Journal of Mathematics and Informatics
Vol. 3, 2015, 71-79
ISSN: 2349-0632 (P), 2349-0640 (online)
Published 15 December 2015
www.researchmathsci.org

Journal of
Mathematics and
Informatics

On the SHA-3 Hash Algorithms

***Sukhendu Kuila*[1], *Dipanwita Roy Chowdhury*[2] and *Madhumangal Pal*[1]**

[1]*Department of Applied Mathematics, Vidyasagar University, India,*
E-mail: {babu.sukhendu,mmpalvu}@gmail.com

[2]*Department of Computer Science & Engineering, Kharagpur, India*
E-mail: drc@cse.iitkgp.ernet.in

Received 1 December 2015; accepted 14 December 2015

Abstract. In order to complement the previous hash standards SHA-1, SHA-2, NIST organizes an open competition in search for new hash algorithm to be named SHA-3. After thorough cryptanalysis and performance studies, NIST finally, in 2012, announces Keccak as the winner of the competition and subsequently standardized it as SHA-3. During SHA-3 competition and afterwards, cryptographic community have witnessed many new cryptanalytic techniques. We try to understand them and provide future directions on the cryptanalytic strategies of SHA-3 hash functions.

Keywords: Hash function; Cryptanalysis; Keccak; SHA-3

1. Introduction

In the era of advanced technology, one has to go through applications in daily lives where cryptography plays a crucial role. With the rapid growth of internet facilities and web based facilities, the gap between traditional market place and global electronic market place has been reduced remarkably. Buying and selling or any form of business transactions in which parties interact electronically through a computer mediate network rather than traditional physical exchanges (popularly known as e-commerce) is increasingly become popular over the world. Along with e-commerce, on-line banking, bank cards and credit cards at ATM, mobile communications, e-voting etc. have also emerged to be necessary now-a-days. Information being an association's or each individual's most important assets irrespective of the computing device being used, efforts have been made on providing information security, data integrity as well as confidence on data origin. The fundamental cryptographic algorithm which deals with information security notions like data integrity, authentication, password verification, pseudorandom generator etc. is known as hash function.

Cryptographic hash function is easily computable deterministic mathematical functions which can take arbitrary long input and produce a fixed length output. The first break through in cryptanalytic point of view came from Chabaud and Joux in 1998 when they showed the way to get full collision on SHA0 with query complexity 2^{61} in (1). Over a short period of time Wang et al. published a number of collisions on MD4, MD5 (2), SHA0 (3) with complexity 2^{39} and SHA1 (4). This great work inspired the cryptanalytic community to improve, extend and apply it to other hash functions. The best reported result till date on collision of full SHA1 is 2^{69} given in (4) and practical

collision for 73 steps in (5). With the advent of new cryptanalytic strategies like (6),(7),(8),(9),(10) MD construction strategy gets a serious security threat including SHA0, SHA1. Particularly even if SHA2 remains secure as there is no attack which breaks full SHA2 in practical complexity, similar design strategy makes it less confidence as far as security is concerned. So time has come to make another hash standard to augment SHA2.

In 2007, NIST published a notice (11) for organizing an open competition similar to AES this time on hash function to develop a new standard to be named SHA-3. The submission gets a massive 64 hash algorithms coming from all over the world and a large variety of new design strategies have been emerged.

An enormous amount of feedback coming from all over the world was discussed in public forum. The security proofs, implementation results and cryptanalytic results also have been published. In December 2010, NIST declares 5 finalists of the competition, one from these 5 will be awarded as SHA-3 by 2012. The 5 finalists are BLAKE, JH, KECCAK, GROSTL, and SKEIN. In this time NIST allowed the designers to tweak small changes in their algorithms remembering the cryptanalytic results. It has to be remembered that no hash functions among the finalists have been broken or is in serious security threat till date. At last in October 2012, NIST finally announced Keccak as the winner of the competition realizing its performance besides solid security it provides.

