I.J. Wireless and Microwave Technologies, **2021, 5, 1-8**
Published Online October 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2021.05.01

An improved IOT Authentication Process based on Distributed OTP and Blake2

Hind EL Makhtoum
Ibn Tofail University/Engineering sciences laboratory, Kenitra, Morocco
Email: hind.elmakhtoum@uit.ac.ma

Youssef Bentaleb
Ibn Tofail University/Engineering sciences laboratory, Kenitra, Morocco
Email: youssef.bentaleb@uit.ac.ma

Received: 16 July 2021; Accepted: 16 August 2021; Published: 08 October 2021

Abstract: The Internet of Things operates in vital areas and involves people, objects, and networks. Indeed, it ensures the interconnection and exchange of critical information that does not tolerate flaws. A successful attack on an object in one of these fields will cause fatal damage. Therefore, securing IOTs requires particular attention to the used protocols. It is essential to choose adapted protocols to the requirements of the objects and provide maximum security. Authentication is the first entry to IOTs that must be optimal and secure, without encumbering objects with limited capacities. To address this issue and to avoid concentration points, we present an improved authentication protocol based on the Blake2 hash function and the distribution of calculating OTP function. We will then evaluate, in terms of spatial and temporal complexity, the improvement brought to the IOT authentication process by the proposed process. We will prove that the use of these technics ensure a secure authentication that consumes less time and less memory.

Index Terms: Authentication, IOT, BLAKE2, OTP, Time Complexity, Space Complexity.

1. Introduction

Despite the benefits, advantages, and evolution that the IoT brought to our lifestyle, security, trust, and privacy still impose several problems [16]. One of the significant challenges presented by the IoT is guaranteeing reliable and not cumbersome authentication.

The Internet of Things is deployed in critical fields such as smart grids, smart cities, e-wellbeing, wearable [17]… that involve many constrained devices .while the IoT is a greedy target, it requires a high level of security and data concentration points could be fatal for an IoT system.

Besides, due to the small size of the communication mechanisms deployed in IoT and its limits related to equipment performance, including memory and limited computational capabilities, the use of a heavy authentication protocol can lead to high latency.

The majority of existing IoT authentication processes is either focused on security or performance. OTP is a technology that has provided remarkable results in terms of both security and performance. However, an unsecured use of the OTP in its classical form can lead to security breaches. Also, the hash function deployed in the authentication process is critical as it is the principal cryptographic function used to ensure the privacy of the exchanged messages and complicate the communication's decryption. Our contribution is based on these two technologies. We will use the OTP differently in association with the blake2 hash function in order to have a secure, light, and faster authentication.

In this work, we propose an authentication based on an association phase where the password function calculation is distributed between the two entities. Also, we deployed the Blake2 hash function for exchanges because it has good performance. The improvement of this process consists of sharing the Fotp password function calculation so that no one will have the full computation expression. In this way, we avoid the concentration of information on a ONE POINT OF FAILURE, and we also let the Gateway, which has a high computation and storage capacity compared to IoT objects, carry out the heavy calculations and storage.

The document is organized as follows; section 2 expresses the authentication process that is subject to improvement, then in sections 3 and 4, we give an overview of the hash functions and the HOTP algorithm. Then, we present our contribution that consists of the distribution of the function. Before concluding, we calculate its time and space complexity.

2. Literature Review

Given its criticality, various studies have addressed authentication based on different approaches.

In [5], the author proposes an authentication mechanism for IEEE 802.1x technologies based on the expandable authentication protocol (EAP). Authentication begins with an exchange of identities. An authentication server then verifies the entities' authenticity, using algorithms and mechanisms like MD5 or TLS protocol. This solution is flexible and interoperable because it uses conventional mechanisms but requires many exchanged messages that generate high execution time and power consumption.

