The ISC Int'l Journal of Information Security

ISeCure

July 2023, Volume 15, Number 2 (pp. 260–273)

http://www.isecure-journal.org

Oblivious Transfer Using Generalized Jacobian of Elliptic Curves

Maryam Rezaei Kashi [1], and Mojtaba Bahramian [1,*]

[1] *Department of Pure Mathematics, Faculty of Mathematical Sciences, University of Kashan, Kashan, Iran*

ARTICLE INFO.

Article history:
Received: April 6, 2022
Revised: November 11, 2022
Accepted: June 7, 2023
Published Online: July 1, 2023

Keywords:
Elliptic Curves, Generalized Jacobians, Oblivious Transfer, t-out-of-k Oblivious Transfer

Type: Research Article

doi: 10.22042/isecure.2023.336301.779

dor: 20.1001.1.20082045.2023.15.2.4.3

ABSTRACT

Oblivious transfer (OT) is one of the essential tools in cryptography, in which a sender sends a message to a receiver with a probability between 0 and 1. In contrast, the sender remains oblivious that the receiver has received the message. t-out-of-k oblivious transfer (OT_k^t) is a variant of OT schemes in which a sender transfers k messages to a receiver, but the receiver can only learn t of them. Moreover, the sender remains oblivious to which secrets the receiver has extracted. In this paper, We offer several novel protocols for secure communication using elliptic curve cryptography. First, we propose a new type of Diffie-Hellman key exchange protocol that utilizes the generalized Jacobian of elliptic curves. Next, we introduce simple and secure two-round algorithms for several variants of OT schemes, including OT, OT_2^1, and OT_k^t. The security of our proposed protocols relies on the intractability assumption of solving the discrete logarithm problem. Furthermore, in our OT schemes, it is unnecessary to map the messages to the points on the elliptic curve, which reduces the computational overhead and improves the efficiency of the protocols.

1 Introduction

Oblivious transfer (OT) is an essential cryptographic tool widely used for designing a safe protocol to secure computation. The concept of OT was proposed in 1981 by Rabin [1]. In an OT scheme, a sender sends a message to a receiver such that the receiver receives it with a fixed probability between 0 and 1. In contrast, the sender remains oblivious that the receiver has received the message. In 1985, Even *et al.* [2] generalized OT to 1-out-of-2 OT (OT_2^1), in which the sender transfers two messages to the receiver such that the receiver receives only one of them, while the sender's chance to know which one is obtained by the receiver must be 1/2. Brassard *et al.* [3] extended OT_2^1 to 1-out-of-k OT (OT_k^1), where the sender holds k messages, and the receiver selects one of them to obtain without revealing his choice. The natural generalization of OT_k^1 is t-out-of-k OT (OT_k^t). In OT_k^t the sender sends k messages to the receiver such that the receiver can obtain only t ($t < k$) of them, and the sender does not know which of them are given by the receiver [4–8].

The oblivious transfer has found many applications in cryptography, such as the electronic signing of contracts [2, 9], playing mental games [10], two-party and multi-party secure computation [11, 12], and privately retrieving information from database [13–15]. For example, if a database has k secrets, then an OT_k^t scheme can allow a user to obtain t of them such that the database manager cannot extract which t secrets learned by the user.

Various papers exploit key exchange protocols sim-

* Corresponding author.
**This article is an extended/revised version of an ISCISC'18 paper.
Email addresses: mrezaei.k@grad.kashanu.ac.ir, bahramianh@kashanu.ac.ir
ISSN: 2008-2045

ilar to the Diffie-Hellman to build oblivious transfers [16, 17]. Key exchange protocol is a method in cryptography by which a cryptographic key is shared securely between two parties in a public channel. In 1976, Diffie and Hellman [18] proposed a key exchange algorithm that relies on the discrete logarithm problem, called the Diffie-Hellman key exchange. The elliptic curve Diffie-Hellman key exchange uses elliptic curve point multiplication and is very similar to the classical Diffie-Hellman key exchange.

In this paper, we propose a new type of Diffie-Hellman key exchange on a particular group of the generalized Jacobian of elliptic curves introduced by Dechene [19], and we exploit the generalized Jacobian of elliptic curves to propose simple two-round oblivious transfer schemes.

1.1 Related works

Rabin [1] proposed the idea of the OT based on the RSA cryptosystem for solving the problem of the mutual exchange of messages between two distrustful parties without a trusted third party and the simultaneous transfer of them. In 2006, Parakh [20] proposed a protocol for oblivious transfer using elliptic curve Diffie-Hellman key exchange and used it to build an OT_2^1 protocol. In the schemes, the parties must communicate three times in a public channel (a three-round protocol), and the receiver's security is compromised [6]. Moreover, mapping messages to points on the elliptic curve is necessary.

In recent years, OT_2^1 protocols have received increasing attention in classical and post-quantum cryptography systems [21–24]. Chou and Orlandi [16] introduced a simple OT_2^1 protocol using Diffie-Hellman key exchange. It is one of the most efficient OT protocols in the written works. In 2017, Hauck and Loss [17] proposed a more secure OT protocol under the computational DH assumption. The security of the OT_2^1 protocols proposed in [16, 17] relies on the computational hardness of the discrete logarithm problem. In 2022, Esmaeilzade *et al.* [25] extended the results of [16, 17] to propose a generic construction based on various cryptosystems to build simple, secure, and efficient OT_2^1 protocols. The OT_2^1 protocols proposed in [16, 17, 25] are three-round protocols.

1.2 Our motivation

The protocol constructed by Parakh for oblivious transfer relies on the intractability assumption of solving the discrete logarithm problem. The scheme needs three rounds of communication, while lower communication cost creates a more efficient protocol. Also, it needs to map the messages to the points on the elliptic curve, which has always been challenging.

In this paper, first, we propose a new type of Diffie-Hellman key exchange protocol using the generalized Jacobian of elliptic curves and introduce a simple, secure, and efficient two-round oblivious transfer scheme. Unlike Parakh's scheme, in our OT scheme, it is unnecessary to map the messages to the points on the elliptic curve. Next, we extend the proposed OT scheme to a two-round OT_2^1 protocol. Finally, we generalize our proposed OT_2^1 protocol to a simple and secure OT_k^t protocol.

1.3 Our contributions

The contributions of this paper are summarized as follows:

- We propose a new type of elliptic curve Diffie-Hellman key exchange protocol using generalized Jacobian of elliptic curves, in which its security relies on the hardness of solving the discrete logarithm problem.
- Next, we use the proposed key exchange protocol to introduce a simple and secure OT protocol based on the intractability assumption of solving the discrete logarithm problem on the generalized Jacobian of elliptic curves. Because the computations are in the generalized Jacobian of elliptic curves, it is unnecessary to map the messages to points on the elliptic curve.
- Then, we extend the proposed OT protocol to secure and simple OT_2^1 and OT_k^t protocols. The proposed protocols are secure against all passive attacks.
- We provide a comparison between the introduced protocols and some other protocols.
- Finally, we present a few codes related to the generalized Jacobian of an elliptic curve. These codes are designed to facilitate the calculation of various parameters and values used in the protocols.

1.4 Organization of the paper

The remainder of this paper is organized as follows: Section 2 provides details preliminaries of elliptic curves and the generalized Jacobian of elliptic curves. We present the definition and security model of oblivious transfer and some flavors of oblivious transfer in Section 3. In Section 4, we present a digital signature using the generalized Jacobian of elliptic curves to protect our OT protocols against man-in-the-middle attacks. We propose our schemes and examine their security in Section 5. In Section 6, we compare our proposed OT protocols and some other protocols. Finally, we draw our implementation achievement and

Pari codes in Section 7.

2 Preliminaries

In this section, first, we provide a brief overview of elliptic curves. Next, we introduce the generalized Jacobian of elliptic curves and give some of their properties.

2.1 Elliptic curves

Let $\mathbb{F}_q$ be the finite field of order $q = p^n$, where n is a positive integer, and p is a prime not equal to 2 and 3. An elliptic curve E over $\mathbb{F}_q$ is a curve defined by an equation in the form

$$y^2 = x^3 + ax + b,$$

where $a, b \in \overline{\mathbb{F}}_q$, and its discriminant $\Delta = 4a^3 + 27b^2$ is nonzero. If $a, b \in \mathbb{F}_q$, then we say that E is defined over $\mathbb{F}_q$; in this case, the set of $\mathbb{F}_q$-rational points of E is defined as

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\},$$

where $\mathcal{O}$ is the point at infinity. The points on the elliptic curve E form an abelian group with respect to an operation defined as follows, with $\mathcal{O}$ as the identity element.

