Performance and Security Analysis of Hash Functions

Sultan Almuhammadi and Omar Bawazeer
College of Computer Sciences and Engineering
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
Emails: (muhamadi, g201407380)@kfupm.edu.sa

***Abstract*—Hashing plays a major role in network security. A cryptographic hash function is an important component used in many applications, such as blockchain, authentication, data integrity, and digital signature. With the rapid increase usage of mobile devices, more attention goes towards wireless network to evaluate the trade offs between performance and security of hash functions on mobile devises and wireless networks. This paper presents a comprehensive study of the most common cryptographic hash functions and compares them in terms of their performance and security. The hash functions we considered in this study are: MD4, MD5, Whirlpool, and the hash functions in the SHA family. We compare the security of these hash functions based on recent attacks. In addition, we measured and compared the performance of these hash functions. We also designed a sensitivity test algorithm and implemented it to compare the sensitivity of these hash functions. We summarized the results and provided useful recommendations.**

***Keywords*—Secure hash function, collision resistance, integrity, authentication.**

I. Introduction

The importance of cryptography and information security has grown rapidly to meet security goals in the digital world. As networking and communication fields grow, computer specialists have developed many tools to satisfy the needs of security for individuals and organizations. With today's usage of mobile devices, wireless network become essential. However, the limited resources in mobile devices and wireless network creates the need to study the trade offs between performance and security in these technologies.

Improvements of security and cryptographic tool include: symmetric/asymmetric-key encipherments, cryptographic hash functions and many other tools for different applications [1], [2], [3]. On the other hand, the development of information security coincided with the improvement of attacking techniques that try to get access to confidential information, harm the system, etc. [2].

Besides network security, secure hashing functions have many applications. The security of data should be maintained both when it's static or dynamic [4]. For static security, on the computing devices, the stored data must be legitimately encrypted and controlled [4]. For dynamic security, suitable network security actions must be set up to ensure the information through its transmission [4].

Cryptographic hash functions are used in many services and mechanisms such as Data Integrity, Authentication, Non-repudiation, Digital Signature, Blockchain, and so on.A system needs to guarantee at least one of these relying on the security prerequisites for a specific system. The cryptographic hash functions techniques are used to accomplish some of these security services. Cryptographic hash functions are designed in two ways: one way is to make it from scratch (like MD5, MD6, SHA-x) and the other way is based on Block cipher (like Whirlpool) [2]. Unfortunately, there are several ways to attack the hashed information by using some attacking techniques such as preimage attack, second preimage attack, and collision attack [2], [3].

This paper presents a detailed study on the current hash functions, including their strengths and weakness points. In addition, two experiments are conducted to test the performance of the hash functions and their sensitivity to the input change. Based on the results of this study, suggestions and useful recommendations are provided for mobile devices and wireless networks.

II. Cryptography and Network Security

Cryptography performs a significant part in securing the confidential information in different applications such as medical databases, e-commerce, e-mail, e-banking, etc. It also plays a major role in network security applications, including: confidentiality, integrity, and availability [2].
Confidentiality: ensures the privacy of data in such a way that no one can read the message unless authorized.
Integrity: means that any change or modification must be done by the authorized entities.
Availability:means that the information must be available for the authorized entities.

A. Cryptographic Hash Functions

A cryptographic hash functions is one-way function that takes an input of a variable-length and produces a message digest (or hash-value) which has a fixed output-size. Thus, $H(M) = h$ [2], [3].

A cryptographic hash function, H shown in Figure 1, has the following properties: [1]

1) The input data or message M has a variable-length.
2) The output (message digest) of H has a fixed output-size $|h|$.

3) It is computationally infeasible to find x for a given H and $H(x)$.
4) It is computationally infeasible to find x and y such that $H(x) = H(y)$.

