The ISC Int'l Journal of Information Security
ISeCure

November 2023, Volume 15, Number 3 (pp. 91–100)
http://www.isecure-journal.org

Presented at the ISCISC'2023 in Tehran, Iran.

New Variations of Discrete Logarithm Problem **

Mahdi Mahdavi Oliaee [1], Sahar Khaleghifard [1], and Zahra Ahmadian [1,*]
[1] *Electrical Engineering Department, Shahid Beheshti University, Tehran, Iran*

ARTICLE INFO.

Keywords:
Discrete Logarithm, Hard Problems, Reduction, Diffie-Hellman problem, Cyclic group, Elliptic Curves, Bilinear Pairing

Type:
Research Article

doi:
10.22042/isecure.2023.184477

ABSTRACT

The security of public key cryptography relies on the complexity of certain mathematical hard problems. It is vital to comprehend the intricacy of these problems to develop secure cryptographic schemes and security protocols. This paper provides an overview of some widely recognized hard problems associated with the discrete logarithm problem, including the reductions among them. Furthermore, we introduce a novel hard problem that is equivalent to the discrete logarithm problem, which also has a decisional version. Additionally, a set of new problems is presented, which can be instrumental in the design of secure encryption schemes. This paper is intended to provide crucial insights into the realm of hard problems in cryptography, facilitating a better understanding of security measures.

1 Introduction

The Diffie-Hellmen (DH) problem, introduced in 1976 [1], was the first practical method for establishing a shared secret over an unprotected communications channel. This problem is the foundation of many cryptographic schemes such as other variants of DH key-exchange, Elgamal encryption and variants, and BLS signatures and variants. DH problem is related to the Discrete Logarithm (DL) problem on which the security of many cryptographic schemes such as Schnorr and DSA signature relies.

Public key cryptography relies on the security of certain mathematical problems that are believed to be unsolvable by any Probabilistic Polynomial Time (PPT) adversary. These problems, including discrete logarithm and Diffie-Hellman problems, are known as the cryptographic hard problems.

Hard problems in cryptography are categorized into two main groups: computational problems and decisional problems. Computational problems require the attacker to calculate a parameter, while decisional problems require the attacker to choose between two options. It is evident that any attacker can make the correct choice with a probability of $\frac{1}{2}$. So, the attacker is regarded to be successful if it can choose correctly with a probability of $\frac{1}{2} + \epsilon$, where ϵ is non-negligible. Computational problems are the foundation of one-wayness property, while decisional problems are the basis for achieving indistinguishability in cryptographic schemes. An important instance of such problems for demonstrating one-wayness is DL. However, there is no hard problem for the decisional version of the DL problem. Consequently, it cannot be utilized for establishing indistinguishability. This paper addresses this issue by introducing a

* Corresponding author.
**The ISCISC'2023 program committee effort is highly acknowledged for reviewing this paper.

Email addresses: m_mahdavioliaee@sbu.ac.ir, SaharKhaleghifard@gmail.com, z_ahmadian@sbu.ac.ir

ISSN: 2008-2045 © 2023 ISC. All rights reserved.

novel problem that is equivalent to DL in hardness and has a decisional version, enabling it to establish indistinguishability.

Reduction is a tool used to establish relationships among problems with respect to how difficult they are to solve. When designing cryptographic schemes, prioritizing security requires reducing the hardest possible problem to the scheme's security. However, as the problem becomes more difficult, security reduction usually becomes increasingly complex and potentially unattainable. Therefore, it would be highly advantageous to offer a diverse range of hard problems, though not as difficult as the DL or DH problems, which can be utilized in various designs based on their specific characteristics. This approach enhances the flexibility of the designs, accommodating a wider range of features, albeit with a lower level of security.

This paper explores the significant hard problems that pertain to the Diffie-Hellman problems by providing reduction relations among them. Although some previous studies, such as [2–5] have touched upon the reduction between some of these problems, there remain some reductions that have not been covered or have been proven in a more complex way.

Furthermore, this paper presents the first comprehensive study undertaken on pairing inversion problems and their relation with other computational bilinear Diffie-Hellman problems.

Also, through our analyses, we can classify a variety of computational problems associated with the Diffie-Hellman problems and a variety of bilinear problems based on their hardness.

The contributions of this paper can be summarized as follows:

- Providing an overview of well-known computational problems relevant to the Diffie-Hellman problem and the reductions between them.
- Presenting the first hard problem that is equivalent to the discrete logarithm, which also has a decisional version.
- Introducing a set of new problems and exploring their reductions that could be useful in constructing secure cryptographic schemes.
- Investigating pairing inversion problems and their relation to other computational bilinear Diffie-Hellman problems.
- Proposing a simpler proof for the equivalency of computational Diffie-Hellman and divisible computational Diffie-Hellman problems compared to the previous proof.

This paper is organized as follows: Section 2 provides essential background information that will be referred to throughout the paper. Section 3 investigates established problems pertaining to the Diffie-Hellman problem, and includes discussions on potential reductions between them. In Section 4, a novel hard problem is presented as an equivalent to the discrete logarithm, and its decisional version is proposed, along with the introduction of new problems and an analysis of their level of complexity. Finally, Section 5 examines the variations of the bilinear pairing problem, accompanied by an evaluation of their level of complexity.

2 Preliminaries

This section introduces the notations used in the paper, followed by a concise overview of the cyclic group, discrete logarithm problem, discrete logarithm problem over elliptic curves, and bilinear pairing.

2.1 Notation

The notation $B \Leftarrow A$ demonstrates that if an oracle exists for solving problem A in polynomial time, then there exists a PPT algorithm that solves problem B, as well. Put simply, B is reduced to A. The notion of reduction serves to establish a fundamental relationship between two problems and is essential in developing potential solutions.

The notation $B \Leftrightarrow A$ denotes that problems A and B are equivalent through reductions. This means that if an oracle is accessible to solve problem A, problem B can also be solved, and vice versa.

An oracle, represented by the symbol $\mathcal{O}_{\approx}(a, b) \rightarrow c$, refers to a computational black box that takes the inputs of problem π (i.e., a, b) and produces its output (i.e., c) of it.

2.2 Cyclic Group

A group can be thought of as a collection of elements, represented by the symbol $\mathbb{G}$, along with an operation, denoted by $*$, that combines any two elements from this collection. A group should have four properties, which are briefly known as closeness, associativity, identity element, and inverse element.

