Cryptographic Protocols based on Group Theory

Yang Xiao[1,*], Tim Campion[2]

[1] *Department of Computer Science, Johns Hopkins University*

[2] *Department of Mathematics, Johns Hopkins University*

Abstract

Several cryptographic schemes based on group theory and complexity theory emerged within decades are talked about. This paper mainly describes Diffie-Hellman Key Exchange protocol[4] and its variants with the related group theory material and the computational-hard problems they based on, and systematically builds connection between group theory and complexity theory.

Keywords: Cryptography, Group Theory, Diffie-Hellman, computation theory, computational complexity

Contents

*Corresponding author

Email addresses: yxiao40@jh.edu (Yang Xiao), tcampio1@jh.edu (Tim Campion)

[1] Student

[2] Lecturer

1. Introduction

In the past few decades, as an intersection of mathematics, computer science and security informatics, cryptography becomes a hot topic and attracts a lot of researchers in both mathematics and theoretical computer science. Plenty of crypto-systems emerged in order to ensure computational secure communication. Most of them are based on Groups, Rings, Fields, Lattice and many other algebraic structures. Specifically, some famous and broadly-used cryptosystem such as Diffie-Hellman Key Exchange protocal[4] and its variants is based on group theory. This paper gives a high-level overview of these cryptosystems, platform group structures and computational group theory[9] problems. The rest of this paper follows the following structure: (1) We first describe the background and motivation of cryptography, specifically, the key exchange phase, give several basic definition in cryptography, and talk a little about computational complexity. (2) Then we introduce the Diffie-Hellman Key Exchange protocal (we will represent this as DH in the rest of this paper), corresponding group theory principle and computational problem behind the protocal, and the reduction to prove the underlying problem is in the complexity class $BQP \cap NP \cap coNP$ which provides DH strong security level. (3) Lastly, We introduce several variants of DH and the underlying principles.

2. Fundamental Cryptography

Modern cryptography models crytographic sheme as two parties, Alice and Bob, and the channel between them. Suppose that Alice (Sender) is trying to send Bob (Receiver) a private message through the channel, and wishes no one else (Eavesdropper) except Bob could find out what this message is about. Note that the channel is considered to be public, **i.e.** each bit of data on channel is considered to be eavesdropped. Obviously, directly put the original message on channel is not a feasible solution.

Definition 2.1. *(Informally) During one communication over channel, the readable message that sender wants receiver to obtain is called* ***the plaintext message****.* ***Ciphertext*** *message is the result of encryption performed on plaintext message using an algorithm, called a* ***cipher****.*

Definition 2.2. *(Informally) Encryption is the process of encoding information. Correspondingly, Decryption is the process of decoding information.*

Normally, encryption algorithms are classified into two categories: symmetric key encryption and public key encryption. In a symmetric-key protocol, Alice and Bob share a pair of keys which are exactly the same. Alice encrypt plaintext message using the key, and send the result (ciphertext message) to Bob. Then Bob decrypt the ciphertext using the same key, and get the original plaintext message. Lots of symmetric-key encryption protocol has been developed and broadly-used, **e.g**. DES[10], AES[7]. However, this scenario requires both Alice and Bob agrees on a symmetric key with the assumption that any communication over channel is eavesdropped. This is the classical **key exchange problem**.

Definition 2.3. ***Key exchange*** *(also key establishment) is a method in cryptography by which cryptographic keys are exchanged between two parties, allowing use of a cryptographic algorithm.*

3. Fundamental Computational Complexity Theory

In computer science, the computational complexity or simply complexity of an algorithm is the amount of resources required to run it. Particular focus is given to time and memory requirements. The complexity of a problem is the complexity of the best algorithms that allow solving the problem. A problem is regarded as inherently difficult if its solution requires significant resources, whatever the algorithm used.

Definition 3.1. *A problem is assigned to the* ***P*** *(polynomial time) class if there exists at least one algorithm to solve that problem, such that the number of steps of the algorithm is bounded by a polynomial in n, where n is the length of the input. A problem is assigned to the* ***NP*** *(nondeterministic polynomial time) class if it is solvable in polynomial time by a nondeterministic Turing machine. A problem is said to be* ***NP-hard*** *if an algorithm for solving it can be translated into one for solving any other NP-problem. A problem which is both* ***NP*** *and* ***NP-hard*** *is called an* ***NP-complete*** *problem.*

It is easy to prove that $P \subset NP$, but the converse is still an open problem.

