Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Modern Hash Collision CyberAttacks and Methods of Their Detection and Neutralization

To cite this article: Olga Safaryan *et al* 2021 *J. Phys.: Conf. Ser.* **2131** 022099

View the article online for updates and enhancements.

You may also like

- Double hashing technique in closed hashing search process
Robbi Rahim, Iskandar Zulkarnain and Hendra Jaya
- Research on Discrete Hash Algorithm Based on Deep Semantics
Qinjin Jia
- Detecting similarity in color images based on perceptual image hash algorithm
Nada Hussein M. Ali and Marwa Emad Mahdi

IOP ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection–download the first chapter of every title for free.

This content was downloaded from IP address 209.242.205.171 on 30/12/2021 at 01:38

Modern Hash Collision CyberAttacks and Methods of Their Detection and Neutralization

Olga Safaryan[1*], Larissa Cherckesova[1], Nikita Lyashenko [1], Pavel Razumov [1], Vladislav Chumakov[1], Boris Akishin[1] and Andrey Lobodenko [1]

[1] Don State Technical University, Gagarin Square, 1, Rostov-on-Don, 344003, Russia

E-mail: safari_2006@mail.ru

Abstract. This article discusses the issues related to the possibility of realization of collision cyberattacks (based on hash collisions). Since post–quantum cryptography has become relevant, classical cryptosystems do not provide the sufficient resistance to the modern quantum cyberattacks. Systems based on outdated hashing algorithms become vulnerable to cyberattacks with hash collision. As replacement for unreliable algorithms, such as various modifications of MD5 and SHA–1, new algorithms have been created, for example, SHA–3 standard based on the Keccak function and AES–based hashing. This article discusses modern collision cyberattacks and possible methods of their detection. Because of this study, theoretical description of cyberattacks with hash collision was considered; modern cyberattacks on hash collisions and possible ways of detecting and countering them (weak hash detection) are described; software tool that detects vulnerable and unreliable hash is implemented; software testing is carried out. Based on the conducted research, it can be concluded that the main advantages of implementing software tool are effective detection of vulnerable hash, the ability to generate new hash protected from collisions, convenient and user– friendly interface, small memory requirements and small size of the program code.

1 Introduction

For several decades, quantum computer technologies have been actively developing and scientists from many countries have developed new algorithms that ensure information security. Attackers, in turn, have implemented more complex and effective cryptographic cyberattacks. In the modern world, postquantum cryptography has become relevant, since classical cryptosystems do not provide sufficient resistance against modern quantum cyberattacks. It means that systems based on outdated hashing algorithms become vulnerable to hash collision cyberattacks.

As replacement for unreliable algorithms, such as various modifications of MD5 and SHA-1, new algorithms have been created, for instance, the SHA-3 standard, which is based on the Keccak function and AES-based hashing. In this paper the modern hash collision cyberattacks and some possible methods of countering them will be considered.

The object of research is hash collision cyberattack algorithm.

The subject of the research is the algorithmic complexity of the hash function algorithm development. The aim of this research is to develop the special software tool capable of weak hash detection in attempt to resist to the modern cyberattacks of hash collisions. In accordance with this goal, several research objectives were identified and implemented:

- to investigate the algorithm of hash collision attacks;
- to consider the modern types of hash collision attacks;
- to implement the special software tool in attempt to resist to the modern cyberattacks of hash collisions.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

2 Basic theoretical information about collision attacks

A hash function H collision is two different input data blocks x and y such that H(x) = H(y). Collision cyberattack is based on the searching of hash collision [1].

To perform hash collision cyberattack the attacker needs to find two messages that hash values are the same [2]. Classic collision attack is a search for 2 different messages m_1 and m_2 such that $H(m_1)=H(m_2)$, where H is hash function [3]. Collision cyberattack with prefix, given before, is finding of messages m1 and m2 such that $H(p_1 \| m_1) = H(p_2 \| m_2)$, where ‖ is operation of concatenation; and p1 and p2 are predefined messages (prefixes) [4].

Let us consider an authentication system in which the user chooses some password X, for which the value H(X) is calculated and stored on the server. To successfully complete the authentication procedure, the user must enter the password, and if the hash value for this password matches the value stored on the server, then the authentication procedure is considered passed. The advantage of this system is that the server does not store the password itself, but its hash is stored, which cannot be utilized to recover unambiguously the password.