After the selection of the winner of the competition, on May 2014 NIST published a draft FIPS PUB 202 \SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions". In this draft they have encapsulated SHA-3 family which contains four hash functions SHA3-224, SHA3-256, SHA3-384, SHA3-512 together with two Extandable Output Functions (XOFs) called SHAKE128, SHAKE256. They also have categorically mentioned that XOFs' particular applications will be published later.

In this paper, we state the attacks mounted on Keccak so far and provide directions for future work on the extension of cryptanalytic results on Keccak. Rest of the paper is organizes as follows. The Keccak hash function is discussed in brief in Section 2. The cryptanalytic results known so far are detailed in Section 3. Section 4 makes concluding remark and future directions on cryptanalysis of Keccak.

2. Brief note on SHA-3

After the selection of the winner of the competition, on May 2014 NIST published a draft FIPS PUB 202 SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions". In this draft they have encapsulated SHA-3 family which contains four hash functions SHA3-224, SHA3-256, SHA3-384, SHA3-512 together with two Extensible Output Functions (XOFs) called SHAKE128, SHAKE256. They also have categorically mentioned that XOFs' particular applications will be published later. All the SHA-3 functions are instances of Keccak hash function.

Four versions of Keccak| Keccak-512, Keccak-384, Keccak-256, Keccak-224 for target hash length 512, 384, 256, 224 are submitted to SHA-3 competition. The size of the full state for all the versions is set to b = 1600, capacity c is assigned to 2n and r = b-c. The b bit state is treated as 3-dimensional binary matrix where each bit is represented by 3-dimensional coordinates a[x][y][z]--x; y ranges from 0 to 4 and z ranges from 0 to 63. For fixed values of y and z coordinates set of 5 bits is called a row and it is denoted

by a[*][y][z]. When x and z coordinates are kept constant, a column is formed with the 5 bits i.e. a[x][*][z]. For constant x and y coordinates the set of 64 bits a[x][y][*] is called a lane. A slice a[*][*][z] is formed with 25 bits when z coordinate is kept fixed. So there are 320 rows, 320 columns, 25 lanes and 64 slices in a state. Sponge function is used as the domain extension operation of Keccak hash function. Sponge function introduced by Bertoni et al. (12) received immediate attention in the research community for its simple, regular, non-compressive intermediate mode. For clear understanding, here we briefly sketch the sponge function. It uses a fixed internal transformation to be iterated to produce hash value. The internal state is divided into two parts: c-bit capacity and r-bit rate. Initially the state is fed with some fixed value. A suitable padding rule is applied so that the length of the padded message becomes a multiple of r and with the sole constraint that the last message block be never all zero. It is then partitioned into r-bit blocks. These blocks go into the state sequentially by XORing with the rate part. Then a fixed internal transformation (h) is executed over b = (c + r) bit state. This process is known as absorbing phase. Whenever all the message blocks have been absorbed, it starts producing output which is referred as squeezing phase.

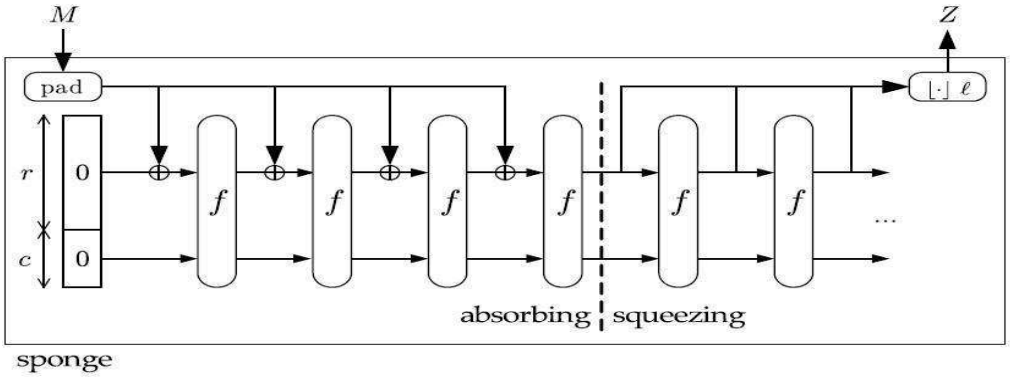

Figure 1: Sponge function

Padding is an important step in the preprocessing phase and must be properly dealt with to avoid attacks like the length-extension. The message to be hashed is first padded and then split into blocks each of length r bits. Initially the state (b bits) is filled with 00s. Message blocks are XORed with the bitrate part of the state interleaved with application of a fixed permutation Keccak-f. When all message blocks are absorbed, first r bit of the state is returned and the permutation f is applied.