The work in [10] intends to secure IoT devices. They propose an authentication mechanism based on the IPsec principle using a key shared between constrained (Cd) and non-forced (Ud) devices. Authentication begins with an agreement between entities on security policy. Then they exchange the keying equipment, authenticate each other and eventually create a secure channel. The mechanism is robust, but it has some weaknesses. Both Gateway and Device generate and send keys without authentication. So, a malicious object can launch DOS attacks.

The author [11] proposed an authentication for OCARI Networks. It focuses on the association of a device with the Gateway that manages the authentication operation. Indeed, the proposed protocol is based on the calculation of the OTP function by each participant. As inputs, the function has a Ki key derived from the Gateway's Kd key and a challenge calculated by the Gateway. Authentication is successful if the values resulting from the OTP calculation are identical. This solution is optimal, but it presents vulnerability: A successful attack on the Gateway will lead to total control over the systems. An attacker can impersonate the identity of a device and infiltrate the network.

Each of the above proposals has some limitations. In this work, we are interested in the [11] process that presents the vulnerability of ONE POINT OF FAILURE. Indeed, all authentication information is concentrated on the Gateway. If it is attacked, the attacker will have total control of the network.

3. Authentication process for IOT

3.1 IOT security features

Authentication in the IoT is challenging because it must take into consideration the special features of these networks. Indeed, IoT networks are subject to various types of attacks since they involve different entities, namely devices, humans, service providers….. All these components are connected and must authenticate each other. In addition, lightweight solutions are required to address security challenges in the IoT field adequately. Devices of the IoT are limited in terms of computational, power, and memory capabilities. Due to the nature of this network, it is complicated to adapt existing infrastructures and protocols with the new features. These restrictions must be considered while designing IoT solutions and security protocols [18, 19]

3.2 IOT authentication Process

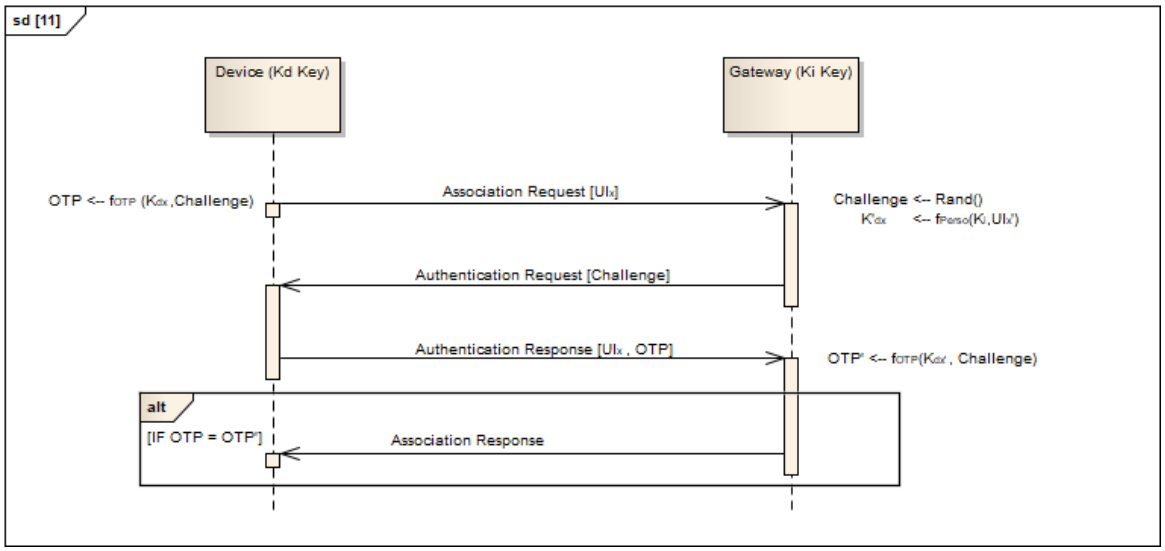

Fig.1. Version 1[11]

The first authentication version of [11] is based on four exchanges of association and authentication messages. The protocol switches from a traditional authentication based on exchanging messages between the Device and the Gateway (ID and password) to a new protocol that includes another message based on a two-entry Fotp function. Authentication is requested by the Device that sends its ID (Unix) to the Gateway. The Gateway first calculates a challenge and sends it to the Device; then, it calculates the Device's key Kdx through the Fperso function. Both entities calculate the

challenge-based OTP and the Kdx key, configured in an out-of-band way on the Device. The Device sends the calculated OTP to the Gateway, which compares the value received and the calculated value and returns to the Device the association response. The mechanism is described in Fig.1. (CPAN is the Gateway).

