Study on Secure Message Encryption and Authentication

Rashed MAZUMDER

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

Study on Secure Message Encryption and Authentication

Rashed MAZUMDER

Supervisor: Professor Atsuko Miyaji

School of Information Science
Japan Advanced Institute of Science and Technology

September, 2017

Abstract

An important problem in cryptography is to satisfy secure data communication over an insecure channel. Usually, computing technologies such as E-mail, on-line Banking, ATMs, Mobile Applications, IoT, Big data, Cloud Network, VANET, and MANET require secure and efficient cryptographic solutions. Now-a-days, the trade-off between security and efficiency is the most significant issue for designing a handy cryptographic solution. Generally, the cryptographic solution should be suitable to implement in variety of platforms like IoT environment, cloud network, metropolitan area network, big data environment, and smart city. In addition, it should satisfy reasonable efficiency with satisfactory security margin. Under symmetric key cryptography, certain tools are used to keep secure data communication. Among these tools, one of the important tool is cryptographic compression function.

The Cryptographic Compression Function (CCF) is used as a component of cryptographic hash (CH). The cryptographic hash is defined as to proceed data from an arbitrary domain to a fixed domain. Applications of the CH are enormous such as message verification, password verification, and pseudo-random generation. Generally, the CCF is built by scratch or block-cipher. Interestingly, the block-cipher based CCF is more apposite than the scratch based CCF because of direct hardware implementation of block-cipher. Depending on the output size of block-cipher, there are two more groups such as single block-length (SBL) and double block-length (DBL). However, the SBL is not appropriate because of the birthday-bound attack. On the contrary, the DBL has three more sub-groups depending on the key size such as (n, n), $(n, 2n)$ block-cipher based CCF, and (n, k) light-weight cipher based CCF. Under the (n, n) CCF, there are certain familiar schemes such as MDC-2, MDC-4, MJH, and Bart-12. The most of the schemes of (n, n) based CCF can not support variable size of message encryption. In addition, padding mechanism is mandatory for small and flexible size of message. These familiar schemes have less collision security bound and less efficiency-rate. Oppositely, the existing schemes of $(n, 2n)$ block-cipher CCF are classified under two classes which are based on rigorous security bound and efficiency. Usually, the efficiency-rate (r) is defined as $r = \frac{|m|}{n \times \#E}$ where $|m|$ means length of message, n directs block-length, and $\#E$ indicates number of calling block-cipher. The schemes of Weimar-DM, Hirose-DM, Tandem-DM, and Abreast-DM are members of the rigorous security bound group. These schemes are secure under the ideal cipher model (ICM). However, the security assumption of the ICM is very rigid. Hence, adversary model is weak under this security proof model. In addition, the ICM is close to the ideal world rather than the real world. On the contrary, Nandi and ISA-09 belongs to the efficiency class. These schemes need three calls of block-cipher. In addition, key scheduling is $KS = 3$. Moreover, the operating mode is serial. From the above discussions of CCF, the desired targets are:

- Upper security bound, Higher efficiency-rate, Less call of block-ciphers, Less key scheduling, and Better security proof model (close to the real world)

Next, we use Cryptographic Compression Function as a building tool in the domain of Authenticated Encryption (AE). Generally, Authenticated Encryption is a procedure that satisfies both data privacy and authenticity. The AE has many applications in the field of secure data communication such as e-banking, mobile banking, IoT, big data, and cloud network. Generally, AE consists of two modules such as Encryption and Decryption. Input of encryption module is respectively message, key, nonce, and optional associated data (AD). Usually, nonce is defined as the counter or unique number. On the contrary, cipher-text and authentication tag are output of encryption module. Generally, this tag is noted as T (n-bit value) that is used for message authentication. Moreover, cipher-text, key, authentication tag, and optional AD are the input of decryption module. In addition, if authentication tag is matched for the supplied cipher-text then output of decryption module is plain-text else error. According P. Rogaway, there is another concept of IV (Initialization of vector) that can be used instead of nonce and AD. Hence, we define probabilistic-IV-based AE or IV-based AE in short as an authenticated encryption algorithm that has a random IV without associated data whose security goal is indistinguishability with random bits with respect to an adaptive-chosen-plaintext-and-known-IV attack, unlike the common security goal of AE with nonce and associated data, which is indistinguishability with an adaptive-chosen-plaintext-and-IV attack. Under these circumstances, we classified two groups of probabilistic-IV-based AE and nonce respect (including AD) AE. Under the nonce, AD based AE schemes, there are two more subgroups in respect of security notions such as nonce respect and nonce reuse. Usually, nonce respect means the value of each nonce is unique like counter. Oppositely, if nonce value is repeated, then it is called nonce reuse. In the domain of nonce, AD based AE, the most important argument is whether the AE is secure and authentic under the nonce reuse. Interestingly, E. Fleischmann et al. claimed that nonce reuse is acceptable in the aspect of security notions of AE through the scheme of McOE. Following that, several schemes have been proposed like APE, PoE, TC, COPA, and ElmE-D. However, Hoang et al. proved that usage of nonce reuse is not secure and proper in the aspect of security notions of online AE. Hence, a door is re-opened for security notions of nonce respect in AE. Usually, nonce and AD based AE satisfies rigorous security bound. However, overhead costs are increased because of strong security model. Under the recent trends of information technology, the IoT, big data, and cloud network are emerging applications. Interestingly, the main challenges of IoT-end devices, big-data end devices and cloud network low level devices are to keep a certain level of security margin with low cost. Thus, the AE should satisfy the properties of low cost and resources also including reasonable security bound. We actually try to address that using secure cryptographic compression function its possible to build secure and efficient authenticated encryption. In addition, we have some proposals of authentication mode under the authenticated encryption scheme those have opportunity to provide higher authenticity security margin.

- Propose certain ideas of AE that are based on cryptographic compression function, Better efficiency, Application based AE (light, heavy scheme, and secure scheme), Parallel operating mode, Less call of block-cipher function.

In addition, Small Domain Message Encryption (SDE) is another application of CCF. It is one of the most prominent branch of message encryption where message domain should be small. Usually, it is defined as to encrypt short message where plain-text and cipher-text are equal in length including similar format. The SDE is widely implementable

under personal identification, and ATMs. There are certain familiar schemes those are based on card shuffling algorithm where block-cipher is used as primitive. The schemes of Swap-or-not shuffle, Mix and Cut, Thorp shuffle, and SRS are based on block-cipher (e. g. AES), where the number of calling block-ciphers or functions are high. Hence, these schemes are not efficient for encryption under the resource constrained devices and IoT-end devices. From the above discussions of TDE, the desired targets are:

- Supports small encryption function, Satisfy partial security margin

Keywords. Cryptographic Compression Function, Authenticated Encryption, Small Domain Encryption, Collision Resistance, Efficiency-rate

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Professor Atsuko Miyaji for the continuous support during my doctoral study. Moreover, her patience, motivation, and immense knowledge in the field of cryptography and security makes me confident to explore new direction in the world of cryptography. Her guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Ryuhei Uehara, Prof. Mineo Kaneko, Prof. Eiichiro Fujisaki, and Dr. Mitsuru Matsui for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I thank to all Miyaji-lab members for their different types of supports in the research related works. In addition, I would like to convey my heartiest thanks to Assistant Professor Chunhua Su, Jiageng Chen, and Satoru Tanaka for their great support during my PhD works.

We are very much grateful to all anonymous reviewers of the conferences and journals for their valuable comments. This Ph.D work has been partially supported by JAIST Doctoral Research Fellow (DRF) program. I am honoured to Japan Association for Mathematical Sciences Foundation (JAMS) because of their research-funding to me for attending several conferences.

Last but not the least, I would like to thank my family: my beloved wife and daughter, my parents and to my brothers for supporting me spiritually throughout the Ph.D work including writing this thesis. Moreover, Special thanks to Bangladeshi community people of JAIST, and Kanazawa in Japan.

Dedication

This theses is dedicated to all Employees of JAIST who helped me in every steps of my life in JAIST. In addition, dedicate to all the Japanese people outside of JAIST who made my life comfortable and easier in Japan.

Contents

Chapter 1

Introduction

Cryptography is a process of encoding data that is readable by valid senders and receivers [1, 2]. Encryption/Decryption is the major concern issue under the cryptography [1, 2, 3]. Generally, cryptography is used to satisfy secure data communication over an insecure channel. For example, computer aided services such as e-mail, e-banking, e-learning, online shopping, IoT, cloud network, and big data require secure and efficient cryptographic solutions [4, 5, 6, 7]. One of the cracking issue is the trade-off between security and efficiency for designing a cryptographic solution. Furthermore, encryption/decryption modes play significant roles for any cryptographic solution. Usually, three types of encryption mode are available such as symmetric cryptography, asymmetric cryptography, and digital envelope. However, our thesis arena is limited under the symmetric cryptography. One of the prominent cryptographic tools is Cryptographic compression function (CCF). In addition, the CCF is a useful tool for building Authenticated Encryption (AE) and Small domain encryption (SDE). Our primary works are focused for proposing secure and efficient cryptographic compression function. Next, we make a drive to use cryptographic compression function for building secure and efficient schemes of authenticated encryption and small domain encryption.

1.1 Backgrounds

Cryptographic Compression Function. A cryptographic (CH) hash function is a kind of hash function. It is suitable to use in the application of cryptography because of it's certain properties [58, 59, 60, 61]. Usually, it is noted as an algorithm where it takes arbitrary size of message and output a fixed size of message [62, 63, 64, 73]. In addition, it is assumed as a one-way function where inversion is infeasible [62, 63, 64]. There are many usage of CH such as password verification, message authentication, key derivation, and data identifier [2, 25, 61, 62, 63, 64, 70]. Now-a-days, it is an important cryptographic tool under the IoT-end device and resource constrained device [7, 66, 80]. The effectiveness of the CH depends on the internal structure. Generally, the internal structure depends on the compression function which we define here as cryptographic compression function (CCF). The CCF can be built by scratch or block-cipher [61, 63, 65]. However, the block-cipher based CCF is more suitable than the scratch based CCF because of direct hardware implementation of the block-cipher [25, 39]. Under the block-cipher, there are two branches such as single block length (SBL) and double block length (DBL). Interestingly, the DBL is more secure than the SBL due to birthday attack [25, 39, 64].

The DBL has three sub-branches. In addition, these sub-branches are classified based on the key size of the block-cipher. An (n, n), $(n, 2n)$, and lightweight-cipher are the sub-branches under the DBL [26, 27]. Our research is focused on the (n, n) and $(n, 2n)$ block-cipher based CCF. These block-cipher based CCF have some common properties such as key scheduling, number of block-ciphers, operational mode, security margin, padding oracle attack, and efficiency-rate. According to these properties, we made two groups of efficiency and security. Moreover, we evaluate existing all familiar schemes based on these two groups.

In addition, cryptographic compression function can be used as a building tool of authenticated encryption (AE). The AE is suitable for maintaining secure data communication [8, 9, 10, 49, 50, 52]. Usually, it has many usage in the applications of data and computer communication [8, 9, 10, 11, 12]. From the very beginning of security system, encryption is used in the defense organization of advanced countries like USA, Germany, France, Russia, and Japan [13, 90, 91]. Moreover, it is useful in the e-governance framework also [4, 14, 15]. Interestingly, the usage of encryption is becoming more popular in the domain of public life such as e-mail, e-banking, e-learning, and on-line shopping [4, 14, 15]. Now-a-days, data communication plays very important role in every nodes of life [2, 3, 4, 16, 17, 18]. Therefore, security and privacy are prime concern issues now-a-days. However, security and privacy issues are very complex and excessive during implementation time because of multi types and dimensions of applications [17, 37, 39, 40, 44]. For example, the characteristics of big data, IoT, and cloud network are based on multiple attributes. Thus, it is very tough job to design a flexible cryptographic solution under big data or cloud network. Usually, different types of devices are used to seek data periodically in the arena of big data, IoT, and cloud network. Therefore, there is a chance for an intruder to inject fake data. In addition, an intruder can steal information data from the insecure channel. An authenticated encryption is an important cryptographic tool. Usually, it is used to satisfy data privacy and data integrity under any insecure channel. In the domain of big data and cloud network a large number of data are needed to process via distributed network. As a result, data revocation (inject false data) and alteration of data are very common phenomena. So, authenticated encryption can play a major role under these scenarios. For example, the city (HoPe) takes a decision to merge all hospital's information including patients, doctors, and staffs. Hence, patient keeps a single medical card that can connect to central. In addition, all records are kept in the central that can access by authorized person like doctors. However, if all nodes (hospitals) are connected through cloud network then there are chances to inject false data and revoke sensitive data of patient in respect of adversary. Actually, shared network increases the risk of data falsification or data revocation. Therefore, authenticated encryption can be a good choice to maintain security and integrity in this kind of shared network. It can check data integrity (protect: inject false data) and data privacy.

In this current decade including upcoming decade, IoT is an important technology in the field of data communication [40, 71, 72]. There are many kinds of devices are used under the IoT [40, 71, 72]. Most of the devices are resource constrained device where memory and power capacity are limited [3, 40, 71, 72]. For example, RfID-tag, IoT-end device, and wireless sensor network devices are the resource constrained device that have many usage in our daily life. In addition, credit card, personal information number card, student card, and resident card are also important in our daily life. However, the cracking indicators are the behavior of these devices: whether these are secure and efficient. The

main challenges are to keep balance among cost, security margin, and efficiency. Now-a-days, researchers are trying to invent efficient crypto-device. Under this circumstance, encryption cost plays an important role. Hence, small domain encryption (SDE) is very important in the context of IoT [30, 31, 32, 33, 34]. Generally, SDE is defined as a procedure where small size of message can be encrypted. In addition, the size of message and cipher are equal [35, 36]. Moreover, SDE keeps the same format for the plain-text and cipher-text.

1.2 Motivations

Cryptographic Compression Function has certain properties such as collision resistance, preimage resistance, efficiency rate, number of calling block-cipher, operational mode, and key scheduling that reflect the effectiveness of the block-cipher based CCF [23, 24, 25, 39, 54, 55, 56]. Hence, we studied these properties and tried to find out the gaps. For example, the parameter of key scheduling is very vital for any schemes of CCF. Usually, single key schedule consumes 176-bytes of memory [74]. In the aspect of resource constrained device and IoT-end device, to keep the less key scheduling is very important. Moreover, the higher efficiency-rate encloses that the scheme is efficient. On the contrary, CR and PR are vital in the aspect of security. These two directs the security margin of any scheme. Under the (n, n) block-cipher based CCF, there are certain familiar schemes such as MDC-s, MDC-4, MJH, and Bart-12 [38, 68, 69, 70]. The MDC-2 and MDC-4 are the pioneer schemes under the domain of (n, n) block-cipher. The MDC-2 needs double key scheduling including 1/2 efficiency-rate. It needs padding mechanism for variable length of message encryption [38, 68, 69]. The CR is bounded as $O\left(2^{n/2}\right)$ for MDC-2. Moreover, the MDC-4 needs to execute four block-ciphers. The key scheduling of MDC-4 is four. It also needs the padding mechanism for flexible size of message. Another pioneer scheme is MJH. It needs single key scheduling. In addition, it is bounded as $O\left(2^{n/2}\right)$ for collision resistance. Most recently, there is another scheme of Bart-12 [59]. It has upper security margin. However, it needs three calls of block-cipher. Moreover, the number of key scheduling is three. Under these circumstances, a new scheme is needed under (n, n) block-cipher that can satisfy less key scheduling and higher security margin. In addition, it can support padding free encryption for variable size of message. Under the $(n, 2n)$ block-cipher, there are some well-known schemes such as Weimar, Hirose, Tandem, Abreast, Nandi, and ISA09. The most recent and best scheme is the Weimar-DM [25]. It has double key scheduling including 1/2 efficiency-rate. In addition, it is secure under the ideal cipher model (ICM: close to ideal world) where ICM has rigid security assumption. Interestingly, there is another security model (WCM: weak cipher model) that has less strict security assumption. Hence, it is close to the real world [62, 63]. The rest of the schemes of Hirose, Tandem, and Abreast are also secure under the ICM [28, 29, 39, 53]. For the schemes of Nandi and ISA09 are good in the aspect of efficiency-rate [64, 65]. However, the number of key scheduling is three for these two schemes. In addition, three calls of block-cipher are required for a single set of message encryption. According to the above discussions, there are certain gaps under the group of security margin and efficiency. For the group of security margin, there is an opportunity to propose a scheme of CCF that can satisfy upper security bound. In addition, it is expected to be secure under the security model of ICM, WCM, and ext. WCM. On the contrary, there is a possibility to

propose a higher efficient scheme of CCF that can satisfy less call of block-cipher and key scheduling.

Authenticated Encryption is built by scratch or block-cipher. Many of existing constructions are based on a block-cipher because we can use a well established block-cipher such as AES as a component. Usually, block-cipher based authenticated encryption is suitable for IoT and resource constrained device's encryption [7, 43]. We classify the AE in two categories. For the first one, we define probabilistic-IV-based AE or IV-based AE in short as an authenticated encryption algorithm that has a random IV without associated data whose security goal is indistinguishability with random bits with respect to an adaptive-chosen-plaintext-and-known-IV attack, unlike the common security goal of AE with nonce and associated data, which is indistinguishability with an adaptive-chosen-plaintext-and-IV attack. The second one is based on nonce based AE including associated data. In addition, IV-based AE is expected to be more suitable for resource constrained devices that the nonce and AD based AE due to its weaker security model. But, nonce and associated data based AE is more secure than that of the IV-based AE. Actually, construction of AE depends on application properties and characteristics. There are certain block-cipher based authenticated encryption such as McOE, PoE, OAE, CLOC, COPA, COBRA, SILC, and PPAE [11, 12, 42, 43, 46, 48, 77, 92]. At first we emphasize IV-based AE because it is expected to be a light solution due its weaker security model. Next, we also draw an attention on nonce and associated data based AE because of better security margin. In 2012, McOE scheme has been proposed by E. Fleischmann et. al. where nonce can be repeated [11]. Thus, many schemes have been proposed based on these concepts. If nonce can be repeated then it is light for resource constrained device encryption like IV. Interestingly, V. T. Hoang et. al. claimed in the scheme of OAE that nonce-reused is not valid concept for secure AE [42]. Therefore, doors are re-opened for doing research under the unique nonce based AE and IV based AE. Furthermore, there are certain parameters that reflect the effectiveness of the AE schemes such as efficiency-rate, number of calling encryption functions in encryption mode and authenticity mode, and operational mode (Parallel/Serial) [42, 45, 46, 47]. Hence, these properties should evaluate for identifying the efficient construction of authenticated encryption. Under these circumstances, we found certain gaps between existing familiar schemes of AE and properties of efficiency after brief study of the existing works such as operational mode (parallel/serial) in authentication of AE, less call of block-cipher in authentication of AE, and probabilistic-IV based AE. In addition, we try to focus how we can use secure cryptographic compression function as a building tool of authenticated encryption.

Small Domain Encryption can be built by traditional block-cipher or scratch [30, 31, 32, 33, 34]. For example, AES or DES are the examples of traditional block-cipher. Generally, the size of AES and DES are fixed including key size like 256, 192, 128, and 64 bits. Just think of 16 or 24 bits message, you are going to use 64 or 128 bit block-cipher. Under this condition, the key management costs are increasing. In addition, storage and operation costs are increased also. Furthermore, certain lightweight-ciphers are popular in recent days. However, the key size of these ciphers are 32, 48, 64, and 128 bits. Actually, predefined block-cipher is not appropriate for small domain encryption. According to the above discussions, small block-ciphers are suitable for small domain encryption. But the problem is security. Usually, small block-cipher is more efficient rather than the security. Thus, small block-cipher based SDE is efficient but not secure enough. Interestingly, there are some studies where it is found that resource

constrained devices rely on the less execution time, less power consumption, and less number of gates rather than the rigorous security bound. In respect of authenticated encryption and cryptographic compression function, inadequate studies have been done under the small domain encryption. At first, J.Black and P.Rogaway addressed this burning issue in [31]. However, there was no follow-up studies after that proposal for the long time. Interestingly, the SDE issue become popular when applications of IoT are increasing. Next B. Morris, P. Rogaway, T. Stegers proposed a SDE oriented construction where block-cipher e. g. AES/DES is used as primitive [30, 31, 32]. The execution time of this scheme is $O\left(\log^3 N\right)$ [30, 31, 32]. Moreover, this scheme is required to call a large number of block-ciphers. Recently a scheme has been proposed by V. T. Hoang, B. Morris, and P. Rogaway that satisfies the small domain message encryption including the format preserving encryption [30, 31, 32, 33]. This scheme is based on card shuffling of Swap-or-Not. In addition, this scheme needs to execute block-cipher also. Moreover, it invokes a large number of block-cipher (e. g. AES). The security of this scheme is bounded by $q = (1 - \varepsilon)\, 2^n$ (q : number of query, n : block-length) [30, 31, 32, 33]. There is an another scheme of Mix-and-Cut that is proposed by T. Ristenpart and S. Yilek [32, 33, 34, 35]. Interestingly, it follows by a card shuffling algorithm where the basic primitive is block-cipher. The encryption time of the Mix-and-Cut is $O\left(\log^2 N\right)$ [32, 33, 34, 35]. The one of the best construction is Sometimes-Recursive Shuffle (SRS). It is proposed by B. Morris, P. Rogaway [32, 33, 34, 35]. In addition, the execution time of this scheme $(O\left(\log N\right))$. It needs less call of block-cipher (1000 calls of block-cipher) in compare to all schemes [34, 35]. This scheme needs 80K clock cycles, or 25 μsec of recent Intel processor [34, 35]. Therefore, it can be said that the SRS is not light for resource-constrained device and IoT-end device. Generally, the constructions of SDE can be classified into two domains such as partial security domain and full security domain. Under these circumstances, it is obvious that most of the schemes are feasible for heavy system rather than the light system. Thus, we rethink about the SDE based construction that can encrypt short message. In addition, it takes less resources. Moreover, it preserves the size of message and cipher-text.

1.3 Summary of Contributions

Under the Cryptographic Compression Function, we define two classes such as group of security bound and group of efficiency. Under the group of security margin, we proposed our first scheme that satisfies higher security margin [54]. In addition, it satisfies single key scheduling. Moreover, the proposed first scheme is secure under three types of security model such as ideal, weak, and extended.weak cipher model. Interestingly, extended.weak cipher model (ext. WCM) is proposed by us in [54]. Under the group of efficiency, we proposed our second CCF construction. This construction has upper efficiency-rate. Moreover, it satisfies less key scheduling and less number of calling block-ciphers. For the (n, n) block-cipher based CCF, we proposed two more constructions. We noted as third and fourth scheme of CCF [57]. The proposed third scheme support variable message encryption. In addition, it is free from padding mechanism. Moreover, it has higher efficiency-rate. However, the security bound is less than the proposed fourth scheme's collision security bound. The proposed fourth scheme needs three calls of block-ciphers. Moreover, it needs double key scheduling.

Under the Authenticated Encryption, we proposed a scheme of IV-based AE. It satisfies padding free mechanism. In addition, it satisfies inverse freeness of block-cipher. The encryption mode of the first scheme of AE is based on serial operation. Under the first scheme, we proposed three variants of authentication mode or tag generation. The encryption mode of the first scheme needs $(n) + n \times F^{\text{prng}} + 2$ functions. In addition, the first variant authentication of the first AE scheme needs $(n-1)+1$ block-cipher function. The first variation of authentication runs in semi-parallel approach. The proposal of second variant needs $(n+1)+3$ block-cipher function for authentication. However, the second variant runs in serial mode and needs more resources. But this proposal can be an interesting future work because of getting upper authenticity security margin. However, we do not provide detail security proof under this work. The main principle is: we use $3n \rightarrow 2n$-bit secure compression function for the second variant of authentication under the first scheme. And we show that this cryptographic compression function is secure under preimage and collision resistance. In addition, the most attractive variant is third variant which runs in parallel and needs only two calls of block-cipher for generating tag under the first (Serial Scheme) scheme of AE.

For clear understanding, we mention the name of first scheme of authenticated encryption (AE) as Serial-AE. In addition, the variation of tag generation under the Serial-AE is named as Semi-Parallel Tag generation (Semi-Parallel-T.G). Hence, in combine it is formed as Serial-AE: Semi-Parallel-T.G. According to the above explanation rest of the constructions are defined as: Serial-AE: Serial-T.G and Serial-AE: Parallel-T.G., where Serial-T.G means tag generation is based on serial operation and Parallel-T.G represents parallel tag generation operation. In principle, encryption mode is similar for all constructions.

Our proposed second scheme is based on nonce respect. It operates in parallel approach. In principle, it does not suitable for associated data. However, it can only supports n-bit associated data in its initialization phase. Hence, it is suitable for IoT applications. The second scheme's encryption mode requires $(n) + n \times GF + 2$ calling functions. Moreover, the second scheme of AE has two variants of authentication mode. The first variant authentication needs $(n-1)+1$ block-cipher function. And, it operates in semi-parallel. The second variation of authentication under the second scheme of AE depends on serial operation. It needs $n+2$ block-cipher function. Actually, the proposal of second variant authentication under the second scheme is an opportunity where we can achieve better authenticity security margin. However, the detail security proof is not provided in this work. But, we provide a security proof sketch where we show that this variant can achieve upper authenticity security margin.

For clear understanding, we mention the name of second scheme of authenticated encryption (AE) as Parallel-AE. In addition, the first variation of tag generation under the Parallel-AE is named as Semi-Parallel Tag generation (Semi-Parallel-T.G). Hence, in combine it is formed as Parallel-AE: Semi-Parallel-T.G. According to the above explanation rest of the construction is defined as: Parallel-AE: Serial-T.G., where Serial-T.G represents serial tag generation operation. In principle, encryption mode is similar for all constructions.

Under the Small Domain Encryption, we proposed a scheme, which is based on small keyed function. Our proposed scheme can encrypt small chunk of message. It can encrypt arbitrary size of message without padding. It preserves the length of plain-text and cipher-text. Moreover, it is light in operation because of using small keyed function. Additionally,

it satisfies partial security margin.

1.4 Organization

This thesis consists of 7 chapters as depicted in Figure 1.1. In Chapter 2, we discuss preliminaries where we mention the basic definitions in the cryptographic tool, security notions, mathematical notions, and cryptographic primitives. We briefly discuss the existing contributions under the CCF, AE, and SDE in Chapter 3. In chapter 4, we propose

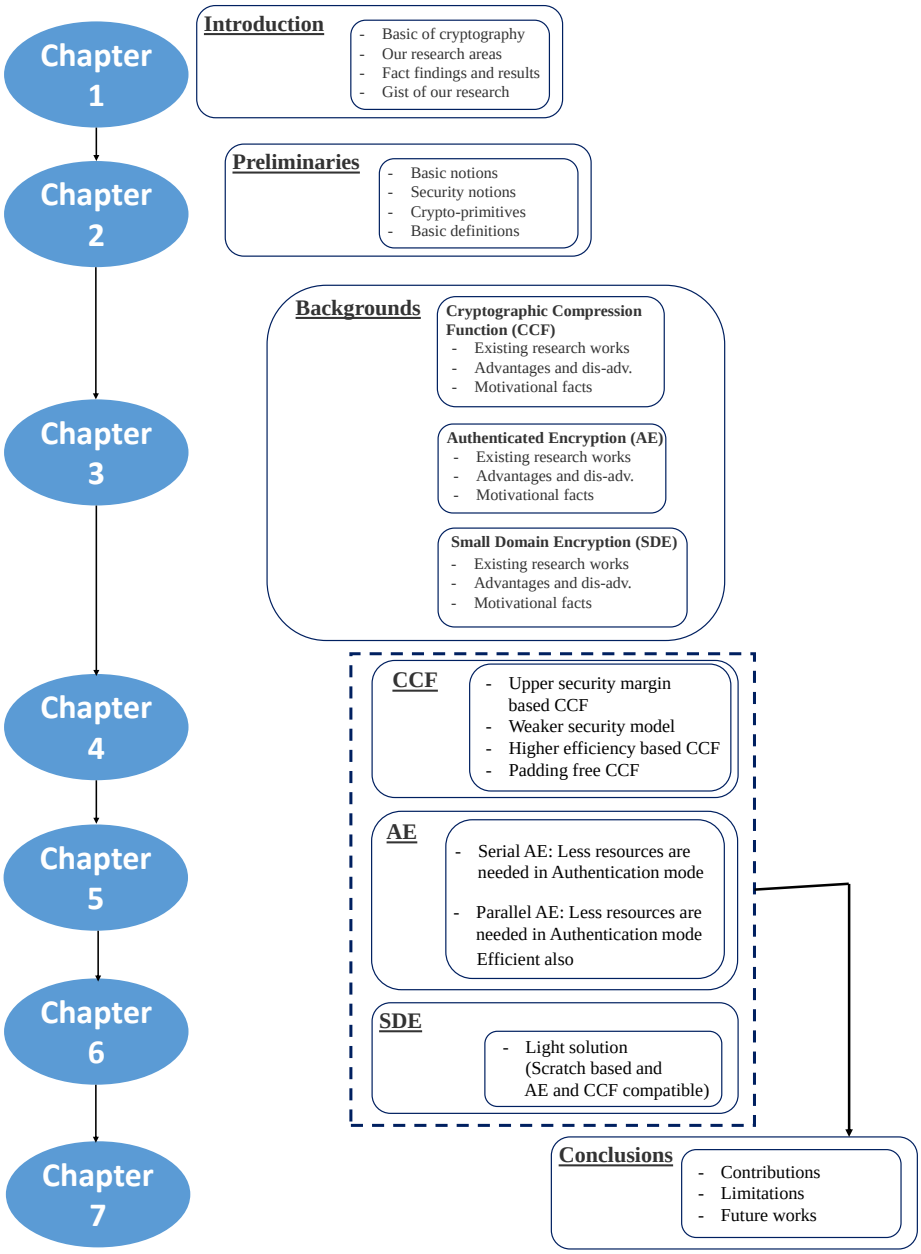


Figure 1.1: Flow Chart of the Thesis Book

better security based $(n, 2n)$ cryptographic compression function [54]. However, it is extended version of [53] paper. In addition, we propose another scheme of $(n, 2n)$ cryptographic compression function that satisfies upper efficiency. Under the (n, n) block-cipher compression function, we propose a scheme that satisfies upper security margin, padding free property, and better efficiency [57]. Furthermore, we propose another construction

of (n, n) block-cipher compression function that is secure under ideal cipher model, and weak cipher model [56].

In chapter 5, we propose a serial authenticated encryption (Serial-AE). Under the Serial-AE, we have three variants of Tag Generation (T.G: Authentication) such as Semi-Parallel-T.G, Serial-T.G, and Parallel-T.G. Therefore, in combine form these are as Serial-AE: Semi-Parallel-T.G, Serial-AE: Serial-T.G, and Serial-AE: Parallel-T.G. The scheme of Serial-AE is proposed in [49]. However, we have certain observations in Doctoral Preliminary Defense under this scheme. Therefore, we make corrections and revise in this document as Serial-AE: Semi-Parallel-T.G, Serial-AE: Serial-T.G, and Serial-AE: Parallel-T.G.

Furthermore, our proposed second scheme of authenticated encryption is named as Parallel-AE. Under the Parallel-AE, we have two variants Tag Generation (T.G) such as Semi-Parallel-T.G, and Serial-T.G. Therefore, in combine these are Parallel-AE: Semi-Parallel-T.G, Parallel-AE: Serial-T.G. However, we have certain observations in Doctoral Preliminary Defense under the scheme of Parallel-AE. Hence, we make corrections and revise in this document as Parallel-AE: Semi-Parallel-T.G, and Parallel-AE: Serial-T.G.

In chapter 6, we propose a solution of small domain encryption [51]. Finally, we conclude and draw certain future works under the chapter 7.

Chapter 2

Preliminaries

Our study is focused on secure message encryption and authentication. For encryption and authentication, one of the strongest tool is authenticated encryption. In addition, cryptographic compression functions is used for data encryption. Moreover, the tool of small domain encryption is suitable for small and flexible message encryption. Authenticated Encryption is a special encryption process that simultaneously preserves confidentiality and authenticity of the data [11, 12, 19, 20, 21]. It has two basic modules such as encryption and tag generation. Encryption module satisfies the data confidentiality. In addition, MAC or tag generation fulfils the data authenticity. There are certain standards for the authenticated encryption (Figure 2.1) according to [41, 43, 46, 47]. Usually, encryption module can be based on Initialization vector (IV) or Nonce (security aspect) [45]. IV-based encryption module is noted as $C = (E/F)^K_{IV}(m)$, where E/F, C, IV, K and m mean block-cipher/function, cipher-text, initialization vector, key and message. This encryption module is called probabilistic encryption scheme also [41, 43, 45] and denoted as $C = E^K_R(m)$ (R: random number).

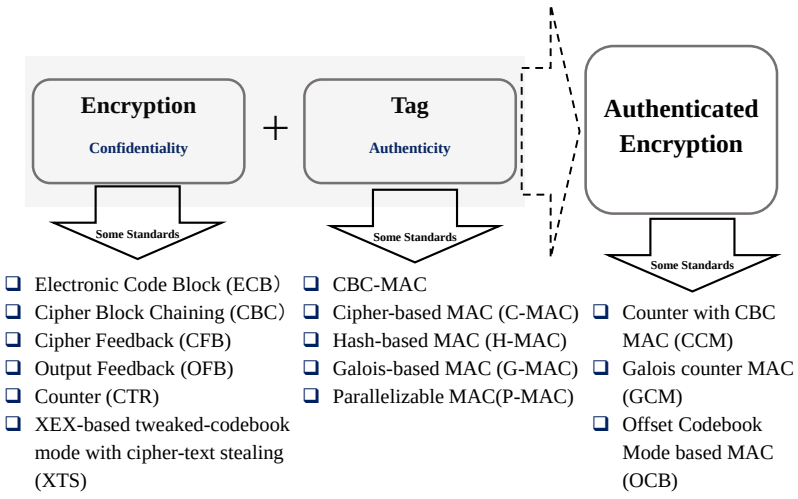


Figure 2.1: Certain Basic Modes for Confidentiality and Authenticity [1, 2, 43, 45]

On the contrary, there is another nonce based encryption scheme. Generally, nonce based encryption is denoted as $C = (E/F)^K_N(m)$ where N stands for nonce. For the tag generation or message authentication code (MAC): $T = (E/F)_K(M)$ is used, where T stands for tag. Moreover, nonce based MAC is denoted as $T = (E/F)^K_N(M)$. Fi-

nally, nonce based authenticated encryption with associated data represents as $C = (E/F)^{K}_{N,A}(M)$.

Cryptographic Compression Function (*CCF*). A cryptographic hash a special type cryptographic function that takes arbitrary length of string and returns a fixed size of string [58, 72, 73, 74]. In addition, it is hard to invert [60, 61, 62]. In the modern cryptography, there are many applications those are based on cryptographic hash such as message authentication, password verification, file identifier, pseudo-random generation, and key derivation [53, 60, 61, 66]. There are certain basic characteristics of cryptographic hash such as: Usually, cryptographic hash is deterministic such that similar message has always similar hash value. In addition, it operates very fast for computing hash in respect of given message. Moreover, it is infeasible to backtrack. Furthermore, change of a single bit of message makes a great impact on the new hash value. Additionally, to find different message under a hash value is not feasible [60, 61, 62, 63]. Generally, a cryptographic hash is built by compression function. We denoted the compression function as cryptographic compression function (CCF) (Figure 2.2).

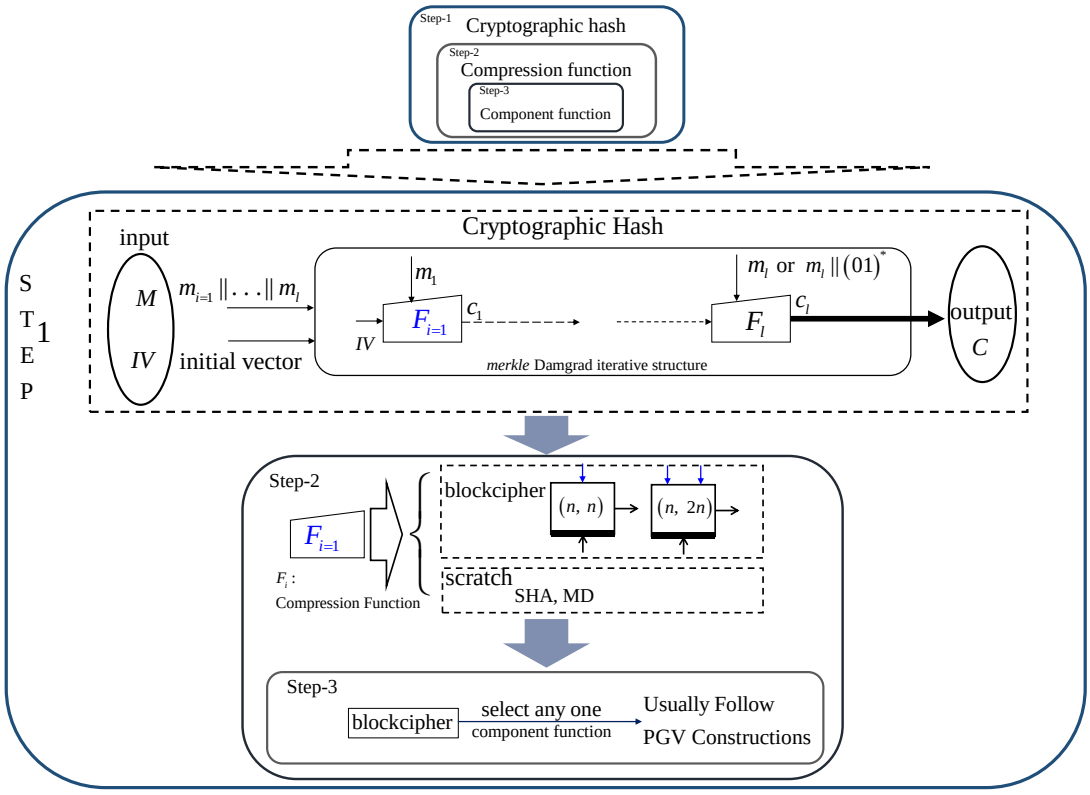


Figure 2.2: Basic of Cryptographic Hash [25, 39, 55, 56]

A cryptographic compression function is called as one-way function also. It invokes two fixed size of messages and return a fixed size of processed data [25, 39, 60]. In addition, to generate a set of input from the output is infeasible [58, 59, 60]. Moreover, it has certain differences with the conventional compression function. The CCF is used in the MerkleDamgard construction for making cryptographic hash [25, 39, 60, 61, 62].

Small Domain Encryption (*SDE*). The small domain encryption is one of the most prominent branches in message encryption [30, 31, 32, 33, 34]. Usually, small domain encryption is defined as to encrypt short message [31, 32, 33]. For example, the length of

the message is shorter than the regular block-size of AES/DES [27, 33, 34]. In addition, the final length of plain-text and cipher-text should be equal. Moreover, the format of plain-text and cipher-text should be unique in certain cases [32, 33, 34]. Day by day, the applications of small domain encryption are increasing, such as personal identification, credit card, and debit card [33, 34]. Therefore, it is a great challenge to construct an efficient and secure scheme of SDE. There are some traditional block-cipher such as AES/DES. Generally, these are suitable as a primitive for big size of data encryption. However, the situation is quite different for message encryption under the resource constrained devices in real life application [30, 31, 32]. Usually, a small chunk of message needs to encrypt for example 8, 16, 24, 32 bits. Usually, the key size of AES/DES is 128, 192 or 256 bits. Even for lightweight-cipher, the key size is 32, 48, 64, 96 bits. Under these circumstances, key and energy managements are the biggest challenges for the resource constrained devices. Therefore, the concept of SDE is very vital. However, the security is low under the SDE because of small size of the key. On the contrary, the efficiency is better because of low resource requirements. According to [30, 31, 32, 35, 36], the implementation of resource constrained devices depends on the speed, memory-utilization, power consumption, and number of gate operation rather than the rigorous security bound in certain cases.

2.1 Encryption Modes

Actually, there are notable six constructions for encryption mode [8, 11, 12, 19, 45]. Usually, these constructions should follow for proposing an encryption mode of AE. One of the simple and oldest encryption mode is ECB (Electronic Codebook). In the following Figure 2.3 we describe the summary of the notable encryption modes.

Block-ciphers/Functions. Usually, block-cipher is used under the encryption mode for encrypting message [11, 12, 19, 20, 21]. A block-cipher is a kind of function where it follows the principle of $E : K \times \{0,1\}^n \rightarrow \{0,1\}^n$. Generally, K directs the key set and n means the message and cipher-text length for the desire block-cipher. On the contrary, the inverse of block-cipher is denoted as $D = E^{-1} : K \times \{0,1\}^n \rightarrow \{0,1\}^n$. In addition, it returns the cipher-text by invoking plain-text and key. Furthermore, any one can use n to n mapping function such as $F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ where F directs the function. In addition, k and n mean key and message/cipher-text length.

Basic security of block-cipher. Generally, the security of block-cipher depends on the randomness of the key set and random permutation on n bits. Let there is an adversary $\mathcal{A}$ that tries to access on E_K and π. In addition, $\text{Perm}(n)$ directs the permutation on n-bit strings. Under this circumstance, adversary tries to distinguish between E_K and π. Hence, the advantage of adversary is denoted as:

$$\text{Adv}_E^{\text{prp}}(\mathcal{A}) = \Pr\left[K \leftarrow^{\$} \mathcal{K} : \mathcal{A}^{E_K(\cdot)} \Rightarrow 1\right] - \Pr\left[\pi \leftarrow^{\$} \text{Perm}(n) : \mathcal{A}^{\pi}(\cdot) \Rightarrow 1\right] \tag{2.1}$$

On the contrary, the basic security of block-cipher is re-defined in respect of random function instead of random permutation. Under this circumstance, the advantage of adversary is quantified as:

$$\text{Adv}_E^{\text{prf}}(\mathcal{A}) = \Pr\left[K \leftarrow^{\$} \mathcal{K} : \mathcal{A}^{E_K(\cdot)} \Rightarrow 1\right] - \Pr\left[\rho \leftarrow^{\$} \text{Func}(n,n) : \mathcal{A}^{\pi}(\cdot) \Rightarrow 1\right] \tag{2.2}$$

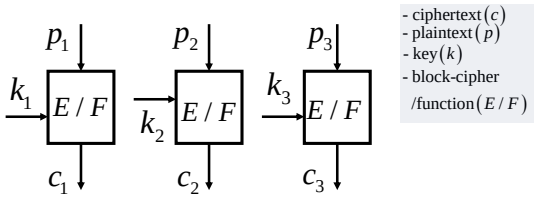
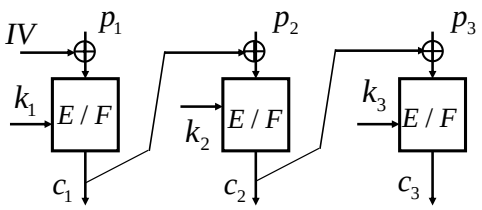
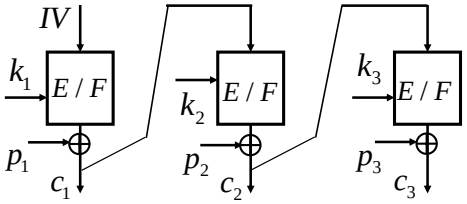
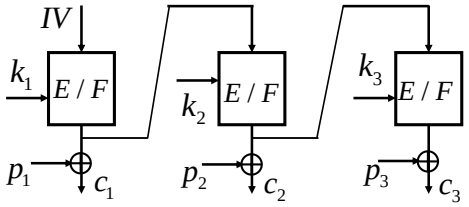
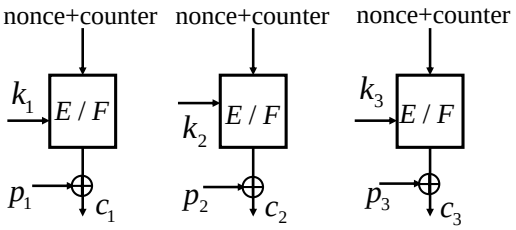
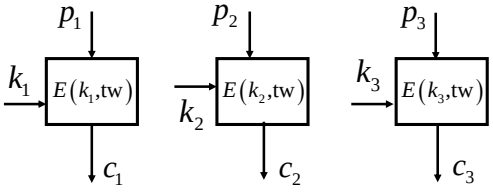
ECB	Message is clustered and each chunk of message is encrypted individually. The positive point is simple and parallel. However, the main disadvantage is identical encryption.	
CBC	Message is partitioned into several blocks and each chunk of message is XoR-ed with previous ciphertext. IV needs to use in the first block. Main drawback: it operates in serial mode.	
CFB	CFB mode is closely related to CBC. It is suitable for stream cipher. Main advantage is padding free construction. Decryption can be parallelized. Plaintext XoR-ed with encrypted value.	
OFB	OFB mode satisfies a synchronous stream cipher. It generates keystreams blocks. Then ciphertext is built by XoR operation with plaintext. IV is needed and operates in serial.	
CTR	CTR mode is called as integer counter mode (ICM) also. CTR is flexible for multi-processor machine. Every iteration, unique nonce or counter value is required. It operates in parallel fashion.	
XTS	XTS is based on tweakable block-cipher. It is secure as a strong PRP. It is also IV based encryption. XTS takes input: key, tweak (tw), and plaintext. It satisfies parallel mode.	

