NOVEL CRYPTOGRAPHY METHOD BASED ON ELLIPTIC CURVE USING LIE ALGEBRAS

AMOR HASIC & EMINA HASIC
Department of Computer Science International University
University of Novi Pazar, Serbia

ABSTRACT

This paper explores a novel approach to elliptic curve cryptography by leveraging Lie algebras, a branch of mathematics that provides powerful tools for studying symmetries in geometric objects we begin by introducing the fundamental concepts of elliptic curves and Lie algebras, highlighting their relevance in contemporary cryptographic systems. Subsequently, we delve into the intricate connections between elliptic curves and Lie algebras, elucidating how Lie algebras offer a unique framework for understanding the underlying structures of elliptic curves. Our paper presents a comprehensive analysis of how Lie algebras can be effectively utilized to enhance various aspects of elliptic curve cryptography, including key generation, encryption, and decryption processes. We explore the advantages of employing Lie algebras, such as increased computational efficiency and resistance against emerging cryptographic attacks. Furthermore, we discuss practical implementations and provide insights into the feasibility of integrating Lie algebras techniques into existing elliptic curve cryptographic systems. We also discuss potential avenues for future research and development in this promising area of cryptographic study. Through theoretical analysis and practical considerations, this paper underscores the potential of Lie algebras as a valuable tool for advancing elliptic curve cryptography, offering new perspectives and avenues for enhancing the security and efficiency o cryptographic systems in the digital age.

Keywords: Elliptic curve, cryptography, Diffie-Hellman, Lie algebras, data protection.

1.0 INTRODUCTION

Algebra is a mathematical discipline that arose from the problem of solving equations [1]. If one starts with the integers Z, it is known that every equation $a + x = b$,, where a and b are integers, has a unique solution. However, the equation $ax = b$ does not necessarily have a solution in Z, or it may have infinitely many solutions (take $a = b = 0$). So let's increase Z to rational numbers Q, consisting of c/d of all fractions, where d ≠ 0 . Then both equations have a unique solution in Q, provided that a ≠ 0 for the equation $a \neq 0$. So Q is a field. In the study of fields obtained by joining the roots of polynomial equations, a new concept appeared, namely, symmetries of fields that permute the roots of the equation. The theory of groups arose from the study of polynomial equations. [2]. The solvability of the equation is determined by the group of permutations of its roots; before Abel [1824] and Galois [1830] mastered this relation, it led Lagrange [1770] and Cauchy [1812] to investigate permutations and prove the precursors of the theorems that bear their names. The term "group" was coined by Galois. Interest in transformation groups, and in what we now call classical groups, grew after 1850; thus Klein's Erlanger program [1872] emphasized their role in geometry. Modern group theory began when the axiomatic method was applied to these results; Burnside's theory of groups of finite order [1897] marks the beginning of a new discipline, abstract algebra, in which structures are defined by axioms and the nature of their elements is irrelevant.

Definition 1.1. A group is a set G, together with a map $G \times G$ into G with the following properties:

- Closure: For all $a, b \in G$ we have

$$a \cdot b \in G$$

- Associativity: For all $a, b, c \ \in \ G$.

$$(ab)c = a(bc)$$

There exists an element e in G such that for all $a \in G$,

$$a \cdot e = a \cdot x = x$$

The element e is unique and is called the group identity element or simply identity.

- and such that for all $a \in G$, there exists $a' \in G$ with

$$a \cdot a' = a' \cdot a = e$$

is called the inverse of a.

Definition 1.2. A group G is said to be commutative or abelian if for all $a, b \ \in \ G$ we have $a \cdot b \ = \ b \cdot a$. A group that is not abelian is said to be non-abelian.

Almost all groups where cryptography is encountered are abelian, since the commutative property is what makes them cryptographically interesting.

Proposal 1.3. (Uniqueness of identity). Let G be a group, and let $e, f \ \in \ G$ such that for all $a \in G$.

$$e \cdot a = a \cdot e = a$$

$$f \cdot a = a \cdot f = a$$

Then $e \ = \ f$.

Proof. Since e is an identity, we have

$$e \cdot f = f$$

On the other hand, since f is an identity, we have

$$e \cdot f = e$$

So $e = e \cdot f = f$..

Definition 1.4. A subgroup of G is a subgroup H of G with the following properties:

1) Identity is an element of H.

2) If $h \in H$, then $h^{-1} \in H$.

3) If $h_1, h_2 \in H$, then $h_1 \cdot h_2 \in H$.

Definition 1.5. Let G and H be groups. A homomorphism from G to H is a map $\varphi: G \rightarrow H$ such that, for all a, b in G, $\varphi(a \cdot b) = \varphi(a)\varphi(b)$.

Definition 1.6. A ring is a set with two operations, usually denoted by + and • for addition and multiplication, that satisfy the following properties:

1. Addition is closed :

$$\forall a, b \in G: a + b \in G$$

2. Addition is associative :

$$\forall a, b, c \in G: a + (b + c) = (a + b) + c$$

3. 0 is an additive identity :

$$\forall a \in G: a + 0 = 0 + a = a$$

4. The additive inverse always exists :

$$\forall a \in G: a + (N - a) = (N - a) + a = 0$$

5. Addition is commutative :

$$\forall a, b \in G: a + b = b + a$$

6. Multiplication is closed :

$$\forall a, b \in G: a \cdot b \in G$$

7. Multiplication is associative :

$$\forall a, b, c \in G: (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

8. 1 is a multiplicative identity

$$\forall a \in G: a \cdot 1 = 1 \cdot a = a$$

9. Multiplication and addition satisfy the distributive law :

$$\forall a, b, c \in G: (a + b) \cdot c = a \cdot c + b \cdot c$$

10. Multiplication is commutative:

$$\forall a, b \in G: a \cdot b = b \cdot a$$

The ring and its two operations can be denoted by a triple $(R,\cdot,+)$.

If the multiplication also happens to be commutative, we say that the ring is commutative.

Definition 1.7. Let m be an integer different from zero, for $a, b \in G$;

$$a \equiv b(mod\ m) \Rightarrow m|a - b$$

is called the mod m of a with b.

2.0 PRELIMINARIES

Definition 2.1. The scientific field that deals with ways to protect data is called cryptography. It is the science of secrecy or hidden writing. Cryptography is based on mathematical models of algorithms. An algorithm represents a set of procedures that is performed on the initial information that is in an open form. After the mentioned set of operations which are defined by the algorithm called the encryption process, the result is an unintelligible beginning. The exchange of data in the modern world multiplies every day, confidential data can be intercepted by attackers and thus reach the person and organization they are not intended for.

The basic elements of a cryptographic system are a cryptographic algorithm and a cryptographic key, and the basic operations are encryption and decryption. The length of the cryptographic key is defined by the algorithm.

The implementation of the cryptographic technique used are ciphers and digital signatures. The basic element of protection uses a complex encryption system. Confidentiality of data is realized by using symmetric encryption systems, while the application of digital signature technology is based on asymmetric encryption systems.

There are two basic types of encryption systems, symmetric and asymmetric encryption systems.

2.1 Symmetric cipher systems

Symmetric cipher systems is a system where the same key is used for encryption and decryption. That key represents a secret value known to the sender and receiver of the data. In addition, that key represents a shared secret where special attention should be paid. In order to pay special attention, special algorithms for symmetric key exchange were developed.