4. IOT Hash Functions

Our improvement is mainly based on integrating hash functions to complicate exchanges between the Gateway and the Device.

A hash function produces outputs with a size much smaller than the size of the input message. This hash is helpful in several applications, such as storing passwords, verifying integrity, and authentication.

Hash functions work in a one-way manner. They convert an arbitrary size data into a set of data with a fixed length. This length varies from one algorithm to another. A hash function hides the original data so that a slight change in the output results in a radical change in the output [7]. Also, cryptography functions require heavy mathematical calculations, which may be resource-intensive. So, it is necessary to opt for lighter algorithms to improve equipment performance and safety with limited capabilities, such as IoT equipment.

The most popular lightweight hash functions include Snefru, Message Digest (MD5), the SHA family, HAVAL, KECCAK, and BLAKE [4]. Several comparative studies have been done under different conditions and environments to find the best among these algorithms.

A comparative study between SHA and MD5 has shown that the SHA algorithm family works faster than the MD5 [8].

The Secure Hash Algorithm (SHA) is a widely used hash algorithm. In embedded systems, SHA algorithms are used to transfer and validate data integrity securely. Although SHA-2 offers strong encryption, it is still vulnerable to attacks because it uses the hash generated in the last round to generate the next hash token [7]

Therefore, on November 2, 2007, the National Institute of Standards and Technology (NIST) opened a competition to develop a new cryptographic hash algorithm - SHA-3 with a modified hash for key generation and offering better performance. BLAKE and Keccak were among the finalists selected for the final round of the competition and received much more support than the others, thanks to the high level of security they presented. Finally, KECCAK was announced the winner and designated as SHA3 [14,12]

Keccac, the winner of the competition, uses the sponge-based hash design based on bit permutations. In addition to being quick in hardware implementations, it has shown resistance against multiple attacks, a high level of security, high throughput, and simplicity of design [4,13].

Blake is a function based on stream encryption to generate the hash digests [13]. Indeed, a 16-word constant is used with the message word, the salt value, and the initial vector to form a 4x4 matrix, each line is subjected to eight sets of permutations and combinations. [4,6]

Later, BLAKE was improvised into BLAKE2, based on the same BLAKE design for high efficiency and safety. It went under several changes to optimize the protocol for modern applications with simplicity and ease of use. BLAKE2 is competitive in speed with MD5; it is faster than Blake, even on long messages, thanks to the reduced number of permutations [13]. It exists in two variants: BLAKE2b is optimized for 64-bit platforms, and the BLAKE2s is optimized for 8 to 32-bit platforms. [13,7]

