Salud, Ciencia y Tecnología - Serie de Conferencias. 2023; 2:526
doi: 10.56294/sctconf2023526

SERIE DE CONFERENCIAS

Categoría: Congreso Científico de la Fundación Salud, Ciencia y Tecnología 2023

ORIGINAL

Unlocking Security: Pioneering a Novel Elliptic Curve-Based Hashing Scheme

Desbloquear la seguridad: Un novedoso sistema de cifrado basado en una curva elíptica

Mbarek LAHDOUD[1], Ahmed ASIMI[1]

[1]Laboratoire des Syst`emes Informatiques & Vision (LabSiv). S´ecurit´e, Cryptologie, Contrˆole d'acc`es et Mod´elisation (SCCAM). Department of Mathematics-Faculty of Sciences, University Ibn Zohr- Agadir.

Citar como: LAHDOUD M, ASIMI A. Unlocking Security: Pioneering a Novel Elliptic Curve-Based Hashing Scheme. Salud, Ciencia y Tecnología - Serie de Conferencias 2023; 2:526. https://doi.org/10.56294/sctconf2023526

Recibido: 15-06-2023 **Revisado:** 17-08-2023 **Aceptado:** 19-10-2023 **Publicado:** 20-10-2023

ABSTRACT

Low-power networks and devices are becoming increasingly prevalent globally. These networks facilitate the exchange of concise messages, such as measurements and instructions. However, ensuring security, particularly concerning message integrity and sender authentication, presents a challenge in constrained environments. This article introduces a major breakthrough in the field of cryptography through the development of an innovative hash function leveraging the torsion subgroup on an elliptic curve. By incorporating the unique properties of this group, our approach redefines data security standards. We demonstrate the heightened resilience of our hash function against current attacks while maintaining exceptional performance. This novel method represents a significant advancement in safeguarding sensitive information, paving the way for more robust cybersecurity and practical applications across various domains. Experimental results confirm the effectiveness and security of our approach, establishing new perspectives for the evolution of modern cryptography.

Keywords: Hash; Security; Elliptic Curve; Subgroup.

RESUMEN

Las redes y dispositivos de bajo consumo son cada vez más frecuentes en todo el mundo. Estas redes facilitan el intercambio de mensajes concisos, como mediciones e instrucciones. Sin embargo, garantizar la seguridad, sobre todo en lo que respecta a la integridad de los mensajes y la autenticación del remitente, supone un reto en entornos con limitaciones. Este artículo presenta un importante avance en el campo de la criptografía mediante el desarrollo de una innovadora función hash que aprovecha el subgrupo de torsión de una curva elíptica. Al incorporar las propiedades únicas de este grupo, nuestro enfoque redefine los estándares de seguridad de los datos. Demostramos la mayor resistencia de nuestra función hash frente a los ataques actuales, manteniendo al mismo tiempo un rendimiento excepcional. Este novedoso método representa un avance significativo en la salvaguarda de información sensible, allanando el camino para una ciberseguridad más robusta y aplicaciones prácticas en diversos dominios. Los resultados experimentales confirman la eficacia y seguridad de nuestro enfoque, estableciendo nuevas perspectivas para la evolución de la criptografía moderna.

Palabras clave: Hash; Seguridad; Curva Elíptica; Subgrupo.

INTRODUCTION

In the blockchain, at the IoT level and within IT processes, the Cryptographic hash function plays a fundamental role in security: as- ensure a link between blocks, build a Merkle tree at the transaction level of a block of a chain, sign a message, produce an HMAC, establish an address, draw up hash tables to facilitate access to information etc..., the function The hash therefore behaves like a "Swiss army knife".

H is a hash function of size d if for each message of any size, it associates a string of d bits.(1) This chain is called the imprint, the hash, the digest or the condensed.

It depends on the input file.

