Journal of Engineering Design and Computational Science (JEDCS)

Volume 3, Issue 2, April2024

Extended Diffie-Hellman Key Exchange with Pairing Cryptography to Multiple users

ASSOUJAA ISMAIL[1], EZZOUAK SIHAM[2]

1&2Sidi Mohammed Ben Abdellah University, Faculty of science Dhar El Mahrez, Department of mathematics, Lab: LASMA, Fez, Morocco.

E-mail: [1]ismail.assoujja@usmba.ac.ma, [2]siham.ezzouak@usmba.ac.ma

Abstract - Cryptography plays a pivotal role in ensuring secure communication between two parties, typically represented by Alice and Bob, in the presence of adversaries like Eve. However, modern communication often involves multiple users beyond the traditional pair. This work extends the DiffieHellman key exchange protocol to accommodate multiple users, addressing the need for secure communication in diverse group settings. Additionally, we propose methods for communicating the input points utilized in the pairing application with the DiffieHellman key exchange protocol, enhancing the security and resilience of the exchanged keys. Through these advancements, we aim to fortify cryptographic protocols for robust multi-user scenarios, safeguarding sensitive information in an ever-evolving digital landscape.

Keywords: Pairing, Diffie-Hellman, Multi-users, ...

I. INTRODUCTION

In the realm of secure communication, cryptography serves as an indispensable tool, facilitating confidential exchanges among trusted parties. Diffie-Hellman (D.H) key exchange stands as a seminal method for securely exchanging cryptographic keys over public channels and represents one of the earliest practical examples of public-key protocols. Originally conceived to enable encrypted communication between two parties, the DiffieHellman key exchange method revolutionized secure communication by allowing parties to jointly establish shared secret keys without prior acquaintance. This shared key, established over an insecure channel, enables subsequent communications to be encrypted using symmetric key ciphers, contributing to the security of various Internet services. As the exchange of cryptographic keys over public channels forms a cornerstone of secure communication, our study extends beyond conventional two-party scenarios to address the implications and challenges of multi-user environments in key exchange protocols. Drawing inspiration from previous research, this paper explores the intricacies of multiuser Diffie-Hellman key exchange, aiming to enhance security and resilience in cryptographic protocols. To provide the necessary background for our investigation, Section 2 of this paper delves into mathematical foundations. In Section 3, we introduce Diffie-Hellman Key Exchange Protocol and it's extended form for multi-users. In Section 4, we will introduce the new method for communicating keys based on the DiffieHellman key exchange protocol and pairing cryptography, and extended the algorithm for multiusers. Finally, in Section 5, we conclude our paper by summarizing our main findings in this area.

II. MATHEMATICAL BACKGROUND

Throughout the paper, we will use the following conventions without explicitly stating them e: pairing.

- enc: encryption function.
- dec: decryption function.
- K: Key.
- PT: Plaintext.
- CT: Ciphertext.
- KPUB:Public Key.
- KPRI : Private Key.
- p: A prime number.
- D-H: Diffie-Hellman.

$\mathbb{G}_1 \subset (E(\mathbb{F}_q))$: additive group of cardinal n
$\mathbb{G}_2 \subset (E(\mathbb{F}_{q^k}))$: additive group of cardinal n
$\mathbb{G}_3 \subset \mathbb{F}^*_{q^k} \subset \mu_n$: cyclic multiplicative group
$n \in \mathbb{N}^*$.
$\mu_n = \{u \in \bar{\mathbb{F}_q} | u^n = 1\}$.

- P_∞ : the point at infinity of the elliptic curv

A. Pairing definition and proprieties

Property 1: [16], We let $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ and $(\mathbb{G}_3, .)$ denote three finite abelian groups of order r.

- Bilinearity: for all $S, S_1, S_2 \in \mathbb{G}_1$ and for all $T, T_1, T_2 \in \mathbb{G}_2$

$$e(S_1 + S_2, T) = e(S_1, T)e(S_2, T)$$
$$e(S, T_1 + T_2) = e(S, T_1)e(S, T_2)$$

- Non degeneracy: there exist $S \in \mathbb{G}_1$ with $S \neq 1, T \in \mathbb{G}_2$ with $e(S, T) \neq 1$.