A group $\mathbb{G}$ is cyclic if there is an element $g \in \mathbb{G}$ such that for each $\alpha \in \mathbb{G}$ there exists an integer i such that $\alpha = g^i$. Such an element g is called a generator or primitive element of $\mathbb{G}$ [6]. In a cyclic group, each element acts as the generator of a cyclic subgroup.

2.3 Discrete Logarithm Problem

Let $\mathbb{G}$ be a finite cyclic group of order p, with generator g and $\beta \in \mathbb{G}$. The discrete logarithm (DL) problem is finding the integer a, where $2 \leq a \leq p-1$, such that the following relation is satisfied [7]:

$$\beta = \underbrace{g * g * \cdots * g}_{a \ times} = g^a \tag{1}$$

Therefore, the discrete logarithm problem oracle $\mathcal{O}_{DL}$ is demonstrated as follows:

$$\mathcal{O}_{DL}(g, g^a) \rightarrow a \tag{2}$$

In order to prevent brute-force attacks on discrete logarithm-based cryptosystems in real-world scenarios, it should be ensured that the underlying group's order is sufficiently large. To achieve this, a large prime number $p = 2q + 1$ is chosen, where q is prime. The discrete logarithm problem in the subgroup $\mathbb{G} \subset \mathbb{Z}_p^*$ is assumed to be hard, where $\mathbb{G}$ is a cyclic group of prime order q, with a generator g.

2.4 Discrete Logarithm Problem Over Elliptic Curves

In cryptographic applications, we need to consider the elliptic curve over a finite field, which is defined as follows.

Definition 1 (Elliptic Curve Over Finite Fields). The elliptic curve E over $\mathbb{Z}_p$, where $p > 3$ is prime, is the set of all pairs $(x, y) \in \mathbb{Z}_p^2$ which fulfill

$$E : y^2 \equiv x^3 + ax + b \bmod p \tag{3}$$

together with an imaginary point of infinity P_∞, where $a, b \in \mathbb{Z}_p$, satisfying $4a^3 + 27b^2 \neq 0 \bmod p$.

For an elliptic curve to be defined, it should have no intersections or singular points on its plot, which is achieved if the discriminant of the curve, defined as $-16(4a^3 + 27b^2)$, is nonzero. For convenience, we are dealing with an elliptic curve defined by a short Weierstrass equation, given in (3), because any general Weierstrass equation (4) can be transformed into the short form [8].

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{4}$$

The elliptic curve E over $\mathbb{Z}_p$ is an additive group. The addition law in elliptic curves is as follows. For points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ on E, the line through P and Q (or the tangent line at P, if $P = Q$) intersects E at a third point R. The line through the point at infinity P_∞ and R intersects E at a fourth point, which is defined to be $P + Q$ (or $P + P = 2P$, if $P = Q$). In other words, $P + Q$ is geometrically the reflection of point R across the x-axis.

The number of points on the elliptic curve, defined over the field $\mathbb{Z}_p$, is denoted by $\#E$, which is bounded by Hasse's theorem [9] as follows:

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p} \tag{5}$$

The discrete logarithm problem over the elliptic curve is defined as follows.

Definition 2 (Elliptic Curve Discrete Logarithm Problem – ECDLP). Let E be an elliptic curve group with generator element P. For a given element $T \in E$. The DL problem is finding the integer d, where $1 \leq d \leq \#E$, such that:

$$T = \underbrace{P + P + \cdots + P}_{d \ times} = dP \tag{6}$$

2.5 Bilinear Pairing

Pairing-based cryptography is a large class of cryptography providing solutions for digital signatures, key establishment, functional encryption, attribute-based encryption, etc. It relies on a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, called bilinear pairing over finite groups, such that $\mathbb{G}_1$, $\mathbb{G}_2$ are additive cyclic groups and $\mathbb{G}_T$ is a multiplicative cyclic group, all of prime order q. This bilinear map has the following properties [9]:

(1) Non-degeneracy, which means for all $P \in \mathbb{G}_1 \setminus \{1\}$ and $Q \in \mathbb{G}_2 \setminus \{1\}$, it holds $e(P, Q) \neq 1$.
(2) Bilinearity, which means that $e(aP, bQ) = e(P, Q)^{ab}$ for $a, b \in \mathbb{Z}$,
(3) There is a polynomial-time algorithm to compute $e(P, Q)$.

In cases where $\mathbb{G}_1$ equals $\mathbb{G}_2$, the pairing is considered symmetric, whereas, when $\mathbb{G}_1$ differs from $\mathbb{G}_2$, the pairing is referred to as asymmetric.

Although $\mathbb{G}_1$ and $\mathbb{G}_2$ are typically written in additive notation due to their origin as subgroups of elliptic curves over finite fields, in pairing groups, we often use the multiplicative notation for them.

3 Existing Variations of Diffie-Hellman Problem

This section reviews a collection of prominent computational and decisional problems that are pertinent to the Diffie-Hellman problem. We will also explore how reduction relations can be defined among them.

3.1 Variations of Computational Diffie-Hellman Problem

In the following, we will review different forms of the computational Diffie-Hellman problem, and explore the relationships between them through reduction.

Definition 3 (Computational Diffie-Hellman problem – CDH [1]). Given a triple (g, g^a, g^b) of elements in $\mathbb{G}$ to compute g^{ab}.

$$input : g, g^a, g^b \rightarrow output : g^{ab} \tag{7}$$

Definition 4 (Square Computational Diffie-Hellman – SCDH [10]). Given a pair (g, g^a) of elements in $\mathbb{G}$ to compute g^{a^2}.

$$input : g, g^a \to output : g^{a^2} \tag{8}$$

Definition 5 (Inverse Computational Diffie-Hellman – InvCDH [3]). Given $g, g^a \in \mathbb{G} \setminus \{1\}$ to compute $g^{a^{-1}}$ (Clearly, the case $a = 0$ must be excluded from the set of instances.)

$$input : g, g^a \to output : g^{a^{-1}} \tag{9}$$

Theorem 1 ($InvCDH \Leftrightarrow SCDH \Leftrightarrow CDH$ [2]). InvCDH, SCDH, and CDH problems are equivalent.

Definition 6 (Divisible Computational Diffie-Hellman – DivDH [2]). Given a triple (g, g^a, g^b) of elements in $\mathbb{G}$ to compute $g^{b/a}$.

$$input : g, g^a, g^b \to output : g^{b/a} \tag{10}$$

Theorem 2 ($DivDH \Leftrightarrow CDH$). DivDH and CDH problems are equivalent.

