Journal of Management and Social Studies

Journal of Management and Social Studies

ISSN 2383-2320

Volume 1, Issue 12, December 2023, pp. 14-29

homepages: http://jmssjournal.com/

Double Length Sponge Construction DLP

Baraa Tareq Hammad[1*], Norziana Jamil2, Muhammad Reza Z'aba[3], Mohd Ezanee Rusli[4], Ismail Taha Ahmed[5]

[1] University of Tenaga Nasional, Jalan IKRAM-, Kajang, Selangor, Malaysia.

[2] University of Tenaga Nasional, Jalan IKRAM-, Kajang, Selangor, Malaysia.

[3] MIMOS Berhad, Technology Park Malaysia, Bukit Jalil, Kuala Lumpur, Malaysia.

[4] University of Tenaga Nasional, Jalan IKRAM-, Kajang, Selangor, Malaysia.

[5] University of Tenaga Nasional, Jalan IKRAM-, Kajang, Selangor, Malaysia.

*Corresponding author's E-mail: *Omrami82@yahoo.com*

ABSTRACT

In this paper, a new sponge construction called DLP sponge is developed, which takes an arbitrary length of input and yields an output of random length. It can easily be proven that the construction is resistant against generic attacks when the capacity c = n. Not only that, it is flexible enough that it can also be used to build other cryptographic primitives such as block ciphers, Message Authentication Codes (MACs) and hash functions. This construction is also suitable for lightweight cryptography because it solves problem of capacity c, the hash digest n and the small key k.

Keywords: sponge construction, DLP sponge, hash function, double length, authenticated encryption, duplex construction.

1. Introduction

A cryptographic hash function is a function that takes an arbitrary-length input and produces a fixed-length output called a digest, without any secret parameters. It typically consists of two parts, i.e. internal part known as compressionfunction f (which is iterated sufficiently many times) and external part knownas hash construction such as Merkle-Damg°ard(Diffie, 1976) and sponge (Bertoni, 2007), to name a few.

It looks impossible to have an iterated hash function behaving like a randomoracle. Similar shortcoming can be observed in sponge construction due to involved inner collision. Some researchers have reported that sponge constructionshows a weakness when $c = n$(Bertoni, 2011b,Alahmad, 2013, Hammad 2016). So, in this paper, a new sponge construction is proposed by creating two permutation lines run in parallel wherethe length of input in every permutation line is doubled from n to $2n$. We referthis construction as DLP sponge, and the security of this construction can bereduced to the security of sponge construction, i.e.:

1. Collision-resistance: It is difficult to find any two different inputs m_0 and $m_1$such that $H(m_0) = H(m_1)$ and this requires at least $2^{c/2}$ work.
2. Preimage-resistance: Given a hash value $H(m)$, it is difficult to find m andthis requires at least 2^cwork.
3. Second preimage-resistance: Given an inputm_0, it is difficult to find a different input m_1 such that $H(m_0) = H(m_1)$, and this requires at least 2^cwork.

Examples of hash function designs based on the sponge construction areKeccak (Bertoni, 2009), PHOTON (Guo, 2011), Spongent (Bogdanov, 2011), quark (Aumasson, 2010) and LHash (Wu, 2013). All theseprimitive designs are based on permutations of many variants, and most of themare designed for constrained devices with $c = n$.

In order to have a more secure and efficient hash function, a double length (from single length) construction has been considered. Hirose (Hirose, 2004, Hirose, 2006) proposeda Double Block Length Construction with two different block ciphers, and the collision resistance of this construction is $2^{n/2}$. Nandi (Nandi, 2005) proposed a 2/3-ratedouble length compression function, and this double length construction takes $n$inputs and produces $2n$ outputs. Nandi (Nandi, 2005) prove that the complexity of collision attack is $2^{2n/3}$, which is more secure than the Merkle-Damgard construction.

It is important to note that these constructions were designed based onMerkle-Damg°ard, which takes n bit input and produces $2n$ output, by using twodifferent compression functions. On the other hand, our construction is basedon sponge construction which takes an arbitrary input and produces a variablelength output using the same compression function without the need of extrahardware.

In this paper, general descriptions on sponge (Bertoni, 2011b) and duplex construction(Bertoni, 2011a) are firstly provided in Chapter 2. Next, our construction called DLP spongeis explained in Chapter 3. We also elaborate in this chapter how this construction can be used as Authenticated Encryption (AE), Block Cipher, Message Authentication Code MAC and

HashFunction. The security analysis of DLP sponge is given in Chapter 4 and weconclude our work in Chapter 5.

2. Literature Review

The Sponge Construction

Sponge construction is an iterative construction designed by Bertoni, G. et.al.((Bertoni, 2007, Bertoni, 2011b)that maps a variable length input to a variable length output. Thisfeature renders this construction suitable for many applications such as hashfunction, stream cipher, mask generation function, and Message AuthenticationCode (MAC) (Borowski, 2013). The length state $b = r+c$ is fixed, where r denotes the bitrateand c represents the capacity determined through a function f that generates atransformation or permutation of b. Sponge construction operates in two phasesas shown in Figure1.