Figure 2.3: Basic Encryption Modes [8, 11, 12, 19, 45]

However, Func(n, n) directs the set of all functions from n-bit strings to n-bit strings. According to PRP to PRF switching lemma, the adversarial $(\mathcal{A})$ advantage over $\Pr[\mathcal{A}^{\pi} \rightarrow 1]$– $\Pr[\mathcal{A}^{\rho} \rightarrow 1]$ is $q^2/2^{n+1}$ in respect of at most q number of queries [20, 21, 48].

2.1.1 Security of Encryption Modes

There are certain basic security notions based on encryption modes and characteristics of IV, nonce, and tweak. Generally, these are probabilistic-IV based encryption, nonce-based encryption, tweak-able block-cipher.

Probabilistic Encryption. Let probabilistic-IV-based AE or IV-based AE in short as an authenticated encryption which has a random IV without associated data whose security goal is indistinguishability with random bits with respect to an adaptive-chosen-plaintext-and-known-IV attack, unlike the common security goal of AE with nonce and associated data, which is indistinguishability with an adaptive-chosen-plaintext-and-IV attack. It is noted as $C \leftarrow (E/F)_K^{IV}(m)$. The IV should be used in each iteration for each chunk of message. In addition, IV treats as set of uniform strings which are selected randomly. Furthermore, it assumes that user and adversary have no control on IV [19, 45]. This notion formalized by Bellare, Desai, Jokipii, and Rogaway [22, 45]. *Nonce-based Encryption.* Nonce-based encryption is noted as $C \leftarrow (E/F)_K^{N}(m)$. Nonce should be unique for each iteration of message encryption. In addition, user has control on the nonce. A counter is a kind of nonce [22, 45]. It was first proposed by Rogaway [22, 45]. *Tweak-able Encryption.* Tweak-able encryption invokes tweak property of block-cipher instead of IV and nonce. It is noted as $C \leftarrow (E/F)_K^{T}(m)$. Usually, it invokes message, key and tweak and return cipher-text. In reverse, message is provided in respect of cipher-text, key and tweak. First this notion is formalized by Liskov, Rivest, and Wagner [45, 63]. However, we are concern for probabilistic and nonce-based encryption. Hence, we mention the security principles for these two types.

Security of Probabilistic Encryption Scheme

SemCPA security is considerable for the IV based encryption mode [19, 45]. Usually, SemCPA means semantic security with respect to an adversarial advantage for chosen plain-text attack [19, 45]. Under the IV based encryption, the IV is uniformly distributed and chosen randomly. According to the grammar of encryption mode, there are three types of input such as key, message and IV. However, user or adversary has no control on IV under the IV based encryption mode. Hence, user can provide only two input such as message and key. Under these circumstances, let there is an adversary $\mathcal{A}$. In addition, notation of IV based encryption is $E : \mathcal{K} \times \{0,1\}^n \times \mathcal{X} \rightarrow \mathcal{Y}$. Furthermore, two types of oracle are considered such as Real and Ideal oracles. For the real oracle, key K is selected randomly from the set of $\mathcal{K}$. In addition, a message chunk m is used as input of encryption mode ($m \in \mathcal{X}$). As an inside operation, a random IV is selected as $IV \xleftarrow{\$} \{0,1\}^n$. Hence, output of IV encryption mode under the real world is $C \leftarrow (E/F)_K^{IV}(m)$. If $m \notin \mathcal{X}$ then oracle returns null value. On the contrary, message m is the input for the random oracle model. In addition, IV and C are selected randomly from the random oracle. Moreover, if $m \notin \mathcal{X}$ then random oracle terminate from the process. The adversary $\mathcal{A}$ tries to distinguish the output of real oracle from the output of random oracle. Hence, the advantage of adversary is defined from the success probability of distinguishing between two output domain. Mathematically, it is noted as:

$$\mathrm{Adv}_E^{\mathrm{ind}}(\mathcal{A}) \leq \Pr\left[\mathcal{A}^{E_K(\cdot)} \rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot)} \rightarrow 1\right] \tag{2.3}$$

Usually, security of IV based encryption mode depends on the value of $\mathrm{Adv}_E^{\mathrm{ind}}(\mathcal{A})$. If $\mathrm{Adv}_E^{\mathrm{ind}}(\mathcal{A})$ is small then informally one can claim that the IV based encryption mode is

reasonably secure.

Security of Nonce-based Encryption Scheme

Under the nonce-based encryption, nonce is chosen randomly. In addition, it is assumed that nonce never repeats. Generally, three types of input are used in encryption mode such as key, message, and nonce. In this domain, user has control on these three input. However, adversary has access on these input also. Generally, nonce based encryption is noted as $E : \mathcal{K} \times \mathcal{U} \times \mathcal{X} \rightarrow \mathcal{Y}$. For security notion, real oracle and random oracle are used for nonce-based encryption mode. Under the real oracle, $\mathcal{K}$ provides random key. A message ($m \in \mathcal{X}$) directly used in the encryption mode. Moreover, nonce ($N \in \mathcal{U}$) is also used as input of nonce-based encryption mode. Therefore, the output of the real world is $C \leftarrow (E/F)_K^N(m)$. However, if $m \notin \mathcal{X}$ and $N \notin \mathcal{U}$ then real oracle returns invalid. In the random oracle model, input $m \in \mathcal{X}$ and $N \in \mathcal{U}$ are used. In the respect of these two input oracle computes C. In addition, random oracle commits null character if $m \notin \mathcal{X}$ and $N \notin \mathcal{U}$ satisfies. Actually, the task of $\mathcal{A}$ is to distinguish the output of real oracle from the output of random oracle. In addition, the advantage of adversary is formalized from the success probability of distinguishing the two domain's output. Hence, the advantage is denoted as:

$$\mathrm{Adv}_E^{\mathrm{ind}}(\mathcal{A}) \leq \Pr\left[\mathcal{A}^{E_K(\cdot,\cdot)} \rightarrow 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot)} \rightarrow 1\right] \tag{2.4}$$

The value of $\mathrm{Adv}_E^{\mathrm{ind}}(\mathcal{A})$ directs the security margin. Generally, $\mathrm{Adv}_E^{\mathrm{ind}}(\mathcal{A})$ should be nominal. In addition, nonce-based encryption mode achieves marginal security if $\mathrm{Adv}_E^{\mathrm{ind}}(\mathcal{A})$ is small [19, 45].

2.2 Authenticity Modes

Message authentication code or tag generation is a special kind of procedure where K (Key) and m (message) are used as input. In the respect K and m, the output is T. Formally, MAC is defined as $MAC(E/F) : \mathcal{K} \times \mathcal{M} \rightarrow \{0,1\}^{|T|}$.

According to the syntax, one can claim that MAC is deterministic. However, MAC can be probabilistic, stateful or nonce-based [19, 45]. Usually, a MAC has three steps such as key generation algorithm that selects a key randomly, MAC generation algorithm that outputs a tag under the given key and message, and verification of crated tag including message [ref]. There are many cryptographic primitives to generate MAC algorithms. Among them HMAC, block-cipher, OMAC, CBC-MAC, CMAC, and PMAC are prominent [19, 45] (Figure 2.4). Moreover, UMAC and VMAC are constructed based on universal hashing which are faster in operation [19, 45].

2.2.1 Security Notion of Authentication

Basic security notion for MAC depends on two phases. Moreover, it is simple and easy to explain. Usually, MAC is noted as $T \leftarrow (E/F)_K(M)$. In addition, two types of oracle are given to adversary $\mathcal{A}$ for making query. However, key K is chosen randomly from the uniform distribution string set of $\mathcal{K}$ [19, 20, 45]. The core task for the adversary is to make a query to the MAC verification oracle for getting output 1. However, M is not the member of MAC generation algorithm under the MAC generation oracle. Under

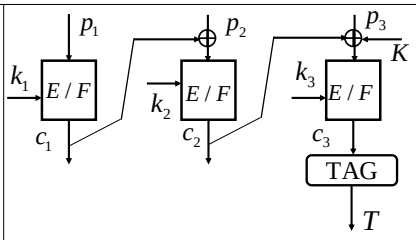
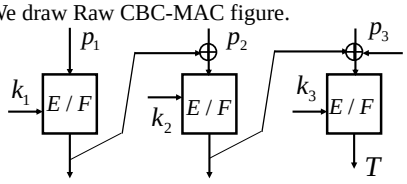
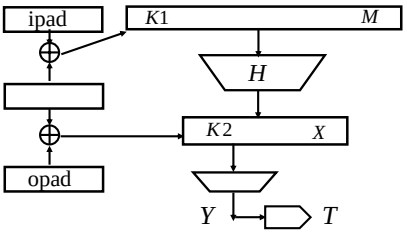
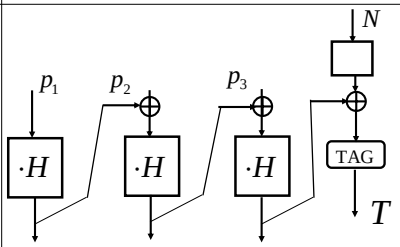
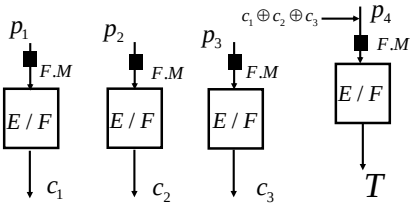
CMAC	CMAC (Cipher-based Message Authentication Code) is a block-cipher based message authentication code algorithm. CMAC is used to provide authenticity and integrity of binary data.	
CBC-MAC	CBC-MAC is a method of constructing MAC using a block-cipher. Serial fashion is used for encryption. A change to any of the message- bits will affect the final output. Hence, to know desire value without knowing the key is infeasible.	It has six types of algorithms such as MAC-1, MAC-2, MAC-3, MAC-4, MAC-5, MAC-6. We draw Raw CBC-MAC figure. 
HMAC (keyed-hash message authentication code)	HMAC is a specific type of message authentication code. It invokes cryptographic hash function and secret key. It is first formalized by Mihir Bellare. Usually, security of HMAC depends on size of the used secret key.	
GMAC (Galois/Counter MAC)	GMAC is nonce-based message authentication code scheme. It is symmetric key based. It is widely used due to its efficiency. GCM is apposite for protecting packetized data because it has min. latency and operation overhead.	
PMAC (Parallelizable MAC)	PMAC is a MAC algorithm, which was created by Phillip Rogaway. PMAC is similar in functionality to the OMAC algorithm. PMAC satisfies provable security underlying the block-cipher. It operates in parallel fashion.	

Figure 2.4: Basic Authenticity Modes [19, 20, 45, 47, 48]

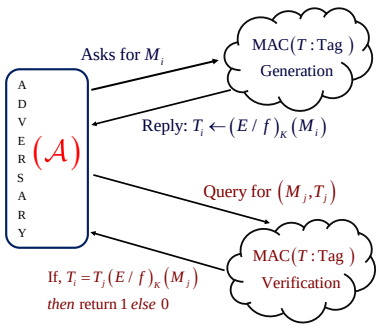


Figure 2.5: Basic Security of MAC [19, 20, 45, 47, 48]

MAC (Tag) Generation	Under this oracle, adversary sends a request for generating tag under the given message (M_i). In response, the oracle returns tag which is called as MAC.	Let $M \in \mathcal{M}$ Then, $T \leftarrow (E/F)_K(M)$
MAC (Tag) Verification	This oracle take input as M_j and T_j. It returns 1 if tags are collide in respect of generated tag.	If $T_i = T_j \leftarrow (E/F)_K(M_j)$ *then* return 1 *else* 0

Figure 2.6: Scenario for MAC security [19, 20, 45, 47, 48]

these circumstances, (M, T) is called as pair of forgery. Generally, adversary is looking to do forgery. In addition, adversary's probability of finding a forgery is defined as the advantage of adversary. Moreover, it is noted as $\mathrm{Adv}^{\mathrm{mac}}_{E/F}(\mathcal{A})$. Furthermore, the total resources are q_{gen}, q_{ver}, and t which are defined as respectively number of queries come from the MAC generation oracle and number of queries come from the MAC verification oracle, and execution time [19, 20, 21, 49, 50].

Authenticated Encryption with Associated Data. Authenticated encryption with associated data is formalized under this section. Usually, authenticated encryption (AE) is defined as:

$$E : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{X} \rightarrow \mathcal{X} \tag{2.5}$$

In addition, $\mathcal{K}$, $\mathcal{N}$, $\mathcal{A}$, and $\mathcal{X}$ means respectively key set, nonce set, associated-data set, and set of message space. Authenticated encryption invokes key, nonce, associated-data, and message as input and returns cipher-text. For example, encryption function is $C \leftarrow E_K^{N,A}(M)$ where $K \in \mathcal{K}$, $N \in \mathcal{N}$, $A \in \mathcal{A}$, $M \in \mathcal{X}$. On the contrary, decryption function is noted as $D = E^{-1} : M \leftarrow D_K^{N,A}(C)$ where $K \in \mathcal{K}$, $N \in \mathcal{N}$, $A \in \mathcal{A}$, $C \in \mathcal{X}$. However, $D_K^{N,A}(C) \rightarrow M$ is valid if $C \leftarrow E_K^{N,A}(M)$ satisfies for $M \in \mathcal{X}$. In addition, $D_K^{N,A}(C) \rightarrow$ invalid when there is no M for $C \leftarrow E_K^{N,A}(M)$.

Security Notions of Authenticated Encryption with Associated Data.

PRF Security. An experiment $\mathrm{Exp}(v)$ is defined where $v = 0$ or 1. A function Fn_K is defined as $Fn_K : K \times X \rightarrow Y$ where key (K) is chosen randomly. On the contrary, a random function is formalized as $Rfn[X, Y]$: the set of all functions from X to Y. For example, under the experiment $\mathrm{Exp}(v)$ the function Fn_K will be executed where $v = 0$. Moreover, the random function (Rfn) will be executed under the $\mathrm{Exp}(v)$ when $v = 1$. Under these circumstances, there is an adversary $\mathcal{A}$ that tries to distinguish between Fn_K and Rfn. Therefore, the advantage of adversary is quantified as:

$$\mathrm{Adv}_{\mathrm{PRF}}[\mathcal{A}] = [\Pr[\mathrm{Exp}(0) \rightarrow 1] - \Pr[\mathrm{Exp}(1) \rightarrow 1]] \text{ is negligible} \tag{2.6}$$

Furthermore, Fn_K is PRF secure iff the condition of 2.6 is satisfied.

PRP Security. The block-cipher is defined as E. Moreover, inverse of block-cipher is denoted as $D = E^{-1}$. The block-cipher function is $E/D : K \times X \rightarrow Y$ where K is selected as randomly. Furthermore, there is an random permutation like Rnp. In addition, random

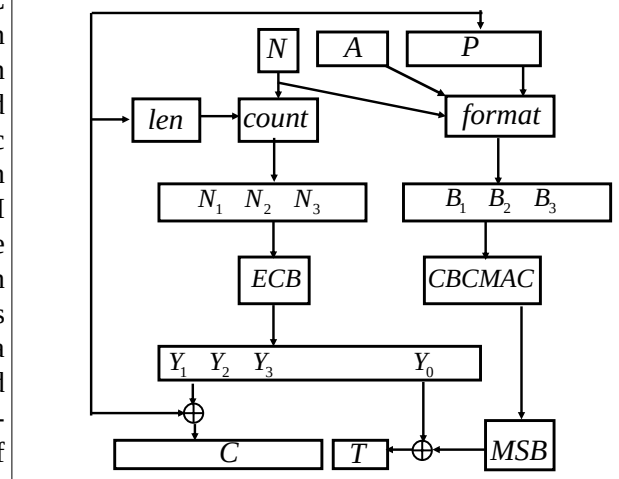
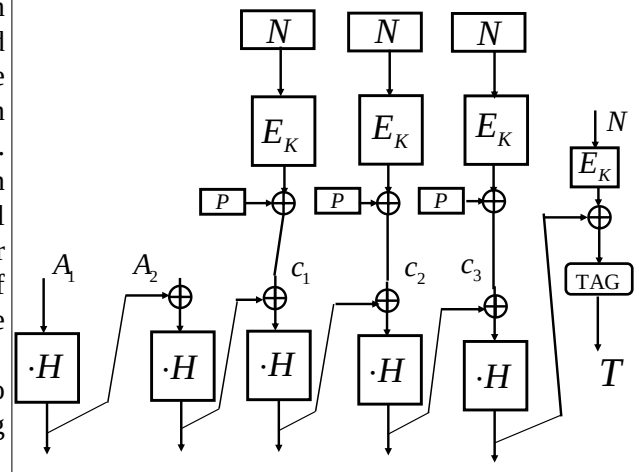
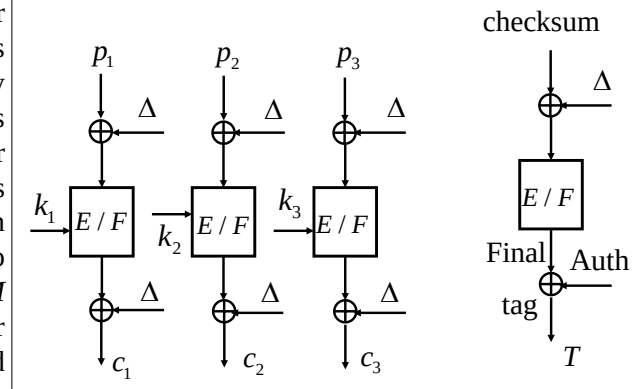
CCM	Counter with CBC-MAC (CCM) mode of operation is the first generation scheme. Usually, it is said that CCM is a generic authenticated encryption block-cipher mode. CCM is defined for use with the AES block cipher. In addition, AES-CCM has four inputs: an AES key, a nonce, a plaintext, and associated-data. AES-CCM returns the output of cipher-text and MAC (tag).	
GCM	GCM is the combination of counter mode and Galois MAC mode. The Galois field multiplication is used for authentication. In addition, it operates in parallel. It has additional useful properties. For example, it is capable of acting as a stand-alone MAC, authenticating messages when there is no data to encrypt, including no modifications.	
OCB	OCB is a block-cipher based AE that provides privacy and authenticity for the data. OCB is notable because of faster operation. OCB satisfies on-line property. In addition, it is needed to know the length of A or M for encryption, and for decryption length of A and C are needed.	

Figure 2.7: Certain Basic Constructions of AEAD [19, 20, 21, 45]

permutation depends on the random permutation properties of the key space. Under the experiment $\mathrm{Exp}(v)$, the block-cipher will be executed when $v = 0$. Additionally, random permutation will be run under the experiment when $v = 1$. However, there is an adversary $\mathcal{A}$ that tries to distinguish between block-cipher and random permutation. Moreover, the

PRP advantage of adversary is bounded as:

$$\mathrm{Adv}_{\mathrm{PRP}}[\mathcal{A}] = [\Pr[\mathrm{Exp}(0) \to 1] - \Pr[\mathrm{Exp}(1) \to 1]] \text{ is negligible} \tag{2.7}$$

Privacy. Privacy security is based on encryption oracle. The advantage of adversary is denoted as:

$$\mathrm{Adv}_{E}^{\mathrm{priv}}(\mathcal{A}) \leq \Pr\left[K \leftarrow^{\$} \mathcal{K} : \mathcal{A}^{E_K(\cdot,\cdot,\cdot)} \to 1\right] - \Pr\left[\mathcal{A}^{\$(\cdot,\cdot,\cdot)} \to 1\right] \tag{2.8}$$

Adversarial query to encryption oracle contains N, A, and M. In addition, it returns the C. Usually, this oracle is defined as real world. On the contrary, adversary makes query to random oracle gets uniformly distributed set of strings. Generally, this domain is called as ideal world. However, the main challenge for the adversary is to distinguish the real world from the ideal world. The basic assumptions are: adversary can not repeats query. Moreover, it is based on unique nonce and associated-data.

Authenticity. For the authenticity security notion, adversary is able to access encryption and decryption oracle. Adversary forges if it makes a query of $(N,\ AC)$ to the oracle and the reply is $D_K^{N,A}(C) \neq$ invalid. In addition, $(N,\ A,\ M) \to C$ is not member of encryption oracle. Therefore, the advantage of adversary is quantified as:

$$\mathrm{Adv}_{E}^{\mathrm{auth}}(\mathcal{A}) \leq \Pr\left[K \leftarrow^{\$} K : \mathcal{A}^{E_K(\cdot,\cdot,\cdot),D_K(\cdot,\cdot,\cdot)}\text{forges}\right] \tag{2.9}$$

2.3 Building Modes of Compression Function

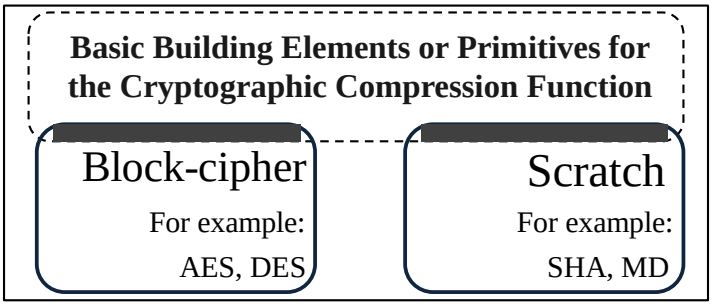


Figure 2.8: Basic Primitives for building CCF [58, 59, 60, 61, 79]

According to the Figure 2.2, let the message is M. The size of message is $l < n$, where n means the size of F in Step-1. The main task is to encrypt the message (M) through cryptographic hash. Hence, M is partitioned as $m_1, m_2, \ldots m_i$, (where $i \leq l$). Then MerkleDamgard construction is needed to organize the set of cryptographic compression function (F), where message is encrypted in serial fashion under the cryptographic hash. Next the question is how to build the compression function (F) or what the primitives are. Generally, this F can be built by the primitives of block-cipher or scratch (Figure 2.2, Step-2).

According to the Figure 2.9, block-cipher has two types such as single block-length and double-block-length [25, 63, 64, 65]. Usually, single block-length block-cipher returns the output which size is equal to block-length. On the contrary, the double block-length

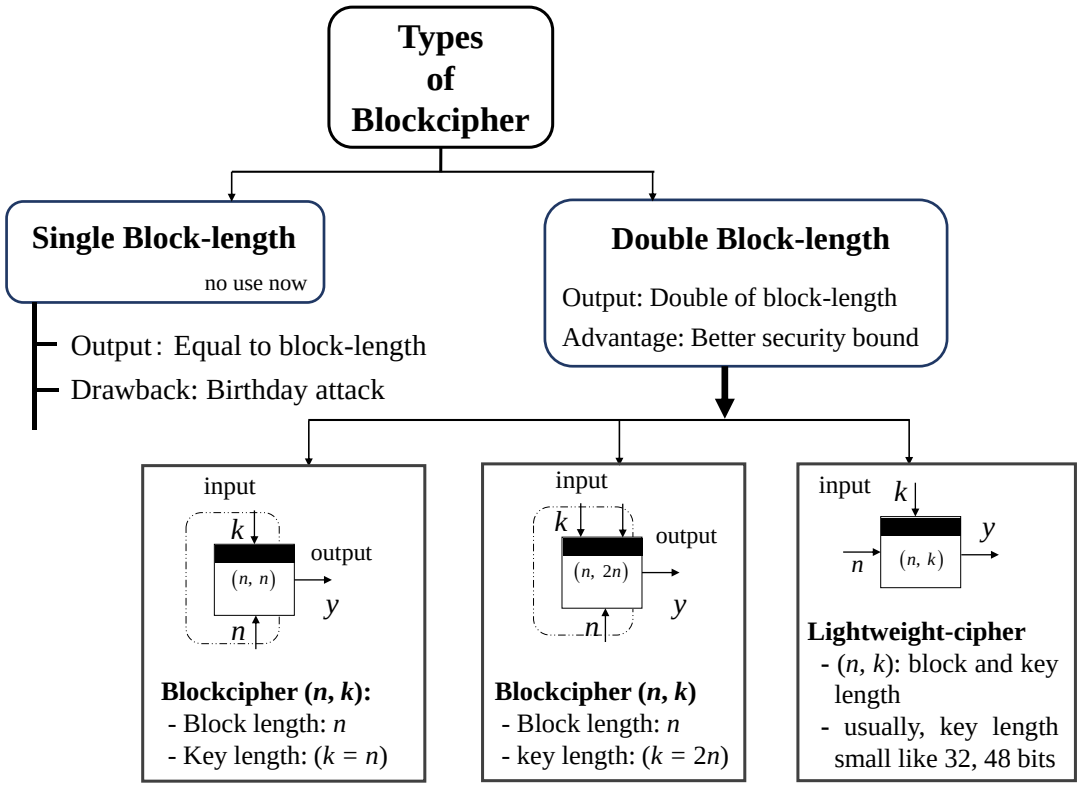


Figure 2.9: Types of Block-cipher [1, 2, 3, 55, 56]

block-cipher generates the output that size is double of block-length. Because of birthday-bound attack, single block-length block-cipher has no usage now [4, 25, 80]. However, double block-length block-cipher is appropriate for building cryptographic compression function. Interestingly, double block-length block-cipher has three classes. First one is defined as block-cipher(n, n) where n directs the size of block-length (message) and k means key-length [53, 64, 65]. Moreover, block-cipher$(n, 2n)$ is the second-one, where key size is double of block-length. Usually, this $(n, 2n)$ block-cipher is good for making cryptographic compression function in the aspect of security-margin because of greater key size. The third and last one is lightweight-cipher. This concept becomes popular under the resource constrained devices very recently. Generally, the lightweight-cipher supports small size of block-length and key-length. Moreover, AES and DES are used in the (n, n) and $(n, 2n)$ block-cipher in the perspective of implementation [25, 39, 55, 56]. On the contrary, lightweight-cipher is suitable for invoking PRESENT, XTEA, TWINE, KATAN, and KASUMI [62, 68, 80, 81].

There is another way or primitive to build CCF (Figure 2.8). Usually, this is called as scratch. The scratch is based on random function like universal function, SHA-series, and MD-series. The main advantage of the scratch function is the size flexibility. Usually, it supports greater size of message including long key. However, in the aspect of hardware implementation block-cipher is more appropriate than the scratch because of direct implementation of block-cipher [25, 49, 50, 55, 56].

Basic Properties of Cryptographic Compression Function. According to the Figure 2.10, there are two broad aspects of properties for the cryptographic compression function.

Efficiency	Security
➢ Key scheduling key schedules per blockcipher	➢ Collision resistance (CR) find an output for different two input
➢ Number of blockciphers (#*E*) no. of blockcipers are used in compression function	➢ Preimage resistance (PR) find a collision for a predefined output
➢ Operational mode (OM) multiple blockciphers run in serial/parallel	➢ Padding oracle attack (PA) length extension attack, when extra bit is added
➢ Efficiency-rate (r) r = message length/$\#E \times$ blocklength	

Figure 2.10: Classifications and Properties of CCF

First we notify here security and second one is efficiency. Under the security aspect there are three parameters such as collision resistance, preimage resistance, and second preimage resistance [60, 61, 62]. Moreover, efficiency-rate, number of key scheduling, number of calling block-ciphers or function, and padding-free are the parameters under the efficiency [49, 50, 55, 56].

Collision resistance is an important security notions for any cryptographic compression function. Usually, it is defined as a procedure where adversary tries to find M and M' (M, M' : message) such that $F(M) = F(M')$ when $M \neq M'$ [60, 61, 62]. In addition, to find $F(M) = F(M')$ is infeasible for adversary. Moreover, if adversary tries to find $F(M') = C'$ for the given the value of C such that $(F(M') \rightarrow C') = C$ [25, 39, 60, 61, 62]. Actually, this is also hard to find for the adversary [25, 39, 60]. Moreover, to find M' such that $F(M') = F(M)$ for the given M is difficult for the adversary under the second preimage resistance [39, 61, 62]. Under the efficiency aspect, the efficiency rate (r), number of block-cipher ($\# E$), number of key scheduling (KS) are prominent [25, 39, 53, 55, 56]. The efficiency rate is defined as ($r = |m|/\#E \times n$) where r means the symbol of efficiency-rate, $|m|$: message length, $\#E$: number of calling block-cipher per message encryption [25, 39]. Moreover, Key scheduling comes from the required number of key sets for single message encryption or compression [25, 39, 53, 55].

2.3.1 Security Notions of Compression Function

For cryptographic compression function, there are certain standard security notions such as collision resistance, preimage resistance, and second preimage resistance. Generally, two models are used for security proof of the CCF. The first one is based on block-cipher and the second-one is random oracle (based on random function). The block-cipher based model has two major variants such as ideal cipher model (ICM) and weak cipher model (WCM).

Ideal cipher model (ICM)

Under the ideal cipher model, a block-cipher is defined as $block(n, k)$ where n and k directs respectively block-length and key-length. Let $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block-cipher of $block(n, k)$. Moreover, $E(k, \cdot)$ is a permutation under the every $K \in \{0,1\}^k$. Let $BL(n, k)$ is the set for all block-ciphers $block(n, k)$. Generally, E is selected randomly from $BL(n, k)$. Furthermore, two types of operations are available under $BL(n, k)$ such

as encryption query and decryption query [25, 39, 61]. Usually, the input of encryption query is message and key where output is cipher-text. In addition, cipher-text and key are used as input of decryption query. On the contrary, the output of he decryption operation is message. For example, $E : (m, k) \rightarrow c$ directs the encryption query where m, k, and c represents message, key, and cipher-text. However, $D = E^{-1}$ is denoted as decryption query [ref]. Usually, a database is used to keep the query records. Moreover, three elements are stored under a single transaction such as key, message, and cipher-text. In addition, it is assumed that similar query has no chance to be executed for the second time [25, 39, 61, 62]. For example, if $[E : (m_j, k_j) \rightarrow c_j$ contains under the encryption query then there is no chance to evaluate $D = E^{-1} : (c_j, k_j) \rightarrow ?$ query. We noted block-cipher as $Block_n^k$ also.

Weak cipher model (ICM)

Weak cipher model is the fabrication of ideal cipher model [63]. It was first introduced by Liskov in [54, 61, 63] as weak ideal compression function. After that Hirose and Kuwakado re-formalized the concept of weak ideal compression function into weak cipher model [62]. Let $BL(n, k)$ be the block-cipher like ideal cipher model. Moreover, three types of operations exist instead of two types of operation. The first-one is defined as encryption query. In addition, the decryption query is the second-one. Moreover, key disclosure query is available under the weak cipher model [61, 62, 63]. For example, $E : (m_i, k_i) \rightarrow c_i$ and $D = E^{-1} : (c_j, k_j) \rightarrow m_j$ are the encryption and decryption query. Furthermore, key disclosure query is $E^k : (m_l, c_l) \rightarrow k_l$. In principle, duplicate query can not be queried under the weak cipher model also.

Security definition of compression function

There are some famous constructions standards under the PGV [60, 61, 62, 63]. Usually, it is obvious to follow those standard constructions for creating a block-cipher based cryptographic compression function (Figure 2.11). Generally, collision resistance and preimage resistance are used for mentioning security notions for any cryptographic compression function (Figure 2.12). Collision resistance is defined as to find two different input under a single output [60, 61, 62, 63]. However, the task is infeasible to the adversary in respect of computational time and resources. Moreover, to find a message (m') for the given hash value (H^{given}) such that $H(m') = H^{given}$ is called preimage resistance [25, 39, 61].

Collision Resistance of CCF. Assume there is an adversary $\mathcal{A}$ that is allowed to access through the block-cipher $(BL(n, k))$. If the output of the compression function are c_1 and c_2 under the $f_E(m_1, k_1) \rightarrow c_1$ and $f_E(m_2, k_2) \rightarrow c_2$. Then there is an experiment $(\exp_{coll}(\mathcal{A}))$ that output iff the following criteria satisfies:

$$f_E(m_1, k_1, c_1,) = f_E(m_2, k_2, c_2,) \wedge \{(m_1, k_1, c_1,) \neq (m_2, k_2, c_2,)\}$$

, where f_E is a block-cipher compression function and m, c, and k are the elements of message, cipher-text and key. The advantage of $\mathcal{A}$ is to find a collision under the f_E. Let $\mathrm{Adv}_{f_E}^{coll}(\mathcal{A}) = \Pr[\exp_{coll}(\mathcal{A}) \rightarrow 1]$ $(coll : \text{collision})$. The advantage of adversary $\mathcal{A}$ is evaluated by the number of queries that are allowed to ask block-cipher oracle. Hence, $\mathrm{Adv}_{f_E}^{\mathrm{coll}}(q) = \max_{\mathcal{A}}\left\{\mathrm{Adv}_{f_E}^{\mathrm{coll}}(\mathcal{A})\right\}$ where the maximum is taken from all the adversaries that ask at most q oracle queries [60, 61, 62, 63].

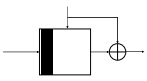
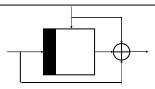
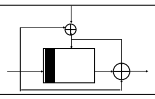
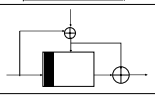
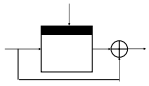
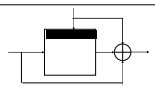
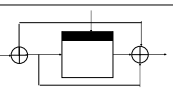
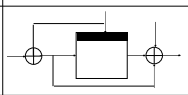
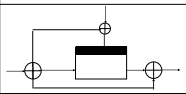
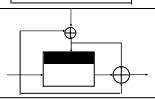
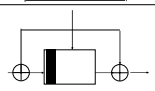
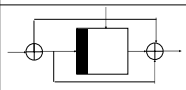
f_1	$z = E_h(m) \oplus m$		CR= $q^2/2^n$
f_2	$z = E_h(m) \oplus m \oplus h$		CR= $q^2/2^n$
f_3	$z = E_h(m \oplus h) \oplus m$		CR= $q^2/2^n$
f_4	$z = E_h(m \oplus h) \oplus h \oplus m$		CR= $q^2/2^n$
f_5	$z = E_m(h) \oplus h$		CR= $q^2/2^n$
f_6	$z = E_m(h) \oplus h \oplus m$		CR= $q^2/2^n$
f_7	$z = E_m(h = \alpha \oplus \beta) \oplus h \oplus \alpha$		CR= $q^2/2^n$
f_8	$z = E_m(h \oplus m) \oplus h \oplus m$		CR= $q^2/2^n$
f_9	$z = E_{h \oplus m}(h) \oplus h$		CR= $q^2/2^n$
f_{10}	$z = E_{h \oplus m}(h) \oplus m$		CR= $q^2/2^n$
f_{11}	$z = E_m(h \oplus m) \oplus m$		CR= $q^2/2^n$
f_{12}	$z = E_{h \oplus m}(m) \oplus h$		CR= $q^2/2^n$

Figure 2.11: Famous 12 PGV Constructions out of 64 [60, 61, 62]

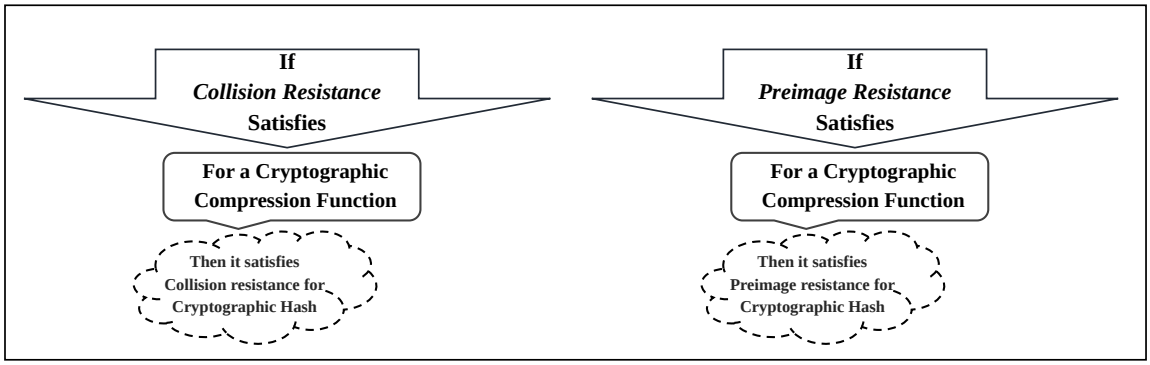


Figure 2.12: Basics of Security Standards [60, 61, 62, 63]

Preimage Resistance of CCF. Adversary $\mathcal{A}$ is able to access the block-cipher. Furthermore, $\mathcal{A}$ randomly selects a set of value: rd_1 and rd_2. If x' and y' are the responses

form the block-cipher oracle then an experiment $\exp_{pre}(\mathcal{A})$ is defined where pre stands for preimage. In addition, the output of the defined experiment is 1 iff the following condition satisfies:

$$f_E(x', y', m) = (rd_1, rd_2)$$

, where f_E is a block-cipher based compression function. The advantage of adversary for finding a preimage under f_E is defined by $\mathrm{Adv}^{pre}_{f_E}(\mathcal{A}) = \Pr[\exp_{pre}(\mathcal{A}) \to 1]$. Furthermore, the advantage of $\mathcal{A}$ is quantified in respect of the total number of queries. Hence, $\mathrm{Adv}^{pre}_{f_E}(q) = \max_{\mathcal{A}}\{\mathrm{Adv}^{pre}_{f_E}(\mathcal{A})\}$ where the maximum is taken from all the adversaries that ask at most q queries [60, 61, 62, 63].

2.4 Building modes of Small Domain Encryption

The scheme of small domain encryption can be built by block-cipher or scratch. Usually, small domain encryption means $\{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ where the length of n is small like 16, 24 or 32 bits. One can use traditional block-cipher. However, it needs padding. In addition, waste of resources because of using 128-bit AES for 32 bits message [31, 32, 33]. Therefore, bit by bit encryption is being used through certain well defined shuffling algorithms [33, 34, 35] such as Knuth shuffle, Thorp shuffle [32], Swap-or-not shuffle, Mix-and-cut shuffle [33], SRS shuffle [35]. On the contrary, Feistel network or general Feistel network are used to build the scheme of small domain encryption [30, 36].

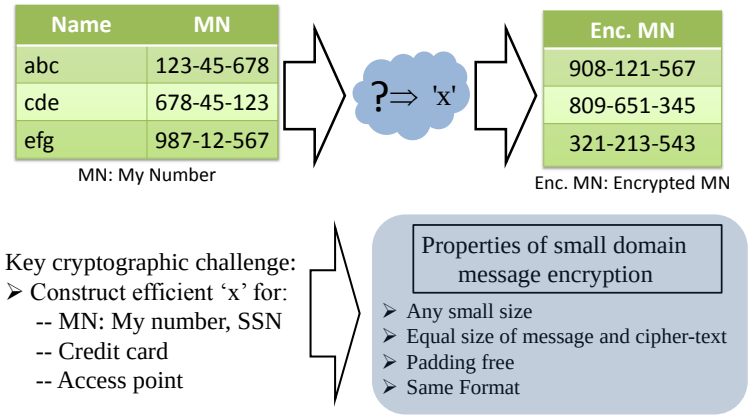


Figure 2.13: Basics SDE [30, 31, 32, 33, 35, 36]

Moreover, a small keyed-function is defined as $f : K \times M \to M$ where K means key space and M directs message space. In addition, $f_k(\cdot) = f(k, \cdot)$ is a permutation over M for every $k \in K$.It is assumed that there is an adversary $\mathcal{A}$ that can access an encryption (Enc) oracle of the proposed scheme and an oracle of random function (RO). The advantage of an adversary is defined to distinguish between the output of random-oracle and the output of the proposed scheme. Moreover, the adversary has access on ideal permutation ($). Hence, $\mathrm{Adv}^{\mathrm{cca}}_f(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathrm{Enc}(\cdot)}_{\$} = 1\right] - \Pr\left[\mathcal{A}^{\mathrm{RO}(\cdot)}_{\pi} = 1\right]$. We assume adversary $\mathcal{A}$ has non-adaptive query feature. In addition, chosen plain-text attack by any adversary is defined as each query runs under encryption query [31, 32, 33, 35]. Furthermore, we define PRNG functions as f_{pr_1} and f_{pr_2}. The operation of PRNG functions is $f_{pr_{1,2}} \to^{\mathrm{u,r}} \{0,1\}^n$, where $u:$ uniform $r:$ random.

2.4.1 Security Notions of SDE

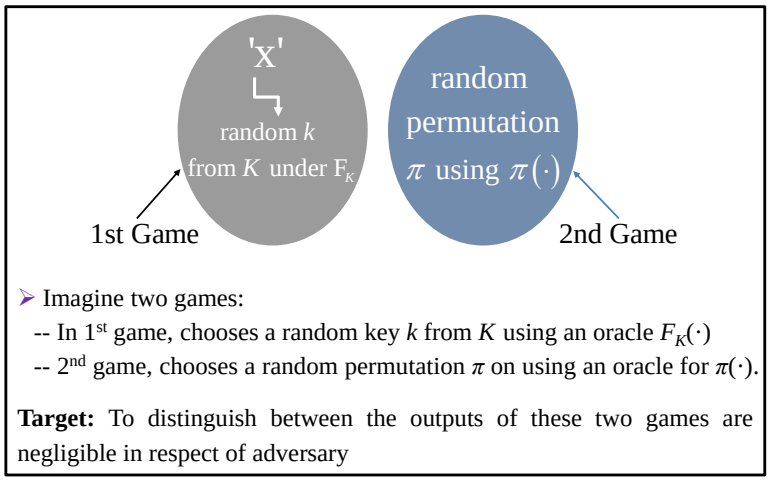


Figure 2.14: Security Model of SDE [30, 31, 32, 33, 35, 36]

The security notions (SN) is defined as SETM$[f]$ = (SETM-E, SETM-D), where SETM-E and SETM-D directs respectively encryption and decryption oracle. Encryption oracle takes M and delivers C. Furthermore, decryption oracle receives cipher-text as input and passes message through ideal permutation. In addition, adversary $\mathcal{A}$ has access to random oracle that releases corresponding $\mathcal{M}$ or $\mathcal{C}$ by random permutation (Figure 2.14). Therefore, the advantage of adversary is defined as:

$$\mathrm{Adv}^{\mathrm{SN}}(\mathcal{A}) =^{\mathrm{def.}} \Pr\left[\mathcal{A}_{\$}^{\mathrm{SETM-E}(\cdot),\mathrm{SETM-D}(\cdot)} \to 1\right] - \Pr\left[\mathcal{A}_{\pi}^{\mathrm{RO}(\cdot)} \to 1\right]$$

In addition, the first probability depends on key space and randomness of $\mathcal{A}$. Moreover, the second-one relies on random-bits oracle and $\mathcal{A}$.

Chapter 3

Existing Research Works

In this modern cryptography, message encryption and authentication are noted as "Swiss Army Knife of Cryptography" [2, 37, 44]. In each sector of information technology, message encryption and authentication are essential. For example, the verify process of integrity for files or messages, verification of the password, file/data identifier, pseudo-random generation, key derivation, social card security, credit card security, and ATM card security. In this chapter, we briefly discuss the motivations for our works through three sections. In addition, these motivations are based on the existing works of authenticated encryption, cryptographic compression function, and small domain encryption.