Keccak internal permutation consists of 24 rounds and each round consists of 5 operations θ, ρ, π, χ, ι and the default ordering of the operations are as shown below

$$\iota \circ \chi \circ \pi \circ \rho \circ \theta \quad (1)$$

1. The operation θ is linear and it produces diffusion. The parity of two neighboring columns of each bit is added with it. Additions are performed over GF(2).
   θ: $a[x,y,z] = a[x,y,z] + \sum_{y'=0}^{4} a[x-1,y',z] + \sum_{y'=0}^{4} a[x+1,y',z-1]$.
2. The operation ρ translates each lane by some predetermined value T(x,y) called ρ-offset

ρ: a[x, y, z] = a[x, y, T(x, y)]

3. Another linear operation π is a permutation on slice which is defined as follows

π: a[x, y, z] = a[x', y', z]

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

4. The operation χ is the non-linear operation and it operates on each of 320 rows independently and parallel. The operation is defined as follows

χ: a[x, y, z] $= a[x, y, z] + ((\neg\, a[x+1, y, z]) \wedge a[x+2, y, z])$.

If a single bit is changed in the input at most 2 bits are changed at the output and hence it produces slow diffusion.

5. Here 64 bit pre-determined round constant is added with the first lane of the state.

3. Cryptanalysis of reduced round Keccak

The first ever published cryptanalytic result on Keccak is reported in (13). Two known methods-triangulation algorithm, cube tester have been used separately to distinguish the Keccak internal permutation. Triangulation algorithm has been used to sort out free variables when some specific input bits as well as output bits are kept fixed. The presence of such free variables in constrained- input-constraint-output environment is utilized to exhibit the non-randomness of the Keccak internal permutation. Another distinguisher has been developed in (13) based on cube tester which exploits low algebraic degree of the round function. Practical preimage attack has been reported in (14) with the help of cube attack. In (15), a new kind of distinguishing strategy called zero-sum distinguisher has been introduced and applied to Keccak internal permutation. It tries to build structures which sum to zero as well as their images also sums to zero. Starting from the middle, the idea is to apply higher order derivative to exploit low algebraic degree of round function and its inverse. Distinguisher up to 16 rounds of Keccak-f has been reported here. In the work (16) the authors extend the basic zero-sum structure and mount distinguisher which can penetrate up to 18 rounds. The improvement arises by making tighter bound on algebraic degree and exploiting a special property of χ operation. Suppose the degree of the forward round is d. Construct subspace V of dimension d by fixing$(320 - \lfloor \frac{d}{5} \rfloor)$ rows to arbitrary value. Since χ acts row wise, for every constant a, there exists b for which following holds$\chi(a + V) = b + V$. The authors call this property as multi set property. The authors in (17) extend their previous work to obtain multi set property for two consecutive rounds. When applied to Keccak permutation, it has been shown the attack can penetrate up to 20 rounds.

Preimage attack up to 3 rounds of Keccak hash function have been found in (18) experimentally with the help of SAT-solver. In the work (19), the authors analyze iterated permutation with regard to its algebraic degree. It has been noticed that the algebraic degree, for larger number of rounds, does not increase exponentially to number of rounds. This novel observation leads to construct zero-sum structure for full 24 rounds. While the previous work requires a heavy complexity of about 2^{1590}, here the work (20) intends to reduce the work load by exploiting a property of χ^{-1}. It has been shown that the algebraic degree of the product of two output coordinates of χ^{-1} does not increase optimally. By lowering the inverse permutation degree significantly, it becomes possible

to construct zero-sum structure for full Keccak in 2^{1575} calls to the permutation. In (21), a second preimage attack is mounted which requires large memory and hence becomes inefficient.