- The absorbing phase: - First, message M is padded with 1 followed by many0s to make total length a multiple of r. Then, the message is divided into$r$-bit blocks. Each input block m_i is XORed with the r part of internal state$S$. Finally, process S is iterated until all blocks are exhausted. The absorbingphase is denoted by $S = S_f\,(S_r \oplus m_i, S_c)$.
- The squeezing phase: The state progresses according to S_f; however, ther$r$ parts of the states are returned at each iteration as output blocks in thisinstance. The size of the final output is determined by the user. The squeezingphase is denoted by $z_j = S_r$; and $S = S_f(S)$.

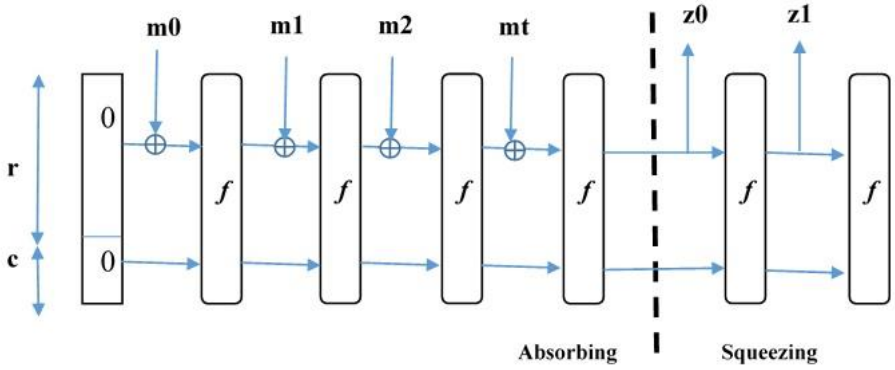

Fig. 1. SpongeConstruction.

Algorithm 1:- The sponge construction sponge [*f*; *pad*,*r*]

Require: $r<b$
Interface Z= sponge (M, ℓ) with $M \in Z^2$, integer $\ell > 0$ and $Z \in Z^2$.
$m = M$|| pad[r] ([M]).
Let $M = m_0||m_1||\ldots||m_w$ with $[m_i] = r$.
$s = 0^b$.
for $i = 0$ **to** w **do**
$s = s \oplus (m_i || 0^{(b-r)}$
$s = f(s)$
end for
$Z = [s]^r$
While $[z] < \ell$ **do**
$s = f(s)$
$Z = Z \,||[s]_r$
end While
return $[Z]^\ell$

Duplex Sponge

This construction is very similar to the design of the sponge construction. Themain difference between the two designs is that in the former, there is no squeezing phase before the final digest is produced (Bertoni, 2011a).

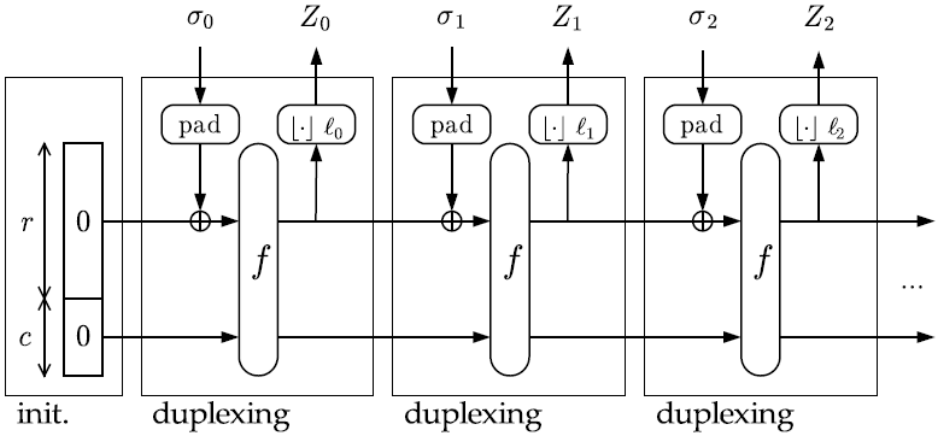

Fig. 2. The duplex construction (Bertoni, 2011a).

SpongeWrap

The SpongeWrap construction (Bertoni, 2011a) is designed for authenticated encryption as shown in Figure 3. First, the key *k* is initialized and loaded into the state. Next,the header *A*padded and absorbed into the state. The message *M* padded and divided into *p* blocks, after that the encryption(or decryption) runs in duplex mode. as shown in Figure 3. The resulted tag *T* compared with the recited tag to check the validity of this tag (Yalçın, 2012).

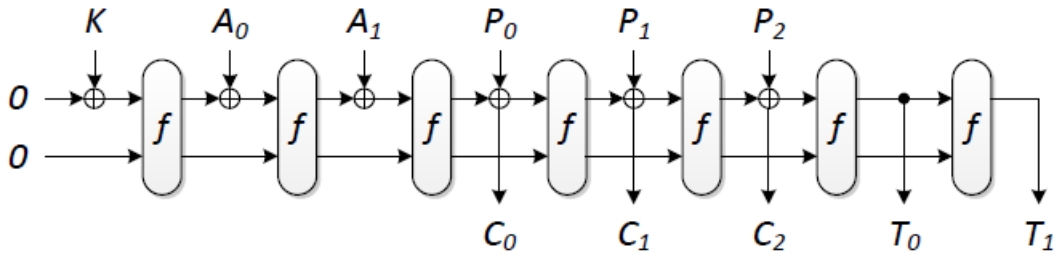

Fig. 3. SpongeWrap authenticated encryption (Yalçın, 2012).

