On the Hardness of the Diffie-Hellman Decision Problem

March 20, 1998

Abstract

It is shown that in the model of generic algorithms, the Diffie-Hellman decision problem is *not* polynomial-time computationally equivalent to the Diffie-Hellman problem.

Keywords. Diffie-Hellman protocol, Diffie-Hellman decision problem, discrete logarithms, generic algorithms, complexity, lower bounds.

Definition 1 Let G be a cyclic group with generator g. The *Diffie-Hellman (DH) problem* is to compute, given two group elements g^u and g^v, the element g^{uv}. The *Diffie-Hellman decision (DHD) problem* on the other hand is, given a triple (g^u, g^v, g^w), to decide whether $w \equiv uv \pmod{|G|}$.

Definition 2 Let G be a cyclic group with generator g. A *Diffie-Hellman decision oracle* (DHD oracle for short) takes as input a triple (g^u, g^v, g^w) of group elements and outputs `yes` if $w \equiv uv \pmod{|G|}$ and `no` otherwise.

Theorem 1 *Let n be a positive integer and let p be a prime factor of n. Assume that a generic algorithm is given that works for groups of order n, makes calls to a DHD oracle for G and runs in time at most T. Then the probability, taken over the input and the coin tosses of the algorithm, that the algorithm correctly solves the DH problem is at most*

$$\alpha \leq \frac{(T+3)(T+2)+4}{2p} .$$

Proof. Let $n = p^t s$ with $t \geq 1$ and $\gcd(s,p) = 1$. We can assume $n = p^t$. The generic algorithm takes as inputs $\sigma(1)$, $\sigma(x)$, and $\sigma(y)$, where σ is the randomly chosen encoding function, and should compute $\sigma(xy)$. The algorithm is allowed to call, in addition to the usual oracles for addition and inversion, an oracle that solves the DHD problem, i.e., that computes the function DHD with

$$\mathrm{DHD}(\sigma(u), \sigma(v), \sigma(w)) = \texttt{yes}$$

if $w \equiv uv \pmod{|G|}$ and $\mathrm{DHD}(\sigma(u), \sigma(v), \sigma(w)) = \texttt{no}$ otherwise. Assume that the algorithm makes A calls to the addition or inversion oracle and B calls to the DHD oracle in a particular execution. Hence we have $A+B \leq T$. By calling the oracles, the algorithm can compute $P_i(x, y)$, $i = 1, \ldots, A+3$, for bivariate polynomials $P_i(X, Y)$ with $P_1(X, Y) = 1$, $P_2(X, Y) = X$, $P_3(X, Y) = Y$, and for $i > 3$ either $P_i(X, Y) = P_k(X, Y) + P_l(X, Y)$ or $P_i(X, Y) = -P_k(X, Y)$ for some $k, l < i$. Clearly, $P_i(X, Y)$ is a linear polynomial for all i. We can assume that all the polynomials are distinct. Furthermore, the algorithm calls the DHD oracle for B input triples $(P_i(x, y), P_j(x, y), P_k(x, y))$. Here, we can assume that none of these polynomials is constant, in particular, that the answer of the DHD oracle is not trivially $\texttt{yes}$.

Let $\mathcal{E}$ be the event that either $P_i(x, y) = P_j(x, y)$ for some $i \neq j$, or that the DHD oracle answers $\texttt{yes}$ at least once. Observe first that, given $\mathcal{E}$, everything the algorithm sees is statistically independent from x. Second,

$$P[\mathcal{E}] \leq \frac{(A+3)(A+2)}{2p} + \frac{2B}{p} \leq \frac{(T+3)(T+2)}{2p} \ . \tag{1}$$

The first expression in (1) is the number of two-element sets

$$\{i, j\} \subseteq \{1, \ldots, A+3\}$$

times the probability that a linear polynomial takes the value 0 for random values of the variables. The second expression on the other hand is B times the probability $2p$ that a relation of the form

$$P_i(x, y) \cdot P_j(x, y) = P_k(x, y)$$

is satisfied for random x and y (i.e., that a certain *quadratic* polynomial takes on the value 0).

Finally, the success probability α of the algorithm satisfies

$$\alpha \leq P[\mathcal{E}] + P[\overline{\mathcal{E}}] \cdot \frac{2}{p \cdot P[\overline{\mathcal{E}}]} \leq \frac{(T+3)(T+2)+4}{2p} \ .$$

The reason is that, given $\overline{\mathcal{E}}$, the best thing the algorithm can do is output one of the values $P_i(x, y)$. However $P_i(x, y) = xy$ holds with probability at most $2/(p \cdot P[\overline{\mathcal{E}}])$ over the random choices of x and y. $\square$

Corollary 2 *Let n be an integer and p be a prime factor of n. Let a generic reduction of the DH problem to the DHD problem for groups of order n be given with expected running time T. Then*

$$T \geq \sqrt{p}/2 - 3/2 \ .$$

Proof. Assume that the execution of the probabilistic algorithm is aborted after $2T$ steps. This new algorithm has running time at most $2T$ and answers correctly with probability at least $1/2$. Hence the result follows from Theorem 1. $\square$

Corollary 3 *For groups whose orders n have a prime factor p which is not of order $(\log n)^{O(1)}$, the DHD problem is not polynomial-time equivalent to the DH problem in a generic sense.*

References

[1] D. Boneh and R. J. Lipton, Algorithms for black-box fields and their application to cryptography, *Advances in Cryptology - CRYPTO '96*, Lecture Notes in Computer Science, Vol. 1109, pp. 283–297, Springer-Verlag, 1996.

[2] B. den Boer, Diffie-Hellman is as strong as discrete log for certain primes, *Advances in Cryptology - CRYPTO '88*, Lecture Notes in Computer Science, Vol. 403, pp. 530–539, Springer-Verlag, 1989.

[3] W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644–654, 1976.

[4] J. L. Massey, Advanced Technology Seminars Short Course Notes, pp. 6.66–6.68, Zürich, 1993.

[5] U. M. Maurer, Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, *Advances in Cryptology - CRYPTO '94*, Lecture Notes in Computer Science, Vol. 839, pp. 271–281, Springer-Verlag, 1994.

[6] U. M. Maurer and S. Wolf, Lower bounds on generic algorithms in groups, to appear in *Advances in Cryptology - EUROCRYPT '97*, Lecture Notes in Computer Science, Springer-Verlag, 1998.

[7] U. M. Maurer and S. Wolf, The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms, to appear in *SIAM Journal of Computing*, 1998.

[8] U. M. Maurer and S. Wolf, Diffie-Hellman oracles, *Advances in Cryptology - CRYPTO '96*, Lecture Notes in Computer Science, Vol. 1109, pp. 268–282, Springer-Verlag, 1996.

[9] K. S. McCurley, The discrete logarithm problem, in *Cryptology and computational number theory*, C. Pomerance (Ed.), Proc. of Symp. in Applied Math., Vol. 42, pp. 49–74, American Mathematical Society, 1990.

[10] A. J. Menezes, *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 1993.

[11] S. C. Pohlig and M. E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Transactions on Information Theory*, Vol. 24, No. 1, pp. 106–110, 1978.

[12] J. T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, *Journal of the ACM*, Vol. 27, No. 4, pp. 701–717, 1980.

[13] V. Shoup, Lower bounds for discrete logarithms and related problems, *Advances in Cryptology - EUROCRYPT '97*, Lecture Notes in Computer Science, Vol. 1233, pp. 256–266, Springer-Verlag, 1997.