New strategies have been developed in (22) to construct low weight differential path. Based on this path, distinguisher on Keccak hash function reduced to 4 rounds has been devised. Preimage and collision have been found for 2 rounds of Keccak while near collision attack has been devised for 3 round. In the work (23), exploiting algebraic properties, authors developed a novel algorithm named Target Difference Algorithm.

Reference	Type	Exploiting Property	Rounds	Version	Complexity
(21)	Second Preimage	Polynomial enumeration	6/7/8	512	$2^{506}/2^{507}$ $/2^{511.4}$
(22)	Second Preimage	Differential path	2	256	2^{33}
(18)	Preimage	SAT solver	3	512	2^{506}
(24)	Preimage	Rotational	4	512	2^{506}
(22)	Collision	Differential path	2		2^{33}
(25)	Collision	Internal differential	3	384	Practical
(23)	Collision	Algebraic and differential	4	224/256	Practical
(25)	Collision	Internal differential	5	224/256	2^{115}
(23)	Near collision	Algebraic and differential	5	224/256	Practical
(22)	Near collision	Differential path	3		2^{25}
(26)	Key recovery	$Cube\ attack^b$	5/6/7	512^a	2^{128}
(26)	Forgery	$Cube\ attack^b$	7	512^a	2^{128}
(26)	Forgery	$Cube\ attack^b$	8	512^a	2^{256}

[a]r = 1024, c = 576
[b]Keyed Keccak

Table 1: Summary of core attacks on SHA-3

The algorithm takes arbitrary state difference as input and produces messages satisfying initial constraint together with given target difference after one round. Though the algorithm is heuristic, it has been claimed that most of the time it exhibits desired solution. Utilizing this algorithm in the conjunction of differential path, practical collisions have been found for Keccak-224, Keccak-256 reduced to 4 rounds. In [24] new techniques developed to construct differential paths extend up to 5 rounds of Keccak permutation. The strategy of rebound attack has been applied with the aid of newly constructed path. As a result several distinguishing attacks on round reduced internal

permutation have been reported. The attack is able to distinguish Keccak permutation reduced to 8 rounds in query complexity $2^{491.47}$. Rotational cryptanalysis technique is applied to cryptanalyze Keccak hash function in [25]. A statistical distinguisher on 5 round Keccak-f is reported which work with query complexity 2. Later a preimage attack on 4 round Keccak with complexity 2 is devised. In [26], internal difference, a new concept of difference is introduced. While classical difference considers difference between a pair states, internal difference considers difference between 2 or more parts of a state. Subspace cryptanalysis in the context of internal differential path yields collision on Keccak hash function reduced up to 5 rounds. In [27], the authors study the propagation of difference and looks for biased output bits. A distinguisher for Keccak hash function reduced to 6 rounds is reported which work with complexity 252. Boomerang technique is used to devise distinguisher on Keccak internal permutation. Unlike classical boomerang attack, here in [28], it uses internal differential path along with classical differential path. When applied to Keccak internal permutation reduced to 7, 8 rounds, it becomes possible with complexity 212; 218 respectively. It is important to note that for 8 round distinguisher, the attack starts from round 4 and attacking strategy become inefficient for the permutation starting from round 1. Recently Dinur et al. [29] applies cube attack on keyed mode of round reduces Keccak sponge function. The attack successfully recovers key in practical complexity to Keccak reduced to 6 rounds when working in MAC mode as well as in stream cipher mode. All the cryptanalytic results published till date can be summarized in *Table 1* and *Table 2*.

Reference	Exploiting property	Rounds	Version	complexity
(20)	Zero sum	24	Permutation	2^{1579}
(22)	Differential path	4	Permutation	2^{36}
(22)	Differential path	4	Hash function	2^{24}
(27)	Differential path	6	Hash function	2^{52}
(24)	Rotational symmetry	4	Permutation	$2^{8.6}$
(28)	Differential path	4	Permutation	2^{2}
(29)	Self symmetry	4	Permutation	Single query
(17)	Zero sum	7	Permutation	2^{20}
(30)	Internal differential Boomerang	7	Permutation	2^{13}
(30)	Internal differential Boomerang	7^{w}	Permutation	2^{10}
(17)	Zero sum	8	permutation	2^{30}