Proof. To prove the theorem, we go through the following steps:

- $(DivDH \Leftarrow CDH)$: Assuming (g, g^a, g^b) as inputs for DivDH problem, with $\mathcal{O}_{CDH}$ acting as a perfect oracle for solving CDH problem. Choose $r_1, r_2, r_3 \in \mathbb{Z}_q^*$, and let $h = g^{r_1 . a}, g^{r_2} = h^{\frac{r_2}{r_1 . a}}, g^{r_3 . b} = h^{\frac{r_3 . b}{r_1 . a}}$. Consequently, $g^{b/a}$ can be computed as follows:
$$\mathcal{O}_{CDH}(h, h^{\frac{r_2}{r_1 . a}}, h^{\frac{r_3 . b}{r_1 . a}}) \to h^{\frac{r_2 . r_3 . b}{(r_1 . a)^2}} = g^{\frac{r_2 . r_3 . b}{r_1 . a}} \tag{11}$$
we can extract $g^{b/a}$ by knowing r_1, r_2, r_3.
- $(CDH \Leftarrow DivDH)$: Assuming (g, g^a, g^b) as inputs for CDH problem, with $\mathcal{O}_{DivDH}$ acting as a perfect oracle for solving DivDH problem. Choose $r_1, r_2, r_3 \in \mathbb{Z}_q^*$, and let $h = g^{r_1 . a}$, $g^{r_2} = h^{\frac{r_2}{r_1 . a}}$, and $g^{r_3 . b} = h^{\frac{r_3 . b}{r_1 . a}}$. Consequently, g^{ab} can be computed as follows:
$$\mathcal{O}_{DivDH}(h, h^{\frac{r_2}{r_1 . a}}, h^{\frac{r_3 . b}{r_1 . a}}) \to h^{\frac{r_3 . b / r_1 . a}{r_2 / r_1 . a}} = g^{\frac{r_1 . r_3 . ab}{r_2}} \tag{12}$$
we can extract g^{ab} by knowing r_1, r_2, r_3. □

This theorem has been previously proven in [2], demanding the invocation of the DivDH and InvCDH oracles twice and once, respectively, for the forward direction $(CDH \Leftarrow DivDH)$ and CDH and InvCDH oracles, each once, for the backward direction $(DivDH \Leftarrow CDH)$. However, the evidence presented in this paper only requires invoking the CDH oracle once for each of the forward and backward directions and excludes the use of the InvCDH oracle in both directions. Hence, our proof is more efficient.

It is worth noting that r_1 and r_2 in the proof of Theorem 2 are serving as the general case. However, they can simply be set to 1. This point holds true throughout the rest of the paper.

3.2 Variations of Decisional Diffie-Hellman Problem

In this section, we will provide an overview of two significant variations of the decisional Diffie-Hellman problem. This problem is used in designing cryptographic schemes and protocols [11, 12]

Definition 7 (Decisional Diffie-Hellman problem – DDH [13]). Given a quadruple (g, g^a, g^b, g^c) of elements in $\mathbb{G}$ to determine whether or not $g^c = g^{ab}$.

$$input : g, g^a, g^b, g^c \to output : \begin{cases} 1 & if\ g^c = g^{ab} \ , \\ 0 & otherwise. \end{cases} \tag{13}$$

Definition 8 (Inverse Decisional Diffie-Hellman – InvDDH [2]). Given a triple (g, g^x, g^z) to determine whether or not $g^z = g^{\frac{1}{x}}$.

Theorem 3 ($InvDDH \Leftarrow DDH$ [2]). Having an oracle capable of solving the DDH problem enables one to solve the InvDDH problem. In short, InvDDH is at most as hard as DDH problem.

4 Introducing New Problems

In this section, we aim to present a novel hard problem, called the Discrete Logarithm Diffie-Hellman (DLDH) problem, that is of equivalent hardness to the discrete logarithm problem. The main significance of DLDH problem, making it suitable for distinguishing proofs, is that despite the DL problem, it possesses a decisional variant. To the best of our knowledge, this is the first DL-equivalent hard problem, supporting the decisional variant. Moreover, we establish the hardness of its decisional version by demonstrating its equivalence to the InvDDH problem. In addition, we propose five new problems that can serve as useful tools in designing secure cryptographic schemes. We also conduct an extensive analysis of the hardness of these introduced problems.

4.1 A DL-Equivalent Problem and Its Decisional variant

The DLDH problem, which is defined below, closely resembles the Diffie-Hellman problem; however, it returns ab as its output instead of g^{ab}.

Definition 9 (Discrete Logarithm Diffi-Hellman Problem – DLDH). Given a triple (g, g^a, g^b) of elements in $\mathbb{G}$ to compute ab.

$$input : g, g^a, g^b \rightarrow output : ab \tag{14}$$

Theorem 4 **(**$DLDH \Leftrightarrow DL$**).** DLDH and DL problems are equivalent.

Proof. To prove the theorem, we go through the following steps:

- ($DLDH \Leftarrow DL$): Suppose g, g^a, g^b are the inputs of the $DLDH$ problem, and $\mathcal{O}_{DL}$ is a perfect oracle that solves the DL problem. Choose $r_1, r_2 \in \mathbb{Z}_q^*$. Then call the oracle $\mathcal{O}_{DL}$ on (g, g^{ar_1}) and (g, g^{br_2}) to get ar_1 and br_2:

  $$\mathcal{O}_{DL}(g, g^{ar_1}) \rightarrow ar_1 \tag{15}$$

  $$\mathcal{O}_{DL}(g, g^{br_2}) \rightarrow br_2 \tag{16}$$

  It follows that ab can be computed by knowing r_1 and r_2.
- ($DL \Leftarrow DLDH$): Suppose g, g^a are the inputs of the DL problem and $\mathcal{O}_{DLDH}$ is a perfect oracle that solves the $DLDH$ problem. Choose $r_1, b \in \mathbb{Z}_q^*$, then call the oracle $\mathcal{O}_{DLDH}$ on (g, g^{ar_1}, g^b):

  $$\mathcal{O}_{DLDH}(g, g^{ar_1}, g^b) \rightarrow ar_1 b \tag{17}$$

  Knowing $r_1 b$ enables the computation of a. So, if one has an oracle for solving the DL problem, one can solve the DLDH problem and vice versa. Hence, these two problems are equivalent. □

Definition 10 (Decisional DLDH – DDLDH). Given a triple (g, g^a, g^b) of elements in $\mathbb{G}$ and $c \in \mathbb{Z}_q^*$ to determine whether or not $c = ab$.

$$input : g, g^a, g^b, c \rightarrow output : \begin{cases} 1 & if\ c = ab\ , \\ 0 & otherwise. \end{cases} \tag{18}$$

Theorem 5 **(**$DDLDH \Leftrightarrow InvDDH$**).** The Decisional DLDH problem is equivalent to the InvDDH problem.

