The Hardness of the DHK Problem in the Generic Group Model

Alexander W. Dent

Royal Holloway, University of London
Egham, Surrey, TW20 0EX, U.K.
a.dent@rhul.ac.uk

Abstract. In this note we prove that the controversial Diffie-Hellman Knowledge problem is secure in the generic group model. This appears to be the first paper that presents any evidence as to whether the Diffie-Hellman Knowledge problem is true or false, although a similar result was developed independently and in parallel by Abe and Fehr [1].

1 The Generic Group Model

It is clear that the way in which we represent a group to a polynomial-time algorithm affects the computational power of that algorithm. For example, the computational Diffie-Hellman problem is (almost) always presented as a problem on a representation of the group C_p, where p is a large prime number. However, it is clear that the difficulty of solving the Diffie-Hellman problem depends on the way the group C_p is represented: if C_p is presented as additive arithmetic modulo p then the Diffie-Hellman problems is easy, whereas if C_p is presented as the order p subgroup of the multiplicative group of a finite field or of an elliptic curve group then we believe the Diffie-Hellman problem is hard to solve.

The generic group model is a theoretical model that aims to analyse the success of algorithms against groups whose representations reveal no information to the attacker. There are various attempts to formalise the idea of a generic group [2, 12, 15, 17]. The most popular (and intuitively obvious) of these is that provided by Shoup [17]. In this model, the attacker is not given direct access to group elements, but to the images of group elements under the the action of a random one-to-one mapping $\sigma : G \rightarrow \{0,1\}^k$. Group operations can be computed by the algorithm by way of a series of oracles. The attacker is given access to addition oracle ADD and an inversion oracle INV such that

$$ADD(\sigma(x), \sigma(y)) = \sigma(x+y) \qquad \text{and} \qquad INV(\sigma(x)) = \sigma(-x) \tag{1}$$

It is clear that in this situation, the attacker can gain no advantage in solving a computational problem from the representation $\sigma(x)$ of the group element x.

The model has been used to provide evidence as to the hardness of several computational problems [6, 13, 17, 18]. However, we remind the reader that the generic group model can only ever be used to provide evidence as to the hardness of the problem, not to provide any kind of proof. This is because (1) it does not tell us anything the difficulty of a problem in any one particular group representation, and (2) it has been shown that there exists problems that are provably difficult in the generic group model and yet insecure when this problem is instantiated on *any* particular group representation [8]. Nevertheless, the generic group model has been used to justify the use of several new assumptions recently, particular in situations where authors wish to prove the security of cryptosystems without using the random oracle model.

In the next section, when we consider the Diffie-Hellman Knowledge (DHK) assumption, one valid strategy that the attacker might be able to employ is to pick group elements at random (i.e. in such a way that the attacker does not know their discrete logarithm with respect to any base). This ability is not usually considered in the generic group model. We model this ability by setting $k = \lceil \log |G| \rceil$. The attacker may now generate random group elements by choosing random strings $\hat{\sigma} \in \{0,1\}^k$: these will be a representation of some group element with probability at least $1/2$.[1] If a random $\hat{\sigma}$ is the representation of some new group element, then it will be the representation of a random group element whose image under σ has not already been computed. It may be assumed that the addition and inversion oracles return an error message when queried with a bitstring $\hat{\sigma} \notin Im\ \sigma$. If we wish to consider groups for which it is impossible to pick random group elements, then we should take $k \gg \log |G|$. It should be noted that all of known results on the difficulty of solving computational problems in the generic group model remain true when the attacker is allowed to sample group elements at random.

2 The Diffie-Hellman Knowledge (DHK) Assumption

In this section we will consider the difficulty of the Diffie-Hellman Knowledge (DHK) problem in the generic group model.

