IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 9, No. 2, June 2020, pp. 229~235
ISSN: 2252-8938, DOI: 10.11591/ijai.v9.i2.pp229-235

New concept for cryptographic construction design based on noniterative behavior

Abdallah Abouchouar, FouziaOmary, Khadija Achkoun
Department of Computer Science, Mohammed the V University, Morocco

Article Info

Article history:

Received Nov 13, 2019
Revised Aug 20, 2020
Accepted Apr 22, 2020

Keywords:

Authentication
Cryptanalysis
Cryptographic hash function
Domain extension
Security

ABSTRACT

Nowadays, cryptography especially hash functions require to move from classical paradigms to an original concept able to handle security issues and new hardware architecture challenges as in distributed systems. In fact, most of current hash functions apply the same design pattern that was proved vulnerable against security threats; hence the impact of a potential weakness can be costly. Thus, the solution begins with a deep analysis of divers attack strategies; this way can lead to finding a new approach that enables new innovative and reliable candidates as alternative hash functions. So to achieve this goal, in this article we introduce a new construction design that consists of a non-iterative behavior by combining a parallel block processing and a sequential xor addition process, in order to provide a secure design without changing the expected goal of a hash function, at the same time avoid the use of vulnerable structures.

Corresponding Author:

Abdallah Abouchouar,
Department of Computer Science,
Mohammed the V University,
Avenue des Nations Unies, Agdal 10000, Rabat, Morocco.
Email: abdollah.abouchouar@gmail.com

1. INTRODUCTION

In modern cryptography, we recognize hash functions by two mechanisms; the first is the internal transformation that process fixed length data; the second one is the construction design, also called domain extension algorithm, which iterate the internal transformations on the input message. The second mechanism comes as a solution to reduce compression process complexity. In practice hash function security depends on the way how the construction was designed, but now it presents serious security weakness. Most of collision attacks on hash functions take advantage of the model itself or through existing flaws in the internal compression function. The National Institute of Standards and Technology (NIST) deprecate their use in several applications and have decided that it is prudent to develop a new hash algorithm [1], even for those standardized as a secure algorithm, especially SHA2 family. In 2007 NIST announced a request for candidate algorithm nominations for a new cryptographic hash algorithm SHA-3 [2]; the aim was expecting a new candidate that offers features or properties exceeding, or improving upon SHA-2. The competition finished in 2012 and the winning algorithm was Keccak (based on wipe function design). Now it becomes the NIST's SHA-3 algorithm standard [3].

In fact, cryptanalysts fend their attacks on construction flaws, such as the length extension attack [4], the fixed-point attack [5], multi-collision attack [6], differential and linear attack and more other generic or specific attacks [7-9]. Despite the efforts made on hash functions paradigms to be more secure, there exists a potential threat while they inherit the properties of vulnerable design constructions as reported in a survey on authentication systems [10], IoT systems [11-12] and network architecture [13].

Thus, to build a secure construction design, a deep understanding of various attack strategies is required to prevent any flaws.

In the present paper, we introduce a new domain extension algorithm as an alternative to classic paradigms, based on a new approach to process compression transformations. Therefore, we let new perspectives for new hash functions conception which can be implemented for distributed architecture as in Cloud Computing, Big Data, etc.... The first section gives a preliminary about design construction and hash functions; in the second section we present a summary of different attack strategies based on either design constructions or hash functions flaws; in the third section we define the new extension domain algorithm based on a pseudo parallel behavior and sequential xor addition process, then we bring up a security analysis with a discution and comparison results.

2. PRELIMINARY

2.1. Cryptographic hash functions

Hash functions are fundamental tools in modern cryptography. Conventionally a hash function maps a large data set to a smaller one. But in this specific category, as they operate in several sensitive applications as authentication system, data integrity, key generation …etc, cryptographic hash functions should consider more required security properties [14]:

- Ease of computation ➔ for a kwon function H with input x, H(x) is easy to compute.
- One way function ➔ for each y = H(x) in range of H, it's computationally infeasible to find x in the domain of H.
- Preimage resistance ➔ for a given digest y of H, it is infeasible to find x with H(x) = y.
- 2nd preimage resistance ➔ for a given x, it is infeasible to find x' such that x'≠ x and H (x') = H(x).
- Collision resistance ➔ it is infeasible to find separate x and x' such that x'≠ x and H (x') = H(x).