Soit:

$H : \{0,1\}^* \quad \rightarrow \{0,1\}^d$

$M \quad \rightarrow 7 \quad H(M)$

It is cryptographic if it is one-way. That is to say, given $y \in \{0,1\}^d$, it is "difficult", with the currently accessible computing power, to determine x such that $H(x) = y$.(1,2,3,4,5,6,7,8)

Furthermore, this application, nowadays, and according to the designer's objectives, satisfies the following main properties:(9)

- The calculation of the result is very fast;
- The antecedent of a given image is extremely difficult to calculate by current technologies [Pre-image]
- For a given x, it is difficult to determine x' such that $H(x') = H(x)$; [Second pre-image]
- It is almost impossible to determine two different messages whose digests coincide;[Collision]

Table 1. Minimum security

Attack	Security Limit
Pre-image	2d
Second pre-image	2d
Collision	$2^{d/2}$ (Birthday Paradox)

This function is used to verify the integrity of a saved or received file, the signature of the issuer of a file, authentication and information security. Our goal in this work is to propose an elliptic curve based hash function.

In section 2, we will overview the state of the art in the design and construction of conventional hash functions or those intended for restricted environments, in section 3 we propose a function hash where the digest is the point of an elliptic curve, our conclusion will be carried by the section 4.

Notations

IoT	:	Internet Of Things
RFID	:	Radio Frequency Identification
MAC	:	Message Authentication Code
a== b	:	Value of variable a is equal to the value of variable b
$A \leftarrow B$	:	Assign the value of variable B to variable A
HMAC	:	Keyed-Hash Message Authentication Code

ODHFQF	:	One-way Dynamic Hash Function based on Quadratic Fields
$\{0,1\}^*$	:	Set of finite chains formed by 0 and 1
$\|A\|$	:	Cardinal of the set A
$\underbrace{0^s}_{stimes}$	:	$\underbrace{0\|\|0\|\|...\|\|0}$: concatenation

State of the art

The hash function is characterized by the size of its fixed output, however, recently, we find in the literature dynamic hash functions whose output size varies in a finite interval of integers $[l_{min}, l_{max}] \cap$N, the example is given by ODHFQF hash function cited in.[3] In other words, for a given hash function, the output size is either fixed or adjustable. To each file or set of data, it associates a string of *d* bits which represents all of the input data.[2,4]

Historically, the hash function is composed of two elements:[1]

• A transformation (a compression or permutation) *f* where the sizes of the inputs and output are fixed; f: $\{0,1\}^n \times \{0,1\}^m$ 7→ $\{0,1\}^n$ where $m,n \in$ N. • An extension of the domain by an iterative process using message slicing and transformation to obtain the same size as the output of *f* for any input message

Compression function[1]

These functions are a fundamental component of iterated hashing schemes. They are built on the basis of block encryption. The obvious example is $h(h_{i-1},m_i) = h_i$, where we can take $h(h_{i-1},m_i) = E_{mi}(h_{i-1})$. But, using D_{mi}, a weakness appears, consisting of the determination of h_{i-1} given h_i. What facilitates a preimage and second preimage attack.

In the sense of correcting, solutions in the form $H_i = h(H_{i-1},m_i) = E_{x1}(x) \oplus x3$, where x_1,x_2,x_3 are linear combinations of H_{i-1} and m_i, have been put forward. But the most used are indicated in the figure 1.[10,11]