Symmetric cipher systems are divided into two basic groups:

i. Sequential cipher systems
ii. Block cipher systems

Sequential cipher systems use addition according to model 2, i.e. addition in the binary number system. Addition according to model 2 is denoted by the mathematical symbol $\oplus$ or in various programming languages by the XOR command. The binary system consists of two elements, namely 0 (zero) and 1 (one). Sequential encryption systems are based on the fact that the binary string to be encrypted is added according to model 2 with a cryptographic key, which is also a binary string, and in this way a string representing the cipher is obtained. the cryptographic key in the form of a binary string is generated using an algorithm. Such an algorithm that generates numbers is called a deterministic algorithm.

The period of the cipher string indicates the number of binary elements after which the elements are repeated and must be of the same length as the length of the string they encode.

Block cipher systems are used by dividing the original message into blocks that are encrypted using cryptographic keys in blocks of two or more elements. Each element is encrypted differently, depending on the encryption method of neighboring elements located in the same block.

The basic elements of the block cipher system are:

i. Initial or initial transformation: may contain one or two functions. The first function masks input data ie. swaps blocks that contain only zeros and ones. The second function aims to make certain attacks on block cipher systems as difficult as possible.

ii. A cryptographically weak function that repeats n times: it forms a non-linear function whose parameters are parts of the cryptographic key and parts of the input data. Nonlinear functions can contain only one operation that is very complex or a series of consecutive, mutually different, simple transformations. Each repetition is connected to each other by adding according to model 2, each individual bit with the data coming from the previous repetition.

iii. Final or final transformation: which ensures that encryption and decryption operations are symmetric.

iv. Key Development Algorithm: which has the role of converting a key, which is usually of limited length, into multiple sub-keys consisting of multiple bits.

A block cipher is used to encrypt short messages such as passwords, digital signatures, identification data, and the like.

One of the fundamental problems in secret key cryptography is the exchange of cryptographic keys. Two users communicating with each other and exchanging encrypted data must choose a secret key before starting communication. They must use a secure channel to exchange the secret key. One of them that is publicly available is called the public key, while the other key is called the secret key and is accessible only to its owner. With this encrypted system, there are two families of function pairs. The first function is used for encryption and is marked with E, which is the symbol of the English word for encryption - encryption. The second function serves for decryption and is marked with D, which is the symbol of the English word for decryption - decryption. The first key or k_1 is used for encryption while the second key k_2. s used for decryption. The functions for encryption and decryption are also denoted as E_(k_1)- encryption with key k_1 i D_(k_2) decryption with key k_2. With M we denote a set of open messages, which is a symbol of the English word - message. X denotes a set of ciphers, since in mathematics an unknown quantity is most often denoted by x.

We write as follows:

$$E_{k_1}: M \rightarrow X \qquad (1)$$

$$D_{k_2}: X \rightarrow M \qquad (2)$$

For each open message m that is in the set M, the following applies:

$$D_{k_2}\left(E_{k_1}(m)\right) = m \qquad (3)$$

Before the use of the description system begins, communication participants A and B must agree on which pair of keys k_1, k_2they will use. Pairs of mutual prime numbers that can be used as cryptographic keys must also be determined. Suppose that communication participant A sends a message m from set M to communication participant B. Communication participant A encrypts and sends message m using the function E_{k_1} as follows:

$$E_{k_1}(m) = c \qquad (4)$$

After the calculation, the participant A sends the calculated value c using B. The communication participant B, to whom the message is intended, wants to reconstruct the message sent by A, B must decrypt the cipher c received from A. For this purpose, he will use the decryption function D_{k_2}

To use a cryptographic system with public keys in practice, it is necessary to define a function called a one-way function (One-Way Function OWF).

$$f: M \rightarrow X \qquad (5)$$

The defined one-way function is calculable as follows

$$f(m) = c \qquad (6)$$

Necessarily, a function that is the inverse of a one-way function is difficult to calculate. The inverse function is represented as follows:

$$f^{-1}(c) = m \qquad (7)$$

If communication participant A wants to send a message m to another communication participant B, it is necessary that the public key found in the catalog of public keys of participant B, which is E_b. Poslije toga, After that, A sends the message $f_b(m) = c$ to user B. Only participant B can calculate the inverse function f_b^{-1} and thus only participant B can reconstruct the original message m.

$$f_b^{-1}(c) = f_b^{-1}\left(f_b(m)\right) = m \qquad (8)$$

The security of public key cryptosystems in use today is measured by the number of operations that must be performed to compute the inverse function, even though there is no algorithm that easily computes the inverse function. [4]

3.0 ELLIPTIC CURVES

Elliptic curves are a family of smooth algebraic curves, so they can be described by an algebraic expression and their first derivative is defined at every point of the domain where the curve is defined. It is used in various areas of mathematics from number theory to complex analysis. They have a special application in cryptography.

Definition 3.1. Elliptic curves represent a set of points in a plane whose position is defined by the following algebraic expression:

$$y^2 = x^3 + ax + b \qquad (9)$$

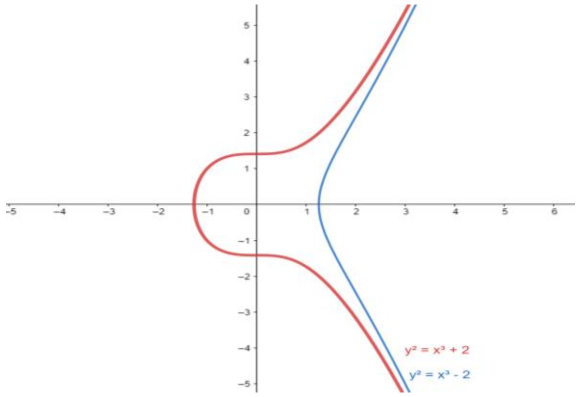
The above expression (9) is also called Weierschrass normal form of elliptic curves.

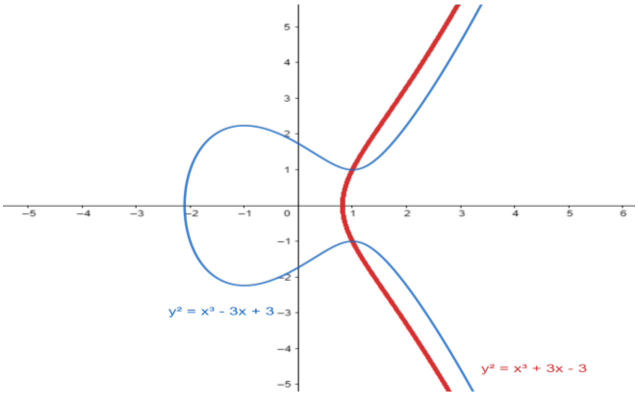
Whereby the expression

$$D = 4a^3 + 27b^2 \qquad (10)$$

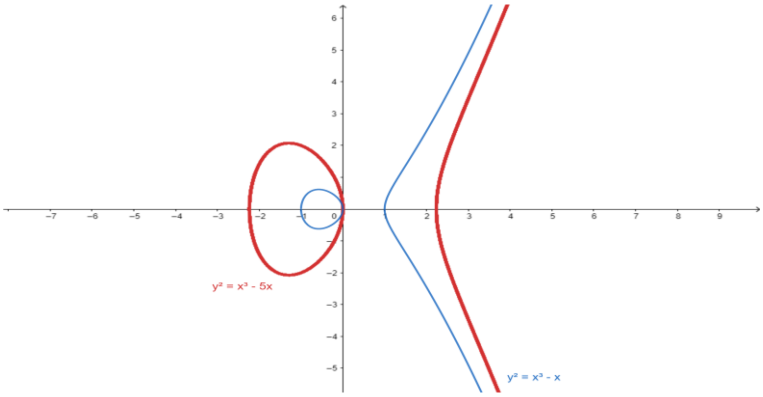
It is called the discriminant of the polynomial and $D \neq 0$..

