November-December 2019
ISSN: 0193-4120 Page No. 972- 979

Design of Hash based Extractor and Stream based Expander – The HSKDF Scheme

Chuah Chai Wen
Information Security Interest Group (ISIG)
University Tun Hussein Onn Malaysia
**Corresponding author: cwchuah@uthm.edu.my*

Article Info
Volume 81
Page Number: 972- 979
Publication Issue:
November-December 2019

Article History
Article Received: *3 January 2019*
Revised: *25 March 2019*
Accepted: *28 July 2019*
Publication: *25 November 2019*

Abstract

Key derivation function (KDF) is a basic cryptographic algorithm which is used to generate an arbitrary length of pseudorandom cryptographic keys from a secret string and some public string. These cryptographic keys are used to protect the confidentiality of the electronic data when transmitting over the Internet. The design of KDF is based on two phase which are extractor and expander. To date, all the key derivation functions are composed using the same cryptography ciphers for the extractor and the expander. The cryptography ciphers are stream cipher, keyed-hash message authentication code (HMAC), and block ciphers. This paper intended to show an alternative design in constructing the KDFs with combination of two different cryptography ciphers. The results have shown that extractor based on keyed-hash message authentication code and expander based on stream ciphers preserved the existing the highest security level and offer significant efficiency advantages in term of execution or running time over the existing HMAC, block cipher or stream cipher based KDFs.

Keywords*: Key derivation functions, extractor, expander, keyed-hash message authentication codes, block ciphers, stream ciphers*

1 INTRODUCTION

A key derivation function (KDF) is a cryptographic algorithm in generating one or more arbitrary length of cryptographically strong (pseudorandom) cryptographic keys from a secret string and public string. The security of the KDFs rely on the entropy of secret string. The recent KDF proposals are mostly two phase, which consists of an extractor and an expander [1 – 4]. This two phase KDFs are flexible, which allow the researchers conduct the design and perform the security analyse for the extractor and the expander separately. KDFs are widely used in the Host identity protocol [5]and Transport Layer Security [6]. Therefore, it is critical to design the secure KDF proposals in securing the electronic data when transmitting over these insecure channels.

To date, the existing KDF proposals are constructed using keyed-hash message authentication code (HMAC) [1], block ciphers[2] and stream ciphers [3, 7]. HMAC and block cipher take a variable length of input and produce a fixed-length output. It is slightly contradicting with the design for the

November-December 2019
ISSN: 0193-4120 Page No. 972- 979

KDF to generate arbitrary length of output. Therefore, HMAC and block cipher need to be modified to suit with the design for the KDF. Such that using HMAC or block cipher to product a number blocks of output until obtain the required length, it may exceed the number bits, hence discard any leftover bits. This is a waste. Therefore, a KDF based on stream ciphers is proposed which may generate arbitrary length of cryptrographic key without discarding any bits in excess of the required length. But, the security of KDF based on HMAC is higher than the KDF based on stream ciphers. Noted that, the security of KDFs are relied on the key of the ciphers itself. For example, if the HMAC is from SHA256, the brute force complexity is 2^{256} or collision complexity is 2^{128}, compared with Trivium, the brute force complexity is 2^{80}, or collision complexity is 2^{40}.

For current twophase KDF, both the extractor and the expander are based on the same ciphers. Stream ciphers are fast and able to generate arbitrary length of output. HMAC and block ciphers can only generate fixed-length output and slower, but better security. Therefore, this research is to propose an alternative KDF proposal with different ciphers to construct the extractor and the expander. Then, examine the security and efficiency in term of execution time using the real data size are investigated.

2 KEY DERIVATION FUNCTION

KDF generates the arbitrary length of cryptographic key from secret string (password, Diffie-Hellman share secret key) and public strings (salt and context information). The current design of KDF consists of twophase. The rational of designing the KDF in twophase is it more flexible and the process of designed and analysed the extractor and expander can be conducted separately.

Definition 1. (KDF) [1, 3 – 4]. A key derivation function is defined as: $K \leftarrow (p, s, c, n)$, where

- p is secret string with length of l^p. It is chosen from the space of α.
- s is public random string known as salt with length of l^s.It is chosen from the salt space β.
- c is public context string with length of l^c. It is chosen from a context space γ.
- K is cryptographic key.
- n is the length of cryptographic key, K.