Proof. To prove the theorem, we go through the following steps:

- ($InvDDH \Leftarrow DDLDH$): Suppose (g, g^a, g^z) are the inputs of the InvDDH problem and $\mathcal{O}_{DDLDH}$ is a perfect oracle for solving the DDLDH problem. Choose r_1, r_2, and c randomly and let $h = g^{r_1}$. Next, call $\mathcal{O}_{DDLDH}$ on the quadruple $(h, h^{r_2 a}, h^{\frac{z.c}{r_2}}, c)$. If $z = a^{-1}$, then this oracle outputs 1. Because in this situation we have $(r_2 a)(\frac{z.c}{r_2}) = c$.
- ($DDLDH \Leftarrow InvDDH$): Suppose (g, g^a, g^b, c) are the inputs of the DDLDH problem and $\mathcal{O}_{InvDDH}$ is an oracle to solve InvDDH. Choose r_1, r_2, and c randomly and let $h = g^{r_1}$. Next, invoke $\mathcal{O}_{InvDDH}$ on the triple $(h, h^{r_2 a}, h^{\frac{b}{c.r_2}})$. If $c = ab$, then this oracle outputs 1. Because in this situation we have $\frac{b}{c.r_2} = (r_2 a)^{-1}$. □

The innovation of DDLDH problem is twofold. Firstly, it is the first decisional version of a DL-equivalent problem. Secondly, it translates the problem of comparing two elements in group $\mathbb{G}$ into the comparison of numbers in $\mathbb{Z}_q^*$.

4.2 Some New Variations of DH Problem

In this section, we present five novel hard problems derived from the DH problem and explore their respective levels of difficulty.

Definition 11 (Weighted InvDH (WInvDH)). Given a pair (g, g^a) of elements in $\mathbb{G}$ and $c \in \mathbb{Z}_q^*$ to compute $g^{\frac{c}{a}}$.

$$input : g, g^a, c \rightarrow output : g^{\frac{c}{a}} \tag{19}$$

As it will be proved in Theorem 6, Definitions 5 and 11 are equivalent. However, we defined the WInvDH problem as an independent problem with two distinct purposes in mind. Firstly, it serves as a warm-up exercise to acquaint oneself with the proofs that will be presented in this section. Secondly, it is important to highlight that the output of WInvDH can be considered as the last input for DLDH (14), and vice versa, i.e. WinvDH receives c and outputs $g^{c/a}$, while DLDH receives $g^{c/a}$ and outputs c.

Theorem 6 **(**$WInvDH \Leftrightarrow InvCDH$**).** The WInvDH problem is equivalent to the InvCDH problem.

Proof. To prove the theorem, we go through the following steps:

- ($WInvDH \Leftarrow InvCDH$): Suppose (g, g^a, ab) are the inputs of the WInvDH and $\mathcal{O}_{InvCDH}$ is a perfect oracle for solving *InvCDH*. Choose $r \in \mathbb{Z}_q^*$ and compute $g^{\frac{ra}{ab}} = g^{\frac{r}{b}}$. Then query $\mathcal{O}_{InvCDH}$ on $(g, g^{rb^{-1}})$ to get $g^{br^{-1}}$.

  $$\mathcal{O}_{InvCDH}(g, g^{rb^{-1}}) \rightarrow g^{r^{-1}b} \tag{20}$$

  So, g^b can be computed by knowing r^{-1}.
- ($InvCDH \Leftarrow WInvDH$): Suppose (g, g^a) are the inputs of the InvCDH and $\mathcal{O}_{WInvDH}$ is a perfect oracle for solving WInvDH. We choose $c \in \mathbb{Z}_q, r \in \mathbb{Z}_q^*$ at random and call $\mathcal{O}_{WInvDH}$ on (g, g^{ar}, c). Thus

  $$\mathcal{O}_{WInvDH}(g, g^{ar}, c) \rightarrow g^{cr^{-1}a^{-1}} \tag{21}$$

Now, we can compute $g^{a^{-1}}$ from knowing cr^{-1}. □

Definition 12 (Additive Hidden DH (AHDH)). To compute $g^{a(b+r)}$ and g^r, given a triple (g, g^a, g^b) of $\mathbb{G}$ elements, where $r \in \mathbb{Z}_q^*$.

$$input : g, g^a, g^b \rightarrow output : g^{a(b+r)}, g^r. \quad (22)$$

Theorem 7 (*AHDH* ⇐ *CDH*). Having an oracle for solving the CDH problem enables one to solve the AHDH problem.

Proof. Assuming that the inputs (g, g^a, g^b) are given for the AHDH, and $\mathcal{O}_{CDH}$ is a perfect oracle for solving CDH problem. We randomly choose $r_1, r_2, r_3, r_4 \in \mathbb{Z}_q^*$, and set $h = g^{r_1}$. Then, call the $\mathcal{O}_{CDH}$ on $(g^{r_1}, g^{ar_2}, g^{br_3+r_4})$:

$$\mathcal{O}_{CDH}(h, h^{a\frac{r_2}{r_1}}, h^{b\frac{r_3}{r_1}+\frac{r_4}{r_1}}) \rightarrow h^{a\frac{r_2}{r_1}(b\frac{r_3}{r_1}+\frac{r_4}{r_1})} = g^{a\frac{r_2r_3}{r_1}(b+\frac{r_4}{r_3})}$$

Knowing $r_1, r_2,$ and r_3, we can easily compute $g^{a(b+\frac{r_4}{r_3})}$ as follows:

$$(g^{a\frac{r_2r_3}{r_1}(b+\frac{r_4}{r_3})})^{(\frac{r_2r_3}{r_1})^{-1}} = g^{a(b+\frac{r_4}{r_3})} \quad (23)$$

Also by knowing r_3 and r_4, we can compute $g^{\frac{r_4}{r_3}}$. Therefore, setting $r = \frac{r_4}{r_3}$, this theorem is proved. □

Definition 13 (Additive Hidden DivDH – AHDivDH). To compute $g^{\frac{(b+r)}{a}}$ and g^r, given a triple (g, g^a, g^b) of $\mathbb{G}$ elements, where $r \in \mathbb{Z}_q^*$.

$$input : g, g^a, g^b \rightarrow output : g^{\frac{(b+r)}{a}}, g^r. \quad (24)$$

Theorem 8 (*AHDH* ⇔ *AHDivDH*). The AHDH and AHDivDH problems are equivalent.