Definition 1. *Let λ be a security parameter and σ be a representation of the cyclic group $\mathbb{Z}/n\mathbb{Z}$ where n contains a prime factor p of bit-length λ. Let $\mathcal{A}$ be any algorithm (attacker) that takes the group elements $(\sigma(1), \sigma(x))$ as input, where x is chosen at random from $\{1, 2, \ldots, n\}$, and outputs bitstrings $(B, C) \in \{0,1\}^k \times \{0,1\}^k$. The Diffie-Hellman Knowledge (DHK) assumption states that for each polynomial-time attacker $\mathcal{A}$, there exists a polynomial-time extractor $\mathcal{A}^*$ that takes as input the group elements $(\sigma(1), \sigma(x), B, C)$ and the random coins $R[\mathcal{A}]$ used by $\mathcal{A}$, and outputs an element $r \in \{1, 2, \ldots, n\}$ such that $B = \sigma(r)$ and $C = \sigma(xr)$ (if such an r exists).*

[1] An alternative solution would be to provide the attacker with access to an oracle that randomly generates group elements. This has the advantage that the attacker can always generate a random group element with probability one. Since our analysis will assume that every new bitstring $\hat{\sigma} \in \{0,1\}^k$ that the attacker produces is the encoding of some group element, our results will hold regardless of how we define the attacker's ability to sample group elements.

The DHK assumption is designed to capture the notion that it is impossible to create a Diffie-Hellman tuple $(\sigma(1), \sigma(x), \sigma(r), \sigma(xr))$ from $(\sigma(1), \sigma(x))$ without knowing r. This is a very strong assumption that, despite being used in several high-profile papers [4, 5, 7, 9–11], has been heavily criticised. Opponents of the assumption have pointed out that it is not efficiently falsifiable [14] and so any proof that it is false must be complex and as difficult to check as a proof that it is true. In particular, experimental evidence cannot be used to check whether this assumption is false or true.

We have presented the 'standard model' version of the DHK problem. We will actually show that, in the generic group model, there exists a single extractor $\mathcal{A}^*$ that can recover the value r produced by *any* polynomial-time attacker $\mathcal{A}$ when given the oracle queries that $\mathcal{A}$ used to produce its output. This is clearly sufficient to show that the DHK assumption is true for a generic group. It can be noted that the difference between the 'standard model' version of the DHK assumption and this 'generic group' version of the DHK assumption is similar to the difference between plaintext awareness in the random oracle model [3] and in the standard model [5]. This result is important because it is the first piece of evidence presented that suggests whether the DHK assumption is true or false.

The proof is comparatively simple, and uses techniques suggested by Shoup [17]. It relies on the following crucial lemma [16, 17].

Lemma 1. *Let $F(x_1, x_2, \dots, x_m)$ be a polynomial of total degree $d \geq 1$. Then the probability that $F(x_1, x_2, \dots, x_m) = 0 \bmod n$ for randomly chosen values $(x_1, x_2, \dots, x_m)$ in $\mathbb{Z}/n\mathbb{Z}$ is bounded above by d/p where p is the largest prime dividing n.*

Theorem 1. *The DHK assumption holds in a generic group.*

Proof The extractor $\mathcal{A}^*$ keeps track of the oracle queries of $\mathcal{A}$ as monomials. We set $F_0 = 1$ and $F_1 = X$ — these represent the group elements $\sigma(1)$ and $\sigma(x)$. If $\mathcal{A}$ makes an oracle query using a bitstring σ_i that has not been an input or output by the addition or inversion oracles, then we assign a new variable Z_i to the group element σ_i. The result of applying the addition oracle on the group elements σ_i and σ_j (represented by monomials F_i and F_j) is a new group element σ_l represented by the monomial $F_l = F_i + F_j$. The result of applying the inversion oracle to a group element σ_i (represented by monomial F_i) is a group element σ_l represented by the monomial $-F_i$.

We may think of these monomials as representing the group because each element σ_i can be thought of as $\sigma(F_i(x, z_1, z_2, \dots, z_m))$. This representation is completely consistent unless the attacker computes two group elements $\sigma_i = \sigma_j$ such that $F_i \neq F_j$. Note that in this case we must have $F_i(x, z_1, z_2, \dots, z_m) = F_j(x, z_1, z_2, \dots, z_m)$ for the randomly chosen values $(x, z_1, z_2, \dots, z_m)$. This occurs with probability at most $O(1/p)$. Hence, the probability that the monomial representation is not consistent with the representation given by σ is bounded by $O(m^2/p)$, which is negligible as a function of the security parameter.