Definition2. (Computational extractor)[1]. Let α and β be set spaces of $\{0,1\}^{l^p}$ and $\{0,1\}^{l^s}$ respectively. An extractor, $EXT: \{0,1\}^{l^p} x \{0,1\}^{l^s} \rightarrow \{0,1\}^{l^{PKR}}$ is called a $(m, t_X, q_X, \varepsilon_X)$ – min-entropy computational extractor where exist of an adversary $\mathbb{A}$ running in a polynomial number of time t_X who can make atmost q_X queries to the EXT, the $\mathbb{A}$ may distinguish between the derived PKR from p with m min-entropy or a random string of the same length, with probability not larger than $(\frac{1}{2} + \varepsilon_X)$ where ε_X is negligible.

Definition 3.(Expander)[1]. Let γbe set spaces $\{0,1\}^{l^c}$. An expander, $EXP: \{0,1\}^{l^{PKR}} x \{0,1\}^{l^c} \rightarrow \{0,1\}^n$ is called a(t_Y, q_Y, ε_Y)-secure arbitrary length of pseudorandom function family where exist of $\mathbb{A}$ running in a polynomial number of time $t_Y$who can make at most q_Y queries to the EXP , the $\mathbb{A}$ may distinguish the cryptographic key generated by the

Published by: The Mattingley Publishing Co., Inc.

November-December 2019
ISSN: 0193-4120 Page No. 972- 979

$EXP$from a random string of the samelength with probability not larger than $(\frac{1}{2}+\varepsilon_Y)$, where$\varepsilon_Y$is negligible.

2.1 The Security of KDF – Adaptive Chosen Public Inputs Model with Multiple Salts (CPM)

The primary security property for a KDF is that the s and c arepublicly known but the arbitrary length of K generated by the KDF are indistinguishable from truly random binary strings of the same length. The CPM is the formal security model which is used to examine the security of the KDFs. CPM is using the modern cryptography security proof which is an indistinguishable game played between a challenger $\mathbb{C}$ and an adversary $\mathbb{A}$ in polynomial time t. $\mathbb{A}$ try to find the flawof the KDF in polynomial time. The KDF is secure if $\mathbb{A}$ can win the indistinguishable game with probability of$(\frac{1}{2}+\varepsilon)$, where$\varepsilon$ is negligible.

Definition 4.(CPM-secure) [8].A KDF is called (m,t,q,ε) CPM-secure if for all polynomial time adversary $\mathbb{A}$ is making at most $q<|\beta|\times|\gamma|$ queries to the KDF who can win the following indistinguishability game with probability not larger than $\left(\frac{1}{2}+\epsilon\right)$.

1. $\mathbb{C}$selects$p\leftarrow\alpha$.
2. For $i=1,\ldots,q^{'}\leq q$, $\mathbb{A}$ chooses $s_i\leftarrow\beta$ and $c_i\leftarrow\gamma$. $\mathbb{A}$ sends s_i and c_i to $\mathbb{C}$ and $\mathbb{C}$ generates $K_i\leftarrow KDF(p,s_i,c_i,n)$. $\mathbb{C}$ sends the K_i to $\mathbb{A}$.
3. $\mathbb{A}$ chooses $s\leftarrow\beta$ and $c\leftarrow\gamma$. $(\{s,x\}\notin\{s_i,c_i\},\ldots,\{s_q^{'},c_q^{'}\})$. $\mathbb{C}$ chooses b randomly, $b\overset{R}{\leftarrow}\{0,1\}$. If $b=0$, $\mathbb{C}$ generates $K^{'}\leftarrow KDF(p,s,c,n)$, $K^{'}\overset{R}{\leftarrow}\{0,1\}^n$. $\mathbb{C}$ sends $K^{'}$ to $\mathbb{A}$.
4. Repeat the step 2 process in learning phase for up to $q-q'$ queries $(\{s_i,c_i\}\notin\{s,c\})$.
5. $\mathbb{A}$ output $b^{'}=0$, if $\mathbb{A}$ believes $K^{'}$ is K generated by KDF, else output $b^{'}=1$. A wins if $b^{'}=b$.