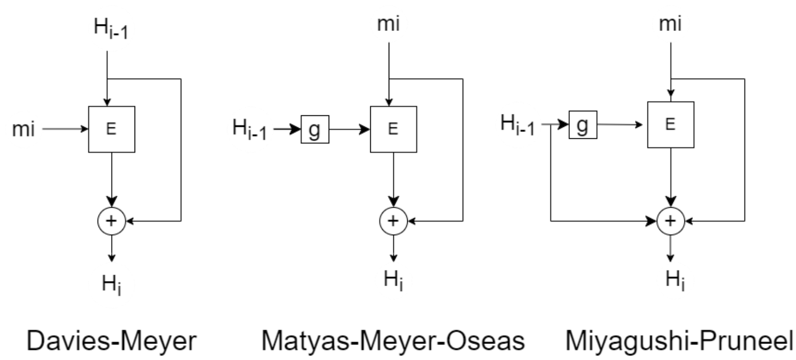

Figure 1. Compression Functions

The relationships governing them are respectively:
1. $Hi = Emi(Hi-1) \oplus Hi-1$; 2. $Hi = Eg(Hi-1)(mi) \oplus mi2$
3. $Hi = Eg(Hi-1) \oplus mi \oplus Hi-1$.

Various hash construction schemes

The different models for constructing hash functions are:

1. *Merkle-Damgaard.*[5,6] The process described can be summarized as follows:

The initial message is divided into k blocks of equal size, where the last block is padded with zeros to reach the required size (an operation called padding). We use a compression function f with two inputs and one output.

https://doi.org/10.56294/sctconf2023526

The domain extension is done by using an initial vector (IV) and compressing the blocks associated with the chaining values. These chaining values come either from the previous compression or from the initial vector (IV) when initializing the iterative process.

The final result is obtained by taking the transform of the last block

Vulnerabilities inherent to iterations in the Merkle-Damgaard scheme include:[8]

- Collision recycling: repetitive use of a collision in the compression function;
- Length extension attack: if we know $h(M)$ then we can calculate $h(M||S)$ where S is any string;
- Multi-collisions: problem, discovered by A. Joux in 2014, linked to the iterated character on the compression function.

2. *HAIFA*.[24][14] To solve the internal collisions of Merkle Damgaard(MD), HAIFA (HAsh Iterative FrAmework) adds two chains C_i, S to the block m_i of MD, the C_i counts the message bits processed up to rank i , the S is the salt (a fixed bit string), see figure 2. This construction and Wide-pipe[1,11] are similar in the sense that the latter generates internal chains of sizes larger than the final output.

3.

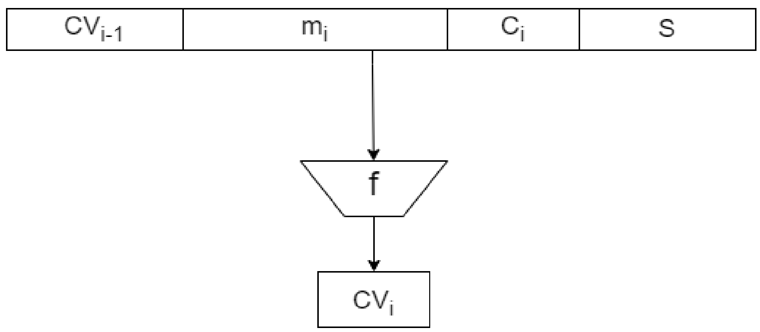

Figure 2. HAIFA scheme

4. *Sponge construction*.[4,11,12,13,14,15,16,17,18,29,20,21] The hash is the product of a process of iterations where the internal state S is the partition: Y of size c (capacity of the sponge set by the user) and X of size $r = b - c$ (sponge rate) corresponding to the number of bits absorbed per iteration; the process takes place in two phases as illustrated in figure 3 A state vector $S = (X||Y)$ evolves by iteration, The state is obtained after applying a permutation or transformation f.