DLP Sponge Construction

In this section, a new construction based on sponge construction is proposedi.e DLP sponge. In DLP sponge construction, a problem of small keys and theresistance against generic attacks of sponge construction when the capacity *c* equals to the output *n* is addressed. Most of the previous work show that spongeconstruction is similar to random oracle when the capacity $c \geq 2n$(Bertoni, 2011,Thomsen, 2005,Bertoni, 2008).

The DLP construction uses two *b*-bit compression functions *f*. Two chains arekept same during the absorbing process. Subsequently, in the squeezing phase,the compression function *f* takes input from the two chains as *r* and *c* in orderto produce the output as shown in Figure 4. The attractive features inherited inthe double length and the wide pipe construction serve as the building block ofour new design.

In the absorbing phase, *M* is divided into *m* blocks (=2*r*). Then, each blockis padded by zeros followed by 1. Here, the inputs of *f* and *f`* are $m_i$ and $m`_i$,respectively. It is important to note that the use of same *m* ensures that the twocompression functions *f* and *f`* in producing the same value. The decision ofusing the same/different compression functions *f*; *f`* is determined by the user, and*m*satisfies the following properties:

1. It is easy to compute both *m* and m^{-1}.
2. It has no fixed points.

The DLP sponge construction phase:

1- First: - The message *M* is padded by ‘1’ bit followed by the least number of `0' bits to make the length of padded *M* a multiple of *r*-bit.
2- Initialization the capacity *c* and the bitrate *r* with zeros.
3- Divide *M* into *r*-bit blocks and processed sequentially.
4- The absorbing phase: Each input block m_i is XORed with the *r* part of internal state *S*. Finally, process *S* is iterated until all blocks are exhausted. The absorbing phase is denoted by

$s = s \oplus (m_i || 0^{(b-r)}$

$s` = s` \oplus (m`_i || 0^{(b-r)}$

$s = f(s)$

$s` = f`(s`)$

S= s||s`

5- The squeezing phase: The state progresses according to S_f; however, the r parts of the states are returned at each iteration as output blocks in this instance. The size of the final output is determined by the user. The squeezing phase is denoted by $Z = Z \,||\, [S]_r$.

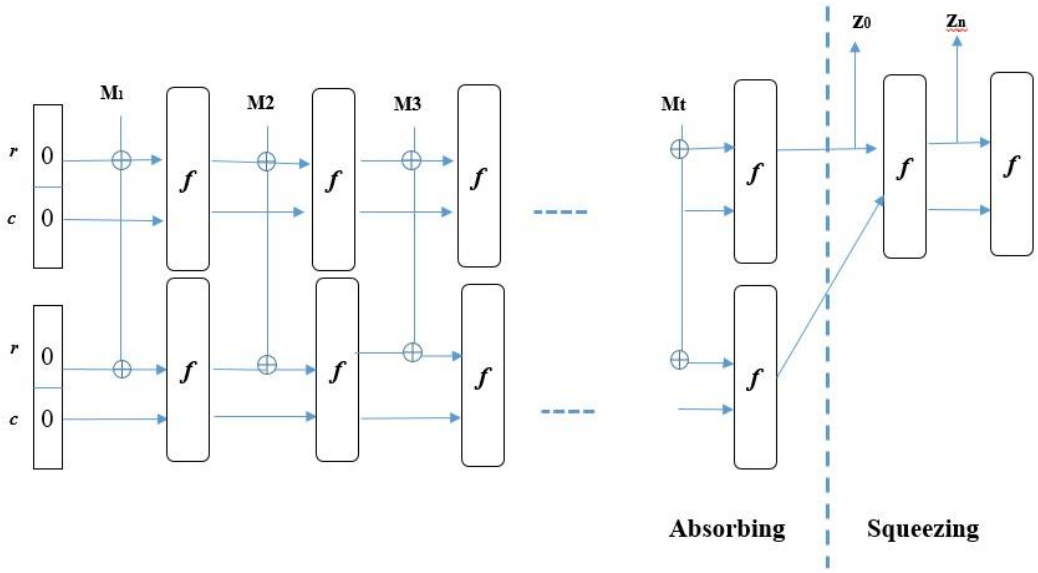

Fig. 4. DLP sponge construction

In the keyed sponge, the indifferentiability bound suggests that the size ofkey k is twice of the capacity c(Bertoni, 2011). With the underlying permutation $b = c+r$,decreasing capacity will affect the bitrate, Therefore, this motivates us to proposea new construction where $k = 2c$ without the effects of the size of b. Thismakes DLP sponge construction to be flexible for many applications as shownin Section 3.1.

Algorithm 2:- DLP sponge construction DLPS[*f*,*pad*,*r*]

Require: $r<b$
Interface Z= DBLS (M, ℓ) with $M \in Z^2$, integer $\ell > 0$ and $Z \in Z^2$.
$m = M$|| pad[r] ([M]).
Let $M = m_0||m_1||\ldots||m_w$ with [m_i] = r.
$s = 0^b$.
for i = 0 **to** w **do**
$s = s \oplus (m_i || 0^{(b-r)}$
$s` = s` \oplus (m`_i || 0^{(b-r)}$
$s = f(s)$
$s` = f`(s`)$
S= s||s`
end for
$Z = [S]^r$
While [z] <ℓ **do**
$s = f(S)$
$Z = Z \,|||[S]_r$
end While
return $[Z]^{\ell}$

DLP Sponge as Authenticated Encryption (AE)

Authenticated Encryption is a technique which combines both authenticationand encryption in order to provide integrity and confidentiality. A sponge function, SpongeWrap mode (Bertoni, 2011a)has also been used for AE (Aumasson, 2012). Another constructionutilizing AE has been proposed such as Monkey sponge and Donkey sponge(Bertoni, 2012).Both constructions have been designed for lightweight cryptography primitives(Bertoni, 2012).