However, an attacker can gain access to the system by performing hash collision cyberattack on the hash function. If an attacker finds certain string Y for which H(X) = H(Y), then he will be able to pass the authentication procedure by entering the password Y without knowing the password X [5]. It should be mentioned that hash collision cyberattack with given before prefix is much stronger, since it allows detecting hash collisions with certain predefined properties. In practice, it means that this cyberattack can be employed to forge the electronic documents that was secured by electronic signatures based on the hash function.

In 2017, the “SHAttered prefix hash collision cyberattack” was developed and implemented in practice [6]. This cyber attack allows to change signed by SHA–1 files in the way that criminal needs, starting from falsifying documentation and ending with TLS certificates falsifying [7]. The implementation of “SHAttered prefix hash collision cyberattack” requires computation with complexity of 263 [8].

As a consequence, SHA–1 ceased to be reliable algorithm and it should be utilized the algorithms SHA–2 or SHA–3 in the modern applications to ensure their security [9].

Let us consider the application of hash collision cyber attack to forge an electronic signature. Let Alice and Bob be the users who send documents and use digital signature based on hashing, and Eve is an attacker. In this case, the following algorithm is used to carry out hash collision cyberattack:

If Eve manages to create successfully two different documents A and B that have about the same hash values, then she can trick Bob into thinking of her document as Alice's.

1. Eve sends document A to Alice.
2. Alice signs its hash and sends the signature to Eve.
3. Eve attaches the document A signature to document B.
4. Eve sends to Bob the signature and the document B, claiming that Alice signed the document. Since the electronic signature verifies the hash value of document B only, Bob does not know about the substitution [10].

In Figure 1 the counterfeiting process is demonstrated.

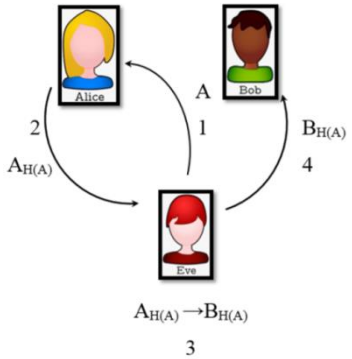

Fig. 1. Counterfeiting digital signature by hash collision cyberattack.

3 Analysis of modern hash collision cyberattacks

The application of quantum computer allows faster search for collisions for most of the existing algorithms. So, it is known that effective hash collision cyberattacks have been found against AES–MMO hashing algorithm [11]. Upon analyzing of modification of algorithm with seven rounds, two models of quantum collision attack were developed [12].

Cyberattack by first model has computational complexity of 245 and requires 216 qubits of memory. Second model allows implementing hash collision cyberattack without storing data in quantum memory, but requires more computations – 259 [13].

To search for collisions of SHA–3 hash, new modern hash collision cyberattacks have been developed, based on quantum Grover's search algorithm [14].

Modifications SHA3–224 and SHA3–256 are not sufficiently resistant to the quantum cyberattacks, and modifications with much longer hash length SHA3–384 and SHA3–512 should be used to ensure sufficient security [15].

Table 1. Demonstrate the complexity of hash collision cyber attacks on the different modifications of SHA–3 algorithm:

Modification	Classical hash collision cyberattack	Quantum hash collision cyberattack
SHA3–224	112	74⅔
SHA3–256	128	85⅓
SHA3–384	192	128
SHA3–512	256	170⅔

Differential cryptanalysis is also used to search for hash collisions. This method is based on the study of the differences between the values of the function on different rounds of encryption [16]. When attacking the SHA–1 algorithm, it is effective to use two techniques of differential cryptanalysis.

The first common technique is linearization. It is based on the representation of the function in the linear form, which allows finding pairs of messages that are close to collisions (almost the same hash values). Then the Hamming distance is calculated to estimate similarity of the hash values. Message modification is another technique of differential cryptanalysis. To use it, an attacker selects some message and tries to find the collision by changing this modification [17]. Another method of finding collisions, which is based on using of differential cryptanalysis, is the Rebound cyberattack.

The main idea of this cyberattack is to study the differential characteristics of block cipher (or its fragments), permutation or other low–level cryptographic algorithms [18].

In this case, some of the characteristics are determined deterministically, and the rest are written in probabilistic form. Rebound cyberattack consists of two phases: internal and external. In the internal phase, an attacker needs to find many solutions for some of differential characteristics that are difficult to perform in probabilistic form. This is achieved by composing and subsequent solving corresponding system of equations. In the external phase, the solutions obtained are used for calculations in forward and reverse directions [19].