Definition 1: [16], A pairing is a bilinear and non degenerate function:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_3$$
$$(P, Q) \longrightarrow e(P, Q)$$

III. MULTI-USERS DIFFIE-HELLMAN KEY EXCHANGE PROTOCOL

B. Diffie-Hellman key exchange protocol

Remark 1: In this paper, our primary objective is to extend the Diffie-Hellman Key Exchange to multiple users.

1

parties but not to any potential eavesdroppers.

2) Key Generation: Each party generates their own private key: a for party A and b for party B.

These keys are kept secret.

Each party computes their public key: Party A computes $A = g^a \bmod p$

Party B computes $B = g^b \bmod p$.

3) Exchange Public Keys: Both parties exchange their public keys A and B over the insecure channel.

The Diffie-Hellman key exchange (DH) is a foundational method for securely sharing cryptographic keys over a public channel, initially proposed by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. Unlike traditional methods requiring a secure physical channel for key exchange, DH enables two parties with no prior acquaintance to establish a shared secret key over an insecure channel. This shared key can subsequently encrypt communications using symmetric key ciphers. While widely utilized in Internet services, recent research indicates that some parameters used in DH implementations may be vulnerable to compromise by well-funded attackers, prompting concerns about the protocol's overall security.

Here's a simplified overview of how it works:

1) Initialization (Public Parameter Creation): Both parties agree on public parameters: a large prime number p and a primitive root g. These parameters are known to both 4) Secret Key Calculation:

Party A computes the shared secret key K using B^amodp Party B computes the shared secret key K using A^bmodp

Since $A^b = B^a$, both parties arrive at the same shared secret key K.

Both parties can now use the shared secret key K for encryption and decryption using symmetric encryption algorithms.

Alice & Bob can exchange their keys in just one step.

C. Three-party Diffie-Hellman (3DH protocol)

The Diffie-Hellman Tripartite Key Exchange is an extension of the traditional Diffie-Hellman key exchange protocol that enables three parties to establish a shared secret key. It's also known as the Three-party Diffie-Hellman or 3DH protocol.

Here's how it works:

1) Initialization (Public Parameter Creation): The parties agree on public parameters: a large prime number p and a primitive root g. These parameters are known to all parties but not to any potential eavesdroppers.

2) Key Generation: Each party generates their own private key: a for party A and b for party B and c for party C.

These keys are kept secret.

Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p and an integer g having large prime order in $\mathbb{F}_p^*$.	
Private Computations	
Alice	**Bob**
Choose a secret integer a.	Choose a secret integer b.
Compute $A \equiv g^a \pmod{p}$.	Compute $B \equiv g^b \pmod{p}$.
Public Exchange of Values	
Alice sends A to Bob $\longrightarrow$	A
B $\longleftarrow$	Bob sends B to Alice
Further Private Computations	
Alice	**Bob**
Compute the number $B^a \pmod{p}$.	Compute the number $A^b \pmod{p}$.
The shared secret value is $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$.	

Each party computes their public key: Party A computes A = g^amodp Party B computes B = g^bmodp. Party C computes C = g^cmodp. 3) Exchange Public Keys: All parties exchange their public keys A and B and C over the insecure channel.

4) Secret Key Calculation:

Party A computes the shared secret key A' using C^amodp

Public Parameter Creation		
A trusted party chooses and publishes a large prime p with g his generatrice		
Private Computations		
Alice	Bob	Charlie
Choose a secret integer a. Compute $A = g^a$	Choose a secret integer b. Compute $B = g^b$	Choose a secret integer c. Compute $C = g^c$
Public Exchange of Values		
All parties exchange their public keys A and B and C over the insecure channel.		
Further Private Computations		
Alice	Bob	Charlie
Compute $A' = C^a$	Compute $B' = A^b$	Compute $C' = B^c$
Public Exchange of Values		
All parties exchange their public keys A' and B' and C' over the insecure channel.		
Further Private Computations		
Alice	Bob	Charlie
Compute $K = (C')^a$	Compute $K = (A')^b$	Compute $K = (B')^c$
The shared secret value is $K = g^{abc}$		

Diffie-Hellman Tripartite key exchange cryptosystem.

Alice, Bob & Charlie exchange their keys in two steps.