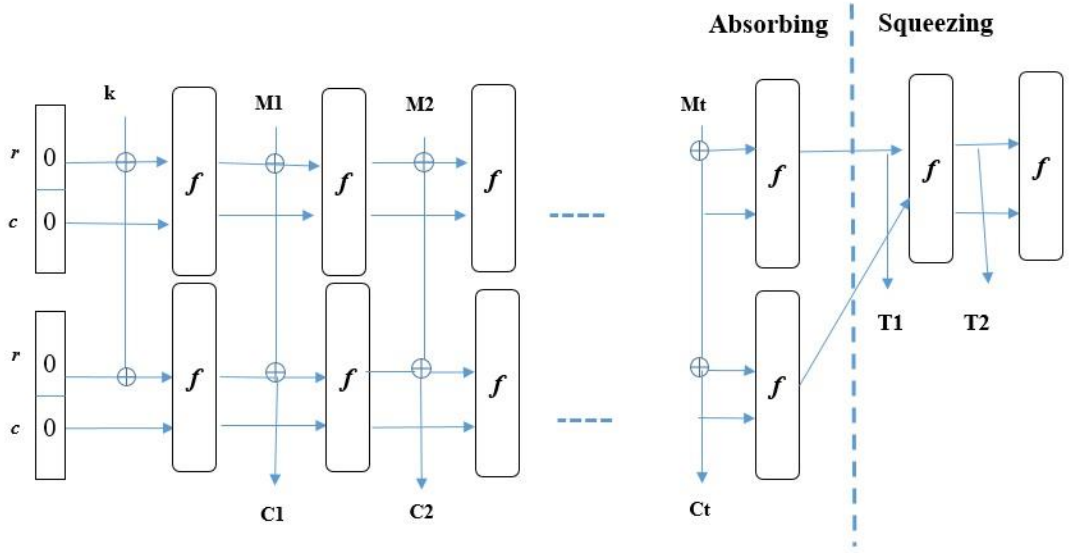

Fig. 5. DLP sponge construction as authenticated encryption

In order for DLP sponge to serve as a block cipher, duplex construction (Bertoni, 2011a)is firstly performed, followed by the tag production. The challenge is how to dealwith small keys, and how and where to use these keys.

In principle, encryption and authentication are not performed concurrentlyin resource constrained devices due to processing power capability and memoryconstraints. Therefore, authenticated encryption is usually adopted, where thesame block cipher performs both functions (Yalçın, 2012).

Encryption *E*= $Z_2^k \times (Z_2^*) \rightarrow Z_2^* \times Z_2^t$: $(K, H, M) \rightarrow (C, T)$

Decryption *D*=$Z_2^k \times (Z_2^*) \times Z_2^t \rightarrow Z_2^* \cup \{error\}$: $(K, H, C, T) \rightarrow M \; or \; error$

Given two inputs *M* and *M`* produce the same output, cause an issue, so,the header *H* added, it can be expanding by zeros.

DLP Sponge as a Block Cipher

Block cipher provides inverse function (encryption and decryption) of data, usesdifferent modes of operation like Offset Code Book mode (OCB), Cipher BlockChaining (CBC) and Electronic Code Book (ECB). In this mode, a key *k* and amessage *M* are used as input to produce a cipher *C* that has a same size as *M*,as shown in Figure 6.

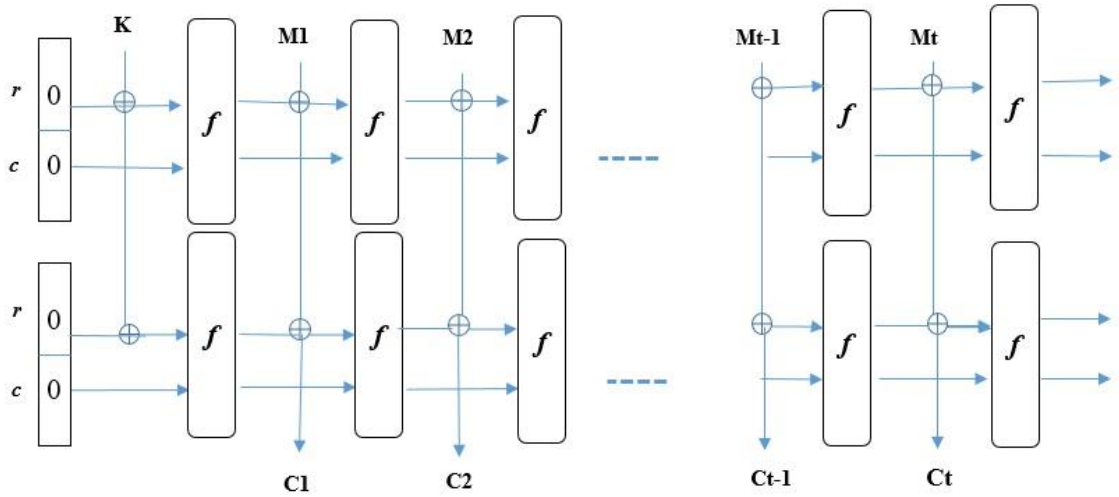

Fig. 6. DLP sponge construction as block cipher

DLP Sponge as a the Message Authenticated Code (MAC)

MAC is one of the most important applications of hash function, which is usedto validate the integrity and authenticity of information communicated over aninsecure channel. A sender uses a secret message as an input to MAC functionin order to produce a tag that will be sent to a receiver. Then, the receiver usesthe same MAC function, message and key to produce a tag. The results are thencompared as shown in Figure 7.