Proof. To prove the theorem, we go through the following steps:

- (*AHDivDH* ⇐ *AHDH*): Assuming that the inputs (g, g^a, g^b) are given for the AHDivDH, and $\mathcal{O}_{AHDH}$ is a perfect oracle for solving AHDH. We randomly choose $r_1, r_2, r_3 \in \mathbb{Z}_q^*$, and set $h = g^{ar_1}$. Then invoke the $\mathcal{O}_{AHDH}$ on $(g^{ar_1}, g^{r_2}, g^{br_3})$:

  $$\mathcal{O}_{AHDH}(h, h^{\frac{r_2}{ar_1}}, h^{b\frac{r_3}{ar_1}}) \rightarrow h^{\frac{r_2}{ar_1}(b\frac{r_3}{ar_1}+r')}, h^{r'} \quad (25)$$

  $$h^{\frac{r_2}{ar_1}(b\frac{r_3}{ar_1}+r')} = g^{\frac{r_2r_3}{r_1}(\frac{b+ar'r_1r_3^{-1}}{a})} \quad (26)$$

  By knowing $r_1, r_2,$ and r_3, we can easily compute $g^{\frac{b+ar'\frac{r_1}{r_3}}{a}}$ as follows:

  $$(g^{\frac{r_2r_3}{r_1}(\frac{b+ar'r_1r_3^{-1}}{a})})^{(\frac{r_2r_3}{r_1})^{-1}} = g^{\frac{b+ar'\frac{r_1}{r_3}}{a}} \quad (27)$$

  Also by knowing r_3, we can compute $(h^{r'})^{\frac{1}{r_3}} = g^{\frac{ar_1r'}{r_3}}$. Therefore, by considering $r = \frac{ar_1r'}{r_3}$, this theorem is proved.

- (*AHDH* ⇐ *AHDivDH*): Assuming that the inputs (g, g^a, g^b) are given for the AHDH and $\mathcal{O}_{AHDivDH}$ is a perfect oracle for solving AHDivDH. We randomly choose $r_1, r_2, r_3 \in \mathbb{Z}_q^*$, and let $h = g^{ar_1}$. Then call the $\mathcal{O}_{AHDivDH}$ on $(g^{ar_1}, g^{r_2}, g^{br_3})$:

  $$\mathcal{O}_{AHDivDH}(h, h^{\frac{r_2}{ar_1}}, h^{b\frac{r_3}{ar_1}}) \rightarrow h^{\frac{b\frac{r_3}{ar_1}+r'}{\frac{r_2}{ar_1}}}, h^{r'} \quad (28)$$

  The first output can be written as $g^{\frac{ar_1r_3}{r_2}(b+\frac{ar_1r'}{r_3})}$, by knowing $r_1, r_2,$ and r_3, we can easily compute $g^{a(b+\frac{ar_1r'}{r_3})}$ as follows:

  $$(g^{\frac{ar_1r_3}{r_2}(b+\frac{ar_1r'}{r_3})})^{(\frac{r_1r_3}{r_2})^{-1}} = g^{a(b+\frac{ar_1r'}{r_3})} \quad (29)$$

  Also by knowing r_3, we can compute $(h^{r'})^{\frac{1}{r_3}} = g^{\frac{ar_1r'}{r_3}}$. Therefore, by considering $r = \frac{ar_1r'}{r_3}$, this theorem is proved. □

Definition 14 (Additive Hidden SDH – AHSDH). To compute $g^{a(a+r)}$ and g^r, given a pair (g, g^a) of $\mathbb{G}$ elements, where $r \in \mathbb{Z}_q^*$.

$$input : g, g^a \rightarrow output : g^{a(a+r)}, g^r. \quad (30)$$

Theorem 9 (*AHSDH* ⇐ *AHDH*). Having an oracle for solving the AHDH (22) enables one to solve the AHSDH.

Proof. Suppose g, g^a are the inputs of the AHSDH problem and $\mathcal{O}_{AHDH}$ is a perfect oracle for solving the AHDH problem. We randomly choose $r_1, r_2, r_3 \in \mathbb{Z}_q^*$, and let $h = g^{r_1}$. Then call the $\mathcal{O}_{AHDH}$ on $(g^{r_1}, g^{ar_2}, g^{ar_2+r_3})$:

$$\mathcal{O}_{AHDH}(h, h^{a\frac{r_2}{r_1}}, h^{\frac{ar_2+r_3}{r_1}}) \rightarrow h^{a\frac{r_2}{r_1}(\frac{ar_2+r_3}{r_1}+r')}, h^{r'} \quad (31)$$

$$h^{a\frac{r_2}{r_1}(\frac{ar_2+r_3}{r_1}+r')} = g^{\frac{ar_2^2}{r_1}(a+\frac{r_3+r_1r'}{r_2})} \quad (32)$$

By knowing r_1 and r_2, we can easily compute $g^{a(a+\frac{r_3+r_1r'}{r_2})}$, as follows:

$$(g^{\frac{ar_2^2}{r_1}(a+\frac{r_3+r_1r'}{r_2})})^{(\frac{r_2^2}{r_1})^{-1}} = g^{a(a+\frac{r_3+r_1r'}{r_2})} \quad (33)$$

Also by knowing r_2, we can compute $(h^{r'})^{\frac{1}{r_2}} = g^{\frac{r_1r'}{r_2}}$. Therefore, by considering $r = \frac{r_3+r_1r'}{r_2}$, this theorem is proved. □

Definition 15 (Hidden Weighted InvDH – HWInvDH). To compute $g^{\frac{r}{a}}$ and g^r, given a pair (g, g^a) of $\mathbb{G}$ elements, where $r \in \mathbb{Z}_q^*$.

$$input : g, g^a \rightarrow output : g^{\frac{r}{a}}, g^r. \tag{34}$$

Theorem 10 $(HWInvDH \Leftarrow AHDivDH)$**.** Having an oracle for solving AHDivDH (24) enables one to solve HWInvDH.