In practice, all these properties are not satisfied, actually hash function definition confirms collisions existence, due to the birthday theory [14].

2.2. Domain extension

Domain extension, construction design or operation mode, are all the same concept that handles the compression transformation on a non-fixed length input to get as a result a fixed length output. It is difficult to implement a compression function that processes by itself a non-fixed length message. So, a domain extension algorithm intervenes here to reduce this complexity.

The Merkel-Damgard construction was the leading solution proposed independently by [15-16]. This construction provides an iterated behavior which influences the conception of many popular hash functions (MD/SHA family). Thus, the cost of finding a resistant hash function reduced to find a resistant internal compression function. As a consequence, an iterated construction with an internal collision resistant compression function can be extended to a collision resistant hash function [14].

Merkel-Damgârd construction [15-16] deals with arbitrary length message, divided into input blocks of fixed length. An initialization vector IV is used during the process, and iteratively the internal compression function takes as input the current message block and the output of the last iteration as an IV block as illustrated in Figure 1.

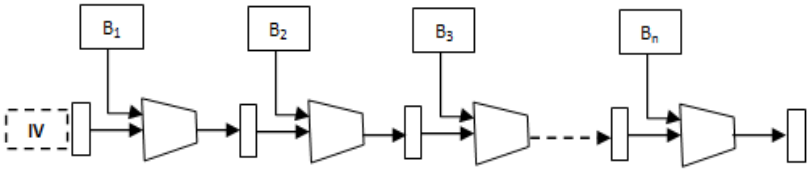

Figure1.The Merkel-Damgârd construction

Figure 1, here (B_i) are the divided input blocks and (IV) the initial vector. Because of its efficient design, most of known constructions are based on the Merkel-Damgârd [15-16], like [17], EMD [18], ROX [19] and HIFA [20]. Despite its relevant approach, the Merkel-Damgard algorithm is exposed to risk. The variants mentioned above came in order to enhance and fix the classic model flaws. But they still inherit its vulnerable structure.

3. ATTACK STRATEGIES ON DOMAIN CONSTRUCTION

Before starting a new design construction, it is important to begin by understanding diverse attack strategies. Advanced attacks concern both design constructions and specific hash functions, thus there are two categories: generic and specific ones. Below we give a non-exhaustive and brief introduction to well-known generic attacks.

3.1. Brute force attack

Theoretically, none of current hash functions can prevent it; this attack consists of an exhaustive search by testing all possible inputs. In this case the security level depends on the output size; the more it is bigger, more it is difficult to apply. [14]

3.2. Birthday attack

It is a generic attack algorithm which can be applied to any design or hash function, based on mathematical properties that assume if a given function (H) with size domain is (m) and co-domain size is (n) with (m > n), so at least there are two different elements (x ,y) with H(x) = H(y) [14]

3.3. Collision attack

For a given hash function H, we look for two messages (x and y) such $x \neq y$ and H(x) = H(y), this attack takes 2n/2 computations to find (x, y), based on birthday attack the probability to produce a collision is ½.

3.4. Preimage attack

For a given output (y) of a hash function H, we look for an input x such that H (x) = y. In theory, it takes 2n computation to find such (x); we call it a brute force preimage attack.

3.5. 2nd preimage attack

For a given message (x) and a hash function H, we look for (x') to have H (x') = H(x). In theory, it takes 2n computation to find such (x); we call it a brute force 2nd preimage attack.

3.6. Fixed point attack

Whit a compression function F with Davies-Meyer structure, we seek a pair (Hi-1,Xi) such that: F(Hi-1,Xi)=Hi-1. It can be arranged that the chaining variable (H) has a value for which a fixed point is known. This property allows 2nd preimage collision to be produced [21]. The following Figure 2 illustrates this structure. Figure 2, here (X_i) is the input block, (H_{i-1}) is the chaining variable for the current iterationand (F) is the compression function.

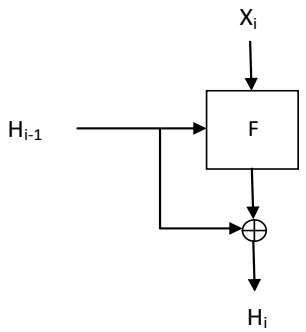

Figure 2. Davies-Meyer structure

3.7. Multi collision attack

Introduced by Joux [6], based on the iterative behavior of the hash functions construction and, in particular, exploits chaining variables and IV. It focuses on the internal compression function to find k distinct input blocks with the same collision.