Using the Rebound cyberattack allows to perform the effective search the collisions for Whirlpool function, which is resistant to all other types of hash collision cyberattacks. There is also theoretical description of the Rebound cyber attack on the Grostl–256 hash function [20].

4 Software development and testing

The implemented software tool is designed to check the reliability of the hash. Since many hashing algorithms (for example, SHA–1, MD–5) are vulnerable to modern hash collision cyberattacks, their usage does not guarantee the integrity of the file. The program determines if generated hash is quite

resistant to collision cyberattacks. If hash function is vulnerable, the program allows the user to generate new hash value using the modern SHA–3 algorithm that is proven to be safe. This implementation uses the SHA–3 256 modification. The flowchart is shown in Figure 2.

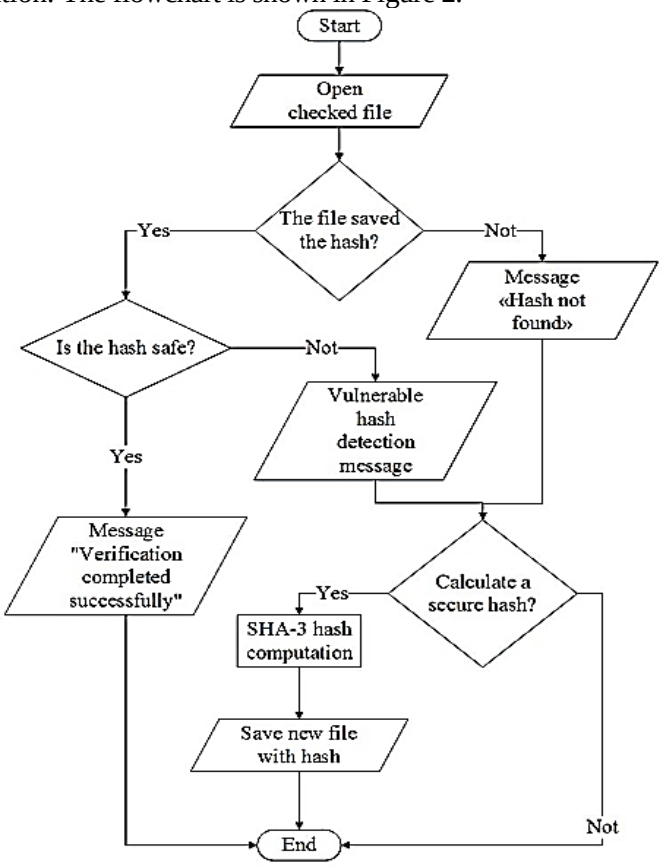

Fig. 2. Flowchart of software tool algorithm for checking hash function reliability

The software implementation of the hash vulnerability check is written in the programming language Python. The graphical interface of the program is realized by using the SimpleGUI library. In order to run the program, user needs to click the file selection button and choose the file that should be checked in the opened dialog box.

The interface of the software tool is shown in the Figure 3.

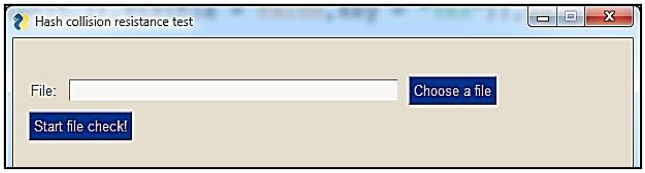

Fig. 3. Software interface

As an example, let us consider the file File _1. txt, which contains the hash that was generated by vulnerable to collision cyberattacks MD–5 algorithm. File is shown in the Figure 4.

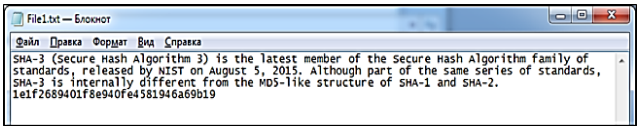

Fig. 4. File with vulnerable hash MD–5

After determining the required file, it is necessary to select "Check file" button. The program finds the hash in the file and outputs the message stating that this hash is vulnerable. The user is also asked to create the secure hash. The vulnerability message is demonstrated in the Figure 5.