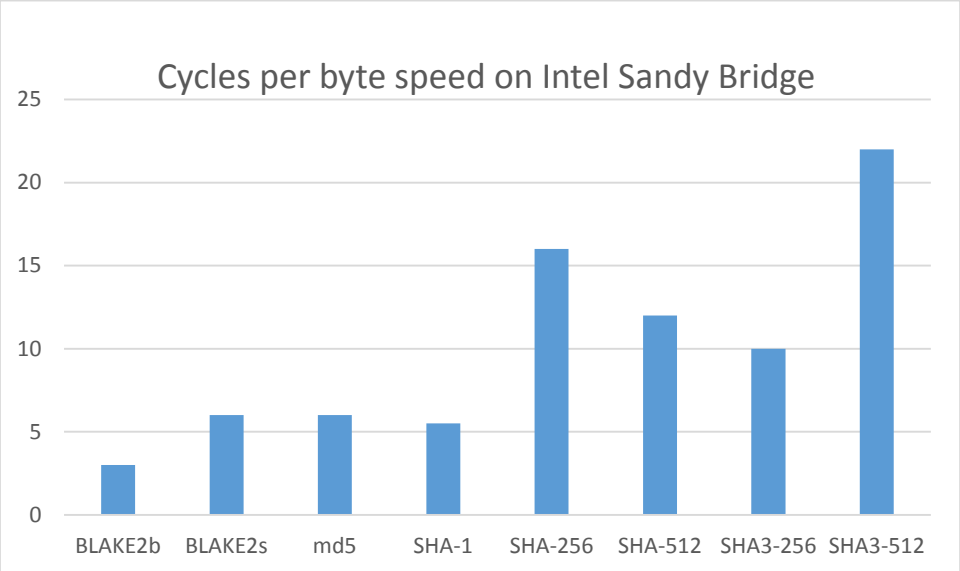

Fig.2. Hash Functions Speed [3]

Despite the many advantages of KECCAK, we will opt for Blake 2 because of the high processing speed. In order to compare our results with [11] results, we will especially choose the blake2s since it is more optimized on 32 processors and we will use its mathematical algorithm [20] as it is mentioned in RFC7693

Algorithm 1: Blake2 Algorithm [20]

Blake2 Algorithm

```
FUNCTION BLAKE2( d[0..dd-1], ll, kk, nn )
    |   h[0..7] := IV[0..7]          // Initialization Vector.
    |   // Parameter block p[0]
    |   h[0] := h[0] ^ 0x01010000 ^ (kk << 8) ^ nn
    |   // Process padded key and data blocks
    |   IF dd > 1 THEN
    |   |       FOR i = 0 TO dd - 2 DO
    |   |   |       h := F( h, d[i], (i + 1) * bb, FALSE )
    |   |       END FOR.
    |   END IF.
    |   // Final block.
    |   IF kk = 0 THEN
    |   |       h := F( h, d[dd - 1], ll, TRUE )
    |   ELSE
    |   |       h := F( h, d[dd - 1], ll + bb, TRUE )
    |   END IF.
    |   RETURN first "nn" bytes from little-endian word array h[].
    END FUNCTION.
```

5. HOTP

The OTP is based on a one-time password. It changes with the change of a value called counter that is valid for only one authentication, and it is incremented if it has succeeded [15]. HOTP is an authentication algorithm based on the association of HMAC with SH1 (default). It is computed symmetrically between the authenticated and the authenticator. It is based on comparing obtained results to decide whether the authentication has been successful or not [2].

The method is based on factors that must be shared previously between the two entities concerned. Namely, the C counter, which is a random number, a private secret key Ks, the length d of the HOTP (recommended to be between 6 and 8), and whether the authentication will keep the default hash function sha1 or that the entities will use a different function and this must also be communicated.[2]

The algorithm is described briefly in the fig.3. below:

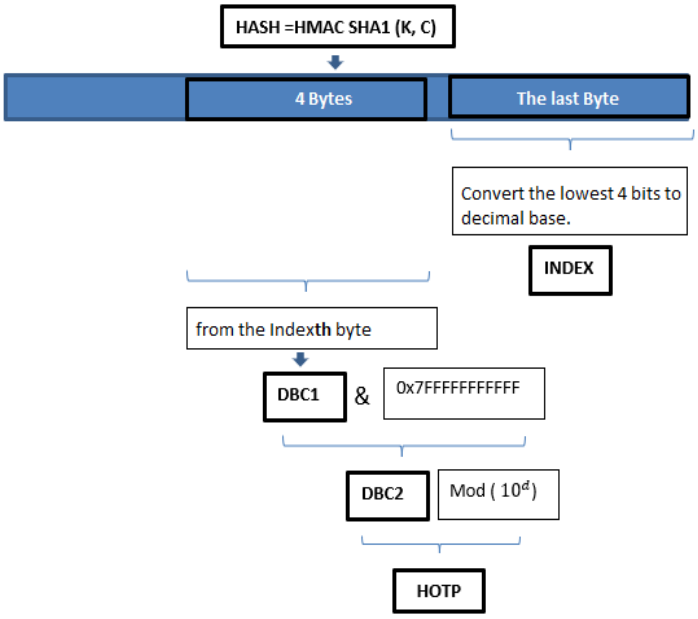

Fig.3. HOTP Function

First, a hash HMAC-SHA is established based on the values of C and Ks. The four least significant bits of the hash are converted into a decimal base to obtain the INDEX. Then we look for the Index byte of the hash and extract 4 bytes called BDC1. Since processors handle the arithmetic calculations of signed and unsigned numbers differently, we overcome this interoperability problem by subjecting the BDC1 value to a 0x7FFFFFFFFFFF to mask the byte of the sign and force the value to be an unsigned integer BDC2. Finally, to force the obtained value to the desired length d, a modulo 10^d is applied. Thus, we obtain the HOTP [1].

6. Contribution: Distribution of Fotp function

6.1 Principle

In this work, we propose an authentication protocol based on the [11] process with an improved computation of the OTP function to avoid the ONE POINT FAILURE on the Gateway.

In this paper, we are only interested in the authentication step of the Device to the Gateway. We assume that confidentiality and integrity are well assured. We will also work on the exact requirements of the thesis in terms of technical characteristics, particularly the 32-bit processor.

We will respect the number of exchanged messages to avoid the congestion of channels by the authentication messages.