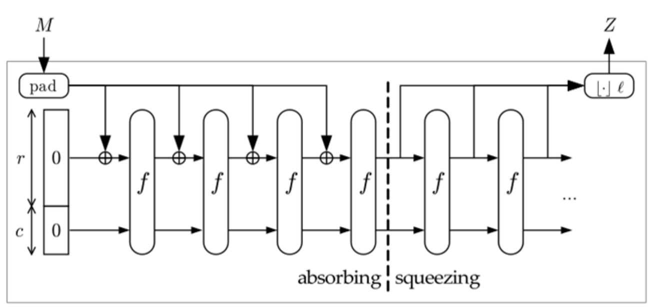

Figure 3. Sponge Diagram

- *Padding*. The message M is divided into N blocks of size r bits (the last block is possibly completed by zeros: padding) i.e. $M_0||M_1||M_2||M_3...||M_{N-1}$;
- *Absorption*. The initial vector $IV = 0$; the first block M_0 is xored (bit by bit) with the X of the IV (i.e. the first r elements of IV), the block M_i of the message is xored (bit by bit) with the X part of the state vector $S_i = f(M_i XORX)||Y$), so on until the last block of the message. The vector IV can take other values in this case to be fixed by the correspondents for more entropy;

- *Squeezing*. We concatenate the X after the end of the absorption up to a rank fixed by the user or the designer. (The spin can be made interleaved with the absorption: duplex construction).[22,23,24,25]

On the vector $(X||Y)$, we can establish $(c + r)!$ permutations f or $(c + r)^{c+r}$ transformations. Which can deter an attacker. Additionally, the r can be increased at the last transformation of the "Absorption" process to obtain an 'extended sponge' scheme.[15] In sum, the sponge model is a tool for constructing hash functions with output sizes chosen by the user or designer and will be suitable for resource-constrained environments. In this sense, it is necessary to design a permutation (transformation) f (component of the compression function) which is dependent on the message.

Applications

Conventional hash functions

The sponge scheme serves as the foundation for the expansion of the field in many hash functions that have emerged over the past decade. A notable example is Keccak, which became the SHA-3 standard, resulting from the competitive process initiated by NIST on October 2, 2012, including 14 candidates.[1] In this context, SHA-3 adopts the sponge model, while the Luffa, Fugue and CubeHash functions use derived algorithms.

Lightweight hash functions

An overview of hash function families suitable for resource-constrained environments was presented by.[17] Among them, Quark[13], Photon[15], Spongent[16] use the sponge scheme. Similarly, the low-cost Gluon[18,19] family shares the extension scheme with SHA-3 and the three previously mentioned families.

In the context of low-cost cryptography, a growing demand currently,[25] NIST announced on March 29, 2021, inviting ten finalists to complete their submissions by May 17, 2021. The fifth virtual workshop of Lightweight Cryptography hosted by NIST will be held May 9-11, 2022, providing an opportunity to discuss various aspects of the finalists and gather feedback toward the standardization of lightweight cryptographic primitives.

It is important to note that low-cost cryptography will also influence hash functions through their compression functions, as reported in section 2.

Fundamentals for our proposal

We recall below

Field

Definition 1. A field is a triple (K,+,.) where E is a non-empty set, + and . two internal laws which satisfy: - (K,+) and (K*,.) are two commutative groups of neutral elements 0 and 1. (K*=K -0) - . is distributive with respect to +.

Definition 2. The cardinal of a field (K,+,x) is the total number of elements forming E, denoted $|K|$. When $n = |K| < \infty$, the field is finite and denoted F_n.

Definition 3. The characteristic of a field (K,+,.), denoted car(K) is the smallest integer c such that 1+1+...+1 (c times) is equal to 0.

Theorem 1. Every finite field has a cardinality of the form p^n, where p and n are natural integers, p prime and $n > 0$.

Theorem 2. All finite cardinal fields p^n are isomorphic.

Theorem 3. for all $(p,n) \in N^* x N^*$ with p prime, there exists a cardinal field p^n

Example of finite field and cardinal equal to p^n.

$F_p[X]/P(X)$ the set of remainders of the Euclidean division of polynomials with coefficients in F_p by the polynomial $P(X)$ (p is a prime integer and $P(X)$ is irreducible of degree = n). This set has the laws + and . is a finite field with cardinality equal to p^n .