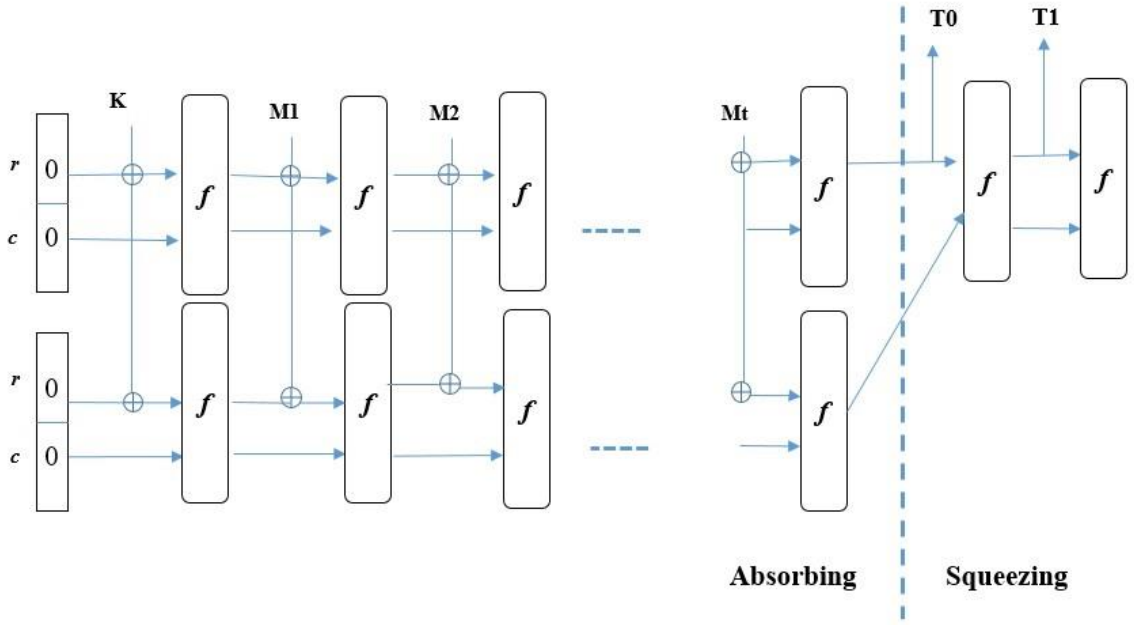

Fig. 7. DLP sponge construction as MAC

3. Methodology

When a new cryptographic hash function is designed, one chooses a constructionand a compression function to build a new function with arbitrary input and output sizes (Bertoni, 2011a). When the security of this construction is proven, these primitives are guaranteed to be secure against generic attacks such as multicollision(Joux, 2004), herding (Kelsey, 2006), second pre-image (Kelsey, 2005), faster multicollision (Aumasson, 2008) and length extension attacks (Gligoroski, 2010).

Collision resistance is the main security necessity for hash functions. Thebirthday paradox theory shows that finding two persons with the same birthday is similar to finding two messages that yield to the same hash digest. Thebirthday attack can occur in all compression functions in iterated hash functionswith complexity ($2^{n/2}$). Therefore, the construction of a hash function can alsobe attacked. An attack on the construction function that does not significantlyaffect the compression function is called a generic attack. This type of attack isbased on several parameters, such as hash size and internal state.

The security of sponge construction mainly depends on capacity *c*(Bertoni, 2007,Bertoni, 2011).Therefore, the attacks focus more on capacity *c* than output *n*. When the capacity *c* is small, the collision is limited to *c* with a complexity of $2^{c/2}$. However,this scenario is not true in lightweight cryptography where 2^n is costly in termsof area, such as in PHOTON (Guo, 2011), quark (Aumasson, 2010) and LHash (Wu, 2013).

In order to study the security of DLP sponge construction, birthday attackis analyzed against. This attack can be applied to any iterated hash functionwhich requires $2^{n/2}$ complexity, while in sponge construction it requires $2^{c/2}$. Itis more susceptible to an attack if *c* is small. Therefore, DLP doubles the size of *c*i.e. doubles the sponge length, to provide better security.

The two main concepts in DLP sponge construction are state collision and inner collisions as shown in Figure 8. A state collision involves a pair of differentmessages $M \neq M`$ under the same state $S_f\ [M] = S_f\ [M`]$. The state collisionsobtained during the absorbing phase may lead to identical hash function values $S_f\ [M] = S_f\ [M`]$. The squeezing phase produces the same output values$S_f\ [M\ |0^j\ ] = S_f\ [M`\ |0^j\ ]$ for all j.

