Chapter 3
Variants of the Diffie–Hellman Problem

Kannan Balasubramanian
Mepco Schlenk Engineering College, India

ABSTRACT
Many variations of the Diffie-Hellman problem exist that can be shown to be equivalent to one another. We consider following variations of Diffie-Hellman problem: square computational and Square decisional Diffie-Hellman problem, inverse computational and inverse computational decisional Diffie-Hellman problem and divisible computational and divisible decisional Diffie-Hellman problem. It can be shown that all variations of computational Diffie-Hellman problem are equivalent to the classic computational Diffie-Hellman problem if the order of a underlying cyclic group is a large prime. We also describe other variations of the Diffie-Hellman problems like the Group Diffie-Hellman problem, bilinear Diffie-Hellman problem and the Elliptic Curve Diffie-Hellman problem in this chapter.

INTRODUCTION
The Diffie-Hellman problem is to compute $g^{xy}$ given $g^x$ and $g^y$ using a generator in a cyclic group. This is a difficult problem and is used in the Diffie-Hellman Exchange protocol. Many other problems can be shown to be equivalent to this protocol by reducing one problem to another. The Diffie-Hellman problems can be either computational or decisional. Both the computational and the decisional variants of the square Diffie-Hellman problem, Inverse Diffie-Hellman problem and the divisible Diffie-Hellman problems are considered in this chapter and shown to be equivalent if the order of the underlying cyclic group is large prime.

The basic tools for relating the complexities of various problems are polynomial reductions and transformations. We say that a problem $A$ reduces in polynomial time to another problem $B$, denoted by $A \leq_p B$ if and only if there is an algorithm for $A$ which uses a subroutine for $B$, and each call to the subroutine for $B$ counts as a single step, and the algorithm for $A$ runs in polynomial-time. The latter implies that the subroutine for $B$ can be called at most a polynomially bounded number of times. The practical implication comes from the following proposition: If $A$ polynomially reduces to $B$ and there is a polynomial time algorithm for $B$, then there is a polynomial time algorithm for $A$ also.

DOI: 10.4018/978-1-5225-2915-6.ch003
DIFFIE-HELLMAN PROBLEM

In this chapter, we are considering useful variations of Diffie-Hellman problem: square Computational (and decisional) Diffie-Hellman problem, inverse computational (and decisional) Diffie-Hellman problem and divisible computational (and decisional) Diffie-Hellman problem. We are able to show that all variations of computational Diffie-Hellman problem are equivalent to the classic computational Diffie-Hellman problem if the order of an underlying cyclic group is a large prime (Bao et al., 2003).

Let $p$ be a large prime number such that the discrete logarithm problem defined in $\mathbb{Z}_p^*$ is hard. Let $G \in \mathbb{Z}_p^*$ be a cyclic group of prime order $q$ and $g$ is assumed to be a generator of $G$. It is assumed that $G$ is prime order, and security parameters $p; q$ are defined as the fixed form $p = 2q + 1$ and $\text{ord}(g) = q$. A remarkable computational problem has been defined on this kind of set by Diffie and Hellman (Diffie, 1976). The Diffie-Hellman assumption (CDH assumption) is stated as follows:

- **Computational Diffie-Hellman Problem (CDH):** On input $g, g^x, g^y$, computing $g^{xy}$. An algorithm that solves the computational Diffie-Hellman problem is a probabilistic polynomial time Turing machine, on input $g, g^x, g^y$, outputs $g^{xy}$ with non-negligible probability. The Computational Diffie-Hellman assumption means that such a probabilistic polynomial time Turing Machine does not exist. This assumption is believed to be true for many cyclic groups, such as the prime sub-group of the multiplicative group of finite fields.

- **Square Computational Diffie-Hellman Assumption:** The square computational Diffie-Hellman problem, introduced in (Maurer et al., 1998) is defined as follows:

- **Square Computational Diffie-Hellman Problem (SCDH):** On input $g, g^x$, computing $g^{x^2}$. An algorithm that solves the square computational Diffie-Hellman problem is a probabilistic polynomial time Turing machine, on input $g, g^x$, outputs $g^{x^2}$ with non-negligible probability. The square computational Diffie-Hellman assumption means that there no such a probabilistic polynomial time Turing machine does not exist.

- **Inverse Computational Diffie-Hellman Problem (invCDH):** On input $g, g^x$, outputs $(g^x)^{-1}$. An algorithm that solves the inverse computational Diffie-Hellman problem is a probabilistic polynomial time Turing machine, on input $g, g^x$, outputs $(g^x)^{-1}$ with non-negligible probability. Inverse computational Diffie-Hellman assumption means that such a probabilistic polynomial time Turing machine does not exist. It can be shown that that the SCDH assumption and InvCDH assumption are equivalent.

- **Divisible Computational Diffie-Hellman Problem (DCDH):** On random input $g, g^x, g^y$, computing $g^{y/x}$. An algorithm that solves the divisible computational Diffie-Hellman problem is a proba-
Secure Two Party Computation
[www.igi-global.com/chapter/secure-two-party-computation/188520?camid=4v1a](www.igi-global.com/chapter/secure-two-party-computation/188520?camid=4v1a)

ICA and PCA-Based Cryptology
[www.igi-global.com/chapter/ica-and-pca-based-cryptology/108031?camid=4v1a](www.igi-global.com/chapter/ica-and-pca-based-cryptology/108031?camid=4v1a)

Reviewing the Security Features in Contemporary Security Policies and Models for Multiple Platforms
[www.igi-global.com/chapter/reviewing-the-security-features-in-contemporary-security-policies-and-models-for-multiple-platforms/153088?camid=4v1a](www.igi-global.com/chapter/reviewing-the-security-features-in-contemporary-security-policies-and-models-for-multiple-platforms/153088?camid=4v1a)

Provable Security for Public Key Cryptosystems: How to Prove that the Cryptosystem is Secure
[www.igi-global.com/chapter/provable-security-for-public-key-cryptosystems/153082?camid=4v1a](www.igi-global.com/chapter/provable-security-for-public-key-cryptosystems/153082?camid=4v1a)