Proof. Assuming that the inputs (g, g^a, g^b) are given for the HWInvDH and $\mathcal{O}_{AHDivDH}$ is a perfect oracle for solving AHDH. We randomly choose $r_1, r_2, r_3 \in \mathbb{Z}_q^*$ and set $h = g^{r_1}$. Next, we call the $\mathcal{O}_{AHDivDH}$ on $(g^{r_1}, g^{ar_2}, g^{r_1 r_3})$:

$$\mathcal{O}_{AHDivDH}(h, h^{\frac{ar_2}{r_1}}, h^{r_3}) \rightarrow h^{\frac{r_3+r'}{ar_2/r_1}} = g^{\frac{1}{a}\cdot\frac{r_1^2 r_3 + r_1^2 r'}{r_2}}, h^{r'} \tag{35}$$

By knowing $r_1, r_2,$ and r_3, we can compute $(h^{r'})^{\frac{r_1}{r_2}} g^{\frac{r_1^2 r_3}{r_2}} = g^{\frac{r_1^2 r' + r_1^2 r_3}{r_2}}$. Therefore, by considering $r = \frac{r_1^2 r' + r_1^2 r_3}{r_2}$, this theorem is proved. □

5 Variations of Bilinear Pairing Problem

Pairing is a momentous aspect of public key cryptography, particularly in Identity-based and Attribute-based Encryption. Therefore, it is essential to have a systematic study and comprehensive understanding of pairing-related problems. This section will examine pairing inversion problems and provide a detailed analysis of them. Afterward, we will address computational bilinear Diffie-Hellman problems and present the reductions that occur between them. This paper does not delve into the hard problems associated with multilinear maps, a generalization of the bilinear pairing. Due to space constraints, we are unable to address these problems here, although some of them have been introduced in [14].

5.1 Pairing Inversion Problems

In the following, we will explore the various forms of the pairing inversion problem.

Definition 16 (Generalized Pairing Inversion – GPI). Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order q and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a non-degenerate bilinear pairing. The Pairing Inversion problem is: Given $\alpha \in \mathbb{G}_T$, to compute $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ such that $\alpha = e(P, Q)$ [15].

$$input : \alpha \rightarrow output : P, Q \ s.t. \ \alpha = e(P, Q) \tag{36}$$

Theorem 11. The GPI problem (36) doesn't have a unique solution. The number of solutions is equal to $q - 1$.

Proof. Suppose that $a.P, b.Q$ is the solution to the *GPI* problem. We can choose any $c \in \mathbb{Z}_q^*$, then compute $a.c.P$, and $\frac{b}{c}.Q$. So, this is also a distinct solution to *GPI* problem. That is clear that the number of solutions is $q - 1$. □

Definition 17 (Fixed Argument Pairing Inversion1 – FAPI1). Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order q, and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ non-degenerate bilinear pairing. The Fixed Argument Pairing Inversion1 problem is: Given $\alpha \in \mathbb{G}_T$ and $P \in \mathbb{G}_1$, to compute $Q \in \mathbb{G}_2$ such that $\alpha = e(P, Q)$ [15].

$$input : \alpha = e(P, Q), P \rightarrow output : Q \tag{37}$$

Theorem 12 $(GPI \Leftarrow FAPI1)$**.** Having an oracle for solving the FAPI1 problem (37) enables one to solve the GPI problem (36).

Proof. Suppose α is the given input of the GPI problem, and $\mathcal{O}_{FAPI1}$ is a perfect oracle for solving FAPI1. Choose $P' \in \mathbb{G}_1$ randomly and call the oracle $\mathcal{O}_{FAPI1}$ on (α, P') to get Q' such that $e(P', Q') = \alpha$.

$$\mathcal{O}_{FAPI1}(\alpha, P') \rightarrow Q' \ s.t. \ e(P', Q') = \alpha \tag{38}$$

So, if one has an oracle for FAPI1 problem, one can solve the GPI problem. □

Definition 18 (Fixed Argument Pairing Inversion2 – FAPI2). Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order q and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ non-degenerate bilinear pairing. The Fixed Argument Pairing Inversion2 problem is: Given $\alpha \in \mathbb{G}_T$ and $Q \in \mathbb{G}_2$, to compute $P \in \mathbb{G}_1$ such that $\alpha = e(P, Q)$ [15].

$$input : \alpha = e(P, Q), Q \rightarrow output : P \tag{39}$$

Theorem 13 $(GPI \Leftarrow FAPI2)$**.** Having an oracle for solving the FAPI2 problem (39) enables one to solve the GPI problem (36).

Proof. The proof is similar to that of Theorem 12. □

This paper specifically examines symmetric bilinear pairing problems, represented by $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. Consequently, the FAPI1 and FAPI2 problems are deemed to be equivalent.

5.2 Computational Bilinear Diffie-Hellman Problem

In the following, we will explore the various forms of the pairing inversion problem.

Definition 19 (Gap Computational Diffie-Hellman Problem – GCDH). A Gap Diffie-Hellman group is a group where the CDH problem is hard, but the DDH problem is easy [16]. Let $\mathbb{G}$ be a group of prime order q generated by P. The GCDH problem is: Given a triple $(P, aP, bP) \in \mathbb{G}$, and a distinguisher $\mathcal{D}$ for the DDH problem to determine abP.

$$input : P, aP, bP, \mathcal{D} \rightarrow output : abP \tag{40}$$

Remark. Let $\mathbb{G}$ be a group over which a symmetric bilinear pairing transformation, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is defined, then $\mathbb{G}$ is necessarily a GCDH group. Let (P, aP, bP, cP) be an instance of the DDH problem. One can simulate distinguisher $\mathcal{D}$ for DDH by computing $e(aP, bP) = e(P, P)^{ab}$ and $e(cP, P) = e(P, P)^{c}$.

Theorem 14 **(*GCDH* $\Leftarrow$ *CDH*).** With an oracle that can solve the CDH problem, one can solve the GCDH problem.