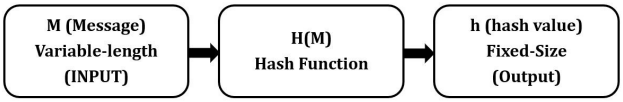

Figure 1. Block Diagram of Cryptographic Hash Function $H(M) = h$

To produce the message digest, all cryptographic hash functions now are using iteration. So, the long message input of an arbitrary size is divided into k fixed size segments that will be compressed by a compression function which accepts an n-bits input to produce m-bits output, where $n \geq m$ typically, and iterated k times to create the output [2]. Depending on the designing of compression function, the cryptographic hash functions are classified into two types. The first one is made from scratch, which originally designed for the hash functions like the Message Digest (MD) family (MD2, MD4, MD5, and MD6) and the Secure Hash Algorithms (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512). The second type is the hash functions that based on block ciphers such as Whirlpool [2], [3]. Table I summarizes the details of the most common hash functions.

Table I. SUMMARY OF STANDARD HASH FUNCTION

Hash	Block size	Word size	Output size	Rounds	Year
MD4	512 bits	32 bits	128 bits	48	1990
MD5	512 bits	32 bits	128 bits	64	1992
SHA-0	512 bits	32 bits	160 bits	80	1993
SHA-1	512 bits	32 bits	160 bits	80	1995
SHA-224	512 bits	32 bits	224 bits	64	2004
SHA-256	512 bits	32 bits	256 bits	64	2002
SHA-384	1024 bits	64 bits	384 bits	80	2002
SHA-512	1024 bits	64 bits	512 bits	80	2002
Whirlpool	512 bits	-	512 bits	10	2003

There is a trade-off between the security of the hash function and its performance. Many cryptographic hash functions nowadays have different strength, and they are typically categorized into two types: (1) more secure but slow to run, and (2) less secure but fast to run [5].

According to [1], a good hash function should meet some important properties that insure its security. Basically, it should resist the collision, preimage, and second preimage attacks. It also should meet the avalanche criterion and completeness requirements. C

ompleteness means that any change in the input affects the whole message digest. Avalanche criterion relates between completeness and avalanche effect which means that tiny change in input produces a vital variation in output.

The following three important properties or criteria must be satisfied by any cryptographic hash function [1], [6].

1) *Preimage Resistance*: Let H be given as a hash function and $H(M) = y$ it must be computationally infeasible for an adversary to find any preimage, M', such that $h(M') = y$.
2) *Second Preimage Resistance*: The second preimage resistance guarantees that a preimage of a message cannot be computed from its digest. Thus, given M and $H(M)$, it is computationally infeasible to find another message $M' \neq M$ that has the same digest $H(M') = H(M)$.
3) *Collision Resistance*: It should be difficult to create two messages M and M' such that have equal message digest $h(M) = h(M)$.

III. SECURITY OF HASH FUNCTIONS

Kishor and Kapoor [4] mentioned that SHA-1 and MD5 are commonly used in data integrity and other security applications, include: password protection, digital forensics, digital signature, and image encryption. In addition, they claimed that the SHA-3 works better in performance than SHA-2 on multicore processor. They presented a survey on attacks on secure hash functions. Tables II and III summarizes the known attacks on SHA0, SHA1, and SHA2.

Table II. ATTACKS ON SHA0 AND SH1

Hash	Attack	Year
SHA-0	Collisions with complexity 2^{61}	1998
SHA-0	Full Collisions of 65 round and collision with 142 bits equal	2004
SHA-0	Collisions with complexity 251	2004
SHA-0	Collisions with complexity 2^{40}	2004
SHA-0	Collisions with complexity 2^{39}	2005
SHA-1	Collisions possible for 53 rounds instead of 80	2005
SHA-1	Collisions with complexity $< 2^{69}$ operations	2005
SHA-1	Collisions with complexity 2^{63}	2005
SHA-1	two-block collision for 64-round	2006
SHA-1	Complexity equivalent to $2^{57.5}$	2010

