# On the hash function of ODH

Xianhui Lu[1], Xuejia Lai[2], Dake He[1], Guomin Li[1]
Email:luxianhui@gmail.com

1:School of Information Science & Technology, SWJTU, Chengdu, China
2:Dept. of Computer Science and Engineering, SJTU, Shanghai, China

**Abstract.** M. Abdalla, M. Bellare and P. Rogaway proposed a variation of Diffie-Hellman assumption named as oracle Diffie-Hellman(ODH) assumption. They recommend to use a one-way cryptographic hash function for the ODH assumption. We notice that if the hash function is just one-way then there will be an attack. We show that if the the hash function is non-malleable then the computational version of ODH assumption can be reduced to the computational Diffie-Hellman(CDH) assumption. But we can not reduce the ODH assumption to the decisional Diffie-Hellman(DDH) even if the hash function is non-malleable. It seems that we need a random oracle hash function to reduce the ODH assumption to the DDH assumption.

**Keywords:** ODH, hash, one-way, non-malleable

## 1 Introduction

The Diffie-Hellman problem [1] and its variants are widely used in modern cryptographic systems[6, 8, 7, 9, 10]. There are several works to study classical and variable Diffie-Hellman problems [2, 3, 11, 5]. M. Abdalla, M. Bellare and P. Rogaway proposed a variation of Diffie-Hellman assumption named as oracle Diffie-Hellman(ODH) assumption. The ODH assumption suppose that provide an adversary $A$ with $(g \in G, g^v, g^u, W)$ ($G$ is a group of large prime order $q$) and an oracle $\mathcal{H}_v$, which computes the function $H(X^v)$, the adversary can not tell whether $W = H(g^{uv})$ or not. M. Abdalla, M. Bellare and P. Rogaway point out that for some simple choices of functions $H$, an adversary can use the oracle $\mathcal{H}_v$ and the result of Boneh and Venkatesan[4] to solve the Diffie-Hellman problem. They recommend to use a one-way cryptographic hash function for the ODH assumption and points that these attacks do not appear to work when a one-way hash cryptographic hash function is used.

### 1.1 Our Contributions

We notice that if the hash function is one-way but not non-malleable then we can constructs an attack on the ODH problem. We show that if the the hash function is non-malleable then the computational version of ODH assumption can be reduced to the computational Diffie-Hellman(CDH) assumption. But we can not reduce the ODH assumption to the decisional Diffie-Hellman(DDH) even if the hash function is non-malleable. It seems that we need a random oracle hash function to reduce the ODH assumption to the DDH assumption.

## 2 Preliminaries

We will review the standard definitions of ODH assumption[11] and non-malleable hash function.

In describing probabilistic processes, we write $x \overset{R}{\leftarrow} X$ to denote the action of assigning to the variable $x$ a value sampled according to the distribution X. If $S$ is a finite set, we simply write $s \overset{R}{\leftarrow} S$ to denote assignment to $s$ of an element sampled from uniform distribution on $S$. If $A$ is a probabilistic algorithm and $x$ an input, then $A(x)$ denotes the output distribution of $A$ on in put $x$. Thus, we write $y \overset{R}{\leftarrow} A(x)$ to denote of running algorithm $A$ on input $x$ and assigning the output to the variable $y$.

### 2.1 The Oracle Diffie-Hellman Problem

Now we review the definition of oracle Diffie-Hellman assumption[11]. Let $G$ be a group of large prime order $q$, $H : \{0,1\}^* \to \{0,1\}^{hLen}$ be a cryptographic hash function and consider the following two experiments:

experiments $\mathrm{Exp}_{H,A}^{odh-real}$:

$$u \overset{R}{\leftarrow} Z_q^*; U \leftarrow g^u; v \overset{R}{\leftarrow} Z_q^*; V \leftarrow g^v; W \leftarrow H(g^{uv})$$

$$\mathcal{H}_v(X) \overset{def}{=} H(X^v); b \leftarrow A^{\mathcal{H}_v(\cdot)}(U,V,W); \text{return } b$$

experiments $\mathrm{Exp}_{H,A}^{odh-rand}$:

$$u \overset{R}{\leftarrow} Z_q^*; U \leftarrow g^u; v \overset{R}{\leftarrow} Z_q^*; V \leftarrow g^v; W \leftarrow \{0,1\}^{hLen}$$

$$\mathcal{H}_v(X) \overset{def}{=} H(X^v); b \leftarrow A^{\mathcal{H}_v(\cdot)}(U,V,W); \text{return } b$$

Now define the advantage of the $A$ in violating the oracle Diffie-Hellman assumption as

$$Adv_{H,A}^{odh} = \Pr[\mathrm{Exp}_{H,A}^{odh-real} = 1] - \Pr[\mathrm{Exp}_{H,A}^{odh-rand} = 1]$$

Here $A$ is allowed to make oracle queries that depend on the $g^u$ with the sole restriction of not being allowed to query $g^u$ itself.

### 2.2 Non-malleable hash function

A hash function $H$ is said to be non-malleable if it is infeasible for an adversary to find two functions $f$ and $g$ that $H(x) = y, H(f(x)) = g(y)$. Non-malleable hash function is a stronger notion than one-way hash function. It is clear that if we can reverse a hash function then we can get $g(y) = H(f(H^{-1}(y)))$ for any function $f$.