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on E in which $P_1, P_2 \neq \mathcal{O}$. We put $P_1 + P_2 = P_3 = (x_3, y_3)$ where,

(1) If $x_1 = x_2$ and $y_1 \neq y_2$, then $P_3 = \mathcal{O}$, that means $P_2 = -P_1$.
(2) If $x_1 \neq x_2$, then $x_3 = m^2 - x_2 - x_1$ and $y_3 = m(x_1 - x_3) - y_1$, where $m = (y_2 - y_1)/(x_2 - x_1)$.
(3) If $P_1 = P_2$ and $y_1 \neq 0$, then $x_3 = m^2 - 2x_1$ and $y_3 = m(x_1 - x_3) - y_1$, where $m = (3{x_1}^2 + a)/(2y_1)$.
(4) If $P_1 = P_2$ and $y_1 = 0$, then $P_3 = \mathcal{O}$.
(5) $P + \mathcal{O} = \mathcal{O} + P = P$, for each point P on E.

Further information on the elliptic curves can be obtained from a variety of sources, including books and research papers. Two important references in this area are [26, 27].

2.2 Usual and generalized Jacobians

The generalized Jacobian of a curve C is a commutative algebraic group related to the curve C with an effective divisor on C, introduced by Rosenlicht [28] in 1954. Dechene [19] introduced the use of the generalized Jacobian of elliptic curves as a candidate for discrete logarithm-based cryptography. This paper uses the generalized Jacobians to propose a key exchange protocol and apply it in several different elliptic curve oblivious transfer versions.

Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over the finite field $\mathbb{F}_q$. The divisor group of E, denoted by $Div(E)$, is the free abelian group consisting of formal sums $\sum_{P\in E} n_P(P)$, where the n_P's are integers, all but finitely many of n_P's will equal to 0. Given any divisor $D = \sum_{P\in E} n_P(P)$, the degree and the sum of D are defined as $deg(D) = \sum_{P\in E} n_P$ and $sum(D) = \sum_{P\in E} n_P P$, respectively. Furthermore, the support of D is the set $Supp(D) = \{P \in E | n_P \neq 0\}$. The divisors D and D' are called co-prime, if $Supp(D) \cap Supp(D^{'}) = \emptyset$. The subgroup of $Div(E)$ contains all divisors of degree 0 and is denoted by $Div^0(E)$. A divisor $D = \sum_P n_P(P)$ is called effective, if $n_P \geq 0$ for each $P \in Supp(D)$. For a non-zero function f in $\overline{\mathbb{F}}_q(E)$, the function field of E, divisor of f is $div(f) = \sum_{P\in E} ord_P(f)(P)$, where $ord_P(f)$ is the order of f at the point P. A divisor D is called principal, if $D = div(f)$, for a function f in $\overline{\mathbb{F}}_q(E)$. The set of all principal divisors of E is denoted by $Princ(E)$. Two divisors D and D' on E are said to be linearly equivalent, denoted by $D \sim D'$, if they differ by a principal divisor, i.e., $D - D' = div(f)$ for some non-zero function f on E. Principal divisors on E have degree zero, which means that they are divisors of functions on E that have no poles or zeros. The quotient group of degree-zero divisors modulo principal divisors, denoted by $Pic^0(E) = \frac{Div^0(E)}{Princ(E)}$, is called the (usual) Jacobian of E. Let $\mathfrak{m} = \sum_{i=1}^{r} m_i(P_i)$ be an effective divisor with the support $S_{\mathfrak{m}} = \{m_1, \cdots, m_r\}$. Suppose D and $D^{'}$ are two divisors relatively prime with respect to $\mathfrak{m}$. We say that D and D' are $\mathfrak{m}$-equivalent, denoted by $D \sim_{\mathfrak{m}} D'$, if they are linearly equivalent ($D - D' = div(f)$) and satisfy the condition $ord_{P_i}(1 - f) \geq m_i$ for $1 \leq i \leq r$. The $\mathfrak{m}$-equivalence class of divisor D is denoted by $[D]_{\mathfrak{m}}$. The set of all divisors of E relatively prime to $\mathfrak{m}$ is denoted by $Div_{\mathfrak{m}}(E)$, and the degree zero part of $Div_{\mathfrak{m}}(E)$ is denoted by $Div^0_{\mathfrak{m}}(E)$. We define

$$\begin{aligned} &Princ_{\mathfrak{m}}(E) = [o]_{\mathfrak{m}} \\ &= \left\{ div(f) | f \in \overline{\mathbb{F}}_q(E)^*, ord_P(1 - f) \geq m_i, (1 \leq i \leq r) \right\}. \end{aligned}$$

The quotient group $\dfrac{Div^0_{\mathfrak{m}}(E)}{Princ_{\mathfrak{m}}(E)}$ is called the generalized Jacobian of E with respect to $\mathfrak{m}$, and denoted by $Pic^0_{\mathfrak{m}}(E)$ or $J_{\mathfrak{m}}(E)$. Dechene in [19] proved that if M and N are two distinct non-zero points on E, then the generalized Jacobian of E can be represented as

$$J_{\mathfrak{m}}(E) = \left\{ (k, P) : k \in \mathbb{F}_q^*, P \in E(\mathbb{F}_q) \right\},$$

and given $(k_1, P_1), (k_2, P_2) \in J_{\mathfrak{m}}(E)$ such that $P_1, P_2, \pm(P_1 + P_2) \notin \{M, N\}$, we have

$$(k_1, P_1) + (k_2, P_2) = (k_1 k_2 c_{\mathfrak{m}}(P_1, P_2), P_1 + P_2), \quad (1)$$

where $c_{\mathfrak{m}}(P_1,P_2)=\frac{L_{P_1,P_2}(M)}{L_{P_1,P_2}(N)}$, and $L_{P_1,P_2}\in\overline{\mathbb{F}}_q(E)^*$ is such that $div(L_{P_1,P_2})=(P_1)+(P_2)-(P_1+P_2)-(\mathcal{O})$. The basic properties of the group law include the following:

(1) $(1,\mathcal{O})$ is the identity element of $(J_{\mathfrak{m}},+)$.
(2) $\mathbf{c}_{\mathfrak{m}}(P_1,P_2)=\mathbf{c}_{\mathfrak{m}}(P_2,P_1)$, therefore $J_{\mathfrak{m}}$ is an abelian group.
(3) The inverse of $(k,P)\in J_{\mathfrak{m}}$ is $-(k,P)=\left(\frac{1}{k}\frac{l_{P,\mathcal{O}}(N)}{l_{P,\mathcal{O}}(M)},-P\right)$, where $l_{P,\mathcal{O}}$ denotes the equation of the straight line passing through P and $\mathcal{O}$.
(4) For all $P\in E\setminus\{M,N\}$, $c_{\mathfrak{m}}(\mathcal{O},P)=1$. Therefore, $(k_1,\mathcal{O})+(k_2,P)=(k_1k_2,P)$ and in particular $(k_1,\mathcal{O})+(k_2,\mathcal{O})=(k_1k_2,\mathcal{O})$, as a result $J_{\mathfrak{m}}$ contains a subgroup isomorphic to $\mathbb{F}_q^*$.

Now, for the integer n, and $(1,P)\in J_{\mathfrak{m}}(E)$, let $\alpha_n(P)$ be the first component of $n(1,P)$. Then, we have

$$\alpha_n(P)=c_{\mathfrak{m}}(P,P)c_{\mathfrak{m}}(P,2P)\cdots c_{\mathfrak{m}}(P,(n-1)P),$$

also, if $(k,P)\in J_{\mathfrak{m}}$, then $n(k,P)=(k^n\alpha_n(P),nP)$. The following theorem, which is proved in [29] is used in our proposed scheme.

Theorem 1. *Let m and n be integers, and P be a point on E. Then*

a) $\alpha_{m+n}(P)=\alpha_m(P)\alpha_n(P)c_{\mathfrak{m}}(mP,nP)$,
b) $\alpha_{mn}(P)=\alpha_n(P)^m\alpha_m(nP)=\alpha_m(P)^n\alpha_n(mP)$.

3 Oblivious transfer

In this section, we present the definition and the security model of oblivious transfer and the flavors of oblivious transfer. In all schemes presented in this paper, there are two characters: Alice (the sender) and Bob (the receiver). We assume that both parties are willing to participate honestly in the protocol.

3.1 Definition of OT

In an oblivious transfer, Alice transfers a message to Bob such that Bob receives it with a fixed probability p, and Alice remains oblivious to whether or not Bob received the message. In this paper, we assume that $p=1/2$. At the end of the protocol, Bob must check whether he has received Alice's private key.

Correctness: An OT protocol is correct if Bob achieves the messages with the probability of $1/2$ when Alice and Bob follow the protocol steps.

3.2 The security model of OT

According to the definition, the security requirements of OT are:

Receiver's privacy: The sender should not be able to know whether or not the receiver received the message.
Sender's security: In an OT protocol, Bob can obtain Alice's message only with the probability $1/2$. Equivalently, Bob cannot realize any information about Alice's message with the probability of $1/2$.