3.1 Previous Works in Cryptographic Compression Function

According to the construction properties, block-cipher based cryptographic compression function is broadly categorized into two groups such as (n, n) and $(n, 2n)$ block-cipher based compression function [58, 59, 60, 61, 79]. According to the Table 3.1, the security bound of Weimar-DM is the best. However, the key scheduling is twice. On the contrary, Hirose-DM satisfies single key scheduling. In addition, all scheme's mode are Davies Meyer (DM). Moreover, the proof technique of these scheme is based on ideal cipher model (ICM). Interestingly, the assumption of ICM is very rigid. Hence, it is not suitable in respect of real world. There is another proof technique of weak cipher model which has less strict assumption. Therefore, WCM is close to the real world. Under these circumstances, there is a scope to provide a new scheme that can provide better security bound. In addition, it can satisfy single key scheduling property. Moreover, there is another possibility to introduce weak cipher model security proof technique. However, the WCM has certain dis-advantage such as: under any single instance only single type query is allowed. Hence, there is a fact to provide more realistic security proof.

The parameters of CR, PR, r, $\#E$, OM, and KS are vital for any satisfactory scheme of block-cipher based compression function [39, 54, 55, 56]. Firstly, certain gaps are identified from the current familiar schemes based on the above parameters. Thus, the importance of the findings are shown in the field of efficient and secure communication. For example, the key scheduling cost is analysed in respect of construction of compression function. Usually, 176 bytes are needed for operating of single key scheduling [74]. Hence, minimization of key scheduling is a common practice. Additionally, the operation mode is

Table 3.1: Result Analysis of Different $(n, 2n)$ block-cipher based Compression Function [25, 39, 54, 55, 58, 64, 66]

	CF	CR	PR	KS	PT	OM	r	$\#E$
Weimar	$3n \to 2n$	$2^{126.23}$	$2^{252.5}$	2	ICM	DM	1/2	2
Hirose	$3n \to 2n$	$2^{124.55}$	2^{251}	1	ICM	DM	1/2	2
Abreast	$3n \to 2n$	$2^{124.42}$	2^{246}	2	ICM	DM	1/2	2
Tandem	$3n \to 2n$	$2^{120.87}$	2^{246}	2	ICM	DM	1/2	2
ISA-09	$4n \to 2n$	$O(2^n)$	-	3	ICM	DM	2/3	3
Nandi	$4n \to 2n$	$O(2^n)$	-	3	ICM	DM	2/3	3

CF: Compression Function
CR: Collision Resistance
PR: Preimage Resistance
KS: Key Schedule
PT: Proof Technique
OM: Operation Mode
MM: Matyas Meyer Oseas
DM: Davies Meyer
r: Efficiency rate
$\#E$: Number of calling block-ciphers/functions

very crucial for resource limited devices, where the parallel mode can provide maximum support in respect of memory system [25, 39, 55, 56, 62]. Moreover, the efficiency-rate needs to reach the landmark ($r = 1$) [25, 39]. There are some well-known schemes of block-cipher compression function such as Weimar, Hirose, Tandem, Abreast, Nandi, and ISA09 (Table 3.1). For example, Weimar-DM provides tight security bound such as $q = 2^{126.23}$ [25]. Moreover, it follows double key scheduling including 1/2 efficiency-rate. The scheme of Hirose delivers marginal security bound as $q = 2^{124.55}$ but it ensures a single key scheduling. However, the CR and PR bound of the Tandem-DM and Abreast-DM are not satisfactory as that of the Weimar, and Hirose [25, 39]. Moreover, the efficiency-rate of Tandem-DM and Abreast-DM is 1/2 like Weimar, and Hirose [28, 29]. Though the scheme of Nandi is bounded by $q = O\left(2^{2n/3}\right)$ but it provides higher efficiency-rate ($r = 2/3$) [64]. Additionally, the construction of ISA09 provides better efficiency-rate ($r = 2/3$) [65]. According to the above discussions and Table 3.1, most of the existing schemes have rigorous security margin. However, the efficiencies are low for the constructions of Weimar, Hirose, Tandem and Abreast. On the other hand, the schemes of Nandi and ISA09 satisfies higher efficiency-rate. Moreover, the constructions of Nandi and ISA09 satisfies $KS = 3$ and $\#E = 3$ [64, 65]. On the contrary, the OM is serial for Nandi and ISA09 schemes. Thus, the overall efficiencies are not adequate for the ISA09 and Nandi schemes. Under these circumstances, there is a scope to provide an efficient scheme of compression function.

Under the (n, n) block-cipher compression function there are many constructions such as MDC-2, MDC-4, MJH, Bart-12, and SKS-15 (Table 3.2). Generally, certain features

Table 3.2: Comparison of the Existing Familiar Compression Function: Based on (n, n) block-cipher [27, 38, 56, 57, 59, 70]

	CF	r	$\#E$	**KS**	**CR**	**PR**	**OM**	**PF**	**RM**
MDC-2	$3n \to 2n$	1/2	2	2	$\mathcal{O}\left(2^{\frac{n}{2}}\right)$	$\mathcal{O}(2^n)$	P	×	176×2
MDC-4	$3n \to 2n$	1/4	4	4	$\mathcal{O}(2^{\frac{5n}{8}})$	$\mathcal{O}\left(2^{\frac{5n}{4}}\right)$	P	×	176×4
MJH	$3n^{+c} \to 2n$	1/2	2	1	$\mathcal{O}\left(2^{\frac{n}{2}}\right)$	$\mathcal{O}(2^n)$	P	×	176×2
Bart-12	$3n \to 2n$	1/3	3	3	$\mathcal{O}(2^n)$	$\mathcal{O}\left(2^{\frac{3n}{2}}\right)$	S	×	176×3
SKS	$3n \to 2n$	1/3	3	1	$\mathcal{O}(2^n)$	$\mathcal{O}(2^{2n})$	P	×	176×3

CF: Compression function, KS: Key Scheduling
CR: Collision resistance, PR: Preimage resistance
$\#E$: Number of blockcipher calls
r: Efficiency rate
OM: Operational mode, P: Parallel, S: Serial
PF: Padding free, RM: Required memory in bytes

are used to identify the better (n, n) block-cipher compression function. We summarize those features and make two groups of efficiency and security. The group of efficiency has certain sub-branches such as key scheduling, number of block-ciphers call, operational mode and efficiency-rate. On the contrary, the security group is focused for collision resistance (CR), preimage resistance (PR), and padding oracle attack. Initially, we address the point of efficiency (Table 3.2). Hence, there is a scope to reduce the storage for key scheduling. Moreover, the less call of block-cipher utilizes less memory resources. Additionally, the efficiency-rate should be close to 1. On the other hand, the parallel mode scheme is suitable for faster operation. The current familiar schemes of MDC-2, MDC-4, Bart, and MJH have lower efficiency-rate (Table 3.2). In certain cases, the numbers of block-cipher call are high. Moreover, the key scheduling is high also (Table 3.2). Thereafter, we point out the issue of security. The current schemes of MDC-2, MDC-4, MJH have lower collision and preimage security bound. Furthermore, the Bart-12 and SKS-15 have higher security bound. However, the efficiency-rate of Bart-12 and SKS-15 is not satisfactory. Interestingly, the current familiar schemes need padding mechanism for variable size of message $(n \neq m)$. Hence, the schemes of MDC-2, MDC-4, MJH, Bart-12 and SKS-15 are not risk free from padding oracle attack [41, 75, 76]. On the contrary, supporting short and flexible message encryption under different platform are important features for the resource constrained device, and IoT-end device [83, 84, 85, 86, 87, 88, 89]. Thus the block-cipher compression function should have capability to encrypt short message. However, the current schemes can not deal variable message encryption without padding because of internal constructions. Furthermore, the security bound of the existing schemes is based on block-length (fixed size, e. g. n-bits) rather than the flexible size of message.

3.2 Previous Works in Authenticated Encryption

Authenticated encryption plays very vital role in encryption and authentication of message [19, 20, 21, 22]. There are many schemes under the authenticated encryption

[11, 12, 19, 20, 21, 22]. We classified two types of AE. The first one is defined as probabilistic-IV-based AE or IV-based AE in short as an authenticated encryption algorithm that has a random IV without associated data whose security goal is indistinguishability with random bits with respect to an adaptive-chosen-plaintext-and-known-IV attack, unlike the common security goal of AE with nonce and associated data, which is indistinguishability with an adaptive-chosen-plaintext-and-IV attack. And the second one is nonce based authenticated encryption. Moreover, nonce based authenticated encryption is classified into two group in respect of security notion such as nonce reuse and nonce respect. Generally, IV-based authenticated encryption is efficient for encryption process under the resource constrained device due to its weaker security model. Most recently, majority of the works of authenticated encryption are based on nonce and associated data such as McOE, COPA, COBRA, OTR, CLOC, PoE, and SILC [11, 12, 43, 46, 47, 77, 92]. These schemes need extra overhead cost to withstand a strong adversary that can choose nonce. Additionally, a finite field multiplication is used in the construction of the OCB and OTR. Hence, the actual efficiency is decreased for the OTR schemes [47]. The privacy security of the block-cipher based message authenticated encryption is $O\left(2^{n/2}\right)$ for most of the constructions [47]. Moreover, many constructions need padding mechanism for encryption of arbitrary (variable) length of message. However, the padding mechanism itself has certain drawbacks such as padding oracle attack, and length extension attack [75, 76]. In addition, a bunch of sensors, actuators, and RFID-tags are active in the domain of IoT. The existing familiar constructions use various kind of primitive such as block-cipher, scratch, ideal permutation, hash for component function of encryption. For example, McOE-X has two variants of McOE-D and McOE-G which are based on respectively THC-CBC and HCBC-2 [11]. Moreover, the OAE$(1, 2)$ and PoE used block-cipher also [12, 42]. The construction of F.PRF used tweak-able block-cipher in the component function and the APE used ideal permutations [21, 78]. In addition, the schemes of COBRA, and OTR used feistel block-cipher network [47, 77]. Therefore, there is a scope for proposing a new variant of primitive in component function for the message authenticated encryption where authenticated encryption runs under IV instead of nonce and associated data. Moreover, it can satisfy low resources for authentication mode.

Under the category of nonce and associated data based authenticated encryption there are bunch of schemes. Interestingly, many of recently proposed schemes claim security in a nonce-reuse scenario [11, 12]. The scheme of McOE brings a breakthrough in the domain of nonce reuse AE [12]. Thereafter, a bunch of schemes have been proposed based on the properties of the McOE such as COPA, PoE, APE, and ELmE (Table 3.3). However, Hoang et. al. showed that the concept of nonce reusing is no more secure for any online authentication scheme [12]. In addition, Hoang et. al. claimed that the online characteristic is a parameter of efficiency. Therefore, a window is re-opened for on-line and nonce respect AE. Furthermore, the McOE needs block-cipher inversion and it's privacy security is bounded by $O\left(2^{n/2}\right)$. Most recently, there are two more proposals such as CLOC and SILC [43, 46]. The constructions of CLOC and SILC are good for short input. Additionally, these two schemes are free from block-cipher inversion. However, the operation mode of CLOC and SILC is serial.

In addition, numbers of calling function or block-cipher is very important to evaluate the efficiency of authenticated encryption. Usually, numbers of calling function or block-cipher means the required numbers of function for encrypting a chunk of message and cipher-text [45, 46]. For example, the scheme of OTR [47] needs $a+m$ calls of block-cipher

Table 3.3: Certain Existing Researches on Authenticated Encryption

S. N.	O.M	$\#Bc$	r	N1	N2	M. Merits
COPA [77]	Parallel	$a+2m+2$	2	-	Y	Parallel, Supports Nonce misuse
PoE [12]	Non-Sequential	$(m \times 2HF)^* + m$	-	-	Y	Non-Sequential, CCA Secure
COBRA [92]	Parallel	$(m+GF)^* + 1 + 2$	-	-	Y	Parallel, inverse freeness of block-cipher
McOE [11]	Serial	$(m+1)^*$	1	Y	Y	Supports Nonce Reuse Security Notion
CLOC [43]	Serial	$a+2m+1$	2	Y	-	Optimized Resources, Suitable for Short Input
SILC [46]	Serial	$a+2m+3$	2	Y	-	Reduces Resources than CLOC like Tweaking function
OTR [47]	Parallel	$(a+m)^*$	1	Y	-	Parallel, Rate 1
APE [78]	Serial	$(a+m)^*$	-		Y	Permutation based, Nonce Reuse, Suitable for Constrained device
ElmE [93]	Parallel	$a+2m+1$	2	-	Y	Fully Parallel, Nonce Reuse
EAX [94]	Serial	$a+m+N$	2	Y	-	Simpler, Efficient than CCM

1. S. N.: Scheme Name, O.M: Operational Mode, r: Efficiency-rate,
2. $\#Bc$: Number of calling function (depends on number of associated data, message),
3. N1: Nonce respect, N2: Nonce reuse, M. Merits: Main Merits,
4. a, m, N: Number of Associated-data, Message, and Nonce
5. HF: Universal hash function, GF: Finite Field Multiplication
6. $*$: May varies

or function. According to the construction of [47], each chunk of message (m) needs one call of block-cipher or function. In addition, one call of block-cipher is required to encrypt one chunk of associated-data (a). Therefore, the total number of required block-ciphers is $a+m$. Generally, #Bc is evaluated through the above way. Moreover, the efficiency-rate (r) is defined as r = message size/numbers of function $\times$ function-length. Most recently, block-cipher inverse freeness becomes an important criteria for any authenticated encryption. It reduces the resource-cost for computing inversion of block-cipher/permutation. Therefore, there are certain scopes to propose nonce-respect authenticated encryption that can satisfy less call of block-cipher. In addition, it can satisfy inverse freeness of block-cipher.

3.3 Previous Works in Small Domain Encryption

Small domain encryption is one of the blazing issue in the arena of modern cryptography [30, 31, 32, 33]. Usually, block-cipher is used to develop a scheme of small domain encryption. There are certain traditional block-cipher such as AES/DES. Generally, these are suitable as a primitive for big size of data encryption. However, the situation is quite different for message encryption under the resource constrained devices in real life application [35, 36, 51]. Usually, a small chunk of message needs to encrypt for example 8, 16, 24, 32 bits. Generally, the key size of AES/DES is 128, 192 or 256 bits. Even for lightweight-cipher, the key size is 32, 48, 64, 96 bits [7, 80, 81]. Under this circumstance, key and storage management cost are increased. Therefore, the concept of SDE is important for resource constrained devices. However, the security is low under the SDE because of the small size of the key. On the contrary, the efficiency is higher. In certain cases, the implementation of resource constrained devices depends on the speed, memory-utilization, power consumption, and number of gate operation rather than the rigorous security bound in certain cases.

In 2002, J.Black and P.Rogaway formally addressed the issue of small domain encryption, including format preserving encryption for the first time in [31]. Following that B. Morris, P. Rogaway, T. Stegers proposed a construction of [32]. The basic primitive of this scheme is block-cipher e. g. AES/DES. The encryption time of this scheme is $O\left(\log^3 N\right)$ [34]. In addition, it needs a certain number of calling AES block-ciphers. Later, another construction was proposed by V. T. Hoang, B. Morris, and P. Rogaway in [34]. This scheme satisfies small domain message encryption. In addition, it also satisfies the format preserving encryption. It uses Swap-or-Not shuffle algorithm [33, 34]. However, the basic primitive is block-cipher as well. It also needs a large number of calling block-cipher (e. g. AES). The security margin of this scheme is $q = (1-\varepsilon)\,2^n$ (q : number of query, n : block-length) [33, 34, 35]. In the next year, T. Ristenpart and S. Yilek proposed a scheme of Mix-and-Cut [33]. Interestingly, it is also based on a card shuffling algorithm where the basic primitive is a block-cipher. Moreover, the encryption time of this scheme is $O\left(\log^2 N\right)$ [33, 34, 35]. Most recently, the scheme of Sometimes-Recursive shuffle has been proposed by B. Morris, P. Rogaway [35]. Authors of the SRS-construction re-conceptualized the scheme of Mix-and-Cut [33]. In addition, the SRS is the best scheme in respect of low encryption time ($O\left(\log N\right)$) and full security margin [35]. However, it needs 1000 calls of AES block-ciphers. Authors of [35] claimed that the resources are required to call 1000 of AES is 80K clock cycles, or 25 μsec, on a recent Intel processor. Hence, the SRS is also heavy for resource-constrained device and IoT-end device. According to the above discussions, the mentioned constructions are broadly classified into two groups such as partial security margin and full security margin. The schemes of Thorp-shuffle, and Swap-or-Not are under the partial security margin group. On the contrary, Knuth-shuffle, Mix-and-Cut, and SRS are the member of the full security margin group. Most recently, two more proposals are available such as FNR and BPS [30, 36]. In addition, the scheme of FNR is based on classical Feistel structure where Galois Field $\mathrm{GF}(2^n)$ is used. Furthermore, the scheme of BPS is based on tweak block-cipher and it uses general Feistel structure [30]. Under these circumstances, there is an opportunity to propose a scheme of small domain encryption that can satisfy less resource. In addition, it can provide better efficiency.

Chapter 4

Some Probable Secure Constructions of Compression Function (CF)

A cryptographic hash is an important tool in the area of a modern cryptography [25, 39, 53, 59]. It comprises a compression function (CF), where the compression function can be built by a scratch or block-cipher. There are some familiar schemes of block-cipher compression function such as Weimar, Hirose, Tandem, Abreast, Nandi, ISA-09 [25, 28, 29, 39, 64, 65]. Interestingly, the security proof of all the mentioned schemes are based on the ideal cipher model (ICM), which depends on ideal environment [61, 67]. Therefore, it is desired to use such a proof technique model, which is closed to the real world such as weak cipher model (WCM). Hence, we propose our first scheme that is based on $(n, 2n)$ block-cipher compression function. In addition, it is secure under the ideal cipher model, weak cipher model and extended weak cipher model (ext. WCM). Additionally, the majority of the existing schemes need multiple key schedules, where the proposed first scheme and the Hirose-DM follow single key scheduling property. The efficiency-rate of our scheme is $r = 1/2$. Moreover, the number of block-cipher calls of the first scheme is 2 and it runs in parallel mode.

A constrained device is an emerging technology that has enormous applications in our daily life such as access control, inventory control, luggage tracking, bar-code reader, and IoT [14, 37, 40, 71, 72, 80]. However, it has certain drawbacks of low memory and less computing power [37, 53, 55, 56, 68, 74]. Thus, one of the cracking challenge is to provide efficient and secure cryptographic solution for the constrained device in the aspect of security issue. An (n, n) block-cipher based cryptographic compression function is applicable to provide provable security to the constrained device. Though, there are many constructions of (n, n) block-cipher such as MDC-2, MDC-4, MJH, Bart-12, and SKS-15 [29, 38, 59, 79]. However, most of the familiar schemes are not suitable for short and variable message encryption without padding due to internal structures. Furthermore, the security margin is provided based on block length rather than the flexible message size. Therefore, we present two different (n, n) block-cipher compression function schemes. The second scheme (SS) satisfies better efficiency such as less call of block-cipher, less key scheduling, and higher efficiency-rate. On the contrary, the third scheme (TS) has upper security bound. Moreover, both of the schemes are suitable for small and variable message encryption (message size = tn, such that $t < 1, n :$ block length), which is handy for the constrained device.

Under the (n, n) block-cipher compression function, most of the schemes are secure

under the ideal cipher model. In addition, the scheme of MJH and Bart-12 are secure under the Finite-Field-Multiplicative model. Both of the models are based on strong assumptions. Hence, we propose an (n, n) block-cipher based compression function which is secure under the weak cipher model. In addition, we also show that the proposed fourth scheme is secure under the ideal cipher model. Generally, weak cipher model has less strict assumptions than that of the ideal cipher model. Therefore, WCM is more close to the real world.

4.1 An Upper Bounded Secure Scheme of CF

Our proposed first scheme is secure under the three types of security model. Secondly, it follows single key scheduling and it's number of block-cipher call is 2. Additionally, the efficiency rate of proposed scheme is 1/2. We compare our first scheme of $(n, 2n)$ block-cipher based compression function with existing familiar schemes in Table 4.1.

Table 4.1: Comparison among the First Scheme and Existing Familiar Schemes [25, 28, 29, 39, 56, 64, 65]

				Security Proof Technique		
Scheme Name	**CF**	**KS**	r	**ICM**	**WCM**	**ext. WCM**
First Scheme	$3n \to 2n$	1	1/2	$\surd$	$\surd$	$\surd$
Weimar	$3n \to 2n$	2	1/2	$\surd$	N.Y.	N.Y.
Hirose	$3n \to 2n$	1	1/2	$\surd$	N.Y.	N.Y.
Abreast	$3n \to 2n$	2	1/2	$\surd$	N.Y.	N.Y.
Tandem	$3n \to 2n$	2	1/2	$\surd$	N.Y.	N.Y.
Nandi	$4n \to 2n$	3	2/3	$\surd$	N.Y.	N.Y.
ISA-09	$4n \to 2n$	3	2/3	$\surd$	N.Y.	N.Y.

CF = Compression function, N.Y. = Not yet
KS =Key scheduling, Efficiency rate = r
ICM, WCM, ext.WCM = Ideal, Weak, extended weak (cipher model)

4.1.1 Proposed First Scheme of Compression Function (FS)

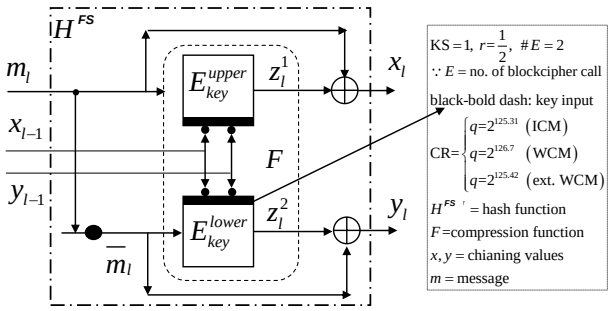


Figure 4.1: Block Diagram of the First Proposed Scheme (FS)

The proposed first scheme follows two calls of block-cipher call, where key scheduling is single. It satisfies the class of $(n, 2n)$ block-cipher because of key size is double of the block-length. It runs under the Matyas Meyer Oseas mode (MMO). The definition and block diagram of this scheme are notified in Definition 4.1 and Figure 4.1.

Definition 4.1. Let $E \in Block_n^k$ be a block-cipher, where k = key length and n = block length. The H^{FS} is a hash that is constructed by F. Let $F = \{0,1\}^n \times \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ be a block-cipher based compression function. In this scheme, two independent block-ciphers are used for a single iteration such as E_{key}^{upper} and E_{key}^{lower}. Therefore, the final output of H^{FS} is:

$$H^{FS}(x_{l-1}, y_{l-1}, m_l) = x_l,\ y_l \text{ such that,}$$
$$x_l = z_l^1 \oplus m_l, y_l = z_l^2 \oplus \bar{m}_l \Big[\text{where, } z_l^1 \leftarrow E^{upper}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(m), z_l^2 \leftarrow E^{lower}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(\bar{m})\Big]$$

In addition, we proposed a new security proof model of ext.WCM. In this section, we define this new security model. An Extended weak cipher model (ext.WCM) follows the basic properties of the ICM and WCM. It adds a new feature for making the adversary powerful, where the adversary can ask any type of query for a single instance (Table 4.2 and Figure 4.2). Additionally, the assumptions of the ext.WCM is weaker than the ICM and WCM. Under the ext.WCM, the adversary gets a set of message and corresponding encrypted message (ciphertext) based on a key. However, the query process is based on non-adaptive. The block-cipher oracle is defined as ${}^{ext.}WCM^{k,m,c}(\cdot)$, where the adversary can ask and gets a set of key (k), message (m) and cipher-text (c).

Table 4.2: Operation Characteristics of the Security Proof Model

$\mathcal{A} \rightarrow^{allowed}$ for Game. Such that, **Game** (G)**:** **Input:** $[x, x']$ **where,** $x \neq x' \wedge H(x) = H(x')$	ICM	WCM	ext. WCM
	Allow, $\mathcal{A} \rightarrow E^f/E^b$ for G	Allow, $\mathcal{A} \rightarrow E^f$ for G then, $\mathcal{A} \rightarrow E^b$ for G then, $\mathcal{A} \rightarrow E^k$ for G	Allow, $\mathcal{A} \rightarrow E^f/E^b//E^k$ for G

Security Proof of Collision Resistance of the First Scheme (ICM Based)

An adversary $\mathcal{A}$ can make two types of query such as forward query (E^f) and backward query (E^b) [39, 61, 62, 63]. Under the ICM, a game is defined as $Game_{ICM}^{coll}$ (Algorithm 1), where adversary $\mathcal{A}$ tries to find (x, y, m) and (x', y', m'). Therefore, the adversary gets success iff $H^{FS}(x, y, m) = H^{FS}(x', y', m')$ where, $(x, y, m) \neq (x', y', m')$. In addition, the $Game_{ICM}^{coll}$ is categorized into three sub-games (Table 4.3). Adversary $\mathcal{A}$ runs through these three subgames for getting success. However, the first subgame stands for dual queries. Under the first subgame, adversary tries to find two different queries for a collision. Secondly, the subgame of $subGame_{sole,ICM}^{coll}$ is responsible for finding a collision within a single query. Finally, a collision through initial chaining values are occurred by the third subgame.

Theorem 4.1. Let H^{FS} be a two calls of $2n$ bit key block-cipher compression function. The task of adversary $\mathcal{A}$ is to find collision under the compression function F (H^{FS}).

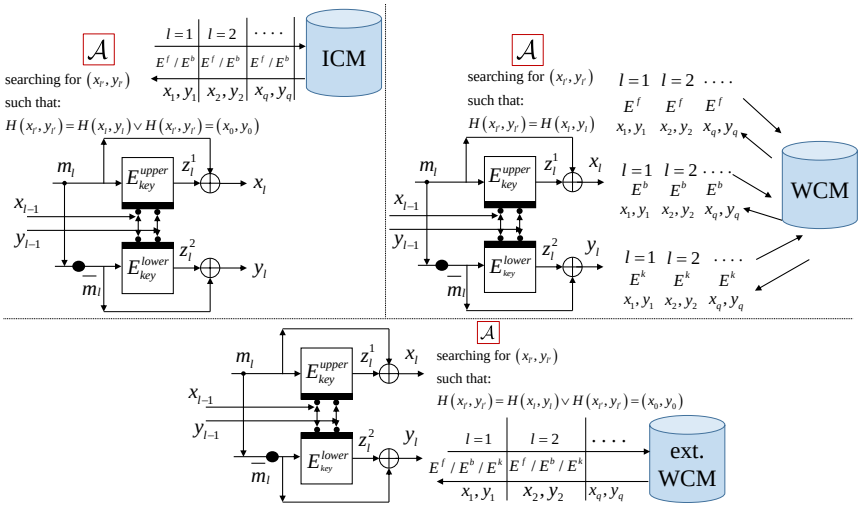


Figure 4.2: Security Proof Model

Table 4.3: Branches of $Game^{coll}_{ICM}$

Branch name	Condition
$subGame^{coll}_{dual,ICM}$	$(x_l, y_l, m_l) \neq (x_{l'}, y_{l'}, m_{l'}) \wedge$ $H^{NEW}(x_l, y_l, m_l) = H^{NEW}(x_{l'}, y_{l'}, m_{l'})$
$subGame^{coll}_{sole,ICM}$	$x_l = y_l$, when $H^{NEW}(x_{l-1}, y_{l-1}, m_l) = (x_l, y_l)$
$subGame^{coll}_{pri,ICM}$	$(x_l, y_l) = (x_0, y_0)$, when $H^{NEW}(x_{l-1}, y_{l-1}, m_l) = (x_l, y_l)$

Hence, the advantage of $\mathcal{A}$ is bounded after q pairs of queries as:

$$\mathrm{Adv}^{ICM^{coll}}_{H^{FS}}(q) \leq \frac{q^2 + q}{(2^n - 2q)^2} + \frac{2q}{(2^n - 2q)}$$

Proof. Let an adversary $\mathcal{A}$ can ask any relevant query and never makes any duplicate query through E^f or E^b. It can ask upto l-th queries, where $l \leq q$.

$subGame^{coll}_{dual,ICM}$. Adversary $\mathcal{A}$ uses the ICM oracle for E^f or E^b query. At first, the adversary checks whether the most recent query is collide with the previous any queries or not. Let the current iteration is l, where the outputs are x_l, y_l. For example, $l'|(l' < l \leq q)$ is previously executed any iteration and the corresponding output are $x_{l'}, y_{l'}$. If $(x_l, y_l) = (x_{l'}, y_{l'})$ is satisfied then a trigger will be defined and the $subGame^{coll}_{dual,ICM}$ will be over. Otherwise, the adversary $\mathcal{A}$ stores the value of x_l, y_l into the query database $(\mathcal{Q})$ and goes for next iteration. Let the outcome of l'-th iteration are $x_{l'} \leftarrow E^{upper}_{\bar{x}_{l'-1}||\bar{y}_{l'-1}}(m_{l'}) \oplus m_{l'}$ and $y_{l'} \leftarrow E^{lower}_{\bar{x}_{l'-1}||\bar{y}_{l'-1}}(\bar{m}_{l'}) \oplus \bar{m}_{l'}$. For an iteration of $l|(l' < l \leq q)$, the output are $x_l \leftarrow E^{upper}_{\bar{x}_{l-1}||\bar{y}_{l-1}}(m_l) \oplus m_l$ and $y_l \leftarrow E^{lower}_{\bar{x}_{l-1}||\bar{y}_{l-1}}(\bar{m}_l) \oplus \bar{m}_l$. If $(x_{l'}, y_{l'})$ and (x_l, y_l) collides each other then a trigger will be defined as $tri^{coll}_{dual,ICM}$. However, the x_l, y_l come from the set size $2^n - (2l - 2)$ and $2^n - (2l - 1)$. Hence, under the trigger of $tri^{coll}_{dual,ICM}$ the probability will be

$l-1/(2^n-(2l-2))\times(2^n-(2l-1))$. More explicitly, under the $subGame^{coll}_{dual,ICM}$ through $tri^{coll}_{dual,ICM}$, the following states are responsible for collision:

$$\{(x_l = x_{l'}) \wedge (x_l = y_{l'})\} \vee \{(y_l = y_{l'}) \wedge (y_l = x_{l'})\} \tag{4.1}$$

where,

$$x_l = E^{upper}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(m_l) \oplus m_l, x_{l'} = E^{upper}_{\bar{x}_{l'-1}\|\bar{y}_{l'-1}}(m_{l'}) \oplus m_{l'}$$
$$y_l = E^{lower}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(\bar{m}_l) \oplus \bar{m}_l, y_{l'} = E^{lower}_{\bar{x}_{l'-1}\|\bar{y}_{l'-1}}(\bar{m}_{l'}) \oplus \bar{m}_{l'}$$

Therefore, the probability of collision under the l-th query will be $\Pr\left[Tri^{coll}_{dual,ICM}\right] = \Pr\left[tri^{coll}_{2,dual,ICM}, \cdots, tri^{coll}_{q,dual,ICM}\right]$, which implies that,

$$\sum_{l=2}^{q} \Pr[Tri^{coll}_{l,dual,ICM}] = \sum_{l=2}^{q} \frac{2(l-1)}{(2^n-2l-2)(2^n-2l-1)} \leq \sum_{l=2}^{q} \frac{2(l-1)}{(2^n-2l)^2} \leq \frac{q^2}{(2^n-2q)^2} \tag{4.2}$$

$subGame^{coll}_{sole,ICM}$. The $subGame^{coll}_{sole,ICM}$ is responsible for finding a collision within l-th iteration of query, where $l \leq q$. Assume that, the output are x_l and y_l at the point of l-th iteration. Therefore, there is a chance for creating a collision when $x_l = y_l$. If collision occurs, a trigger $\left(tri^{coll}_{sole,ICM}\right)$ will be called. Therefore, the probability of collision under the subgame $\left(subGame^{coll}_{sole,ICM}\right)$ through $tri^{coll}_{sole,ICM}$ is $\Pr\left[Tri^{coll}_{sole,ICM}\right] = \Pr\left[tri^{coll}_{1,sole,ICM}, tri^{coll}_{2,sole,ICM}, .., tri^{coll}_{q,sole,ICM}\right]$. After q pairs of queries, it implies that,

$$\sum_{l=1}^{q} \Pr[Tri^{coll}_{l,sole,ICM}] = \sum_{l=1}^{q} \frac{1}{(2^n-2l-2)(2^n-2l-1)} \leq \sum_{l=1}^{q} \frac{1}{(2^n-2l)^2} \leq \frac{q}{(2^n-2q)^2} \tag{4.3}$$

$subGame^{coll}_{pri,ICM}$. Usually, the initial vectors or chaining values need to provide at the beginning of encryption process. Therefore, the generated output can be collide with the initial or primary values at the any phase of l. For example, in the iteration of $l\,|(l \leq q)$, the outcome are $x_l = E^{upper}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(m_l) \oplus m_l$ and $y_l = E^{lower}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(\bar{m}_l) \oplus \bar{m}_l$. If collision occurs for x_0, y_0 and x_l, y_l, a trigger will be defined as $tri^{coll}_{pri,ICM}$ and query process will be terminated from the $subGame^{coll}_{pri,ICM}$. Hence, the probability of collision under l-th query will be $\Pr\left[Tri^{coll}_{pri,ICM}\right] = \Pr\left[tri^{coll}_{1,pri,ICM}, tri^{coll}_{2,pri,ICM}, .., tri^{coll}_{q,pri,ICM}\right]$. After q pairs of queries, it implies that,

$$\sum_{l=1}^{q} \Pr[Tri^{coll}_{l,pri,ICM}] = \sum_{l=1}^{q} \frac{2}{(2^n-2l)} \leq \frac{2q}{(2^n-2q)} \tag{4.4}$$

Adding the value of 4.2, 4.3 and 4.4, **Theorem 4.1** is satisfied.

Security Proof of Collision Resistance of the First Scheme (WCM Based)

An adversary $\mathcal{A}$ will make an additional query E^k with E^f and E^b under the WCM, where E^k is defined as a key-disclosure query [62]. According to the WCM, the adversary $\mathcal{A}$ is able to make any relevant query with non-repetition. A $Game^{coll}_{WCM}$ (Algorithm 2) will be defined for finding collision under the WCM. The target of the adversary $\mathcal{A}$ is to find X, Y such that $H(X) = H(Y)$, where X, Y = input, H = hash outout. Additionally, the $Game^{coll}_{WCM}$ will be classified into three subgames (Table 4.4), where $subGame^{coll}_{forw(E^f),WCM}$

Algorithm 1 ($Game^{coll}_{ICM}$)

Initailization : $l = 0,\ q = 2^n,\ \mathcal{Q}$: Empty query database
procedure $Game^{coll}_{ICM}$
    $Execution$: E^f or E^b
    $Answer$: from ICM oracle
    $E^f/E^b \to x_l = \left(z^1_l \oplus m_l\right) = E^{upper}_{\bar{x}_{l-1}||\bar{y}_{l-1}}\left(m_l\right) \oplus m_l$
    $E^f/E^b \to y_l = \left(z^2_l \oplus \bar{m}_l\right) = E^{lower}_{\bar{x}_{l-1}||\bar{y}_{l-1}}\left(\bar{m}_l\right) \oplus \bar{m}_l$
    switch (input) **do**
        case 1
            `assert`$(subGame^{coll}_{dual,ICM})$
            if $l' < l \leq q$ **then**
                searching for $(x_{l'}, y_{l'})$ from $\mathcal{Q}$
                if $\{(x_l, y_l) = (x_{l'}, y_{l'})\} \to \mathcal{A}$ wins **then**
                    call: collision event $tri^{coll}_{dual,ICM}$ and break: from $subGame^{coll}_{dual,ICM}$
                end if
            else
                store: $(x_l, y_l) \to \mathcal{Q}$
            end if
        case 2
            `assert`$(subGame^{coll}_{sole,ICM})$
            if $\{(l \leq q) \wedge (x_l = y_l)\} \to \mathcal{A}$ wins **then**
                call: collision event $tri^{coll}_{sole,ICM}$ and break:from $subGame^{coll}_{sole,ICM}$
            else
                store: $(x_l, y_l) \to \mathcal{Q}$
            end if
        case 3
            `assert`$(subGame^{coll}_{pre,ICM})$
            if $\{(l \leq q) \wedge (x_l, y_l) = (x_0, y_0)\} \to \mathcal{A}$ wins **then**
                call: collision event $tri^{coll}_{pri,ICM}$ and break: from $subGame^{coll}_{pri,ICM}$
            else
                store: $(x_l, y_l) \to \mathcal{Q}$
            end if
end procedure

Table 4.4: Branches of $Game^{coll}_{WCM}$

Branch name	Condition
$subGame^{coll}_{forw(E^f),WCM}$	$E^f \to (x, y, m) \neq (x', y', m')$ $\wedge H^{NEW}(x, y, m) = H^{NEW}(x', y', m')$
$subGame^{coll}_{back(E^b),WCM}$	$E^b \to (x, y, m) \neq (x', y', m')$ $\wedge H^{NEW}(x, y, m) = H^{NEW}(x', y', m')$
$subGame^{coll}_{key(E^k),WCM}$	$E^k \to (x, y, m) \neq (x', y', m')$ $\wedge H^{NEW}(x, y, m) = H^{NEW}(x', y', m')$

is defined for finding collision through cipher-text and $subGame^{coll}_{back(E^b),WCM}$ is used for exploring plain-text. Additionally, the adversary $\mathcal{A}$ will execute the game of $subGame^{coll}_{key(E^k),WCM}$ for getting collision through the key-disclosure query.

Theorem 4.2. Let H^{FS} be a two calls of $2n$ bit key block-cipher hash function. It invokes the block-cipher based compression function F, where the advantage of the adversary $\mathcal{A}$ is to find collision under H^{FS} (F). Therefore, the adversarial advantage is bounded after q pairs of queries as:

$$\mathrm{Adv}^{WCM^{coll}}_{H^{FS}}(q) \leq \frac{3q(q-1)}{2^{2n}}$$

Proof. Let $\mathcal{A}$ be the adversary that can make query upto l-th queries, where $l \leq q$. The collision probability of these three subgames will be evaluated under the adversary $\mathcal{A}$ in the following way.

Algorithm 2 $(Game^{coll}_{WCM})$

Initailization : $l = 0$, $q = 2^n$, $\mathcal{Q}$: Empty query database
procedure $Game^{coll}_{WCM}$
    run: $subGame^{coll}_{forw,WCM}$, $subGame^{coll}_{back,WCM}$ and $subGame^{coll}_{key,WCM}$
    function $subGame^{coll}_{forw(E^f),WCM}$
        for $(l \leq q)$ **do**
            run an oracle (E^f) from WCM
            reply:
            $E^f \rightarrow x_l = \left(z^1_l \oplus m_l\right) = E^{upper}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(m_l) \oplus m_l$
            $E^f \rightarrow y_l = \left(z^2_l \oplus \bar{m}_l\right) = E^{lower}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(\bar{m}_l) \oplus \bar{m}_l$
            Check for collision hit event:
            if $l' < l \leq q$ **then**
                searching for $(x_{l'}, y_{l'})$ from $\mathcal{Q}$
                if $(x_l, y_l) = (x_{l'}, y_{l'}) \rightarrow$ Adversary wins **then**
                introduce event $tri^{coll}_{E^f,WCM}$ and terminate from $subGame^{coll}_{forw(E^f),WCM}$
                end if
            else
                keep: $(x_l, y_l) \rightarrow \mathcal{Q}$
            end if
        end for
    end function
    function $subGame^{coll}_{back(E^b),WCM}$
        run an oracle (E^b) from WCM
        do same procedure as $subGame^{coll}_{forw(E^f),WCM}$ but use a different oracle
    end function
    function $subGame^{coll}_{key(E^k),WCM}$
        run an oracle (E^k) from WCM
        do same procedure as $subGame^{coll}_{forw(E^f),WCM}$ but use a different oracle
    end function
end procedure

$subGame^{coll}_{forw(E^f),WCM}$. The adversary $\mathcal{A}$ will execute the $subGame^{coll}_{forw(E^f),WCM}$, where a forward query returns the query result and stores a couple of output into the $\mathcal{Q}$. There are three basic phases under the $subGame^{coll}_{forw(E^f),WCM}$ such as making query, checking and trigger/store. In the first phase, the adversary is allowed to make query through E^f under the WCM. Then in second phase, $\mathcal{A}$ checks whether the last output pair collides with the previous any query pair. The third phase depends on the second phase where a trigger will be called if collision occurs. On the contrary, the output pair will be stored into $\mathcal{Q}$ and the adversary will be allowed for next query. For example, the adversary $\mathcal{A}$ gets a couple of outputs $(x_{l'}, y_{l'})$ at the l'-th iteration. Let there is an another iteration of $l|\,(l' < l)$, where output pair will be x_l, y_l. If $(x_l, y_l) = (x_{l'}, y_{l'})$ then a collision will be occurred and a trigger $\left(tri^{coll}_{E^f,WCM}\right)$ will be called. However, the sets of queries are:

$$E^f \rightarrow^{WCM} (l' < q) : x_{l'} = z^1_{l'} \oplus m_{l'} = E^{upper}_{\bar{x}_{l'-1},\bar{y}_{l'-1}}(m_{l'}) \oplus m_{l'}, y_{l'} = z^2_{l'} \oplus \bar{m}_{l'} = E^{lower}_{\bar{x}_{l'-1},\bar{y}_{l'-1}}(\bar{m}_{l'}) \oplus \bar{m}_{l'}$$
$$E^f \rightarrow^{WCM} (l' < l < q) : x_l = z^1_l \oplus m_l = E^{upper}_{\bar{x}_{l-1},\bar{y}_{l-1}}(m_l) \oplus m_l, y_l = z^2_l \oplus \bar{m}_l = E^{lower}_{\bar{x}_{l-1},\bar{y}_{l-1}}(\bar{m}_l) \oplus \bar{m}_l$$

Hence, the conditions of collision are:

$$\left\{ \begin{array}{l} \left(z^1_{l'} \oplus m_{l'} = z^1_l \oplus m_l\right) \vee \\ \left(z^2_{l'} \oplus \bar{m}_{l'} = z^1_l \oplus m_l\right) \end{array} \right\} \wedge \left\{ \begin{array}{l} \left(z^1_{l'} \oplus m_{l'} = z^2_l \oplus \bar{m}_l\right) \vee \\ \left(z^2_{l'} \oplus \bar{m}_{l'} = z^2_l \oplus \bar{m}_l\right) \end{array} \right\} \tag{4.5}$$

From 4.5, the collision probability is:

$$\sum_{l=2}^{q} \Pr\left[Tri^{coll}_{l,E^f,WCM}\right] = \sum_{l=2}^{q} \frac{2(l-1)}{\left(2^n - 2l\right)^2} \leq \sum_{l=2}^{q} \frac{2(l-1)}{\left(2^n\right)^2} \leq \frac{q\,(q-1)}{2^{2n}} \tag{4.6}$$

$subGame^{coll}_{back(E^b),WCM}$. Let the adversary $\mathcal{A}$ will execute the $subGame^{coll}_{back(E^b),WCM}$, where backward query will be provided an output pair. A trigger $\left(tri^{coll}_{E^b,WCM}\right)$ will be defined, if collision occurs. According to this subgame and the previous explanation of the $subGame^{coll}_{forw(E^f),WCM}$, the collision probability is:

$$\sum_{l=2}^{q} \Pr\left[Tri^{coll}_{l,E^b,WCM}\right] = \sum_{l=2}^{q} \frac{2(l-1)}{\left(2^n - 2l\right)^2} \leq \sum_{l=2}^{q} \frac{2(l-1)}{\left(2^n\right)^2} \leq \frac{q\,(q-1)}{2^{2n}} \tag{4.7}$$

$subGame^{coll}_{key(E^k),WCM}$. The explanation of probability of $subGame^{coll}_{key(E^k),WCM}$ is as that of the $subGame^{coll}_{forw(E^f),WCM}$. Therefore, the probability of collision is:

$$\sum_{l=2}^{q} \Pr\left[Tri^{coll}_{l,E^k,WCM}\right] = \sum_{l=2}^{q} \frac{2(l-1)}{\left(2^n - 2l\right)^2} \leq \sum_{l=2}^{q} \frac{2(l-1)}{\left(2^n\right)^2} \leq \frac{q\,(q-1)}{2^{2n}} \tag{4.8}$$

Adding the values of 4.6, 4.7 and 4.8, **Theorem 4.2** is proved.