Elliptic curve

Definition 4. An elliptic curve E on a field K is the set of points $P = (x,y) \in E$ such that:

https://doi.org/10.56294/sctconf2023526

$$y^2 = x^3 + ax + b \quad \text{si } car(K) \neq 2,3. \text{ si } car(K) = 2 \text{ si}$$
$$car(K) = 3 \tag{1}$$
$$y^2 + y = x^3 + ax + b$$
$$y^2 = x^3 + ax^2 + bx + c$$

or a, b, $c \in K$ and check $4a^3 + 27b^2 \neq 0$.

(N. Koblitz, A Course in Number Theory and Cryptography, Springer 1987)

To have an estimate of the number of points on an elliptic curve on a finite field K_q, we use Hasse's theorem or Schoof's algorithm (see Book: Elliptic Curves

Number Theory and Cryptography Second Edition)

√

Theorem 4 (Hasse). : If $|E|$ is the order of E then $|q + 1 - |E|| < 2 \quad q$

The addition on E: $R = P + Q$ on an elliptic curve on R, is defined by the secant and the tangent. Expressed in Cartesian coordinates, we will have:

- $P \neq Q$ and $x_P \neq x_Q$:

$\lambda = x$______$yQQ--xyPP$

$$(2 - xP - xQ \; x_R = \lambda \; y_R = \lambda(x_P - x_R) - y_P \tag{2}$$

$$\lambda = \frac{3 \;\; _P+}{2y_P} \quad \bullet \quad P = Q \text{ and } \; x^2 \;\; a \quad y_P \neq 0:$$

$$(2 - 2xP \; x_R = \lambda \; y_R = \lambda(x_P - x_R) - y_P \tag{3}$$

- ($P \neq Q$ and $x_P = x_Q$) or ($P = Q$ and $y_P \neq 0$) then R=O (neutral element of +)

These expressions valid on any field K confer to $E \cup \{O\}$ a structure of the commutative group.

Multiplication on E: It follows from addition such that for $n \in$ N and $P \in E$ we will have 0.P=O and n.P = P+P+...+P (n times).

Definition 5. The order of a point $P \in E$ is the smallest integer n such that $n.P = O$.

Definition 6. For $n \in$ N, The n-torsion $E[n]$ of the commutative group ($E \cup \{O\}$,+) is the set points that have an order equal to n.

Theorem 5. The n-torsion of E is a commutative subgroup.

Our contribution

The choice of parameters, such as the coefficients a and b of the elliptic curve, as well as the size of the n-twist, is crucial to guarantee security. Research focuses on determining secure settings.

Let the set of messages be represented by finite streams of bits, E an elliptic curve on a field K_q, P a point of $E[n]$ and l a positive integer. We define our hash function as follows:

$$H : \{0,1\}^* \rightarrow E \cup \{O\}$$
$$m \mapsto H(m) = \Sigma^k_{i=0} n_i.P$$

where n_i is expressed by the following recurrence:

$$n_i = \begin{cases} m_0 & \text{if } i = 0 \\ m_i \oplus n_{i-1} & \text{si } i \geq 1 \end{cases} \tag{4}$$

mi is the slice i of size l of the message $m = m_0||m_1||...||m_k$. We complete with "1" if the last slice has a length less than l.

The expression $\Sigma^k_{i=0} n_i.P$ is well defined whatever the message m. it corresponds to a point on the elliptic curve ($ni.P \in E$ and ($E \cup \{O\}$,+) is a group).

https://doi.org/10.56294/sctconf2023526

$n_i = a_0 + 2.a_1 + ... + 2^{i-1}.a_{i-1}$ where $a_k \in \{0,1\}$, the calculation will therefore be done by the method " double and add".

Properties

Reciprocity. The structure of the *H* function makes it very difficult to determine a message given a point on the elliptic curve.

Collisions. To reduce collisions at the slice level, we choose *P* with an order greater than *l*.