2.2 HMAC Based KDF

KDF using HMAC-SHA families (HKDF) is proposed by Krawczyk[1]. The HKDF is proof CPM-secure[8]. The HMAC is a pseudorandom function, PRF. The two-phase HKDF consists of computational extractor and a pseudorandom expander. The computational extractor is, $EXT(p,s):PRK\leftarrow F((s\oplus opad)\parallel F((s\oplus ipad)\parallel p))$, where F is a hash function (SHA1 or SHA2), opad and ipad are constant values of $0x36$ and $0x5C$ respectively. Krawczyk proposed the length of s is equal to the length of hash digest. If l^s is greater than block size of hash function l^f, s is hashed using F [1], else if l^s is less than block size of hash function, $s$is padded with zero until l^s is equal to the block size length of F [1]. PRK is the intermediate value that output from the extractor phase. The length of PRK, we denoted it as l^{PRK} is depended on the length of hash digest.

The pseudorandom expander of the HKDF generates n bits cryptographic key from PKR and c . The pseudorandom expander is, $K(i+1)\leftarrow F\left((PRK\oplus opad)\parallel F\left((PRK\oplus ipad)\parallel K(i)\parallel c\parallel i\right)\right),1\leq i<L$, $L=\left\lceil\frac{n}{l^f}\right\rceil$. The l^{PRK} is in the number blocks of l^f . The $K=K(1)||\ldots||K(L-1)$, the

Published by: The Mattingley Publishing Co., Inc.

first n bits is used as cryptographic keys K, and discard the remaining bits.

Definition 5.(HMAC-PRF) [9 - 10]. For a HMAC, $f: \mathcal{K}\, x\, \mathcal{D} \rightarrow \mathcal{R}$, under a key $\mathcal{k} \in \mathcal{K}$. An arbitrary length of input keyed function, $\mathcal{G}: \{0,1\}^c x\, \{0,1\}^{b*} \rightarrow \{0,1\}^c$ is (ε, t, g, l)- PRF secure, if for any $\mathbb{A}$ running in time t who makes at most q queries, each of length at most l (the b bits block), a $\mathcal{R}: \{0,1\}^{b*} \rightarrow \{0,1\}^c$ and a uniformly random key $\mathcal{K} \leftarrow \{0,1\}^c$, we have $\Delta^{\mathbb{A}}(\mathcal{G}_{\mathcal{K}}, \mathcal{R}) \leq \varepsilon$.

2.3 Block Cipher Based KDF

NIST SP 800-56C document specify a two-phase KDF using Advanced Encryption Standard (AES) with mode operation of Cipher-based Message Authentication Code (CMAC) [2], denoted it as BKDF. BKDF is proved CPM-secure [8]. The extractor of BKDF for the case of $1 \leq i \leq D$ is, $EXT(p,s): PRK_i \leftarrow F_s(PRK_{i-1} \oplus D_i), 1 \leq i < t, t = \left\lceil \frac{l^p}{128} \right\rceil$, where F is block cipher such as AES with key length of 128, 192 or 256,D_i is p divided into 128 bits blocks, $s$is used as AES key and $PRK_0 = 0^{128}$. The length of PRK is 128 bits.

The expander phase of BKDF is only can be constructed using AES-128. The expander of BKDF for the case of $1 \leq i \leq M$ is, $EXP(PRK, c, n): K(i) \leftarrow F_{PRK}(K_{i-1} \oplus M_i), 1 \leq i < t, t = \left\lceil \frac{l^c}{128} \right\rceil$,where F is AES-128, M_i is $c$divided into 128 bits per blocks, PRK is used as the AES key and $K(0) = 0^{128}$. If$n > 128$, process iterations ($\left\lceil \frac{n}{128} \right\rceil$) in generating the $K$are continues until exceeds the required length. The first n bits of K is used as cryptographic keys, and the remaining bits are discarded.

2.4 Stream Cipher Based KDF

Another two-phase KDF proposal proposed by Chuahet. al. [3]using stream ciphers (SCKDF) and proved that it is CPM-secure. The two-phased SCKDF consists of extractor and expander where both designs are referred to an ideal pseudorandom key stream generator (PKG)[11]. The design of SCKDF is much more flexible compare with HKDF and BKDF as the extractor and the expander of SCKDF are not restricted to same type of PKG, it allows combination of different types of PKG[3].