Security Proof of Collision Resistance of the First Scheme (ext.WCM)

According to the definition of ext.WCM, the adversary $\mathcal{A}$ will make three types of query under a single instance non-adaptively (Table 4.5), where the adversary has no chance for repeated query. A $Game^{coll}_{(E^f,E^b,E^k),ext.WCM}$ (Algorithm 3) will be defined in this section for providing the security proof of the proposed scheme and it is categorized into three subgames with their task into Table 4.5.

Table 4.5: Branches of $Game^{coll}_{(E^f,E^b,E^k),ext.WCM}$

Branch name	Condition
${}^{outer}subGame^{coll}_{(E^f/E^b/E^k),ext.WCM}$	$E^{f,b,k}\rightarrow^{ext.}WCM^{k,m,c}(\cdot) \Rightarrow$ $(x_l,y_l,m_l) \neq (x_{l'},y_{l'},m_{l'}) \wedge$ $H^{NEW}(x_l,y_l,m_l)$ $= H^{NEW}(x_{l'},y_{l'},m_{l'})$
${}^{inner,IV}subGame^{coll}_{(E^f/E^b/E^k),ext.WCM}$	$E^{f,b,k}\rightarrow^{ext.}WCM^{k,m,c}(\cdot) \Rightarrow$ $x_l = y_l$ when, $H^{NEW}(x_{l-1},y_{l-1},m_l) = (x_l,y_l)$ $\vee$ $(x_l,y_l) = (x_0,y_0)$ when, $H^{NEW}(x_{l-1},y_{l-1},m_l) = (x_l,y_l)$ and (x_0,y_0) = initial value

Theorem 4.3. Let H^{FS} be a two calls of $2n$ bit key block-cipher hash function, where it consists of block-cipher compression function F. The advantage of adversary $\mathcal{A}$ is to find collision through H^{FS} (F) after q pairs of queries. Therefore, the adversarial advantage is bounded as:

$$\mathrm{Adv}^{ext.WCM^{coll}}_{H^{FS}}(q) = q^2 - q/2N^2 + 3q/N$$

Proof. Let adversary $\mathcal{A}$ asks any relevant query and never makes any duplicate query through $E^f/E^b/E^k$. Under the ext.WCM model, the query is being asked non-adaptively at first. Therefore, adversary looks for collision based on those executed queries.

${}^{outer}subGame^{coll}_{(E^f/E^b/E^k),ext.WCM}$. The subgame of ${}^{outer}subGame^{coll}_{(E^f/E^b/E^k),ext.WCM}$ will be assigned for finding collision under any iteration of the query process $l|\,(l \leq q)$. For an example, at the point of $l'\,(l' \leq q)$-th iteration, the resultant output are $x_{l'}, y_{l'}$. However, in the iteration of $l|\,(l' < l \leq q)$, the output are x_l, y_l. If the adversary $\mathcal{A}$ finds that there is a collision between $x_{l'}, y_{l'}$ and x_l, y_l then a trigger will be called. Hence, the conditions of collision are:

$$\left(\mathcal{A}\rightarrow^{make\ query}\left(E^f,E^b,E^k\right)\right) \wedge (\text{for two iterations of queries } (l,l')\,|(l' < l \leq q)) \tag{4.9}$$

Furthermore, 4.9 can be derived as:

$$\begin{gathered} z^1_l\left(E^{upper}_{\bar{x}_{l-1},\bar{y}_{l-1}}(m_l)\right) = z^1_{l'}\left(E^{upper}_{\bar{x}_{l'-1},\bar{y}_{l'-1}}(m_{l'})\right) \text{ or } z^1_l\left(E^{upper}_{\bar{x}_{l-1},\bar{y}_{l-1}}(m_l)\right) = z^2_{l'}\left(E^{lower}_{\bar{x}_{l'-1},\bar{y}_{l'-1}}(,\bar{m}_{l'})\right) \\ z^2_l\left(E^{lower}_{\bar{x}_{l-1},\bar{y}_{l-1}}(\bar{m}_l)\right) = z^2_{l'}\left(E^{lower}_{\bar{x}_{l'-1},\bar{y}_{l'-1}}(\bar{m}_{l'})\right) \text{ or } z^2_l\left(E^{lower}_{\bar{x}_{l-1},\bar{y}_{l-1}}(\bar{m}_l)\right) = z^1_{l'}\left(E^{upper}_{\bar{x}_{l'-1},\bar{y}_{l'-1}}(m_{l'})\right) \end{gathered} \tag{4.10}$$

If, 4.10 occurs then ${}^{outer}tri^{coll}_{(E^f,E^b,E^k),ext.WCM}$ will be called. Hence, the probability of collision under the subgame of ${}^{outer}subGame^{coll}_{(E^f/E^b/E^k),ext.WCM}$ is:

$$\Pr\left[{}^{outer}Tri^{coll}_{(E^f,E^b,E^k),ext.WCM}\right] = \Pr\left[{}^{outer}tri^{coll}_{1,(E^f,E^b,E^k),ext.WCM},\ldots,{}^{outer}tri^{coll}_{q,(E^f,E^b,E^k),ext.WCM}\right] \tag{4.11}$$

From 4.11,

$$\sum_{l=1}^{q} \Pr\left[{}^{outer}Tri^{coll}_{l,(E^f,E^b,E^k),ext.WCM}\right] = \sum_{l=1}^{q} \frac{(l-1)}{(2^{2n})} \leq \frac{q^2-q}{2.2^{2n}} \tag{4.12}$$

${}^{inner}subGame^{coll}_{(E^f/E^b/E^k),ext.WCM}$. Let, there is an iteration l, where $l \leq q$. Under the l-th iteration, the output is $z_l^1 = E^{upper}_{\bar{x}_{l-1},\bar{y}_{l-1}}(m_l) \Rightarrow m_l \oplus z_l^1 = x_l$ and $z_l^2 = E^{lower}_{\bar{x}_{l-1},\bar{y}_{l-1}}(\bar{m}_l) \Rightarrow \bar{m}_l \oplus z_l^2 = y_l$.

There is a chance to make collision between x_l and y_l. So, a trigger $\left({}^{inner}tri^{coll}_{(E^f,E^b,E^k),ext.WCM}\right)$ will be called when a collision occurs. Hence,

$$\Pr\left[{}^{inner}Tri^{coll}_{(E^f,E^b,E^k),ext.WCM}\right] = \Pr\left[{}^{inner}tri^{coll}_{1,(E^f,E^b,E^k),ext.WCM}, \cdots, {}^{inner}tri^{coll}_{q,(E^f,E^b,E^k),ext.WCM}\right] \tag{4.13}$$

From 4.13, the collision probability is:

$$\sum_{l=1}^{q} \Pr\left[{}^{inner}Tri^{coll}_{l,(E^f,E^b,E^k),ext.WCM}\right] = \sum_{l=1}^{q} \frac{1}{2^n} \leq \frac{q}{2^n} \tag{4.14}$$

Algorithm 3 $\left(Game^{coll}_{(E^f,E^b,E^k),ext.WCM}\right)$

Initailization : $l = 0$, $q = 2^n$, $\mathcal{Q}$: Empty query database
procedure $\left(Game^{coll}_{(E^f,E^b,E^k),ext.WCM}\right)$
    for $(l \leq q)$ **do**
        Execution: $E^f/E^b/E^k$ through ${}^{ext.}WCM^{k,m,c}(\cdot)$
        Answer from ext. WCM oracle
        $E^f/E^b/E^k \rightarrow x_l = \left(E^{upper}_{x_{l-1}\|y_{l-1}}(m_l) \oplus m_l\right)$
        $E^f/E^b/E^k \rightarrow y_l = \left(E^{upper}_{\bar{x}_{l-1}\|\bar{y}_{l-1}}(\bar{m}_l) \oplus \bar{m}_l\right)$
        Store into $\mathcal{Q}$
    end for
    (∗calling three subgames∗)
    CALL $\rightarrow {}^{outer}subGame^{coll}_{(E^f,E^b,E^k),ext.WCM}$
    searching for (x_l, y_l) and $(x_{l'}, y_{l'})$ from $\mathcal{Q}$
    if $\{(x_l, y_l) = (x_{l'}, y_{l'})\} \rightarrow \mathcal{A}$ wins **then**
        call ${}^{outer}tri^{coll}_{(E^f,E^b,E^k),ext.WCM}$ and terminate ${}^{outer}subGame^{coll}_{(E^f,E^b,E^k),ext.WCM}$
    end if
    CALL $\rightarrow {}^{inner}subGame^{coll}_{(E^f,E^b,E^k),ext.WCM}$
    searching for (x_l, y_l)
    if $(x_l = y_l) \rightarrow \mathcal{A}$ wins **then**
        call ${}^{inner}tri^{coll}_{(E^f,E^b,E^k),ext.WCM}$ and terminate ${}^{inner}subGame^{coll}_{(E^f,E^b,E^k),ext.WCM}$
    end if
    CALL $\rightarrow {}^{iv}Game^{coll}_{(E^f,E^b,E^k),ext.WCM}$
    searching for (x_l, y_l)
    if $\{(x_l, y_l) = (x_0, y_0)\} \rightarrow \mathcal{A}$ wins **then**
        call ${}^{iv}tri^{coll}_{(E^f,E^b,E^k),ext.WCM}$ and terminate ${}^{iv}subGame^{coll}_{(E^f,E^b,E^k),ext.WCM}$
    end if
end procedure

${}^{iv}subGame^{coll}_{(E^f/E^b/E^k),ext.WCM}$. Under this subgame, there is a possibility for a collision such as $(x_l, y_l) = (x_0, y_0)$. Therefore, the probability of collision is:

$$\sum_{l=1}^{q} \Pr\left[{}^{iv}Tri^{coll}_{l,(E^f,E^b,E^k),ext.WCM}\right] = \sum_{l=1}^{q} \frac{2}{2^n} \leq \frac{2q}{2^n} \tag{4.15}$$

Therefore, **Theorem 4.3** is proved by taking the union bound of 4.12, 4.14 and 4.15.

4.2 A Pair of Constructions of Compression Function

In this section, we propose two schemes of (n, n) block-cipher based compression function for short message encryption. The second scheme is defined as SS and later one is noted as TS. The proposed second scheme has higher efficiency rate, less call of block-ciphers and less key scheduling (Table 4.6 and Table 4.7). It operates in parallel. On the contrary, the third scheme is bounded for upper security margin (Table 4.7). Both of the schemes are padding free construction for short and variable size of message. Moreover, the proposed two schemes follow the Davies Meyer mode (DM) and satisfy the feature of double key scheduling (KS).

Table 4.6: Comparison: FS, SS and other (n, n) based blockciphers scheme [27, 38, 56, 59, 69]

	CF	r	$\#E$	**KS**	**CR**	**PR**	**OM**	**PF**	**RM**
MDC-2	$3n \to 2n$	1/2	2	2	$\mathcal{O}\left(2^{\frac{n}{2}}\right)$	$\mathcal{O}(2^n)$	P	×	176×2
MDC-4	$3n \to 2n$	1/4	4	4	$\mathcal{O}(2^{\frac{5n}{8}})$	$\mathcal{O}\left(2^{\frac{5n}{4}}\right)$	P	×	176×4
MJH	$3n^{+c} \to 2n$	1/2	2	1	$\mathcal{O}\left(2^{\frac{n}{2}}\right)$	$\mathcal{O}(2^n)$	P	×	176×2
Bart	$3n \to 2n$	1/3	3	3	$\mathcal{O}(2^n)$	$\mathcal{O}\left(2^{\frac{3n}{2}}\right)$	S	×	176×3
SKS	$3n \to 2n$	1/3	3	1	$\mathcal{O}(2^n)$	$\mathcal{O}(2^{2n})$	P	×	176×3

CF: Compression function
KS: Key Scheduling
CR: Collision resistance
PR: Preimage resistance
$\#E$: Number of blockcipher calls
r: Efficiency rate
OM: Operational mode, P: Parallel, S: Serial
PF: Padding free
RM: Required memory in bytes

Table 4.7: Characteristics of Second Scheme and Third Scheme

	CF	**KS**	**CR**	**PR**	$\#E$	r	**OM**	**PF**	**RM**
SS	$2n + tn \to 2n$	2	$\mathcal{O}(2^{tn/2})$	$\mathcal{O}(2^{tn})$	2	t	Parallel	$\surd$	176×2
TS	$2n + tn \to 2n$	2	$\mathcal{O}(2^{tn})$	$\mathcal{O}(2^{2tn})$	3	$t/3$	Serial	$\surd$	176×3

CF, KS, CR, PR, $\#E$, r, OM, PF, RM (refer to Tabel 4.6)

4.2.1 Proposed Second Scheme of Compression Function (SS)

In this section, we define the construction of SS through diagram and definitions of Figure 4.3 and Definition 4.2, 4.3. We use two set of short message under a single compression function of SS. The key scheduling is double but construction is based on (n, n) block-cipher. It consists of two calls of block-cipher per compression function. Furthermore, the SS follows the Davies Meyer (DM) mode. There are certain notations which are related

to proposed second scheme such as F^{SS}: Compression function, $N = 2^n$: Domain size, $\mathcal{A}$: Adversary, Q: Query Triplet, $\oplus$: XoR operation, coll: collision, $cv_{i,j}, cv_{i,j}$: Chaining value, $\mathcal{B}$: Adversary, $\mathcal{Q}$: Query database, $\mathcal{ADV}$: Advantage of Adversary, $\mathcal{C}_i$: coll. event, and $\Pr[\cdot]$: Probability.

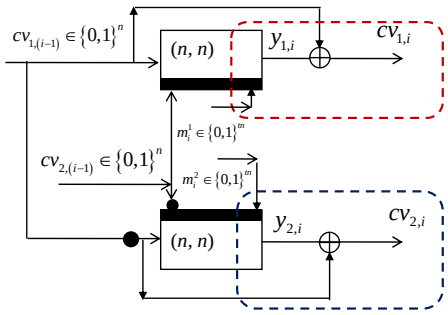


Figure 4.3: Second scheme (SS)

Definition 4.2. *Let $E \in Block_n^k$ be a block-cipher taking k-bit key and an n-bit block size. The compression function of $F^{SS} : \{0,1\}^n \times \{0,1\}^{n-tn} \times \{0,1\}^{2tn} \to \{0,1\}^{2n}$ $(t < 1)$ is defined as:*

$$F^{SS}\left(cv_1, cv_2, m^1, m^2\right) = \left(E_{cv_2 \| m^1}\left(cv_1\right) \oplus cv_1, E_{\overline{cv}_2 \| m^2}\left(\overline{cv}_1\right) \oplus \overline{cv}_1\right)$$

Definition 4.3. *Let $F^{SS} : \{0,1\}^n \times \{0,1\}^{n-tn} \times \{0,1\}^{2tn} \to \{0,1\}^{2n}$ be a compression function where $(cv_{1,i}, cv_{2,i}, m_i^1, m_i^2) = F^{SS}(cv_{1,i}, cv_{2,i}, m_i^1, m_i^2)$. The notations are $cv_{1,i} \in \{0,1\}^n, cv_{2,i} \in \{0,1\}^{n-tn}, (m^1, m^2) \in \{0,1\}^{tn}$. Therefore, F^{SS} consists of $((n+m) = k, n)$ ideal block cipher E as like,*

$$F_u^{SS}(cv_{1,i-1}, cv_{2,i-1}, m_i^1) = E(cv_{1,i-1}, cv_{2,i-1} \| m_i^1) \oplus cv_{1,i-1}$$
$$F_l^{SS}(\overline{cv}_{1,i-1}, \overline{cv}_{2,i-1}, m_i^2) = E(\overline{cv}_{1,i-1}, \overline{cv}_{2,i-1} \| m_i^2) \oplus \overline{cv}_{1,i-1}$$

Security Analysis of the Second Scheme (SS)

A computationally unbounded adversary $\mathcal{A}$ is given oracle access to a block-cipher E/E^{-1}. In block-cipher oracle, all block-cipher's key (n-bit) and block-data (n-bit) are uniformly distributed. The query history of $\mathcal{A}$ is the set of triples, where Q consists x_i : plaintext, y_i : ciphertext, and k_i : key. The queries are stored in query database $\mathcal{Q}$. Therefore, adversary $\mathcal{A}$ gets success for finding a collision under any i-th iteration. For example, there exist two distinct set of queries such as $w \leftarrow (cv_{1,i-1}, cv_{2,i-1}, m_i^1)$, $x \leftarrow (\overline{cv}_{1,i-1}, \overline{cv}_{2,i-1}, m_i^2)$, $y \leftarrow \left(cv_{1,j-1}, cv_{2,j-1}, m_j^1\right)$ and $z \leftarrow \left(\overline{cv}_{1,j-1}, \overline{cv}_{2,j-1}, m_j^2\right)$. Hence, $\mathcal{A}$ gets success iff,

$$F^{SS}(w) = F^{SS}(y), F^{SS}(w) = F^{SS}(z) \text{ or } F^{SS}(x) = F^{SS}(y), F^{SS}(x) = F^{SS}(z)$$
$$\text{when}, (w \neq x \neq y \neq z)$$

Collision security of SS. An adversary $\mathcal{A}$ is allowed to make any relevant query to E/E^{-1} under the ideal cipher model (ICM). However, $\mathcal{A}$ is not allowed for duplicate query. For example, adversary never makes a query of $E(k, x) = y$ if y is already part of

$E^{-1}(k, y) = x$ query. Adversary $\mathcal{A}$ is limited to make an arbitrary query upto q. Therefore, $\mathcal{A}$ tries to find a collision under F^{SS} (compresion function of the second scheme) through the block-cipher oracle. The output $(cv_{1,i}, cv_{2,i})$ of compression function (F^{SS}) depends both on the plain-text and cipher-text including key for any i-th iteration. Thus, one of these is fixed by an adversarial query. On the other hand, the rest of the values are determined randomly from the block-cipher oracle. Usually, the adversary makes a query through block-cipher oracle, where size of plain-text or cipher-text is n bit. Yet, we use short message (tn such that $t < n$) in the SS scheme. Hence, it is necessary to accommodate the feature of short message in query response mechanism. Firstly, we try to address the traditional query response mechanism. For example, the collision length is n-bit when adversary finds a collision. If collision occurs under any n-bit, hence it is not problem for adversary to find a collision under tn-bit. However, $\mathcal{A}$ doesn't allow to get success for finding collision through tn-bit message directly. Under this circumstance, we invoke adversary $\mathcal{B}$ which works based on the query response of $\mathcal{A}$.

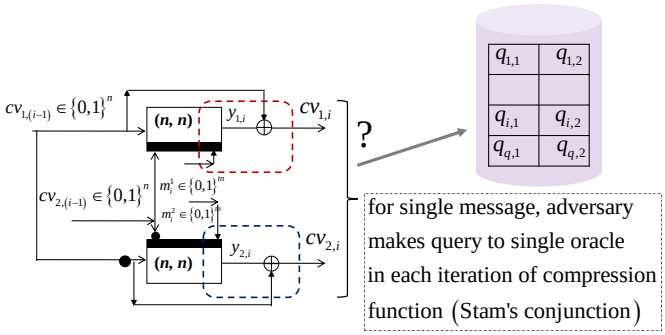


Figure 4.4: SS Security Analysis (single message set)

The adversary $\mathcal{A}$ calls adversary $\mathcal{B}$. Thereafter, $\mathcal{B}$ is allowed to access the query response database of $\mathcal{A}$. Furthermore, the adversary $\mathcal{B}$ is allowed to search the query, where size is tn-bits instead of n-bits. Thus, the query database is being reduced 2^n to 2^{tn}. Hence, $\mathcal{B}$ is more powerful than $\mathcal{A}$ due to less size of database. For example, adversary $\mathcal{A}$ makes a query and get the output $(cv_{1,i}, cv_{2,i}) \in \{0,1\}^n$. Therefore, the adversary $\mathcal{B}$ takes that output and tries to prune tn-bit from n-bit. If it finds a collision under tn-bit block size then the adversary $\mathcal{B}$ wins. Moreover, $\mathcal{B}$ sends true to adversary $\mathcal{A}$ and adversary $\mathcal{A}$ stops the query process. On the contrary, adversary $\mathcal{B}$ sends false to adversary $\mathcal{A}$. In this way adversary $\mathcal{B}$ tries to find a collision for the size of tn bit instead of n bit. Hence, it is clear that the adversary $\mathcal{B}$ gets more advantage and the result is more tight.

Theorem 4.4. Let F^{SS} be a block length compression function specified in Definition 4.2, 4.3. Therefore, an adversary $\mathcal{A}$ is defined for finding a collision under F^{SS}. Hence, the advantage of adversary is upper bounded after q queries such as:

$$\mathcal{ADV}^{coll}_{F^{SS}}(q) \leq 2\left(\frac{q(q-1)}{2^{2tn}} + \frac{q}{2^{tn}} + \frac{2q}{2^{tn}}\right)$$

Proof. There are three cases for defining the security issues of the SS scheme. We define and explain these cases in the following ways.

Case 1$(c1)$. For any iteration of i $(j < i < q)$, we assume C be the event of collision-hit under the F^{SS}. The conditions of collision-hit for different two types of query are as

follows:

$$y_{1,i} \oplus cv_{1,(i)} = y_{1,j} \oplus cv_{1,(j)} \wedge y_{1,i} \oplus cv_{1,(i)} = y_{2,j} \oplus \overline{cv}_{1,(j)}$$

$$y_{2,i} \oplus \overline{cv}_{1,(i)} = y_{1,j} \oplus cv_{1,(j)} \wedge y_{2,i} \oplus \overline{cv}_{1,(i)} = y_{2,j} \oplus \overline{cv}_{1,(j)}$$

Let $Coll_{c1}$ be the event that find a collision for different two sets of queries under the F^{SS}. Thus, the probability of collision events are $\Pr[Coll_{c1}] = \Pr[C_2 \vee C_3 \vee ..... \vee C_q]$. Hence,

$$\Pr[Coll_{c1_i}] \leq \sum_{i=2}^{q} \frac{2(i-1)}{\left(2^{tn} - (2i-1)^2\right)} \leq \sum_{i=2}^{q} \frac{2(i-1)}{2^{2tn}} \Rightarrow \frac{q(q-1)}{2^{2tn}} \tag{4.16}$$

Case 2$(c2)$. This case is defined for single query for the iteration of i $(i \leq 1)$, where the collision condition is $cv_{1,(i)} = cv_{2,(i)}$. Let $Coll_{c2}$ be the event that find a collision for single set of query under F^{SS}. Therefore, the probability of collision events are $\Pr[Coll_{c2}] = \Pr[C_1 \vee C_2 \vee ..... \vee C_q]$. Hence,

$$\Pr[Coll_{c2_i}] \leq \sum_{i=1}^{q} \frac{1}{(2^{tn} - i)} \leq \frac{q}{2^{tn}} \tag{4.17}$$

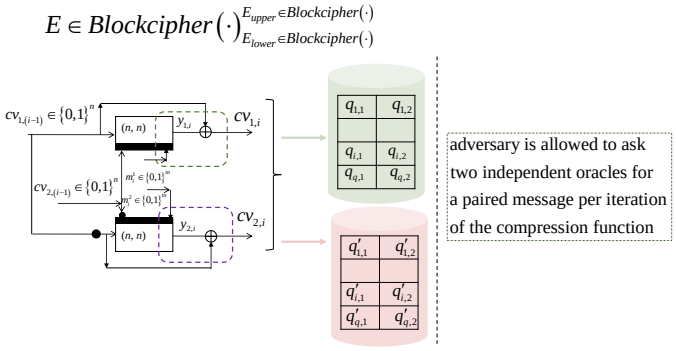


Figure 4.5: Collision security analysis of SS

Case 3$(c3)$. For any iteration of i, there is chance to occur a collision through initial chaining values. The conditions are as follows:

$$\{(cv_{1,i}, cv_{2,i}, k_i) \rightarrow y_{1,i}\}, \{(\overline{cv}_{1,i}, \overline{cv}_{2,i}, k'_i) \rightarrow y_{2,i}\} = (cv_{0,1}, cv_{0,2}) \vee (cv_{0,2}, cv_{0,1})$$

We assume that $Coll_{c3}$ be the event for finding a collision through initial vector values under F^{SS}. Therefore, the probability of collision events are $\Pr[Coll_{c3}] = \Pr[C_1 \vee C_2 \vee ..... \vee C_q]$. It implies that,

$$\Pr[Coll_{c3_i}] \leq \sum_{i=1}^{q} \frac{2}{2^{tn}} \leq \frac{2q}{2^{tn}} \tag{4.18}$$

Taking the union bound of 4.16, 4.17, and 4.18 we get,

$$\frac{q(q-1)}{2^{2tn}} + \frac{q}{2^{tn}} + \frac{2q}{2^{tn}} \tag{4.19}$$

The above result (Figure 4.4) comes from two block-ciphers response according to single set of message. However, we use two set of different messages under two block-ciphers. According to Stam's principle, usually oracle response follows through single

message [26]. Due to use of two set of messages, adversary can ask two different oracles according to each message. Therefore, adversary calls two oracles for query under each iteration of compression function. Thus, the collision-hit probability comes from the upper and lower block based two independent oracles (Figure 4.5). The value of 4.19 is valid for single oracle. Hence, the upper block and lower block probability are as follows:

$$\Pr^{u}_{coll_i} = \frac{q\,(q-1)}{2^{2tn}} + \frac{q}{2^{tn}} + \frac{2q}{2^{tn}} \tag{4.20}$$

$$\Pr^{l}_{coll_i} = \frac{q\,(q-1)}{2^{2tn}} + \frac{q}{2^{tn}} + \frac{2q}{2^{tn}} \tag{4.21}$$

Finally, adding the values of 4.20, and 4.21 **Theorem 4.4** is proved.

Preimage Security of SS. Let $\mathcal{A}$ be an adversary that tries to find a preimage for predetermine output (δ). The preimage security proof of SS is followed by Armknecht [58]. Adversary $\mathcal{A}$ asks a pair of queries through E/E^{-1}. Thus, it is needed to bound the probability for preimgae-hit under i-th query pair. The adversary wins iff the output of compression function collides with the δ. For example, the compression function output are $cv_{1,i}$, $cv_{2,i}$. Therefore, the condition of preimage-hit is:

$$(cv_{1,i}, cv_{2,i}) = \delta \in \{rsv_1, rsv_2\}$$

where, rsv_1, rsv_2 : randomly selected value by the adversary $\mathcal{A}$.

Theorem 4.5. Let F^{SS} be a block length compression function. $\mathcal{A}$ is an adversary for finding preimage-hit under the F^{SS}. The advantage of $\mathcal{A}$ is bounded after q queries such as:

$$\mathcal{ADV}^{pre}_{F^{SS}}(q) \leq 2\left(16/2^{2tn}\right)$$

Proof. The adversary $\mathcal{A}$ maintains a query database Q in the form of $cv_{1,i}, cv_{2,i}$. If the size of database reaches to $N/2$ (N : size of database (2^n)) then all remaining queries under this key are given for free to the adversary. The first half of $N/2$ is called normal query database and later one is defined as super query database [25, 58]. Thereafter, the successful conditions (Figure 4.6) of preimage-hit for the adversary are:

$$\{(cv_{1,i-1} \oplus y_{1,i} = cv_{1,i}) = rsv_1\} \wedge \{(cv_{1,i-1} \oplus y_{2,i} = cv_{2,i}) = rsv_2\}$$

$$\{(cv_{1,i-1} \oplus y_{1,i} = cv_{2,i}) = rsv_2\} \wedge \{(cv_{1,i-1} \oplus y_{2,i} = cv_{1,i}) = rsv_1\}$$

The above conditions can be occurred under the following any event:

- Normal query win under the Normal query database
- Super query win under the super query database

Therefore, we need to find out the preimage-hit probability under the above events. Moreover, we need to take the union bound of the above two results such as:

$$\Pr[\text{Normal query win}(Q)] + \Pr[\text{Super query win}(Q)]$$

Case 1(c_1). Adversary $\mathcal{A}$ makes forward or backward query such as:

$$E_{cv_{1,i-1}\|m^1}(cv_{2,i-1}), E^{-1}{}_{cv_{1.i-1}\|m^2}(\bar{cv}_{2,i-1})$$

where, the goal is to find the Normal query win(Q). According to definition of normal query and adjacent query [58], the set size of fresh value is needed to evaluate. Under this circumstances, two sub cases can be happened such as:

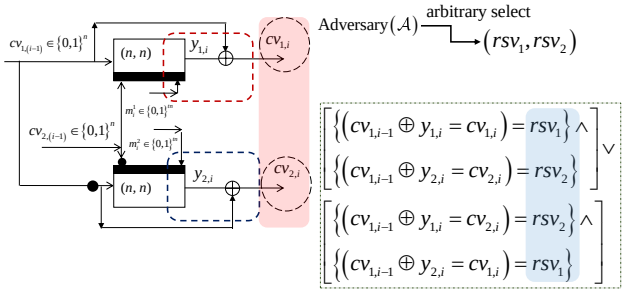


Figure 4.6: Preimage Security Analysis of SS

- Sub-Case 1.1. For example, $\mathcal{A}$ makes a forward $E_{cv_{2,i-1}\|\bar{m}^1}(cv_{1,i-1})$ query, where at most $(2^{tn}/2-1)$ queries could be answered previously. Therefore, the query of $E_{\overline{cv}_{2,i}\|m^2}(\overline{cv}_{1,i})$ and corresponding answer comes from the set size $(2^{tn}/2-1)$. If adversary fails to find any preimage-hit under this scenario, it implies that super query will be occurred. Hence, the value of $y_{1,i}$ and $y_{2,i}$ come uniformly and independently from the set size $2^{tn}/2$. Thus, the probability is $2/2^{tn}/2$.

- Sub-Case 1.2. If $cv_{1,i-1} \oplus y_{1,i} = cv_{1,i}$ satisfies, then the probability for the free query $E_{cv_{2,i-1}\|m^2}(\overline{cv}_{1,i-1})$ comes from the set size $(2^{tn}/2+1)$. Hence, the probability is $1/2^{tn}/2 = \frac{2}{2^{tn}}$.

Therefore,

$$\Pr[\text{Normal query win}(Q)] = 8/2^{2tn} \tag{4.22}$$

Case 2(c_2). In this case, we try to find the probability of preimage-hit under the super query database. For example, the value of $E_{cv_{2,i-1}\|m^1}(\cdot)$ and $E_{\overline{cv}_{2,i-1}\|m^2}(\cdot)$ already have been known on exactly $2^{tn}/2$ points. Therefore, $E_{cv_{2,i-1}\|m^1}(\cdot)$ is the part of super and the corresponding $E_{\bar{cv}_{2,i-1}\|m^2}(\cdot)$ query must be the member of the super query domain. From the above discussions, the probability of $E_{cv_{2,i-1}\|m^1}(cv_{1,i}) = cv_{1,i}$ is either 0 or $\frac{2}{2^{tn}}$. The probability is 0, if the $cv_{1,i}$ is not in the part of super query. It means $cv_{1,i}$ is the part of normal query. On the contrary, the result comes from the set size $2^{tn}/2$ due to super query. Therefore, the probability is $\frac{2}{2^{tn}}$. For simplicity, the conditions of preimage-hit under case-2 are:

$$\{(y_{1,i} \oplus cv_{1,i-1} \in cv_{1,i}) = rsv_1 \wedge (y_{2,i} \oplus \overline{cv}_{1,i-1} \in cv_{1,i}) = rsv_2\} \vee$$

$$\{(y_{1,i} \oplus cv_{1,i-1} \in cv_{2,i}) = rsv_2 \wedge (y_{2,i} \oplus \overline{cv}_{1,i-1} \in cv_{2,i}) = rsv_1\}$$

- Sub-Case 2.1. For the query of $E_{cv_{2,i-1}\|m^1}(cv_{1,i-1}) \oplus cv_{1,i-1} = cv_{1,i}$, the answer comes from the set size $2^{tn}/2$. Hence, the probability is $\frac{2}{2^{tn}}$. Moreover, the probability of $E_{\overline{cv}_{2,i-1}\|m^2}(\overline{cv}_{1,i-1}) \oplus \overline{cv}_{1,i-1} = cv_{1,i}$ is $\frac{2}{2^{tn}}$. Hence, the total probability of this subcase is $\left(\frac{2}{2^{tn}}\right)^2$.

- Sub-Case 2.2. According to subcase 2.2, the total probability of $E_{cv_{2,i-1}\|m^1}(cv_{1,i-1}) \oplus cv_{1,i-1} = cv_{2,i}$ and $E_{\overline{cv}_{2,i-1}\|m^2}(\overline{cv}_{1,i-1}) \oplus \overline{cv}_{1,i-1} = cv_{2,i}$ is $\left(\frac{2}{2^{tn}}\right)^2$.

Now, we analyse the probability of case-1 and case-2 with the cost of super query occurrence. The colliding cost of super query is $2^{tn}/2$. Hence, the probability of collision

of super queries is at most $q/(2^{tn}/2)$. Thus,

$$\Pr\left[\text{Super query win}\,(Q)\right] \leq q/(2^{tn}/2) \times (2^{tn}/2) \times 2 \times \left(\frac{2}{2^{tn}}\right) = \frac{8q}{2^{2tn}} \tag{4.23}$$

Taking the union bound of 4.22 and 4.23 the adversarial advantage is $\frac{16q}{2^{2tn}}$. This is true for single oracle. However, we use two set of message in our scheme. Hence, the adversary gets chance to make query from two oracles for preimgae-hit. Therefore, the probability of preimage-hit under F^{SS} is $2 \times \left(\frac{16q}{2^{2tn}}\right)$ (**Theorem 4.5** is satisfied).

4.2.2 Proposed Third Scheme of Compression Function (TS)

In this section, we define third scheme (TS). It is based on three calls of (n, n) block-cipher. By construction, the TS (Figure 4.7) invokes a single set of message per iteration through three block-cipher. The message size is tn-bits, where $t < 1$.

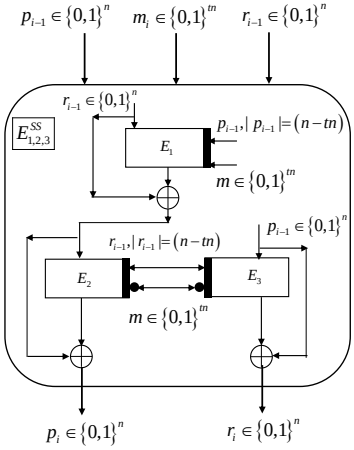


Figure 4.7: Block Diagram of Proposed Third scheme (TS)

Definition 4.4. *Let $E \in Block_n^k$ be a block cipher, where key size and block size are respectively k-bit and n-bit. The compression function $F^{TS} : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^{tn} \rightarrow \{0,1\}^{2n}$ $(t < 1)$ consists of three block-ciphers which are defined as follows*:

$$\begin{aligned} &E^{TS}_{1,2,3} = \\ &E_{1,m\|p_{i-1}}\left(r_{i-1}\right) \oplus r_{i-1} \rightarrow x_i, \\ &E_{2,\bar{m}\|r_{i-1}}\left(x_i\right) \oplus x_i \rightarrow p_i, \\ &E_{3,\bar{m}\|r_{i-1}}\left(p_{i-1}\right) \oplus p_{i-1} \rightarrow r_i \end{aligned}$$

where, $(|p_{i-1}| = n - tn\,(\text{for } E_1)\,; |r_{i-1}| = n - tn\,(\text{for } E_2,\, E_3))$. Therefore,

$$\begin{aligned} &F^{TS}\left(p_{i-1}, r_{i-1}, m\right) \\ &= \begin{pmatrix} E_{2,\bar{m}\|r_{i-1}}\left(E_{1,m\|p_{i-1}}\left(r_{i-1}\right)\right) \oplus \left(E_{1,m\|p_{i-1}}\left(r_{i-1}\right)\right), \\ E_{3,\bar{m}\|r_{i-1}}\left(r_{i-1}\right) \oplus \left(r_{i-1}\right) \end{pmatrix} \end{aligned}$$

Security Analysis of the Third Scheme (TS)

Collision security of TS. Adversary $\mathcal{A}$ is a collision finding experiment based on ideal cipher model oracle. It can make any query E or E^{-1}. On the iteration of i, the scenario looks $Q_i \in$ (note that $m_{i-1}, k_{i-1}, c_{i-1}$) or $Q_i \in (c_{i-1}, k_{i-1}, m_{i-1})$ where m, k, c respectively stands for plain-text, key, cipher-text. The responses are stored in $\mathcal{Q}$ such that $\mathcal{Q} \in (Q_1, Q_2, ...Q_i)$, where $i < q$. Adversary $\mathcal{A}$ gets success iff, $F^{TS}(m,k,c) = F^{TS}(m',k',c')$. According to definition of TS, the above condition rewrite as $F^{TS}(p,r,m) = F^{TS}(p',r',m')$.

Theorem 4.6. Let F^{TS} be a compression function consists of triple block-cipher. An adversary $\mathcal{A}$ is defined to find collisions under the F^{TS} after q queries. Hence, the advantage of $\mathcal{A}$ is bounded by:

$$\mathcal{ADV}^{coll}_{F^{TS}}(q) \leq \frac{3q^2 - 5q}{\left(2^{n-\sqrt{tn}} - 3q\right)^2} + \frac{q}{(2^{n-tn} - 3q)}$$

Proof. Adversary $\mathcal{A}$ is allowed to make any relevant query to E/E^{-1} under the ideal cipher model. In addition, adversary $\mathcal{A}$ is computationally unbounded in respect of memory and time. Furthermore, it is bounded in respect of time and memory. We use three calls of block-cipher under the $E^{SS}_{1,2,3}$. The output of three block-ciphers fed into final output of $E^{SS}_{1,2,3}$. Therefore, we will find out the probability of collision under three calls of block-cipher. For example, the output (p,r) of F^{TS} comes from $E^{TS}_{1,2,3}$. At first we define the conditions of collision occurrence (Figure 4.8). There are two main conditions for causing collision such as non-matching pair and matching query. The non-matching conditions are defined as $(a < b < i < q)$:

$$\{(p_{a+1} = r_{b+1}) \wedge (p_{a+1} = r_{i+1})\} \vee \{(p_{b+1} = r_{i+1}) \wedge (p_{b+1} = r_{a+1})\} \vee \{(p_{i+1} = r_{b+1}) \wedge (p_{i+1} = r_{a+1})\}$$

When,

$$[f(p_a, r_a, m), f(p_b, r_b, m), f(p_i, r_i, m) \,|\text{s.t.}\, (p_a, r_a, m) \neq (p_b, r_b, m) \neq (p_i, r_i, m)]$$

That means three different set of queries are needed to occur collision. Moreover, matching query means a couple of output makes collision for any single set of query. Hence,

$$p_{i+1} = r_{i+1} \,\big|\big(F^{TS}(p_i, r_i, m_i) = (p_{i+1}, r_{i+1})\big)$$

Additionally, a collision can be occurred through initial chaining values. Usually a set of chaining values inject in compression function for initialization. Thus, an opportunity is arisen for making collision through p_0, r_0 under any iteration of $i|\,(i \geq 1)$.

non-matching query. Let the collision event is C_i. According to the definition of non-matching collision, three calls of block-cipher are used under $F^{TS}\left(E^{TS}_{1,2,3}\right)$. Hence, the event (C_i) is created for any i $(a < b < i < q)$. Thereafter, the probability is:

$$\Pr[C_i] = \frac{2^{tn} \times 3(i-3)}{(2^n - 3i)(2^n - 3i)}$$

Let C be the event that find a collision under non-matching query through F^{TS} for q queries. Thus, the probability of collision events are $\Pr[C] = \Pr[C_3 \vee C_4 \vee ..... \vee C_q]$. Thereafter,

$$\sum_{i=3}^{q} \Pr[C_i] = \sum_{i=3}^{q} \frac{2^{tn} \times 3(i-3)}{(2^n - 3i)(2^n - 3i)} \leq \frac{3q(q-2)}{\left(2^{n-\sqrt{tn}} - 3q\right)^2} \tag{4.24}$$

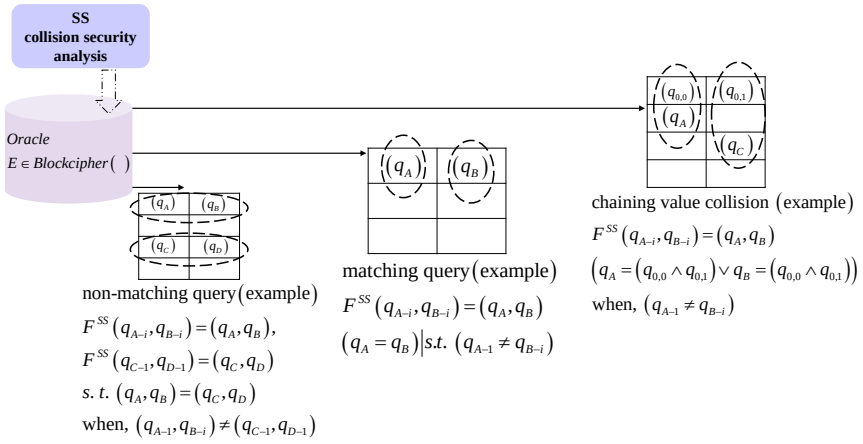


Figure 4.8: Collision Security Analysis of TS

matching query. For matching query, we define collision an event C_i. The probability of C_i will be $2^{tn}/(2^n - 3i)$ according to the definition of matching query condition. Therefore, C be the events of collision occur for q queries such as:

$$\Pr[C] = \Pr[C_1 \vee C_2 \vee ..... \vee C_q]$$

Hence, the probability is:

$$\sum_{i=1}^{q} \Pr[C_i] = \sum_{i=1}^{q} \frac{2^{tn}}{(2^n - 3i)} \leq \frac{q}{(2^{n-tn} - 3q)} \tag{4.25}$$

collision with initial chaining value. Under this condition, the initial chaining values are p_0, r_0. Therefore, a collision can be occurred under any i-th iteration iff:

$$(p_i = (p_0 \wedge r_0) \vee r_i = (p_0 \wedge r_0))$$

We define a collision event C_i. Therefore, the probability of C_i is $2 \times 2^{tn} \times 1/(2^n - 3i) \times 1/(2^n - 3i)$. Let, C be the events for colliding pair for q queries such as $\Pr[C] = \Pr[C_1 \vee C_2 \vee ..... \vee C_q]$. Hence, the probability of collision events is:

$$\sum_{i=1}^{q} \Pr[C_i] = \sum_{i=1}^{q} \frac{2^{tn} \times 2}{(2^n - 3i)(2^n - 3i)} \leq \frac{2^{tn+1} \times q}{(2^n - 3q)^2} \tag{4.26}$$

Adding the values of 4.24, 4.25, and 4.26 Theorem 4.6 is satisfied.

Preimage Security of TS. Let adversary $\mathcal{A}$ tries to find a preimage-hit for pre-define output of p', r'. Initially, $\mathcal{A}$ selects arbitrarily p', r'. The adversary can ask query through E and E^{-1}. In any point of i-th iteration, there is a chance to occur preimage-hit for $F^{TS}(p, r, m) = (p', r')$ (Figure 4.9).