One of the most important properties of elliptic curves is that an operation can be introduced on them, in a natural way, with which they become Abelian groups.Depending on the parameters a and b, the graph of the elliptic curve can have different shapes. In the following figure 1, several characteristic forms o elliptic curve graphics are shown using the Geogebre program graphic calculator.





a) $y^2 = x^3 - 3x + 3$ $\qquad$ $y^2 = x^3 + 3x - 3$



a) $y^2 = x^3 - 5x$ $y^2 = x^3 + x$

Figure 1: Some characteristic shapes of elliptic curve graphs (a,b -1 component and c-2 component)

Let $P, Q \in E,$ such that the points P and Q belong to the graph of the elliptic curve. Let's set the line p so that it contains the points P and Q. The line p is a segment of the graph of the elliptic curve. If the points P and Q are with coordinates $P(x_1, y_1)$ and $Q(x_2, y_2)$ then the coefficient of the line p is calculated as follows:

$$tg\alpha = \frac{y_2 - y_1}{x_2 - x_1}$$

In the general case, the line p intersects the graph of the elliptic curve at one point, i.e. in the third point belonging to the graphic of the elliptic curve. We will mark the third point with R. We will take the line q containing the point R, which is parallel to the y axis of the rectangular coordinate system and is normal to the x axis of the rectangular coordinate system. Let's mark the intersection of the line q and the graph of the elliptic curve with $-R$. In this way, the addition of points P and Q, which are located on the graphic of the elliptic curve, is defined. The sum of two elements P and Q from the set E is denoted by $P \oplus Q$ or $P + Q$. Graphically, all of the above is shown in the following figure 2.

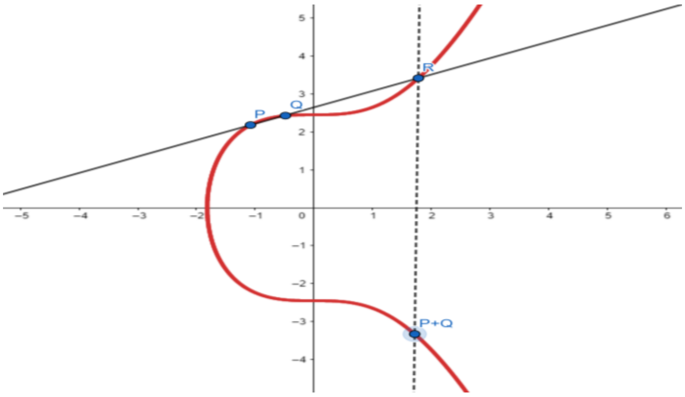


Figure 2: Graphical interpretation of the addition of two elements in set E

Let $P \in E$. To determine the opposite element, it is necessary to add $P \bigoplus -P$, i.e. introduce a third point of intersection with the graph of the elliptic curve O which is neutral and which is considered a point of and line parallel to the *y*-axis. The graphic interpretation is shown in Figure 3.

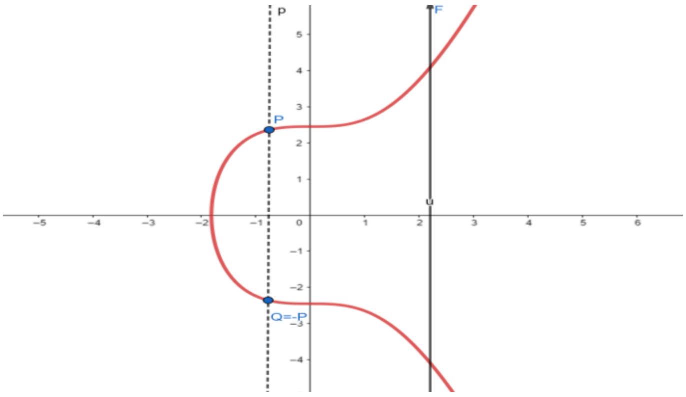


Figure 3: Graphical interpretation of the determination of the opposite element in the set E.

After the graphical interpretation, it is necessary to interpret the operations on the set E algebraically.

Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. The following properties can then be defined:

1. $P + O = O + P = P$
2. $P + (-P) = O$
3. $P + (Q + R) = (P + Q) + R$
4. $P + Q = Q + P$
5. $-O = O$
6. $-P = (x_1, -y_1)$
7. If $Q = -P$, then $P + Q = O$
8. If $Q \neq -P$, then $P + Q = (x_3, y_3)$ where

$$x_3 = \lambda^2 - x_1 - x_2, \qquad y_3 = -y_1 + \lambda(x_1 - x_3)$$

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, \text{if } x_2 \neq x_1 \\ \dfrac{3x_1^2 + a}{2y_1}, \text{if } x_2 = x_1 \end{cases}$$

The number λ is the coefficient of the direction of the line through P and Q, i.e. the tangent at the point P in the case of $P = Q$.

3.1 Public key cryptography and elliptic curves

The basic task of cryptography is to enable communication between two people (the sender and the receiver) in such a way that the third person cannot understand the messages. The message that the sender wants to send to the receiver is called plaintext. The sender forms a public key using a pre-agreed key K, such a process is called encryption, and the resulting format is a cipher. After that, the sender sends the cipher through some communication channel.

Diffie-Hellman (1976) offered one possible solution to the key exchange problem, based on the fact that in some groups exponentiation is much simpler than logarithmization. They are considered the originators of public key cryptography. The idea of public key is to construct cryptosystems where, from knowing the encryption function E_k, it would be practically impossible (in a reasonable amount of time) to calculate the decryption function D_k. Then the function E_k could be public. Thus, in a public-key cryptosystem, each user K has two keys: public E_k and secret D_k. If the sender wants to send the recipient a message k, she encrypts it using the recipient's public key E_{k_1} i.e. sends the receiver code $y = E_{k_1}(x)$. The receiver decrypts the cipher using his secret key D_{k_1}, $D_{k_1}(y) = D_{k_1}(E_{k_1}(x)) = x$. Let's note that the receiver must have some additional information (so-called trapdoor - hidden entrance) about the function E_{k_1}, so that only he can calculate its inverse D_{k_1}, while this is impossible for everyone else. Such functions whose inverse is difficult to calculate without knowing some additional data are called personal one-way functions.

Discrete logarithm problem. Let $(G,*)$ be a finite group, $g \in G$, $H = \{g^i : i \geq 0\}$a subgroup of G generated by g, and $h \in H$. We need to find the smallest non-negative integer *x* such that $h = g^x$, where

$$g^x = g * g *\ldots* g \qquad (11)$$

x times

he number *x* is called the discrete logarithm and is denoted by $\log_g h$ [5]

The fact that there are groups where the discrete logarithm problem is hard was exploited by Diffie and Hellman (1976) in their solution to the key exchange problem.

If the sender and receiver want to agree on a secret random element in the group G, which they could use as an encryption key in some symmetric cryptosystem. They have to carry out that agreement through some insecure communication channel, of course they didn't exchange any information beforehand, except for the information of group G and its generator g. Using the Deffie-Hellman protocol for key exchange.

1. The sender generates a random natural number$a \in \{1,2,\ldots,|\mathrm{G}| - 1\}$, and sends the element g^a to the recipient. where $|G|$ denotes the number of elements in group G
2. The recipient generates a random natural number $b \in \{1,2,\ldots,|\mathrm{G}| - 1\}$, and sends the element g^b to the sender.
3. Calculation sender $(g^b)^a = g^{ab}$.
4. Calculation recipient $(g^a)^b = g^{ab}$.