3.3 Definition of OT_2^1

In a chosen 1-out-of-2 oblivious transfer, Alice sends two messages, denoted by m_0 and m_1, and Bob chooses a bit c without revealing which message he has chosen. Additionally, the protocol requires that Bob should not know what has been extracted by Alice.

Correctness: An OT_2^1 protocol is correct if Bob achieves one of the messages m_0,m_1 of his choice c, when Alice and Bob follow the protocol steps.

3.4 The security model of OT_2^1

Security requirements of OT_2^1 are:

Receiver's privacy: Alice should not be able to realize Bob's choice. Specifically, she should not be able to obtain whether $c=0$ or $c=1$, which means $Pr[c=0]=Pr[c=1]=1/2$.
Sender's security: In this protocol, the receiver should be unable to distinguish any information about s_{1-c}.

3.5 Definition of OT_k^t

The most commonly used form is t-out-of-k OT (OT_k^t), in which Alice sends k messages $m_1,\ldots,m_k$ to Bob and Bob selects $i_1,\ldots,i_t$ to receive $m_{i_1},\ldots,m_{i_t}$, such that Bob can learn only t of k messages and Alice does not understand which of them have been extracted by Bob.

Correctness: An OT_k^t protocol is correct if Bob achieves the messages of his choices when Alice and Bob follow the steps of the protocol.

3.6 The security model of OT_k^t

Security requirements of OT_k^t are:

Receiver's privacy: In throughout of an OT_k^t Alice should not be able to realize Bob's choice set $B=\{i_1,\ldots,i_t\}$. In other words, given any two sets of choices, B_0 and B_1, she should not be able to obtain whether $B=B_0$ or $B=B_1$.
Sender's security: In this protocol, the receiver should be unable to distinguish any information about m_j for all j such that $j\neq i_1,\ldots,i_t$.

4 A digital signature using generalized Jacobian of elliptic curves

In this section, we use the elliptic curve digital signature algorithm (ECDSA) [30] and present a digital signature using the generalized Jacobian of elliptic curves to protect our *OT* protocols against man-in-the-middle attacks.

A certificate authority (CA) decides upon an elliptic curve $E : y^2 = x^3+ax+b$ over a finite field $\mathbb{F}_q$ ($q = p^n$, p is a prime), and a divisor $\mathfrak{m} = (M) + (N)$ where M and N are distinct finite points on E. He chooses a point $P \in E$ such that the points M and N are not in the subgroup generated by P, denoted by $\langle P\rangle$. After choosing P, a point G is initialized as $(1, P)$. Further, he considers a hash function $hash : \{0,1\}^* \longrightarrow \mathbb{U}_n$ where $\mathbb{U}_n = \{1 \leq x < n : \gcd(x, n) = 1\}$, and n is the order of G. He chooses a private key $k_{CA} \in \mathbb{U}_n$ and publishes $\{E, \mathbb{F}_q, \mathfrak{m}, G, n, hash\}$. Suppose Alice wants to sign a message m. Alice authenticates herself to the CA by her identity $ID_A \in \mathbb{U}_n$. The CA computes and sends $k_A = k_{CA}ID_A \pmod n$ to Alice. K_A is Alice's private key. The CA publishes $A = k_AG$ as "Alice's public key is A, signed CA." Once Alice has received her private and public keys, she performs a series of steps to sign a message m.

(1) calculates $h = hash(m)$,
(2) chooses a random integer c coprime to n, and calculates $r_m = \alpha_c(P)$,
(3) computes $s_m = c^{-1}(h + r_mk_A) \pmod n$; if $\gcd(s_m, n) \neq 1$, then returns to step 2,
(4) puts $sign_A(m) = (r_m, s_m)$, and sends $\{m, sign_A(m)\}$ to Bob.

To verify the signed message, Bob performs the following steps:

(1) calculates $h = hash(m)$, $s_m^{-1} \pmod n$, and $R = (hs_m^{-1})G + (r_ms_m^{-1})A$,
(2) puts $r'_m =$ the first coordinate of R,
(3) verifies the validity of the signature by checking that $r'_m = r_m$.

Remark 1. Since the CA publishes A as "Alice's public key is A, signed the CA," MITM cannot generate such a certificate. The CA will not give Alice's certificate to MITM and lists Alice as the owner. The MITM cannot reuse Alice's certificate, as he does not know Alice's private key, and MITM cannot generate his certificate, as he does not know CA's private key.

5 Our schemes

5.1 Key agreement protocol using generalized Jacobian of elliptic curves

This section uses the generalized Jacobian of an elliptic curve and presents an algorithm for securely exchanging a key between two parties, Alice and Bob, over a public channel.

Alice and Bob decide upon an elliptic curve $E : y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_q$ such that $q = p^n$ and p is a large enough prime number ($p \approx 2^{160}$) and a divisor $\mathfrak{m} = (M)+(N)$, where M and N are distinct finite points on the elliptic curve. They choose a point $P \in E$ such that $M, N \notin \langle P\rangle$. Alice and Bob carry out the following steps for exchanging a key using the generalized Jacobian of an elliptic curve:

(1) Alice chooses an integer a, computes $G_A = a(1, P) = (\alpha_a(P), aP)$, and sends $\{G_A, sign_A(G_A)\}$ to Bob.
(2) Bob verifies the validity of the signature, chooses an integer b and computes $K_B = \alpha_a(P)^b\alpha_b(aP)$, $G_B = b(1, P) = (\alpha_b(P), bP)$, and sends $\{G_B, sign_B(G_B)\}$ to Alice.
(3) Alice verifies the validity of the signature and computes $K_A = \alpha_b(P)^a\alpha_a(bP)$.

According to Remark 2, $K_A = K_B$. Therefore, Alice and Bob have a shared secret key.

Remark 2. According to part b of Theorem 1, in the preview algorithm, $K_A = K_B$.

5.2 Oblivious transfer using generalized Jacobian of elliptic curves

This subsection aims to provide a protocol to exchange private keys n_A and n_B between two distrustful parties, Alice and Bob, without using a trusted third party and without simultaneous exchange of information. We introduce an algorithm to exchange the secret key n_A with oblivious transfer from Alice to Bob using the generalized Jacobian of elliptic curves, which Bob can use in the same way to transfer his private key n_B to Alice.

Alice and Bob decide upon an elliptic curve $E : y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_q$, $q = p^n$, p is a large prime ($p \approx 2^{160}$). Also, they agree on a divisor $\mathfrak{m} = (M) + (N)$, where M and N are distinct finite points on E, and they select one x-coordinate. Suppose P_1 and P_2 are corresponding points to this x-coordinate, hence; we can write $P_1 = -P_2$. The selection of x-coordinate should be such that $M, N \notin \langle P_1\rangle$. Alice secretly chooses one of the two points P_1 and P_2, which we call P_A. Similarly, Bob chooses $P_B \in \{P_1, P_2\}$. Suppose $n = |\langle(1, P_1)\rangle|$. Consider that Alice wants to send secret key n_A with oblivious transfer to Bob, where $n_A < p$. Alice and Bob perform the following steps:

(1) Bob computes

$$G_B = n_B(1, P_B) = (\alpha_{n_B}(P_B), n_BP_B),$$

and sends $\{G_B, sign_B(G_B)\}$ to Alice.

Alice		Bob
(1) calculates $h = hash(m)$, (2) chooses c, $(\gcd(c, n) = 1)$, calculates $r_m = \alpha_c(P)$, (3) $s_m = c^{-1}(h + r_m k_A) \mod n$, if $(s_m, n) \neq 1$ returns to (2), (4) puts $sign_A(m) = (r_m, s_m)$.	$\xrightarrow{\{m, sign_A(m)\}}$	(1) calculates $h = hash(m)$, $s_m^{-1} \pmod n$, $R = (hs_m^{-1})G + (r_m s_m^{-1})A$, (2) puts r'_m = the first coordinate of R, (3) checks $r'_m = r_m$.

Figure 1. A digital signature using generalized Jacobian of elliptic curves

Alice		Bob
chooses a private integer a,		chooses a private integer b,
computes $G_A = (\alpha_a(P), aP)$	$\xrightarrow{\{G_A, sign_A(G_A)\}}$	computes $G_B = (\alpha_b(P), bP)$
	$\xleftarrow{\{G_B, sign_B(G_B)\}}$	
computes $K_A = \alpha_b(P)^a \alpha_a(bP)$		computes $K_B = \alpha_a(P)^b \alpha_b(aP)$

Figure 2. The key agreement protocol using generalized Jacobian of elliptic curves

(2) Alice verifies the validity of the signature, and calculates $K_A = \alpha_{n_B}(P_B)^{n_A}\alpha_{n_A}(n_B P_B)$, $M_A = n_A 1_{\mathbb{F}_q} + K_A$ and $G_A = n_A(1, P_A) = (\alpha_{n_A}(P_A), n_A P_A)$. She sends

$$\{M_A, G_A, sign_A(M_A, G_A)\}$$

to Bob.