Avalanche. equation (1) allows the propagation of the change of a bit to impact the position of the end point. Let $m = a_0a_1...0...m_k$ (0 placed at rank *i*) $m' = a_0a_1...1...m_k$ (1 placed at rank *i*) then the Hamming distance δ(m,m') = 1

Security Evidence

The sum of the multiples of a point *P* and the XOR (bit by bit) with accumulation of the slices to obtain the scalar makes it very difficult to find a message from a given point *R*. (Discrete Logarithm Problem and stream encryption)

Complexity

The "Double and Add" Algorithm is in the form

Require: *n,P* **Ensure:** *R* $R \leftarrow O$ **repeat**
if *n* mod 2 == 1 **then**
$R \leftarrow R + P$
end if
$P \leftarrow 2P$ $n \leftarrow n/2$
until n==0

There would be, in the worst case, $2(log_2 n-1)$ of doublings and additions. or a complexity of $O(log_2 n)$ don't forget the linear complexity of formula (4). without forgetting other methods such as Montgmory Ladder (Wikipedia and N.M´eloui, Arithmetic for Cryptography bases on Elliptic Curves, Thesis University Montpellier II, 2007)

CONCLUSION

The essential role of hashing in the processes of confidence in data is expressed through the control of their integrity and the authentication of the interlocutors. In the current era, the Internet connects an increasing number of machines and objects, falling into two categories: heavy machines such as computers, PCs, MACs, and machines or objects with limited resources such as IoT, sensors, smart cards, RFID, etc. This diversification sparks our interest in paying particular attention to hash functions, with an emphasis on their efficiency and economics.

Our hashing solution, based on elliptic curves and able to work with a base point on the elliptic curve, aims to assign each message a point on this curve. gives a large family of hash functions dependent on the chosen elliptic curves and allowing adjustments in terms of output size and security level subordinate to customer needs. SMART will benefit from this approach in the Blockchain ecosystem, with extensive applications in various sectors such as agriculture, health, education, logistics, home automation and military.

REFERENCES

1. Boura, Christina" Analyse de fonctions de hachage cryptographiques", Th`ese de Doctorat-Universit´e Pierre et Marie Curie-Paris VI, 2012

https://doi.org/10.56294/sctconf2023526

2. Idrissi Ahmed, ”Nouvelles conceptions fond´ees sur la cryptologie et le code de Goppa”, Th`ese de Doctorat -Universit´e Ibn Zohr Agadir-Maroc-2014

3. Asimi Younes,” Lightweight and Robust cryptographic applications for a Secure Wireless Network Protocol”,Th`ese d’habilitation universitaire - Universit´e Ibn Zohr Agadir-Maroc-2022

4. Bertoni, Guido and Daemen, Joan and Peeters, Micha¨el and Van Assche, Gilles ”Sponge functions”, ECRYPT hash workshop, vol=2007, no=9, 2007

5. Ralph C. Merkle ”One Way Hash Functions and DES”. Dans Gilles Brassard, ´editeur : CRYPTO’89, volume 435 de Lecture Notes in Computer Science, pages 428–446. Springer, 1990

6. Ivan Damgard ”A Design Principle for Hash Functions”. Dans Gilles Brassard,´editeur CRYPTO’89, volume 435 de Lecture Notes in Computer Science, pages 416–427. Springer, 1990

7. Backes, Michael and Barthe, Gilles and Berg, Matthias and Gr´egoire, Benjamin and Kunz, C´esar and Skoruppa, Malte and B´eguelin, Santiago Zanella ”Verified security of merkle-damg˚ard”, IEEE 25th Computer Security Foundations Symposium, pp=354–368, 2012

8. Denton, B and Adhami, R ”Modern hash function construction”, Proceedings of the International Conference on Security and Management (SAM), pp=1, 2011

9. Salem, Israa Ezzat and Salman, Adil M and Mijwil, Maad M ”A Survey: Cryptographic Hash Functions for Digital Stamping”, Journal of Southwest Jiaotong University,vol:54, no:6, 2019