## 3 Attack on ODH

In sections 2 we showed that non-malleable hash function is a stronger notion than one-way hash function. A non-malleable hash function must be a one-way hash function, a one-way hash function may not be a non-malleable hash function. According to the recommendation in [11], the hash function $H$ used in ODH assumption is a one-way hash function. Suppose that $H$ is not non-malleable and $f, g$ are functions that $H(x) = y, H(f(x)) = g(y)$. Give $(g \in G, g^v, g^u, W)$ and an oracle $\mathcal{H}_v$ where $G$ is a group of large prime order $q$, $(v, u) \in Z_q^*$ are choose randomly, the adversary can calculate $W'$ as follow:

$$x' \leftarrow f(g^u); y' \leftarrow \mathcal{H}_v(x'); w' \leftarrow g^{-1}(y')$$

From the definition we have that $w' = H(g^{uv})$. The adversary then checks whether $w' = W$. That's finish the attack on the ODH assumption. We see that when the hash function is not non-malleable the oracle $\mathcal{H}_v$ will yield an attack.

## 4 Computational version of ODH

In section 3 we showed that it is not enough if the hash function is only a one-way hash function. But what if instead we used a non-malleable hash function? We find that if the hash function is non-malleable then the oracle $\mathcal{H}_v$ will not help the adversary to calculate $H(g^{uv})$. But we do not know if the oracle $\mathcal{H}_v$ will help the adversary to tell whether $H(g^{ur}) = W$ or not. That's to say we can not reduce the ODH assumption to DDH assumption even if we use a non-malleable hash function. Now let's consider the computational version of the ODH assumption. Let $G$ be a group of large prime order $q$, $H : \{0,1\}^* \rightarrow \{0,1\}^{hLen}$ be a cryptographic hash function, consider the experiment: $\mathrm{Exp}_{H,A}^{c-odh}$:

$$u \overset{R}{\leftarrow} Z_q^*; U \leftarrow g^u; v \overset{R}{\leftarrow} Z_q^*; V \leftarrow g^v;$$

$$\mathcal{H}_v(X) \overset{def}{=} H(X^v); Z \leftarrow A^{\mathcal{H}_v(\cdot)}(U, V);$$

$$if\ Z = g^{uv}\ return\ 1,\ else\ return\ 0.$$

Now define the advantage of the $A$ in violating the computational oracle Diffie-Hellman assumption as

$$Adv_{H,A}^{c-odh} = \Pr[\mathrm{Exp}_{H,A}^{c-odh} = 1]$$

Here $A$ is allowed to make oracle queries that depend on the $g^u$ with the sole restriction of not being allowed to query $g^u$ itself.

From the definition above we can see that, if $H$ is a non-malleable hash function then it will not give the adversary any help with calculating $Z$. Then the only way the adversary get $H(g^{uv})$ is to calculate $g^{uv}$ and then use the hash function $H$ to get $H(g^{uv})$. That's to say the oracle $\mathcal{H}_v$ is useless and the computational ODH assumption is reduce to the classical CDH assumption.

What hash function should we use to reduce the ODH assumption to the DDH assumption? If we can reduce the ODH assumption to the DDH assumption then the oracle $\mathcal{H}_v$ will not tell any information of $H(g^{uv})$, that's to say $H(g^{u'v})$ is independent to $H(g^{uv})$ from the adversary's view. It seems that only the random oracle hash function works.

## 5 Conclusion

We give an attack on the ODH assumption when the underlying hash function is not non-malleable. We showed that the computational ODH assumption can be reduced to the classical CDH assumption when the hash function is non-malleable. But we can not reduce the ODH assumption to the DDH assumption even the hash function is non-malleable. It seems that the hash function must to be a random oracle if we want to reduce the ODH assumption to the DDH assumption.

## References

1. Whitfield Diffie and Martin Hellman. New directions in cryptography. IEEE Transactions on Information Theory, IT No.2(6):644C654, November 1976.
2. Ueli M. Maurer and Stefan Wolf. Diffie-Hellman, Decision Diffie-Hellman, and discrete logarithms. In IEEE Symposium on Information Theory, page 327, Cambridge, USA, August 1998.
3. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, Advances in Cryptology-EUROCRYPT97, number 1233 in Lecture Notes in Computer Science, pages 256C266. International Association for Cryptologic Research, Springer Verlag, Berlin Germany, 1997.
4. D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in diffiehellman and related schemes. In N. Koblitz, editor, Advances in Cryptology C CRYPTO96, volume 1109 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, Aug. 1996.
5. Feng Bao, Robert H. Deng and HuaFei Zhu. Variations of Diffie-Hellman Problem. Information and Communications Security, volume 2836 of Lecture Notes in Computer Science, pages301-312. Springer-Verlag, 2003.
6. R. Cramer and V. Shoup, ”A Practical Public Key Cryptosystem Provably Secure Against Chosen Ciphertext Attack”, *Adv. in Cryptology - Crypto 1998*, LNCS vol. 1462, Springer- Verlag , pp. 13-25, 1998;
7. V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. In B. Preneel, editor, Advances in Cryptology - Eurocrypt 2000, volume 1807 of Lecture Notes in Computer Science, pages 275-288. Springer-Verlag, 2000.
8. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing, 33(1):167-226, 2003.
9. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryp- tion scheme. In M. Franklin, editor, Advances in Cryptology - Crypto 2004, volume 3152 of Lecture Notes in Computer Sciene, pages 426-442. Springer-Verlag, 2004.
10. Eike Kiltz. Chosen-Ciphertext Secure Key Encapsulation based on Hashed Gap Decisional Diffie-Hellman. Proceedings of the 10th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2007, pp. 282–297 LNCS 4450 (2007).Springer-Verlag. Full version available on Cryptology ePrint Archive: Report 2007/036
11. M. Abdalla, M. Bellare and P. Rogaway. DHIES: An encryption scheme based on the Diffie-Hellman Problem Extended abstract, entitled The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES, was in Topics in Cryptology - CT-RSA 01, Lecture Notes in Computer Science Vol. 2020, D. Naccache ed, Springer-Verlag, 2001