Table III. ATTACKS ON SHA2

Hash	Attack	collision	Year
SHA-256	Deterministic Collision	In 24/64 rounds with $2^{28.5}$complexity	2008
SHA-512	Deterministic Collision	In 24/80 rounds with $2^{32.5}$ complexity	2008
SHA-256	Preimage, Meet-in-the-middle	In 42/64 rounds with $2^{251.7}$ complexity	2009
SHA-256	Preimage, Meet-in-the-middle	In 43/64 rounds with $2^{254.9}$ complexity	2009
SHA-512	Preimage, Meet-in-the-middle	In 42/80 rounds with $2^{502.3}$ complexity	2009
SHA-512	Preimage, Meet-in-the-middle	In 46/80 rounds with $2^{511.5}$ complexity	2009
SHA-256	Preimage, Meet-in-the-middle	In 42/64 rounds with $2^{248.4}$ complexity	2010
SHA-512	Preimage, Meet-in-the-middle	In 42/80 rounds with $2^{494.6}$ complexity	2010
SHA-256	Pseudo Collision, Differential	In 46/64 rounds with 2^{178} complexity	2011
SHA-256	Pseudo Collision, Differential	In 46/64 rounds with 2^{46} complexity	2011
SHA-256	Preimage, Biclique	In 45/64 rounds with $2^{555.5}$ complexity	2011
SHA-512	Preimage, Biclique	In 50/80 rounds with $2^{511.5}$ complexity	2011
SHA-256	Preimage, Biclique	In 52/64 rounds with 2^{555} complexity	2011
SHA-512	Preimage, Biclique	In 57/80 rounds with 2^{511} complexity	2011

Lathwal and Khanchi [7] indicated that there is an increasing in applications that depend on message authentication code (MAC) which derived from hash values and the main characteristic of the cryptographic hash function like SHA-1 and MD5 is the execution speed in comparison

with other techniques such as symmetric block cipher like DES. Also, they mention that the MD5 has been defined for using in Internet Protocol Security (IPSEC), as the reason for an HMAC.

According to Sobti and Ganesan [1], hash functions are one of the most important techniques that are spread in the cryptography field and used to execute a number of network security services and achieve the desired goals. These services included: digital signature, authenticity, digital time stamping, integrity, and authenticating users of computer systems using message digest. Moreover, they are used in steganography, pseudo random number generator, session key derivations to protect the successful communication session, constructions of block ciphers (SHACAL-1,SHACAL-2), indexing data in hash table, fingerprinting, duplicate data detection, uniquely identifying files, checksum to find the corruption data, and generation random numbers.

Sobti and Ganesan also categorized the attacks on hash functions into two wide classifications (Figure 2): Brute Force Attacks and Cryptanalytical Attacks. Brute-force attacks are similar to exhaustive search and contain collision, preimage, second preimage attacks, K-Way Collision attack for $K \geq 2$, and K-Way Second Pre-image attack for $K \geq 1$. Whereas, cryptanalytical attacks concentrate with the underlying construction of hash function and the algorithm of compression function, and include two types: (1) generic attacks (which include: Length Extension attacks, Joux-Multicollision attacks, Multi-Second Preimage attacks based on Joux Technique, Generic Second Preimage attacks, Correcting block attack, Fixed-Point attacks, Herding attacks, and Meet-in-the-Middle attack) and (2) specific attacks, like: Differential Cryptanalysis, Linear Cryptanalysis, Rotational Cryptanalysis, and attacks on underlying encryption algorithm.

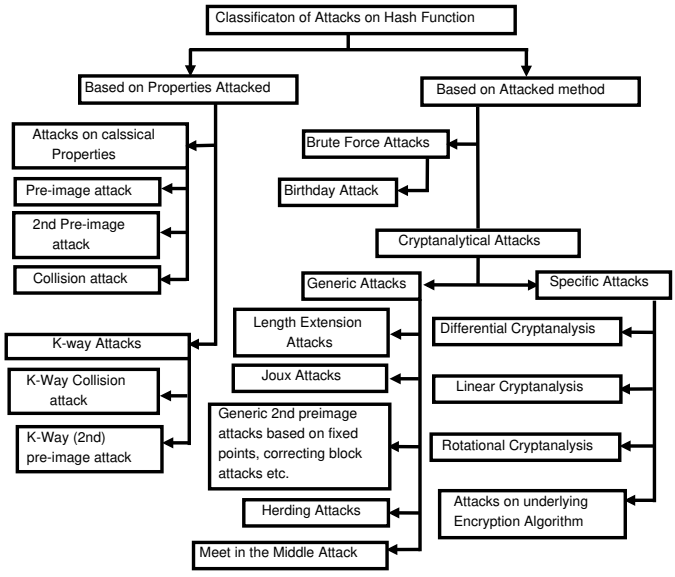

Figure 2. Classification of Attacks on Hash Functions [1]

Verma and Prajapati [8] conducted experiments on SHA using their own algorithm. They claimed that the proposed algorithm is more powerful than others because it takes less time to produce the message digest and it has more sensitivity to avalanche effect with about 50% of bit difference if only one bit is different in the input message. Table IV shows these results.