3.8. Length extension attack

As the other attack strategies, it exploits the iterative hash function structure. This attack affects the MAC applications [4], by manipulating the internal state; it extends a valid MAC digest which can be used to produce a valid MAC digest without information on the MAC key.

4. NEW CONSTRUCTION DESIGN

In this section we present a basic concept of the new construction design. We focus our reflection on how to combine efficiency, security and originality.

It is an extension domain algorithm with a specific internal transformation that processes non fixed length messages, and generates a fixed length block as final output. The following points describe some specific characteristics:

- There is no initialization vector to handle.
- The internal transformation has a roleto extend the input blocks, not to compress them as in the classic paradigm.
- Each input block is processed separately.
- This model can provide a pseudo parallel processing based on separate input blocs.
- It applies a sequential xor addition, involving all extension function outputs.
- This construction design joins a pseudo parallel behavior to an iterative one.

The following Figure 3, illustrates the design characteristics:

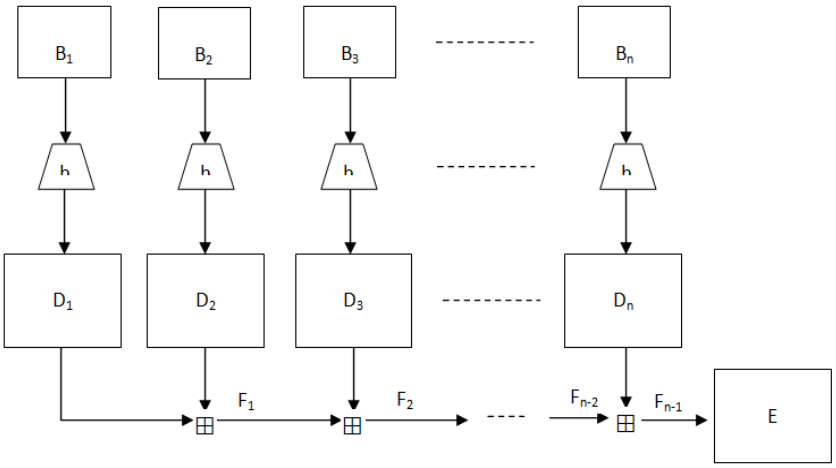

Figure 3. The new design construction

Figure 3, we have the following parameters:

- $B_i$theith message block input.
- h: the internal extension function.
- $D_i$the (h) function output in theithstep
- $F_i$thexor addition output in theithstep
- E: the final output block.

Hence, we canrepresent it as amathematical system where (^) denotes the Xor addition:

$$E = D_n \text{ ^} \cdots (D_{i+1} \text{ ^} (D_i \text{ ^} (\cdots (D_3 \text{ ^} (D_2 \text{ ^} D_1)) \cdots))$$

Otherwise:

$$F_i = F_{i-1} \text{ ^ } D_{i+1} \text{ where } \begin{cases} D_i = \mathrm{h}(B_i) \forall\ 1 \leq \mathrm{i} \leq \mathrm{n} \\ F_1 = D_2 \text{^} D_1 \end{cases}$$

$$E = F_{n-1}$$

5. SECURITY ANALYSIS

5.1. Distinctive secure properties

We based our approach on the most important points that make it different from the classic paradigm:

- Except the definition of the internal function (h), the random content of the input message bloc can be extended to the design behavior.
- Avoiding use of the initialization vector increases the randomness and prevents any content manipulation.
- The internal transformation is not the kernel as in the classic design, so a potential local weakness cannot impact automatically the whole design.
- The operation mode can be parallelized because of input blocks independence, this point can be applied in separate data architecture as in Cloud and distributed database.
- Even if the internal function extends the input blocks, applying a sequential xor addition on the output blocks insures the compression role.
- The compression effect is seen after processing the entire input blocks.

