm to invert F which for a nonnegligible proportion of instances $f: S \to T$ of F of size m succeeds on a nonnegligible subset of $f(S)$.

(*e*) F is quasiperiodic if for almost all instances $f: S \to T$ of F of size m there is an $r > 1$ such that f is an r: 1 map from a nonnegligible subset of S onto $f(S)$.

(*f*) F preserves nonnegligibility if for almost all instances $f: S \to T$ of F of size m, the inverse image under f of a nonnegligible subset of $f(S)$ is a nonnegligible subset of S.

Lemma 2:

(i) Collision free + many collisions ⇒ one way.

(ii) One way + preserves nonnegligibility ⇒ output one way.

(iii) Quasiperiodic ⇒ preserves nonnegligibility + many collisions.

Proof: Let F be a hash function family, and $f: S \to T$ be an instance of F of size m.

(i) Choose $x \in S$ uniformly at random and compute $z = f(x)$. If F is not one way there is a polynomial in m algorithm to invert F which, with nonnegligible probability, finds $y \in S$ with $z = f(y)$. If also F has many collisions then $x \neq y$ with probability at least almost 1/2, which means F is not collision free. A weaker result not requiring the many collisions property was given by Damgard [3]. A preprint of his paper attempted to prove the stronger version given here without the many collisions property, prompting Gibson [6] to give an example showing that this is required.

(ii) This follows immediately from the definitions.

(iii) Suppose there is an $r > 1$ such that f is an r: 1 map from a nonnegligible subset X of S onto $f(S)$. Then clearly the many collisions property applies to f. Now let J be a nonnegligible subset of $f(S)$, and let $I = f^{-1}(J)$. Then

$$|I|/|S| = |I|/|X| \times |X|/|S|$$
$$= |J|/|f(S)| \times |X|/|S|$$

which is nonnegligible.

7 Strong DL hash function family is one way

Definition: A set D of integers is hard to factor if the number of m-bit members of D is not polynomially bounded in m, it is easy to select polynomially in m many m-bit members of D, but every polynomial in m factoring algorithm fails for almost all m-bit members of D.

Of course we do not know whether such a set exists, but the set of DL-superstrong integers defined in Section 3 is a good candidate.

Theorem 3: Let D be a hard to factor set of DL-strong integers. Let $s: N \to N$ be polynomially bounded with $s(m) > m$, where N denotes the natural numbers, and let F be the hash function family whose instances of size m are all the functions $f: \{0, 1\}^{s(m)} \to \{0, 1\}^m$ given by $f(x) = a^x \bmod n$, where n is an m-bit member of D, and a is a DL-strong base for n. Then F is collision free, one way, and output one way.

Proof: Theorem 2 shows F is collision free, it is easy to show it is quasiperiodic, so by Lemma 2 it is both forms of one way.

Theorem 1 means we can drop the requirement that a be DL-strong for n. Theorem 4 implies that we can even drop the requirement that members of D be DL-strong!

8 DL hash functions with hard to factor moduli

We sketch below generalisations of theorems 1(*a*) and 2 that apply to integers n with the property that for every prime factor p of n, $p - 1$ has a large prime factor. We show that if n has this property then for almost all $a \in Z_n^*$, collisions for a DL hash function with modulus n and base a reveal a factor of n. Now if n does not have this property a factor of n can almost certainly be found easily by the Pollard $p - 1$ method [10]. Thus our results imply that if F is a hash function family of DL hash functions with a hard to factor set of moduli, then F is collision free and one way.

Theorem 4: Let c, d be positive integers with $c < d$. Let the odd t-bit integer n have $k > 1$ distinct prime factors, and suppose that for each prime factor p of n, $p - 1$ is of the form $2up_1$, where $u \leqslant c$, and all the prime factors of p_1 are $> d$. Then

(*a*) The Carmichael function $\lambda(n)$ is of the form $2UP$, where $U \leqslant c^k$, any prime factors of U are $\leqslant c$, and all the prime factors of P are $> d$.

(*b*) The proportion of members of Z_n^* whose order is not a multiple of P is less than the sum of the reciprocals of the distinct prime factors of P.

(*c*) If the order of $a \in Z_n^*$ is a multiple of P, $x \neq 0$ satisfies $a^x = 1 \bmod n$, and $2Ux$ is s-bit, then there is an algorithm with input a, n, x that outputs a factor of n with probability $\geqslant 0.5$ in at most $2Us$ modular multiplications and one gcd computation of t-bit integers.

Proof (sketch):

(*a*) This follows immediately from the way $\lambda(n)$ is calculated.

(*b*) This follows from the decomposition of Z_n^* into a direct product of cyclic groups of prime power order [7], noting that these orders must divide $2UP$, and that $2U$ is coprime to P.

(*c*) Apply the Miller-Bach lemma, noting that $2Ux$ is a multiple of $\lambda(n)$.

9 References

1 BACH, E.: 'Discrete logarithms and factoring'. Report no. UCB/CSD 84/186, Comp. Sc. Division (EECS), Univ. of California, Berkeley, June 1984.

2 BOYAR, J.F., KURTZ, S.A., and KRENTEL, M.W.: 'A discrete logarithm implementation of perfect zero-knowledge blobs', *J. Cryptol.*, 1990, **2**, (2)

3 DAMGARD, I.: 'Design principles for hash function'. Advances in Cryptology, Crypto 89, Lecture Notes in Comp. Sci., vol. 435, Springer, 1990.

4 DAVIES, D.W., and PRICE, W.L.: 'Security for computer networks' (Wiley, 1984)

5 DIFFIE, W., and HELLMAN, M.: 'New directions in cryptography', *IEEE Trans., Inf. Theory*, 1976, **22**

6 GIBSON, J.K.: 'Some comments on Damgard's hashing principle', *Electron. Lett.*, 1990, **26**, (15)

7 KAPLANSKI, I.: 'Infinite abelian groups' (Univ. of Michigan Press, 1954)

8 McCURLEY, K.: 'A key distribution system equivalent to factoring', *J. Cryptol.*, 1988, **1**, (2)

9 MILLER, G.: 'Riemann's hypothesis and tests for primality', *J. Comput. & Syst. Sci.*, 1976, **13**

10 POLLARD, J.: 'Theorems on factorisation and primality testing', *Proc. Cambridge Philos. Soc.*, 1974, **76**

11 SHMUELY, Z.: 'Composite Diffie-Hellman public key systems are hard to break'. Technical Report No. 356, Comp. Sc. Dept., Technion-Israel Inst. of Tech., 1985.

12 SCHRIFT, A.W., and SHAMIR, A.: 'The discrete log is very discreet'. Proc. 22nd Annual ACM Symp. on Theory of Computing (STOC). Baltimore, May 1990.

13 WILLIAMS, H.C.: 'A p + 1 method of factoring', *Math. Comput.*, 1982, **39**

14 YAO, A.: 'Theory and applications of trapdoor functions'. Proc. 23rd Annual IEEE Symp. on Foundations of Computer Science (FOCS) 1982.