International Journal of Scientific Research in Mathematical and Statistical Sciences
Vol.**11**, Issue.**4**, pp.**29-33**, August **2024**
E-ISSN: 2348-4519
Available online at: www.isroset.org

Research Article

Design and Formulation for Multi-Party Computation in Elliptic Curve Cryptography

Domven Lohcwat[1*], D.G. Yakubu[2], M.I. Bello[3]

[1,2,3]Dept. of Mathematics, Abubakar Tafawa Balewa University, Bauchi

**Corresponding Author:lohcwatlar@gmail.com. +2348034205642*

Received: 07/Jun/2024; **Accepted**: 09/Jul/2024; **Published**: 31/Aug/2024

Abstract— This article discusses the use of elliptic curve cryptography (ECC) in creating secure communication schemes that do not require a pre-shared secret key. ECC, a type of public key cryptography, supports applications such as key agreement and digital signatures. Traditionally, secure communication necessitates a secret key exchange beforehand, but the paper demonstrates a method where multiple parties can publicly select a key without others being able to determine it. The proposed protocol leverages elliptic curves and secure multiparty computation (MPC) to allow secure communication over insecure channels, even in the presence of adversaries. MPC protocols enable parties to jointly compute functions of their private inputs while only revealing the output, with potential applications in privacy-preserving auctions, private DNA comparisons, private machine learning, and threshold cryptography.

Keywords— Multiparty Computation, Elliptic Curve, Elliptic Curve Cryptography and Cryptography

1. Introduction

Elliptic curve cryptography is a form of public key cryptography that relies on the algebraic structure of elliptic curves over finite fields [1,2]. Elliptic curves can be defined over two types of finite fields: prime fields F_p where p is a large prime number, and binary fields F_2^m .This research focuses on the utilization of elliptic curves over prime fields $E(E_p)$. A non-supersingular elliptic curve E over F_p is defined as the solution of $(x, y) \in F_p \times F_p$ to the cubic equation $Y^2 = X^3 + AX + B \ mod\, p$ where $A, B \in F_p$ such that $4A^3 + 27B^2 \neq 0 \ mod\, p$ together with the special point ∞ called the point at infinity. The group of points constitutes an abelian group under the addition operation, ensuring that the sum of any two points lies on the same curve. The security of ECC-based cryptographic protocols relies on the elliptic curve discrete logarithm problem (ECDLP). ECDLP is the challenge of determining the scalar kkk such that R=kP, given that R and P are generator points. Private set computation is a critical component of secure multiparty computation, and secure multiset computation is practically significant [3]. The elliptic curve cryptosystem, a vital public key cryptosystem with additive homomorphism, plays an important role in secure multiparty computation [4,5,6,7, 8,9].

Let G be a group, with g∈G and h∈⟨g⟩. The discrete logarithm problem (DLP) in G involves finding the integer k such that h=g^k, where k is the discrete logarithm of h to the base g. For the group of points EEE on an elliptic curve, this problem is known as the elliptic curve discrete logarithm problem (ECDLP). The difficulty of solving the discrete logarithm problem in certain groups forms the foundation of many cryptographic systems, including the Diffie-Hellman key exchange protocol, the ElGamal public-key encryption scheme, and the Digital Signature Algorithm (DSA). Public key cryptography relies on the concept of one-way functions, such as the exponential function in a large finite field.

2. Related Work

[10] initially introduced and explored the millionaire problem, marking the beginning of secure two-party computation. Subsequently, [11] introduced and analyzed the general secure multiparty computation (SMC) problem. Today, SMC is a highly active research area in cryptography, encompassing range queries [12, 13], location queries

[14,15], set problems [16, 17], and vector problems [18,19]. Multisets are extensively utilized in various fields. For instance, banks use multisets to record information about customers' ages and occupations, while hospitals collect data on patients' blood types, blood pressure, and heart rates. This information is sensitive and private. When multiple parties need to perform collaborative computations using this data, a secure multiparty computation protocol is essential. Multi-secret sharing schemes are employed to protect multiple secrets by distributing them among numerous participants, allowing reconstruction only by specific authorized groups of participants.