Table IV. VERMA AND PRAJAPATI RESULUTS [8]

Techniques	Hashing time	Bit Diference between 2 messages
SHA-160	1.541	41.625%
Propoded SHA	1.699	49.99%
SHA-192	2.93	36.375%

In addition, Verma and Prajapati mentioned that Bihamet lunched a cryptanalyst attack in 2004 and he broke SHA-0 collision domain at 2^{41}. They indicated that the SHA-1 is the most extensively utilized algorithm for integrity because of its robustness and its time efficiency. However, in 2011 the collision was produced with a complexity of 2^{61} operations by Marc Stevens. Also they mention that SHA-2 is more secure than SHA-1 but is not time efficient.

Gupta et al. [5] made a comparative study between Secure Hash Algorithm SHA (SHA-1, SHA-256, SHA-384, and SHA-512) and they found that the most secure algorithm is SHA-512 which produces 512 bits as a message digest and accepts block size of 1024 bits. On the other hand, SHA-512 is time consuming and a bit lengthy [5]. They claimed that they made their analysis completely depending on a level of security that was tested in terms of the output randomness. Table V summarizes the results of experiments in [5].

Table V. RESULTS OF GUBTA ET AL. EXPERIMENTS

Test	Values	SHA-1	SHA-256	SHA-384	SHA-512
K-S	0.33	0.31	0.16	0.08	0.11
Chi square	16.9	8.0	9.25	2.41	12.8
Auto-correlation	1.96	1.35	1.16	0.24	0.47

Wang and Yu presented [9] a robust attack on MD5, MD4, and other hash functions. They claimed that the modular differential attack is able in 15 to 60 minutes computation time to find the collision of MD5 and it needs less than one second to calculate the collision of MD4. They mentioned that the digital signature security depends on the strength of cryptographic hash functions. They stated many other applications such as group signature, e-cash, data integrity, and many other cryptographic protocols. These hash functions are used not just to ensure the security of the applications but also to provide a great efficiency for them [9]. Table VI shows the results of complexity time to compute the collision of MD4 and SHA-0 hash functions.

Table VI. WANG AND YU RESULTS

modification	MD4	SHA-0
Without multi-message	2^{23}	2^{61}
With multi-message	2^{8}	2^{45}

Sasaki et al. [10] claimed that by using preimage attack, they improved the complexity of memory on full MD5 to 2^{13} compared with 2^{45} in the previous attack as shown in Table VII.

According to Mironov [11], cryptographic hash functions are used in many applications such as digital signature, message authentication code (MAC), password tables, key derivation, universally Unique Identifier also known as Globally

Table VII. PREIMAGE ATTACK ON FULL MD5

#Steps	Min Preimage Length	Complexity	Memory	Reference
64 (full)	2^{33} blocks	$2^{123.4}$	2^{45}	(Sasaki, 2009)
64 (full)	2^{33} blocks	$2^{123.4}$	2^{13}	[10]

Unique Identifier (UUID/GUID), and hash table. Mironov presented a summary of some attacks on standard hash functions as shown in Table VIII

Table VIII. SUMMARY OF ATTACKS ON STANDARD HASH FUNCTION

Hash	Attack type	complexity	Year
MD4	collision	2^{22}	1996
MD4	collision	2^{8}	2005
MD5	pseudo-collision	2^{16}	1993
MD5	free-start	2^{34}	1996
MD5	collision	2^{39}	2005
SHA-0	collision	2^{61} (theory)	1998
SHA-0	near-collision	2^{40}	2004
SHA-0	collision	2^{51}	2005
SHA-0	collision	2^{39}	2005
SHA-1	collision (40 rounds)	verylow	2005
SHA-1	collision (58 rounds)	2^{75} (theory)	2005
SHA-1	collision (58 rounds)	2^{33}	2005
SHA-1	collision	2^{63} (theory)	2005

According to their experiments in [12], Galbally et al. claimed that the attacking of SHA-3 hashes are more slower than attacking hashes of MD5. They found that the cracking speed of SHA-3 was 0.3×10^9 hash/sec. Whereas the cracking speed of MD5 was 20×10^9 hash/sec. SHA-3 is used in many applications such as authenticated encryption, message authentication coding, and the sponge construction in pseudo-random number generation. And as other types of hash algorithms, it can be used for digital signature, in protection of passwords, data identifier or files[13].