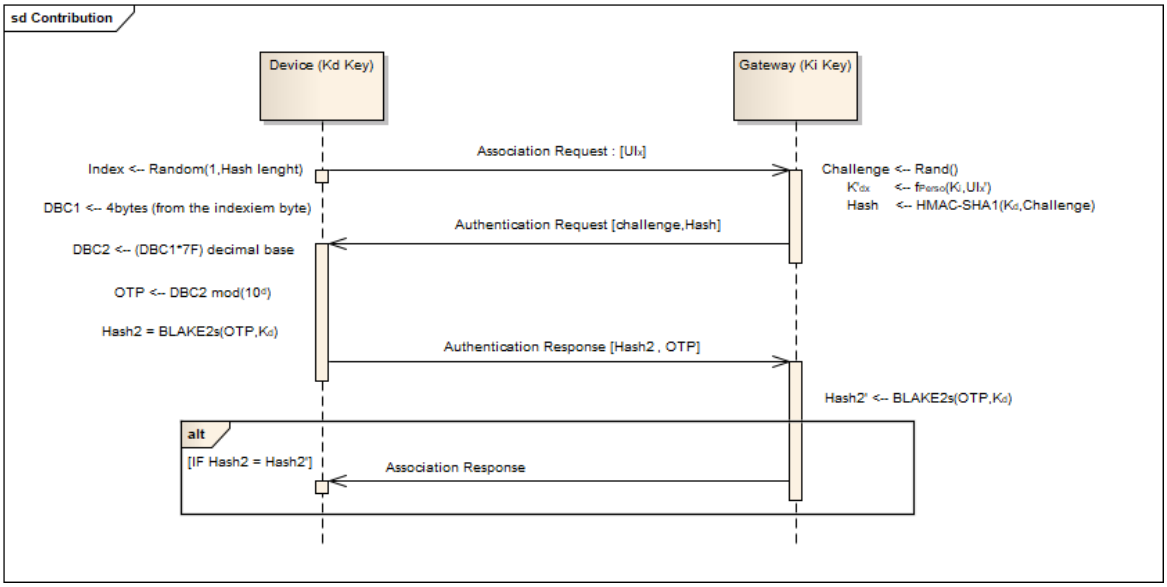

Fig.4. The proposed schema

First, the Device will send its ID (Ui) to the Gateway. The Gateway will extract the Kd key related to the Device and calculates a 32-bit Random challenge and the hash. The Gateway will be the only one to compute the HMAC-SHA256 hash to avoid heavy computation on the Device. After receiving the hash, the Device will compute an INDEX, the decimal value of the 4 bits following a random value between 1 and the Hash length. The Device will then continue the classical OTP calculation by converting the value of the 4 bytes according to the Index Byte, then applying an x7F mask relative to the value's sign and forcing it to the length d by the modulo d. The Device will then calculate a Hash2 based on the BLAKE2s function, which is more optimal in our context (according to Chapter 4), with the Kd (32 bits) and the calculated OTP as input arguments. The Device sends the value of OTP and Hash2 to the Gateway, performing the same calculation and comparing it to the received Hash2. If the value is the same as the one received, the authentication is successful; otherwise, it fails. After three attempts of the same Ui, the Gateway will block the Ui for a defined amount of time. Thus, none of the devices will have the expression or the complete authentication data.

7. Results and Discussion

7.1 Security Analysis

We have studied several security scenarios, as follows:

Scenario 1: Attacked Device: The attacker will not have the Kd key thanks to the key customization process [11]

Scenario 2: Attacked Gateway: The calculation required for authentication is based on calculations made separately by the Gateway and the Device, so full authentication requires both devices' computation expressions.

7.2 Time Complexity

The temporal complexity is divided between the two participants: Gateway (CTg) and Device (CTd). Since they have different computing, processing, and storage capacities, we will treat the complexity separately.

We assume that CT is the number of operations of the process [11]and CT' those of our proposal.

$$CT' = CTg(Random,\ Fperso, Hmac - sha256, Blake2s, Comparison) + CTd(Random, DBC1, DBC2, OTP, Blake2s) \quad (1)$$

We assume that the operations made by the two solutions are negligible; we obtain :

$$CT' = O\left[CTg(Blake2s) + CTd(Blake2s)\right] \quad (2)$$

We move on to the calculation of the number of operations of the solution [11]

$$CT = CTg(Random, Fperso, Hmac - sha256, Index, DBC1, DBC2, OTP, Comparison) + CTd(HMAC - 256, Index, DBC1, DBC2, OTP) \quad (3)$$

We proceed in the same way as for the previous complexity, and we get:

$$CT = O\left[CTg(Index, DBC1, DBC2, OTP) + CTd(HMAC - SHA256)\right] \quad (4)$$

Since the Gateway has advanced computational capabilities, what will infect the execution time is the time consumed in the calculation by the Device, which is limited in computational capacity. We are faced with a comparison between the speed of HMAC-SHA256 and the one of BLAKE2s, which is represented in Fig.2.

7.3 Space Complexity

For calculating spatial complexity, we will proceed in the same way as we calculated the temporal complexity. Among the variables that occupy storage space on our proposal, we conclude that the argument that makes the real difference is the digest produced by the blake2s function and the sum of arguments (DBC1, dbc2, and index). Comparing the sizes of these two will give us the optimal solution in spatial terms.

On the Device, our proposition consumes the sum of hash length (4bits) and the size of Blake2s digest (32bytes). Otherwise, the first proposition only consumes the size of the challenge produced by the authenticator and stored by the Device (32bytes). So, it implies a storage difference of extra 4bits consumed on the Device in our solution. For Gateway, the stored value that makes the difference is the size of the hash blake2s (32bytes) versus a 48bit storage divided between DBC1 (32bits), DBC2 (32bits), and index (4bits).

7.4 Experimental Results

Based on the Time complexity equations calculated in the last subsection, we implemented algorithms using the JavaScript language on a station with a 32bites processor as the Device and a station with 64bites as the Gateway. The algorithm intends to calculate the time of execution. Regarding the memory consumption, we will be based on the calculation of 1.4. Subsection.