(3) Bob verifies the validity of the signature, and computes $K_B = \alpha_{n_A}(P_A)^{n_B}\alpha_{n_B}(n_A P_A)$, and obtains $M_B = M_A - K_B$. He receives Alice's secret key with probability $1/2$ according to Remark 5.

Remark 3. Alice and Bob must agree on an x-coordinate so that the order of $P_1 = (x, y)$ is not 2, because if that happens, then $P_1 = -P_2$, and Bob always receives Alice's message.

In the final phase of the algorithm, Bob must check whether or not he has received Alice's private key. To accomplish this, we present a technique in Remark 4.

Remark 4. Suppose c is the value obtained by Bob in the last step; Bob calculates $\alpha_c(P_1)$ and $\alpha_c(P_2)$. If one of these two numbers equals to $\alpha_{n_A}(P_A)$, Bob ensures he gets Alice's private key, and $c = n_A$. Otherwise, Bob does not receive Alice's private key.

Correctness: Remark 5 guarantees the correctness of our *OT* protocol.

Remark 5. Two cases may occur in this protocol:

(1) $P_A = P_B$; In this case, according to Theorem 1, $K_A = K_B$, as a result, $M_B = n_A 1_{\mathbb{F}_q}$, Moreover, Bob receives Alice's message.

(2) $P_A = -P_B$; in this case, $K_A \neq K_B$, so $M_B \neq n_A 1_{\mathbb{F}_q}$, Furthermore, Alice's message is not received by Bob.

Therefore, Bob achieves the message with probability $1/2$.

Security of the protocol

Here, we examine the security of the proposed *OT* protocol:

Receiver's privacy:

If Alice knows the answer to "Did Bob receive the message?" then Bob's privacy is compromised. Bob sends the value $n_B(1, P_B) = (\alpha_{n_B}(P_B), n_B P_B)$ to Alice. If Alice achieves P_B, she realizes whether $P_B = P_A$, and according to Remark 5, she finds out whether Bob achieved the message. To find P_B, Alice must solve one of the following discrete logarithm problems, which are computationally difficult and require significant computational resources:

$$x(1, P_1) = (\alpha_{n_B}(P_B), n_B P_B),$$
$$x(1, P_2) = (\alpha_{n_B}(P_B), n_B P_B).$$

Otherwise, it is hard to get P_B. Therefore, privacy of Bob relies on the difficulty of solving the discrete logarithm problem on the generalized Jacobian of elliptic curves.

Sender's security:

If $P_A \neq P_B$, and Bob can realize n_A, then privacy of Alice is compromised. Alice sends $M_A = n_A 1_{\mathbb{F}_q} + K_A$ and

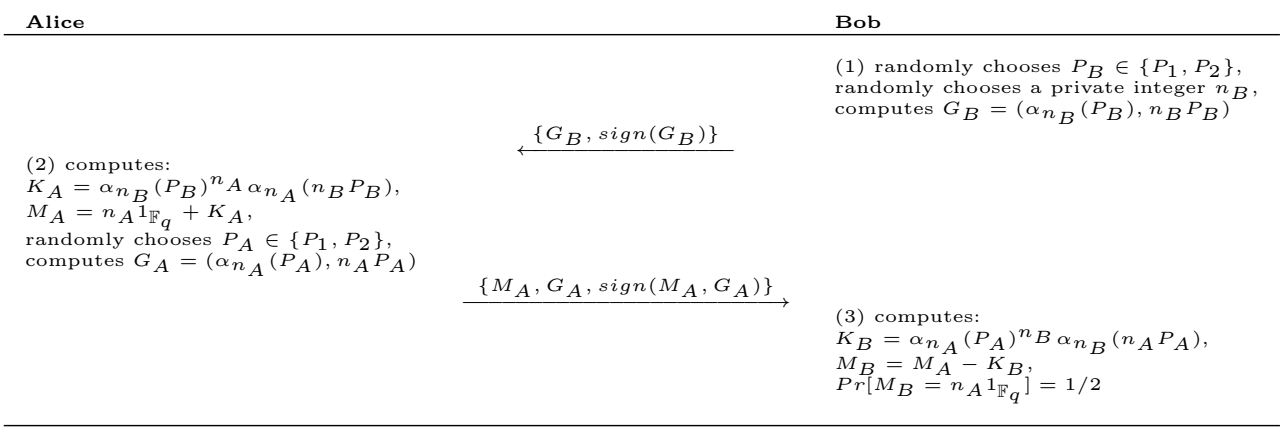

Figure 3. The oblivious transfer protocol using generalized Jacobian of elliptic curves

$n_A(1, P_A) = (\alpha_{n_A}(P_A), n_A P_A)$ to Bob, where $K_A = \alpha_{n_B}(P_B)^{n_A}\alpha_{n_A}(n_B P_B)$. If Bob solves the discrete logarithm $x(1, P_A) = (\alpha_{n_A}(P_A), n_A P_A)$ for $P_A = P_1$ or $P_A = P_2$, then he obtains n_A anyway. Another way to get n_A is to find the value of K_A. Remark 5 states that Bob can easily achieve K_A when $P_A = P_B$, but when $P_A \neq P_B$, Bob faces a more challenging task to achieve K_A. In this case, Theorem 1 provides a solution that involves using the values of $\{\alpha_{n_A}(P_B), n_A P_B\}$. In fact, if he has the values of $\{\alpha_{n_A}(P_B), n_A P_B\}$, he can compute

$$K_A = \alpha_{n_A}(P_B)^{n_B}\alpha_{n_B}(n_A P_B).$$

However, this solution requires that Bob has access to the values of $\{\alpha_{n_A}(P_B), n_A P_B\}$, which he may not have in practice. Therefore, without solving the discrete logarithm problem, it is hard for Bob to obtain n_A when $P_A \neq P_B$. Hence, Alice's privacy is based on the difficulty of the discrete logarithm problem.

5.3 Chosen 1-out-of-2 oblivious transfer using generalized Jacobian of elliptic curves

This subsection aims to present an algorithm using the generalized Jacobian of elliptic curves, in which Alice sends one of the keys n_1 and n_2 to Bob with an OT_2^1, where $n_1, n_2 < p$. In this algorithm, Bob can get only one of two Alice's keys, and Alice remains oblivious to which of the two keys Bob retrieved.

Alice and Bob decide upon an elliptic curve $E : y^2 = x^3 + ax + b$ over the field $\mathbb{F}_q$, $q = p^n$, p is a large prime ($p \approx 2^{160}$). Also, they agree on a divisor $\mathfrak{m} = (M) + (N)$, where M and N are distinct points on E. They choose a random $x \in \mathbb{F}_q$ such that there are two distinct points P_1 and $P_2 = -P_1$ on E with x-coordinate x, and $M, N \notin \langle P_1 \rangle$. Bob chooses $P_B = P_i \in \{P_1, P_2\}$. Suppose Alice wants to send one of the secret keys n_1, n_2 to Bob with oblivious transfer. Alice and Bob perform the following steps:

(1) Bob chooses an integer n_B, computes $G_B = n_B(1, P_B) = (\alpha_{n_B}(P_B), n_B P_B)$, and sends $\{G_B, sign_B(G_B)\}$ to Alice.
(2) Alice verifies the signature and then computes $K_1 = \alpha_{n_B}(P_B)^{n_1}\alpha_{n_1}(n_B P_B)$, $K_2 = \alpha_{n_B}(P_B)^{n_2}\alpha_{n_2}(n_B P_B)$, $M_1 = n_1 1_{\mathbb{F}_q} + K_1$, $M_2 = n_2 1_{\mathbb{F}_q} + K_2$, $G_1 = n_1(1, P_1) = (\alpha_{n_1}(P_1), n_1 P_1)$ and $G_2 = n_2(1, P_2) = (\alpha_{n_2}(P_2), n_2 P_2)$. She sends
$$\{M_1, M_2, G_1, G_2, sign_A(M_1, M_2, G_1, G_2)\}$$
to Bob.
(3) Bob verifies the signature and then computes $K_B = \alpha_{n_i}(P_i)^{n_B}\alpha_{n_B}(n_i P_i)$ and $M_B = M_i - K_B$. According to Remark 7, he obtains one of Alice's private keys.

Remark 6. The value $x \in \mathbb{F}_q$ must be selected so that the order of P_1 is not 2; otherwise, $P_1 = P_2$, and Bob always receives both of Alice's keys.

Correctness: Remark 7 guarantees the correctness of our OT_2^1 protocol.

Remark 7. Two cases may occur in this protocol:

(1) $P_B = P_1$; in this case, according to Theorem 1, $K_B = K_1$, therefore $M_B = n_1 1_{\mathbb{F}_q}$, Moreover, Bob achieves Alice's first key.
(2) $P_B = P_2$; similar to the previous case, $K_B = K_2$, therefore $M_B = n_2 1_{\mathbb{F}_q}$, Furthermore, Bob obtains Alice's second key.