Theorem 4.7. Let F^{TS} be a block length compression function $\left(E \in Block_n^k\right)$, where adversary $\mathcal{A}$ is defined for finding a preimage-hit. The advantage of adversary is bounded after q queries such as:

$$\mathcal{ADV}^{\text{pre}}_{F^{TS}}(q) \leq \frac{q(q-2)}{(2^n - 3q)^2 \times 2^{-tn}} + \frac{q-2}{(2^{n-tn} - 3q)}$$

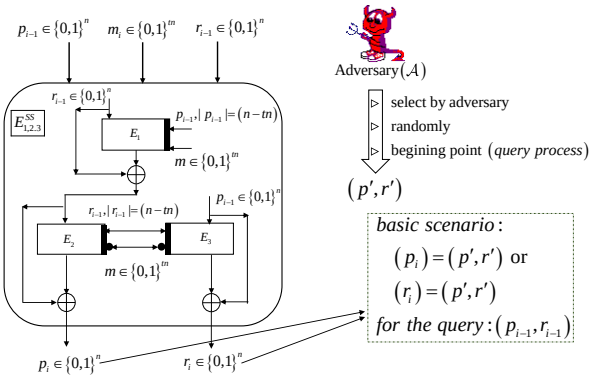


Figure 4.9: Preimage Security Analysis of TS

Proof. An adversary $\mathcal{A}$ selects randomly p', r' for preimage attack. Therefore, the adversary needs to execute three calls of block-cipher according to the definition of our proposed scheme. If $F_{TS}(p, r, m) = (p', r')$, we can say that adversary get success for preimage-hit. Our define conditions are:

$$\begin{array}{l}(p_{a+1} = p'), (p_{a+1} = r') \wedge (r_{a+1} = p'), (r_{a+1} = r') \vee \\ (p_{b+1} = p'), (p_{b+1} = r') \wedge (r_{b+1} = p'), (r_{b+1} = r') \vee \\ (p_{i+1} = p'), (p_{i+1} = r') \wedge (r_{i+1} = p'), (r_{i+1} = r')\end{array}$$

When,

$$[f(p_a, r_a, m), f(p_b, r_b, m), f(p_i, r_i, m) | s.t. (p_a, r_a, m) \neq (p_b, r_b, m) \neq (p_i, r_i, m)]$$

We assume that the preimage hit event is Pre_i. According to the above conditions of F^{TS}, Pre_i can be occurred at any point of i-th iteration $(a < b < i < q)$. Therefore, the events of preimage-hit for q queries are:

$$\Pr[\text{Pre}] = \Pr[\text{Pre}_3 \vee \text{Pre}_4 \vee ..... \vee \text{Pre}_q]$$

Hence, the probability is:

$$\sum_{i=3}^{q} \Pr[\text{Pre}_i] = \sum_{i=3}^{q} \frac{2^{tn} \times 3(i-3)}{(2^n - 3i)(2^n - 3i)} \leq \frac{q(q-2)}{2^{-tn} \times (2^n - 3q)^2} \tag{4.27}$$

After execution of three calls of block-cipher, there is chance to occur $p' = r'$. Let, Pre be the event for occurring this scenario. Therefore, the probability of preimage-hit events are:

$$\Pr[\text{Pre}] = \Pr[\text{Pre}_3 \vee \text{Pre}_4 \vee ..... \vee \text{Pre}_q]$$

Hence, the probability is:

$$\sum_{i=3}^{q} \Pr[\text{Pre}_i] = \sum_{i=3}^{q} \frac{2^{tn}}{(2^n - 3i)} \leq \frac{q-2}{(2^{n-tn} - 3q)} \tag{4.28}$$

Adding the values of 4.27and 4.28 Theorem 4.7 is satisfied.

Table 4.8: Comparison of efficient rate for different schemes [27, 38, 56, 59, 69]

SS	message size ($2tn$)	efficiency-rate
	$t = 1/2$	0.5
	$t = 2/3$	0.66
	$t = 3/4$	0.75
TS	message size (tn)	efficiency-rate
	$t = 1/2$	0.166
	$t = 2/3$	0.22
	$t = 3/4$	0.25
	message size ($m = n$)	efficiency-rate
MDC-2	m	$m/2n = 0.5$
MDC-4	m	$m/4n = 0.25$
MJH	m	$m/2n = 0.5$
Bart-12	m	$m/3n = 0.33$
SKS-15	m	$m/3n = 0.33$

4.2.3 Efficiency Analysis Second and Third Scheme

In this section, we explain the efficiency analysis of different schemes. The efficiency-rate is defined as r = (message size)/(blocklength) × (number of blockcipher call). The schemes of MDC-2, MDC-4, MJH, Bart-12 are not fit for variable and short message. Therefore, the message size is n-bit per compression function for the above mentioned schemes. On the contrary, the proposed schemes of SS and TS are suitable for short and variable message. According to the definition of SS and TS, the message size are respectively $2tn$ and tn ($t < 1$). That's why the efficiency of SS and TS also varies (based on message size). In Table 4.8, we mention the efficiency rate of various schemes.

4.3 A Light Scheme of (n, n) block-cipher compression Function

In this section, we proposed fourth scheme (FrS) of (n, n) block-cipher hash that satisfies a single key scheduling with better security bound (Table 4.9). We use three calls of block-cipher through the Davies Meyer (DM) mode. The result of collision resistance and preimage resistance are $O(2^n)$ and $O(2^{2n})$ under the ICM. Additionally, the proposed scheme is bounded by CR = $O(2^n)$ and PR = $O(2^n)$ under the WCM. The efficiency rate of proposed scheme is 1/3.

Table 4.9: Comparison study of existing schemes and new scheme (Aspect: Security Proof) [27, 38, 56, 59, 69]

Scheme Name	**ICM**		**WCM**		**FFM**	
	CR	**PR**	**CR**	**PR**	**CR**	**PR**
FrS	$\mathcal{O}(2^n)$	$\mathcal{O}(2^n)$	$\mathcal{O}(2^n)$	$\mathcal{O}(2^n)$	-	-
MDC-2	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$	-	-	-	-
MDC-4	$\mathcal{O}(2^{5n/8})$	$\mathcal{O}(2^{5n/4})$	-	-	-	-
MJH	-	-	-	-	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$
Bart	-	-	-	-	$\mathcal{O}(2^n)$	$\mathcal{O}(2^{3n/2})$
MSR	$\mathcal{O}(2^{tn})$	$\mathcal{O}(2^{2tn})$	-	-	-	-
CIDM	$\mathcal{O}(2^n)$	$\mathcal{O}(2^{2n})$	-	-	-	-

4.3.1 Proposed Fourth Scheme of Compression Function

In this section, we proposed fourth scheme of (n, n) block-cipher cryptographic compression function as FrS (Figure 4.10). It satisfies a single key scheduling (KS = 1). It is based on three calls of (n, n) block-cipher under the Davies Meyer mode (DM) (message goes as key input). The efficiency rate of our scheme is 1/3.

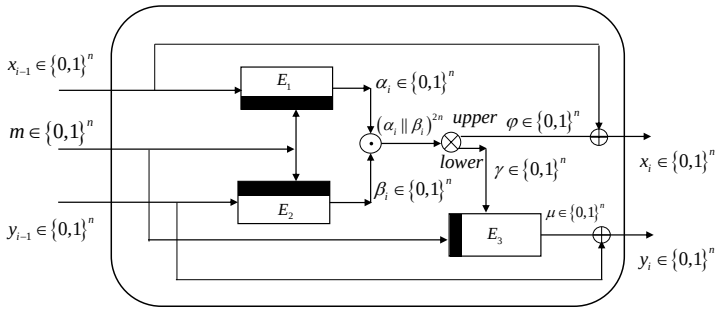


Figure 4.10: Proposed Fourth Scheme

Definition 4.5. Let, $E \in Block_n^k$ be the block-cipher, where $(k, n) \in \{0,1\}^n$ means key and block length. The compression function $F^{E^{\text{FrS}}}(H^{\text{FrS}}) : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \rightarrow$

$\{0,1\}^{2n}$ contains of three block-ciphers, defined as follows:

$$
\begin{array}{l}
E_1 \leftarrow E_1\left(m_i, x_{i-1}\right) = \alpha_i \\
E_2 \leftarrow E_2\left(m_i, y_{i-1}\right) = \beta_i \\
E_3 \leftarrow E_3\left(m_i, \gamma_i\right) = \mu_i \\
where, \\
\left.\begin{array}{l}
x_i \leftarrow \varphi_i \oplus x_{i-1} \,|\varphi_i \leftarrow^{\otimes^{upper}} \{(\alpha_i \| \beta_i)\} \\
\gamma_i \leftarrow^{\otimes^{lower}} \{(\alpha_i \| \beta_i)\} \\
y_i \leftarrow \mu_i \oplus y_{i-1}
\end{array}\right| upper, lower \in \{0,1\}^n
\end{array}
$$

Finally,

$$
\begin{array}{l}
H^{\mathrm{FrS}}\left(x_{i-1}, y_{i-1}, m_i\right) \\
= \left(\begin{array}{l}
E_1\left(x_{i-1}, m_i\right) \| E_2\left(y_{i-1}, m_i\right) \rightarrow^{\otimes} \varphi \oplus x_{i-1} \rightarrow x_i, \\
E_3\left(\gamma_i, m_i\right) \rightarrow \mu \oplus y_{i-1} \rightarrow y_i
\end{array} \right)
\end{array}
$$

Security Analysis of the Proposed Fourth scheme

Usually, the ICM is widely used for the security proof of block-cipher hash [61, 62, 63]. The WCM is better than the ICM for its weaker assumption [61, 62, 63]. The target of the adversary will be unique under the both security proof models. We assume that the adversary $\mathcal{A}$ can get access to the block-cipher $(Block_n^k)$ oracle. It tries to find a collision under $H^{\mathrm{FrS}}\left(F^{E^{\mathrm{FrS}}}\right)$ through the following conditions (Table 4.10).

Table 4.10: Conditions of collision occurrence

Conditions (x, y : chaining value, x_0, y_0 : initial value, m : message)
1. $\left(x_{i-1}, y_{i-1}, m_i\right), \left(x_{i'-1}, y_{i'-1}, m_{i'}\right) \,\vert i' < i < q$ $H^{\mathrm{FrS}}\left(x_{i-1}, y_{i-1}, m_i\right) = H^{\mathrm{FrS}}\left(x_{i'-1}, y_{i'-1}, m_{i'}\right)$ $\vert \left(x_{i-1}, y_{i-1}, m_i\right) \neq \left(x_{i'-1}, y_{i'-1}, m_{i'}\right)$
2. $\left(x_{i-1}, y_{i-1}, m_i\right) \vert i < q \,; H^{\mathrm{FrS}}\left(x_{i-1}, y_{i-1}, m_i\right) \rightarrow \left(x_i = y_i\right)$
3. $\left(x_{i-1}, y_{i-1}, m_i\right) \vert i < q \,;\; H^{\mathrm{FrS}}\left(x_{i-1}, y_{i-1}, m_i\right) = x_0, y_0$

Collision Security Analysis (*ICM based*). Under the ICM, the adversary $\mathcal{A}$ is allowed to make two types of query to the oracle of block-cipher $(Block\,(n,k))$ such as forward and backward query. An adversary $\mathcal{A}$ can get cipher-text through forward query, where a backward query provides plain-text. The query is noted as $Q_i | i < q$. After each iteration, a query will be stored at $\mathcal{Q} | (\mathcal{Q} \in Q_i, Q_{i+1}, \ldots, Q_q) \wedge (1 < i \leq q)$, where query looks $Q \in (x, y, m)$ $[x, y, m =$ chaining value, message$]$. We will follow the certain conditions from 5.10 for finding collision hit under the ICM.

Theorem 4.8. Let H^{FrS} be a block-cipher hash function consists of compression function $F^{E^{\mathrm{FrS}}}$ (Definition 4.5 and Figure 4.10). The task of the adversary $\mathcal{A}$ is to find collision through H^{FrS} after q queries. Therefore, the adversarial advantage is bounded as:

$$
\mathrm{Adv}^{\mathrm{coll}}_{H^{\mathrm{FrS}}}(q) \leq \frac{q^2 - 2q}{\left(2^n - 3q\right)^2}
$$

Proof. The adversary $\mathcal{A}$ will ask to the block-cipher oracle until it doesn't get success. As for example, after i'-th query the query set looks $(Q_{i'} \in (x', y', m'))$. For next any iteration, there is a chance to find a query $(Q_i \in (x, y, m))|\,(i' < i)$ that produces the same output as the output of i' iteration. There are two more conditions for collision hit, which are available in the Table 4.10.

Condition-1. For the first condition, it needs two iterations of H^{FrS}. It means, the adversary $\mathcal{A}$ tries to find a collision (Table 4.10) for two different set of query. We assume that $Ev^{\mathrm{coll}}_{\text{condtion-1}}$ be the event for finding a collision under the $H^{\mathrm{FrS}}\left(F^{E^{\mathrm{FrS}}}\right)$. Our scheme needs three calls of block-cipher per iteration by construction. Therefore, the collision probability for a event of $Ev^{\mathrm{coll}}_{\text{condition-1}}|\,(i' < i)$ will be:

$$\Pr[Ev^{\mathrm{coll}}_{\text{condition-1}}] = \frac{i}{(2^n - 3i)(2^n - 3i)}$$

If $Ev^{\mathrm{coll}}_{\text{condition-1}}|\,(i' < i)$ be the event of finding a collision under the $F^{E^{\mathrm{FrS}}}$, then the probability of collision events after q queries will be $\Pr[Ev^{\mathrm{coll}}_{\text{condition-1}}] = \Pr[Ev^{\mathrm{coll}}_{3,\text{condtion-1}} \vee Ev^{\mathrm{coll}}_{4,\text{condition-1}} \vee \ldots \vee Ev^{\mathrm{coll}}_{q,\text{condition-1}}]$.

$$= \sum_{i=3}^{q} \Pr[Ev^{\mathrm{coll}}_{i,\text{condition-1}}] = \sum_{i=3}^{q} \frac{i}{(2^n - 3i)\,(2^n - 3i)} \leq \frac{(q-2)\,(q-3)}{(2^n - 3q)^2} \tag{4.29}$$

Condition-2. According to Table 4.10, there is a scope for collision hit within a single query. Let, $Ev^{\mathrm{coll}}_{\text{condition-2}}$ be the event for finding a collision under the $H^{\mathrm{FrS}}\left(F^{E^{\mathrm{FrS}}}\right)$, where three calls of block-cipher will be executed per iteration. Hence, the collision probability for the event of $Ev^{\mathrm{coll}}_{\text{condition-2}}$ will be $\Pr\left[Ev^{\mathrm{coll}}_{\text{condition-2}}\right] = \frac{1}{(2^n-3i)(2^n-3i)}$. The probability of collision events are:

$$\Pr[Ev^{\mathrm{coll}}_{\text{condition-2}}] = \Pr[Ev^{\mathrm{coll}}_{3,\text{condition-2}} \vee Ev^{\mathrm{coll}}_{4,\text{condition-2}} \vee \ldots\ldots \vee Ev^{\mathrm{coll}}_{q,\text{condition-2}}]$$

$$= \sum_{i=3}^{q} \Pr[Ev^{\mathrm{coll}}_{i,\text{condtion-2}}] = \sum_{i=3}^{q} \frac{1}{(2^n - 3i)(2^n - 3i)} \leq \frac{(q-2)}{(2^n - 3q)^2} \tag{4.30}$$

Condition-3. We know for any block-cipher based hash (compression function), it needs initialization value. Let, there is a possibility for the adversary to get a collision under these initializing values at any stage of query process. We assume that $Ev^{coll}_{\text{condition-3}}$ be the event for finding a collision against the set of initialization value through $H^{\mathrm{FrS}}\left(F^{E^{\mathrm{FrS}}}\right)$. The collision probability for the event of $Ev^{\mathrm{coll}}_{\text{condition-3}}$ will be:

$$\Pr[Ev^{\mathrm{coll}}_{\text{condition-3}}] = \frac{2}{(2^n - 3i)(2^n - 3i)}$$

Therefore, the probability of collision events are:

$$\Pr[Ev^{\mathrm{coll}}_{\text{condition-3}}] = \Pr[Ev^{\mathrm{coll}}_{3,\text{condition-3}} \vee Ev^{\mathrm{coll}}_{4,\text{condition-3}} \vee \ldots \vee Ev^{\mathrm{coll}}_{q,\text{condition-3}}] =$$

$$\sum_{i=3}^{q} \Pr[Ev^{\mathrm{coll}}_{i,\text{condition- 3}}] = \sum_{i=3}^{q} \frac{2}{(2^n - 3i)(2^n - 3i)} \leq \frac{2\,(q-2)}{(2^n - 3q)^2} \tag{4.31}$$

Taking the values of 4.29, 4.30, and 4.31 **Theorem 4.8** is satisfied.

Preimage Security Analysis(*ICM based*). The preimage resistance of the (n, n) block-cipher hash usually is bounded by $O\,(2^n)$ [61, 62, 63]. The probability of preimage hit

comes from the set size of $1/(2^n - q)$ where the parameters are defined as (2^n = domain size) and (q = number of query) [25, 39]. If the number of query goes to the equal value of 2^n, the denominator will be 0 and result will be useless. The above problem is first addressed by [58] in Asiacrypt 2011. Also authors of [58] provide a new technique for eliminating this problem as well as better preimage security bound. We will follow the proof technique of [58] for our scheme's security proof and implement according to our scheme's definition. We assume that a $\mathcal{A}$ be the adversary that can ask a set of pair-query to the block-cipher oracle. Initially, $\mathcal{A}$ picks the value of (x', y') randomly. The target of $\mathcal{A}$ is to find the probability for any iteration $(i \leq q)$, where $H^{\text{FrS}}(x, y, m) = \{(x', y')\}$.

Theorem 4.9. Let H^{FrS} ($F^{E^{\text{FrS}}}$) be a double block length compression function and $\mathcal{A}$ be an adversary to find a preimage hit under the H^{FrS} after q queries. Then the advantage of adversary $\mathcal{A}$ is bounded as:

$$\text{Adv}^{\text{pre}}_{H^{\text{FrS}}}(q) \leq {8(q-2)^2}/{(2^n - 3q)^2} + \frac{4q^2}{2^{2n}}$$

Proof. According to the [58], we will take the concept of query classification. The query classification is classified as super query and normal query [25, 58]. The normal query is based on adaptive query, where non-adaptive method is true for super query [25, 58]. At first, adversary $\mathcal{A}$ will ask to the oracle adaptively until the size of database reaches into $N/2$. Then the rest of the queries provide to the adversary as free [25, 58], where database size will be $N/2$. In the later half, the query will be asked non-adaptively. If the preimage hit occurs in the database of normal query, then it is defined as NormalQueryWin otherwise it is called SuperQueryWin [25, 58]. Therefore, we need to find out the probability of hitting for NormalQueryWin and SuperQueryWin.

Condition-1. For NormalQueryWin, the adversary $\mathcal{A}$ will ask to the oracle through either forward or backward query. We assume that $\mathcal{A}$ makes a forward query. The result will be come from the set size $(N - 3i)/2$ (due to three calls of blockcipher). Therefore, the probability of the output $(x_i, y_i | (i \leq N/2))$ will be $2 \times 2/(N - 3i)$. Let $Ev^{\text{pre(forward)}}_{\text{condition-1}}$ be the event of preimage hitting. Hence, the probability of preimage hitting events are $\Pr[Ev^{\text{pre(forward)}}_{\text{condtion-1}}] = \Pr[Ev^{\text{pre}}_{3,\text{condtion-1}} \vee Ev^{\text{pre}}_{4,\text{condtion-1}} \vee \ldots \vee Ev^{\text{pre}}_{q,\text{condtion-1}}]$.

$$= \sum_{i=3}^{q} \Pr[Ev^{\text{pre(forward)}}_{i,\text{condtion-1}}] = \sum_{i=3}^{q} \frac{4}{(2^n - 3i)} \leq \frac{4(q-2)}{(2^n - 3q)} \tag{4.32}$$

For any forward query (encryption), there is a chance to occur a preimgae hit under the backward query (decryption). If $Ev^{\text{pre(backward)}}_{\text{condition-1}}$ be the event of preimage hit, then the probability will be:

$$\sum_{i=3}^{q} \Pr[Ev^{\text{pre(backward)}}_{i,\text{condtion-1}}] = \sum_{i=3}^{q} \frac{2}{(2^n - 3i)} \leq \frac{2(q-2)}{(2^n - 3q)} \tag{4.33}$$

From 4.32 and 4.33,

$$\Pr[\text{Condition-1}] = \frac{8(q-2)^2}{(2^n - 3q)^2} \tag{4.34}$$

Condition-2. For SuperQueryWin [25, 58], the adversary $\mathcal{A}$ will pick the value of x_i, y_i non-adaptively from the super query database, where domain size is $N/2$. The sub conditions of condition-2 are as follows for either forward or backward query:

$$H^{\text{FrS}}(x_{i-1}, y_{i-1}) = (x_i, y_i) = (x'/y') \tag{4.35}$$

$$H^{\mathrm{FrS}}(x_{i-1}, y_{i-1}) = (x_i, y_i) = (y'/x') \tag{4.36}$$

We assume that $Ev^{\mathrm{pre(forward)}}_{\text{condition-2}}$ be the event of preimage hit. Then the probability of preimage hitting events are $\Pr[Ev^{\mathrm{pre(forward)}}_{\text{condtion-2}}] = \Pr[Ev^{\mathrm{pre}}_{3,\text{condtion-2}} \vee Ev^{\mathrm{pre}}_{4,\text{condtion-2}} \vee \ldots \vee Ev^{\mathrm{pre}}_{q,\text{condtion-2}}]$.

$$= \sum_{i=3}^{q} \Pr[Ev^{\mathrm{pre(forward)}}_{i,\text{condtion-2}}] = \sum_{i=1}^{q} \frac{2}{2^n} \leq \frac{2q}{2^n} \tag{4.37}$$

The result of 4.36 is (as same explanation of 4.37):

$$\sum_{i=3}^{q} \Pr[Ev^{\mathrm{pre(backward)}}_{i,\text{condtion-2}}] = \sum_{i=3}^{q} \frac{2}{2^n} \leq \frac{2q}{2^n} \tag{4.38}$$

Therefore, from 4.37 and 4.38,

$$\Pr[\text{Condition-2}] = \frac{4q^2}{2^{2n}} \tag{4.39}$$

Finally, adding the values of 4.34 and 4.39, **Theorem 4.9** is satisfied.

Collision Security Analysis (*WCM based*). The adversary $\mathcal{A}$ is allowed to make three types of query under the WCM. These are defined as forward, backward and key-disclosure query (E, E^{-1}, E^k). The query is noted as $Q_i | i < q$ and defined as $Q \in (x, y, m)$. The $\mathcal{Q} | ((\mathcal{Q} \in Q_i, Q_{i+1}, .., Q_q) \wedge (1 < i \leq q))$ be the query database, where after each iteration query has been stored.

Theorem 4.10. Let H^{FrS} be a block-cipher based hash consists of a compression function $F^{E^{\mathrm{FrS}}}$ (Definition 4.5 and Figure 4.10) and $\mathcal{A}$ be an adversary to find a collision hit under the H^{NEW}. After q queries, the advantage of adversary $\mathcal{A}$ is bounded as:

$$\mathrm{Adv}^{\mathrm{coll}}_{H^{\mathrm{FrS}}}(q) \leq \frac{3q^2 - 12q}{(2^n - 3q)^2}$$

Proof. The adversary $\mathcal{A}$ can make forward, backward and key-disclosure query under the WCM. In forward query, the adversary can ask for cipher-text through plain-text and key. A backward query returns the plain-text and key-disclosure query is responsible for the key. Under any j-th iteration, a query will be $Q_j \in \{x_{j-1}, y_{j-1}, m_j\}$. We assume there is another iteration $i | (j < i < q)$, where query looks $Q_i \in \{x_{i-1}, y_{i-1}, m_i\}$. For finding a collision under $Q_j \in \{x_{j-1}, y_{j-1}, m_j\}$ and $Q_i \in \{x_{i-1}, y_{i-1}, m_i\}$ the probable conditions are:

$$\begin{aligned} &H^{\mathrm{FrS}}(x_{j-1}, y_{j-1}, m_j) = H^{\mathrm{FrS}}(x_{i-1}, y_{i-1}, m_i) \\ &\vee H^{\mathrm{FrS}}(x_{j-1}, y_{j-1}, m_j) = (x_0, y_0) | (x_{j-1}, y_{j-1}, m_j) \neq (x_{i-1}, y_{i-1}, m_i) \end{aligned} \tag{4.40}$$

forward query. Under the forward query, we assume that $Ev^{\mathrm{coll}}_{\mathrm{forward}}$ be the event for finding a collision through $H^{\mathrm{FrS}}\left(F^{E^{\mathrm{FrS}}}\right)$. Our scheme needs three calls of block-cipher per iteration by construction. According to 4.40, the collision probability for the event of $Ev^{\mathrm{coll}}_{\mathrm{forward}} | (j < i)$ is:

$$\Pr[Ev^{\mathrm{coll}}_{\mathrm{forward}}] = \frac{i}{(2^n - 3i)(2^n - 3i)} + \frac{1}{(2^n - 3i)(2^n - 3i)}$$

If $Ev^{\text{coll}}_{\text{forward}}|\,(j<i)$ be the event for finding a collision under the $F^{E^{\text{FrS}}}$ for q pairs of queries. Then the probability of collision events are $\Pr[Ev^{\text{coll}}_{\text{forward}}] = \Pr[Ev^{\text{coll}}_{3,\text{forward}} \vee Ev^{\text{coll}}_{4,\text{forward}} \vee \ldots \vee Ev^{\text{coll}}_{q,\text{forward}}]$.

$$= \sum_{i=3}^{q} \Pr[Ev^{\text{coll}}_{i,\text{forward}}] = \sum_{i=3}^{q} \frac{i}{(2^n-3i)\,(2^n-3i)} + \frac{1}{(2^n-3i)\,(2^n-3i)} \leq \frac{q^2-4q+3}{(2^n-3q)^2} \tag{4.41}$$

backward query. According to the 4.40 and including similar explanation of the forward query, the probability of backward query is:

$$= \sum_{i=3}^{q} \Pr[Ev^{\text{coll}}_{i,\text{backward}}] = \sum_{i=3}^{q} \frac{i}{(2^n-3i)\,(2^n-3i)} + \frac{1}{(2^n-3i)\,(2^n-3i)} \leq \frac{q^2-4q+3}{(2^n-3q)^2} \tag{4.42}$$

key-disclosure query. According to the 4.40 and including similar explanation of the forward query, the probability of key-disclosure is:

$$= \sum_{i=3}^{q} \Pr[Ev^{\text{coll}}_{i,\text{key-disclosure}}] = \sum_{i=3}^{q} \frac{i}{(2^n-3i)\,(2^n-3i)} + \frac{1}{(2^n-3i)\,(2^n-3i)} \leq \frac{q^2-4q+3}{(2^n-3q)^2} \tag{4.43}$$

Adding the results of 4.41, 4.42 and 4.43, **Theorem 4.10** is proved.

Preimage Security Analysis (*WCM based*). In preimage security analysis, adversary $\mathcal{A}$ randomly selects (x', y') at the beginning point of query process. Therefore, adversary looks for the query-input of x, y, m that can produce the output of $H^{\text{FrS}}(x, y, m)$ such that $H^{\text{FrS}}(x, y, m) = (x', y')$

Theorem 4.11. Let H^{FrS} be a block-cipher based compression function and $\mathcal{A}$ be an adversary to find a preimage hit under the $H^{\text{FrS}}\left(F^{E^{\text{FrS}}}\right)$ after q pairs of queries. Hence, the advantage of $\mathcal{A}$ is bounded as:

$$\text{Adv}^{\text{pre}}_{H^{\text{FrS}}}(q) \leq \frac{2q-4}{(2^n-3q)^2}$$

Proof. Adversary $\mathcal{A}$ can make forward, backward or key-disclosure query for finding the following condition:

$$H^{\text{FrS}}(x, y, m) = (x', y'), \text{where}\,(i<q) \tag{4.44}$$

forward query. Under the forward query, we assume that $Ev^{\text{pre}}_{\text{forward}}$ be the event for finding a preimage hit through the $H^{\text{FrS}}\left(F^{E^{\text{FrS}}}\right)$. According to the 4.44, the preimage hit probability is:

$$\Pr[Ev^{\text{pre}}_{\text{forward}}] = \frac{2}{(2^n-3i)(2^n-3i)}$$

If $Ev^{\text{pre}}_{\text{forward}}|\,(i<q)$ be the event for finding a preimage hit through $F^{E^{\text{FrS}}}$ for q pairs of queries. Then the probability of preimage hitting events are $\Pr[Ev^{\text{pre}}_{\text{forward}}] = \Pr[Ev^{\text{pre}}_{3,\text{forward}} \vee Ev^{\text{pre}}_{4,\text{forward}} \vee \ldots \vee Ev^{\text{pre}}_{q,\text{forward}}]$.

$$= \sum_{i=3}^{q} \Pr[Ev^{\text{pre}}_{i,\text{forward}}] = \sum_{i=3}^{q} \frac{2}{(2^n-3i)\,(2^n-3i)} \leq \frac{2q-4}{(2^n-3q)^2} \tag{4.45}$$

backward query. As same explanation of the forward query, the probability of backward query is:

$$= \sum_{i=3}^{q} \Pr[Ev^{\text{pre}}_{i,\text{backward}}] = \sum_{i=3}^{q} \frac{2}{(2^n - 3i)(2^n - 3i)} \leq \frac{2q-4}{(2^n - 3q)^2} \tag{4.46}$$

key-disclosure query. The probability of the key-disclosure query is:

$$= \sum_{i=3}^{q} \Pr[Ev^{\text{pre}}_{i,\text{key-disclosure}}] = \sum_{i=3}^{q} \frac{2}{(2^n - 3i)(2^n - 3i)} \leq \frac{2q-4}{(2^n - 3q)^2} \tag{4.47}$$

Adding the results of 4.45, 4.46 and 4.47, **Theorem 4.11** is satisfied.

Chapter 5

A Pair of Constructions of Authenticated Encryption

An authentication encryption (AE) scheme satisfies to transfer an authenticated data between two parties or more [1, 2, 3]. There are vast applications of the AE such as access control, encryption, enhancing trust between multiple parties, and assure the originality of a message [11, 12, 19, 20]. However, the main challenge of the AE is to maintain low-cost features for it's construction. Furthermore, there is another emerging issue of IoT in the field of data and network communication [7, 82, 86, 87, 88]. The numbers of application of the IoT are increasing expeditiously, where various kinds of device have been used such as IoT-end device, constrained device, and RfID. Moreover, the main challenge of the IoT-end devices, and resource constrained devices is to keep a certain level of security bound including minimum cost. However, the IoT-end devices, resource constrained devices, and RfID have lack of resources such as memory, power, and processors. Interestingly, the AE can play a vital role between data acquisition (sensors, actuators) and data aggregation of usual platform of the IoT. Thus, the construction of the AE should satisfies the properties of low-cost, least resources and less operating-time. Though, there are many familiar constructions of AE such as OTR, McOE, POE, OAE, APE, COPE, CLOC, and SILK but most of the schemes depend on the features of nonce and associate data. In the aspect of security, the usage of nonce and associated data are adequate. However, these two features increase the overhead cost. Therefore, we propose a simple construction of probabilistic-IV based AE (First Scheme: FS) where block-cipher compression function is used as encryption function.

Security, privacy and data integrity are the critical issues in Big Data application of IoT-enable environment and cloud-based services. There are many upcoming challenges to establish secure computations for Big Data applications. Authenticated encryption (AE) plays one of the core roles for Big Data's confidentiality, integrity, real-time security, authenticity [7, 12, 95]. There are many proposals in the research area of authenticated encryption. Among those schemes, one of the prominent issues is security notion of nonce-reuse in AE. Interestingly E.Fleischmann et. al. claimed that the scheme of McOE satisfies the properties of nonce-reuse AE. However, the concept of nonce-reuse online AE is reconciled later by V.T.Hoang et. al. in Crypto2015. Therefore, we consider the issue of nonce respect and probabilistic-IV in authenticated encryption and propose two simple constructions, which are efficient in certain contexts and suitable for IoT applications. Our first scheme is based on probabilistic-IV. This scheme operates in serial. Hence, we notify this

scheme as Serial-AE also. In addition, it is expected to be a light solution due its weaker security model. The first scheme (Serial-AE) needs $n + n \times f^{\text{prng}} + 2$ resources for encryption mode. Moreover, we provide three types (variant) of tag generation (authentication) under the first scheme. The first variant needs $(n-1)+1$ calling of block-cipher and it operates in semi-parallel. On the contrary, the second variant of tag generation is based on serial operation and it needs $(n+1)+3$ calling of block-cipher to create tag. And the third variant needs only two calls of block-cipher function. Our second scheme (SS) is based on nonce respect AE and it operates in parallel mode. Therefore, we call this scheme as Parallel-AE also. It only supports fixed size of associated data like n-bit in the initialization phase. Under this context, it is suitable for IoT application. The second scheme (Parallel-AE) needs of resources $m + (m \times GF) + 2$ for encryption. Moreover, we provide two types of tag generation under the second scheme (Parallel-AE). The first variant runs in semi-parallel. It needs $(n-1)+1$ calling of function. The second variant is based cryptographic compression function $(3n \rightarrow 2n)$-bit where $n+2$ encryption functions are needed.

5.1 Probabilistic-IV based AE

The first scheme of AE is based on probabilistic-IV. It is secure under the weaker security model. It runs in serial mode. Hence we call this scheme as Serial-AE also. In addition, we provide three variants (types) of tag generation (authentication) under the scheme of Serial-AE. These are named as Semi-Parallel Tag generation (Semi-Parallel-T.G), Serial Tag generation (Serial-T.G), and Parallel Tag generation (Parallel-T.G). Hence, in combine form these are Serial-AE: Semi-Parallel-T.G, Serial-AE: Serial-T.G, and Serial-AE: Parallel-T.G. In principle, encryption mode is similar for all constructions. More clearly, the scheme of Serial-AE consists the part of encryption and tag generation. And, we provide three variants of tag generation under the scheme of Serial-AE, where encryption part is fixed.

5.2 Preliminaries for Serial Authenticated Encryption

At first, we mention that we provide three variants of authentication or tag generation under the scheme of Serial-AE where encryption part is unique. Under the scheme of Serial-AE, M is defined as message set where $m_i^j \in M$ such that $(m_1^1, m_1^2), .., \left(m_l^1, m_l^2\right)$, where $j \in \{1,2\}, i \leq l$. On the contrary, cipher-text (C) is noted as $c_i^j \in C$ such that $(c_1^1, c_1^2), .., \left(c_l^1, c_l^2\right)$ where $j \in \{1,2\}, i \leq l$. Moreover, T is defined as final Tag. We also define certain operators such as $\oplus$: ex-or, and $\bullet$: inverse. We define block-cipher $(\mathcal{E})$ in the scheme of Serial-AE as $\mathcal{E}_{k \oplus a}(b) \rightarrow c$. In principle, our assumption is $c \leftarrow \mathcal{E}_{k \oplus a}(b) \neq c' \leftarrow \mathcal{E}_{k \oplus b}(a)$. Moreover, we define a PRNG function (F^{prng}) in the encryption mode of the fist scheme. The operation of F^{prng} is to take n-bit string and return $2n$-bit random string. In mathematically, we can deduce like $F^{\text{prng}}(x) \rightarrow y_1, y_2$ where $x, y_1, y_2 \in \{0,1\}^n$. In principle, the first scheme operates in serial mode. Under the scheme of Serial-AE, n numbers of block-cipher plus $n \times f^{\text{prng}}$ functions plus 2 initialization block-ciphers are needed for encryption part. However, the cost of tag generation (authentication) under the scheme of Serial-AE is varied

because three types of authentication.

5.2.1 Proposed Scheme of Serial-AE: Semi-Parallel-T.G

The scheme of Serial-AE: Semi-Parallel-T.G is noted as $\text{AE}^{\text{FS}}_{T.V1}$ where FS: First Scheme, T tag, $V1$: First variant (Fig. 5.1). This scheme has three phases. The first phase (PH-1) is responsible for initialization. The second phase is based on an encryption module of e-$\text{AE}^{\text{FS}}_{T.V1}$. The task of this phase is to generate cipher-text and tag. Moreover, third phase (PH-3) represents a decryption module $\left(\text{d-AE}^{\text{FS}}_{T.V1}\right)$ of Serial-AE: Semi-Parallel-T.G. In addition, the scheme of Serial-AE: Semi-Parallel-T.G follows serial operation but the authentication mode is based on semi-parallel. Algorithm 4 is called as initialization or Phase 1 (PH1). Encryption and decryption module are defined by algorithm 5 and 6.

Explanation of authentication Procedure of Semi-Parallel T.G There are list of: $(x_{i,1}, x_{i,2}, x_{i+1,1}, x_{i+1,2}, ..., x_{l,2}, x_{l,1}) \in x$ that are generated in encryption module of the Serial-AE: Semi-Parallel-T.G (Fig. 5.2, Algorithm 5, 6). Then, we re-index x like $(x_1, x_2, ..., x_l, x_{l+1}, x_{l'}) \in x$. For example, we take x as $(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, ..., x_{4,1}, x_{4,2}) \in x$. Then, we re-index x like $(x_1, x_2, ...x_7, x_8) \in x$.

Explanation of "For Each Level do: $x_i \leftarrow \mathcal{E}_{k \oplus x_{2i-1}}(x_{2i})$ [for, $i \in \{1, 2, .., |x|/2\}$]" command (Algorithm 5, 6: Line 13):

We encrypt $(x_1, x_2, ..., x_7, x_8) \in x$ pair by pair in each level (Fig. 5.1). In level 1, encrypt operation will perform like $x_1 \leftarrow \mathcal{E}_{k \oplus x_1}(x_2)$, $x_2 \leftarrow \mathcal{E}_{k \oplus x_3}(x_4)$, $x_3 \leftarrow \mathcal{E}_{k \oplus x_5}(x_6)$, $x_4 \leftarrow \mathcal{E}_{k \oplus x_7}(x_8)$. In level 2, operation is like $x_1 \leftarrow \mathcal{E}_{k \oplus x_1}(x_2)$, $x_2 \leftarrow \mathcal{E}_{k \oplus x_3}(x_4)$. Finally, the encryption is $x_1 \leftarrow \mathcal{E}_{k \oplus x_1}(x_2)$ in level 3. This x_1 is the final input of creating Tag (T). For this, we encrypt x_1 and $c_l^1 \oplus iv_l^1, c_l^2 \oplus iv_l^2$ as $T \leftarrow \mathcal{E}_{k \oplus c_l^2 \oplus iv_l^2}\left(x_1 \oplus c_l^1 \oplus iv_l^1\right)$. Moreover, if number of x is odd then the last $x_{i=l}$ is encrypted with x_1 of top level. In principle, our assumption is like $c \leftarrow \mathcal{E}_{k \oplus a}(b) \neq c' \leftarrow \mathcal{E}_{k \oplus b}(a)$.

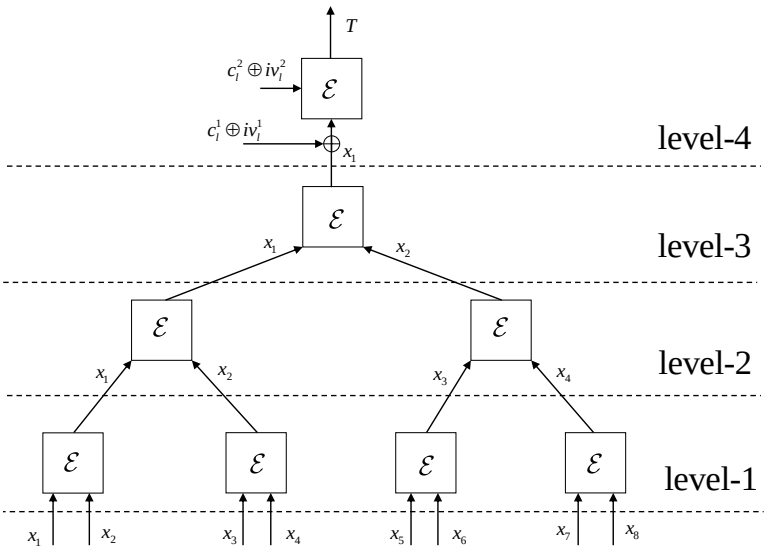


Figure 5.1: Explanation of Semi-Parallel-T.G

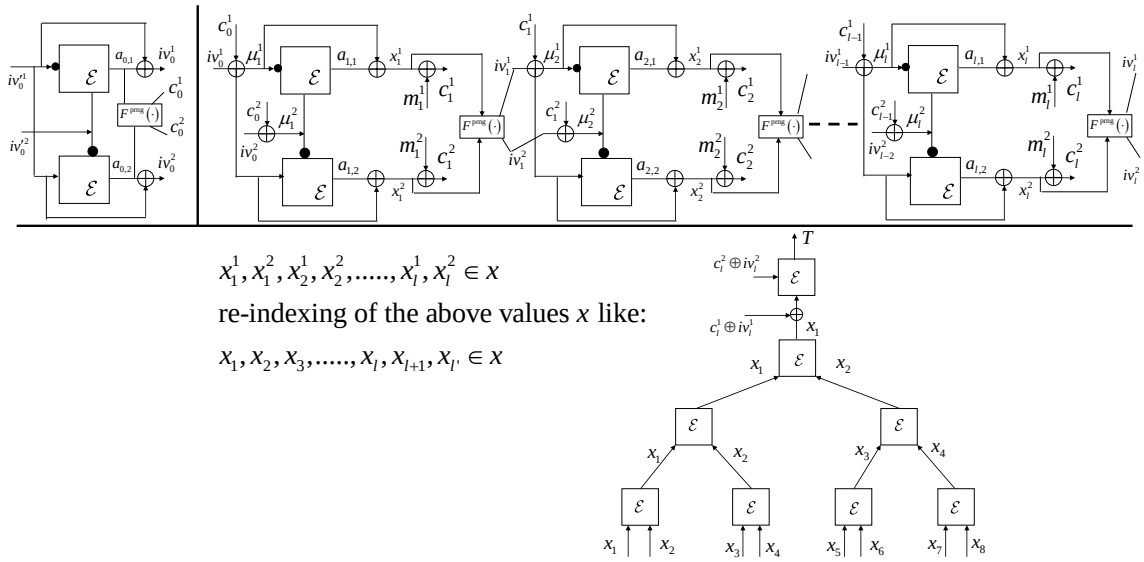


Figure 5.2: Proposed Scheme of Serial-AE: Semi-Parallel-T.G

Algorithm 4 Phase-1 (PH-1) for the Scheme of Serial-AE: Semi-Parallel-T.G

1: Initialization: iv'^1_0, iv'^2_0
2: $a_{0,1} \leftarrow \mathcal{E}_{k \oplus \overline{iv'^1_0}}\left(iv'^2_0\right),\ a_{0,2} \leftarrow \mathcal{E}_{k \oplus iv'^1_0}\left(\overline{iv'^2_0}\right)$
3: $iv^1_0 \leftarrow a_{0,1} \oplus iv'^1_0,\ iv^2_0 \leftarrow a_{0,2} \oplus iv'^1_0$
4: $c^1_0,\ c^2_0 \leftarrow F^{\text{prng}}\left(a_{0,1} \oplus a_{0,2}\right)$

Algorithm 5 Encryption module for the Scheme of Serial-AE: Semi-Parallel-T.G

1: Call PH-1
2: Encrypt M
3: Partitioning: $m^j_i \in M$ s. t. $\left(m^1_1, m^2_1\right), ., \left(m^1_l, m^2_l\right)$, where $j \in \{1,2\}, i \le l$
4: **for** $i = 1$ to l **do**
5: $\quad \mu^1_i \leftarrow iv^1_{i-1} \oplus c^1_{i-1},\ \mu^2_i \leftarrow iv^2_{i-1} \oplus c^2_{i-1}$
6: $\quad a_{i,1} \leftarrow \mathcal{E}_{k \oplus \overline{\mu^1_i}}\left(\mu^2_i\right),\ a_{i,2} \leftarrow \mathcal{E}_{k \oplus \mu^1_i}\left(\overline{\mu^2_i}\right)$
7: $\quad x_{i,1} \leftarrow a_{i,1} \oplus \mu^1_i,\ x_{i,2} \leftarrow a_{i,2} \oplus \mu^1_i$
8: $\quad c^1_i \leftarrow x_{i,1} \oplus m^1_i,\ c^2_i \leftarrow x_{i,2} \oplus m^2_i$
9: $\quad iv^1_i,\ iv^2_i \leftarrow F^{\text{prng}}\left(x_{i,1} \oplus x_{i,2}\right)$
10: **end for**
11: $C \leftarrow \left(c^1_1, c^2_1, c^1_2, c^2_2, ..., c^1_l, c^2_l\right),\ x \leftarrow \left(x^1_1, x^2_1, ..., x^1_l, x^2_l\right)$
12: re-indexing of x like $\left(x_1, x_2, ..., x_l, x_{l+1}, x_{l'}\right)$
13: **for** Each Level **do** $x_i \leftarrow \mathcal{E}_{k \oplus x_{2i-1}}\left(x_{2i}\right)$ [for, $i \in \{1, 2, .., |x|/2\}$]
14: **end for**
15: $T \leftarrow E_{k \oplus c^2_l \oplus iv^2_l}\left(x_1 \oplus c^1_l \oplus iv^1_l\right)$
16: Return (C, T)

Algorithm 6 Decryption module for the Scheme of Serial-AE: Semi-Parallel-T.G

Call PH-1
Decrypt C
Partitioning: $c_i^j \in C$ s. t. $\left(c_1^1, c_1^2\right), .., \left(c_l^1, c_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $\mu_i^1 \leftarrow iv_{i-1}^1 \oplus c_{i-1}^1,\ \mu_i^2 \leftarrow iv_{i-1}^2 \oplus c_{i-1}^2$
    $a_{i,1} \leftarrow \mathcal{E}_{k \oplus \overline{\mu_i^1}}\left(\mu_i^2\right),\ a_{i,2} \leftarrow \mathcal{E}_{k \oplus \mu_i^1}\left(\overline{\mu_i^2}\right)$
    $x_{i,1} \leftarrow a_{i,1} \oplus \mu_i^1,\ x_{i,2} \leftarrow a_{i,2} \oplus \mu_i^1$
    $m_i^1 \leftarrow x_{i,1} \oplus c_i^1,\ m_i^2 \leftarrow x_{i,2} \oplus c_i^2$
    $iv_i^1,\ iv_i^2 \leftarrow F^{\text{prng}}\left(x_{i,1} \oplus x_{i,2}\right)$
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right),\ x \leftarrow \left(x_1^1, x_1^2, ..., x_l^1, x_l^2\right)$
re-indexing of x like $\left(x_1, x_2, ..., x_l, x_{l+1}, x_{l'}\right)$
for Each Level **do** $x_i \leftarrow \mathcal{E}_{k \oplus x_{2i-1}}\left(x_{2i}\right)$ [for, $i \in \{1, 2, .., |x|/2\}$]
end for
$T \leftarrow E_{k \oplus c_l^2 \oplus iv_l^2}\left(x_1 \oplus c_l^1 \oplus iv_l^1\right)$
If T is valid then return M or $\perp$

5.2.2 Proposed Scheme of Serial-AE: Serial-T.G

Under the scheme of Serial-AE, we have three types of tag generation. In this section, we propose the second variant of tag generation including encryption under the scheme of Serial-AE. The second variant of tag generation is based on serial operation and named as Serial-T.G. Hence, the scheme of Serial-AE: Serial-T.G is noted as $\text{AE}^{\text{FS}}_{T.V2}$ where FS: First Scheme, T: tag, $V2$: Second variant (Fig. 5.3). This scheme has three phases. The first phase is called from the algorithm 4. The two other phases are encryption and decryption module. The encryption module of e-$\text{AE}^{\text{FS}}_{T.V2}$ generates cipher-text and tag. Moreover, decryption module $\left(\text{d-AE}^{\text{FS}}_{T.V2}\right)$ of the scheme of Serial-AE produces valid tag or not. For the explanation of encryption and decryption module, algorithm 7 and 8 are used. The variant of tag generation is based on $3n \rightarrow 2n$ bit cryptographic block-cipher compression function.