3. Experimental Method/Procedure/Design

Let's consider a scenario where multiple parties $P_i\,,\ \ldots\ ,P_n,$ $(i \in [1,n])$ and ($P_j\,,\ \ldots\ ,P_n),(j \in [1,n])\ and\ i \neq j$ want to compute a function $f(x_1\,,x_2\,,x_3\,,\ \ldots\ ,x_n)$ on their respective inputs $(x_1\,,x_2\,,x_3\,,\ \ldots\ ,x_n)$.

The objective is to compute the function's output without disclosing any individual inputs to the other parties. The key can be any random integer agreed upon by the parties or users, but it remains unknown to anyone else. A distinctive feature of public key cryptography for key exchange is that parties can establish a common key using public information. This allows anyone to send a message to a specific user using the same encryption key, which can be easily found in a public directory. Consequently, there is no need for the sender to have made any secret arrangements with the recipient, nor is any prior contact between the sender and the recipient necessary.

The algorithm(procedure) can be implemented in the multiplicatively written group of any finite field. We consider the most common implementation in a group of a prime field, $(\mathbb{Z}/p\mathbb{Z})$. It is important that $g\ \in (\mathbb{Z}/p\mathbb{Z})$ is a generator since we want to make sure the generated shared key at the end received from a power of g is any element of $(\mathbb{Z}/p\mathbb{Z})$.

3.1 Outline of the Propose Scheme

1. $P_{i's}$ and $P_{j's}$ agree on a prime modulus p and a generator g which are publicly known.
2. P_i selects or choose a private random number $'a$' such that $1 < a < p-1$ and calculate $A = g^a mod\ p$ sending the result publicly to P_j.
3. P_j select or choose his private number $'b$' such that that $1 < b < p-1$ and calculate $B = g^b mod\ p$ sending the result publicly to P_i.
4. P_i takes $P_{j's}$ public result B and raises it to the power of his private number obtaining $B^a mod\ p$.
5. P_j takes $P_{i's}$ public result A and raises it to the power of his private number obtaining $A^b mod\ p$.
6. We notice that ;

   $B^a mod\ p = (g^b)^a mod\ p = (g^a)^b mod\ p = A^b mod\ p = s$

   is a shared key.

We note that only a and b are private knowledge, while all other values used during the exchange are publicly available. Therefore, a potential third party (attacker) would have to work with g, A, and B to obtain $g^{ab} mod\ p$. However, computing this with only these values takes an extremely long time. If p is extremely large, even the fastest computers in the world would be unable to find "a" such that $A = g^a mod\ p$, given only A, p, and g, illustrating the difficulty of the discrete logarithm problem. Additionally, the use of the generator g in $g\ in\ (\mathbb{Z}/p\mathbb{Z})$ complicates the problem for an adversary, as the powers of g can be any element of the field, increasing the number of possible key choices. Because the message can be arbitrarily large, which may cause the asymmetric encryption process to be slow, the obtained shared secret key is used in symmetric encryption, allowing $P_{j's}$ and $P_{i's}$ to send messages across the same open communications channel. After computing their public keys $A\ and\ B$, P_i and $P_j$share these on an unsecured communication channel. The adversary (attacker) can see these computed values. Finally, P_i and P_j use their secret integers to compute:

$A' \equiv B^a(mod\ p)$ and $B' \equiv A^b(mod\ p)$. Note that it is clear that A' and B' are the same shared key since

$$A' \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \equiv B' mod\ p \tag{1}$$

3.2 Elliptic Curves and their Abelian Group Structure.

Elliptic curves are used in many mathematical fields such as cryptography solving Diophantine equations. The security of the schemes that use the group of points on elliptic curves, is based on the hardness of the discrete logarithm problem.

Let K be a field. An elliptic curve E defined over K is a smooth plane cubic curve giving by a long Weierstrass equation:

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

Where $a_1, a_2, a_3, a_4, a_6\ \in K$. The homogenization of the curve E is giving by:

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \tag{3}$$

The only point at infinity on this curve is [0: 1: 0], we denote this point by $\mathcal{O}$ from now on. This point is the neutral element in the group structure on E. If the $char(K) \neq 2{,}3$

then by a suitable change of variables, we have the short Weierstrass equation:

$$E: Y^2 = X^3 + AX + B \qquad (4)$$

Where $A, B \in K$. It is well-known that if E is a curve giving by the Weierstrass equation (4), then E is an elliptic curve if and only if its discriminant $\Delta = 4A^3 + 27B^2$ is nonzero. That is;

$4A^3 + 27B^2 \neq 0$.