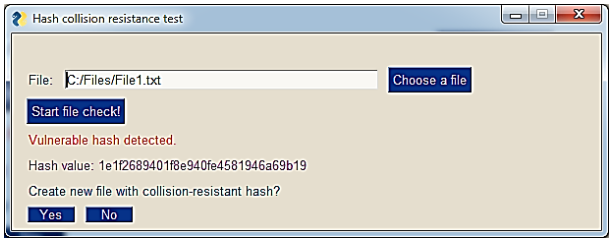

Fig. 5. Vulnerable hash detection

If the user agrees to create new file, the program calculates the SHA–3 hash and saves the contents of the scanned file with the secure hash to the new file. The message about creating the new file is shown at the Figure 6.

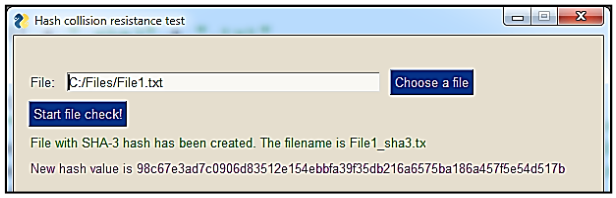

Fig. 6. Generating the hash using the SHA–3 algorithm

New file with hash that was generated by this implemented program is demonstrated in the Figure 7.

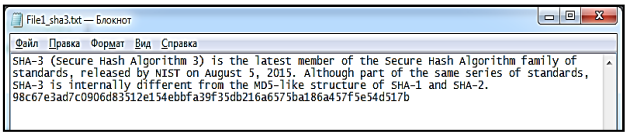

Fig. 7. Text file with SHA–3 hash

Let's check the new file. Unlike MD–5, algorithm SHA–3 is resistant to hash collision cyberattacks and meets security requirements. The result of the check is shown in Figure 8.

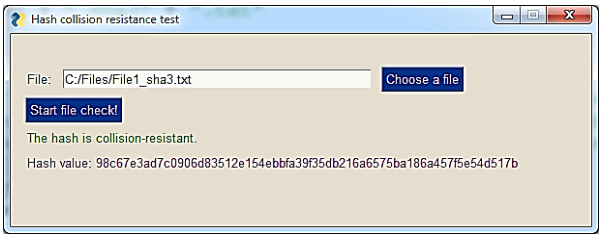

Fig. 8. Checking hash resilience to collision cyberattacks

5 Conclusion

As a result of this investigation:

- the theoretical description of hash collision cyber attacks has been considered;
- the modern hash collision cyberattacks and possible ways to detect and counter them (weak hash detection) have been described;
- the software tool that detects the vulnerable and unreliable hash has been implemented;
- software testing is performed.

Based on the research carried out, it can be concluded that main advantages of the software tool implementation is the effective detection of vulnerable hash, the provided option of generating of new collision–proof hash, the convenient and user–friendly interface, the small memory requirement and the little size of the program code.

References

[1] Razumov P.V., Cherckesova L.V., Safaryan O.A., 2020 et al. "Developing of Algorithm of HTTP FLOOD DDoS Protection", *IEEE 3rd International Conference on Computer Applications & Information Security, IEEE ICCAIS'20. Saudi Arabia, Er–Riyadh*. Pp. 1–6. DOI: 10.1109/ICCAIS48893.2020.9096870. [Electronic Resource] URL: https://ieeexplore.ieee.org/document/9096870.

[2] Razumov P.V., Cherckesova L.V., Safaryan O.A., 2020 et al. "IAS'2019 Cryptographic Protocol Allowing to Protect the Key in the Open Communication Channel", *IEEE 3rd International Conference on Computer Applications & Information Security*, IEEE ICCAIS'20. Saudi Arabia, Er–Riyadh, **19–21**. Publisher IEEE. Pp. 1–5. DOI: 10.1109/ICCAIS48893.2020.9096729. [Electronic Resource] URL: https://ieeexplore.ieee.org/document/9096729.

[3] Revyakina Y., Cherckesova L., Safaryan O., 2020 et al. "Possibilities of Conducting XSS–attacks and the Development of Counter Measures". Topical Problems of Agriculture, Civil and Environmental Engineering. TPACEE 2020. Moscow, MSRU. *E3S Web of Conferences*. Vol. **224**. Pp.1–9. DOI: 10.1051/e3sconf/202022401040.

[4] Razumov P., Safaryan O., Cherckesova L., 2020 et al. "Post–Quantum Cryptosystem NTRUEnCrypt and Its Advantage over Pre–Quantum Cryptosystem RSA", Topical Problems of Agriculture, Civil and Environmental Engineering. TPACEE 2020. Moscow. MSRU. Pub. *In Journal E3S Web of Conferences*. Vol. **224**. Pp. 1–9. DOI: 10.1051/e3sconf/202022401037.