Now their secret key $K = g^{ab}$.

We will now describe ElGamal's cryptocomposition (1985), which is based on the difficulty of computing the discrete logarithm in the grouP Z_p^*.

El Gamal's cryptocomposition

Let p be a prime number and $\alpha \in Z_p^*$ primitivni korijen modulo p. primitive root modulo p. The values of p and α are public. Each user of system K chooses his secret key $a_K \in Z_{p-1}^*$ and announces the value $\beta_K = \alpha^{a_k}(mod\ p)$.

1. The sender sends the recipient a secret message $x \in Z_p^*$ by choosing a random number $k \in Z_{p-1}$ and forwards the public message
$$E_B(x,k) = (y_1, y_2) = (\alpha^k mod\ p, x\beta_B^k mod\ p)$$
2. Now the recipient of the bill:
$$D_B(y_1,y_2) = y_2(y_1^{a_B})^{-1}\ mod\ p\ = x(\alpha^{a_B})^k((\alpha^k)^{a_B})^{-1}\ mod\ p = x\ mod\ p = x$$
Where a_B is its secret key.

Example 3.1. Let ***A*** (sender) and ***B*** (receiver) agree (publicly) to use the group Z_{31}^* and fix the element $\alpha = 3$ of this group. ***A*** chooses his secret key $a_K = 7$, while ***B*** chooses his secret key $a_B = 22$. Now ***A*** sends ***B*** the element $\beta_A = 17(= 37\ mod\ 31)$, and ***B*** responds with $\beta_B = 14(= 322\ mod\ 31)$. A wants to pass secret information to B' at x = 24, chooses some random number k, let's say k = 5 and calculates: $y_1 = 35\ mod\ 31 = 26$ i $y_2 = 24 \cdot 145\ mod\ 31 = 27$. ***B*** now receives the numbers 26 and 27 and calculates: $27 \cdot (2622) - 1 mod 31 = 27 \cdot 5 - 1\ mod\ 31 = 27 \cdot 25 mod\ 31 = 24$. And so ***B*** must have received the information x = 24. Let′s notice that ***B*** in no chance does not need to know the random number $k = 5$ nor ***A***′s secret key $a_A = 7$. [6].

3.2. Match-based cryptography

Bilinear pairings

Definition 3.2.1. A bilinear mapping on (G_1, G_μ) is a map

$$\tau: G_1 \times G_1 \rightarrow G_\mu$$

which satisfies the following conditions:

1) (bilinearity) $\forall R, S, T \in G_1, \tau(R+S,T) = \tau(R,T)\tau(S,T)\ i\ \tau(R,S+T) = \tau(R,S)\tau(R,T)$
2) (non-degeneracy) $\tau(P,P) \neq 1$
3) (computability) τ can be efficiently calculated.

For all $S, T \in G_1$ we have the following bilinear matching properties:

1. $\tau(S,\infty) = 1\ i\ \tau(\infty,S) = 1$
2. $\tau(aS,bT) = \tau(S,T)^{ab}$for all $a, b \in Z$
3. $\tau(S,T) = \tau(T,S)$
4. If $\tau(S,R) = 1$ for all $R \in G_1$, where $S = \infty$
5. $\tau(S,-T) = \tau(-S,T) = \tau(S,T)^{-1}$

Definition 3.2.2. Let τ be a bilinear matching on (G_1, G_μ). The bilinear Diffie-Hellman problem (BDHP) is as follows: Given P, aP, bP, cP, calculate $\tau(P,P)^{abc}$.

Hardness of the BDHP implies the hardness of the DHP in both G_1 and G_μ . First, if the DHP in G_1 can be efficiently solved, then one could solve an instance of the BDHP by computing abP and then $\tau(abP, cP) = \tau(P,P)^{abc}$. Also, if the DHP in G_μ can be efficiently solved, then the BDHP instance could be solved by computing $g = \tau(P,P)$, $g^{ab} = \tau(aP, bP), g^c = \tau(P, cP)$ and then gabc. Nothing else is known about the intractability of the BDHP, and the problem is generally assumed to be just as hard as the DHP in G_1 and G_μ . We note that the decisional Diffie-Hellman problem (DDHP) in G_1 can be efficiently solved. The DDHP is to decide whether a given quadruple (P, aP, bP, cP) of elements in G_1 is a valid Diffie-Hellman quadruple, i.e., whether $cP = abP$. This can be accomplished by computing $\gamma_1 = \tau(\mathrm{P}, cP) = \tau(P,P)^c$ and $\gamma_2 = \tau(aP, bP) = \tau(P,P)^{ab}$; then $cP = abP$ if and only if $\gamma_1 = \gamma_2$. [7]

There are three basic pairing-based protocols:

1. Tripartite single-circuit key agreement. Joux's key agreement protocol [8], as modified by Verheul [9], uses bilinear matching on (G_1, G_μ).for which BDHP is elusive. Sender ***A*** non-randomly selects a secret integer $a \in [1, n-1]$ and transmits the point *aP* to the other two parties. At the same time ***B*** and ***C*** emit points *bP* and *cP*. After receiving *bP* and *cP*, ***A*** (and ***B*** and ***C***) can compute the shared secret

$$K = \tau(bP, cP)^a = \tau(P,P)^{abc} \qquad (12)$$

Joux's protocol can be generalized to a single-circuit protocol by using an efficiently computed key multilinear map $\tau_n: G_1^{i-1} \to G_\mu$ for which the following analog BDHP is irrational: given $P, a_1P, a_2P, \ldots, a_IP$, calculates $\tau_n(P, P, \ldots, P)^{a_1 a_2 \cdots a_i}$.

Joux's protocol serves as an elegant example of the potential of pairing in protocol design.

2. Short signatures. Most discrete logarithmic signature schemes such as DSA are variants of the ElGamal signature scheme. The signatures consist of a pair of integers according to the model n, where n is the row of the basic group $G_1 = \langle P \rangle$. The first signature scheme was proposed by Boneh, Lynn and Shacham (BLS) [10] which signature consists of a single group element.
3. Identity-based encryption. When using public key encryption to securely send a message to ***A***, ***B*** encrypts the message using ***A***'s public key. And then it uses its corresponding private key to decrypt. ***B*** should be sure that he has an authentic copy of A's public key because otherwise an attacker could trick ***B*** into using the attacker's public key, and then he could decrypt ***B***' these messages that were intended only for ***A.***

Definition 3.2.3. Let E be an elliptic curve defined over F_q, n is a prime divisor of $\#E(F_q)$ such that $gcd(n, q) = 1$, and *k* is the smallest positive integer such that $n \mid q^k - 1$. Parameters *q*,*n* and *k* should satisfy the following conditions:

i. *n* should be large enough to make Pollard's rho method for computing discrete logarithms in the order-n subgroup $E(F_q)$ infeasible.

ii. k and should be large enough for index calculus methods to solve the DLP in F_{q^k} is infeasible.

iii. k should be small enough that the arithmetic in F_{q^k} can be an efficient derivative.

Some other conditions can be imposed on the elliptic curve parameters to speed up the computation of the Tate matching.

The original goal of matching in cryptography was to solve the discrete logarithm problem. The matching moves the discrete problem from a subgroup over an elliptic curve to a discrete logarithm problem over a finite field. The interest is that the discrete logarithm problem is easier on finite fields compared to elliptic curves. Also in cryptosystems based on matching we have the MOV attack on the elliptic curve discrete logarithm problem [11].