Party B computes the shared secret key B' using A^bmodp

Party C computes the shared secret key C' using B^cmodp

5) Exchange Public Keys: All parties exchange their public keys A' and B' and C' over the insecure

6) Secret Key Calculation:

Party A computes the shared secret key K using $(C')^a$modp

Party B computes the shared secret key K using $(A')^b$modp

Party C computes the shared secret key K using $(B')^c$modp

Since $(C')^a = (A')^b = (B')^c$, all parties arrive at the same shared secret key K.

All parties can now use the shared secret key K for encryption and decryption using symmetric encryption algorithms.

D. n-party Diffie-Hellman key exchange protocol

K.

All parties can now use the shared secret key K for encryption and decryption using symmetric encryption algorithms.

Public Parameter Creation			
A trusted party chooses and publishes a large prime p with g his generatrice			
Private Computations			
User 1	User 2	...	User n
Choose a secret a_1. Compute $A_1 = g^{a_1}$	Choose a secret a_2. Compute $A_2 = g^{a_2}$	...	Choose a secret a_n. Compute $A_n = g^{a_n}$
Public Exchange of Values			
All parties exchange their public keys A_i over the insecure channel.			
Private Computations			
User 1	User 2	...	User n
Compute $A_1' = A_3^{a_1} = g^{a_1 a_3}$	Compute $A_2' = A_4^{a_2} = g^{a_2 a_4}$	...	Compute $A_n' = A_2^{a_n} = g^{a_2 a_n}$
Public Exchange of Values			
All parties exchange their public keys A_i' over the insecure channel.			
Repeat the same process for key calculation and exchanging n-2 more times			
Further Private Computations			
User 1	User 2	...	User n
Compute $g^{\Pi_{i=1}^n a_i}$	Compute $g^{\Pi_{i=1}^n a_i}$	...	Compute $g^{\Pi_{i=1}^n a_i}$
The shared secret value is $K = g^{\Pi_{i=1}^n a_i}$			

Diffie-Hellman n-users key exchange cryptosystem.

channel.

Extending the Diffie-Hellman key exchange protocol to accommodate more than two parties is a bit more complex but can be achieved using the below process:

1) Initialization (Public Parameter Creation): The parties agree on public parameters: a large prime number p and a primitive root g. These parameters are known to all parties but not to any potential eavesdroppers.

2) Key Generation: Each party generates their own private key: a_i for party A_i. These keys are kept secret.

Each party computes their public key:

Party Ai computes $A_i = g^{ai}$modp

3) Exchange Public Keys: All parties exchange their public keys A_i over the insecure channel.

4) Secret Key Calculation:

Party A computes the shared secret key A_i' using A_j^{ai} modp, j = i + 2modn

5) Exchange Public Keys: All parties exchange their public keys A_i' over the insecure channel.

6) Repeat the same process for key calculation and exchanging n-2 more times 7) Secret Key Calculation:

Each Ai computes the shared secret key K using $g^{ai\ldots an}$modp

All parties arrive at the same shared secret key

The n parties exchange their keys in n steps.

Remark 2: We can used this extend algorithm to communicate the keys between two parties

1) Initialization (Public Parameter Creation): Both parties agree on public parameters: a large prime number p and a primitive root g. These parameters are known to both parties but not to any potential eavesdroppers.

2) Key Generation: Each party generates their own private key: a_i for party A_i and a_j for party A_j.

These keys are kept secret.

Each party computes their public key: Party A_i computes A_i = g^{ai}modp Party A_j computes A_j = g^{aj}modp.

3) Exchange Public Keys: Both parties exchange their public keys A_i and A_j over the insecure channel. 4) Secret Key Calculation:

Party A_i computes the shared secret key K using A_j^{ai} modp

Party A_j computes the shared secret key K using A_i^{aj} modp

Since $A_j^{ai} = A_i^{aj}$, both parties arrive at the same shared secret key K.

Both parties can now use the shared secret key K for encryption and decryption using symmetric encryption algorithms.

IV. CONSTRUCTION OF MULTI-USERS DIFFIEHELLMAN KEY EXCHANGE PROTOCOL WITH PAIRING CRYPTOGRAPHY

E. Diffie-Hellman Key Exchange Protocol with Pairing Cryptography

Alice and Bob want to communicate using pairing application so they follow this method:

1) Public parameter creation:

Alice, Bob and Charlie publish a group G with g his generatrice and a pairing e. 2) Private parameter creation:

Alice chooses a secret a and calculate P = ag □ G.

Bob chooses a secret b and calculate Q = bg □ G.