For the extractor of SCKDF, the input pairs of PKG (key with length of l^v, initial vector (IV) with length ofl^w) are replaced with the input pairs of KDF (p , s) [3].Presuming that$s$ is null, p is divided into blocks where each block is equal to the total length of $l^v + l^w$. If s is not null, the length of s is suggested equal with l^w and p is divided into blocks with length of l^v. If $l^p > l^v$, the first round of process p is divided into block that equal to l^v. Next, the remaining p is divided into blocks with length of $l^v + l^w$. All the blocks of p is executed by the PKG and output PRK with length of l^{PKR} where the l^{PKR} is equal to l^v, the l^v is the key length of PKG. The extractor of SCKDF is, $EXT(p,s): PRK \leftarrow F\left[[(p^1 \oplus s) \oplus (p^2 \parallel p^3)] \oplus \ldots \oplus (p^{l-1} \parallel p^l)\right]$, where F is a PKG and $\oplus$ is exclusive or (XOR).

For the expander phase of SCKDF, the input pairs of PKG are replaced with the input pairs of KDF (PRK, c) [3]. If c is null, then c is padded with '0' which length equal to l^w. If c is null, c is divided into blocks which length equal to l^w.All the blocks of c is executed by the PKG to generate $n$bits K. The expander phase of SCKDF is, $K \leftarrow$

November-December 2019
ISSN: 0193-4120 Page No. 972- 979

$F\left[\left[(PRK \oplus c^1) \oplus c^2\right] \oplus \ldots \oplus c^l\right]$,where the F is PKG. The good of using PKG as expander is that it able to generator arbitrary length of output without discarding any bits in excess of the required length.

Definition 6. (PKG)[11].A PKG is said to pass all polynomial time statistical tests if no polynomial time algorithm can correctly distinguish between an output sequence of the generator and a truly random sequence of the same length with probability significantly greater than $\frac{1}{2}$.

3 GENERAL SECURITY ANALYSIS

The section presents the general attacks on the KDF proposals which are brute force and collision attack with an assumption to the KDFs where no known weaknesses is found. Given the K generated from p, s and c.$\mathbb{A}$may launch a brute force or collision attack to the KDF to find either p or PRK. $\mathbb{A}$ choose the later as $l^{PRK} < l^{p}$. Table 1 presents the complexity in finding the output of expanderPRK. The l^{PKR} is used to determine complexity in performing the brute force or collisionPRK attack. Based on the result, the lowest complexity to find the PRK is Trivium based extractor with 2^{80} complexity for brute force or 2^{40} complexity for collision.The highest complexity to find the PRK isHMAC_SHA512 based extractor with 2^{512} complexity for brute force or 2^{256} complexity for collision. It followed by HMAC_SHA256 based extractor. Overall, if compared with stream cipher or block cipher based extractor, the HMAC based extractor have better security against both brute force and collision to find the PRK

Table 1. Complexity of ideal HMAC, block cipher and stream cipher based extractor.

Extractor	Brute Force	Collision
Trivium based extractor	2^{80}	2^{40}
Sosemanuk based extractor	2^{128}	2^{64}
HMAC_SHA256 based extractor	2^{256}	2^{128}
HMAC_SHA512 based extractor	2^{512}	2^{256}
AES based extractor	2^{128}	2^{64}

4 PERFORMANCE MEASUREMENT (TIME)

This section shows the performance in term of execution time for the combination of different cryptographic primitives (HMAC, block cipher and stream cipher) based extractor and expander. The experiments are executed to measure the execution or running time taken to generate the n bits $K$from p, s and c. The HMAC for this simulation are from SHA256 and SHA512 [12]. The block cipher is based on AES128 [13]. The stream ciphers are Trivium and Sosemanuk which are e-Stream finalists[14]. The lengths p, s, c and n are taken based on Host identity protocol, such that [5]:

Exp 1: p = 128 bytes, s = 8 bytes, c= 32 bytes, n = 64 bytes

Exp 2: p = 256 bytes, s = 8 bytes, c= 32 bytes, n = 192 bytes

The experiments are simulated 100 trials and time is recorded. The average time (mean) for the experiments are calculated. The running time was recorded using CLOCK measured in nanosecond.

November-December 2019
ISSN: 0193-4120 Page No. 972- 979

Figure 1 demonstrates the running time for this experiments which consists of different combination extractor and expander based KDF. The extractor and expander can be same or different cryptographic primitives. Same cryptographic primitives meaning that the extractor and expander, both are constructed using the HMAC or block ciphers or stream ciphers. For example, Trivium based extractor and Trivium based expander. Different cryptographic primitives meaning that the extractor and expander are constructed using either HMAC and block ciphers or HMAC and stream ciphers and so on. For example, HMAC_SHA256 based extractor and AES based expander to form a KDF.