An inner collision involves a pair of two different messages$M \neq M`$ underthe same inner state $S_{c,f}\ [M] = S_{c,f}\ [M`]$. If a state collision on $M \neq M$ exists,then an inner collision on $M \neq M`$ exists as well. However, the reverse is nottrue. The state collision for $M \neq M`$ can be generated through inner collisionssuch that $S_{c,\,f}\ [M] = S_{c,\,f}\ [M`]$.

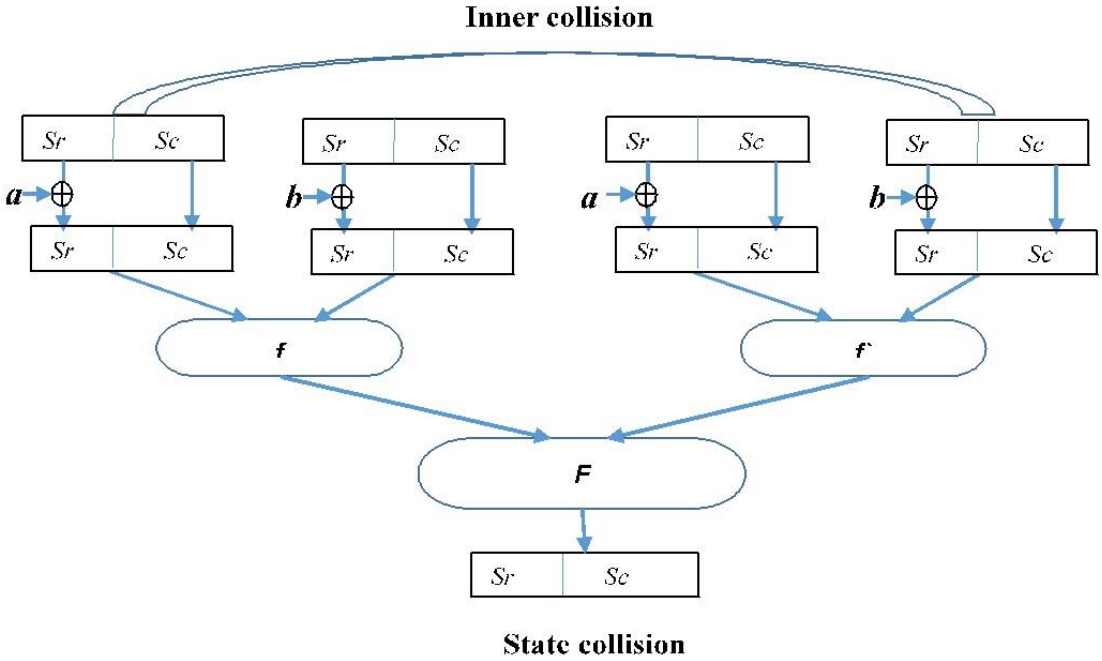

Fig. 8. Inner collision and State collision in DLP sponge

4. Finding

Joux (Joux, 2004) launched a multicollision attackon an iterated hash function. Specifically, this researcher performed an effectivegeneric multicollision attack on the Merkle-Damg°ard hash function and reportedthat the process of determining the multicollisions of messages that yield to thesame hash digests is no more difficult than that of identifying ordinary collisions.The message blocks are larger than the hash and the chaining value assuming$m < n$.

The multicollision attack on DLP sponge construction is not easy as on classical sponge, the concatenation of two compression function f and $f`$, $F = f(H) \| f`(H) = n(i + 1), i > 2$, else the compression function will not work onthe input, so. The complexity of birthday attack will be (2^c) instead of $2^{c/2}$.

The compression function f take input as $f = \{0, 1\}^b \times \{0, 1\}^b \rightarrow \{0, 1\}^n$, assuming that f are collision resistance, we spouse that we have an inner collision$p$, q and v, w, than we have an output collision $p|a|m$, $q|b|m$ and $v|a|m$, $w|b|m$,leads to an output collision with complexity $2^{2(c+3)/2}$. The complexity of Jouxmulticollision attack is $t.2^{c/2}$ when $c \geq 2n$. In Joux attack (Joux, 2004) the attacker presumably utilizes a collision-finding algorithm called machine C, which inducescollisions for compression function f with each call. This machine can apply anycollision finding attack, such as the birthday or brute force attack.

To launch Jouxs attack on DLP sponge construction. We use the birthdayattack as the collision-finding. The attack has two phases. First, birthday attack1(BD1) is employed for messages M_1, $M`_1$, $M_{11}$ and $M`_{11}$, and the initial value$W_1 = 0$is set. This value is inputted into f and $f1$ along with h_0. The outputsof BD1 are X_1, $X`_1$, $X_{11}$ and $X`_{11}$ where $f(h, X_1) = f1\ (h, X`_1) = f\ (h, X_{11}) = f1\ (h, X`_{11}) = d_0$. Second, the values of X_1, $X`_1$, $X_{11}$ and $X`_{11}$ become 0 for thenext iteration of f and $f1$, the outputs are W_2 and h_1. These outputs are theinputs for the next iteration, along with the generic birthday attack2 (BD2) formessages M_2, $M`_2$,$M_{22}$ and $M`_{22}$. The output of BD2 are X_2, $X`_2$, $X_{22}$ and $X`_{22}$where $f\ (h, X_2) = f1\ (h, X`_2) = f\ (h, X_{22}) = f1\ (h, X`_{22}) = d_1$. Figure 9 shows thedetails of this attack.