[w]starts from round 3

Table 2: Distinguishing attack on Keccak

One of the most notable work invalidating core security properties of reduced round version Keccak hash function is reported in [26]. The work successfully finds collision for Keccak-512 reduced to 3 rounds and collisions up to 5 rounds for Keccak 224/256. The attack methodology relies on subset cryptanalysis through which a larger set of inputs are forced to map on a smaller set of outputs with high probability. The methodology is popularly known as squeeze attack. It first construct internal differential characteristic and then link the characteristic to the initial state of the Keccak hash function by the application of an efficient message modification technique TIDA. The method is analogous to Target Difference Algorithm (TDA developed in [23]) which operates on classical difference rather than internal difference environment. The internal difference of initial state of characteristic is known as target difference and the challenge of TIDA is to obtain many single block messages which conform to the target difference after one Keccak permutation. However, as clearly specified in [23], the method is not deterministic and is not applicable to find collision for the largest variant of Keccak hash function [26].

4. Conclusion and future work

In this paper we have reviewed the cryptanalytic results on Keccak reported so far. Despite best efforts, Keccak has shown great strength against all classical and state-of-the-art cryptanalytic techniques. The most effected technique as far as number of attacking rounds are concerned, is prove to be zero-sum distinguishing technique which exploits the low algebraic degree of the round function of Keccak permutation. However it is not known how to mount actual attack against Keccak hash function utilizing the property of internal permutation. On the other hand, the attack which invalidates the core security up to 5 rounds of Keccak hash function is executed by squeeze attack as reported in [26]. The inapplicability of message modification technique TIDA limits the attack to go beyond 3 rounds for the largest variant of Keccak. Moreover, TIDA is heuristic and its success cannot be proved formally. An interesting future work is to construct an efficient, deterministic message modification technique to link internal differential characteristic is required to attack largest variant of Keccak for larger number of rounds.

REFERENCES

1. A.Joux and F.Chabaud, Differential Collisions in SHA-0, In CRYPTO'89, London, UK, Springer,
2. H.Yu and X.Wang, How to break MD5 and other hash functions, Advances in Cryptology-EUROCRYPT 2005, Springer
3. X.Wang, H.Yu and Y.Yin, Efficient collision search attacks on SHA-0. Santa Barbara, California, USA : Springer. In Advances in Cryptology - CRYPTO 2005.
4. X.Wang, Y.Yin and H.Yu, Finding collisions in the full SHA-1, In Proceedings of the 25th Annual International Conference on Advances in Cryptology, CRYPTO'05, Berlin, Heidelberg, Springer, 2005.
5. A.Grechnikov and A.Evgeny, Collisions for 72-step and 73-step sha-1:Improvements in the method of characteristics, IACR Cryptology ePrint Archive, 2010:413, 2010.