Proof. Suppose P, aP, bP, and $\mathcal{D}$ are the given inputs of the *GCDH* problem, and $\mathcal{O}_{CDH}$ is a perfect oracle for solving the *CDH* problem. Thus, call the oracle $\mathcal{O}_{CDH}$ on (P, aP, bP) to get abP.

$$\mathcal{O}_{CDH}(P, aP, bP) \rightarrow abP \tag{41}$$

So, if one has an oracle for *CDH*, one can solve the *GCDH* problem. □

Definition 20 (Computational Bilinear Diffie-Hellman Problem – CBDH [17]). Given a quadruple (P, aP, bP, cP) of elements in $\mathbb{G}$ to compute $e(P, P)^{abc}$.

$$input : P, aP, bP, cP \rightarrow output : e(P, P)^{abc} \tag{42}$$

There are some variations of this problem, which are given in [18–20].

Theorem 15 **(*CBDH* $\Leftarrow$ *GCDH*).** With an oracle that can solve the GCDH problem (40), one can solve the CBDH problem (42).

Proof. Suppose P, aP, bP, and cP are the given inputs of the CBDH problem, and $\mathcal{O}_{GCDH}$ is a perfect oracle that solves the GCDH problem. $r_1, r_2 \in \mathbb{Z}_q^*$ are randomly chosen. Thus, call the oracle $\mathcal{O}_{GCDH}$ on (P, ar_1P, br_2P, e) to get ar_1br_2P. From knowing r_1r_2, the abP can be computed. Then

$$e(abP, cP) = e(P, P)^{abc} \tag{43}$$

So, if one has an oracle for *GCDH*, one can solve the *CBDH* problem. □

Definition 21 (Computational Diffie-Hellman Problem on $\mathbb{G}_T$ – CDH$_{G_T}$). Given a triple $(e(P, P), e(P, P)^a, e(P, P)^b)$ of elements in $\mathbb{G}_T$ to compute $e(P, P)^{ab}$, where $a, b \in \mathbb{Z}_q$.

$$input : e(P, P), e(P, P)^a, e(P, P)^b \rightarrow output : e(P, P)^{ab} \tag{44}$$

Theorem 16 **(*CBDH* $\Leftarrow$ *CDH*$_{G_T}$).** With an oracle that can solve the CDH_{G_T} problem (42), one can solve the CBDH problem (44).

Proof. Suppose P, aP, bP, and cP are the given inputs of the *CBDH* problem, and $\mathcal{O}_{CDH_{G_T}}$ is a perfect oracle that solves the CDH_{G_T} problem. $r_1, r_2, r_3 \in \mathbb{Z}_q^*$ are randomly chosen. Let $\alpha = e(P, P)^{r_1}, \beta = e(P, cP)^{r_2}$ and $\gamma = e(aP, bP)^{r_3}$. Then, call the oracle $\mathcal{O}_{CDH_{G_T}}$ on (α, β, γ):

$$\mathcal{O}_{CDH_{G_T}}(e(P, P)^{r_1}, e(P, cP)^{r_2}, e(aP, bP)^{r_3}) =$$

$$\mathcal{O}_{CDH_{G_T}}(\alpha, \alpha^{\frac{cr_2}{r_1}}, \alpha^{\frac{abr_3}{r_1}}) \rightarrow \alpha^{\frac{r_2r_3abc}{r_1^2}} = e(P, P)^{\frac{r_2r_3abc}{r_1}}$$

Knowing r_1, r_2, and r_3, we can easily compute $e(P, P)^{abc}$ as follows:

$$(e(P, P)^{\frac{r_2r_3abc}{r_1}})^{\frac{r_1}{r_2r_3}} = e(P, P)^{abc} \tag{45}$$

□

Theorem 17 **(*CDH*$_{G_T}$ $\Leftarrow$ *FAPI*).** With an oracle that can solve the FAPI problem, one can solve the CDH_{G_T} problem (44).

Proof. Suppose that the CDH_{G_T} problem has three given inputs, namely $e(P, P), e(P, P)^a$, and $e(P, P)^b$. Additionally, there exists a perfect oracle, $\mathcal{O}_{FAPI}$, that can solve the FAPI problem. We randomly choose three values, $r_1, r_2, r_3 \in \mathbb{Z}_q^*$. Note that P is a known value. Then, we can call the $\mathcal{O}_{FAPI}$ oracle twice as follows:

$$\mathcal{O}_{FAPI}(r_1P, e(P, P)^{ar_2}) \rightarrow \frac{ar_2}{r_1}P \tag{46}$$

$$\mathcal{O}_{FAPI}(r_1P, e(P, P)^{br_3}) \rightarrow \frac{br_3}{r_1}P \tag{47}$$

We can then proceed to calculate $e(\frac{ar_2}{r_1}P, \frac{br_3}{r_1}P) = e(P, P)^{\frac{abr_2r_3}{r_1^2}}$. With r_1, r_2, and r_3 in hand, computing $e(P, P)^{ab}$ becomes a simple task.

□

Theorem 18 **(*GCDH* $\Leftarrow$ *FAPI*).** With an oracle that can solve the FAPI problem, one can solve the *GCDH* problem (40).

Proof. Suppose that the *GCDH* problem has three given inputs, namely P, aP, and bP. Additionally, there exists a perfect oracle, $\mathcal{O}_{FAPI}$, that can solve the FAPI problem. We randomly choose three values, $r_1, r_2 \in \mathbb{Z}_q^*$. Note that P is a known value. Then, we invoke the $\mathcal{O}_{FAPI}$ on $(r_1P, e(P, P)^{abr_2})$:

$$\mathcal{O}_{FAPI}(r_1P, e(P, P)^{abr_2}) \rightarrow \frac{ar_2b}{r_1}P \tag{48}$$

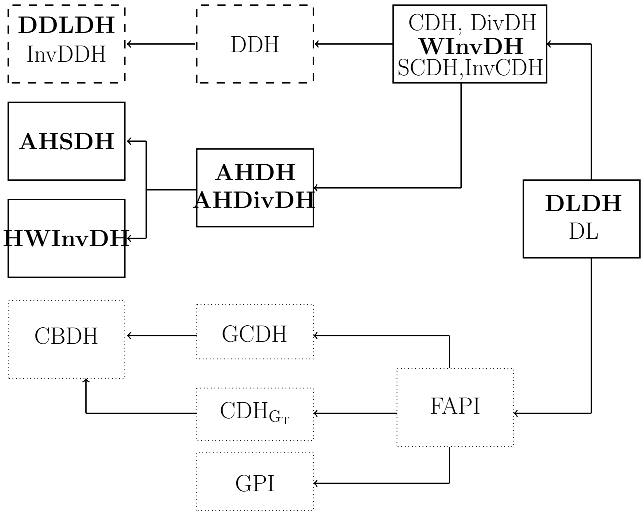

Figure 1. An overwiew of the hardness level for the hard problems explored in this paper