Security of the protocol

The security of the scheme is examined in the following:

Receiver's privacy:

If Alice can obtain i, then privacy of Bob would be compromised. Bob sends $n_B(1, P_B) = (\alpha_{n_B}(P_B), n_B P_B)$ to Alice, where $P_B = P_i$. Therefore, if Alice achieves P_B, she realizes the value of i. Alice needs to solve

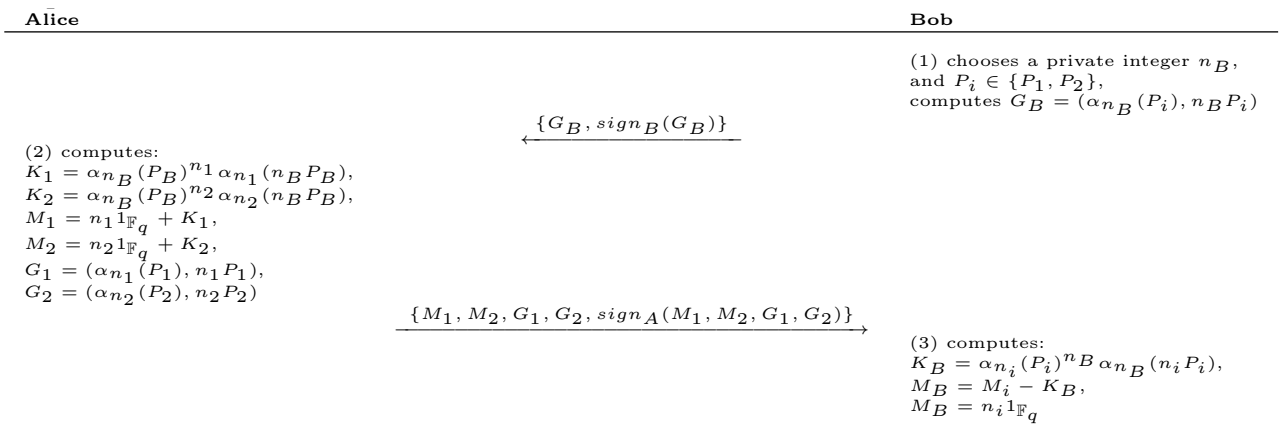

Figure 4. The 1-out-of-2 oblivious transfer protocol using generalized Jacobian of elliptic curves

one of the following discrete logarithm problems to find P_B:

$$x(1, P_1) = (\alpha_{n_B}(P_B), n_B P_B),$$
$$x(1, P_2) = (\alpha_{n_B}(P_B), n_B P_B).$$

Otherwise, it is hard to obtain i. Hence, privacy of Bob is based on the hardness of the discrete logarithm problem on the generalized Jacobian of elliptic curves.

Sender's security:

If Bob gets both of Alice's keys, then privacy of Alice will be compromised. Alice sends

$$M_j = n_j 1_{\mathbb{F}_q} + K_j,$$
$$n_j(1, P_j) = (\alpha_{n_j}(P_j), n_j P_j),$$

for $j = 1, 2$, where $K_j = \alpha_{n_B}(P_B)^{n_j}\alpha_{n_j}(n_B P_B)$ to Bob. If Bob can solve the discrete logarithm $x(1, P_j) = (\alpha_{n_j}(P_j), n_j P_j)$, then he will obtain both n_1 and n_2. Another way to get n_j is to obtain the K_j. If $P_B = P_i$, Bob can easily achieve K_i, as mentioned in Remark 7. For $j \neq i$, we have

$$K_j = \alpha_{n_j}(P_i)^{n_B}\alpha_{n_B}(n_j P_i).$$

Since Bob does not know the values of $\{\alpha_{n_j}(P_i), n_j P_i\}$, he cannot obtain K_j. Therefore, without solving the discrete logarithm problem, it isn't easy to obtain n_j when $j \neq i$. Therefore, Alice's privacy is also based on the hardness of the discrete logarithm problem.

5.4 Chosen t-out-of-k oblivious transfer using generalized Jacobian of elliptic curves

In this subsection, we propose an OT_k^t scheme using a cyclic subgroup of generalized Jacobian of an elliptic curve. In the proposed protocol, Alice sends t of the k keys $n_1, \ldots, n_k \in [1, p-1]$ to Bob with an OT_k^t, so that Bob obtains only t of k Alice's keys, and Alice remains oblivious to the fact that which of k keys are received by Bob.

To this end, Alice and Bob decide upon an elliptic curve $E : y^2 = x^3 + ax + b$ over the field $\mathbb{F}_q$, $q = p^n$, p is a large prime ($p \approx 2^{160}$). Also, they agree on the divisor $\mathfrak{m} = (M) + (N)$, where M and N are distinct finite points on E. They select $P \in E(\mathbb{F}_q)$ such that $M, N \notin \langle P \rangle$. Alice picks $P_1, \ldots, P_k \in \langle P \rangle$ and makes $n_1, \ldots, n_k$ corresponding to $P_1, \ldots, P_k$, publishes $P_1, \ldots, P_k$. Bob wants to receive $n_{r_1}, \ldots, n_{r_t}$. Alice and Bob perform the following steps:

(1) Bob chooses the secret key n_B and computes

$$G_{B_1} = n_B(1, P_{r_1}) = (\alpha_{n_B}(P_{r_1}), n_B P_{r_1}),$$
$$\vdots$$
$$G_{B_t} = n_B(1, P_{r_t}) = (\alpha_{n_B}(P_{r_t}), n_B P_{r_t}),$$

then he sends $\{G_{B_1}, \ldots, G_{B_t}, sign_B(G_{B_1}, \ldots, G_{B_t})\}$ to Alice.

(2) Alice verifies the signature and computes

$$K = \begin{pmatrix} \alpha_{n_B}(P_{r_1})^{n_1}\alpha_{n_1}(n_B P_{r_1}) & \cdots & (\alpha_{n_B}(P_{r_t})^{n_1}\alpha_{n_1}(n_B P_{r_t}) \\ \vdots & \ddots & \vdots \\ \alpha_{n_B}(P_{r_1})^{n_k}\alpha_{n_k}(n_B P_{r_1}) & \cdots & \alpha_{n_B}(P_{r_t})^{n_k}\alpha_{n_k}(n_B P_{r_t}) \end{pmatrix},$$

$$M = \begin{pmatrix} n_1 1_{\mathbb{F}_q} + K_{11} & \cdots & n_1 1_{\mathbb{F}_q} + K_{1t} \\ \vdots & \ddots & \vdots \\ n_k 1_{\mathbb{F}_q} + K_{k1} & \cdots & n_k 1_{\mathbb{F}_q} + K_{kt} \end{pmatrix},$$

and

$$G_1 = n_1(1, P_1) = (\alpha_{n_1}(P_1), n_1 P_1),$$
$$\vdots$$
$$G_k = n_k(1, P_k) = (\alpha_{n_k}(P_k), n_k P_k),$$

then she sends

$$\{M, G_1, \ldots, G_k, sign_A(M, G_1, \ldots, G_k)\}$$

to Bob.

(3) Bob verifies the signature and computes

$$L_1 = \alpha_{n_{r_1}}(P_{r_1})^{n_B}\alpha_{n_B}(n_{r_1}P_{r_1}),$$
$$\vdots$$
$$L_t = \alpha_{n_{r_t}}(P_{r_t})^{n_B}\alpha_{n_B}(n_{r_t}P_{r_t}),$$

and obtains $M_B = [M_{r_1 1} - L_1, \ldots, M_{r_t t} - L_t]$. He achieves $[n_{r_1}, \ldots, n_{r_t}]$ according to the following theorem:

Theorem 2. *In the last step of the algorithm, Bob achieves* $[n_{r_1}, \ldots, n_{r_t}]$.

Proof. According to Theorem 1, for $1 \leq j \leq t$, we have

$$\begin{aligned} M_{r_j j} - L_j &= n_{r_j}1_{\mathbb{F}_q} + K_{r_j j} - L_j \\ &= n_{r_j}1_{\mathbb{F}_q} \\ &+ \alpha_{n_B}(P_{r_j})^{n_{r_j}}\alpha_{n_{r_j}}(n_B P_{r_j}) \\ &- \alpha_{n_{r_j}}(P_{r_j})^{n_B}\alpha_{n_B}(n_{r_j}P_{r_j}) \\ &= n_{r_j}1_{\mathbb{F}_q}. \end{aligned}$$

Since $n_{r_j} < p\,(q = p^n)$, n_{r_j} can be realized by Bob. □

Correctness: Theorem 2 guarantees the correctness of the OT_k^t protocol.