Results are listed in the Tables below:

Time of execution:

Table 1.Time of execution (ms)

Solutions	Gateway	Device
Our proposition	1.052001953125	0.999755859375
The [11] proposition	2.071044921875	3.999755859375

Memory Consumption

Table 2. Memory consumption (bytes):

Solutions	Gateway	Device
Our proposition	MG+ 48	MD+ 36
The [11] proposition	MG+32	MD+32

*MG: The common memory consumed by the gateway in both solutions
*MD: The common memory consumed by the device in both solutions

7.5 Synthesis

The proposed solution responds well to the problem related to the security of the Gateway. According to the presented security scenarios, the solution is optimal for security because it avoids authentication information's centralization. Even in a case of infiltration to the Gateway, the attacker cannot have full control over the authentication and cannot also pass under another device's identity.

The solution proved good time results thanks to the Blake2s hash function, which is compatible with IoT devices and 32bit processors. It is fast and saves time for computation. On the other hand, our solution consumes 32 more bytes by the Gateway and 4bits more on the Device. This difference does not affect our solution's quality because the space consumption on the critical equipment (the Device) is light (4bits). It is high on the Gateway, but the Gateway has significant storage capacity and does not affect the solution's performance.

8. Conclusion and Perspective

In this document, the suggested process is reliable, secure, ensures high speed, and consumes reasonable computational processing. We first proposed an improvement of the association phase of the authentication process [11]. We opted for the distribution of Fotp computation. Indeed, Fotp is calculated in a distributed way between Device and Gateway; each makes a calculation based on a blind value already calculated by the other entities and vice versa. Thus none of the equipment has complete knowledge of the necessary information for authentication. If the Gateway gets attacked, no one will be able to take advantage of data's centralization. Besides, in order to maintain the performance of the connected devices, we have concentrated the heavy computation and storage to be done by the gateway, which is not limited in storage and computational capacities.

The storage issue could be subject to another storage enhancement proposal by transferring it to the cloud or a storage bay. The proposed solution could also be replaced by other effective methods such as asymmetric encryption (Diffie Hellman, elliptical curve) or Smart Cards.

References

[1] T. Patel and O. Kale, "A Secured Approach to Credit Card Fraud Detection Using Hidden Markov Model," vol. 3, no. 5, p. 8, 2014.

[2] B. R. Parikshith Nayaka S K Jeffin Boban, Aishwarya K, Arjun V., "Abnormal Pattern Analysis in Online Transaction," vol. 7, no. 8, 2019.

[3] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "BLAKE2: Simpler, Smaller, Fast as MD5," in Applied Cryptography and Network Security, vol. 7954, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 119–135. doi: 10.1007/978-3-642-38980-1_8.

[4] V. Rao and K. V. Prema, "Comparative Study of Lightweight Hashing Functions for Resource Constrained Devices of IoT," in 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), Bengaluru, India, Dec. 2019, pp. 1–5. doi: 10.1109/CSITSS47250.2019.9031038.

[5] Jyh-Cheng Chen and Yu-Ping Wang, "Extensible authentication protocol (EAP) and IEEE 802.1x: tutorial and empirical experience," IEEE Communications Magazine, vol. 43, no. 12, p. supl.26-supl.32, Dec. 2005, doi: 10.1109/MCOM.2005.1561920.

[6] V. Rao and P. K.V., "Light-weight hashing method for user authentication in Internet-of-Things," Ad Hoc Networks, vol. 89, pp. 97–106, Jun. 2019, doi: 10.1016/j.adhoc.2019.03.003.

[7] V. K. Sarker, T. N. Gia, H. Tenhunen, and T. Westerlund, "Lightweight Security Algorithms for Resource-constrained IoT-based Sensor Nodes," in ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, Jun. 2020, pp. 1–7. doi: 10.1109/ICC40277.2020.9149359.

[8] K. Suankaewmanee, D. T. Hoang, D. Niyato, S. Sawadsitang, P. Wang, and Z. Han, "Performance Analysis and Application of Mobile Blockchain," in 20f International Conference on Computing, Networking and Communications (ICNC), Maui, HI, Mar. 2018, pp. 642–646. doi: 10.1109/ICCNC.2018.8390265.