Under the encryption mode, our defined block-cipher $\mathcal{E}_{1,2}$ are operated like $\mathcal{E}_{k\oplus a}(b) \rightarrow c$. We assume that $c \leftarrow \mathcal{E}_{k\oplus a}(b) \neq c' \leftarrow \mathcal{E}_{k\oplus b}(a)$. In addition, we use E block-cipher in the authentication mode or tag generation. The operation of E is different from the $\mathcal{E}$. Actually, E works as $E_k(b) \rightarrow c$. In final stage of tag generation, we use three calls of block-cipher under W. We define the cryptographic compression function (W) like $\gamma_1 \leftarrow \mathcal{E}_{k\oplus\beta}(V_F)$, $\gamma_2 \leftarrow \mathcal{E}_{k\oplus V_F}(Z_F)$, and $\gamma_3 \leftarrow \mathcal{E}_{k\oplus Z_F}(\beta)$. Then, $t_1 \leftarrow \gamma_1 \oplus \gamma_2$ where $t_2 \leftarrow \gamma_2 \oplus \gamma_3$.

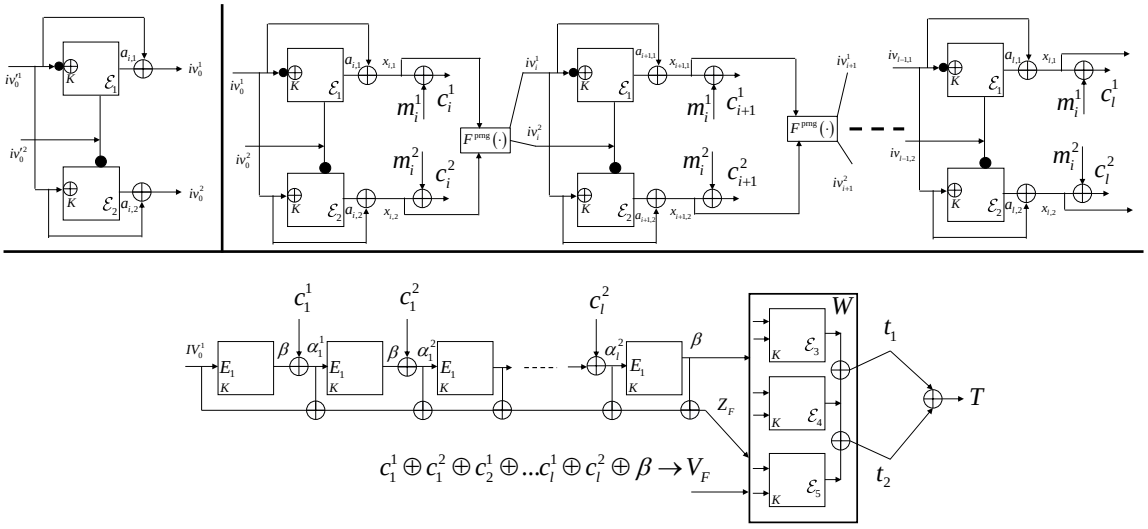


Figure 5.3: Proposed Scheme of Serial-AE: Serial-T.G

Phase-1 (PH-1) We run initialization and get iv'^1_0 and iv'^2_0. Then call two block-ciphers and get $a_{0,1} \leftarrow \mathcal{E}_{k\oplus\overline{iv'^1_0}}\left(iv'^2_0\right)$, $a_{0,2} \leftarrow \mathcal{E}_{k\oplus iv'^1_0}\left(\overline{iv'^2_0}\right)$. Finally, we get $iv^1_0 \leftarrow a_{0,1} \oplus iv'^1_0$, $iv^2_0 \leftarrow a_{0,2} \oplus iv'^1_0$.

Algorithm 7 Encryption module: Serial-AE: Serial-T.G

Call PH-1 (Page 66)
Encrypt M
Partitioning: $m_i^j \in M$ s. t. $\left(m_1^1, m_1^2\right), .., \left(m_l^1, m_l^2\right)$, where $j \in \{1, 2\}, i \leq l$
for $i = 1$ to l **do**
    $a_{i,1} \leftarrow \mathcal{E}_{k \oplus \overline{iv_{i-1}^1}}\left(iv_{i-1}^2\right),\ a_{i,2} \leftarrow \mathcal{E}_{k \oplus iv_{i-1}^1}\left(\overline{iv_{i-1}^2}\right)$
    $x_{i,1} \leftarrow a_{i,1} \oplus iv_{i-1}^1,\ x_{i,2} \leftarrow a_{i,2} \oplus iv_{i-1}^1$
    $iv_i^1,\ iv_i^2 \leftarrow F^{\text{prng}}\left(x_{i,1} \oplus x_{i,2}\right)$
    $c_i^1 \leftarrow x_{i,1} \oplus m_i^1,\ c_i^2 \leftarrow x_{i,2} \oplus m_i^2$
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$
$\beta \leftarrow E_k\left(iv_0^1\right)$
for $i = 1$ to l **do**
    for $j = 1$ to 2 **do**
        $\alpha_i^j \leftarrow \beta \oplus c_i^j$
        $\beta \leftarrow E_k\left(\beta \oplus c_i^j\right)$
    end for
end for
$V_F \leftarrow \left(c_1^1 \oplus c_1^2 \oplus \cdots \oplus c_l^1 \oplus c_l^2 \oplus \beta\right)$
$Z_F \leftarrow \left(iv_0^1 \oplus \alpha_1^1 \oplus \alpha_1^2 \oplus \cdots \oplus \alpha_l^1 \oplus \alpha_l^2 \oplus \beta\right)$
$\gamma_1 \leftarrow E_{k \oplus \beta}\left(V_F\right),\ \gamma_2 \leftarrow E_{k \oplus V_F}\left(Z_F\right),\ \gamma_3 \leftarrow E_{k \oplus Z_F}\left(\beta\right)$
$t_1 \leftarrow \gamma_1 \oplus \gamma_2,\ t_2 \leftarrow \gamma_2 \oplus \gamma_3$
$T \leftarrow t_1 \oplus t_2$
Return C, T

Algorithm 8 Decryption module: Serial-AE: Serial-T.G

Call PH-1 (Page 66)
Decrypt C
Partitioning: $c_i^j \in C$ s. t. $\left(c_1^1, c_1^2\right), .., \left(c_l^1, c_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $a_{i,1} \leftarrow \mathcal{E}_{k \oplus \overline{iv_{i-1}^1}}\left(iv_{i-1}^2\right),\ a_{i,2} \leftarrow \mathcal{E}_{k \oplus iv_{i-1}^1}\left(\overline{iv_{i-1}^2}\right)$
    $x_{i,1} \leftarrow a_{i,1} \oplus iv_{i-1}^1,\ x_{i,2} \leftarrow a_{i,2} \oplus iv_{i-1}^1$
    $iv_i^1,\ iv_i^2 \leftarrow F^{\text{prng}}\left(x_{i,1} \oplus x_{i,2}\right)$
    $m_i^1 \leftarrow x_{i,1} \oplus c_i^1,\ m_i^2 \leftarrow x_{i,2} \oplus c_i^2$
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$
$\beta \leftarrow E_k\left(iv_0^1\right)$
for $i = 1$ to l **do**
    for $j = 1$ to 2 **do**
        $\alpha_i^j \leftarrow \beta \oplus c_i^j$
        $\beta \leftarrow E_k\left(\beta \oplus c_i^j\right)$
    end for
end for
$V_F \leftarrow \left(c_1^1 \oplus c_1^2 \oplus \cdots \oplus c_l^1 \oplus c_l^2 \oplus \beta\right)$
$Z_F \leftarrow \left(iv_0^1 \oplus \alpha_1^1 \oplus \alpha_1^2 \oplus \cdots \oplus \alpha_l^1 \oplus \alpha_l^2 \oplus \beta\right)$
$\gamma_1 \leftarrow E_{k \oplus \beta}\left(V_F\right),\ \gamma_2 \leftarrow E_{k \oplus V_F}\left(Z_F\right),\ \gamma_3 \leftarrow E_{k \oplus Z_F}\left(\beta\right)$
$t_1 \leftarrow \gamma_1 \oplus \gamma_2, t_2 \leftarrow \gamma_2 \oplus \gamma_3$
$T \leftarrow t_1 \oplus t_2$
If T is valid then return M or $\perp$

5.2.3 Proposed Scheme of Serial-AE: Parallel-T.G

In this section, we define the third variant of tag generation including encryption mode under the scheme of Serial-AE. Our define tag generation is based on parallel mode. We notify this as Parallel-T.G. Therefore, in combine we can call as Serial-AE: Parallel-T.G (Figure 5.4). It has three phases. The first phase (Phase-1) is called as initialization. In addition, algorithm 9 represents the first phase. Moreover, 10 and 11 represent the encryption and decryption mode. Our define block-cipher E is worked as $E_{k\oplus a}(b) \rightarrow c$. In principle, our assumption is $c \leftarrow E_{k\oplus a}(b) \neq c \leftarrow E_{k\oplus b}(a)$.

In the initialization phase, we use two calls of block-ciphers including a PRNG function. The block-ciphers are used to build two secret values of $c_{0,1}$ and $c_{0,2}$. In addition, the PRNG function is called to build two more secret values of $iv_0^1,\ iv_0^2$ using the values of $a_{0,1}$ and $a_{0,2}$. In the encryption mode, we use cipher-text as feed forward in the next iteration. Hence, every output of each iteration depends on the previously generated cipher-text. In the tag generation, we xor the last output of $iv_l^1,\ iv_l^1$ with $c_l^1,\ c_l^1$.

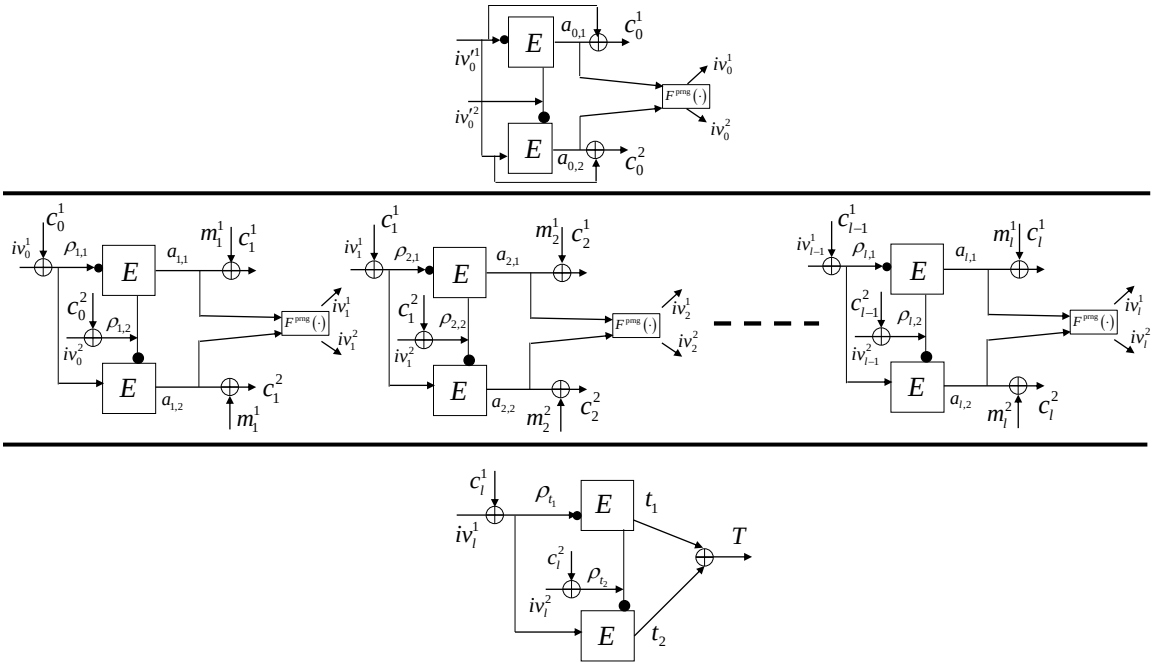


Figure 5.4: Proposed Scheme of Serial-AE: Parallel-T.G

Algorithm 9 Phase-1 (PH-1) of Serial-AE: Parallel-T.G

1: Initialization: $iv'^1_0,\ iv'^2_0$
2: $a_{0,1} \leftarrow E_{k\oplus\overline{iv'}^1_0}\left(iv'^2_0\right),\ a_{0,2} \leftarrow E_{k\oplus iv'^1_0}\left(\overline{iv'}^2_0\right)$
3: $c_0^1 \leftarrow a_{0,1} \oplus \overline{iv'}^1_0,\ c_0^2 \leftarrow a_{0,2} \oplus iv'^1_0$
4: $iv_0^1,\ iv_0^2 \leftarrow F^{\text{prng}}(a_{0,1} \oplus a_{0,2})$

Algorithm 10 Encryption module under Serial-AE: Parallel-T.G

Call PH-1
Encrypt M
Partitioning: $m_i^j \in M$ s. t. $(m_1^1, m_1^2), .., \left(m_l^1, m_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $\rho_{i,1} \leftarrow iv_{i-1}^1 \oplus c_{i-1}^1,\ \rho_{i,2} \leftarrow iv_{i-1}^2 \oplus c_{i-1}^2$
    $a_{i,1} \leftarrow E_{k \oplus \overline{\rho_{i,1}}}(\rho_{i,2}),\ a_{i,2} \leftarrow E_{k \oplus \rho_{i,1}}(\overline{\rho_{i,2}})$
    $iv_{i,1},\ iv_{i,2} \leftarrow F^{\text{prng}}(a_{i,1} \oplus a_{i,2})$
    $c_i^1 \leftarrow a_{i,1} \oplus m_i^1,\ c_i^2 \leftarrow a_{i,2} \oplus m_i^2$
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$
$\rho_{t_1} \leftarrow iv_l^1 \oplus c_l^1,\ \rho_{t_2} \leftarrow iv_l^2 \oplus c_l^2$
$t_1 \leftarrow E_{k \oplus \overline{\rho_{t_1}}}(\rho_{t_2}),\ t_2 \leftarrow E_{k \oplus \rho_{t1}}(\overline{\rho_{t_2}})$
$T \leftarrow t_1 \oplus t_2$
Return (C, T)

Algorithm 11 Decryption module under Serial-AE: Parallel-T.G

Call PH-1
Encrypt M
Partitioning: $m_i^j \in M$ s. t. $(m_1^1, m_1^2), .., \left(m_l^1, m_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $\rho_{i,1} \leftarrow iv_{i-1}^1 \oplus c_{i-1}^1,\ \rho_{i,2} \leftarrow iv_{i-1}^2 \oplus c_{i-1}^2$
    $a_{i,1} \leftarrow E_{k \oplus \overline{\rho_{i,1}}}(\rho_{i,2}),\ a_{i,2} \leftarrow E_{k \oplus \rho_{i,1}}(\overline{\rho_{i,2}})$
    $iv_{i,1},\ iv_{i,2} \leftarrow F^{\text{prng}}(a_{i,1} \oplus a_{i,2})$
    $m_i^1 \leftarrow a_{i,1} \oplus c_i^1,\ m_i^2 \leftarrow a_{i,2} \oplus c_i^2$
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$
$\rho_{t_1} \leftarrow iv_l^1 \oplus c_l^1,\ \rho_{t_2} \leftarrow iv_l^2 \oplus c_l^2$
$t_1 \leftarrow E_{k \oplus \overline{\rho_{t_1}}}(\rho_{t_2}),\ t_2 \leftarrow E_{k \oplus \rho_{t1}}(\overline{\rho_{t_2}})$
$T \leftarrow t_1 \oplus t_2$
If T is valid then return M or $\perp$

5.3 Security Proof Sketch: The scheme of Serial-AE

Under this section, we provide the basic security proof sketch of the Serial-AE. At first, we provide privacy security proof sketch. The privacy security proof sketch is similar for both three variants of authentication under the Serial-AE. Later, we provide authenticity security proof sketch. However, it varies because of three different types of authentication. In summary, the first variant of tag generation (Semi-Parallel-T.G) under the scheme of Serial-AE can achieve birthday bound security margin because $2n \rightarrow n$-bit compression function. In addition, it is expected upper authenticity security under the second variant of tag generation (Serial-T.G) due to the property of $3n \rightarrow 2n$-bit cryptographic compression function. The third variant (Parallel-T.G) is most efficient under the scheme of Serial-AE. We provide the authenticity security proof of Semi-Parallel-T.G, and Parallel-T.G. However, authenticity security proof of second variant (Serial-T.G) under the scheme of Serial-AE is informal in this work.

5.3.1 Privacy Security: The Scheme of Serial-AE

Privacy security proof sketch of the scheme of Serial-AE is similar under the authentication mode of Semi-Parallel-T.G, Serial-T.G, and Parallel-T.G. Because, the encryption mode is similar for these three variants of authentication. However, the privacy security depends on the random behaviour of encryption mode. We encrypt message pair by pair in every iteration under the two calls of block-cipher. Hence, it depends on the behaviour of random output properties of block-cipher. If output is random then the scheme of Serial-AE satisfies the privacy security bound. On the contrary, the input characteristics of block-cipher is also important. It should be random also. Actually, our proposed scheme of Serial-AE is based on serial operation. Hence, it is infeasible to change the order of input in the encryption module in respect of the adversary. In addition, we use PRNG function in every iteration of encryption mode for generating fresh and unique input of next cycle.

5.3.2 Authenticity Security: The Scheme of Serial-AE

In this section, we mention the security proof sketch of authenticity. Basically, two properties ensure the authenticity of any authenticated encryption. First one is the random characteristics of input of any cryptographic compression function which is used in authentication of AE. Secondly, the standard security notions (collision resistance and preimage resistance) of cryptographic compression function that is used in authentication.

Semi-Parallel-T.G (First Variant of Authentication: The scheme of Serial-AE)

According to the construction of the scheme of Serial-AE: Semi-Parallel-T.G, we use $(n-1)$ encryption to generate input for cryptographic compression function. Our process is based on semi-parallel mode. In every level, we encrypt pair by pair (see 5.2.1). In principle, we assume that $c \leftarrow \mathcal{E}_{k \oplus a}(b) \neq c' \leftarrow \mathcal{E}_{k \oplus b}(a)$. Hence, it is infeasible to change the order of values in respect of adversary. Hence, our assumption is input of cryptographic compression function is random. Our compression function is based on $2n \rightarrow n$-bit. In addition, our cryptographic compression function is secure in respect of collision resistance

and preimage resistance. Therefore, it is expected to achieve birthday-bound authenticity security margin for Semi-Parallel-T.G under the scheme of Serial-AE.

Serial-T.G (Second Variant of Authentication under the scheme of Serial-AE)

On the contrary, We use $n+1$ block-cipher function under the second variant authentication (Serial-T.G) of the scheme of Serial-AE. In the aspect of adversary, it is infeasible to change the order of input for cryptographic compression function because of serial operation like CBC. Hence, the input of our define cryptographic compression is random. Next, our cryptographic compression function is based on three calls of block-cipher and it takes $3n$-bit input. Under these circumstances, it produces $2n$-bit output. If cryptographic compression function satisfies the standard security notions like collision resistance, and preimage resistance then it can be said that the scheme of authenticated encryption satisfies the authenticity security bound. In addition, it is expected to achieve upper integrity margin because of $3n \rightarrow 2n$-bit compression function.

Parallel-T.G (Third Variant of Authentication under the scheme of Serial-AE)

We use only 2 calls of block-cipher function in third variant of authentication (Parallel-T.G) under the scheme of Serial-AE. In the aspect of adversary, it is infeasible to change the order of input for cryptographic compression function because of serial operation like CBC. Actually, we use each cipher-text value as a feed forward in the next iteration of encryption mode. In addition, the output of each iteration generates the value of pair iv using F^{prng} function. Hence, each new value of iv depends on the value of previous cipher-text. As a result, the input of our define cryptographic compression is random. Next, our cryptographic compression function is based on two calls of block-cipher. If our defined cryptographic compression function satisfies the standard security notions like collision resistance, and preimage resistance then it can be said that the first scheme of authenticated encryption satisfies the authenticity security bound.

5.4 Security Analysis of the scheme of Serial-AE

Privacy Security Notion of the scheme of Serial-AE. We assume adversary $\mathcal{A}$ is unique IV based game and gets the access from $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ (Encryption procedure of proposed scheme). For example, adversary $\mathcal{A}$ asks to the real oracle $\left(\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}\right)$ and random oracle ($). Under the real oracle, i-th query consists of unique IV, and plain-text (M_i). Furthermore, the reply is $(C,T) \leftarrow \mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}(IV,M)$. On the contrary, the adversarial query to random oracle is IV, M, where the feedback is $(C,T) \leftarrow^{\$} \left\{(0,1)^{|M|} \times (0,1)^{|2n|}\right\}$. Hence, the advantage of the privacy security assumption for the adversary $\mathcal{A}$ is bounded as:

$$\mathrm{Adv}^{\mathrm{priv}}_{\mathrm{AE}_T^{\mathrm{FS}}}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}(.,.),\mathrm{ICM}} = 1\right] - \Pr\left[\mathcal{A}^{\$(.,.),\pi,\pi^{-1}} = 1\right],$$

where the initial probability comes from randomness of the key of the proposed scheme. In addition, the later-one comes from the random oracle. Furthermore, adversary is not allowed for duplicate query.

Integrity Security Notion of the Scheme of Serial-AE. The authenticity of the proposed authentication scheme $\mathrm{AE}_T^{\mathrm{FS}}$ is defined as successful generation of a valid tag (IV, C, T) by the adversary $\mathcal{A}$. However, $\mathcal{A}$ is allowed to ask query to the encryption $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ and decryption oracles $\mathcal{D}.\mathrm{AE}_T^{\mathrm{FS}}$. We assume that the following equations of 5.1 and 5.2 are encryption and decryption queries. These queries are executed by the adversary $\mathcal{A}$.

$$(IV_1, M_1), \cdot\cdot, (IV_3, M_3), \cdots, (IV_{q_\mathcal{E}}, M_{q_\mathcal{E}}) \tag{5.1}$$

$$(IV'_1, C'_1), \cdot\cdot, (IV'_3, C'_3), \cdots\cdot, (IV'_{q_D}, C'_{q_D}) \tag{5.2}$$

Therefore, the query contents of $\mathcal{A}$ are $q_\mathcal{E}, q_\mathcal{D}$. Let, there is an experiment $\mathsf{EXP}^{\mathrm{auth}}_{\mathrm{sim}}$, which outputs 1 iff the adversary successfully forges. Hence, the mathematical notion of the authenticity for the adversary $\mathcal{A}$ is:

$$\mathrm{Adv}^{\mathrm{auth}}_{\mathrm{AE}_T^{\mathrm{FS}}}(\mathcal{A}) = \Pr\left[\mathsf{EXP}^{\mathrm{auth}}_{\mathrm{FS}}(\mathcal{A}) = 1\right] \tag{5.3}$$

However, $\mathcal{A}$ forges and returns bit-string (IV'_i, C'_i, T'_i) for decryption query under the certain condition of $(IV'_i, C'_i, T'_i) \neq (IV_j, C_j, T_j) | \, 1 \leq j \leq q_\mathcal{D}$.

5.4.1 Privacy Security Analysis: The Scheme of Serial-AE

Privacy security of the $\mathrm{AE}_T^{\mathrm{FS}}$ is defined as the probability of distinguish between ciphertext and random string by adversary $\mathcal{A}$, where $\mathcal{A}$ is based on unique IV and FS directs the scheme of Serial-AE. We define certain simulators for finding the advantage of adversary $\mathcal{A}$. Initially, we find the probability of collision for each simulation based game. Then we find the difference between two consecutive simulators. Finally, we take the union bound of all the differentiate values. Generally, we follow the proof technique of [21, 47, 48]. In addition, we customized the privacy security proof technique according to our construction's definition, operation, and nature. Thereafter, we briefly presented the privacy security proof of the proposed first construction in this subsection.

Theorem 5.1. Let $\mathrm{AE}_T^{\mathrm{FS}}$ be the proposed authenticated encryption (Serial-AE), where $n \geq 1$ and $\mathcal{E}.\mathrm{AE}_T^{\mathrm{sim}}$ be encryption algorithm. An adversary $\mathcal{A}$ is allowed to access

random oracle (π/π^{-1}). However, adversary $\mathcal{A}$ can query upto q. The advantage of $\mathcal{A}$ is to distinguish $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ from random oracle, that is noted as:

$$\mathrm{Adv}^{\mathrm{priv}}_{\mathrm{AE}_T^{\mathrm{FS}}}(\mathcal{A}) = \Pr\left[A^{\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}(.,.),\mathrm{ICM}} = 1\right] - \Pr\left[\mathcal{A}^{\$(.,.),\pi/\pi^{-1}} = 1\right] \leq \sigma/2^{n-1} + \sigma^2/2^{2n} + q/2^n$$

KEYPOINTS: σ is the maximum number of queries including q and ideal permutation.

Proof. We use certain simulators for finding the advantage of adversary. In the beginning, simulator $\mathcal{S}_1$ simulates the $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ authenticated encryption under ideal cipher model. Furthermore, the simulator $\mathcal{S}_6$ simulates the random oracle. The rest of the simulators are $\mathcal{S}_2$, $\mathcal{S}_3$, $\mathcal{S}_4$, and $\mathcal{S}_5$. Adversary $\mathcal{A}$ tries to distinguish the consecutive simulators using $\mathcal{COLL}$ event. The advantage of distinguishing two consecutive simulators is evaluated as the probability of the $\mathcal{COLL}$ event. Therefore, we get the advantage of adversary $\mathcal{A}$ for finding the difference between $\mathrm{AE}_T^{\mathrm{FS}}$ and random oracle by adding the probability of all the $\mathcal{COLL}$ events. Additionally, if collisions occur then new value will be taken from uniform distribution of random oracle, Furthermore, K will be chosen in each simulators from the block-cipher's random key set. In addition, IV will be generated in each iteration by F^{prng} for randomness.

First Simulator $(\mathcal{S}_1)$. The proposed authenticated encryption scheme of $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ is simulated by $\mathcal{S}_1$. The queries of $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ are executed under the ideal cipher oracle model for each internal operation. Hence, the each output is random and unique, which is used for next input. Therefore, the $\mathcal{S}_1$ and $\mathrm{AE}_T^{\mathrm{FS}}$ seems to be identical. Hence,

$$\Pr\left[\mathcal{A}^{\mathcal{S}_1} = 1\right] = \Pr\left[\mathcal{A}^{\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}}_{\mathrm{ICM}} = 1\right] \tag{5.4}$$

Second Simulator $(\mathcal{S}_2)$. Under the second simulator $\mathcal{S}_2$, the output of ICM simulate the random permutation based input and output. However, the output of F^{prng} is based on random function, which can't assure the uniqueness for each iteration. If any collision occurs, then a collision event is defined. Therefore, the distinguish between these two simulators is the probability of the $\mathcal{COLL}$ event.

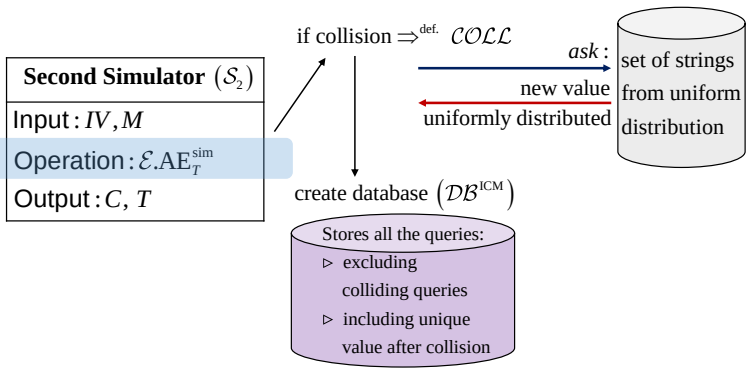


Figure 5.5: Simulator $\mathcal{S}_2$

Additionally, we define a database $\mathcal{DB}^{\mathrm{ICM}}$, which stores all the queries (Figure 5.5). In addition, the queries are queried by σ times under the $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$. Therefore,

$$\Pr\left[\mathcal{A}^{\mathcal{S}_2} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{S}_1} = 1\right] = \Pr\left[\mathcal{COLL}\right] \leq \sigma/2^n \tag{5.5}$$

Third Simulator $(\mathcal{S}_3)$. The $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ is being simulated by the simulator $\mathcal{S}_3$ and executes query through random function. Moreover, $\mathcal{S}_3$ renew and synchronize the database of

$\mathcal{DB}^{\text{ICM}}$ from the last phase of oracle by deleting the collide values. Therefore, the $\mathcal{S}_3$ and $\mathcal{S}_2$ are indistinguishable in the perspective of the adversary $\mathcal{A}$.

$$\Pr\left[\mathcal{A}^{\mathcal{S}_2} = 1\right] = \Pr\left[\mathcal{A}^{\mathcal{S}_3} = 1\right] \tag{5.6}$$

Fourth Simulator $(\mathcal{S}_4)$. Under this simulator we check the randomness of internal iteration of the AE_T^{FS}. There are two issues such as cipher-text and IV. These two should be random and unique for each iteration of authenticated encryption. According to our first construction $\left(\mathcal{E}.\text{AE}_T^{\text{FS}}\right)$, the block-cipher output should be random and unique because of PRP characteristic. The key is being chosen from the block-cipher $\mathcal{BLOCK}(\mathcal{N}, \mathcal{K})$. Therefore, the vital fact is message (m_i). Though, the adversary can control the message (m_i), but it can not control the value of $x_{i,1}$ and $x_{i,2}$ because of the block-cipher output characteristics. Hence, the XOR of $x_{i,1}$, $x_{i,2}$ and m_i^1, m_i^2 produce the random cipher-text (c_i^1, c_i^2). The rest of the issue is unique IV. According to our construction, IV is generated from the output of the block-ciphers. We generate unique and random IV from the function of F^{prng}. However, collision can be occurred under the four scenarios such as:

- collision for output (internal) of block-cipher
- collision for output (external) of block-cipher
- collision for input of F^{prng}
- key attack

Generally, the recovery system of each scenario is similar and simple. If collision occurs under a scenario then unique and random value will be inherited from the uniformly distributed set.

▷ Internal collision of block-cipher: Under this state, $x_{i,1}$ can be collide with $x_{i,2}$ under any iteration of i. In that case, the event $\mathcal{COLL}^{\text{internal}}$ is called. Therefore,

$$\begin{aligned}
&\Pr\left[\mathcal{COLL}^{\text{internal}}\right] = \Pr\left[\mathcal{COLL}_1 \vee \mathcal{COLL}_2 \vee ...\mathcal{COLL}_\sigma\right] \\
&\leq \Pr\left[\mathcal{COLL}_1\right] + \Pr\left[\mathcal{COLL}_2\right] + ... \Pr\left[\mathcal{COLL}_\sigma\right] \\
&\leq \sigma \cdot (1/2^n)
\end{aligned} \tag{5.7}$$

▷ External collision of block-cipher: Under this state, $x_{i,1}$, $x_{i,2}$ and $x_{j,1}$, $x_{j,2}$ can be collide for different two iterations of i, j where $i < j$. Therefore,

$$\begin{aligned}
&\Pr\left[\mathcal{COLL}^{\text{external}}\right] = \Pr\left[\mathcal{COLL}_1 \vee \mathcal{COLL}_2 \vee ... \vee \mathcal{COLL}_\sigma\right] \\
&\leq \Pr\left[\mathcal{COLL}_1\right] + \Pr\left[\mathcal{COLL}_2\right] + ... + \Pr\left[\mathcal{COLL}_\sigma\right] \\
&\leq \sigma\left(\sigma - 1\right)/\left(2^n - 1\right)^2
\end{aligned} \tag{5.8}$$

▷ collision for input of F^{prng}: According to the construction of the proposed scheme, the i-th iteration's input of F^{prng} depends on the $i-1$-th output of F^{prng}. Thus, there is a chance to make a collision of i-th iteration's input or $i-1$-th iteration's output. Hence,

$$\begin{aligned}
&\Pr\left[\mathcal{COLL}^{F^{prng}}\right] = \Pr\left[\mathcal{COLL}_1 \vee \mathcal{COLL}_2 \vee ... \vee \mathcal{COLL}_\sigma\right] \\
&\leq \Pr\left[\mathcal{COLL}_1\right] + \Pr\left[\mathcal{COLL}_1\right] + ... + \Pr\left[\mathcal{COLL}_\sigma\right] \\
&\leq \sigma/2^n
\end{aligned} \tag{5.9}$$

▷ Key attack: Under this state, the key can be attacked, where the probability is:

$$\Pr\left[\text{key-attack}\right] \leq q/2^n \tag{5.10}$$

Therefore, we take the union bound of 5.7, 5.8, 5.9, 5.10. In addition, we stores all the queries into $\mathcal{DB}^{\pi/\pi^{-1}}$ except the colliding queries.

$$\begin{aligned}
&\Pr\left[\mathcal{A}^{S_4} = 1\right] - \Pr\left[\mathcal{A}^{S_3} = 1\right] = \\
&\Pr\left[\mathcal{COLL}^{\text{internal}} + \mathcal{COLL}^{\text{external}} + \mathcal{COLL}^{F_{U_{iv}}} + \text{key-attack}\right] \leq \\
&\Pr\left[\mathcal{COLL}^{\text{internal}}\right] + \Pr\left[\mathcal{COLL}^{\text{external}}\right] + \Pr\left[\mathcal{COLL}^{F_{U_{iv}}}\right] + \Pr\left[\text{key-attack}\right] \\
&\leq \sigma/2^{n-1} + \sigma^2/2^{2n} + 1/2^n
\end{aligned} \tag{5.11}$$

Fifth Simulator $(\mathcal{S}_5)$. The $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ runs under the ideal cipher model. In addition, it deletes the collide values under the last phase of oracle and takes the fresh and new value from the set of strings of uniform distribution. Therefore, the simulator $\mathcal{S}_5$ and $\mathcal{S}_4$ are indistinguishable in the aspect of adversary $\mathcal{A}$.

$$\Pr\left[\mathcal{A}^{S_5} = 1\right] = \Pr\left[\mathcal{A}^{S_4} = 1\right] \tag{5.12}$$

Sixth Simulator $(\mathcal{S}_6)$. In this module, we synchronize the database of $\mathcal{DB}^{\mathrm{ICM}}$ across the last phase of oracle for $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$. In addition, the simulator $\mathcal{S}_6$ perfectly simulates the random oracle. On the contrary, the simulator $\mathcal{S}_5$ inherits the proposed scheme $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ under the ideal cipher oracle model, where all current values are uniformly distributed. Because, all the collide values are deleted already under the simulator 1 to 4. Thus, $\mathcal{S}_6$ and $\mathcal{S}_5$ are indistinguishable in favour of adversary $\mathcal{A}$. Hence,

$$\Pr\left[\mathcal{A}^{S_6} = 1\right] = \Pr\left[\mathcal{A}^{S_5} = 1\right] = \Pr\left[\mathcal{A}^{\$}_{\pi,\pi^{-1}}\right] \tag{5.13}$$

Finally, **Theorem 5.1** is satisfied under the union bound of 5.5, and 5.11.

5.4.2 Authenticity Security Analysis: Serial-AE: Semi-Parallel-T.G

The authenticity of $\mathrm{AE}_T^{\mathrm{FS}}$ scheme is defined to successful inject of false data $(IV'_{i'}, C'_{i'})$ instead of valid data (IV_i, C_i) through an adversary and gets success for valid tag. The $\mathrm{AE}_T^{\mathrm{FS}}$ has encryption and decryption oracle respectively $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ and $\mathcal{D}.\mathrm{AE}_T^{\mathrm{FS}}$. Additionally, it has access to the random oracle (π/π^{-1}). We assume that encryption and decryption queries look IV, M and IV, C, T. Therefore, the adversarial query contents are $q_{\mathcal{E}}, q_{\mathcal{D}}$. Let, there is an experiment $\mathsf{EXP}_{\mathrm{FS}}^{\mathrm{auth}}$, which outputs 1 iff the adversary successfully forges when $(IV', C', T') \neq (IV', C', T')$. Hence, the mathematical notion of the authenticity for the adversary $\mathcal{A}$ is:

$$\mathrm{Adv}^{\mathrm{auth}}_{\mathrm{AE}_T^{\mathrm{FS}}}(\mathcal{A}) = \Pr\left[\mathsf{EXP}^{\mathrm{auth}}_{\mathrm{FS}}(\mathcal{A}) = 1\right] \tag{5.14}$$

Briefly, $\mathcal{A}$ forges and returns bit-string (IV'_i, C'_i, T'_i) using encryption and decryption query under the certain condition of $(IV, C, T) \neq (IV', C', T')$.

Theorem 5.2. Let $\mathrm{AE}_T^{\mathrm{FS}}$ be the proposed authenticated encryption, where $\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}$ and $\mathcal{D}.\mathrm{AE}_T^{\mathrm{FS}}$ be encryption and decryption algorithm respectively. The adversary $\mathcal{A}$ is

allowed to access the oracle of $\mathrm{AE}_T^{\mathrm{FS}}$ $\left(\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}}, \mathcal{D}.\mathrm{AE}_T^{\mathrm{FS}}\right)$ and random oracle. The advantage of $\mathcal{A}$ is noted as the success probability of injecting false data instead of valid data through the defined experiment EXP. Hence, the advantage is bounded as:

$$\Pr\left[\mathrm{Adv}^{\mathrm{auth}}_{\mathrm{AE}_T^{\mathrm{FS}}(\mathcal{E}.\mathrm{AE}_T^{\mathrm{FS}},\mathcal{D}.\mathrm{AE}_T^{\mathrm{FS}}),\mathrm{ICM}}(\mathcal{A}) = 1\right] \leq \mathrm{Adv}^{\mathrm{priv}}_{\mathrm{AE}_T^{\mathrm{FS}}} + \sigma^2/2^{2n} + q/2^n$$

Proof. Let adversary $\mathcal{A}$ has access both the encryption and decryption oracle. Under the decryption oracle, $\mathcal{A}$ is not allowed to make an encryption query if it is already existed. The advantage of adversary is evaluated as the success probability of $(IV, C, T) \neq (IV', C', T')$, when it gets valid T. For simplicity, we assume two games of $\mathcal{D}$.GAME-1 and $\mathcal{D}$.GAME-2 for finding the authenticity advantage of adversary $\mathcal{A}$. The first game directs the collision of cipher-text and randomness of cipher-text (Fig. 5.6). The collision of tag explains under the game of $\mathcal{D}$.GAME-2.

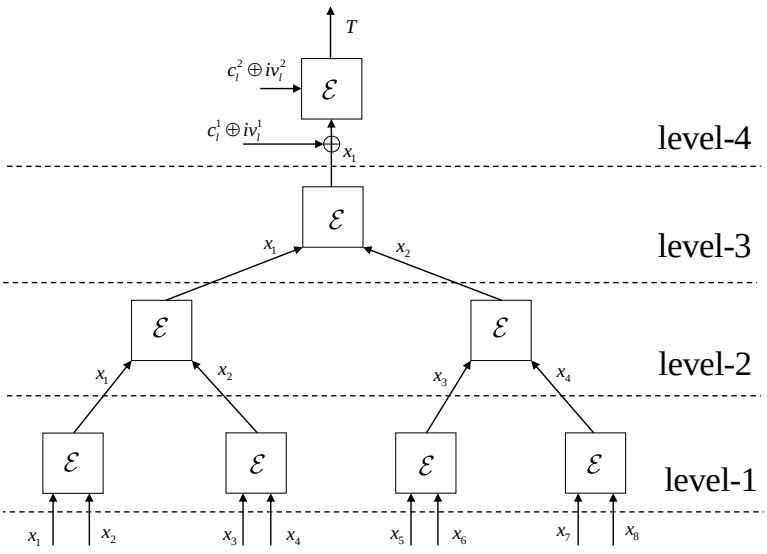


Figure 5.6: Tag Generation Process: Semi-Parallel-T.G under Serial-AE

$\mathcal{D}$.GAME-1. At first, we explain the scenario of the authentication phase. The tag T depends on the input that is generated in the top level of authentication process (Fig. 5.5). If this input is random then the output T will be secure also. Under these circumstances, we will explain form the bottom level of process of authentication. The set of values $(x_{i,1}, x_{i,2}, x_{i+1,1}, x_{i+1,2}, ..., x_{l,2}, x_{l,1}) \in x$ are generated in encryption/decryption module of the first scheme (Fig. 5.2, Algorithm 8, 9). Then, we re-indexing z like $(x_1, x_2, ..., x_l, x_{l+1}, x_{l'}) \in x$. For tag generation, we will use these values. These values are random because of the random properties of block-cipher. We process these values from the bottom of authentication process. We re-encrypt pair by pair like $a \leftarrow E_{k \oplus x_1}(x_2)$, $b \leftarrow E_{k \oplus x_3}(x_4)$, $c \leftarrow E_{k \oplus x_5}(x_6)$, ..., $L1 \leftarrow E_{k \oplus x_{l-1}}(x_l)$.

The first point is weather the value of a, b, c, . . . , $L1$ are random or not. According to the property of block-cipher these values are random and unique. The second point is if order of re-encrypt changes then is there any chances to create similar final input for the tag generation. In this stage, we explain it. At first, we take the bottom level for example. In the bottom level, re-encryptions are $a \leftarrow E_{k \oplus x_1}(x_2)$, $b \leftarrow E_{k \oplus x_3}(x_4)$, $c \leftarrow E_{k \oplus x_5}(x_6)$, ..., $L1 \leftarrow E_{k \oplus x_{l-1}}(x_l)$. Interestingly, $\omega \leftarrow E_{k \oplus x_1}(x_2) \neq \omega' \leftarrow E_{k \oplus x_2}(x_1)$ is true under the block-cipher property. Under these circumstances, our assumption is re-ordering is

also infeasible in respect of adversary. Adversary can only randomly assume the output of each re-encryption. Third point is even the order of cipher-text is changed then similar output can not be generated. Because by definition the building principle of x depends on the previous values of cipher-text like is $x_i^j \leftarrow a_{i,j} \oplus \mu_i^j$ where μ comes from the xor of iv_{i-1} and last generated cipher-text c_{i-1}.