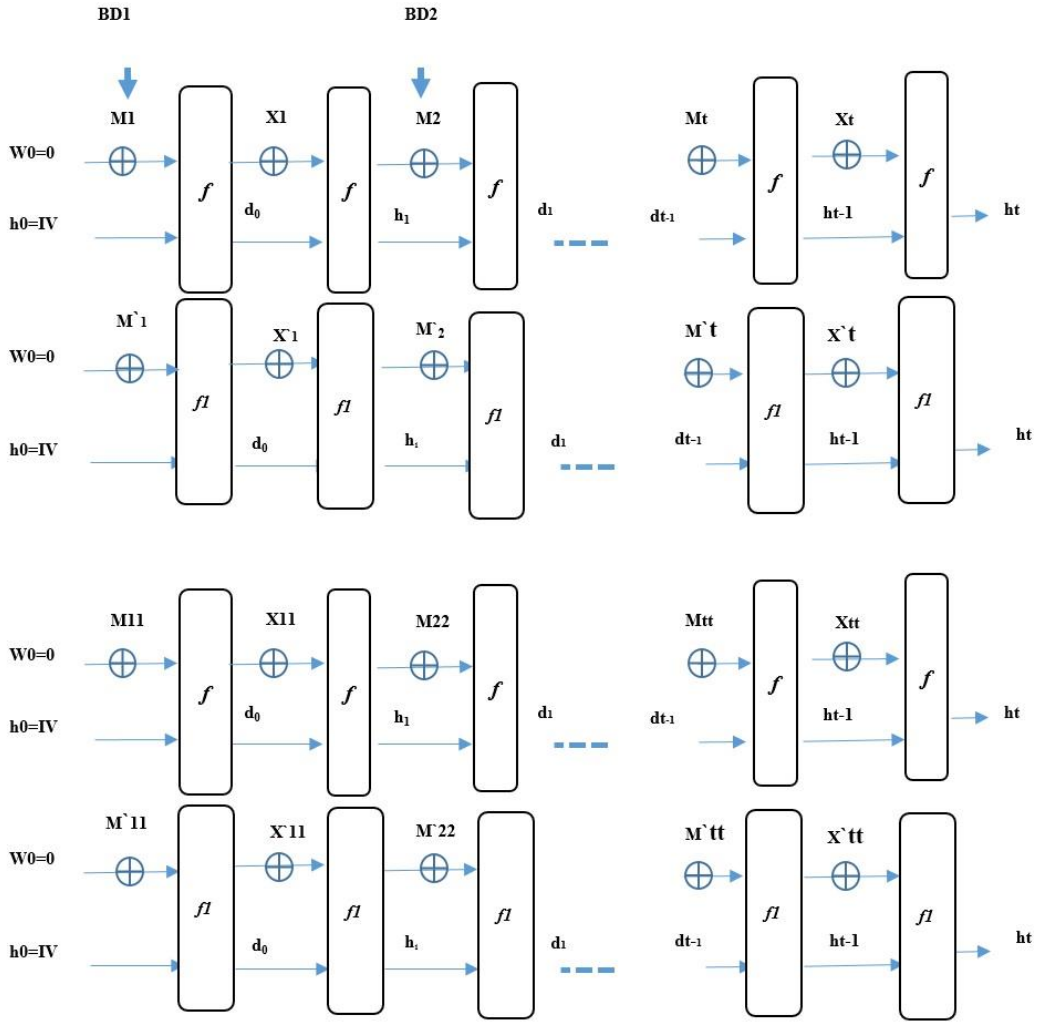

Fig. 9. Multicollision attack on DLP construction.

Analysis against 2nd pre-image Attack and Herding attack

The second pre-image attack(Kelsey, 2005) and Herding attack (Kelsey, 2006) works on Merkle-Damg°ard construction,in sponge they suggest to make it more resistance against these attacks by making $c \geq 2n$, to get a complexity of $2^{2(c+3)/2}$.

Kelsey and Schneier (Kelsey, 2005) launched a \second pre-image attack" on iteratedhash functions by computing multicollisions with the so-called \expandable message-pattern." The difference between their attack and Joux's attack is that the costno longer depends on *t*. Moreover, these researchers detected *t*-collisions whenthe compression function reaches a fixed point. This attack works when the adversary is allowed to choose the value of*IV*.

Herding (Kelsey, 2006) attacks the Merkle-Damgard hash function, they found that anyattacker who can find multicollision (Joux, 2004) on the hash function, then herd anyprefix by chose appropriate suffix. Herding attack focus on the *CTFP* (chosentarget forced prefix preimage resistance) properties of hash function which isalso known as \target value resistance". They also find a differentiable betweenthe Merkle-Damgard and *RO* based on this properties.

If we apply these attacks to DLP sponge construction with capacity 2^c weobtain higher complexity than those obtained (Kelsey, 2005) and (Kelsey, 2006), the second – preimageincludes inner collision. So, for small value of *M*, the workload is near 2^c. InRandom Oracle the expected work to find second pre-image is 2^n. In DLP spongeconstruction there is no more worried about the size of *n* and *c* as in sponge (Bertoni, 2007).