With r_1 and r_2 in hand, computing abP becomes a simple task. □

6 Conclusion

In this paper, we introduced a novel hard problem that is equivalent to the discrete logarithm problem and possesses a decisional variant, a first-of-its-kind achievement. We also proposed five new problems that can aid in the development of secure cryptographic schemes and provided a comprehensive analysis of their hardness. Furthermore, we analyzed various forms of bilinear pairing problems. In Figure 1, we have classified a variety of analyzed computational, decisional, and bilinear problems according to their level of hardness. The notation $B \leftarrow A$ implies that the problem B is at most as hard as problem A. The computational, decisional, and bilinear problems have been distinguished by solid, dashed, and dotted boxes, respectively. Also, the new problems defined in this paper are distinguished in bold. As suggested by the figure, the hardest problems are DLDH and DL, with all other problems branching out from which. Additionally, those problems located in the same box are equivalent in hardness level. We believe that this work will serve as a beneficial resource for researchers and practitioners in the field, and we look forward to further developments and applications of these findings.

References

[1] Whitfield Diffie and Martin E Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 1976.

[2] Feng Bao, Robert H Deng, and Huafei Zhu. Variations of diffie-hellman problem. In *Information and Communications Security: 5th International Conference, ICICS 2003, Huhehaote, China, October 10-13, 2003. Proceedings 5*, pages 301–312. Springer, 2003.

[3] Taiga Mizuide, Atsushi Takayasu, and Tsuyoshi Takagi. Tight reductions for diffie-hellman variants in the algebraic group model. In *Topics in Cryptology–CT-RSA 2019: The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4–8, 2019, Proceedings*, pages 169–188. Springer, 2019.

[4] Naomi Benger, David Bernhard, Dario Catalano, Manuel Charlemagne, David Conti, Biljana Cubaleska, Hernando Fernando, Dario Fiore, Steven Galbraith, David Galindo, et al. Final report on main computational assumptions in cryptography ii. Technical report, ICT-2007-216676 D. MAYA. 6. European Network of Excellence in Cryptology, 2013.

[5] Jung Hee Cheon. Security analysis of the strong diffie-hellman problem. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 1–11. Springer, 2006.

[6] P Nigel Smart. *Cryptography made simple.* Springer, 2016.

[7] Christof Paar and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners.* Springer Science & Business Media, 2009.

[8] Robert Granger and Antoine Joux. Computing discrete logarithms. Cryptology ePrint Archive, Paper 2021/1140, 2021. https://eprint.iacr.org/2021/1140.

[9] Steven D Galbraith. *Mathematics of public key cryptography, Version 2.0.* Cambridge University Press, 2018.

[10] Kannan Balasubramanian. Variants of the diffie-hellman problem. In *Algorithmic Strategies for Solving Complex Problems in Cryptography*, pages 40–54. IGI Global, 2018.

[11] Yael Tauman Kalai, Alex Lombardi, and Vinod Vaikuntanathan. Snargs and ppad hardness from the decisional diffie-hellman assumption. In *Advances in Cryptology- EUROCRYPT 2023: Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 470–498. Springer, 2023.

[12] Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential ddh. In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I*, pages 3–32. Springer, 2021.

[13] Gabor Ivanyos, Antoine Joux, and Miklos Santha. Discrete logarithm and diffie-hellman problems in identity black-box groups. *arXiv preprint arXiv:1911.01662*, 2019.

[14] Mahdi MahdaviOliaee and Zahra Ahmadian. Fine-grained flexible access control: ciphertext policy attribute based encryption for arithmetic circuits. *Journal of Computer Virology and Hacking Techniques*, pages 1–14, 2022.

[15] Steven Galbraith, Florian Hess, and Frederik Vercauteren. Aspects of pairing inversion. *IEEE Transactions on Information Theory*, 54(12):5719–5728, 2008.

[16] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *Journal of cryptology*, 17:297–319, 2004.

[17] Mohammad Ali. Attribute-based remote data auditing and user authentication for cloud storage systems. *ISeCure*, 14(3), 2022.

[18] Sina Abdollahi, Javad Mohajeri, and Mahmoud Salmasizadeh. Highly efficient and revocable cp-abe with outsourcing decryption for iot. In *2021 18th International ISC Conference on Information Security and Cryptology (ISCISC)*, pages 81–88. IEEE, 2021.

[19] Ming Luo and Yuwei Wan. An enhanced certificateless signcryption in the standard model. *Wireless Personal Communications*, 98:2693–2709, 2018.

[20] Han-Yu Lin and Yao-Min Hung. An improved proxy re-encryption scheme for iot-based data outsourcing services in clouds. *Sensors*, 21(1):67, 2020.

Mahdi Mahdavi Oliaee received his B.Sc. degree in Electrical Engineering-Electronic from Urmia University in 2015 and his M.Sc. degree in Cryptography and Secure Communications from Sharif University of Technology, Tehran, Iran, in 2017. He is currently a Ph.D. candidate in the Electrical Engineering Department at Shahid Beheshti University, Tehran, Iran. His research interests include Public Key Cryptography and Cryptographic Protocols. While pursuing his Ph.D., he acted as a visiting researcher at KU Leuven within the COSIC group, under the leadership of Bart Preneel.

Sahar Khaleghifard obtained her B.Sc. degree in Electrical Engineering (Communication) from Shahid Beheshti University in Tehran, Iran in 2018, followed by her M.Sc. degree in Electrical Engineering (Communication Systems) from the same institution focusing on Integral Cryptanalysis of Symmetric Encryptions in 2021. Her research interests include the Cryptanalysis of Symmetric Encryption and Network Security.

Zahra Ahmadian received her B.Sc. degree in Electrical Engineering (Communications and Electronics) from Amirkabir University of Technology, Tehran, Iran, in 2006, and the M.Sc. degree in Electrical Engineering (Secure Communications) and Ph.D. degree in Electrical Engineering (Communication Systems) both from Sharif University of Technology, Tehran, in 2008 and 2014 respectively. Since 2014, she has been with the Electrical Engineering Department of Shahid Beheshti University, Tehran, Iran, as an assistant professor. Her special fields of interest include Wireless Security and Cryptology with an emphasis on Cryptanalysis.