So, we will find out the collision probability in every level under the first game. For example, in the iteration of i and j: for each level, do $x_i \leftarrow E_{k \oplus x_{2i-1}}(x_{2i})\,[\text{for}, i \in \{1, 2, .., |x|/2\}]$ and for each level, do $x_j \leftarrow E_{k \oplus x_{2j-1}}(x_{2j})\,[\text{for}, j \in \{1, 2, .., |x|/2\}]$ such that,

$$\left[\begin{array}{l} \text{for each level,} \\ \text{do } x_i \leftarrow E_{k \oplus x_{2i-1}}(x_{2i})\,; \\ [\text{for}, i \in \{1, 2, .., |x|/2\}] \end{array}\right] = \left[\begin{array}{l} \text{for each level,} \\ \text{do } x_j \leftarrow E_{k \oplus x_{2j-1}}(x_{2j})\,; \\ [\text{for}, j \in \{1, 2, .., |x|/2\}] \end{array}\right]$$

We define an event $\mathcal{COLL}$ to find out the probability of making collision under the iteration of i and j. Hence, the probability of collision under the event $\mathcal{COLL}$ is:

$$\Pr\left[\mathcal{COLL}^{D.Game1}\right] \leq \sum_{i=1}^{\sigma} \frac{2(i-1)}{(2^n - i)} \leq \frac{\sigma(\sigma - 1)}{(2^n - \sigma)^2} \tag{5.15}$$

$\mathcal{D}$.GAME-2. The generated tag can be collide under any iteration of decryption. We assume the generated tag is T for the iteration i. Furthermore, T' is under i'-th iteration. Thus, the collision probability is as follows:

$$\Pr\left[COL\mathcal{L}^{\mathcal{D}.\text{Game-2}}\right] \leq q/2^n \tag{5.16}$$

Finally, **Theorem 5.2** is satisfied under the union bound of 5.15, 5.16 including privacy security advantage.

5.4.3 Authenticity Security Analysis: Serial-AE: Serial-T.G

In this section, we show the brief proof sketch of authenticity (Serial-T.G) under the scheme of Serial-AE (Fig. 5.7). We make two sections here. In the first section, we re-encrypt cipher-text for creating the input of tag generation. In the second section, we use the last generated output as an input of cryptographic compression function and generates the Tag. At first, we will show that the input of cryptographic compression function is random. Furthermore, we will show that the cryptographic compression function of tag generation satisfies the security notion of collision and preimage resistance.

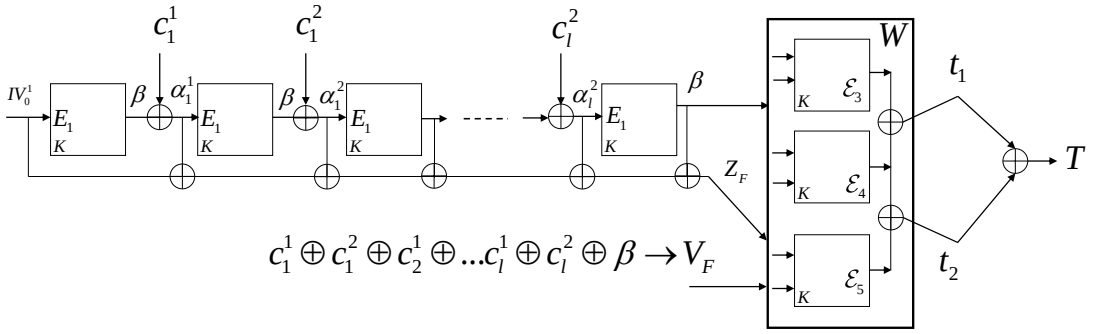


Figure 5.7: Second Variant Authentication under the First Scheme

Under the first section, there are input of β and Z_F. These input are generated through a series of re-encryption process. Let there are iteration of i, and s where β, are Z_F are defined as:

$$\left\{\begin{array}{l} i = 1 \text{ to } l \\ \left\{\begin{array}{l} j = 1 \text{ to } 2 \\ \left\{\begin{array}{l} \alpha_i^j \leftarrow \beta \oplus c_i^j, \\ \beta \leftarrow E_k\left(\beta \oplus c_i^j\right) \end{array}\right\} \end{array}\right\} \end{array}\right\}$$

$$\left\{\begin{array}{l} x = 1 \text{ to } i-1 \\ \left\{\begin{array}{l} y = 1 \text{ to} 2 \\ \left\{\begin{array}{l} \alpha_x^y \leftarrow \beta \oplus c_x^y, \\ \beta \leftarrow E_k\left(\beta \oplus c_x^y\right) \end{array}\right\} \end{array}\right\} \end{array}\right\}$$

Under this circumstance, the collision scenario for any two different i-th and j-th query of β and Z_F are:

$$\left[\begin{array}{l} i = 1 \text{ to } l \\ \left\{\begin{array}{l} j = 1 \text{ to } 2 \\ \left\{\begin{array}{l} \alpha_i^j \leftarrow \beta \oplus c_i^j, \\ \beta \leftarrow E_k\left(\beta \oplus c_i^j\right) \end{array}\right\} \end{array}\right\} \end{array}\right] = \left[\begin{array}{l} x = 1 \text{ to } i-1 \\ \left\{\begin{array}{l} y = 1 \text{ to } 2 \\ \left\{\begin{array}{l} \alpha_x^y \leftarrow \beta \oplus c_x^y, \\ \beta \leftarrow E_k\left(\beta \oplus c_x^y\right) \end{array}\right\} \end{array}\right\} \end{array}\right]$$

We define an event $\mathcal{COLL}$ to find out the probability of making collision under the iteration of i and x. Hence, the probability of collision under the event $\mathcal{COLL}$ is:

$$\Pr\left[\mathcal{COLL}\right] \leq \sum_{i=1}^{q} \frac{2\left(i-1\right)}{\left(2^n - \left(2i-2\right)\right)\left(2^n - \left(2i-1\right)\right)} \leq \frac{q^2 - q}{\left(2^n - 2q\right)^2}$$

In addition, the third input is the xor of all cipher-text and β. Even adversary can change the order of cipher-text but the value of β will be changed in that case. Hence, the third input (V_F) is also random. So, the collision probability for the third input is $1/2^n$. Under these circumstances, total collision probability under the first section is $\frac{q^2-q}{(2^n-2q)^2} + \frac{1}{2^n}$.

In the next phase, we will show that W is the secure cryptographic compression function under collision and preimage resistance. It actually works as one way hash, where component functions are three block-ciphers. It takes $3n$-bit input and generates $2n$-bit output. Now we will show that the input of cryptographic compression function or tag generation produces random output. In addition, the output of cryptographic compression function satisfies the preimage and collision resistance security notions. According to the construction of the second variant authentication of the first scheme, the preimage resistance scenario is:
Let adversary $\mathcal{A}$ is allowed to randomly choose a pair of value such as u_1 and u_2. Hence, it is needed to find out the collision probability of $W(\phi) \leftarrow E_{k\oplus\beta}(V_F), E_{k\oplus V_F}(Z_F), E_{k\oplus Z_F}(\beta)$ where $\phi \ni \beta, V_F, Z_F$ and W is a cryptographic hash that is made by the component function of three calls of block-cipher.

Under these circumstances, u_1, u_2 be the two points where adversary tries to find inversion. In addition, adversary is tried to find out β, V_F, Z_F such that $F(\beta, V_F, Z_F) = (u_1, u_2)$. Under the ideal cipher model, adaptive query is allowed until the domain size of $N/2$. Then, the rest of the queries are given as free to the adversary, where adversary can make query in non-adaptive fashion. Furthermore, the adjacent query triplet is defined as $(k\oplus\beta, V_F, \gamma_1), (k\oplus V_F, Z_F, \gamma_2), (k\oplus Z_F, \beta, \gamma_3)$. Therefore, we need to find out the probability of collision under the domain of adaptive query (NormalQueryWin) and non-adaptive query (SuperQueryWin). Under the NormalQueryWin, the query of response of $(k\oplus\beta, V_F, \gamma_1)$ query can come from the set size at most $N/2-2$. Hence, the probability of collision is approximately $2/N$. Because of adjacent query triplet the total probability is $3\times 2/N$. Therefore, the total probability under the NormalQuerywin is $6/N^2$. For the SuperQueryWin probability is $6/N^2$. Hence, the total preimage security is bounded by $12/N^2$.

In the second section, we point out the collision resistance of the second variant of authentication under the second scheme. Generally, collision resistance is defined as to find x and x' is infeasible such that $F(x) = F(x')$ when $x \neq x'$. According to the construction of the second variant of the first scheme, adversary tries to find $(k\oplus\beta, V_F), (k\oplus V_F, Z_F), (k\oplus Z_F, \beta)$ and $(k\oplus\beta', V'_F), (k\oplus V'_F, Z'_F), (k\oplus Z'_F, \beta')$ s. t.:

$$\begin{bmatrix} E(k\oplus\beta, V_F), E(k\oplus V_F, Z_F), \\ E(k\oplus Z_F, \beta) \end{bmatrix} = \begin{bmatrix} E(k\oplus\beta', V'_F), E(k\oplus V'_F, Z'_F), \\ E(k\oplus Z'_F, \beta') \end{bmatrix}$$

Under these conditions, the collision probability is $\Pr[COLL] \leq \frac{q^2-q}{(2^n-3q)^2}$. Hence, the cryptographic compression function is secure under collision resistance.

5.4.4 Authenticity Security Analysis: Serial-AE: Parallel-T.G

We provide a brief proof sketch of authenticity of the scheme of Serial-AE: Parallel-T.G. In our proof sketch, we make two categories. Under the first category, we show that the input of tag generation process is random. In addition, we show that the module of tag generation process satisfies collision and preimage resistance security bound.

Under the first category, at first we need to explain how the input of tag generation is created. Actually, we use certain values that are generated in the encryption mode. According to the Figure 5.4, we explain the following example. Let there are couple of values iv_i^1 and iv_i^2 under iteration of i. Moreover, there are pair of created cipher-text like c_i^1 and c_i^2 under this iteration of i. Interestingly, the values of iv_i^1 and iv_i^2 depend on the previously generated cipher-text c_{i-1}^1 and c_{i-1}^2. In this way, we can find that the pair of values iv_l^1 and iv_l^2 depend the previously generated cipher-text of $c_1^1, c_1^2, c_2^1, c_2^2, ..., c_{l-1}^1, c_{l-1}^2$ due to block-cipher property and serial operation under encryption mode. Therefore, we can claim that to change the order of cipher-text in respect of adversary is infeasible. Hence, we use iv_l^1 and iv_l^2 values for creating tag.

Under the second category, we show how we create the tag generation module. We use two calls of block-cipher. In addition, our tag generation module is based on block-cipher cryptographic compression function. Hence, if our tag generation module is secure under collision resistance and preimage resistance security bound then we can claim that our proposed scheme (Serial-AE: Parallel-T.G) is secure for authentication also. Our define block-cipher in the tag generation module is E. The working principle of E is $E_{k\oplus a}(b) \rightarrow c$. In addition, our basic assumption is $c \leftarrow E_{k\oplus a}(b) \neq c' \leftarrow E_{k\oplus b}(a)$. In the worst case, the adversary can change the order of c_l^1 and c_l^2. Because, iv_l^1 and iv_l^2 depend on the values upto c_{l-1}^1 and c_{l-1}^2. Interestingly, iv_l^1 and iv_l^2 do not depend on the values of c_l^1 and c_l^2. To overcome this problem, we xor $c_l^1 \oplus iv_l^1$ and $c_l^2 \oplus iv_l^2$. These xor values are used as input of our tag generation module. Moreover, our define block-cipher E satisfy the property of $c \leftarrow E_{k\oplus a}(b) \neq c' \leftarrow E_{k\oplus b}(a)$. Under these circumstances, the final input of our tag generation module are $\rho_{t_1} \leftarrow c_l^1 \oplus iv_l^1$ and $\rho_{t_2} \leftarrow c_l^2 \oplus iv_l^2$. Hence, intermediate of tag t_1 and t_2 are defined as $t_1 \leftarrow E_{k\oplus\overline{\rho_{t_1}}}(\rho_{t_2})$ and $t_2 \leftarrow E_{k\oplus\rho_{t_1}}(\overline{\rho_{t_2}})$. Therefore, adversary even change the order of c_l^1 and c_l^2 but it can not produce similar intermediate tag.

5.5 Nonce Respect Authenticated Encryption

The second scheme is based on nonce respect and operates in parallel. Hence we call this scheme as Parallel-AE. Under the scheme of Parallel-AE, we propose two different types of tag generation (authentication) such as Semi-parallel tag generation (Semi-Parallel-T.G), and Serial tag generation (Serial-T.G). Therefore, in combine these are as Parallel-AE: Semi-Parallel-T.G, and Parallel-AE: Serial-T.G. It can support only fixed size of n-bit associated data. Hence, it is suitable for IoT application in certain cases. The first type of authentication or tag generation needs total $(n-1)+1$ calling function. In addition, second variant is based on $3n \rightarrow 2n$-bit compression function, where we use double keyed function f and it needs total $n+2$ encryption functions.

5.6 Preliminaries for the scheme of Parallel-AE

M is defined as message set where $m_i^j \in M$ such that $(m_1^1, m_1^2), ., \left(m_l^1, m_l^2\right)$, where $j \in \{1,2\}, i \leq l$. In addition, C is defined as set of cipher-text where $c_i^j \in C$ such that $(c_1^1, c_1^2), ., \left(c_l^1, c_l^2\right)$, where $j \in \{1,2\}, i \leq l$. Moreover, T is defined as final Tag. In our proposal, we use function F_k. The operation like F_k is $F_k : c \leftarrow F_{k \oplus a}(b)$. In addition, our assumption is $c \leftarrow F_{k \oplus a}(b) \neq c' \leftarrow F_{k \oplus b}(a)$.

In our proposal, we use Galois field operation [19, 21, 47, 77]. An n-bit string X may be viewed as an element of $\mathrm{GF}(2^n)$ by taking X as a coefficient vector of a polynomial in $\mathrm{GF}(2^n)$. Following [19, 21, 47], we write $2X$ to denote the multiplication of 2 and X over $\mathrm{GF}(2^n)$, where 2 denotes the generator of the field $\mathrm{GF}(2^n)$, by seeing 2 as x in the polynomial representation. This operation is called doubling. Similarly we write $3X$ (where the corresponding polynomial is $x+1$) and 2^2X to denote as $2X \oplus X$. The doubling can be efficiently computed by one-bit shift with conditional XOR of a constant, and other constant multiplications can be done by combining doubling and XOR, as shown above. Throughout the paper we assume $n = 128$ and the corresponding field $\mathrm{GF}(2^n)$ is defined over the polynomial $x^{128} + x^7 + x^2 + x^1 + 1$, which is lexicographically-first primitive polynomial and is quite popular for doubling-based tweaks [19, 21, 47, 77]. For example, we create δ in the Algorithm 1 of initialization phase. Hence, we define $LSH_i(\delta) = \delta$, $LSH_{i+1}(\delta) = 2\delta$, ... , and $LSH_l(\delta) = 2^{l-1}\delta$. Moreover, $LSH_i(\delta') = 3\delta$, $LSH_{i+1}(\delta') = 2 \cdot 3\delta$, ... , and $LSH_l(\delta') = 2^{l-1} \cdot 3\delta$.

5.6.1 Proposed Scheme of Parallel-AE: Semi-Parallel-T.G

In this section, we propose the first variant of tag generation under the second scheme of Parallel-AE. This variant is based on Semi-Parallel mode. Hence tag generation process is called as Semi-Parallel-T.G. In combine, the scheme is called as Parallel-AE: Semi-Parallel-T.G. We represent this scheme as $\mathrm{AE}^{\mathrm{SS}}_{T.V1}$ where SS: Second Scheme, T: tag, $V1$: First variant of authentication (Fig. 5.8). In addition, there are three phases under the scheme of Parallel-AE: Semi-Parallel-T.G. The first phase (PH-1) is responsible for initialization. The second phase simulates encryption module of e-$\mathrm{AE}^{\mathrm{SS}}_{T.V1}$ where cipher-text and tag are generated. Furthermore, third phase (PH-3) represents a decryption module $\left(\text{d-}\mathrm{AE}^{\mathrm{SS}}_{T.V1}\right)$ of the scheme of Parallel-AE: Semi-Parallel-T.G. For the explanation of all phases, algorithm 12, 13 and 14 are used. In addition, we define keyed function F_k. Our defined function is operated as $F_k : F_{k \oplus a}(b) \rightarrow c$ under the encryption mode. In principle,

our assumption is $c \leftarrow F_{k\oplus a}(b) \neq c' \leftarrow F_{k\oplus b}(a)$.

Explanation of Authentication Procedure of Semi-Parallel-T.G. There are list of $\left(z_1^1, z_1^2, z_2^1, z_2^2, ..., z_l^1, z_l^2\right) \in z$ that are generated in encryption module of the scheme of Parallel-AE (Fig. 5.9, algorithm 13, 14). Then, we re-index z like $(z_1, z_2, ...z_l, z_{l+1}, z_{l'}) \in z$. For example, we take z like $(z_1^1, z_1^2, z_2^1, z_2^2, ..., z_4^1, z_4^2) \in z$. Then, we re-index z like $(z_1, z_2, ...z_7, z_8) \in z$.

Explanation of "For Each Level do: $z_i \leftarrow F_{k\oplus z_{2i-1}}(z_{2i})$ [for, $i \in \{1, 2, .., |z|/2\}$]" command (Algorithm 13, 14: Line 16):

We encrypt $(z_1, z_2, ..., z_7, z_8) \in z$ pair by pair in each level (Fig. 5.8). In level-1, encrypt operation is performed like $z_1 \leftarrow F_{k\oplus z_1}(z_2)$, $z_2 \leftarrow F_{k\oplus z_3}(z_4)$, $z_3 \leftarrow F_{k\oplus z_5}(z_6)$, $z_4 \leftarrow F_{k\oplus z_7}(z_8)$. In level-2, operation is liked $z_1 \leftarrow F_{k\oplus z_1}(z_2)$, $z_2 \leftarrow F_{k\oplus z_3}(z_4)$. In level-3, encryption is $z_1 \leftarrow F_{k\oplus z_1}(z_2)$. This z_1 is the final input of creating Tag (T). For this, we encrypt z_1 and secret value of $\delta \oplus \eta$ as $F_{k\oplus z_1}(\delta \oplus \eta) \rightarrow T$. In principle, our assumption is $c \leftarrow F_{k\oplus a}(b) \neq c' \leftarrow F_{k\oplus b}(a)$.

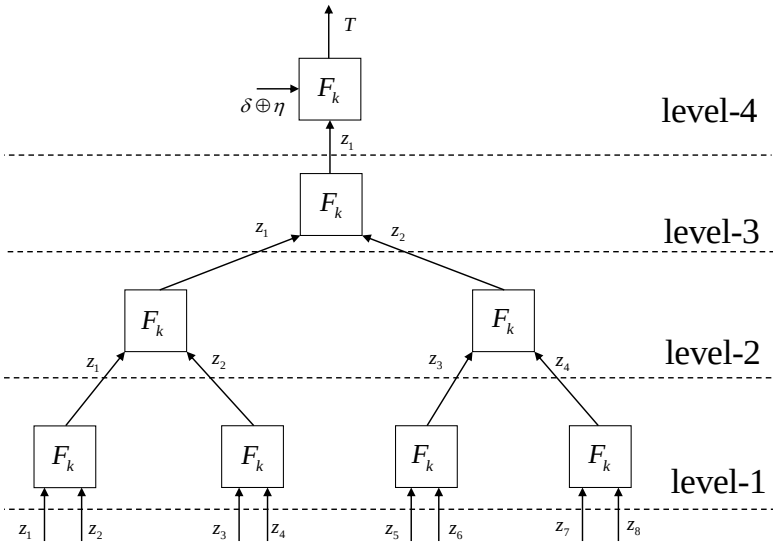


Figure 5.8: Model of Semi-Parallel-T.G (authentication) under the scheme of Parallel-AE

Algorithm 12 Phase-1 (PH-1) of Parallel-AE: Semi-Parallel-T.G

1: Initialization: N_0 and A_0
2: $\delta \leftarrow F_{k\oplus\overline{N_0}}(A_0) \oplus A_0$
3: $\eta \leftarrow F_{k\oplus N_0}\left(\overline{A_0}\right) \oplus A_0$
4: Return: δ, η

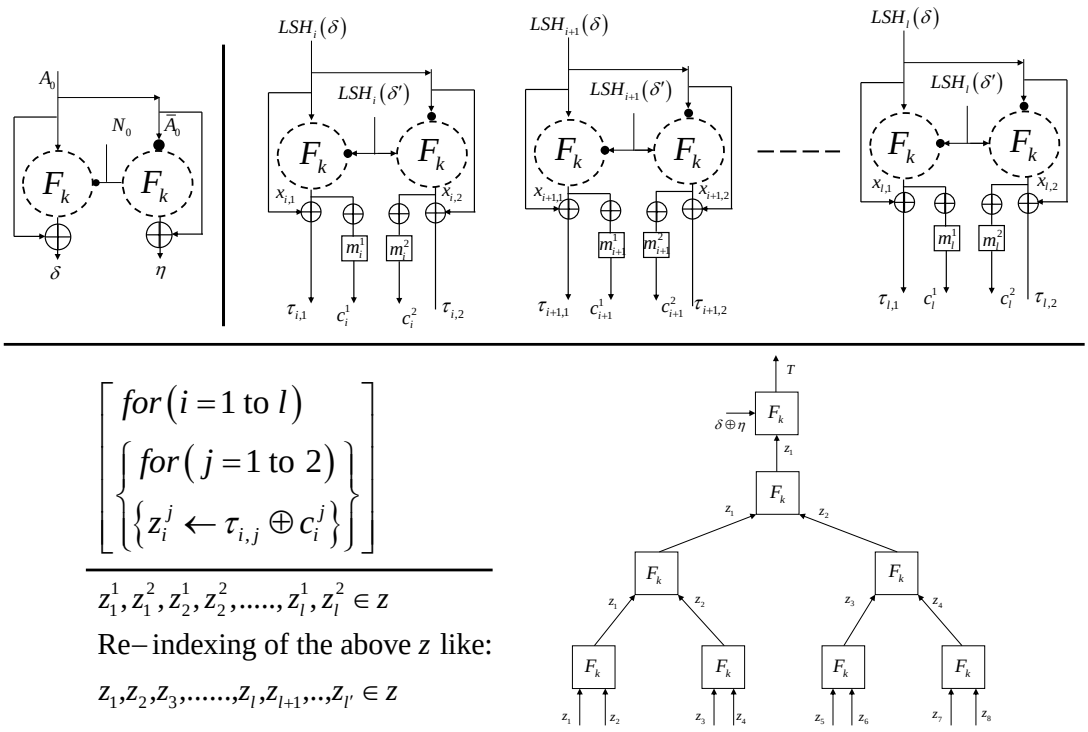


Figure 5.9: Proposed Second Scheme of Parallel-AE: Semi-Parallel-T.G

Algorithm 13 Phase-2 (PH-2) Encryption of Parallel-AE: Semi-Parallel-T.G

Encrypt M and call PH-1
Partitioning: $m_i^j \in M$ such that $(m_1^1, m_1^2), ., \left(m_l^1, m_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $x_{i,1} \leftarrow F_{k \oplus \overline{LSH_i(\delta')}}(LSH_i(\delta))$
    $x_{i,2} \leftarrow F_{k \oplus LSH_i(\delta')}\left(\overline{LSH_i(\delta)}\right)$
    $\tau_{i,1} \leftarrow x_{i,1} \oplus LSH_i(\delta)$, $\tau_{i,2} \leftarrow x_{i,2} \oplus LSH_i(\delta)$
    $c_i^1 \leftarrow x_{i,1} \oplus m_i^1$, $c_i^2 \leftarrow x_{i,2} \oplus m_i^2$
end for
for $i = 1$ to l **do**
    for $j = 1$ to 2 **do**
        $z_i^j \leftarrow \tau_{i,j} \oplus c_i^j$
    end for
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$, $z \leftarrow \left(z_1^1, z_1^2, ..., z_l^1, z_l^2\right)$
re-indexing of z like $(z_1, z_2, ..., z_l, z_{l+1}, z_{l'})$
for Each Level **do** $z_i \leftarrow F_{k \oplus z_{2i-1}}(z_{2i})$ [for, $i \in \{1, 2, .., |z|/2\}$]
end for
$T \leftarrow F_{k \oplus z_1}(\delta \oplus \eta)$
Return (C, T)

Algorithm 14 Phase-2 (PH-2) Decryption of Parallel-AE: Semi-Parallel-T.G

Decrypt C and call PH-1
Partitioning: $c_i^j \in C$ such that $(c_1^1, c_1^2), ., \left(c_l^1, c_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $x_{i,1} \leftarrow F_{k \oplus \overline{LSH_i(\delta')}}(LSH_i(\delta))$
    $x_{i,2} \leftarrow F_{k \oplus LSH_i(\delta')}\left(\overline{LSH_i(\delta)}\right)$
    $\tau_{i,1} \leftarrow x_{i,1} \oplus LSH_i(\delta)$, $\tau_{i,2} \leftarrow x_{i,2} \oplus LSH_i(\delta)$
    $m_i^1 \leftarrow x_{i,1} \oplus c_i^1$, $m_i^2 \leftarrow x_{i,2} \oplus c_i^2$
end for
for $i = 1$ to l **do**
    for $j = 1$ to 2 **do**
        $z_i^j \leftarrow \tau_{i,j} \oplus c_i^j$
    end for
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$, $z \leftarrow \left(z_1^1, z_1^2, ..., z_l^1, z_l^2\right)$
re-indexing of z like $(z_1, z_2, ..., z_l, z_{l+1}, z_{l'})$
for Each Level **do** $z_i \leftarrow F_{k \oplus z_{2i-1}}(z_{2i})$ [for, $i \in \{1, 2, .., |z|/2\}$]
end for
$T \leftarrow F_{k \oplus z_1}(\delta \oplus \eta)$
If T is valid then return M or $\perp$

5.6.2 Proposed Scheme of Parallel-AE: Serial-T.G

We propose second variant of tag generation in this section including encryption mode. This tag generation is based on serial operation. Hence we call this as Serial-T.G. In combine, this scheme is called as Parallel-AE: Serial-T.G. We represent this scheme as $\mathrm{AE}^{\mathrm{SS}}_{T.V2}$ where SS: Second Scheme, T: tag, $V2$: Second variant (Fig. 5.11). The initialization phase is called from the algorithm 12. The encryption module e-$\mathrm{AE}^{\mathrm{SS}}_{T.V2}$ and decryption module $\left(\text{d-}\mathrm{AE}^{\mathrm{SS}}_{T.V2}\right)$ are followed by 15 and 16. The second variant authentication under the second scheme of Parallel-AE is based on $3n \rightarrow 2n$-bit cryptographic compression function. In this second variant of tag generation, we define a function f. This function f is different from the function F_k. The property of f: It has double size of key like $2n$-bit instead of n-bit. We actually, define this f as cryptographic compression function. The operation of f is $f : (\{0,1\}^n \times \{0,1\}^n) \times \{0,1\}^n \rightarrow \{0,1\}^n$. More clearly, it works as $f_{a||b}(c) \rightarrow d$ where $k = a||b$. And our basic assumption is $d \leftarrow f_{a||b}(c) \neq d' \leftarrow f_{b||a}(c)$.

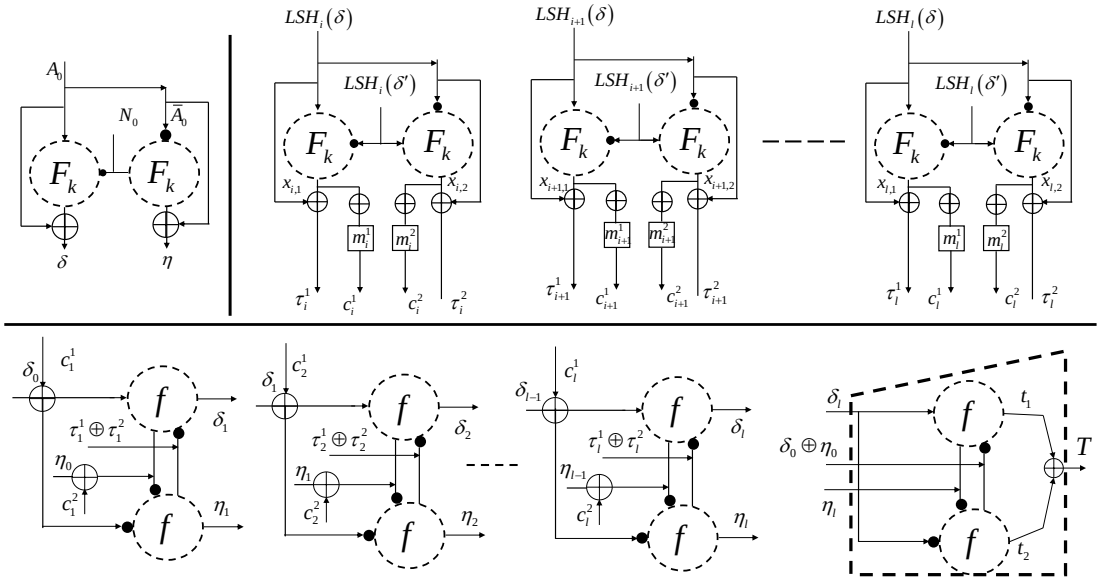


Figure 5.10: Proposed Scheme of Parallel-AE: Serial-T.G

Algorithm 15 Phase-2 (PH-2): Encryption mode of Parallel-AE: Serial-T.G

Encrypt M and call PH-1 (Algorithm15)
Partitioning: $m_i^j \in M$ such that $(m_1^1, m_1^2), .., \left(m_l^1, m_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $x_{i,1} \leftarrow F_{k \oplus \overline{LSH_i(\delta')}}(LSH_i(\delta))$
    $x_{i,2} \leftarrow F_{k \oplus LSH_i(\delta')}\left(\overline{LSH_i(\delta)}\right)$
    $\tau_i^1 \leftarrow x_{i,1} \oplus LSH_i(\delta),\ \tau_i^2 \leftarrow x_{i,2} \oplus LSH_i(\delta)$
    $c_i^1 \leftarrow x_{i,1} \oplus m_i^1,\ c_i^2 \leftarrow x_{i,2} \oplus m_i^2$
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$
$\tau \leftarrow \left(\tau_1^1, \tau_1^2, \tau_2^1, \tau_2^2, ..., \tau_l^1, \tau_l^2\right)$
$\delta_0 \leftarrow \delta,\ \eta_0 \leftarrow \eta$
for $i = 1$ to l **do**
    $\delta_i \leftarrow f_{(\eta_{i-1} \oplus c_i^2) \| \left(\overline{\tau_i^1 \oplus \tau_i^2}\right)}(\delta_{i-1} \oplus c_i^1),\ \eta_i \leftarrow f_{\left(\overline{\eta_{i-1} \oplus c_i^2}\right) \| (\tau_i^1 \oplus \tau_i^2)}\left(\overline{\delta_{i-1} \oplus c_i^1}\right)$
end for
$t_1 \leftarrow f_{\eta_l \| (\overline{\delta_0 \oplus \eta_0})}(\delta_l),\ t_2 \leftarrow f_{\overline{\eta_l} \| (\delta_0 \oplus \eta_0)}\left(\overline{\delta_l}\right)$
$T \leftarrow t_1 \oplus t_2$
Return (C, T)

Algorithm 16 Phase-3 (PH-3): Decryption mode of Parallel-AE: Serial-T.G

Decrypt C and call PH-1 (Algorithm15)
Partitioning: $c_i^j \in C$ such that $(c_1^1, c_1^2), .., \left(c_l^1, c_l^2\right)$, where $j \in \{1,2\}, i \leq l$
for $i = 1$ to l **do**
    $x_{i,1} \leftarrow F_{k \oplus \overline{LSH_i(\delta')}}(LSH_i(\delta))$
    $x_{i,2} \leftarrow F_{k \oplus LSH_i(\delta')}\left(\overline{LSH_i(\delta)}\right)$
    $\tau_i^1 \leftarrow x_{i,1} \oplus LSH_i(\delta),\ \tau_i^2 \leftarrow x_{i,2} \oplus LSH_i(\delta)$
    $m_i^1 \leftarrow x_{i,1} \oplus c_i^1,\ m_i^2 \leftarrow x_{i,2} \oplus c_i^2$
end for
$C \leftarrow \left(c_1^1, c_1^2, c_2^1, c_2^2, ..., c_l^1, c_l^2\right)$
$\tau \leftarrow \left(\tau_1^1, \tau_1^2, \tau_2^1, \tau_2^2, ..., \tau_l^1, \tau_l^2\right)$
$\delta_0 \leftarrow \delta,\ \eta_0 \leftarrow \eta$
for $i = 1$ to l **do**
    $\delta_i \leftarrow f_{(\eta_{i-1} \oplus c_i^2) \| \left(\overline{\tau_i^1 \oplus \tau_i^2}\right)}(\delta_{i-1} \oplus c_i^1),\ \eta_i \leftarrow f_{\left(\overline{\eta_{i-1} \oplus c_i^2}\right) \| (\tau_i^1 \oplus \tau_i^2)}\left(\overline{\delta_{i-1} \oplus c_i^1}\right)$
end for
$t_1 \leftarrow f_{\eta_l \| (\overline{\delta_0 \oplus \eta_0})}(\delta_l),\ t_2 \leftarrow f_{\overline{\eta_l} \| (\delta_0 \oplus \eta_0)}\left(\overline{\delta_l}\right)$
$T \leftarrow t_1 \oplus t_2$
If T is valid then return M or $\perp$

5.7 Security Proof Sketch: The Scheme of Parallel-AE

Under this section, we provide basic security proof sketch of the second scheme of Parallel-AE. At first, we provide privacy security proof sketch. Actually, we provide three types tag generation under this second scheme of Parallel-AE. But, the privacy security proof sketch is similar for three types of authentication (T.G: tag generation) under the scheme of Parallel-AE. Later, we provide authenticity security proof sketch for different three types of authentication (tag generation). However, it varies because of three design variations. In summary, the first variation of authentication (Semi-Parallel-T.G) under the scheme of Parallel-AE can achieve birthday bound security margin because $2n \rightarrow n$-bit compression function. In addition, the second variant authentication (Parallel-T.G) has birthday bound authenticity security margin. The third variant authentication (Serial-T.G) is based on $3n \rightarrow 2n$-bit compression function. Hence, it is expected to achieve higher authenticity security margin.

5.7.1 Privacy Security: The Scheme of Parallel-AE

Privacy security proof sketch of the scheme of Parallel-AE is similar under the first, second, and third variant of authentication (Semi-Parallel-T.G, Parallel-T.G, and Serial-T.G). Because, the encryption mode is unique for three variants authentication. However, the privacy security depends on random behaviour of encryption mode. We encrypt message pair by pair in every iteration under the two calls of keyed function. Hence, it depends on the behaviour of random output properties of keyed function. If output is random then the second scheme satisfies privacy security margin. On the contrary, the input characteristics of PRF function is also important. It should be random also. Actually, our proposed second scheme is based on parallel operation. We use GF operation in every rotation of encryption for generating unique and fresh nonce values. Hence, it is difficult to change the order of input in the encryption module in respect of the adversary. Under these circumstances, it is expected to achieve birthday-bound privacy security margin under the second scheme.

5.7.2 Authenticity Security: The Scheme of Parallel-AE

In this section, we mention the security proof sketch of authenticity. Basically, we assume two properties ensure the authenticity of any authenticated encryption. First one is the random characteristics of input of any cryptographic compression function for generating tag. And the second one is standard security notions of cryptographic compression function such as collision resistance, and preimage resistance.

First Variation of Authentication: Semi-Parallel-T.G

According to the construction of the second scheme of Parallel-AE and the first variant of tag generation (authentication), we use $(n-1)$ encryption for generating input of cryptographic compression function. Our process is based on semi-parallel. In every level, we encrypt pair by pair (see 5.6.1). In addition, our basic assumption is $c \leftarrow F_{k \oplus a}(b) \neq c' \leftarrow F_{k \oplus b}(a)$. Hence, it is infeasible to change the order in respect of adversary. Therefore,

the input of cryptographic compression is random. Our compression function is based on $2n \rightarrow n$-bit. In addition, it satisfies collision and preimage resistance. Therefore, it is expected to achieve birthday-bound authenticity security margin.

Second Variation of Authentication: Serial-T.G

According to the construction of the scheme of Parallel-AE and the second variant authentication (Serial-T.G), we need total $n+2$ encryption call for generating tag. However, the authentication process is serial. It is based on $3n \rightarrow 2n$-bit compression function. In addition, the input of tag generation are satisfied random characteristic because of MD fashion cryptographic compression function. Hence, it is expected to get better authenticity security notion.

5.8 Security Analysis of the Scheme of Parallel-AE

5.8.1 Privacy Security Analysis: The Scheme of Parallel-AE

Privacy security of the proposed second scheme of Parallel-AE is defined as to distinguish between the output of encryption module and the output of random oracle RO in respect of adversary $\mathcal{A}$. In addition, the characteristics of $\mathcal{A}$ are unique nonce and associated data. Our security proof is based on multiple games, where first game runs the proposed scheme and last game directs the random oracle. For example, adversary $\mathcal{A}$ is allowed to ask (N_1, m_1) . . . . (N_l, m_l).

Theorem 5.3. Let $\mathrm{AE}_T^{\mathrm{SS}}$ be the proposed second authenticated encryption scheme (Parallel-AE) where $n \geq 1$. We assume there is an adversary $\mathcal{A}$ that is allowed to ask proposed scheme oracle (based on ideal permutation). In addition, it is allowed to make query on random oracle (RO). Furthermore, $\mathcal{A}$ can query at most q, where total number of query is σ. Under these circumstances, the advantage of adversary is defined to distinguish between $\mathcal{E}.\mathrm{AE}_T^{SS}$ and RO. Therefore, the advantage is quantified as:

$$\mathrm{Adv}^{\mathrm{priv}}_{\mathrm{AE}_T^{\mathrm{SS}}}(\mathcal{A}) \leq \sigma(\sigma-1)\big/2^{2n} + \sigma/2^n + 3/2^n$$

Proof. Our mentioned security proof is based on multiple games. In addition, it is very simple and easy to understand. The first game is noted as game_1. The task of this game is to implement the proposed scheme. In this way, we define certain games for different issues. Moreover, the last game is noted as game_4. The purpose of this game is to inherit the random oracle. We actually show the transition of game_1 to game_4. Moreover, there are certain collision events of this transition process. The probability of these collision events are defined as the advantage of an adversary. In addition, the fresh values are selected from the set of uniform distribution $(\mathcal{U}, \mathcal{V}, \mathcal{Y})$ if any collision occurs.

game_1. This game invokes the proposed scheme $\mathrm{AE}_T^{\mathrm{SS}}$. In addition, it receives input as N, M , where the output are C, T (under ideal permutation). Hence,

$$\Pr\left[\mathcal{A}_{\pi}^{\mathrm{AE}_T^{\mathrm{SS}}} \rightarrow 1\right] = \Pr\left[\mathcal{A}^{\mathrm{game}_1} \rightarrow 1\right] \tag{5.17}$$

game_2. This game is executed under random permutation. Hence, there are certain chances to collide some values. Hence, the advantage of adversary is:

$$\Pr\left[\mathcal{A}^{\mathrm{game}_2} \rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathrm{game}_1} \rightarrow 1\right] \leq \sigma/2^n \tag{5.18}$$

game_3. According to our scheme's construction, there is a chance to make collision for different two iteration's outputs. In addition, a pair of output is produced in each iteration. Hence, there is another chance to collide within pair. Moreover, a collision can be occurred under initialization values. Furthermore, a collision can be occurred under tag generation. We assume an event HiT for making any collision. Hence, for different two iterations the probability of collision is $\Pr[\mathrm{HiT}] \leq \sigma(\sigma-1)/2^{2n}$. Next, the collision probability under single iteration is $\Pr[\mathrm{HiT}] \leq \sigma/2^n$. Furthermore, the collision probability under initialization values is $\Pr[\mathrm{HiT}] \leq 2/2^n$. Additionally, $\Pr[\mathrm{HiT}] \leq 1/2^n$ is the collision probability value under the tag generation. However, if any collision occurs then fresh values will be randomly chosen from the set of uniformly distributed strings $(\mathcal{U}, \mathcal{V}, \mathcal{Y})$. The union bound of all the collision events are defined as to distinguish between game_2 and game_3.

game_4. This game inherits the random oracle. Literally, the last game doesn't contain any collide events. In addition, all values are fresh and unique. Hence, the difference between game_3 and game_4 is nominal. Fianally,

$$\Pr\left[\mathcal{A}^{\text{RO}} \to 1\right] = \Pr\left[\mathcal{A}^{\text{game}_3} \to 1\right]$$

Therefore, **Theorem 5.3** is satisfied by taking the union bound of the probability of all collision events.

5.8.2 Authenticity Security Analysis: The Scheme of Parallel-AE: Serial-T.G

In this section, we show an informal proof sketch of authenticity under the second scheme of third variant authentication (Serial-T.G) (Fig. 5.12). We make two groups here. In the first group, we process or re-encrypt cipher-text. In the second group, we use the last encrypted value of the first group as input of tag generation. At first, we show that the input of tag generation is random. Next, we show that the output of cryptographic compression function or tag generation satisfies collision and preimage resistance.