In the interest of brevity of this paper, I will not go into further details of the matching of cryptosystems based on elliptic curves.

4.0 LIE ALGEBRA

Definition 4.1. A vector space V over a field F is a Lie algebra if there exists a bilinear multiplication $V \times V \rightarrow V$, with the operation denoted $(x, y) \mapsto [xy]$, such that:

1) It is skew symmetric where $[x, x] = 0$ for all x in V (this is equivalent to $[x, y] = -[y, x]$ since the character is $F \neq 2$).

2) It satisfies the Jacobi identity $\left[x[yz] + y[z, x] + z[x, y]\right] = 0$ $(x, y, z \in V)$.[12]

Example 4.2. We can show that the real vector space $R^{3}$is a Lie algebra. Recall the following properties of the cross product when a ,b and c are arbitrary vectors and α,β and γ are arbitrary scalars:

1. $a \times b = -(b \times a)$
2. $\alpha x(\beta b + \gamma c) = \beta(a \times b) + \gamma(a \times c)$ i

$$(\alpha a + \beta b)xc = \alpha(a \times c) + \beta(b \times c)$$

Proof. Note, $axa = -(axa)$, by property (1), letting $b = a$, so $a \times a = 0$. According to the above properties, the cross product is both skew symmetric (property 1) and bilinear (property 2) .Vector triple product expansion: $x \times (y \times z) = y(x \times z) - z(x \times y)$. To show that the cross product satisfies the Jacobi identity, we have:

$$\left[x[yz] + y[z, x] + z[x, y]\right] = x \times (y \times z) + y \times (z \times x) + z \times (x \times y)$$

$$= [y(x * z) - z(x * y)] + [z(x * y) - x(y * z)] + [x(z * y) - y(z * x)] = 0$$

Example 4.3. L is itself a left L-module.

The left action of L on L is defined as $x \cdot y = [xy]$ Then we have

$$[[xy]z] = [x[yz]] - [y[xz]]$$

which is a consequence of the Jacobi identity. This shows that L is a left L-module. This is called a coupled module. We define $adx : L \to L$ by

$$ad\ x \cdot y = [xy] \text{ za } x, y \in L$$

Then we have

$$ad[xy] = ad\ x\ ad\ y - ad\ y\ ad\ x$$

Now let V be a left L-module, U is a subspace of V and H a subspace of L. We define HU as the subspace of V covered by all elements of the form xu for $x \in H, u \in U$.

Proposition 4.4: Let g be any Lie algebra. For any $x \in g$, define a linear transformation

$$ad_g(x): g \to g: y \to [x, y]$$

Then $ad_g : g \to gl(g)$ is the representation of g on g itself.

Proof: Clearly ad_g is a linear map. We need to show that

$$ad_g([x, y]) = [ad_g(x), ad_g(y)]$$

For all $x, y \in g$, ie that

$$[[x, y], z] = [x, [y, z]] - [y, [x, z]]$$

For all $x, y, z \in g$.

This, however, is only a form of Jacobian identity.

Definition 4.5- [13] The map $L \to DerL$ that sends x to ad_x is called an adjoint representation of L. This is similar to taking an ad homomorphism from $g \to gl(g)$. The representation ad is a homomorphism:

$$ad_{([x;y])} = [ad_{(x)}, ad_{(y)}] = ad_{(x)}ad_{(y)} - ad_{(y)}ad_{(x)}$$

$$[[x, y];\ z] = [x, [y, z]] - [y, [x, z]]$$

$$0 = [x, [y, z]] + [y, [z, x]] + [z, [x, y]]$$

That is, if and only if the Jacobi identity is satisfied, where $x, y, z = -[z, [x, y]]$ i $-[y;\ [x, z]] = [y, [z, x]]$ by oblique symmetry.

Example 4.6:[13] The set of all internal derivatives $ad_x, x \in L$, is an ideal for $Der(L)$. Let $\delta \in Der(L)$. By definition of inner derivatives, for all $y \in L$:

$$[\delta, ad_x](y) = (\delta(ad_x) - (ad_x)\delta)(y) = \delta[x,y] - ad_x(\delta(y))$$

$$= [\delta(x), y] + [x, \delta(y)] - [x, \delta(y)]$$

$$= ad(\delta(x)y)$$

So, ad_x is an ideal of $Der\ (L)$.

Definition 4.7. Let L be a finite-dimensional Lie algebra over F. We define a map

$$L \times L \to F$$

$$x, y \to \langle x, y \rangle$$

gives

$$\langle x, y \rangle = tr(ad(x)\ ad(y))$$

for $x, y \in\ L$. We call the Killing form on L .

Proposal 4.8. Let L be a finite-dimensional Lie algebra over F. The Killing form on L is a symmetric bilinear form. Moreover, we have

$$\langle [xy], z \rangle = \langle x, [yz] \rangle$$

Proof:

$$\langle [xy], z \rangle = tr(ad[xy]ad(z)) = tr\left(\left(ad(x)ad(y) - ad(y)ad(x)\right)ad(z)\right)$$

$$tr(ad(x)ad(y)ad(z)) - tr(ad(y)ad(x)ad(z))$$

$$tr(ad(x)ad(y)ad(z)) - tr(ad(x)ad(z)ad(y))$$

$$tr\big(ad(x)(ad(y)ad(z) - ad(z)ad(y))\big)$$

$$tr(ad(x)ad[yz]) = \langle x, [yz] \rangle$$

Lemma 4.9: [14] Let F have characteristic zero and let it be algebraically closed. Let n be a positive integer. For $x, y \in gl\ (n, F)$ define

$$t(x, y)\ =\ tr(xy)$$

The function $t:\ gl(n,F) \times\ gl(n,F) \to F$ is an associative, symmetric bilinear form. If L is a Lie subalgebra of $gl(n,F)$, L is simple, and the limit t on $L \times\ L$ is nonzero, then L is not degenerate.

Proof: Clearly, t is F-linear in each variable. Also, t is symmetric because $tr\ (xy)\ =\ tr\ (\ yx)$ for $x, y \in gl(n,F)$. To see that t is associative, let $x, y, z \in gl\ (n, F)$. Then

$$t(x,[y,z]) = tr(x(yz-zy))$$

$$= tr(xyz) - tr(xzy)$$

$$= tr(xyz) - tr(yxz)$$

$$= tr((xy - yx)z)$$

$$= t([x,y],z)$$

Suppose that L is a subalgebra of gl(n,F), L is simple, and the limit t on $L \times L$ is nonzero.. Let $J = y \in L : t(x,y) = 0, x \in L$. We have to prove that J = 0. We claim that J is an ideal of L. Let $y \in L$ and $z \in J$, we have to see that $[y,z] \in J$. Let $x \in L$. Now $t(x,[y,z]) = t(x,y,z) =$ 0 because $z \in J$.

Definition 4.10. Let G be a Lie algebra over F. Consider the following set of ideals:

$$G \supset G' = [G,G] \supset (G')' = [G',G'] \supset ((G')')' = [(G')',(G')'] \dots ..$$

Each member of the sequence is an ideal of G; consecutive quotients are abelian. To improve the notation, we have

$$G^0 = G,$$

$$G^1 = [GG] = G',$$

$$G^2 = [GG^1] = (G')',$$

$$\dots \dots \dots \dots \dots.$$

$$G^i = \left[GG^{i-1}\right]$$

$$G^{i+1} = \left(G^i\right)'$$

$$\dots \dots \dots \dots \dots \dots$$

Then we have

$$G = G^0 \supseteq G^1 \supseteq G^2 \supseteq \cdots G^n = 0.$$

This is called a derived sequence G. Given a Lie algebra G and sequence length $n > 0$, we say that G is solvable if $G^{(n)} = 0$, that is, a Lie algebra is solvable if its derived sequence ends by the zero subalgebra.