[9] P. Hagenlocher, "Performance of Message Authentication Codes for Secure Ethernet," 2018, doi: 10.2313/NET-2018-11-1_04.

[10] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi, "Secure Communication for Smart IoT Objects: Protocol Stacks, Use Cases and Practical Examples," p. 8.

[11] M. T. Hammi, "Sécurisation de l'Internet des objets," p. 164.

[12] E. Andreeva, B. Mennink, B. Preneel, and M. Škrobot, "Security Analysis and Comparison of the SHA-3 Finalists BLAKE, Grøstl, JH, Keccak, and Skein," in Progress in Cryptology - AFRICACRYPT 2012, vol. 7374, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 287–305. doi: 10.1007/978-3-642-31410-0_18.

[13] J.-P. Aumasson, W. Meier, R. C.-W. Phan, and L. Henzen, The Hash Function BLAKE. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. doi: 10.1007/978-3-662-44757-4.

[14] S. Chang et al., "Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition," p. 84.

[15] B. Hamdane, A. Serhrouchni, A. Montfaucon, and S. Guemara, "Using the HMAC-Based One-Time Password Algorithm for TLS Authentication," p. 8

[16] Jawahar Lal Nehru Technological University Kakinada, 533003, India, N. S. Chauhan, A. Saxena, and J. Murthy, "A Privacy-Aware Dynamic Authentication Scheme for IoT Enabled Business Services," IJCNIS, vol. 11, no. 6, pp. 29–37, Jun. 2019, doi: 10.5815/ijcnis.2019.06.04.

[17] Military Technical College/Electrical Engineering Department, Cairo,11571, Egypt, M. M. Samy, W. R. Anis., A. A. Abdel-Hafez, and H. D. Eldemerdash, "An Optimized Protocol of M2M Authentication for Internet of Things (IoT)," IJCNIS, vol. 13, no. 2, pp. 29–38, Apr. 2021, doi: 10.5815/ijcnis.2021.02.03.
[18] T. Yousuf, R. Mahmoud, F. Aloul, and I. Zualkernan, "Internet of Things (IoT) Security: Current Status, Challenges and Countermeasures," IJISR, vol. 5, no. 4, pp. 608–616, Dec. 2015, doi: 10.20533/ijisr.2042.4639.2015.0070.
[19] B S Abdur Rahman University, Vandalur, Chennai and 600048, India and S. Revathi, "Protocols for Secure Internet of Things," IJEME, vol. 7, no. 2, pp. 20–29, Mar. 2017, doi: 10.5815/ijeme.2017.02.03.
[20] Saarinen, M-J., & Aumasson, J-P. (2015). "The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC): IETF RFC 7693". (Request for Comments; No. 7693). Internet Engineering Task Force. https://doi.org/10.17487/RFC7693

Authors' Profiles

Hind EL MAKHTOUM, She got her engineering degree in Electronics and telecommunications from ENSEM, Casablanca, Morocco. Currently, she is a Ph.D. researcher of the Engineering sciences laboratory of Ibn Tofail university-Ensa, Kenitra, Morocco. His area of research includes Networks, security, and the Internet of things.

Pr. Youssef BENTALEB, He is a Professor researcher, entitled to direct research, holder of a Ph.D. in applied mathematics and computer science, a researcher in the fields related to modeling and digital simulation, signal and image processing, security, and analysis of massive data. He has a professional experience of ten years in the field of human resources management (administration and computer management of databases), President of the Moroccan Center for Polytechnic Research and Innovation (CMRPI), Director of the International Journal of Scientific Research and Innovation IJRSI-CMRPI, Editor-in-Chief of JCCE Journal, Member of the research laboratory EECOMAS, Guest Speaker.

How to cite this paper: Hind EL Makhtoum, Youssef Bentaleb, "An improved IOT Authentication Process based on Distributed OTP and Blake2", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.11, No.5, pp. 1-8, 2021.DOI: 10.5815/ijwmt.2021.05.01