We will now define the group structure on E. Let E be an elliptic curve over K, defined by equation (4). Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on E. The addition

$R = P + Q = (x_3, y_3)$ is defined as follows:

- If $x_1 = x_2$ and $y_1 \neq y_2$, then $R = \mathcal{O}$
- If $x_1 \neq x_2$, then $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$, where $\boldsymbol{\lambda} = \frac{y_2 - y_1}{x_2 - x_1}$
- If $P = Q$, and $y_1 = 0$, then $R = \mathcal{O}$
- If $P = Q$, and $y_1 \neq 0$ then $x_3 = \lambda^2 - 2x_1$ and $y_3 = \lambda(x_1 - x_3) - y_1$ where $\boldsymbol{\lambda} = \frac{3{x_1}^2 + A}{2y_1}$

The points on E form an additive abelian group with $\mathcal{O}$ as the identity element. We know that from the ElGamal cryptosystem which is a public key cryptosystem proposed by ElGamal in 1985, it is easy to create a direct analogue of the ElGamal public key cryptosystem. Two users say, A and B agree to use a particular prime p, elliptic curve E, and point $P \in E(F_p)$. A chooses a secret multiplier n_A and publishes the point $Q_A = n_A P$ as her public key, likewise B also chooses a secret multiplier n_B and publishes the point $Q_B = n_B P$ as her public key. $B's$ plaintext is a point $M \in E(F_p)$. He chooses an integer k to be his ephemeral key and computes:

$$C_1 = kP \; and \; C_2 = M + kQ_A. \qquad (4a)$$

He then sends the two points (C_1, C_2) to A who computes:

$$C_2 - n_A C_1 = M + \; kQ_A - n_A(kP) = M + k(n_A P) - n_A(kP) = M \qquad (4b)$$

to recover the plaintext. An adversary who can solve the ECDLP can of course, determine n_A from the publicly known information n_A and $n_A P$ but as everybody knows, there is no efficient way to compute discrete logarithms, so the system is secure, [20].

3.3 Design and formulation.

We first gave an intuition on how one might compute a function securely without relying on trusted parties. This requires that we specify a protocol, i.e., a set of instructions that participants are supposed to follow to obtain the desired result. For simplicity, we will assume for now that participants always follow the protocol. We will also assume that any pair of participants can communicate securely, i.e., it is possible for P_i to send a message M to P_j, such that no third party sees M and P_j knows that M came from P_i. Let $n (n \geq 2)$ be the number of participants joining a distributed computing network over insecure channel, in which the i^{th} party keeps a private input $v_i (i = 1, \; . \; . \; . \; , \; n)$, and all inputs have the same length $|v_i| = |v_j|$ with $\forall i, j$. The multiparty computation function f is defined as follows:

$$f : \left(\{0,1\}^*\right)^n \to \left(\{0,1\}^*\right)^n$$
$$\bar{v} = (v_1, \; . \; . \; . \; , v_n) \to f(\bar{v})$$
$$= f_1(\bar{v}, \; . \; . \; . \; , f_n(\bar{v})) \qquad (5)$$

The i^{th} party who owns the private input value v_i wishes to obtain the i^{th} element in $f(v_1 \; . \; . \; . \; , \; v_n)$ that is $f_i(v_1 \; . \; . \; . \; , \; v_n)$ denoted as y_i. A multi-party computation function f can fall into one of the following types:

Deterministic functions: this returns a unique output with the same input value, and includes: i). Symmetric deterministic functions are deterministic functions in which $f_i(v_1 \; . \; . \; . \; , \; v_n) \equiv f_j(v_1, . \; . \; . , v_n)$ with $\forall i \neq j$. (ii). Asymmetric deterministic functions are deterministic functions where $f_i(v_1 \; . \; . \; . \; , \; v_n) \neq f_j(v_1, . \; . \; . , v_n)$ with $\forall i \neq j$. General functions (including both deterministic and indeterministic functions) that can return different outputs with the same input in different executions.