For all KDF proposals, the execution time increases when the lengths of the inputs are increased. The results have shown that when the l^p and n are shorter the fastest execution time is Trivium based KDF and followed by HMAC_SHA256 based extractor and Trivium based expander. But, vice versa when l^p and n are longer. The slowest execution time for both experiments are AES based KDF.

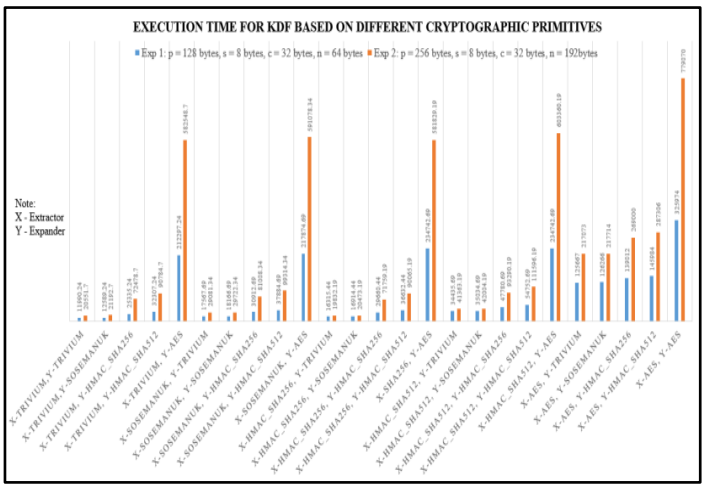

Figure 1. Execution Time for Different Combination of Extractor and Expander based Cryptographic Primitive.

5 RESULT AND DISCUSSION WITH FORMAL SECURITY ANALYSIS

By considering then general security analysis and execution time in Section 3 and Section 4 respectively, one can be concluded that the better design for KDF is HMAC-SHA256 based extractor and Trivium based expander, which denoted as HSKDF. The security level is lower than HMAC-SHA512 based extractor but to compromised the security of the HMAC-SHA256 based extractor is still a hard problem by using the current technology. Trivium based expander has high throughput and requires less resource [15] is an elegant design which can generate any$n$ bits of cryptographic key.

Hence, HSKDF is recommended as an alternative twophase KDF proposal, where the extractor is formulated using HMAC _SHA256 and the expander is formulated using Trivium. Here, the HSKDF is CPM-secure is presented. Assuming the HMAC_SHA256 is an ideal PRF satisfying the Definition 5and the Trivium is an ideal PRG satisfying the Definition 6.

Theorem 1: Suppose that a HMAC-SHA256 satisfies Definition 5 and Trivium satisfies Definition 6. If a HSKDF is built from the extractor based HMAC_SHA256 and the expander based Trivium, then the two phase HSKDF scheme is $(m, \min\{t_X, t_Y\}, \min\{q_X, q_Y\}, \varepsilon_X + \varepsilon_Y)$-CPM secure w.r.t thesecret string p with m entropy.

Proof: To proof the Theorem 1, we need to satisfy two conditions:

(a) the HMAC_SHA256 based extractor is a $(m, t_X, q_X, \varepsilon_X)$-PRF computational extractor;

(b) the Trivum based expander is a $(t_Y, q_Y, \varepsilon_Y)$-secure arbitrary length PKG.

Published by: The Mattingley Publishing Co., Inc.

November-December 2019
ISSN: 0193-4120 Page No. 972- 979

To prove (a) assuming the extractor based HMAC_SHA256 is not a $(m, t_X, q_X, \varepsilon_X)$-PRF computational extractor. This means that $\mathbb{A}$ has a polynomial time t_X algorithm to distinguish either thePRK is extracted from p with m entropy or a random string of the same length with greater than the probability of $\frac{1}{2} + \varepsilon_X$, where ε_X is not negligible. For the underlying PRF, this means $\mathbb{A}$ has a polynomial time algorithm to distinguish between PRK and a truly random string of the same length. This is contradicting with the assumption ofHMAC_SHA256 is PRF satisfies Definition 5. Therefore, (a) is correct.

To prove (b) assuming the expander based Trivium is not a $(t_Y, q_Y, \varepsilon_Y)$-secure arbitrary length PKG. This means that $\mathbb{A}$has a polynomial time $t_Y$algorithm to distinguish whether K is derived from PRK or a random string of the same length with greater than the probability of $\frac{1}{2} + \varepsilon_Y$, where ε_Y is not negligible. For the underlying PRG, this means$\mathbb{A}$ has a polynomial time algorithm to distinguish between K and a truly random string of the same length. Again, this is contradicting with the assumption of Trivium satisfies Definition 6. Therefore, (b) is correct.