Ma et al. [14] claimed that the results of their attack are the best cryptanalytic findings on Whirlpool based on the number of rounds analyzed under the setting of the hash function. By using their proposed limited-birthday distinguisher, they obtained a new 9-round distinguisher and decreased the time complexity of the preceding 7-round distinguisher. The summary of attacks on Whirlpool hash function is shown in Table IX

Table IX. SUMMARY OF ATTACKS ON WHIRLPOOL

Attack Type	Rounds	Time	Memory	Year
Collision	5.5	2^{120}	2^{64}	2013
Preimage	6	2^{481}	2^{256}	2012
limited-birthday distinguisher	7	2^{440}	2^{128}	2013
	7.5	2^{368}	2^{144}	2014
	9	2^{354}	2^{158}	2014

Almuhammadi and Amro [15] conducted a sensitivity test experiment to measure the avalanche effect of changing one bit in the input message on the message digest. They applied their algorithm on a number of hash functions, including: MD4, MD5, and the SHA family. They found that the SHA-512 has the best sensitivity score than other hash functions. Table X shows their sensitivity tests results.

From the above discussions, we summarize the following points regarding the security issues of cryptographic hash functions:

Table X. SENSITIVITY TESTS RESULTS

Function	***k*(b)**	**Average (%)**	**Minimumm (%)**	**Maximumm (%)**
SHA-512	512	49.79 - 50.23	39.65 - 44.34	55.47 - 60.35
SHA-256	256	49.74 - 50.24	36.72 - 42.19	57.81 - 62.89
MD5	128	49.71 - 50.28	31.25 - 38.28	61.72 - 69.53
MD4	128	49.51 - 50.36	28.13 - 38.28	60.94 - 72.66

1) Hash functions are used in many application.
2) Researchers show great interest in the performance and security analysis of hash functions
3) There are many attacks on hash functions.
4) Hash functions vary in security level against these attacks.

The following table summarizes the most effective attacks on cryptographic hash functions with their complexities.

Table XI. EFFECTIVE ATTACKS ON CURRENT HASH FUNCTIONS (CONCLUSION TABLE)

Hash	O/P Size	Types of Attacks (Best Results)	Year
MD4	128	collision with 2^8 complexity	2005
MD5	128	pseudo-collision with 2^{16} complexity	1993
SHA-0	160	collision with 2^{39} complexity	2005
SHA-1	160	collision (58 round) with 2^{33} complexity	2005
SHA-256	256	Deterministic Collision In 24/64 rounds with $2^{28.5}$ complexity	2008
SHA-512	512	Deterministic Collision In 24/80 rounds with $2^{32.5}$ complexity	2008
Whirlpool	512	collision with 2^{120} complexity	2014

We show next the performance analysis of the hash functions. Unlike the security analysis presented in this section, we base the performance analysis on our own experiments and observations.

IV. PERFORMANCE ANALYSIS

The performance analysis presented here is based on our experiments using OpenSSL libraries. The discussion we made here is based on our observations. OpenSSL library is a general-purpose and open source project that designed for cryptographic research. We use OpenSSL library to measure the performance of some hash functions that are supported by OpenSSL. We performed the experiment in our labs and run it 20 times to calculate the average of performance scores for more accurate results. Table XII shows the system specifications of the machines in our lab where conducted the experiments.