$\mathcal{A}$ eventually terminates and outputs two group elements (σ_i, σ_j) which $\mathcal{A}^*$ represents as monomials (F_i, F_j). If $F_i = r$ and $F_j = rX$ for some value of r, then

$\mathcal{A}^*$ outputs r. Otherwise $\mathcal{A}^*$ outputs $\perp$ — that the tuple is not a Diffie-Hellman tuple. If $(\sigma(1), \sigma(x), \sigma_i, \sigma_j)$ is a Diffie-Hellman tuple, then

$$\begin{aligned} & x \cdot F_i(x, z_1, z_2, \dots, z_m) = F_j(x, z_1, z_2, \dots, z_m) \\ \iff & x \cdot F_i(x, z_1, z_2, \dots, z_m) - F_j(x, z_1, z_2, \dots, z_m) = 0 \,. \end{aligned}$$

This can occur because $X \cdot F_i = F_j$ (in which case $F_i = r$ and $F_j = rX$, and the extractor $\mathcal{A}^*$ returns the correct value r), or because $X \cdot F_i \neq F_j$ but the equation holds for the particular random values $(x, z_1, z_2, \dots, z_m)$ used (in which case the extractor fails). However, this latter event occurs with probability at most $2/p$. Hence, the extractor works with non-negligible probability. □

It should be noted that a similar result was developed independently (and concurrently) by Abe and Fehr [1].

Acknowledgements

Thanks to Paul Crowley and Martijn Stam for pointing out grammatical errors, and to Serge Fehr for pointing out the similarities between this note and his own work.

References

1. M. Abe and S. Fehr. Perfect NIZK with adaptive security. Unpublished manuscript, 2006.
2. L. Babai and E. Szemerédi. On the complexity of matrix group problems I. In *25th Annual ACM Symposium on the Theory of Computing*, pages 229–240, 1984.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology – Crypto '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer-Verlag, 1998.
4. M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In M. Franklin, editor, *Advances in Cryptology – Crypto 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 273–289. Springer-Verlag, 2004.
5. M. Bellare and A. Palacio. Towards plaintext-aware public-key encryption without random oracles. In P. J. Lee, editor, *Advances in Cryptology – Asiacrypt 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 48–62. Springer-Verlag, 2004.
6. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *Advance in Cryptology – Eurocrypt 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer-Verlag, 2005.
7. I. B. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, *Advances in Cryptology – Crypto '91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer-Verlag, 1991.
8. A. W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Y. Zheng, editor, *Adavcnes in Cryptology – Asiacrypt 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 100–109. Springer-Verlag, 2002.

9. A. W. Dent. The Cramer-Shoup encryption scheme is plaintext aware in the standard model. In S. Vaudenay, editor, *Advances in Cryptology – Eurocrypt 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 289–307. Springer-Verlag, 2006.
10. S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In H. Krawcyzk, editor, *Advances in Cryptology – Crypto '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 408–423. Springer-Verlag, 1998.
11. H. Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In V. Shoup, editor, *Advances in Cryptology – Crypto 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 546–566. Springer-Verlag, 2005.
12. U. Maurer. Abstract models of computation in cryptography. In N. P. Smart, editor, *Coding and Cryptography: 10th IMA International Conference*, volume 2796 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 2005.
13. U. Maurer and S. Wolf. Lower bounds on generic algorithms in groups. In K. Nyberg, editor, *Advances in Cryptology – Eurocrypt '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 72–84. Springer-Verlag, 1998.
14. M. Naor. On cryptographic assumptions and challenges. In D. Boneh, editor, *Advances in Cryptology – Crypto 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer-Verlag, 2003.
15. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994. Translated from *Matematicheskie Zametki*, 55(2):91–101, 1994.
16. J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.
17. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology – Eurocrypt '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer-Verlag, 1997.
18. N. Smart. The exact security of ECIES in the generic group model. In B. Honary, editor, *Coding and Cryptography*, volume 2260 of *Lecture Notes in Computer Science*, pages 73–84. Springer-Verlag, 2001.