3) Public exchange of values:

Alice and Bob receive and send the points P and Q to each other in a public channel of communication.

4) Further private computation:

Alice compute the value $e(P, Q)^a = e(g, g)^{ab}$.

Bob compute the value $e(P, Q)^b = e(g, g)^{ab}$.

The shared secret values are $K = e(g, g)^{ab}$.

Public Parameter Creation	
A trusted party chooses and publishes a large prime p with g his generatrice	
Private Computations	
Alice	Bob
Choose a secret integer a. Compute $P = a.g$	Choose a secret integer b. Compute $Q = b.g$
Public Exchange of Values	
All parties exchange their public keys A and B over the insecure channel.	
Further Private Computations	
Alice	Bob
Compute $e(P, Q)$	Compute $e(P, Q)$
The shared secret value is $K = e(g, g)^{ab}$	

Diffie-Hellman Key Exchange Protocol with Pairing Cryptography.

F. Three party Diffie-Hellman Key Exchange Protocol with Pairing Cryptography

Alice, Bob and Charlie want to communicate using pairing application so they follow this method:

1) Public parameter creation:

Alice, Bob and Charlie publish a group G with g his generatrice and a pairing e. 2) Private parameter creation:

Alice chooses a secret a and calculate P = ag □ G.

Bob chooses a secret b and calculate Q = bg □ G.

Charlie chooses a secret c and calculate R = cg □ G.

3) Public exchange of values:

Alice, Bob and Charlie receive and send the points P, Q and R to each other in a public channel of communication.

4) Further private computation:

Alice compute the value $e(Q, R)a = e(g, g)^{abc}$. Bob compute the value $e(P, R)b = e(g, g)^{abc}$. Charlie compute the value $e(P, Q)c = e(g, g)^{abc}$.

The shared secret values are $K = e(g, g)^{abc}$.

Public Parameter Creation		
A trusted party chooses and publishes a group G with g his generatrice and a pairing e		
Private Computations		
Alice	Bob	Charlie
Choose a secret integer a. Compute $P = a.g$	Choose a secret integer b. Compute $Q = b.g$	Choose a secret integer c. Compute $R = c.g$
Public Exchange of Values		
Alice sends P to Bob and Charlie ⟶ P		
Bob sends Q to Alice and Charlie ⟶ Q		
Charlie sends R to Alice and Bob ⟶ R		
Further Private Computations		
Alice	Bob	Charlie
Compute $e(Q, R)^a$	Compute $e(P, R)^b$	Compute $e(P, Q)^c$
The shared secret value is $K = e(g, g)^{abc}$		

Three party Diffie-Hellman Key Exchange Protocol with Pairing Cryptography.

G. Four party Diffie-Hellman Key Exchange Protocol with Pairing Cryptography

Alice, Bob and Charlie want to communicate using pairing application so they follow this method: Alice, Bob, Charlie and Jermy want to communicate using pairing application so they follow this method:

1) Public parameter creation:

Alice, Bob, Charlie and Jermy publish a group G with g his generator and a pairing e.

2) Private parameter creation:

Alice choose a secret a1 and calculate $P_1 = a_1 g \in G$.

Bob choose a secret a2 and calculate $P_2 = a_2 g \in G$.

Charlie choose a secret a3 and calculate $P_3 = a_3 g \in G$.

Jermy choose a secret a4 and calculate $P_4 = a_4 g \in G$.

3) Public exchange of values:

Alice, Bob, Charlie and Jermy receive and send the points P_1, P_2, P_3, P_4 to each other in a public channel of communication.

4) Further private computation:

Alice computes the value

$K = e(P_2, P_3)_{a1P4} = e(g, g)_{a1a2a3a4g}$

$= e(g_2, g)_{a1a2a3a4}$

Bob compute the value

$K = e(P_1, P_3)_{a2P4} = e(g, g)_{a1a2a3a4g}$

$= e(g_2, g)_{a1a2a3a4}$

Charlie compute the value

$K = e(P_1, P_2)_{a3P4} = e(g, g)_{a1a2a3a4g}$

$= e(g_2; g)_{a1a2a3a4}$

Jermy compute the value

$K = e(P_1, P_2)^{a4P3} = e(g, g)^{a1a2a4a3g}$

$= e(g^2, g)^{a1a2a3a4}$

The shared secret values are $K = e(g^2, g)^{a1a2a3a4}$.