Table XII. SYSTEM SPECIFICATION FOR EXPERIMENTS

Place and Device	(KFUPM) Operating System Lab, Desktop.
Operating System	Cent OS linux-lab10 3.1 0.0-327.28.3. el7.x86_64
CPU	Intel Core 2 Quad CPU Q9400 @ 2.66GHz x86_64
Ram	4 GB
OpenSSL Version	openssl 1.0.0-fips

We apply the performance analysis test on the following six hash functions: MD4, MD5,SHA-1, SHA-256, SHA-512 and Whirlpool. These functions vary in security, and therefore we targeted them to test their performance for better understanding of the trade-off between security and performance. We apply each hash functions on different input sizes and observed the throughput by each function. The throughput of a function

indicate its speed in processing the data and it is defined by the amout of data processed per second measured in MB/Sec unit. Figure 3 shows the results of our experiment. We observed that most of the hash functions get faster with larger input. However, some hash functions perform much better than others as the input size increases. For example, MD4 is much faster than other functions for large input. Then MD5 comes in the second place and SHA-1 comes next. SHA-512 Almost comes in the middle point of performance between hash functions. Whereas SHA-256 and Whirlpool are respectively slow. Table XIII shows these results numerically.

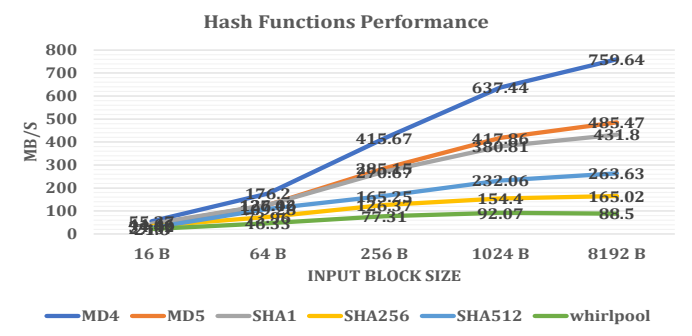

Figure 3. Results of Hash Functions Performance

Table XIII. RESULTS OF HASH FUNCTIONS PERFORMANCE

type/input	16 B	64 B	256 B	1024 B	8192 B
MD4	55.27 MB	176.2 MB	415.67 MB	637.44 MB	759.64 MB
MD5	41.06 MB	126.42 MB	285.15 MB	417.86 MB	485.47 MB
SHA1	44.46 MB	127.06 MB	270.67 MB	380.81 MB	431.8 MB
SHA256	34.42 MB	73.96 MB	126.37 MB	154.4 MB	165.02 MB
SHA512	27.07 MB	109.28 MB	165.25 MB	232.06 MB	263.63 MB
whirlpool	21.6 MB	46.33 MB	77.31 MB	92.07 MB	88.5 MB

V. SENSITIVITY TEST

In this section, we will perform a sensitivity test similar to the one in [15]. The test basically measures the randomness of the hash function. The randomness of a hash function output is a key measurement of its security. To acquire accurate measurements of the randomness of each targeted hash function, an intensive sensitivity test was conducted. The idea here is to see how many output bits may change if a single input bit is changed. Each hash function was tested using the sensitivity test shown in Algorithm 1.

To test a hash function h of output length κ bits, the algorithm generates r random strings of length κ. For each random string w, the hash value h_w is computed. Then a single bit in w is flipped, and a new hash value $h_{\hat{w}}$ is computed. These two hash values are compared to each other and the percentage of their *Hamming distance* to their length is recorded in the sensitivity matrix. The location of the flipped bit varies from 1 to κ for each tested hash function. The algorithm computes and returns the sensitivity matrix, where $Sensitivity[i, j]$ indicates the percentage of the change in the output of hashing the j^{th} random string when a single input bit at location i is flipped. The ideal sensitivity score is 50%.

In our sensitivity test, we run the algorithm for $r = 100$ random strings and calculated the minimum, maximum and

Algorithm 1: Sensitivity Test

Input : hash output length (bits) κ
number of random strings r
Output: sensitivity matrix *Sensitivity*[κ,r]

for $i = 1$ to κ **do**
  for $j = 1$ to r **do**
    $w \leftarrow random\_string(\kappa)$
    $h_w \leftarrow h(w)$
    $\hat{w} \leftarrow bit\_flip(w, i)$
    $h_{\hat{w}} \leftarrow h(\hat{w})$
    $\delta \leftarrow Hamming\_distance(h_w, h_{\hat{w}})$
    $Sensitivity[i, j] \leftarrow (\delta/k) \times 100$
  end for
end for

average sensitivity values for each bit location and for all the five hash functions. The results of the sensitivity test of MD4, MD5, SHA-256 and SHA-512 are given in [15]. The sensitivity results of Whirlpool are given in Figures 4, 5, and 6, which show the average, minimum and maximum sensitivity scores respectively.