6. A.Joux, Multicollisions in iterated hash functions, application to cascaded Constructions, Santa Barbara, California, USA, Springer, 2004, In Advances in Cryptology -CRYPTO 2004.
7. P.Sarkar and S.Sanadhya, New collision attacks against up to 24-step SHA-2. INDOCRYPT, Vol. 5365, *Lecture Notes in Computer Science*, Kharagpur, India, Springer, 2008.
8. E.Biham, R.Chen and A.Joux, Cryptanalysis of SHA-0 and reduced SHA-1, *J. Cryptology*, 28(1) (2015) 110-160.
9. B.Schneier and J.Kelsey, Second preimages on n-bit hash functions for much less than 2n work, In Advances in Cryptology– EUROCRYPT 2005, Aarhus, Denmark, Springer.
10. T.Kohno and J.Kelsey, Herding hash functions and the nostradamus attack, In Proceedings of EUROCRYPT'06. Berlin, Heidelberg, Springer, 2006.
11. National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Supersedes FIPS PUB202,., 2015.
12. G.Bertoni, J.Daemen, M.Peeters and G.Van Assche, Sponge functions. Ecrypt Hash Workshop, (2007).
13. D.Khovratovich and J.P.Aumasson, First Analysis of Keccak. comment on the NIST Hash Competition 2009.
14. J.Lathrop, Cube attacks on cryptographic hash functions, Available at http:// www.cs.rit.edu/ jal6806/thesis/., 2009.
15. A.Canteaut and C.Boura, A zero-sum property for the keccak-f permutation with 18 rounds, NIST mailing list.
16. W.Meier and J.P.Aumasson, Zero-sum distinguishers for reduced keccak-f and for the core functions of luffa, NIST mailing list.
17. C.Boura and A.Canteau, Zero-sum distinguishers for iterated permutations and application to keccak-f and hamsi-256, SAC 2010, Springer.
18. M.Srebrny and P.Morawiecki, A sat-based preimage analysis of reduced keccak hash functions, *Inf. Process. Lett.,* 113(2010-11)392-397.
19. C.Boura, A.Canteaut and C.De Canniere, Higher-order dierential properties of keccak and luffa, Cryptology ePrint Archive, Report 2010/589.
20. X.Lai, M.Duan, Improved zero-sum distinguisher for full round keccak-f permutation, *Chinese Science Bulletin,* 57(6) (2012) 694-697.
21. D.J.Bernstein. Second preimages for 6 (7? (8??)) rounds of Keccak? Available at http://ehash.iaik.tugraz.at/uploads/6/65/NIST-mailinglist.
22. M. N.Plasencia, A. Rock and W.Meier, Practical analysis of reduced-round keccak, In Progress in Cryptology- INDOCRYPT 2011, Chennai, India, Springer.
23. I.Dinur, O.Dunkelman, and A.Shamir. Improved practical attacks on round reduced Keccak. *Journal of Cryptology*, 27(2) (2014) 183-209.
24. P.Morawiecki, J.Pieprzyk, and M.Srebrny. Rotational cryptanalysis of round-reduced Keccak. Cryptology ePrint Archive, Report 2012/546.
25. I.Dinur, O.Dunkelman, and A.Shamir. Collision Attacks on Up to 5 Rounds of SHA-3 Using Generalized Internal Differentials. http://eprint.iacr.org/., 2012.
26. I.Dinur, P. Morawiecki, J.Pieprzyk, M. Srebrny and M.Straus, Cube attacks and cube-attack-like cryptanalysis on the round-reduced keccak sponge function. In Advances in Cryptology - EUROCRYPT 2015, Sofia, Bulgaria, Springer, 2015.

27. W.Meier and S.Das, Marrakesh, Differential biases in reduced-round keccak, In Progress in Cryptology - AFRICACRYPT 2014, Morocco, Springer, 2014.
28. A.Duc, J.Guo, T.Peyrin and L.Wei, Unaligned rebound attack: application to Keccak. *In Fast Software Encryption*, Springer, 2012.
29. S.Kuila, D.Saha, M.Pal and D.Roy Chowdhury, Practical distinguishers against 6-round keccak-f exploiting self-symmetry, In Progress in Cryptology-AFRICACRYPT 2014, Marrakesh, Morocco, Springer, 2014, pp. 88-108.
30. I.Nikolic and J.Jean, Internal differential boomerangs: Practical analysis of the round-reduced keccak-f permutation, *In Fast Software Encryption FSE-2015*, Istanbul, Turkey, Springer, 2015.
31. A.Joux and F.Chabaud, Differential Collision on SHA-0. In CRYPTO'98, London, UK, Springer-Verlag, 1998.
32. M.Srebrny and P.Morawiecki, A sat-based preimage analysis of reduced keccak hash functions, *IACR Cryptology ePrint Archive*, 2010:285.1
33. S.Kuila, D.Saha, M.Pal and D.Roy Chowdhury, CASH: Cellular automata based parameterized hash, *In Security, Privacy, and Applied Cryptography Engineering*, Vol. 8804, *Lecture Notes in Computer Science*, pp. 59–75, Springer.
34. D.Saha, S.Kuila and D.R.Chowdhury, Escape: Diagonal fault analysis of APE. In Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, 2014, Vol. 8885, *Lecture Notes in Computer Science*, pp. 197–216, Springer