10. Kale, AM and Dhamdhere, Shrikant ”Survey paper on different type of hashing algorithm”, International Journal of Advance Scientific Research Algorithm, vol: 3, no: 2, 2018

11. Thomas Fuhr ”Conception, preuves et analyse de fonctions de hachage cryptographiques”, Th`ese de Doctorat- Ecole T´el´ecom Paris, 2011

12. Ga¨etan Leurent ”Construction et Analyse de Fonctions de Hachage”, Th`ese de Doctorat-Universit´e Paris Diderot, 2010

13. Aumasson, Jean-Philippe, Luca Henzen, Willi Meier, and Mar´ıa NayaPlasencia. ”Quark: A lightweight hash”. Journal of cryptology 26, no. 2, pp 313–339, 2012

14. Biham, Eli and Dunkelman, Orr ”A Framework for Iterative Hash Functions— HAIFA”, Computer Science Department, Technion, 2007

15. Guo, Jian and Peyrin, Thomas and Poschmann, Axel, ”The PHOTON family of lightweight hash functions”, Annual Cryptology Conference, pp 222–239, Springer, 2011

16. Bogdanov, Andrey and Kneˇzevi´c, Miroslav and Leander, Gregor and Toz, Deniz and Varıcı, Kerem and Verbauwhede, Ingrid” SPONGENT: A lightweight hash function”, International workshop on cryptographic hardware and embedded systems,Springer, pp 312–325, 2011

17. Kasper Damg˚ard, Tore Kasper Frederiksen ”Whitepaper LIGHTWEIGHT CRYPTOGRAPHY”, Alexandra Institute, 2021

18. Gupta, Deena Nath and Kumar, Rajendra ”Lightweight Cryptography: an IoT Perspective”, Int. J. Innov. Technol. Explor. Eng., volume=8, number=8, pp 700–706, 2019

19. Hammad, B Tareq and Jamil, Norziana and Rusli, Mohd Ezanee and Reza, M, ”A survey of lightweight cryptographic hash function”, journal=Inter. J. Sci. Eng. Res, volume=8, pp 806–814, 2017

20. Meuser, Tobias and Schmidt, Larissa and Wiesmaier, Alex, ”Comparing Lightweight Hash Functions–PHOTON & Quark”, 2015 web

21. Online https://simple.wikipedia.org/wiki/Cryptographic hash function

22. Online https://keccak.team/sponge duplex.html

23. Online https://fr.wikipedia.org/wiki/G

24. Online https://en.wikipedia.org/wiki/HAIFAconstruction

25. Online https://csrc.nist.gov/projects/lightweight-cryptography/round-2-candidates

FINANCIACIÓN

Los autores no recibieron financiación para el desarrollo de la presente investigación.

CONFLICTO DE INTERESES

Los autores declaran que no existe conflicto de intereses.

CONTRIBUCIÓN DE AUTORÍA

Conceptualización: Mbarek LAHDOUD, Ahmed ASIMI.
Curación de datos: Mbarek LAHDOUD, Ahmed ASIMI.
Análisis formal Mbarek LAHDOUD, Ahmed ASIMI.
Adquisición de fondos: Mbarek LAHDOUD, Ahmed ASIMI.
Investigación: Mbarek LAHDOUD, Ahmed ASIMI.
Metodología: Mbarek LAHDOUD, Ahmed ASIMI.
Administración del proyecto: Mbarek LAHDOUD, Ahmed ASIMI.
Recursos: Mbarek LAHDOUD, Ahmed ASIMI.
Software: Mbarek LAHDOUD, Ahmed ASIMI.
Supervisión: Mbarek LAHDOUD, Ahmed ASIMI.
Validación: Mbarek LAHDOUD, Ahmed ASIMI.
Visualización: Mbarek LAHDOUD, Ahmed ASIMI.
Redacción – borrador original: Mbarek LAHDOUD, Ahmed ASIMI.
Redacción – revisión y edición: Mbarek LAHDOUD, Ahmed ASIMI.