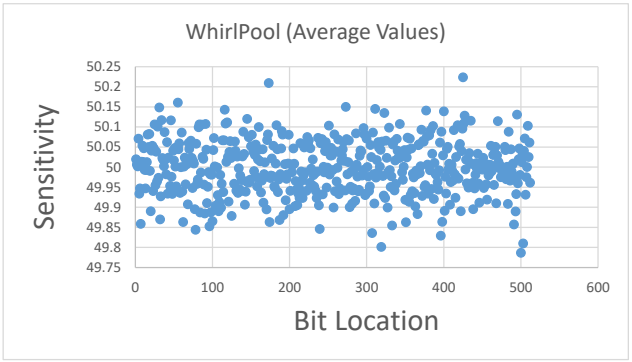

Figure 4. Average Scores of Whirlpool Sensitivity Test

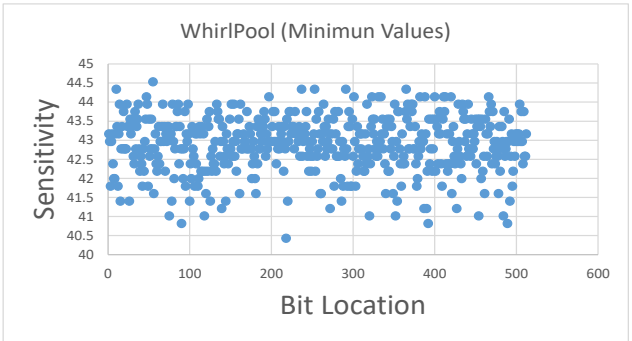

Figure 5. Minimum Scores of Whirlpool Sensitivity Test

Each point (x, y) in Figure 4 represents the average sensitivity score y of the bit at location x computed by Equation 1 below. While the points in Figures 5 and 6 represent the minimum and the maximum scores and are computed by Equations 2 and 3 respectively [15].

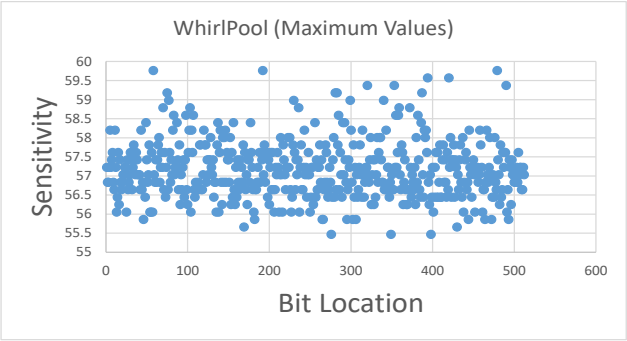

Figure 6. Maximum Scores of Whirlpool Sensitivity Test

$$y = \sum_{j=1}^{r} Sensitivity[x, j]/r \quad (1)$$

$$y = \min_{\forall j}(Sensitivity[x, j]) \quad (2)$$

$$y = \max_{\forall j}(Sensitivity[x, j]) \quad (3)$$

Table XIV shows the numeric sensitivity scores of all the hash functions. We noticed that Whirlpool is slightly more sensitive than SHA-512 and both have better sensitivity scores than other hash functions. There is no significant difference between Whirlpool and SHA-512 sensitivity test results. We found that the Whirlpool and SHA-512 scores are the closest to the ideal sensitivity score (50%), and with relatively low standard deviation compared to other hash functions.

Table XIV. SENSITIVITY TESTS RESULTS

Function	k(b)	Average (%)	Min (%)	Max (%)	Std-Dev
MD4	128	49.51 - 50.36	28.13 - 38.28	60.94 - 72.66	0.154
MD5	128	49.71 - 50.28	31.25 - 38.28	61.72 - 69.53	0.128
SHA-256	256	49.74 - 50.24	36.72 - 42.19	57.81 - 62.89	0.094
SHA-512	512	49.79 - 50.23	39.65 - 44.34	55.47 - 60.35	0.070
Whirlpool	512	49.79 - 50.22	40.43 - 44.53	55.47 - 59.77	0.066

VI. CONCLUSION

Hash functions are considered an efficient tool in many applications, like digital signature, virus detection, intrusion detection, and Blockchain. This paper compares the performance and security of some cryptographic hash functions, namely: MD4, MD5, SHA-1, SHA-256, SHA-512, and Whirlpool. Our study confirms the trade-off between security and performance. It shows that MD4 and MD5 are faster than other hash functions but less secure. It also shows that the most secure hash function with an acceptable performance is SHA-512. The sensitivity test shows that Whirlpool and SHA-512 have the most ideal sensitivity among other hash functions considered in this study.