Let G be a Lie algebra and we have an ideal I

$$[G,I^1] = \left[G,[I,I]\right] \subseteq \left[I,[I,G]\right] + \left[I,[G,I]\right] \subseteq [I,I] + [I,I] = I^1$$

Therefore, if I is an ideal of G, then so is I^1. Unless otherwise stated.

Lemma 4.11. A Lie algebra G is solvable if and only if its derivative sequence ends at zero.

Proof. If the derived sequence ends in zero, then it is a solvable sequence. Conversely, if

$$G = G^0 \supseteq G^1 \supseteq G^2 \supseteq \cdots G^n = 0.$$

is a solvable sequence, then $G^1 \supseteq G'$ because G/G^1 is commutative, $G^2 \supseteq (G^1)' \supseteq G''$ because $G^1 = G^2$ is also commutative , and so on until $0 = G^n$.

Example 4.12. G is solvable if and only if there exists a chain of subalgebras $G = G^0 \supset G^1 \supset \cdots \supset G^n = 0$ such that:

- G^{n+1} is an ideal of G^n.
- Every quotient of G^n/ G^{n+1} is abelian.

Proof:

- If G is solvable, there exists a set of ideals for G such that $G \supseteq G^1 \supseteq \cdots \supseteq G^n = 0$ for some n. By definition, every ideal forms a subalgebra of G. Therefore , $G^{n+1} \subseteq G^n$, where G^{n+1} and G^n are ideals of G.
- Let G^{n+1} be an ideal of $G^n \subseteq G$, G^{n+1} is an ideal of G. if G is solvable, then by the definition of a subnormal sequence, every quotient of G^n/G^{n+1} is abelian. Derived sequences by definition $G^{n+1} = [G^n, G^n]$. Let $[x, y] \in G^{n+1}$ with $x, y \in G^n$:

$$[x + G^{n+1}, y + G^{n+1}] = [x, y] + [x + G^{n+1}] + [G^{n+1}, y] + [G^{n+1}, G^{n+1}]$$

$$[x + G^{n+1}, y + G^{n+1}] = [x, y] + G^{n+1}$$

Therefore G^n/G^{n+1} is abelian.

Example 4.12. Lie groups $G = T_n(F)$ of nonsingular upper triangular matrices over fields F. If A and B are upper triangular matrices

$$A = \begin{pmatrix} a_1 & & & * \\ 0 & a_2 & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & a_n \end{pmatrix}, \ B = \begin{pmatrix} b_1 & & & * \\ 0 & b_2 & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & b_n \end{pmatrix}$$

then the product AB has the form

$$AB = \begin{pmatrix} a_1 b_1 & & & * \\ 0 & a_2 b_2 & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & a_n b_n \end{pmatrix}$$

$$BA = \begin{pmatrix} b_1a_1 & & & * \\ 0 & b_2a_2 & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & b_na_n \end{pmatrix}$$

Therefore, the commutator $AB - BA$ is strictly upper triangle

$$AB - BA = \begin{pmatrix} a_1b_1 & & & * \\ 0 & a_2b_2 & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & a_nb_n \end{pmatrix} - \begin{pmatrix} b_1a_1 & & & * \\ 0 & b_2a_2 & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & b_na_n \end{pmatrix}$$
$$= \begin{pmatrix} 0 & & & * \\ 0 & 0 & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & 0 \end{pmatrix}$$

The elements of G′ consist of strictly upper triangular matrices. We see that the elements of the G″ matrix whose entries are 0 below the diagonal are 2 steps above the main diagonal; that is, G″ consists of matrices a_{ij} such that $a_{ij} = 0$ whenever $i \geq j - 1$. In general, G^i matrices have 0 below the diagonal 2^{i-1} steps above the main diagonal.

Let E_{ij} be an $n \times n$ matrix whose (i, j) entry is 1 and all other entries are 0. E_{ij} satisfy the multiplication rules,

$$E_{ij}E_{kl} = \gamma_{jk}E_{il}$$

and so

$$[E_{ij}E_{kl}] = E_{ij}E_{kl} - E_{kl}E_{ij} = \gamma_{jk}E_{il} - \gamma_{li}E_{kj}$$

$$T_n(F) = \bigoplus (FE_{ij})$$

$$i \leq j$$

Let G_r denote the subspace $T_n(F)$ consisting of those matrices whose entries below the diagonal are r steps above the main diagonal 0. Then

$$G_r(F) = \bigoplus (FE_{kl})$$

$$k \leq l - r$$

For $r \geq 1$ we will show that if E_{ij} and E_{kl} are in G_r, then $[E_{ij}E_{kl}] \in G_{r+1}$. The matrix product $E_{ij}E_{kl}$ lies in G_{r+1}. $E_{ij}E_{kl}$ is nonzero if and only if $j = k$, in which case the product is E_{il}. But this means that $i \leq j - r = k - r \leq l - 2r \leq l - (r + 1)$, since $r \geq 1$.

So $E_{il} \in G_{r+1}$. It is shown that for all $r \geq 0$ $[G_r, G_r] \subset G_{r+1}$, and therefore $G = T_n(F)$ is also solvable.

5.0 PAIRING LIE ALGEBRA WITH CRYPTOGRAPHY

Let E be an elliptic curve defined over a finite field F_p with a prime order subgroup G, and let P and Q be points on E, and the Lie algebra associated with E is generated by tangent vectors at the identity element. A Weil pairing on Lie algebra is defined as $\tau: G \times G \to F^*_{p^k}$, where k is the embedding degree.

For points P and Q in G, the Weil pairing is calculated as: $\tau(P, Q) = \zeta_r^{trace}(\alpha P, Q)$ where ζ_r is a primitive $r-$th root of unity, $\alpha_{P,Q}$ is the rational function associated with divisor $(P) - (O) - (Q) + (P + Q)$.

This is general structure of a Lie Algebra Symmetric Bilinear Pairing, specifically the Weil pairing on elliptic curves. The actual implementation details and security considerations can be more complex and often involve additional parameters and operations.

It's important to note that cryptographic protocols using such pairings should be designed and implemented carefully to ensure security against various attacks.[14].

Proposal 5.1. A scheme based on identity encryption using symmetric bilinear matching Lie algebra.

First of all, one should choose a safe elliptic curve E defined over a finite field F_p with a symmetric bilinear pair $\tau: G \times G \to F^*_{p^k}$, where G is a subgroup of E. After that, one should fix the public parameters including the equation of the elliptic curve, generating point and matching function τ. The master key $g \in Z_q$ should be generated as a random element in the subgroup q of G. Then the master key should be calculated as gG. The user must then register where his public identity such as an email address can be used as input to a cryptographic hash function to obtain the point P_i for the elliptic curve. After that, the user calculates his private key as $D_k = gP_i$ where g is the master key and P_i is a point derived from his identity. To encrypt a message M, the sender randomly chooses $r \epsilon Z_q$ and calculates the ciphertext as $C = M \oplus \tau(P_i, rG)$[15]

To decrypt the ciphertext C, the user computes the matching $\tau(D_k, rG)$ and uses it to recover the original message M as: $M = C \oplus \tau(D_k, rG)$.