6. RESULTS AND DISCUSSION

6.1. Why these structure choices

Here we give a check list point explaining the security reasons behind these structure choices:

- Ideal randomness is a needed property in a secure construction, as quoted above; this design provides a height level of randomness, because it depends on the random input contents, no use of initialization vectors IV.
- Examining a pseudo collision attack, based on a free choice of the IV is infeasible while this option is absent in this construction.
- The Fixed point attack [5] benefits from dependent input blocks process as in the Davies-Meyer construction [21], but in our case such structure is missedbecause each input block is computed separately.
- Taking into consideration the importance of the internal transformation, resistant internal function can be extended to the entire hash function; here with the internal function definition the required properties of a birthday attack [14] are not satisfied.
- A local generic collision attack cannot be provided; final chaining variables in each round are not located before the xor addition process.
- For other generic attacks as first and second pre-image and collision attack it is further difficult to apply in practice, because of missinglocalbirthday attack.
- Using xor addition ensures a height level of complexity and ambiguitydue to one-time pad property [22-23].
- The multi-collision attack [6] looks for multi local hash collisions, and it is based on the iterative nature of the construction and it uses the IV, in our case an intermediate hash is not defined explicitly, the xor addition absorbs all the local output blocks, moreover there is no IV and every input block is processed separately.
- The length extension attack [4], because of missed initial vector IV, there is no way to force a faked hash value to get a valid MAC.
- This structure is well appropriateto use large size input/output blocks, added to independent rounds process and infeasible fixed point attack, it is costlyto apply a brute force attack.

6.2. Comparing with other constructions

To highlight the relevant properties of this new construction, we bring up a models comparison in Table 1, which illustrates some attacks feasibility against some of existing constructions. So, theoritecally and referring to each construction and comparing their operational mode, we can conclude that the new model is safer than the others. Table 1, the first column intitles the applyed attacks, the next columns designate the compared construction models, (NA) the attack can not be applied, (A) the attack can be applied.

In other side, due to pseudo parallel block processing this new operational mode can let interesting perspectives to distributed systems as Big Data, Blockchain, Crypto-currency, Cloud… etc. While the first implemented variant is not ready, it is earlier to discuss deeply the performance and the experimentation in a distributed environment.

Table 1. Model comparison

Collision Attack	New Model	Merkle–Damgård	Sponge function	HAIFA
Fixed point attack	NA	A	A[24]	A[26]
Multi collision attack	NA	A	A[24]	A[26-28]
Length extension attack	NA	A	NA[25]	A[26]

7. CONCLUSION

In this article we have presented a new design construction for cryptographic hash functions. It is an alternative candidate to current vulnerable constructions. Understood the attack strategies lead us to achieve this goal thereby analyzing the structural vulnerabilities in the classical paradigm. So we have determinate four principal properties, sources of design weakness: The iterative process, use of initial vector, chaining variables (dependent input process), and internal compression functions with small co-domain size compared to domain size. This analysis was the key master in our approach; hence we focus our thinking to innovate a design that gives the expected role and do not be inspired by classic model. Also in the pseudo parallel process, we let an opportunity to implement new hash function generation well adapted to the distributed data architecture. Maybe this proposition is not a mature and perfect one, but it gives a new approach to implement hash functions. Now, as a perspective we are working on a new hash function implementation based on this design construction, by the way, we can challenge its efficiency, security and adaptability.