Analysis against length extension attack

According to length extension attack(Gligoroski, 2010), the complexity of finding (*Z*, *x*), $x = H(M||Z)$, greater than or equal to 2^n.Given $h(M)$, *M* unknown input, able to find $h(M||X)$, *X* known input, in DLPsponge it's possible to do this if can find two inputs yield to same output at theend of squeezing of *M*, the complexity is $2^{(c+r)-n}$.

5. Discussion & Conclusion

In this paper, we proposed a new construction of hash function named as DLPsponge. This new construction could be used as a keyed hash in authenticatedencryption and MAC, to solve the issue related with small keys and spongesecurity when c = n. The security analysis shows that DLP sponge is resistantagainst many sophisticated attacks such as multicollision attack and herdingattack.

References:

1. Alahmad, M.A., Alshaikhli, I.F. and Nandi, M., 2013, September. Multicollisions in Sponge construction. In Informatics and Creative Multimedia (ICICM), 2013 International Conference on (pp. 215-219). IEEE.
2. Aumasson, J.P., 2008, December. Faster multicollisions. In International Conference on Cryptology in India (pp. 67-77). Springer Berlin Heidelberg.
3. Aumasson, J.P., Henzen, L., Meier, W. and Naya-Plasencia, M., 2010, August. Quark: A lightweight hash. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 1-15). Springer Berlin Heidelberg.
4. Aumasson, J.P., Knellwolf, S. and Meier, W., 2012. Heavy Quark for secure AEAD. DIAC-Directions in Authenticated Ciphers.
5. Baraa Tareq Hammad, Norziana Jamil, Muhammad Reza Zaba, Mohd Ezanee Rusli and Ismail Taha Ahmed, "Faster Multicollision attack on sponge construction", International Conference on Computer, Communication and Control Technology (I4CT) 2016.
6. Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2007, May. Sponge functions. In ECRYPT hash workshop (Vol. 2007).
7. Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2008, April. On the indifferentiability of the sponge construction. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 181-197). Springer Berlin Heidelberg.
8. Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2009. Keccak sponge function family main document. Submission to NIST (Round 2), 3, p.30.

9. Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2011a, August. Duplexing the sponge: single-pass authenticated encryption and other applications. In International Workshop on Selected Areas in Cryptography (pp. 320-337). Springer Berlin Heidelberg.
10. Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2011b. Cryptographic sponges. Online http://sponge. noekeon. org.
11. Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2011c. On the security of the keyed sponge construction. SKEW.
12. Bertoni, G., Daemen, J., Peeters, M. and Van Assche, G., 2012. Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers.
13. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K. and Verbauwhede, I., 2011, September. SPONGENT: A lightweight hash function. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 312-325). Springer Berlin Heidelberg.
14. Borowski, M., 2013, October. The sponge construction as a source of secure cryptographic primitives. In Military Communications and Information Systems Conference (MCC), 2013 (pp. 1-5). IEEE.
15. Diffie, W. and Hellman, M., 1976. New directions in cryptography. IEEE transactions on Information Theory, 22(6), pp.644-654.
16. Gligoroski, D., 2010, September. Length extension attack on narrow-pipe SHA-3 candidates. In International Conference on ICT Innovations (pp. 5-10). Springer Berlin Heidelberg.
17. Guo, J., Peyrin, T. and Poschmann, A., 2011, August. The PHOTON family of lightweight hash functions. In Annual Cryptology Conference (pp. 222-239). Springer Berlin Heidelberg.
18. Hirose, S., 2004, December. Provably secure double-block-length hash functions in a black-box model. In International Conference on Information Security and Cryptology (pp. 330-342). Springer Berlin Heidelberg.
19. Hirose, S., 2006, March. Some plausible constructions of double-block-length hash functions. In International Workshop on Fast Software Encryption (pp. 210-225). Springer Berlin Heidelberg.
20. Joux, A., 2004, August. Multicollisions in iterated hash functions. Application to cascaded constructions. In Annual International Cryptology Conference (pp. 306-316). Springer Berlin Heidelberg.
21. Kelsey, J. and Kohno, T., 2006, May. Herding hash functions and the Nostradamus attack. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 183-200). Springer Berlin Heidelberg.

22. Kelsey, J. and Schneier, B., 2005, May. Second preimages on n-bit hash functions for much less than 2 n work. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 474-490). Springer Berlin Heidelberg.
23. Nandi, M., Lee, W., Sakurai, K. and Lee, S., 2005, February. Security analysis of a 2/3-rate double length compression function in the black-box model. In International Workshop on Fast Software Encryption (pp. 243-254). Springer Berlin Heidelberg.
24. Thomsen, S.S. and Knudsen, L.R., 2005. Cryptographic hash functions (Doctoral dissertation, Technical University of DenmarkDanmarks Tekniske Universitet, Department of Applied Mathematics and Computer ScienceInstitut for Matematik og Computer Science).
25. Wu, W., Wu, S., Zhang, L., Zou, J. and Dong, L., 2013. LHash: A Lightweight Hash Function (Full Version). IACR Cryptology ePrint Archive, 2013, p.867.
26. Yalçın, T. and Kavun, E.B., 2012, November. On the implementation aspects of sponge-based authenticated encryption for pervasive devices. In International Conference on Smart Card Research and Advanced Applications (pp. 141-157). Springer Berlin Heidelberg.