Hereby, we shown Theorem 1where the HSKDF constructed using the extractor based HMAC_SHA256 and expander based Trivium is $(m, \min\{t_X, t_Y\}, \min\{q_X, q_Y\}, \varepsilon_X + \varepsilon_Y)$-CPM secure w.r.t the secret string p with m entropy.

6 CONCLUSION

In this paper, comparison of different combination of extractor and expander based cryptographic primitive in term of execution time and security analysis are presented. The cryptographic primitives are HMAC_SHA256, HMAC_SHA512, AES128, Trivium and Sosemanuk. The results have shown that extractor based HMAC_SHA256 and expander based Trivum are the ideal combination to form a KDF proposal (HSKDF) with secure and better efficiency in term of execution time. The complexity of brute force and collision attack for HSKDF are 2^{256} and 2^{80} respectively. This paper also proof that the HSKDF is (m, t, q, ε) CPM-secure. In conclusion, the HSKDF is proofed secure and efficient can be used as an alternative KDF proposal to deploy for existing applications.

ACKNOWLEDGEMENT

Author would like to thank H082, Tier 1, RMC, UTHM for the financial support.

REFERENCES

[1] H. Krawczyk, "Cryptographic Extraction and Key Derivation: The HKDF Scheme,"*Advances in Cryptology – CRYPTO 2010*of Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 6223, pp. 631 – 648, 2010.

[2] E. Barker, L. Chen, & R. Davis, "NIST SP 800-56C Rev. 1. Recommendation for Key-Derivation Methods in key-Establishment Schemes," Gaithersburg, MD. United States: NIST, 2017.

[3] C. W. Chuah, E. Dawson, & L. Simpson, "Key Derivation Function: The SCKDF Scheme," In *Security and Privacy Protection in Information Processing Systems*, Springer-Verlag, pp. 125 – 128, 2013.

[4] E. Barker, L. Chen, A. Roginsky, A. Vassilev, & R. Davis, "NIST SP 800-56A Rev. 3. Recommendation for Pair-Wise Key

Establishment Schemes Using Discrete Logarithm Cryptography," Gaithersburg, MD. United States: NIST, 2018.

[5] T. Heer, P. Jokela, T. Henderson, and R. Moskowitz,"RFC 5201: Host Identity Protocol Version 2 (HIPv2)," Technical report, Internet Engineering Task Force, 2012.

[6] E. Rescorla, "RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3," Technical report, Internet Engineering Task Force, 2018.

[7] C. W. Chuah, & W. W. Koh, "Timing Side Channel Attack on Key Derivation Functions. In *Information Science and Applications*, Singapore: Springer Nature Singapore, pp. 266 – 273, 2017.

[8] C. W. Chuah, E. Dawson, and L. Simpson, "A Framework for Security Analysis of Key Derivation Functions," In *Information Security Practice and Experience*of Lecture Notes in Computer Science, Springer Berlin Heidelberg vol. 7232, pp. 199–216, 2012.

[9] S. Hirose, "The PRF Security of Compression-Function-Based MAC Functions in the Multi-User Setting,:*IEICE Transactions* on Fundamentals of Electronics, Communications and Computer Science, vo1. 102, no. 1, pp 270 – 277, 2019.

[10] P. Gaži, K. Pietrzak, & M. Rybár, "The exact PRF-Security of NMAC and HMAC*," CRYPTO 2014, LNCS,vol.8616, pp. 113-130, 2014.

[11] J. Katz, & Y. Lindell, "Introduction to Modern Cryptography," Chapman & Hall/CRC cryptography and network security, 2014.

[12] D. Eastlake, &T. Hansen, "RFC 6234: US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)," Technical report, Internet Engineering Task Force, updated 2017.

[13] J. H. Song, R. Poovendran, J. Lee, & T. Iwata,"RFC 4493: The AES-CMAC Algorithm," Technical report, Internet Engineering Task Force, updated 2006.

[14] M. Robshaw, "The eSTREAM Project," InNew Stream Cipher Designsof Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4986, pp. 1–6, 2008.

[15] C. Canniere, & B. Preneel, "Trivium," In New Stream Cipher Designs of Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol 4986, pp 244–266, 2008.