Security of the protocol

Here, We examine the security of the proposed protocol:

Receiver's privacy:

If Alice achieves at least one of $r_1, \ldots, r_t$, then privacy of Bob will be compromised. Bob sends $n_B(1, P_{r_j}) = (\alpha_{n_B}(P_{r_j}), n_B P_{r_j})$ to Alice for $1 \leq j \leq t$. If Alice obtains P_{r_j}, she will find the value of r_j. To this end, Alice can solve the discrete logarithm problems:

$$x(1, P_i) = (\alpha_{n_B}(P_{r_j}), n_B P_{r_j}),$$

for all $1 \leq j \leq t$ and $1 \leq i \leq k$, to achieve n_B. Also, she will find P_{r_j} and r_j simultaneously. Without solving the discrete logarithm problem, it is difficult for Alice to obtain r_j $(1 \leq j \leq t)$. Therefore, privacy of Bob is based on the difficulty of the discrete logarithm problem on the generalized Jacobian of elliptic curves.

Sender's security:

Privacy of Alice is compromised, if Bob obtains at least one n_i such that $i \neq r_j$ for all $1 \leq j \leq t$. Alice sends two sets of information to Bob. The first set is $\{n_i 1_{\mathbb{F}_q} + K_{ij} \;\; 1 \leq j \leq t \; and \; 1 \leq i \leq k\}$ and the second set is $\{n_i(1, P_i) = (\alpha_{n_i}(P_i), n_i P_i)\; 1 \leq i \leq k\}$. For all $1 \leq i \leq k$, if Bob can solve the discrete logarithms

$$x(1, P_i) = (\alpha_{n_i}(P_i), n_i P_i),$$

then he would achieve n_i. According to Theorem 2, he easily realizes n_{r_j} for $1 \leq j \leq t$. In the case that $i \neq r_j$, if Bob wants to get the value n_i from $n_i + K_{ij}$, he needs to obtain K_{ij}, where

$$K_{ij} = \alpha_{n_i}(P_{r_j})^{n_B}\alpha_{n_B}(n_i P_{r_j}).$$

Since Bob does not know the values of $\{\alpha_{n_{r_j}}(P_{r_j}), n_i P_{r_j}\}$, he cannot calculate K_{ij}. Therefore, without solving the discrete logarithm problem, it is hard for Bob to obtain n_i $(i \neq r_j)$.

6 Comparison

This section compares: i) our OT scheme with schemes proposed in [1, 20], ii) our OT_2^1 scheme with schemes proposed in [16, 17, 20, 25], and iii) our proposed OT_k^t scheme with schemes proposed in [4, 6, 31, 32]. This section assumes that the underlying field has prime order p. In the following, we provide definitions for the notations used in this section:

- T_{Mul}: the time of 1024-bit modular multiplication,
- T_{Exp}: the time of 1024-bit modular exponential; $(T_{Exp} \approx 1536T_{Mul} \approx 62976T_{Mu})$,
- T_{Mu}: the time of 160-bit field multiplication; $(T_{Mul} \approx 41T_{Mu}$ [33]),
- T_{Ex}: the time of 160-bit field exponential; $(T_{Ex} \approx 240T_{Mu})$,
- T_{EC-Add}: the time of addition on an elliptic curve; $(T_{EC-add} \approx 5T_{Mu})$,
- T_{EC-Mul}: the time needed for a scalar multiplication on an elliptic curve; $(T_{EC-Mul} \approx 1200T_{Mu})$,
- T_α: the time for computing $\alpha_n(P)$ in $n(1, P)$; $(T_\alpha \approx 2880T_{Mu})$,
- T_{GJ-Add}: the time of addition on generalized Jacobian; $(T_{GJ-Add} \approx 12T_{Mu})$,
- T_{GJ-Mul}: the time needed for a scalar multiplication on a generalized Jacobian of an elliptic curve; $(T_{GJ-Mul} \approx 4320T_{Mu})$,
- T_{bp}: the time of computation of a bilinear pairing,
- T_{hash}: the time for computing of a hash function,
- T_{RO}: the time for computing of a random oracle in [17],
- T_{sign}: the time for computing of the signature algorithm in [1],
- T_{map}: the time for mapping a message to a point on an elliptic curve in [20],

The double-and-add method is a commonly used algorithm for computing scalar multiplication on elliptic curves. In the context of the generalized Jacobian of an elliptic curve, this method requires 160 generalized Jacobian doublings and, on average, 80 generalized Jacobian additions to compute $\alpha_n(P)$ in $n(k, P)$. The addition formula given in [19] and the Pari codes pre-

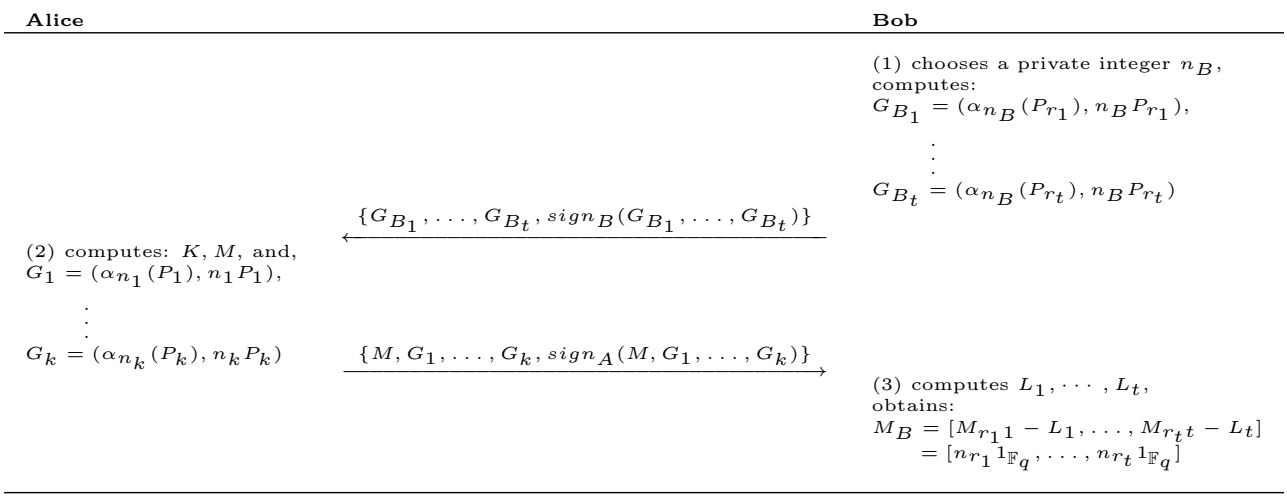

Figure 5. The t-out-of-k oblivious transfer protocol using generalized Jacobian of elliptic curves

sented in Section 7 demonstrate that the generalized Jacobian addition (or doubling) operation requires one elliptic curve operation and seven field multiplications on average. According to [33], an elliptic curve addition or doubling operation requires five field multiplications. Based on this, we can estimate that computing $\alpha_n(P)$ on the generalized Jacobian of an elliptic curve requires approximately 2880 field multiplications. Similarly, computing nP on the elliptic curve needs about 1200 field multiplications. Computation of α^k on the field $\mathbb{F}_p$ by repeated squaring and multiplying needs average of about 240 160-bit field multiplication. Computation α^k mod N by repeated squaring and multiplying needs about 1536 1024-bit modular multiplication. Hence, computing of $n(k, P) = (k^n \alpha_n(P), nP)$ needs about 4320 field multiplications. Note that field addition, modular addition, and bit-XOR have lower computational costs than other operations; therefore, we don't count them.

In Table 1, we compare our proposed scheme in Section 5.2 with the schemes in [1, 20] in terms of the needed rounds, transferred bits between sender and receiver, computational costs of sender and receiver, and security against man-in-the-middle attacks. The Table illustrates that our scheme is better than the schemes [1, 20] in terms of Needed rounds and Transferred bits. In terms of computational costs, we can say that our protocol is better than [1], and if $13162T_{Mu} \leq T_{map} - 2T_{hash}$, it is better than [20], according to Table 1. Our OT protocol and [1] are secure against MITM attack, while the protocol of [20] is not. Additionally, in [20] receiver's security is compromised [6]. Therefore, our scheme is more secure than [20]. Regarding OT schemes, our research has not identified any recently introduced scheme that is similar to ours. However, there are several OT_2^1 schemes that we compare with ours in the following.

Table 2 provides a comparison between our OT_2^1 proposed scheme in Figure 4 and the scheme proposed in [25] based on RSA, the schemes proposed in [16, 17, 20], in terms of the needed rounds, transferred bits between sender and receiver, and computational costs of sender and receiver. Regarding communication costs, we consider the number of rounds and the number of bits transmitted between the parties. Table 2 shows that our protocol is better than others regarding communication costs. In terms of computational costs, the Table shows that if we consider the same computation time for hash functions of all protocols and T_{RO}, our protocol is better than [17]. Moreover, if $T_{hash} \leq 80598T_{Mu}$, our protocol is the best among the compared OT_2^1 schemes.