Public Parameter Creation			
A trusted party chooses and publishes a group G with g his generatrice and a pairing e			
Private Computations			
Alice	Bob	Charlie	Jermy
Choose a secret a_1. Compute $P_1 = a_1.g$	Choose a secret a_2. Compute $P_2 = a_2.g$	Choose a secret a_3. Compute $P_3 = a_3.g$	Choose a secret a_4. Compute $P_4 = a_4.g$
Public Exchange of Values			
Alice sends P_1 to Bob, Charlie and Jermy $\longrightarrow P_1$			
Bob sends P_2 to Alice, Charlie and Jermy $\longrightarrow P_2$			
Charlie sends P_3 to Alice, Bob and Jermy $\longrightarrow P_3$			
Jermy sends P_4 to Alice, Bob and Charlie $\longrightarrow P_4$			
Further Private Computations			
Alice	Bob	Charlie	Jermy
Compute $e(P_2, P_3)^{a_1 P_4}$	Compute $e(P_1, P_3)^{a_2 P_4}$	Compute $e(P_1, P_2)^{a_3 P_4}$	Compute $e(P_1, P_2)^{a_4 P_3}$
The shared secret value is $K = e(g^2, g)^{a_1 a_2 a_3 a_4}$			

Four party Diffie-Hellman Key Exchange Protocol with Pairing Cryptography

H. Multi-party Diffie-Hellman Key Exchange Protocol with Pairing Cryptography

We extended our last method to n users, 1) Public parameter creation:

Users decide to publish a group G with g his generator and a pairing e.

2) Private parameter creation:

Each user choose a secret ai and calculate $P_i = g^{ai}$.

3) Public exchange of values:

Every user sends it public key P_i to all other users in a public channel of communication. 4) Private parameter creation:

Each user calculate $e(g^{n-2}, g)^{a1\ldots an}$

The common key is $K = e(g^{n-2}, g)^{a1\ldots an}$. Calculate the common key: $K = e(P_2, P_3)^{a1P4\ldots Pn} = e(g, g)^{a1\ldots an} \wedge g_{n-3} = e(g_{n-2}, g)^{a1\ldots an}$

Public Parameter Creation			
A trusted party chooses and publishes a group G with g his generatrice and a pairing e			
Private Computations			
User 1	User 2	...	User n
Choose a secret a_1. Compute $P_1 = a_1.g$	Choose a secret a_2. Compute $P_2 = a_2.g$	...	Choose a secret a_4. Compute $P_n = a_n.g$
Public Exchange of Values			
User 1 sends P_1 to all other users $\longrightarrow P_1$			
User 2 sends P_2 to all other users $\longrightarrow P_2$			
.			
.			
.			
User n sends P_n to all other users $\longrightarrow P_n$			
Further Private Computations			
User 1	User 2	...	User n
Compute $e(P_2, P_3)^{a_1 \Pi_{i=4}^{n} P_i}$	Compute $e(P_1, P_3)^{a_2 \Pi_{i=4}^{n} P_i}$	...	Compute $e(P_1, P_2)^{a_n \Pi_{i=3}^{n-1} P_i}$
The shared secret value is $K = e(g^{n-2}, g)^{\Pi_{i=1}^{n} a_i}$			

Multi party Diffie-Hellman Key Exchange Protocol with Pairing Cryptography

Efficiency: With this diagram, we can exchange the keys of all users just in one step.

demonstrates a significant efficiency enhancement compared to the classical Diffie-Hellman Key Exchange Protocol by employing pairing. Specifically, we illustrate that our method enables the exchange of keys for all users in a single step, as opposed to the traditional DiffieHellman Key Exchange Protocol that require n steps. This streamlined process not only simplifies key exchange procedures but also offers enhanced scalability and performance in multi-user environments. Overall, our findings underscore the potential of combining Diffie-Hellman with Pairing Cryptography to address the evolving needs of secure communication across diverse user groups.

V. Conclusion

In conclusion, this paper presents a novel implementation of the Diffie-Hellman Key Exchange Protocol integrated with Pairing Cryptography, extending its applicability to multiuser scenarios. Our approach

References

[1] Victor S. Miller. Use of elliptic curves in cryptography. Crypto 1985, LNCS 218, pp. 417-426, 1985.