4. Results and Discussion

Suppose that, there is an agreement among the communicants, $P_i, \; . \; . \; . \; . \; , P_n, \quad (i \in [1, n])$ and ($P_j, \; . \; . \; . \; . \; , P_n), (j \in [1, n]) \; and \; i \neq j$ to communicate on an insecure communication channel where they agreed to use the prime $E(F_p) = 32611$ and the primitive root $g = 1009$. P_i and P_j chooses their secret-keys as $n_A = 1139 \; and \; n_B = 3589$ respectively. They then compute their public-keys $A \; and \; B$ as follows:

$A \equiv g^{n_A} \equiv 7509 \equiv 1009^{1139} (mod \; 32611)$, and

$B \equiv g^{n_B} \equiv 31004 \equiv 1009^{3589} (mod \; 32611)$.

Thus, the computed value would now be share publicly.

The shared key k will now be:

$$4577 \equiv 1009^{(1139 \times 3589)} \equiv 7509^{3589} \equiv 31004^{1139} (mod \; 32611)$$

.

To see the communicated message, the adversary has to solve any of the congruence equations:

$1009^{n_A} \equiv 7509 \; (mod \; 32611)$ and $1009^{n_B} \equiv 31004 \; (mod \; 32611)$. The adversary can

solve these values of n_A and n_B if p is small prime. In practice, the p value must be at least 1024 bits length, [21]. Then it will be difficult for the adversary to find the secret keys. The calculated value now forms an elliptic curve thus: $E: Y^2 = X^3 + 7509X + 31004$, which off course satisfy the discriminant condition of an elliptic curve. Now consider the curve $E: Y^2 = X^3 + 7509X + 31004$ formulated above, assuming that P_i and P_j want to communicate a secured secret message $M = (11734, 8957)$ through an insecure communication channel, then they have to agree on a particular elliptic curve $E(F_p) = F_{32611}$ and a specific public base point $P = (920, 303)$ on $E(F_p)$. P_i then choose randomly a secret (key) integer $n_A = 1139$ and does not reveal it to anyone. Likewise, P_j also choose randomly a secret (key) integer $n_B = 3589$ and keeps it confidential to himself. They then use their secret key to compute their public key $Q_A \; and \; Q_B$ as follows:
$Q_A = n_A P \; and \; Q_B = n_B P$, and the shared secret value: $n_A Q_B = (n_A n_B)P = n_B Q_A$ which they can use as a key to communicate privately via a symmetric cipher.

$$Q_A = n_A P = 1139 * (920, 303) = (1551, 359) \quad (mod\ 32611) \tag{6}$$

$$Q_B = n_B P = 3589 * (920, 303) = (5231, 28189) \quad (mod\ 32611) \tag{7}$$

P_i then sends Q_A to P_j and P_j also sends Q_B to P_i and they again computes:

$$P_i: n_A \; Q_B = 1139 * (5231, 28189) = (30305, 25466) \quad (mod\ 32611). \tag{8}$$

$$P_j: n_B \; Q_A = 3589 * (1551, 359) = (30305, 25466) \quad (mod\ 32611). \tag{9}$$

Thus P_j and P_i have exchange the secret point: $n_A \; Q_B = (n_A n_B)P = n_B \; Q_A =$ (30305, 25466) $(mod\ 32611)$. (10)

Note that:
$C = M + n_B(n_A P) = (11734, 8957) + (30305, 25466) = (28004, 17212)$. Of course, for P_j to decrypt the message, he computes: $n_A(n_B P) = (30305, 25466)$. Finally, he computes:

$$C - n_A(n_B P) = M + n_B(n_A P) - n_A(n_B P) = M$$

$$(28004, 17212) - (30305, 25466) = (28004, 17212) + (30305, 25466) = (11734, 8957) = M.$$

5. Conclusion and Future Scope

In this paper, we propose a method for secure multiparty computation using elliptic curve cryptography. The purpose of public key validation is to confirm that a public key has certain arithmetic properties. Successful validation indicates the logical existence of an associated private key, although it does not confirm that the private key has been computed or that the claimed owner possesses it. Public key validation is crucial in key establishment protocols, where an entity A derives a shared secret k by combining her private key with a public key received from another entity B, and then uses k in a symmetric key protocol (e.g., encryption or message authentication). Our secure multiparty protocol offers improved efficiency in terms of communication cost compared to secret sharing-based approaches, making it scalable with respect to the number of parties involved.

Conflict of Interest

The author states that there are no conflicts of interest related to this study.

Funding

This research was self-funded.