Table 3 compares our OT_k^t with the first Mu *et al.*'s scheme [4], and the schemes in [6, 31, 32] in terms of needed rounds, the number of bits sent by a sender to a receiver, and the number of bits sent by the receiver to sender. Table 3 shows that our protocol is better than the protocols in [4, 32] In terms of needed rounds, and the protocols of [6, 31] have the same needed rounds as ours. Table 3 demonstrates that our protocol outperforms those proposed in [4, 31, 32] in terms of the number of bits transferred to the receiver. Also, if $160(3t + 2) \leq 160(k + t + 1)$ (or equivalently $2t + 1 \leq k$), our protocol is better than [6]. Based on the number of bits transferred by the sender, our scheme is better than the scheme proposed by Mu *et al.* [4]. Moreover, if $k \leq \dfrac{1024t + 704}{40t - 161}$, our scheme is better than [31, 32]. According to the number of bits the sender transfers, the protocol proposed by Chen *et al.* [6] outperforms our protocol.

Table 4 presents a comparison of the computational costs of our proposed OT_k^t protocol with several existing schemes, including Mu *et al.*'s scheme [4], as

Table 1. Comparisons of the proposed OT with the others

Features	Our scheme	Parakh [20]	Rabin [1]
Needed rounds	2	3	4
Transferred bits	1600 bits	1920 bits	5120 bits
Computational costs	$6T_{GJ-Mul}$ $+2T_{GJ-add}$ $+4T_{\alpha}$ $+2T_{Ex}$ $+10T_{Mu}$ $+4T_{hash}$ $= 38354T_{Mu}$ $+4T_{hash}$	$10T_{EC-Mull}$ $+6T_{EC-add}$ $+2T_{map}$ $= 12030T_{Mu}$ $+2T_{map}$	$2T_{Exp}$ $+2T_{Mul}$ $+4T_{sign}$ $= 126034T_{Mu}$ $+4T_{sign}$
Security against MITM attack	yes	no	yes

Table 2. Comparisons of the proposed OT_2^1 with the others

Features	Our scheme	Parakh [20]	Chou [16]	hauck2017efficient [17]	Esmaeilzade [25]
Needed rounds	2	3	3	3	3
Transferred bits	2400 bits	3040 bits	4096 bits	4096 bits	4096 bits
Computational costs	$7T_{GJ-Mul}$ $+2T_{GJ-add}$ $+5T_{\alpha}$ $+3T_{Ex}$ $+11T_{Mu}$ $+4T_{hash}$ $= 45395T_{Mu}$ $+4T_{hash}$	$16T_{EC-Mul}$ $+10T_{EC-add}$ $= 829560T_{Mu}$	$2T_{Exp}$ $+T_{Mul}$ $+3T_{hash}$ $= 125993T_{Mu}$ $+3T_{hash}$	$2T_{Exp}$ $+T_{Mul}$ $+3T_{hash}$ $+T_{OR}$ $= 125993T_{Mu}$ $+3T_{hash}$ $+T_{RO}$	$3T_{Exp} + T_{Mul}$ $+3T_{hash}$ $= 188969T_{Mu}$ $+3T_{hash}$

Table 3. Comparison of our OT_k^t in terms of communication costs

features	our scheme	Mu *et al.* [4]	Chu *et al.* [31]	Zhang *et al.* [32]	Chen *et al.* [6]
Communication rounds	2	3	2	3	2
Number of bits (sender to receiver)	$160(kt+3k+2)$	$1024(kt+k)$	$1024(k+t+1)$	$1024(k+t+1)$	$160(k+t)$
Number of bits (receiver to sender)	$160(3t+2)$	$1024t$	$1024t$	$1024(t+3)$	$160(k+t+1)$

well as the schemes presented in [6, 31, 32]. The table shows that our protocol outperforms [6, 31] in terms of the computational costs incurred by the receiver. In particular, if the hash function time T_{hash} is less than or equal to $(16575t - 18640)T_{Mu}$, our protocol offers the best performance. In terms of computational costs for the sender, our proposed scheme offers better performance than several existing schemes under certain conditions. Specifically, if the hash function time T_{hash} is less than or equal to $(k(24035t3726) - 5890)T_{Mu}$, our scheme is superior to [4]. If $(k + t - 2)T_{hash} \geq (k(14904t - 55524) - 62976t + 48804)T_{Mu}$, it outperforms [31]. If $2T_{hash} \leq (k(7738719 - 14904t) - 111780)T_{Mu}$, it is better than [32]. Finally, if $kT_{bp} + (k-2)T_{hash} \geq (k(14904t+7452)-1200t+111780)T_{Mu}$, it outperforms [6].

7 Research achievements

This paper introduced a key exchange protocol using the generalized Jacobian of elliptic curves and proposed a two-round algorithm for oblivious transfer using the key exchange protocol. The algorithm was extended to 1-out-of-2 and t-out-of-k oblivious transfer. In these protocols, we use a digital signature algorithm using generalized Jacobian to prevent man-in-the-middle attacks. One of the benefits of these schemes is that it is unnecessary to convert the messages to the points on the elliptic curve. The security of the proposed algorithms relies on the assumption that the discrete logarithm problem on the generalized Jacobian of the elliptic curves is computationally difficult to solve, which is a widely accepted assumption in the field of cryptography.

We compare our protocols with the number of related works in Table 1, Table 2, Table 3, and Table 4. Table 1 shows that our scheme is better than the schemes [1, 20] in terms of Needed rounds and Transferred bits and more secure than [20]. We can see in Table 2 that our OT_2^1 scheme reduced communication costs in comparison with [16, 17, 20, 25]. Moreover, regarding computational costs, Table 2 shows our pro-

Table 4. Comparison of our OT_k^t in terms of computational costs

	our scheme	Mu *et al.* [4]	Chu *et al.* [31]	Zhang *et al.* [32]	Chen *et al.* [6]
Sender	$(k+2)T_{GJ-Mul}$ $+T_{GJ-add}$ $+(kt+1)T_{\alpha}$ $+ktT_{Ex}$ $+(kt+4)T_{Mu}$ $+2T_{hash}$ $=7452(k(2t+1)$ $+15)T_{Mu}+2T_{hash}$	ktT_{Ex} $=62974ktT_{Mu}$	$(k+t+1)T_{Exp}$ $+(k+t)T_{hash}$ $=62976(k+t+1)T_{Mu}$ $+(k+t)T_{hash}$	$3kT_{Exp}$ $+3kT_{Mul}$ $=7746171kT_{Mu}$	tT_{EC-Mul} $+kT_{bp}$ $+kT_{hash}$ $=1200tT_{Mu}$ $+kT_{bp}$ $+kT_{hash}$
Receiver	$(t+2)T_{GJ-Mul}$ $+T_{GJ-add}$ $+(t+1)T_{\alpha}$ $+T_{Ex}$ $+(t+4)T_{Mu}$ $+2T_{hash}$ $=7456(4t+5)T_{Mu}$ $+2T_{hash}$	tT_{Exp} $=62974tT_{Mu}$	$2tT_{Exp}$ $+2tT_{Mul}$ $2tT_{hash}$ $=126034tT_{Mu}$ $+2tT_{hash}$	$(2t+3)T_{Exp}$ $+tT_{Mul}$ $=189051(t+1)T_{Mul}$	$(2(k+t)+1)T_{EC-Mul}$ $+tT_{bp}$ $+tT_{hash}$ $=1200(2(k+t)+1)T_{Mu}$ $+tT_{bp}$ $+tT_{hash}$

tocol is better than [17]. Table 3 shows our OT_k^t protocol is better than [4, 31, 32] in terms of the number of bits transferred by the receiver. Regarding the number of bits transferred by the sender, our scheme is better than the scheme of [4]. In terms of computational costs of the receiver, Table 4 shows our protocol is better than [6, 31]. In terms of computational costs of the sender, if $T_{hash} \leq (k(24035t-3726)-5890)T_{Mu}$, our scheme is better than [4].