[2] Neal Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, Vol. 48, No. 177, pp. 203-209, 1987.

[3] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and publickey cryptosystems. Commun. ACM, Vol. 21, No. 2, pp. 120-126, 1978.

[4] Whelan, C., Scott, M.: The Importance of the Final Exponentiation in Pairings When Considering Fault Attacks. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 225-246. Springer, Heidelberg (2007).

[5] Nadia El Mrabet, Nicolas Guillermin, and Sorina Ionica. A study of pairing computation for curves with embedding degree 15. DBLP volume 2009.

[6] Nadia El Mrabet and Marc Joye. GUIDE TO PAIRINGBASED CRYPTOGRAPHY. Chapman and Hall/CRC CRYPTOGRAPHY AND NETWORK SECURITY, 2018.

[7] Emmanuel Fouotsa, Nadia El Mrabet and Aminatou Pecha. Optimal Ate Pairing on Elliptic Curves with Embedding Degree 9; 15 and 27. Journal of Groups, Complexity, Cryptology, Volume 12, issue 1 (April 17, 2020)

[8] Narcisse Bang Mbiang, Diego De Freitas Aranha, Emmanuel Fouotsa. Computing the optimal ate pairing over elliptic curves with embedding degrees 54 and 48 at the 256-bit security level. Int. J. Applied Cryptography,Vol. 4, No. 1, 2020.

[9] Md. Al-Amin Khandaker, Taehwan Park, Yasuyuki Nogami, and Howon Kim, Member, KIICE. A Comparative Study of Twist Property in KSS Curves of Embedding Degree 16 and 18 from the Implementation Perspective. J. lnf. Commun. Converg. Eng. 15(2): 97-103, Jun. 2017.

[10] Md. Al-Amin Khandaker, Yasuyuki NOGAMI. Isomorphic Mappingcfor Ate-based Pairing over KSS Curve of Embedding Degree 18.c10.1109/CANDAR.2016.0113 November 2016.

[11] Rahat Afreen, S.C. Mehrotra. A REVIEW ON ELLIPTIC CURVE CRYPTOGRAPHY FOR EMBEDDED SYSTEMS. International Journal of Computer Science & Information Technology (IJCSIT), Vol 3.

[12] Md. Al-Amin Khandaker, Yasuyuki NOGAMI. A Consideration of Towering Scheme for Efficient Arithmetic Operation over Extension Field of Degree 18. 19th International Conference on Computer and Information Technology, December 18-20, 2016, North South University, Dhaka, Bangladesh.

[13] Nadia El Mrabet, Aurore Guillevic, and Sorina Ionica. Efficient Multiplication in Finite Field Extensions of Degree 5. DBLP 10.1007/978-3- 642-21969-6-12 June 2011.

[14] Michael Scott, Aurore Guillevic. A New Family of PairingFriendly elliptic curves. May 21, 2018.

[15] Michael Scott, On the Efficient Implementation of PairingBased Protocols, in cryptography and coding, pp. 296-308, Springer, 2011.

[16] Joseph H. Silverman, The Arithmetic of Elliptic Curves, Second Edition, 2000.

[17] Augusto Jun Devegili1, Colm Eigeartaigh, Michael Scott, and Ricardo Dahab, Multiplication and Squaring on PairingFriendly Fields, 2006.

[18] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Compression Point in Field of Characteristic 3. Springer, I4CS 2022, CCIS 1747, pp. 104111, 2022 https://doi.org/10.1007/978-3-03123201-5 7.

[19] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Tower Building Technique on Elliptic Curve with Embedding Degree 36. WSEAS TRANSACTIONS ON COMPUTERS. DOI: 10.37394/23205.2022.21.39.

[20] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Tower Building Technique on Elliptic Curve with Embedding Degree 72. WSEAS Transactions on Computer Research 10:126-138 DOI: 10.37394/232018.2022.10.17

[21] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. TOWER BUILDING TECHNIQUEON ELLIPTIC CURVEWITH EMBEDDING DEGREE 18. Tatra mountains mathematical publications, DOI: 10.2478/tmmp-2023-0008Tatra Mt. Math. Publ. 83 (2023), 103118.

[22] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Pairing based cryptography New random point exchange key protocol. Conference: 2022 7th International Conference on Mathematics and Computers in Sciences and Industry (MCSI), DOI: 10.1109/MCSI55933.2022.00017.