Authors' Contributions

All authors made significant contributions to this work.

References

[1] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. **48,** pp. **203-209, 1987**. [Online]. V. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO '85*, vol**. 218**, pp. **417-426, 1986**.

[2] J. Pan and J. Dou, "Secure multiparty multisets computation," *International Journal of Network Security*, vol. **25**, no. **3**, pp**. 425-430, 2023.**

[3] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Berlin, Germany: Springer Science & Business Media, **2006.**

[4] L. C. Huang and M. S. Hwang, "Two-party authenticated multiple-key agreement based on elliptic curve discrete logarithm problem," *International Journal of Smart Home*, vol. **7**, no. **1**, pp. **9-18, 2013**.

[5] M. S. Hwang, E. F. Cahyadi, et al., "An improvement of the remote authentication scheme for anonymous users using an elliptic curve cryptosystem," in *Proc. IEEE 4th International Conference on Computer and Communications*, Chengdu, China, **2018**, pp. **1872-1877.**

[6] M. S. Hwang, C. C. Lee, J. Z. Lee, and C. C. Yang, "A secure protocol for Bluetooth piconets using elliptic curve cryptography," *Telecommunication Systems*, vol. **29**, no. **3**, pp. **165-180, 2005**.

[7] M. S. Hwang, S. F. Tzeng, and C. S. Tsai, "Generalization of proxy signature based on elliptic curves," *Computer Standards & Interfaces*, vol. **26**, no. **2**, pp. **73-84, 2004**.

[8] C. C. Yang, T. Y. Chang, and M. S. Hwang, "A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem," *Computer Standards & Interfaces*, vol. 25, no. 2, pp. 141-145, 2003.

[9] A. Yao, "Protocols for secure computations," in *Proc. 23rd IEEE Symposium on Foundations of Computer Science*, Chicago, IL, USA, 1**982**, pp. **160-164.**

[10] S. Goldwasser, "Multi-party computations: past and present," in *Proc. 16th Annual ACM Symposium on Principles of Distributed Computing*, New York, USA**, 1997**, pp. **1-6**.

[11] L. H. Liu and Z. J. Cao, "Analysis of a privacy preserving ranked multi-keyword search scheme," *I.J. Electronics and Information Engineering*, vol. **12**, no**. 2**, pp. **76-82, 2022**.

[12] Y. G. Peng, L. Wang, et al., "LS-RQ: A lightweight and forward-secure range query on geographically encrypted data," *IEEE Trans. Dependable Secur. Comput.*, vol. **19**, no**. 1**, pp. **388-401, 2022**.

[13] Y. G. Guan, R. X. Lu, et al., "Toward oblivious location-based k-nearest neighbor query in smart cities," *IEEE Internet Things J.*, vol. 8, no. **18**, pp. **14219-14231, 2021**.

[14] S. N. Zhang, R. X. Lu, et al., "Preserving location privacy for outsourced most-frequent item query in mobile crowdsensing," *IEEE Internet Things J.*, vol. **8**, no**. 11**, pp. **9139-9150, 2021**.

[15] C. Melissa and M. Peihan, "Private set intersection in the internet setting from lightweight oblivious PRF," in *Proc. 40th Annual International Cryptology Conference*, Santa Barbara, CA, USA, **2020**, pp. **34-63**.

[16] M. Peihan, P. Sarvar, et al., "Two-sided malicious security for private intersection-sum with cardinality," in *Proc. 40th Annual International Cryptology Conference*, Santa Barbara, CA, USA, **2020**, pp. **3-33.**

[17] D. Josep, R. Sara, et al., "Outsourcing scalar products and matrix products on privacy-protected unencrypted data stored in untrusted clouds," *Information Science*, vol. **436**, no. **1**, pp. **320-342**, **2018**.

[18] O. Tatsuaki and T. Katsuyuki, "Adaptively attribute-hiding (hierarchical) inner product encryption," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol**. 99**, no. **1**, pp. **92-117, 2016**.

[19] S. Y. Yan, *Elementary Number Theory*. Number Theory for Computing. Berlin, Germany: Springer, , pp.**1-172, 2002**

[20] J. H. Silverman, *The Arithmetic of Elliptic Curves*, vol. **106**. Berlin, Germany: Springer Science & Business Media, , pp. **1, 11, 36, 39, 40, 2009**