[5] Zulfany E., Benfano S., Gunawan W., 2017 Edi A. "A review of collisions in cryptographic hash function used in digital forensic tools", *2nd International Conference on Computer Science and Computational Intelligence*, Bali, Indonesia. Pp. 1–12.

[6] Bruneau1 N., Carlet C., Guilley S. 2017 "Stochastic Collision Attack". *Paris University XIII and Paris University VIII*, France. Pp. 1–3.

[7] Liu Q., Zhandry M. 2018 "*On Finding Quantum Multi-collisions*", Princeton University. Newark, NJ, USA. Pp. 26.

[8] Bitansky, N. Tauman, O. 2018 "*Paneth Multi–collision Resistance: Paradigm for Keyless Hash Functions*". Tel Aviv University. Pp. 67.

[9] Dong X., Sun S., Shi D, 2020 "Quantum Collision Attacks on AES-like Hashing with Low Quantum Random Access Memories", Advances in Cryptology. ASIACRYPT, South Korea, Daejeon, Springer International Publishing, Vol. **12492**, Pp. 727–757 DOI: 10.1007/978-3-030-64834-3.

[10] Chiriaco V., Franzen A., Tayil R., Zhang X., 2016 "Finding Partial Hash Collisions by Brute Force Parallel Programming". 37th IEEE Princeton Section Sarnoff Symposium, 19 – 21 Sept. 2016, Newark, NJ, USA. Pp. 1–2. DOI: 10.1109/SARNOF.2016.7846725. [Electronic Resource] URL: https://ieeexplore.ieee.org/document/7846725.

[11] Leurent G. Peyrin T. 2019 "From Collisions to Chosen–Prefix Collisions: Application to Full SHA–1". Eurocrypt 2019. *38th Annual Internaional Conference on Theory and Applications of Cryptographic Techniques*, 11–14 Sept., Darmstadt University. Germany. Pp. 527–555. DOI: 10.1007/978-3-030-17659-4_18.

[12] Mendel F., Rijmen V, Schlaffer M. 2014 *"Collision attack on 5 rounds of Grostl"*, FSE 2014, Heidelberg. LNCS Vol. **8540**. Pp. 509–521.

[13] Stevens M., Dursztein E., Karpman P. "Freestart Collision for the Full SHA–1", Eurocrypt 2016, Lncs. Vol. **9665**. Pp. 459–483.

[14] Eichlseder M., Mendel F., Schaffer M. 2014 "Branching Heuristics in the Differential Collision

Search with Applications to SHA–512", FSE 2014. Springer. Pp. 1–16.
[15] Dobraunig C., Eichlseder M., Mendel F. 2015 "Security Evaluation of SHA–224, SHA–512/224 and SHA–512/256", Graz University of Technology, Graz, Austria. Pp. 21–44.
[16] Amy M., Matteo O., Gheorghiu V. 2016 "Estimating the Cost of Generic Quantum Pre–Image Attacks on SHA–2 and SHA–3", Selected Areas in Cryptography. Pp. 317–337.
[17] Charmaine C., San Jose G. 2019 "Comparative and Security Performance Analysis of SHA–3", Journal of Advanced Research in Dynamical and Control Systems. Vol. **5**. Issue **11**. Pp. 960–966. DOI: 10.5373/JARDCS/V11SP11/20193121.
[18] Zhou T., Zhu Y., Jing N, Nan T. 2020 "Reliable SoC Design and Implementation of SHA–3–HMAC Algorithm with Attack Protection", *IEEE International Conference on Smart Cloud,*. Pp.88–93. DOI: 10.1109/SmartCloud49737.2020.00025.
[19] Al–Odat Z., Abbas A., Khan S. 2019 "Randomness Analyses of the Secure Hash Algorithms, SHA–1, SHA–2 and modified SHA", 2019 *International Conference on Frontiers of Information Technology (FIT)*. Pp. 3160–3165. DOI: 10.1109/FIT47737.2019.00066.
[20] Karthiga S. Velmurugan T. 2019 "Security Based Approach of SHA–384 and SHA–512 Algorithms in Cloud Environment", *Journal of Computer Science*. Vol. **16(10).** Issue **11**. Pp. 1439–1450. DOI: 10.3844/jcssp.2020.1439.1450