The results shown in this paper confirm that SHA-512 has suitable performance for mobile devices with limited computational power. Also this function is the most secure for wireless network with best sensitivity ratio. Even for mobile devices with extremely low resource, we suggest looking for an alternative functions of similar structures to SHA-512 but with less rounds, instead of choosing a less secure function (MD4, MD5 or even SHA1). Reducing number of rounds in SHA-512 makes it run faster, consume less energy, but of course with less security. More research in this direction is still recommended for such limited resources devices to find the optimal number for rounds in SHA-512 for each such device.

As for future work, we suggest including SHA-3 in the study. Moreover, instead of using the block size only in our experiment, we may consider more factors in the performance test to study the stability of the performance with the execution time. We may also examine modified versions of SHA-512 on mobile devices with reduced number of rounds, and compare their performance, security, and sensitivity parameters for mobile applications.

ACKNOWLEDGMENT

The authors would like to thank King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, for supporting this research.

REFERENCES

[1] R. Sobti and G. Geetha, "Cryptographic hash functions: a review," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 461–479, 2012.

[2] B. A. Forouzan, *Cryptography and Network Security*. McGraw-Hill Higher Education, 2008.

[3] W. Stallings, *Cryptography and network security: principles and practices*. Pearson Education India, 2006.

[4] N. Kishore and B. Kapoor, "Attacks on and advances in secure hash algorithms," *IAENG International Journal of Computer Science*, vol. 43, no. 3, 2016.

[5] S. Gupta, S. K. Yadav, A. P. Singh, and K. C. Maurya, "A comparative study of secure hash algorithms," in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*. Springer, 2016, pp. 125–133.

[6] P. Rogaway and T. Shrimpton, "Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance," in *International Workshop on Fast Software Encryption*. Springer, 2004, pp. 371–388.

[7] D. Lathwal and P. Khanchi, "Review on md5 hash function," *International Journal of Advanced Research in Engineering Technology and Sciences*, vol. 3, no. 4, 2016.

[8] S. Verma and G. Prajapati, "Robustness and security enhancement of sha with modified message digest and larger bit difference," in *Colossal Data Analysis and Networking (CDAN), Symposium on*. IEEE, 2016, pp. 1–5.

[9] X. Wang and H. Yu, "How to break md5 and other hash functions," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2005, pp. 19–35.

[10] Y. Sasaki, W. Komatsubara, Y. Sakai, L. Wang, M. Iwamoto, K. Sakiyama, and K. Ohta, "Meet-in-the-middle preimage attacks revisited new results on md5 and haval," in *Security and Cryptography (SECRYPT), 2013 International Conference on*. IEEE, 2013, pp. 1–12.

[11] I. Mironov *et al.*, "Hash functions: Theory, attacks, and applications," *Microsoft Research, Silicon Valley Campus. Noviembre de*, 2005.

[12] J. Galbally, I. Coisel, and I. Sanchez, "A new multimodal approach for password strength estimation. part i: Theory and algorithms," *IEEE Transactions on Information Forensics and Security*, 2016.

[13] J. James, R. Karthika, and R. Nandakumar, "Design & characterization of sha 3-256 bit ip core," *Procedia Technology*, vol. 24, pp. 918–924, 2016.

[14] B. Ma, B. Li, R. Hao, and X. Li, "Improved cryptanalysis on reduced-round gost and whirlpool hash function (full version)." *IACR Cryptology ePrint Archive*, vol. 2014, p. 375, 2014.

[15] S. Almuhammadi and A. Amro, "Double-hashing operation mode for encryption." *The 7th IEEE Annual Computing and Communication Workshop and Conference, USA*, 2017.