REFERENCES

[1] "NIST Comments on Cryptanalytic Attacks on SHA-1 | CSRC". [Online]. Available: https://csrc.nist.gov/news/2006/nist-comments-on-cryptanalytic-attacks-on-sha-1. [Accessed: 07-Sep.-2019].
[2] "Request for Candidate Algorithm Nominations | CSRC". [Online]. Available: https://csrc.nist.gov/news/2007/request-for-candidate-algorithm-nominations. [Accessed: 07-Sep.-2019].
[3] "NIST Selects Winner of Secure Hash Algorithm (SHA-3) Competition ...". [Online]. Available: https://www.nist.gov/news-events/news/2012/10/nist-selects-winner-secure-hash-algorithm-sha-3-competition. [Accessed: 07-Sep.-2019].
[4] H. Tiwari. Merkle-Damgård Construction Method and Alternatives: A Review. *Journal of Information and Organizational Sciences*. Vol. 41, pp. 283-304, 2017.
[5] G. V., Bard, first, and , "The Fixed-Point Attack." *Springer* US, 2009.
[6] A. Joux. Multi-collisions in Iterated Hash Functions. Application to Cascaded Constructions. In M. Franklin, editor, Advances in Cryptology – CRYPTO 2004, volume 3152 of Lecture Notes in Computer Science, Springer, 2004, pp 306–316.
[7] X. Wang, Y. Lisa Yin, H. Yu: Finding Collisions in the Full SHA-1. In Victor Shoup, CRYPTO'05, Lecture Notes in Computer Science, pp 17-36, 2005.
[8] X. Wang H. Yu. "How to Break MD5 and Other Hash Functions". EUROCRYPT. 2005, ISBN 3-540-25910-4.
[9] S. Marc, B. Elie, K. Pierre, A. Ange, M. Yarik. The First Collision for Full SHA-1. Advances in Cryptology – CRYPTO 2017, 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, Augus.
[10] T. Mehraj, et al., "A critical insight into the identity authentication systems on smartphones". *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, Vol. 13, No. 3, pp: 982-989, 2019.
[11] M. Imdad, et al., "Internet of things (IoT); security requirements, attacks and counter measures." *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, Vol. 18, No. 3, pp: 1520-1530, 2020.
[12] R. Chetan, R. Shahabadkar. "A Comprehensive Survey on Exiting Solution Approaches towards Security and Privacy Requirements of IoT", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol.8, No.4, pp. 2319-2326, 2018.
[13] Vidya M.S, Mala C Patil ,"Reviewing effectivity in security approaches towards strengthening internet architecture." *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 9, No. 5, pp. 3862-3871, 2019.
[14] A. Menezes J., , and S. Vanstone A., Handbook of Applied Cryptography. CRC Press, 1996.
[15] I. Damgård. A Design Principle for Hash Functions. In G. Brassard, editor, Advances in Cryptology – CRYPTO 1989, Lecture Notes in Computer Science, Springer-Verlag, pp 416–427, 1989.
[16] R.C. Merkle. One Way Hash Functions and DES. In G. Brassard, editor, Advances in Cryptology – CRYPTO 1989, Lecture Notes in Computer Science, Springer-Verlag, pp 428–446, 1989.
[17] U.M. Maurer and S. Tessaro. Domain Extension of Public Random Functions: Beyond the Birthday Barrier. In A. Menezes, editor, Advances in Cryptology – CRYPTO 2007, Lecture Notes in Computer Science, Springer-Verlag, pp 187–204, 2007.
[18] M. Bellare, T. Ristenpart. Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In X. Lai and K. Chen, editors, Advances in Cryptology – ASIACRYPT 2006, Lecture Notes in Computer Science, Springer-Verlag, pages 299–314, 2006.
[19] E. Andreeva, G. Neven, B. Preneel and T. Shrimpton. Seven-Property-Preserving Iterated Hashing: ROX. In K. Kurosawa, editor, Advances in Cryptology – ASIACRYPT 2007, Lecture Notes in Computer Science, Springer-Verlag, 2007, pp 130–146.

[20] E. Biham, O. Dunkelman. A framework for iterative hash functions-HAIFA. IACR Cryptology ePrint Archive. 2007. 278.
[21] L. Knudsen, M. Robshaw. The Block Cipher Companion. Springer-Verlag, Berlin, 2011, p. 88.
[22] "XOR and the one-time pad (article) | Ciphers | Khan Academy". [Online]. Available: https://www.khanacademy.org/computing/computer-science/cryptography/ciphers/a/xor-and-the-one-time-pad. [Accessed: 07-Sep.-2019].
[23] O. Oludare, J. Aman, A. Oludare, H. Arshad. An Enhanced Practical Difficulty of One-Time Pad Algorithm for Resolving the Key Management and Distribution Problem. Proceedings of the International MultiConference of Engineers and Computer Scientists 2018 Vol I, March 14-16, 2018, Hong Kong.
[24] B. Tareq, Hammad, first, and, "Faster Multicollisions Attack on Sponge Construction". American Scientific Publishers, 01-Jun.-2017.
[25] "Keccak Team". [Online]. Available: https://keccak.team/keccak_strengths.html. [Accessed: 07-Sep.-2019].
[26] Biham, Eli; Dunkelman, Orr (24 August 2006). A Framework for Iterative Hash Functions - HAIFA. Second NIST Cryptographic Hash Workshop – via Cryptology ePrint Archive: Report 2007/278.
[27] Gaëtan, Leurent, first, and, "The Sum can be Weaker than Each Part". Springer Berlin Heidelberg, 2015.
[28] Zhenzhen, Bao, first, and, "Generic Attacks on Hash Combiners". Springer Science and Business Media LLC, 12-Jul.-2019.