Definition 3.2. *A problem X is assigned to* ***co-NP*** *if and only if its complement $\bar{X}$ is in the complexity class NP.* ***bounded-error quantum polynomial time (BQP)*** *is the class of decision problems solvable by a quantum computer in polynomial time.*

It is also shown to be the case that $P \subset BQP$, $P \subset coNP$. P problems are considered to be **computationally easy**. But the problems that are in $(BQP \cup NP \cup coNP) \setminus P$ are considered to be hard and provide security guarantee for plenty of cryptosystems.

4. Diffie Hellman Key Exchange Protocol

DH[4] is one of the earliest examples of key exchange implemented within the field of cryptography. The protocol can be described as follows:

Let G be a cyclic group, $ord(G) = d$, g is a generator of G, **i.e.** $G = \langle g \rangle = \{g^k | k \in \mathbb{Z}\}$, where g and d is public. Then the following steps are proceeded:

1. Alice chooses an integer $a \in [2, d-1]$ at random and keep it as secret, then compute g^a and send it to Bob.
2. Symmetrically, Bob chooses an integer $b \in [2, d-1]$ at random and keep it as secret, then compute g^b and send it to Alice.
3. Alice computes $(g^b)^a$ as the symmetric key. Bob computes $(g^a)^b$ as the symmetric key.

It is obvious that the symmetric key Alice computes and Bob computes are the same, because $\mathbb{Z}$ is abelian, $(g^a)^b = g^{ab} = g^{ba} = (g^b)^a$. Obviously, for Alice and Bob, the problem to compute g^{ab} is in P, because it needs at most $O(d^2)$ times of multiplication (and modulo operation). Better algorithms can be used to guarantee that the operations it needs to compute symmetric key is at most $O(\log d)$. To be a secure key exchange protocol, DH has to satisfy that it is hard for eavesdropper to obtain g^{ab} after knowing g^a and g^b. Equivalently, if the eavesdropper can easily compute a or b, then the symmetric key is easy to compute.

4.1. Discrete Logarithm Problem

In fact, DH protocol is related to a classical hard-solving problem called the **Discrete Logarithm Problem**: Let G be a cyclic group, g is a generator, given $h \in G$, find an interger t such that $g^t = h$.

Obviously, if DLP is easy to solve then DH is no longer secure. If we take G to be $\mathbb{Z}/d$, then finding out t is just trivial. So DLP relies a lot on the choose of underlying group. In practice, we often select G to be a **finite field** $GF(q)$, where q is a prime that is large enough.

4.2. Hardness of Discrete Logarithm Problem

First, we can easily claim DLP is in BQP by using Shor's algorithm[8] for computing discrete logarithm. For simplicity, we are going to assume $G = \mathbb{Z}_d^\times$. Then a 'YES' instance for the decision version of DLP is $\exists! t \; s.t. \; g^t = h(mod \; d)$, 'No' instance are all the rest instances, according to [2], we can construct witness for instances in both the decision version of DLP and the complement problem by using the **Fermat's Little Theorem**, therefore DLP is in NP and $coNP$. Therefore, the complexity class that DLP belongs to is $BQP \cap NP \cap coNP$.

However, no results have shown DLP is not in P. Neither any NP-hard problem is shown reducible to DLP. It has long been believed that DLP is in NP-intermediate, which, still, is not computationally-easy to solve.

5. Variants of DH

5.1. Conjugacy Search Problem

DH requires the underlying group to be cyclic (abelian). What if the group is not? It turns out replacing exponential operation with conjugacy can solve this problem!

The **Conjugacy Search Problem** (CSP) can be stated as: Let G be a non-abelian group, $g, h \in G \; s.t. \; \exists \; x, \; g^x = h$, where g^x is representing $x^{-1}gx$.

Amazingly, Conjugacy Search Problem for certain groups is proved to be NP-complete[3]. As a result, a variant of DH is proposed in [5].

5.2. Ko–Lee–Cheon–Han–Kang–Park Key Exchange Protocol

The protocol is described as follows: Let G be a non-abelian group, $g \in G$ be an element that is public, A, B are two commuting subgroups of G, i.e. $A, B \leq G$, $[a, b] = a^{-1}b^{-1}ab = e \; \forall a \in A, b \in B$. Then the following steps are proceeded:

1. Alice selects $a \in G$ at random, compute $g^a = a^{-1}ga$ and send it to Bob.
2. Bob selects $b \in G$ at random, compute $g^b = b^{-1}gb$ and send it to Alic.