Creating a complete implementation of a symmetric bilinear elliptic curve and its associated Lie algebra for cryptography and key exchange involves multiple steps, and the code can be quite extensive. Below is a simplified example of using Python and the pycryptodome library for cryptographic operations.

```
from Crypto.Util.number import getPrime

from sympy import mod_inverse

from hashlib import sha256

class EllipticCurvePoint:
```

```
def __init__(self, x, y, a, b, p):
self.x = x
self.y = y
self.a = a
self.b = b
self.p = p
def __add__(self, other):
if self == EllipticCurvePoint.infinity():
return other
if other == EllipticCurvePoint.infinity():
return self
if self.x == other.x and self.y != other.y:
return EllipticCurvePoint.infinity()
if self != other:
m = (other.y - self.y) * mod_inverse(other.x - self.x, self.p)
else:
m = (3 * self.x**2 + self.a) * mod_inverse(2 * self.y, self.p)
x3 = (m**2 - self.x - other.x) % self.p
y3 = (m * (self.x - x3) - self.y) % self.p
return EllipticCurvePoint(x3, y3, self.a, self.b, self.p)
def __eq__(self, other):
return self.x == other.x and self.y == other.y
@staticmethod
def infinity():
return EllipticCurvePoint(None, None, None, None, None)
```

```
class SymmetricBilinearPairing:
def __init__(self, G, p):
self.G = G
self.p = p
def pairing(self, P, Q):
if P == EllipticCurvePoint.infinity() or Q == EllipticCurvePoint.infinity():
return 1
e = pow((P.y * Q.y) % self.p, ((P.x * Q.x) % self.p + (P.x * Q.x) % self.p) // 2, self.p)
return e
# Example usage
if __name__ == "__main__":
# Define elliptic curve parameters
a = 2
b = 2
p = get Prime (128)
# Choose a base point on the curve
G = EllipticCurvePoint(3, 5, a, b, p)
# A's private key
al_private_key = 123
# Compute A's public key
a_public_key = G
for _ in range(a_private_key - 1):
a_public_key += G
# B's private key
b_private_key = 456
```

```
# Compute B's public key

b_public_key = G

for _ in range(b_private_key - 1):

b_public_key += G

# Symmetric bilinear pairing

pairing = SymmetricBilinearPairing(G, p)

# Shared secret computation

shared_secret_a = pairing.pairing(b_public_key, a_public_key)

shared_secret_b = pairing.pairing(a_public_key, b_public_key)

# Check if shared secrets match

assert shared_secret_a == shared_secret_b

# Derive a key from the shared secret using a hash function (e.g., SHA-256)

derived_key = sha256(str(shared_secret_a).encode()).digest()

print("Shared secret:", shared_secret_a)

print("Derived key:", derived_key)  [15]
```

Lemma 5.2. For $x, y \in B$, we have je [x,x] = 0 and $\big[x,[y,z]\big] + \big[y,[z,x]\big] + \big[z,[x,y]\big] = 0.$

Proof. Let $x \in V_i, y \in V_j$ and $z \in V_k$. Let $g \in k_i(G)$, $h \in k_j(G)$ and $n \in k_p(G)$ be preimages for x under T_i, y under T_j and z under T_p. Then

$$[x,x] = T_{2i}((l,l)) = T_{2i}(1) = 0\ i$$

$$\big[x,[y,z]\big] + \big[y,[z,x]\big] + \big[z,[x,y]\big] = \big[x,(T_{j+p}(m,n)\big] + \big[y,(T_{i+p}(l,n)\big] + \big[z,(T_{i+j}(l,m)\big]$$

$$= T_{j+p}\big((m,n)\big) + T_{j+p}\big((n,m)\big) + T_{i+p}\big((l,n)\big) + T_{i+p}\big((n,l)\big) + T_{i+j}\big((l,m)\big) + T_{i+j}((m,l))$$

$$= T_{j+p}\big((m,n)(n,m)\big) + T_{i+p}\big((l,n)(n,l)\big) + T_{i+j}\big((l,m)(m,l)\big)$$
$$= T_{j+p}(1) + T_{i+p}(1) + T_{i+j}(1)$$

$$0 + 0 + 0 = 0$$

Lemma 5.3. For $x_1, x_2, x_3 x_4 \in L$, we have

$$\left[\boldsymbol{x_1},[\boldsymbol{x_2},[\boldsymbol{x_3},\boldsymbol{x_4}]]\right]+\left[\boldsymbol{x_2},[\boldsymbol{x_3},[\boldsymbol{x_4},\boldsymbol{x_1}]]\right]+\left[\boldsymbol{x_3},[\boldsymbol{x_4},[\boldsymbol{x_1},\boldsymbol{x_2}]]\right]+[\boldsymbol{x_4},[\boldsymbol{x_1},[\boldsymbol{x_2},\boldsymbol{x_3}]]]=\boldsymbol{0}$$

Proof. Assume that $x_i \in V_{m_i}$ for $i = 1,2,3.4.$ Let $g_i \in k_{m_i}(G)$ be the preimage of x_i under T_i for i = 1; 2; 3;4.

Then

$$\left[\boldsymbol{x_1},[\boldsymbol{x_2},[\boldsymbol{x_3},\boldsymbol{x_4}]]\right]+\left[\boldsymbol{x_2},[\boldsymbol{x_3},[\boldsymbol{x_4},\boldsymbol{x_1}]]\right]+\left[\boldsymbol{x_3},[\boldsymbol{x_4},[\boldsymbol{x_1},\boldsymbol{x_2}]]\right]+[\boldsymbol{x_4},[\boldsymbol{x_1},[\boldsymbol{x_2},\boldsymbol{x_3}]]]$$

$= T_{m1+m2+m3+m4}\left((g_1,(g_2,g_3,g_4))(g_2,(g_3,g_4,g_1))(g_3,(g_4,g_1,g_2))(g_4(g_1,g_2,g_3)\right)$

Now the result follows from the fact that

$$\left((g_1,(g_2,g_3,g_4))(g_2,(g_3,g_4,g_1))(g_3,(g_4,g_1,g_2))(g_4(g_1,g_2,g_3)\right) \in k_{m1+m2+m3+m4+1}(G):$$

Corollary 5.4. With the product [;] : $L \times L \rightarrow L$ vector space L becomes a Lie algebra.

Example 5.5. [17]: Let G be the group generated by three elements g^1, $g^2, g3$ subject to the relations

$(g_2,g_1) = g_3,(g_3,g_1) = (g_3,g_2) = 1, g_1^2 = g_2^2 = g_3$ and $g_3^2 = 1.$

The frst relation is the same as $g_2g_1 = g_1g_2g_3$, whereas the second and third relations allow us to rewrite any word in the generators to an expression of the form $g_1^{*1}\ g_2^{*2}g_3^{*3}(*)$.