Listing 1 The code for computing $c_{\mathfrak{m}}(P,Q)$ in the elliptic curve $E: y^2 = x^3+2x+3$, where $\mathfrak{m} = (M)+(N)$, $M=(4,50)$, $N=(12,3)$, and $P,Q \in E(\mathbb{F}_{97})$ such that $M,N \notin \langle P,Q\rangle$.

```
print("Input g(P,Q) to compute c_m(P,Q)
in GJ of E:y^2=x^3+2x+3, p=97.");
print("m=(M)+(N)=(4,50)+(12,3),
M,N shouldn't be in <P,Q>");
g(P,Q)={
  p=97;
  e=ellinit([2,3],p);
  M=[4,50];
  N=[12,3];
  R=elladd(e,Q,P);
  if(P==[0],
    P=Q;
    Q=[0];);
  if(Q!=[0],
    if(Q[1]!=P[1],
      m=Mod((Q[2]-P[2])/(Q[1]-P[1]),p);
      b=m*P[1]-P[2];
      l_1=M[2]-m*M[1]+b;
      l_2=N[2]-m*N[1]+b;
    );
    if(Q!=P&&Q[1]==P[1],
      l_1=M[1]-P[1];
      l_2=N[1]-P[1];
    );
    if(Q==P&&Mod(2*P[2]+e[1]*P[1]+e[3],p)!=0,
     m=Mod((3*(P[1]^2)+2*e[2]*P[1]+e[4]
      -e[1]*P[2])/(2*P[2]+e[1]*P[1]+e[3],p);
      b=Mod(m*P[1]-P[2],p);
      l_1=M[2]-m*M[1]+b;
     l_2=N[2]-m*N[1]+b;
    );
    if(Q==P && Mod(2*P[2]+e[1]*P[1]+e[3],p)==0,
      l_1=M[1]-P[1];
      l_2=N[1]-P[1];
    );
     if(R!=[0],
      l_3=M[1]-R[1];
      l_4=N[1]-R[1];
    );
     if(R==[0],
      l_3=1;
      l_4=1;
    );
    );
   if(Q==[0]&&P!=[0],
   l_1=M[1]-P[1];
  l_2=N[1]-P[1];
  );
  if(Q==[0]&&P==[0],
  l_1=1;
   l_2=1;
  );
  if(Q==[0],
    if(R!=[0],
      l_3=M[1]-R[1];
      l_4=N[1]-R[1];
    );
    if(R==[0],
      l_3=1;
      l_4=1;
    );
    );
  return(Mod((l_1/l_3)/(l_2/l_4),p));
}
```

Listing 2 The code for computing $n(a,P)$ in generalized Jacobian of the elliptic curve $E: y^2 = x^3+2x+3$ over the field $\mathbb{F}_{97}$, where $\mathfrak{m} = (M)+(N)$, $M=(4,50)$, $N=(12,3)$ and $P \in E(\mathbb{F}_{97})$ such that $M,N \notin \langle P\rangle$.

```
print("Input f(n,a,P) to calculate n(a,P)
```

```
(M,N shouldn't be in <P>)");
2 f(n,a,P)={
   3    e=ellinit([2,3],97);
   4     bb=binary(n);
   5     print("b="bb);
   6     s=matsize(bb);
   7     t=s[2];
   8     c=1;
   9     h=1;
   10     read(g);
   11     if(n==1,return([c,P]));
   12        for(i=1,t-1,
   13          if(bb[i+1]==1,
   14
       c=c^2*g(ellpow(e,P,h),ellpow(e,P,h));
   15           h=2*h;
   16           c=c*g(ellpow(e,P,h),P);
   17           h=h+1;
   ,
   18          c=c^2*g(ellpow(e,P,h),ellpow(e,P,h));
   19           h=2*h;
   );
   );
   20  return([a^n*c,ellpow(e,P,n)]);
}
```

Acknowledgment

This research was partly supported by the University of Kashan under grant number 1073209/013.

References

[1] Michael O Rabin. How to exchange secrets with oblivious transfer. 1981.

[2] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[3] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 234–238. Springer, 1987.

[4] Yi Mu, Junqi Zhang, and Vijay Varadharajan. m out of n oblivious transfer. In *Australasian Conference on Information Security and Privacy*, pages 395–405. Springer, 2002.

[5] Chin-Chen Chang and Jung-San Lee. Robust t-out-of-n oblivious transfer mechanism based on crt. *Journal of network and computer applications*, 32(1):226–235, 2009.

[6] Yalin Chen, Jue-Sam Chou, and Xian-Wu Hou. A novel k-out-of-n oblivious transfer protocols based on bilinear pairings. *Cryptology ePrint Archive*, 2010.

[7] Der-Chyuan Lou and Hui-Feng Huang. An efficient t-out-of-n oblivious transfer for information security and privacy protection. *International Journal of Communication Systems*, 27(12):3759–3767, 2014.

[8] Jianchang Lai, Yi Mu, Fuchun Guo, Rongmao Chen, and Sha Ma. Efficient k-out-of-n oblivious transfer scheme with the ideal communication cost. *Theoretical Computer Science*, 714:15–26, 2018.

[9] Martin Stanek *et al.* Fast contract signing with batch oblivious transfer. In *IFIP International Conference on Communications and Multimedia Security*, pages 1–10. Springer, 2005.

[10] Haruna Higo, Keisuke Tanaka, Akihiro Yamada, and Kenji Yasunaga. A game-theoretic perspective on oblivious transfer. In *Australasian Conference on Information Security and Privacy*, pages 29–42. Springer, 2012.

[11] Han Jiang, Qiuliang Xu, Changyuan Liu, Zhihua Zheng, Yi Tang, and Mingqiang Wang. Cut-and-choose bilateral oblivious transfer protocol based on ddh assumption. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–11, 2018.

[12] Carmit Hazay, Peter Scholl, and Eduardo Soria-Vazquez. Low cost constant round mpc combining bmr and oblivious transfer. *Journal of Cryptology*, 33(4):1732–1786, 2020.

[13] Sunil B Mane and Pradeep K Sinha. Oblivious information retrieval on outsourced database servers. *International Journal of Scientific and Engineering Research*, 5(5), 2014.

[14] Yu-Guang Yang, Si-Jia Sun, Qing-Xiang Pan, and Peng Xu. Reductions between private information retrieval and oblivious transfer at the quantum level. *Optik*, 126(21):3206–3209, 2015.

[15] Hoda Jannati and Behnam Bahrak. An oblivious transfer protocol based on elgamal encryption for preserving location privacy. *Wireless Personal Communications*, 97(2):3113–3123, 2017.

[16] Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In *International Conference on Cryptology and Information Security in Latin America*, pages 40–58. Springer, 2015.

[17] Eduard Hauck and Julian Loss. Efficient and universally composable protocols for oblivious transfer from the cdh assumption. *Cryptology ePrint Archive*, 2017.

[18] Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[19] I Déchene. Arithmetic of generalized jacobians, algorithmic number theory symposium-ants vii. *Lecture Notes in Computer Science*, pages 421–435.

[20] Abhishek Parakh. Oblivious transfer using elliptic curves. In *2006 15th International Conference on Computing*, pages 323–328. IEEE, 2006.

[21] Paulo SLM Barreto, Bernardo David, Rafael Dowsley, Kirill Morozov, and Anderson CA Nascimento. A framework for efficient adaptively secure composable oblivious transfer in the rom. *arXiv preprint arXiv:1710.08256*, 2017.

[22] Zengpeng Li, Chunguang Ma, Minghao Zhao, and Chang Choi. Efficient oblivious transfer construction via multiple bits dual-mode cryptosystem for secure selection in the cloud. *Journal of the Chinese Institute of Engineers*, 42(1):97–106, 2019.

[23] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 668–699. Springer, 2020.

[24] Nibedita Kundu, Sumit Kumar Debnath, and Dheerendra Mishra. 1-out-of-2: post-quantum oblivious transfer protocols based on multivariate public key cryptography. *Sādhanā*, 45(1):1–12, 2020.

[25] Saeid Esmaeilzade, Nasrollah Pakniat, and Ziba Eslami. A generic construction to build simple oblivious transfer protocols from homomorphic encryption schemes. *The Journal of Supercomputing*, 78(1):72–92, 2022.

[26] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer, 2009.

[27] Lawrence C Washington. *Elliptic curves: number theory and cryptography*. Chapman and Hall/CRC, 2008.

[28] Maxwell Rosenlicht. Generalized jacobian varieties. *Annals of Mathematics*, pages 505–530, 1954.

[29] H Daghigh and M Bahramian. Generalized jacobian and discrete logarithm problem on elliptic curves. 2009.

[30] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1(1):36–63, 2001.

[31] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *International Workshop on Public Key Cryptography*, pages 172–183. Springer, 2005.

[32] Jianhong Zhang and Yumin Wang. Two provably secure k-out-of-n oblivious transfer schemes. *Applied mathematics and computation*, 169(2):1211–1220, 2005.

[33] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. *Designs, codes and cryptography*, 19(2):173–193, 2000.

Maryam Rezaei Kashi received her B.Sc. in Mathematics and Applications and her M.Sc. degree in pure mathematics from the University of Kashan, Kashan, Iran, in 2014, and 2017, respectively. She is a Ph.D. candidate in pure mathematics at the University of Kashan. Her research interests include Algebraic Number theory, Elliptic curve cryptography, and Post-quantum cryptography.

Mojtaba Bahramian is an assistant professor in the Department of Pure Mathematics at the University of Kashan, Kashan, Iran. He received his B.Sc. degree in mathematics from the University of Kashan in 1999, the M.Sc. degree in Pure Mathematics from the Sharif University of Technology, Tehran, Iran, in 2001, and the Ph.D. degree in Pure Mathematics from the University of Kashan in 2010. His research interests include Algebraic Number theory, Elliptic curve cryptography, and Post-quantum cryptography.