3. Alice computes $(g^a)^b$ as key, Bob computes $(g^b)^a$ as key.

Since A, B commutes, then $(g^a)^b = g^{ab} = (g^b)^a$. The eavesdropper needs to solve CSP in order to figure out a or b to compute the key. Ko et al[5] proposed a subtle construction of G. They choose the braid group of n strings, i.e. B_n, which can be represented as:

$$B_n = \left\langle \sigma_1, \sigma_2, ..., \sigma_{n-1} \middle| \begin{array}{ll} \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & for\ |i-j| = 1 \\ \sigma_i\sigma_j = \sigma_j\sigma_i & for\ |i-j| \geq 2 \end{array} \right\rangle$$

Take $0 < l < n$, then we can choose A, B to be:

$$A = \langle \sigma_1, \sigma_2, ..., \sigma_{l-1} \rangle$$
$$B = \langle \sigma_{l+1}, \sigma_{l+2}, ..., \sigma_{n-1} \rangle$$

Then it is satisfied that in this construction, A commutes with B according to the representation of braid group. Also, the multiplication operation can be efficiently done which makes the computation of Alice and Bob fast. CSP for B_n is hard enough and provides strong security. However, since the order of braid group and its subgroups is infinite, so the step of 'select $a \in G$ at random' is out of practice.

5.3. Anshel-Anshel-Goldfeld Key Exchange Protocol

Another variant of DH emerges[1] and provides more flexibility for computation.

Anshel-Anshel-Goldfeld Key Exchange Protocol replaced the step of choosing appropriate commuting subgroups and transforms symmetric key into a commutator. Let G be a non-abelian group, $a_1, a_2, ..., a_k, b_1, b_2, ..., b_m \in G$ are public elements.

1. Alice chooses $x \in \{a_1, a_2, ..., a_k\}$ at random, computes $b_1^x, b_2^x, ..., b_m^x$ and sends to Bob.
2. Bob chooses $\in \{b_1, b_2, ..., b_m\}$ at random, computes $a_1^y, a_2^y, ..., a_m^y$ and sends to Alice.
3. Alice takes $x^{-1} \cdot x^y$ as the key. Bob takes $(y^{-1} \cdot y^x)^{-1}$ as the key.

Then, $x^{-1} \cdot x^y = x^{-1}y^{-1}xy = (y^{-1}x^{-1}yx)^{-1} = (y^{-1} \cdot y^x)^{-1} = [x, y]$. However, even this protocol looks good, the platform group is still not well specified. Some groups that might work for this protocol is discussed in [6].

6. Conclusion

In this paper, we talked about fundamental cryptography, computational complexity theory, group techniques in Diffie-Hellman Key Exchange Protocols and its several variants and gives justification for approaches in cryptography that group theory provides enough security level for protocols.

References

[1] Iris Anshel, Michael Anshel, and Dorian Goldfeld. "An algebraic method for public-key cryptography". In: *Mathematical Research Letters* 6.3 (1999), pp. 287–291.

[2] Gilles Brassard. "A note on the complexity of cryptography (corresp.)" In: *IEEE Transactions on information Theory* 25.2 (1979), pp. 232–233.

[3] B Cavallo and D Kahrobaei. "A family of polycyclic groups over which the conjugacy problem is NP-complete. arXiv math.: 1403.4153 v2, 19 Mar". In: (2014).

[4] Whitfield Diffie and Martin Hellman. "New directions in cryptography". In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[5] Ki Hyoung Ko et al. "New public-key cryptosystem using braid groups". In: *Annual International Cryptology Conference*. Springer. 2000, pp. 166–183.

[6] Alexei Myasnikov, Vladimir Shpilrain, and Alexander Ushakov. *Group-based cryptography*. Springer Science & Business Media, 2008.

[7] Douglas Selent. "Advanced encryption standard". In: *Rivier Academic Journal* 6.2 (2010), pp. 1–14.

[8] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: 10.1137/s0097539795293172. URL: https://doi.org/10.1137%2Fs0097539795293172.

[9] Charles C Sims. *Computation with finitely presented groups*. 48. Cambridge University Press, 1994.

[10] Data Encryption Standard et al. "Data encryption standard". In: *Federal Information Processing Standards Publication* 112 (1999).