Using the remaining relations, we can rewrite this to be a word of the form (*) where $0 \leq i_k \leq 1$. Hence

G contains $2^3 = 8$ elements. The Jennings series of G is $k_1(G) = G$; $k_2(G) = \langle g_3\rangle$, and $k_3(G) = 1$. So $G_1 = \frac{G}{\langle g_3\rangle} = \langle \bar{g}_1, \bar{g}_2\rangle$, where $\bar{g}_2 : \bar{g}_1 = \bar{g}_1\bar{g}_2$. Therefore $G_1 = \{1, \bar{g}_1, \bar{g}_2, \bar{g}_1\bar{g}_2\}$ Let V_1 be a 2-dimensional vector space over F_2 spanned by $\{e_1,\ e_2\}$. Let $\sigma_1 : G_1 \rightarrow V_1$ be the morphism given by $\sigma_1(\bar{g}_i) = e_i, i = 1,2$ (so $\sigma(\bar{g}_1\bar{g}_2) = e_1 + e_2$). Also we have that $G_2 = \frac{\langle g_3\rangle}{1} = \{1, g_3\}$. Let V_2 be a 1-dimensional vector space over F_2 spanned by e_3. Then $\sigma_2 : G_2 \rightarrow V_2$ is given by $\sigma_2(g_3) = e_3$. Now let $L = V1 \oplus V2$. We calculate the Lie product of e_1 and e_2:

$$[e_1,\ e_2] = T_2((g_1,\ g_2)) = T_2(g_3) = e_3:$$

Similarly it can be seen that $[e_1,\ e_3] = [e_2,\ e_3] = 0$

6.0 CONCLUSION AND FURTHER WORKS

Our exploration of ECC using Lie algebras has shed light on the promising intersection between these two fields of mathematics and cryptography. Through our investigation, we have demonstrated the potential benefits and insights that Lie algebras offer in the realm of elliptic curve-based cryptographic systems. Firstly, we have shown that Lie algebras provide a rich mathematical framework for understanding the underlying structures and symmetries inherent in elliptic curves. By leveraging Lie algebras, we gain deeper insights into the geometric properties of elliptic curves, which can be harnessed to enhance the security and efficiency of cryptographic protocols.

Moreover, our analysis has revealed that Lie algebraic techniques can be applied to various aspects of elliptic curve cryptography, including key generation, encryption, and decryption processes. By integrating Lie algebraic methods into cryptographic algorithms, we can potentially improve computational efficiency, mitigate security vulnerabilities, and adapt to emerging threats in the digital landscape. Furthermore, our examination of practical implementations has demonstrated the feasibility of incorporating Lie algebraic concepts into existing elliptic curve cryptographic systems. While challenges and complexities exist in translating theoretical insights into practical applications, our findings suggest that Lie algebras hold promise as a valuable tool for advancing the state-of-the-art in cryptographic research and development.

Overall, the convergence of elliptic curve cryptography and Lie algebras represents an exciting frontier in modern cryptography, with implications for a wide range of applications, including secure communication, digital signatures, and cryptographic protocols in emerging technologies such as blockchain and IoT (Internet of Things).

Looking ahead, there are several avenues for further research and exploration in the field of elliptic curve cryptography using Lie algebras. Investigating techniques to optimize cryptographic algorithms based on Lie algebras for improved performance and scalability, particularly in resource-constrained environments such as IoT devices and mobile platforms. Conducting rigorous security analyses to assess the resilience of Lie algebra-based elliptic curve cryptographic schemes against known attacks and vulnerabilities, and exploring new cryptographic primitives and protocols inspired by Lie algebraic structures. Advocating for the integration of Lie algebraic techniques into cryptographic standards and protocols, and fostering collaboration between mathematicians, cryptographers, and industry stakeholders to promote widespread adoption of advanced cryptographic techniques. Exploring interdisciplinary collaborations between mathematicians, computer scientists, and experts in related fields to leverage insights from diverse domains and develop innovative approaches to cryptographic design and analysis. Promoting awareness and understanding of elliptic curve cryptography and Lie algebras through educational initiatives, workshops, and outreach programs aimed at students, researchers, and practitioners in academia and industry. By addressing these research challenges and opportunities, we can unlock new frontiers in cryptographic theory and practice, paving the way for enhanced security, privacy, and trust in the digital ecosystem. Through continued innovation and collaboration, we can harness the power of mathematics to advance the science of cryptography and address the evolving security challenges of the digital age.

REFERENCES

E. Khamseh „Bilinear cryptography using Lie algebras from p-groups“ Vol 2(1), 2021, pp:71-77DOI:10.30511/mcs.2021.522222.1015https://mcs.qut.ac.ir/article_242145_44ca125b9078901a69de5d7d646e86f1.pdf

A. Dujella "Elliptic curves in cryptography" PMF - MO, University of Zagreb, 2013. https://web.math.pmf.unizg.hr/~duje/elkript/elkripto2.pdf

D. Turković "Cryptographic procedures based on elliptic curves" (2023) University of Rijeka Graduate thesis, Faculty of Engineering Graduate University study of computer science.https://www.unirepository.svkri.uniri.hr/islandora/object/riteh%3A4174/datastream/PDF/view

K. Kunjadić-Ćulibrk "Cryptography of Elliptic Curves" master's thesis, (2016), Singidunum University, Department of Postgraduate Studies.

A. Dujela "Algorithms for elliptic curves" Postgraduate course 2008/2009 https://web.math.pmf.unizg.hr/~duje/elipticke/algelip.pdf

D. Sejdinović "Elliptic curves in cryptography" (2007), 85–97 https://www.researchgate.net/publication/27194713_Elipticke_krivulje_u_kriptografiji

A. Menezes „An Introduction to Pairing-Based Cryptography“ (1991) Mathematics Subject Classification. Primary 94A60. https://www.math.uwaterloo.ca/~ajmeneze/publications/pairings.pdf

A. Joux, “A one round protocol for tripartite Diffie-Hellman”, Algorithmic Number Theory: 4th International Symposium, ANTS-IV, Lecture Notes in Computer Science, 1838 (2000), 385–393. Full version: Journal of Cryptology, 17 (2004), 263–276 https://www.researchgate.net/publication/226904597_A_One_Round_Protocol_for_Tripartite_Diffie-Hellman

E. Verheul, “Evidence that XTR is more secure than supersingular elliptic curve cryptosystems”, Journal of Cryptology, 17 (2004) 277–296. https://link.springer.com/article/10.1007/s00145-004-0313-x

D. Boneh, B. Lynn and H. Shacham, “Short signatures from the Weil pairing”, Advances in Cryptology – ASIACRYPT 2001, Lecture Notes in Computer Science, 2248 (2001), 514–532. Full version: Journal of Cryptology, 17 (2004), 297–319. https://www.iacr.org/archive/asiacrypt2001/22480516.pdf

A. Menezes, T. Okamoto, S.A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Transactions on Information Theory 39(5)1993, 163-1646 https://www.dima.unige.it/~morafe/MaterialeCTC/p80-menezes.pdf

A. Hasić „Introduction to Lie Algebras and Their Representations“ Advances in Linear Algebra & Matrix Theory , 11, 67-91. https://doi.org/10.4236/alamt.2021.113006

Renee Talley, A. (2017) An Introduction to Lie Algebra. California State University, San Bernardino, 38-40. https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1668&context=etd

W.A. Graff, Lie algebras: theory and algorithms, Elsevier, 2000.

M. N. John, U. Otobong. G.A. Musa „ Symmetric Bilinear Cryptography On Elliptic Curve And Lie Algebra"Vol. 06 Issue 10 Oct – 2023. Int. J. Mathematics. 06(10), 01-15 https://www.researchgate.net/publication/375838006_SYMMETRIC_BILINEAR_CRYPTOGRAPHY_ON_ELLIPTIC_CURVE_AND_LIE_ALGEBRA

Roberts, B. (2018-2019) „Lie Algebras". University of Idaho, Moscow, 56-57, 85-86. https://www.freebookcentre.net/maths-books-download/Lie-Algebras-by-Brooks-Roberts.html

B. Huppert, N. Blackbum, „Finite Groups II", Springer-Verlag Berlin Heidelberg New York, 1982 https://archive.org/details/finitegroupsii0000hupp/page/538/mode/2up

A.Hasić, A „An Introduction to Lie Groups"(2020), Advances in Linear Algebra & Matrix Theory , 10, 35-51. https://doi.org/10.4236/alamt.2020.103004