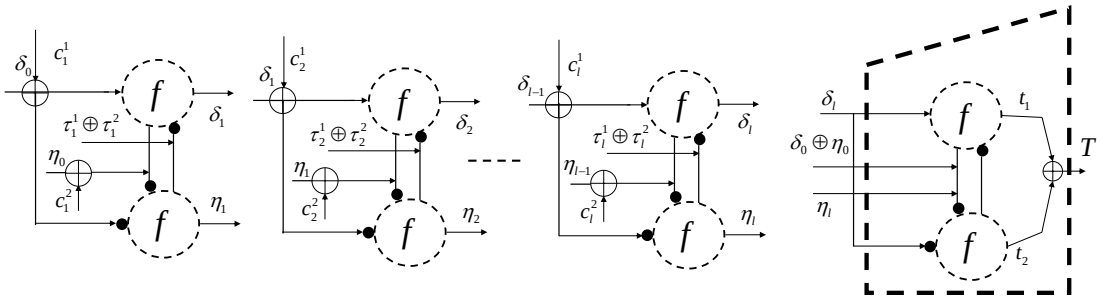


Figure 5.11: Tag Generation: Serial-T.G of the scheme of Parallel-AE

At first, under the first group for any iteration output is $\delta_i \leftarrow f_{(\eta_{i-1} \oplus c_i^2) \| \left(\overline{\tau_i^1 \oplus \tau_i^2}\right)} \left(\delta_{i-1} \oplus c_i^1\right)$, $\eta_i \leftarrow f_{\left(\overline{\eta_{i-1} \oplus c_i^2}\right) \| (\tau_i^1 \oplus \tau_i^2)} \left(\overline{\delta_{i-1} \oplus c_i^1}\right)$. In addition, this operation mode of the first group is serial. Hence, the last output of δ_l and η_l depend on the intermediate all state's output randomness. Due to the block-cipher property, all intermediate output are random also. Under this circumstance, the collision scenario for any two different i-th and j-th query of δ_l and η_l are:

$$\left[\begin{array}{l} \text{for } (1 \text{ to } l) \\ \delta_i \leftarrow f_{(\eta_{i-1} \oplus c_i^2) \| \left(\overline{\tau_i^1 \oplus \tau_i^2}\right)} \left(\delta_{i-1} \oplus c_i^1\right), \\ \eta_i \leftarrow f_{\left(\overline{\eta_{i-1} \oplus c_i^2}\right) \| (\tau_i^1 \oplus \tau_i^2)} \left(\overline{\delta_{i-1} \oplus c_i^1}\right) \end{array}\right] = \left[\begin{array}{l} \text{for } (1 \text{ to } l) \\ \delta_j \leftarrow f_{(\eta_{j-1} \oplus c_j^2) \| \left(\overline{\tau_j^1 \oplus \tau_j^2}\right)} \left(\delta_{j-1} \oplus c_j^1\right), \\ \eta_j \leftarrow f_{\left(\overline{\eta_{j-1} \oplus c_j^2}\right) \| \left(\tau_j^1 \oplus \tau_j^2\right)} \left(\overline{\delta_{j-1} \oplus c_j^1}\right) \end{array}\right]$$

We define an event $COLL$ to find out the probability of making collision under the iteration of i and j. Hence, the probability of collision under the event $COLL$ is:

$$\Pr[COLL] \leq \sum_{i=1}^{q} \frac{2(i-1)}{(2^n - (2i-2))(2^n - (2i-1))} \leq \frac{q^2 - q}{(2^n - 2q)^2}$$

Hence, we can claim that the last output or input of cryptographic compression function is random. Under this circumstance, next we will show that the input of cryptographic compression function or tag generation produces random output. In addition, the output of cryptographic compression function satisfies the preimage resistance security notions. According to the construction of the third variant authentication of the second scheme, the preimage resistance scenario is:
Let adversary $\mathcal{A}$ can arbitrary choose a pair of value such as z_1 and z_2. Hence, it is needed to find out the collision probability of $F(p_1, p_2, p_3) = z_1, z_2$. For example, $F(W) \leftarrow f_{\eta_l||(\overline{\delta_0 \oplus \eta_0})}(\delta_l), f_{\overline{\eta_l}||(\delta_0 \oplus \eta_0)}\left(\overline{\delta_l}\right)$ where $\delta_l, \eta_l, (\delta_0 \oplus \eta_0) \in W$ and F is a cryptographic hash that is made by the component function of two calls of f. Under these circumstances, z_1, z_2 be the two points where inversion is needed. That means, adversary is tried to find out $\delta_l, \eta_l, (\delta_0 \oplus \eta_0)$ such that $F(\delta_l, \eta_l, (\delta_0 \oplus \eta_0)) = (z_1, z_2)$. For simplification, we write as $F(\delta, \eta, (\delta_0 \oplus \eta_0)) = (z_1, z_2)$. Under the ideal cipher model, at first query is allowed in adaptive fashion. When, the query response size becomes $N/2$ then the rest of the queries are given to free. The later half will be queried in non-adaptive fashion. In addition, adjacent query pair looks including response: $\left(\eta_l||\left(\overline{\delta_0 \oplus \eta_0}\right), \delta_l, t_1\right), \left(\overline{\eta_l}||(\delta_0 \oplus \eta_0), \overline{\delta_l}, t_2\right)$. Moreover, adversary gets success if the following condition is true:

$$t_1,\, t_2 = z_1 \text{ and } t_1,\, t_2 = z_2$$

Therefore, we need to find out the probability of collision under the domain of adaptive query (NormalQueryWin) and non-adaptive query (SuperQueryWin). Under the NormalQueryWin, the query of response of $\left(\eta_l||\left(\overline{\delta_0 \oplus \eta_0}\right), \delta_l\right)$ query can come from the set size at most $N/2-2$. Hence, the probability of collision is approximately $2/N$. Because of adjacent query the total probability is $2 \times 2/N$. On the contrary, there is a chance to get a collision between $(t_1 = t_2) = (z_1, z_2)$. The probability of this event is $2/N$. Hence, the total probability under the NormalQuerywin is $8/N^2$. For the SuperQueryWin probability is $8/N^2$ [25]. Hence, the total preimage security is bounded by $16/N^2$.

In this phase, we will show the security of collision resistance. Under the collision resistance, it is hard to find x and x' such that $F(x) = F(x')$ when $x \neq x'$. According to the construction of the third variant of the second scheme, adversary tries to find $\left(\eta_l||\left(\overline{\delta_0 \oplus \eta_0}\right), \delta_l\right), \left(\overline{\eta_l}||(\delta_0 \oplus \eta_0), \overline{\delta_l}\right)$ and $\left(\eta_{l'}||\left(\overline{\delta'_0 \oplus \eta'_0}\right), \delta_{l'}\right), \left(\overline{\eta_{l'}}||(\delta_{0'} \oplus \eta_{0'}), \overline{\delta_{l'}}\right)$ such that:

$$f\left(\eta_l||\left(\overline{\delta_0 \oplus \eta_0}\right), \delta_l\right), f\left(\overline{\eta_l}||(\delta_0 \oplus \eta_0), \overline{\delta_l}\right) = f\left(\eta_{l'}||\left(\overline{\delta'_0 \oplus \eta'_0}\right), \delta_{l'}\right), f\left(\overline{\eta_{l'}}||(\delta_{0'} \oplus \eta_{0'}), \overline{\delta_{l'}}\right)$$

, when $\left(\eta_l||\left(\overline{\delta_0 \oplus \eta_0}\right), \delta_l\right), \left(\overline{\eta_l}||(\delta_0 \oplus \eta_0), \overline{\delta_l}\right) \neq \left(\eta_{l'}||\left(\overline{\delta'_0 \oplus \eta'_0}\right), \delta_{l'}\right), \left(\overline{\eta_{l'}}||(\delta_{0'} \oplus \eta_{0'}), \overline{\delta_{l'}}\right)$. Under these conditions, the collision probability is $\Pr[COLL] \leq \frac{3q^2-q}{(2^n-2q)^2}$. Hence, the cryptographic compression function is secure under collision resistance also.

5.9 Contribution Analysis (Current Result)

A cryptographic hash (CH) is an algorithm that invokes an arbitrary domain of the message and returns fixed size of an output. It has enormous applications in the field of cryptography. The construction of the CH depends on a compression function, where the compression function is constructed through a scratch or block-cipher. *In principle,* we try to use compression function as a building tool of authenticated encryption. In addition, we show that the result of the proposed schemes are almost similar to other prominent existing schemes in certain cases (Table 5.1). Moreover, both of the proposed schemes do not support associated data in principle. Hence, we compare our schemes in respect of encryption mode, authenticity mode, privacy security where we exclude the issue of associated data.

Table 5.1: Comparison Study

S. N.	O.M	TCC	r	Privacy Sec.
COPA [77]	Parallel	$a+2m+2$	2	$O(2^{n/2})$
PoE [12]	Non-Sequential	$(m \times 2HF)^* + m$	-	$O(2^{n/2})$
COBRA [92]	Parallel	$(m+GF)^* + 1 + 2$	-	$O(2^{n/2})$
McOE [11]	Serial	$(m+1)^*$	1	$O(2^{n/2})$
CLOC [43]	Serial	$a+2m+1$	2	$O(2^{n/2})$
SILC [46]	Serial	$a+2m+3$	2	$O(2^{n/2})$
OTR [47]	Parallel	$(a+m)^*$	1	$O(2^{n/2})$
APE [78]	Serial	$(a+m)^*$	-	$O(2^{n/2})$
ElmE [93]	Parallel	$a+2m+1$	2	$O(2^{n/2})$
First Scheme* $(V1)$	Serial	$(m \times f^{\text{prng}}) + 2m + 2 + f^{\text{prng}}$	2+c	$O(2^{n/2})$
Second Scheme** $(V1)$	Parallel	$m + (m \times GF) + (m-1)$	2	$O(2^{n/2})$

1. S. N.: Scheme Name, O.M: Operational Mode, r: Efficiency-rate,
2. TTC: Total Cost Counting, c: Stands for PRNG cost
3. a, m, N: Number of Associated-data, Message, and Nonce,
4. HF: Universal hash function, GF: Finite Field Multiplication, $*$: May varies,
5. First Scheme* $(V1)$: Probabilistic-IV-based AE (Does not supprt Associated data)
6. Second Scheme** $(V1)$: Primarily Nonce respect AE (AD is fixed (n-bit only))

Next, we show that we have certain advantages under some contexts (Fig. 5.12). We proposed three different types of authentication (T.G: tag generation) under the first scheme of Serial-AE such as Semi-Parallel-T.G, Serial-T.G, and Parallel-T.G. The first variant of tag generation (Semi-Parallel-T.G) needs less resources (Fig. 5.13). However, the second variant of tag generation (Serial-T.G) needs more resources and it is based on $3n \rightarrow 2n$-bit cryptographic compression function. Finally, the third variant (Parallel-T.G) under the first scheme of Serial-AE needs two calls of block-cipher. The proposed second construction of Parallel-AE has two variants of tag generation (authentication). The first variant (Semi-Parallel-T.G) needs less resources and operates in semi-parallel. The second variant of tag generation needs more resources and it (Serial-T.G) is based on $3n \rightarrow 2n$-bit compression function. This variant is our conceptual work where we expect

higher authenticity security margin can be achieved.

Figure 5.12 is created by the tag generation (authentication) analysis of the different types of authenticated encryption constructions. For example, the scheme of authenticated encryption OTR: It needs one call of block-cipher function for tag generation including certain pre-computation cost of GF. Another example is CLOC: It needs $n+1$ call of block-cipher function for creating Tag (authentication).

	Tag Generation (Authentication Mode)	
CLOC	**Operational Mode**	Serial
	Encryption Cost	n+1
SILC	**Operational Mode**	Serial
	Encryption Cost	n+2
OTR	**Operational Mode**	Parallel
	Encryption Cost	1 (may vary)
COBRA	**Operational Mode**	Parallel
	Encryption Cost	n+2 (may vary)
COPA	**Operational Mode**	Parallel
	Encryption Cost	n+2

	Cost for Encryption Mode	**Tag Generation (Authentication Mode)** (T.G)			
			Semi-Parallel-T.G	**Serial-T.G**	**Parallel-T.G**
Serial-AE (Proposed First Scheme: FS)	n Encryption + $n \times F^{\text{prng}} + 2$	**Operational Mode** (For T.G)	Semi-Parallel	Serial	Parallel
		Cost for Tag Generation	$(n-1)+1$	$(n+1)+3$	2

	Cost for Encryption Mode	**Tag Generation (Authentication Mode)** (T.G)		
			Semi-Parallel-T.G	**Serial-T.G**
Parallel-AE (Proposed Second Scheme: SS)	n Encryption + Pre-computation of GF + 2	**Operational Mode** (For T.G)	Semi-Parallel	Serial
		Cost for Tag Generation	$(n-1)+1$	$n+2^*$

$n \times \boxed{GF^*}$: It is already computed in encryption mode, $n+\boxed{2^*}$: Key Size of function is Double

Figure 5.12: Significance of the Proposed Schemes [43, 46, 47, 77, 92]: **Note.** Our Proposed Schemes are not based on Associated-Data

We have two proposals of authenticated encryption such as Serial-AE (Based on Probabilistic-IV) and Parallel-AE (Based on nonce-respect). The first scheme of Serial-AE has three different types of tag generation (T.G) such as Semi-Parallel-T.G, Serial-T.G, and Parallel-T.G. Moreover, the proposed second scheme of Parallel-AE places two distinct types of tag generation or authentication. These are Semi-Parallel-T.G, and Serial-T.G. We compare all these tag generation results with existing certain familiar scheme's tag generation outcomes. For example, the third variant of tag generation (Parallel-T.G) of the first scheme (Serial-AE) need less resources for creating tag like two calls of block-cipher. On the contrary, the proposal of the third variant of tag generation or authentication (Serial-T.G) of the second scheme (Parallel-AE) needs more resources because of $3n \rightarrow 2n$-bit compression function. Interestingly, this proposal is an important idea to get better authenticity security margin. However, we do not provide

formal security proof under this variant. But we show that the cryptographic compression function of Serial-T.G variant satisfies the collision and preimage resistance security bound.

Future Target. Our next target is to provide rigorous authenticity security proof for the second and third variant (Serial-T.G, Parallel-T.G) of the scheme of Serial-AE. In addition, We will provide rigorous security proof of the second variant authenticity (Serial-T.G) of the second scheme. However, the current results are based on theoretical analysis. Hence, our next target is to simulate these schemes and compare with the existing schemes in respect of time complexity and hardware efficiency.

Chapter 6

Small and Variable Message Encryption

In modern cryptography, message encryption is an important tool for providing data-privacy and authenticity. In many applications, it is used such as password storage, data integrity check, wireless network, automated teller machine card, and credit card. Usually, the size of a message is arbitrary. In addition, many techniques and constructions are available for the variable and fixed size of message encryption. Interestingly, we address the issue of small domain message encryption (SDE). Generally, the existing constructions are based on blockcipher such as AES/DES and scratch function. Hence, these solutions are reasonable for the bigger size of data. However, these are heavy and expensive for the small size of message encryption under the platform of Internet of Technology-end device (IoT), and resource constrained devices. In addition, the size of plaintext and ciphertext should be equal for satisfying the property of small domain encryption. Actually, J.Black and P.Rogaway formally addressed the above issues for the first time. Following that certain schemes have been launched under the SDE such as Mix-and-Cut shuffle, Swap-or-Not, Thorp-Shuffle, and FNR. Moreover, Sometimes-Recursive shuffle (SRS) is the pioneer construction yet in respect of low encryption time. However, it needs to execute 1000 calls of AES for achieving full security. Therefore, the construction of SRS is also heavy and expensive for the IoT environment and resource constrained devices. Under these circumstances, we propose a simple scheme that follows by Fiestel structure. The internal format is based on small keyed-function. Our construction can encrypt small size of a message. Furthermore, the size of plaintext and ciphertext are equal.

6.1 A Concept of Construction of Small Domain Encryption

Our proposed scheme is based on small keyed-function and noted as SETM where SETM directs "simple encryption for tiny message". In addition, SETM can encrypt a small domain of message. It follows the Feistel structure. Furthermore, we inspired to build the SETM from the constructions of [30, 31, 36]. In addition, our construction is member of a partial security margin group. Our proposed construction satisfies the following objectives:

- It can encrypt small chunk of message

- It can encrypt arbitrary size message without padding
- It preserves the equal length of plain-text and cipher-text
- It is light due to use of small keyed-function

There are certain notations such as f: Small keyed function, tr: Truncation function, $||$: Concatenation, q: query, M, C: Message, Cipher-text, $\boxplus$: XoR, $\mathcal{A}$: Adversary, k: Size of key, n: Size of message, $\otimes$: X-NoR operation, α: message size ($|m_l| = \alpha, \alpha < n$) (arbitrary size message), and c_{end}: cipher, $c_{\text{end}} \leftarrow^{\text{tr}} (c_{l,1} \oplus c_{l,2})$ (arbitrary size message). Furthermore, we assume $\mathcal{X}$ is a finite set of strings where x is uniformly distributed as $\mathcal{X} \leftarrow^{\$} x$. Moreover, $\mathcal{Y}$ satisfies $\mathcal{Y} \leftarrow^{\$} y$ for y. In our proposed scheme, two calls of keyed-functions are used. Therefore, $r = 2$, where r directs the number of calling functions in each iteration. However, the Feistel structure is not secure under $r = 2$ [96, 97]. Usually, it is secure when it satisfies $r > 2$ [96, 97]. Under this circumstance, the value of r is variable in our security proof for providing better security margin. We define a small keyed-function as $f : K \times M \rightarrow M$ where K means key space and M directs message space. In addition, $f_k(\cdot) = f(k, \cdot)$ is a permutation over M for every $k \in K$. We assume there is an adversary $\mathcal{A}$ that can access an encryption (Enc) oracle of the proposed scheme and an oracle of random function (RO). The advantage of an adversary is defined to distinguish between the output of random-oracle and the output of the proposed scheme. Moreover, the adversary has access on ideal permutation ($). Hence, $\text{Adv}_f^{\text{cca}}(\mathcal{A}) = \Pr\left[\mathcal{A}_{\$}^{\text{Enc}(\cdot)} = 1\right] - \Pr\left[\mathcal{A}_{\pi}^{\text{RO}(\cdot)} = 1\right]$. We assume adversary $\mathcal{A}$ has non-adaptive query feature. In addition, chosen plain-text attack by any adversary is defined as each query runs under encryption query [ref]. Furthermore, we define PRNG functions as f_{pr_1} and f_{pr_2}. The operation of PRNG functions is $f_{pr_{1,2}} \rightarrow^{\text{u,r}} \{0,1\}^n$, where $u:$ uniform $r:$ random.

6.2 Definition of the Proposed Scheme of SETM

In this section, we define the proposed scheme through Figure 6.1 and Definition 1. We assume the proposed scheme is based on a small keyed function such as $f_{1,2}^{k_1,k_2} : \{0,1\}^n \rightarrow \{0,1\}^n$ (e. g. n : 8-bits). Our proposed scheme operates in Feistel structure fashion through two calls of the function. We inspired to take the advantage of the Feistel structure from [30, 31, 36]. Moreover, the proposed scheme encrypts small size of the message without padding. In addition, the length of the input-message and output-ciphertext are equal.

Definition 6.1. Let $f : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a small keyed-function where k indicates the key length and n directs the blocklength. We use Feistel structure. Hence, we need two keyed-functions. These two functions are named as $f_1^{k_1}$ and $f_2^{k_2}$. These two functions follow the function of f. Hence, $f_{1,2}^{k_1,k_2} : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$. In addition, we use two more PRNG functions (f_{pr_1}, f_{pr_2}). Under these circumstances, the output

$(c_{i,1}, c_{i,2})$ of the proposed scheme is defined as follows:

$$
\begin{aligned}
&c_{i,1} \leftarrow z_{i,1} \\
&\quad \text{where, } z_{i,1} \leftarrow w_{i,1} \otimes m_{i,1},\ w_{i,1} \leftarrow f_{pr_{i,1}}(y_{i,1}), \\
&\qquad y_{i,1} \leftarrow x_{i,1} \boxplus \overline{m_{i,2}},\ x_{i,1} \leftarrow f_1^{k_{i,1}}(k_{i,1}, m_{i,2}), \\
&c_{i,2} \leftarrow z_{i,2} \\
&\quad \text{where, } z_{i,2} \leftarrow w_{i,2} \otimes m_{i,2}, \\
&\qquad w_{i,2} \leftarrow f_{pr_{i,2}}(y_{i,2}),\ y_{i,2} \leftarrow x_{i,2} \boxplus \overline{\tau_i}, \\
&\qquad x_{i,2} \leftarrow f_2^{k_{i,2}}(k_{i,2}, \tau_i),\ \tau_i \leftarrow z_{i,1} \boxplus m_{i,1}
\end{aligned}
$$

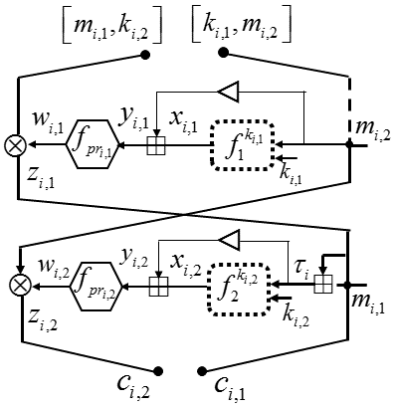


Figure 6.1: Proposed Scheme of SETM

We describe the encryption process of the fixed-size message by algorithm 18. In addition, algorithm 19 represents a decryption method of the fixed-size ciphertext. Furthermore, algorithm 20 and 21 direct the encryption and decryption process for flexible size of message/ciphertext respectively.

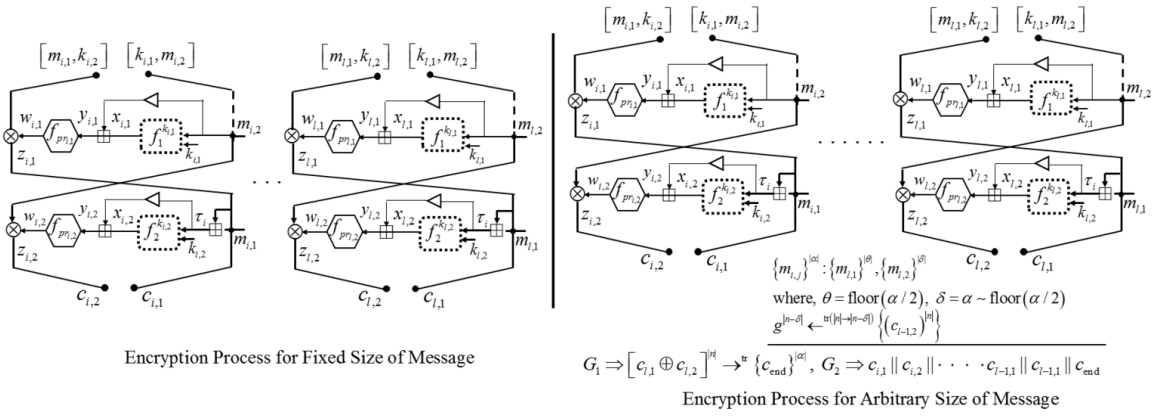


Figure 6.2: Encryption Process for the SETM

Algorithm 17 Process of Encryption-1

Encrypt M

Partition :

$$m_{i,j}, m_{i+1,j+1}, ..., m_{l-1,j}, m_{l,j+1} \in M \left[i \leq l, j = 1\right]$$

and $\forall m_{i,j}$ satisfies the property of $|m_{i,j}| = \left|f_{1,2}^{k_{i,1},k_{i,2}}\right|$

for $i = 1$ to l **do**

$x_{i,1} \leftarrow f_1^{k_{i,1}}(k_{i,1}, m_{i,2}),\ y_{i,1} \leftarrow x_{i,1} \boxplus \overline{m_{i,2}}$

$w_{i,1} \leftarrow f_{pr_{i,1}}(y_{i,1}),\ z_{i,1} \leftarrow w_{i,1} \otimes m_{i,1}$

$c_{i,1} \leftarrow z_{i,1}$

$x_{i,2} \leftarrow f_2^{k_{i,2}}(k_{i,2}, \tau_i), \tau_i \leftarrow z_{i,1} \boxplus m_{i,1}$

$y_{i,2} \leftarrow x_{i,2} \boxplus \overline{\tau_i},\ w_{i,2} \leftarrow f_{pr_{i,2}}(y_{i,2})$

$z_{i,2} \leftarrow w_{i,2} \otimes m_{i,2}$

$c_{i,2} \leftarrow z_{i,2}$

end for

$c_{i,1}||c_{i,2}||\cdots||c_{i,l} \rightarrow C$

if $|C| = |M|$ **then**

Return : C

else

Return : $\perp$

end if

Algorithm 18 Process of Decryption-1

Decrypt C

Partition :

$$c_{i,j}, c_{i+1,j+1}, ..., c_{l-1,j}, c_{l,j+1} \in C \left[i \leq l, j = 1\right]$$

and $\forall c_{i,j}$ satisfies the property of $|c_{i,j}| = \left|f_{1,2}^{k_{i,1},k_{i,2}}\right|$

for $i = 1$ to l **do**

$x_{i,1} \leftarrow f_1^{k_{i,1}}(k_{i,1}, c_{i,2}),\ y_{i,1} \leftarrow x_{i,1} \boxplus \overline{c_{i,2}}$

$w_{i,1} \leftarrow f_{pr_{i,1}}(y_{i,1}),\ z_{i,1} \leftarrow w_{i,1} \otimes c_{i,1}$

$m_{i,1} \leftarrow z_{i,1}$

$x_{i,2} \leftarrow f_2^{k_{i,2}}(k_{i,2}, \tau_i), \tau_i \leftarrow z_{i,1} \boxplus c_{i,1}$

$y_{i,2} \leftarrow x_{i,2} \boxplus \overline{\tau_i},\ w_{i,2} \leftarrow f_{pr_{i,2}}(y_{i,2})$

$z_{i,2} \leftarrow w_{i,2} \otimes c_{i,2}$

$m_{i,2} \leftarrow z_{i,2}$

end for

$m_{i,1}||m_{i,2}||\cdots||m_{i,l} \rightarrow M$

if $|M| = |C|$ **then**, Return: M

else

Return: $\perp$

end if

6.3 Security Analysis of the SETM

We define a game (GE) that has two players such as Pl_1 and Pl_2. Moreover, there is an adversary $\mathcal{A}$ under this game. In addition, Pl_1 simulates the proposed scheme

Algorithm 19 Process of Encryption-2

Encrypt M

Partition :

$$m_{i,j}, m_{i+1,j+1}, ..., m_{l-1,j}, m_{l,j+1} \in M \left[i \leq l, j = 1 \right]$$

$$\triangleright \text{ Condition-1} : |M| \geq l \, |m_{i,j}| \left[\begin{array}{l} |m_{i,j}| = f_{1,2}^{k_{i,1}, k_{i,2}} \\ \text{and } l \text{ is the} \\ \text{number of partition} \end{array} \right]$$

$$\triangleright \text{ Satisy the following Condition-2:}$$
$$|m_{i,j}| = \left| f_{1,2}^{k_{i,1}, k_{i,2}} \right| \text{ (where, } i = l - 1)$$
$$\text{and } |m_{i,j}| < \left| f_{1,2}^{k_{i,1}, k_{i,2}} \right| \text{ (where, } i = l)$$
$$\text{and Let, } m_{l,j} = \{ \}^{|\alpha|} \text{ (where, } \alpha \leq n)$$

for $i = 1$ to $l - 1$ **do**
  $x_{i,1} \leftarrow f_1^{k_{i,1}} (k_{i,1}, m_{i,2}),\ y_{i,1} \leftarrow x_{i,1} \boxplus \overline{m_{i,2}}$
  $w_{i,1} \leftarrow f_{pr_{i,1}} (y_{i,1}),\ z_{i,1} \leftarrow w_{i,1} \otimes m_{i,1}$
  $c_{i,1} \leftarrow z_{i,1}$
  $x_{i,2} \leftarrow f_2^{k_{i,2}} (k_{i,2}, \tau_i), \tau_i \leftarrow z_{i,1} \boxplus m_{i,1}$
  $y_{i,2} \leftarrow x_{i,2} \boxplus \overline{\tau_i},\ w_{i,2} \leftarrow f_{pr_{i,2}} (y_{i,2})$
  $z_{i,2} \leftarrow w_{i,2} \otimes m_{i,2}$
  $c_{i,2} \leftarrow z_{i,2}$
end for
for $i = l$ **do**
  partition $m_{l,j} : \{m_{l,1}\}^{|\lfloor \alpha/2 \rfloor|}$ and $\{m_{l,2}\}^{|\alpha - \lfloor \alpha/2 \rfloor|}$
  let $\lfloor \alpha/2 \rfloor = \theta, \alpha - \lfloor \alpha/2 \rfloor = \delta$
  $g^{|n-\delta|} \leftarrow^{\text{tr}(|n| \to |n-\delta|)} \left\{ (c_{l-1,2})^{|n|} \right\}$
  $x_{l,1} \leftarrow f_1^{k_{l,1}} \left(k_{l,1}, \{m_{l,2}\}^{|\delta|} || g^{|n-\delta|} \right)$
  $y_{l,1} \leftarrow x_{l,1} \boxplus \overline{\{m_{l,2}\}^{|\delta|}},\ w_{l,1} \leftarrow f_{pr_{l,1}} (y_{l,1})$
  $z_{l,1} \leftarrow w_{l,1} \otimes \{m_{l,1}\}^{|\theta|},\ c_{l,1} \leftarrow z_{l,1}$
  $x_{l,2} \leftarrow f_2^{k_{l,2}} (k_{l,2}, \tau_l),\ \tau_l \leftarrow z_{l,1} \boxplus \{m_{l,1}\}^{|\theta|}$
  $y_{l,2} \leftarrow x_{l,2} \boxplus \overline{\tau_l},\ w_{l,2} \leftarrow f_{pr_{l,2}} (y_{l,2})$
  $z_{l,2} \leftarrow w_{l,2} \otimes \{m_{l,2}\}^{|\delta|},\ c_{l,2} \leftarrow z_{l,2}$
end for
$G_1 \leftarrow \left([c_{l,1} \oplus c_{l,2}]^{|n|} \to^{\text{tr}} \{c_{\text{end}}\}^{|\alpha|} \right)$
$G_2 \leftarrow c_{i,1} || c_{i,2} || \cdot \cdot \cdot || c_{l-1,1} || c_{l-1,2} || c_{\text{end}}$
$C \leftarrow G_2$
if $|C| = |M|$ **then** then return C else $\perp$
end if

(SETM-E $(\cdot)$, D $(\cdot)$) and gives feedback to $\mathcal{A}$. On the contrary, Pl_2 mimics random-oracle (RO) and returns corresponding input-output to the adversary. The advantage of an adversary is defined as the success probability for distinguishing of the output of two players. In addition, we assume that $N = 2^n$ and $q \in (1, ..., N)$, where $\mathcal{A}$ can

Algorithm 20 Process of Decryption-2

Decrypt C

Partition :

$$c_{i,j}, c_{i+1,j+1}, ..., c_{l-1,j}, c_{l,j+1} \in C\,[i \le l, j = 1]$$

$$\triangleright \text{ Condition-1} : |C| \ge l\,|c_{i,j}| \left[\begin{array}{l} |c_{i,j}| = f_{1,2}^{k_{i,1},k_{i,2}} \\ \text{and } l \text{ is the} \\ \text{number of partition} \end{array} \right]$$

$\triangleright$ Satisy the Condition-2 as followed:

$|c_{i,j}| = \left| f_{1,2}^{k_{i,1},k_{i,2}} \right|$ (where, $i = l-1$)

and $|c_{i,j}| < \left| f_{1,2}^{k_{i,1},k_{i,2}} \right|$ (where, $i = l$)

and Let, $c_{l,j} = \{\ \}^{|\alpha|}$ (where, $\alpha \le n$)

for $i = 1$ to $l-1$ **do**

  $x_{i,1} \leftarrow f_1^{k_{i,1}}(k_{i,1}, c_{i,2}),\ y_{i,1} \leftarrow x_{i,1} \boxplus \overline{c_{i,2}}$

  $w_{i,1} \leftarrow f_{pr_{i,1}}(y_{i,1}),\ z_{i,1} \leftarrow w_{i,1} \otimes c_{i,1}$

  $m_{i,1} \leftarrow z_{i,1}$

  $x_{i,2} \leftarrow f_2^{k_{i,2}}(k_{i,2}, \tau_i), \tau_i \leftarrow z_{i,1} \boxplus c_{i,1}$

  $y_{i,2} \leftarrow x_{i,2} \boxplus \overline{\tau_i},\ w_{i,2} \leftarrow f_{pr_{i,2}}(y_{i,2})$

  $z_{i,2} \leftarrow w_{i,2} \otimes c_{i,2}$

  $m_{i,2} \leftarrow z_{i,2}$

end for

for $i = l$ **do**

  partition $c_{l,j} : \{c_{l,1}\}^{|\lfloor \alpha/2 \rfloor|}$ and $\{c_{l,2}\}^{|\alpha - \lfloor \alpha/2 \rfloor|}$

  let $\lfloor \alpha/2 \rfloor = \theta, \alpha - \lfloor \alpha/2 \rfloor = \delta$

  $g^{|n-\delta|} \leftarrow^{\mathrm{tr}(|n| \to |n-\delta|)} \left\{ (m_{l-1,2})^{|n|} \right\}$

  $x_{l,1} \leftarrow f_1^{k_{l,1}} \left(k_{l,1}, \{c_{l,2}\}^{|\delta|} || g^{|n-\delta|} \right)$

  $y_{l,1} \leftarrow x_{l,1} \boxplus \overline{\{c_{l,2}\}^{|\delta|}},\ w_{l,1} \leftarrow f_{pr_{l,1}}(y_{l,1})$

  $z_{l,1} \leftarrow w_{l,1} \otimes \{c_{l,1}\}^{|\theta|},\ m_{l,1} \leftarrow z_{l,1}$

  $x_{l,2} \leftarrow f_2^{k_{l,2}}(k_{l,2}, \tau_l),\ \tau_l \leftarrow z_{l,1} \boxplus \{c_{l,1}\}^{|\theta|}$

  $y_{l,2} \leftarrow x_{l,2} \boxplus \overline{\tau_l},\ w_{l,2} \leftarrow f_{pr_{l,2}}(y_{l,2})$

  $z_{l,2} \leftarrow w_{l,2} \otimes \{c_{l,2}\}^{|\delta|},\ m_{l,2} \leftarrow z_{l,2}$

end for

$G_1 \leftarrow \left([m_{l,1} \oplus m_{l,2}]^{|n|} \to^{\mathrm{tr}} \{m_{\mathrm{end}}\}^{|\alpha|} \right)$

$G_2 \leftarrow m_{i,1} || m_{i,2} || \cdot\ \cdot\ \cdot || m_{l-1,1} || m_{l-1,2} || m_{\mathrm{end}}$

$M \leftarrow G_2$

if $|M| = |C|$ **then** then return M else $\perp$

end if

ask at most q queries. Therefore, the advantage of $\mathcal{A}$ is quantified as $\mathrm{Adv}^{\mathrm{SN}}_{\mathrm{SETM}(f)}(\mathcal{A}) = \Pr\left[\mathcal{A}_{\$}^{\text{SETM-E}(\cdot),\mathrm{D}(\cdot)} \to 1\right] - \Pr\left[\mathcal{A}_{\pi}^{\mathrm{RO}} \to 1\right]$. Furthermore, we generalize the value of r (rounds) for satisfying higher security under $f_1^{k_1}, f_2^{k_3}, f_3^{k_3}, ..., f_l^{k_l} \to \left(f_{1,2,...,r}^{k_1,k_2,..,k_r} \right)$ (where, $2 \le r \le l$).

Theorem 6.1. Let $N = 2^n$, $q \in (1, ..., N)$ and $r > 2$. Furthermore, adversary $\mathcal{A}$ has access to SETM-E$(\cdot)$, D$(\cdot)$ and RO through \$ and π. Under these circumstances, the advantage of $\mathcal{A}$ is to distinguish between SETM and RO. Hence, the advantage of $\mathcal{A}$ is bounded as:

$$\mathrm{Adv}^{\mathrm{SN}}_{\mathrm{SETM}(f)}(\mathcal{A}) = \Pr\left[\mathcal{A}^{\mathrm{SETM\text{-}E}(\cdot),\mathrm{D}(\cdot)}_{\$} \to 1\right] - \Pr\left[\mathcal{A}^{\mathrm{RO}}_{\pi} \to 1\right] \leq \frac{q}{r+1}\left(\frac{4q}{2^n}\right)^r + \frac{3q}{2^{nr}}$$

Proof. The security proof concept of the proposed construction is simple and easy. We assume, adversary $(\mathcal{A})$ has access to the proposed construction and random oracle. We define a Stage-0 (ST0). Under this stage, player 1 (Pl_1) simulates the proposed scheme. In addition, there are two more stages such as Stage-1 (ST1) and Stage-2 (ST2), where ST2 directs the random oracle. We will show the transition of ST0 to ST2. Furthermore, there are certain collisions for transition of these stages. Actually, these collisions are defined as cost to distinguish between the proposed scheme and random oracle. The collision events will be removed from the query storage and new values will be injected from the set of uniform distribution.

Stage-0 (ST0). There are two players of Pl_1 and Pl_2 under the game GE. However, Pl_1 is used for current stage only. Furthermore, Pl_1 simulates the proposed construction in this stage. In addition, Pl_1 invokes the input of random key, message and the feedbacks corresponding ciphertext and vice-versa to the adversary. The proposed construction's queries are based on random function. Therefore,

$$\Pr\left[\mathcal{A}^{\mathrm{SETM\text{-}E}(\cdot),\mathrm{D}(\cdot)}_{\$} \leftarrow 1\right] = \Pr\left[\mathcal{A}^{\mathrm{ST0}}_{\pi} \leftarrow 1\right] \tag{6.1}$$

Stage-1 (ST1). Under this stage, queries are executed through random function. Hence, the output should be unique. The output of ST1 and ST0 are identical until certain collisions are occurred. These collisions are based on some conditions such as a pair of distinct query (PDQ), single query (SQ), and initialization query (IQ).

- PDQ. We assume that C^{PDQ} be the event of finding a collision under ST1. Let the output of iteration under i and j $(i < j)$ are $(c_{i,1}, c_{i,2})$ and $(c_{j,1}, c_{j,2})$ respectively. If C^{PDQ} be the event of finding a collision pair under the ST1. Therefore, the probability of collision $(coll)$ is $\Pr[coll]$, where $\Pr[coll] = (4i/2^n)$ Moreover, after r rounds, $\Pr\left[C^{PDQ}_l\right] = \Pr[coll_2 \vee coll_3 \vee \ . \ . \ . \ \vee coll_q] = \sum_{i=2}^{q}(4i/2^n)^r = \frac{q}{r+1}(4q/2^n)^r$.

- SQ. We assume that C^{SQ} be the event of finding a collision under this stage for a single query. It is defined as there is a chance to collide between $c_{i,1}$ and $c_{i,2}$ for any iteration of i. Hence we assume that the output of i-th iteration are $c_{i,1}$ and $c_{i,2}$. If C^{SQ} be the event of finding a collision pair under the ST1. Therefore, the probability of collision $(coll)$ is $\Pr[coll]$, where $\Pr[coll] = (1/2^n)$ Furthermore, after r rounds, $\Pr\left[C^{SQ}_l\right] = \Pr[coll_1 \vee coll_2 \vee \ . \ . \ . \ \vee coll_q] = \sum_{i=1}^{q}(1/2^n)^r = \frac{1}{2^{nr}}(q)$.

- IQ. Generally, two chaining values are used in every iteration of the proposed construction. For example, $c_{0,1}$ $c_{0,2}$ be initial values of the proposed scheme. Hence, there is a chance of hitting a collision against these two values. The condition of this collision occurrence is $(c_{i,1} = (c_{0,1} \text{ or } c_{0,2}) \vee c_{i,2} = (c_{0,1} \text{ or } c_{0,2}))$. We assume that C_{IQ} be the event under this stage of the proposed scheme, where $\Pr[C^{IQ}_l] = \Pr[coll_1 \vee coll_2 \vee coll_3 \vee ..... \vee coll_q] = \sum_{i=0}^{q}\left(\frac{2}{2^n}\right)^r \leq 2q/2^{nr}$ after r rounds.

The collide queries are removed from the query storage. In addition, fresh and unique output is invoked from the uniform distribution of the set. Therefore the difference between the ST0 and ST1 is defined as the probability of collision events such as

$$\Pr\left[\mathcal{A}_{\pi}^{\mathrm{ST1}} \leftarrow 1\right] - \Pr\left[\mathcal{A}_{\pi}^{\mathrm{ST0}} \leftarrow 1\right] \leq \frac{q}{r+1}\left(\frac{4q}{2^n}\right)^r + \frac{3q}{2^{nr}} \tag{6.2}$$

Stage-2 (ST2). Under this stage, player 2 (Pl_2) simulates the random oracle. Hence, the output of ST2 and ST1 are identical in respect of the adversary. Therefore,

$$\Pr\left[\mathcal{A}_{\pi}^{\mathrm{ST2}} \leftarrow 1\right] = \Pr\left[\mathcal{A}_{\pi}^{\mathrm{ST1}} \leftarrow 1\right]$$

In addition, we call random oracle based simulation which is played by player 2 as well. Therefore,

$$\Pr\left[\mathcal{A}_{\pi}^{\mathrm{RO}} \leftarrow 1\right] = \Pr\left[\mathcal{A}_{\pi}^{\mathrm{ST2}} \leftarrow 1\right]$$

Finally, **Theorem 6.1** is satisfied by taking the union bound 6.1 and 6.2.

Chapter 7

Conclusion and Future Works

In modern cryptography, Cryptographic compression function has enormous applications such as password storage, data integrity check, and file identifier. Based on the existing schemes, we categorized cryptographic compression function into two fields such as group of security bound and group of efficiency. Under the group of security bound, there are bunch of schemes those are secure under ideal cipher model. However, ideal cipher model needs strong assumption. Hence, it is far from the real world scenario. In addition, weak cipher model based proof technique satisfies less strict assumption. Therefore, it is relatively close to the real world. Still, there are certain problems under the weak cipher model. Hence, we proposed a new proof technique that needs less assumption than that of the weak cipher model. Furthermore, we proposed an $(n, 2n)$ block-cipher based compression function and proved that it is secure under respectively ideal cipher model, weak cipher model, and extended weak cipher model. There is another tropical issue of variable message encryption through cryptographic compression function. Most of the existing schemes are based on block-cipher. Hence, the size of message also depends of block-size. If message size does not fit with the block size then padding is necessary. However, padding itself has certain dis-advantages such as padding oracle attack. Therefore, we proposed schemes of (n, n) block-cipher based compression function. Our proposed schemes can encrypt variable size of message. Moreover, these proposed schemes satisfy reasonable security bound. Under the (n, n) block-cipher compression function, there are some prominent schemes such as MDC-2, MDC-4, MJH, and Bart-12. These existing schemes are secure under ideal cipher model and finite field multiplication model. Therefore, we proposed an (n, n) block-cipher based compression function. This proposed construction is secure under weak cipher model. Moreover, it satisfies upper bound of collision and preimage security. Under the group of efficiency, the familiar schemes have upper efficiency-rate. However, those schemes key scheduling are higher. Moreover, number of calling block-ciphers are upper also. Therefore, we proposed an $(n, 2n)$ block-cipher based cryptographic compression function for providing better efficiency-rate, less key scheduling, and less call of block-ciphers.

In the next phase, we are focus for practical implementation of cryptographic compression function. Cryptographic compression function can be a useful tool of creating an application of authenticated encryption. Under the authenticated encryption, message authentication and encryption are vital tool for assuring secure communication. Under the authenticated encryption, many researches are running based on rigorous security bound. On the contrary, to satisfy efficiency is one of the challenging tasks under the

platform of resource constrained device and IoT. Moreover, there are lots of researches have been done on the issue of none-reuse and nonce-respect security notions for AE. However, little works have been done on probabilistic-IV based authenticated encryption which are secure under weaker security model. Interestingly, IV-based authenticated encryption is expected to suitable under the resource constrained device because of weaker security model. In addition, it satisfies reasonable security bound. Briefly studying the schemes of authenticated encryption, we classify two groups. First one is probabilistic IV-based authenticated encryption and second-one is nonce based authenticated encryption. In addition, we focus for efficiency-rate of AE. Furthermore, we address the issue of number of calling block-ciphers/function for encrypting message under the AE. Moreover, pointing out that the construction of AE is inverse freeness of block-cipher or not. Under the IV-based authenticated encryption, we proposed scheme that satisfies reasonable privacy security bound. In addition, it satisfies inverse free of block-cipher. Moreover, our proposed construction efficiency-rate is 2. Furthermore, we use block-cipher based compression function as encryption primitive for our IV-based authenticated encryption scheme. On the contrary, our proposed nonce based authenticated encryption depends on nonce-respect. We show our scheme needs less call of block-cipher/function in authentication mode in certain case. In addition, efficiency-rate is 1. Moreover, it satisfies satisfiable privacy security bound.

Small domain encryption (SDE) is one of the hot cryptographic topics. It has large number of usage in the field of commercial arena. Most of the existing familiar constructions of small domain encryption are based on block-cipher. Hence, the size of message depends on block-cipher. In addition, there are two branches of SDE such as partial security based SDE and full security based SDE. Under the full security based SDE, the best security bounded scheme needs 1000 calls of AES. However, this phenomena is not satisfying in respect of efficiency. We proposed a construction that is based on small function such as 8, 16, or 20 bits.

Future Perspective in respect of Application. The cryptographic compression function plays very vital role directly in the application field of cryptography. Hence, it is also interesting to prove that the existing constructions are secure and efficient enough under the IoT environment and big data platform. The most recent and upcoming challenges for authenticated encryption are enormous. It is very interesting to understand how the AE co-ops with the technology of IoT. Moreover, it is quite interesting that AE absorbs under the big data application also. Moreover, our AE work in this domain is based on theoretical approach. In addition, we provide informal security approach for achieving upper authenticity security margin under certain variants of authentication. Under these circumstances, we will provide formal rigorous security margin in the continuation of this work. Furthermore, we will simulate the proposed AE schemes and compare with the existing familiar schemes in respect of time complexity and hardware requirements. For small domain encryption, our proposed scheme is based on small function. Hence, it needs to implement in real life. In addition, to observe that the actual hardness of security and efficiency.

Bibliography

[1] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin , *"Hash Functions and RFID Tags: Mind the Gap," LNCS, CHES,* vol. 5154, pp. 283-299, 2008.

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography,* 5th ed, CRC Press, 2001.

[3] W. Stallings, Data and Computer Communications, 10th Edition, Pearson, 2013.

[4] J. Shearer, P. Gutmann, *Government, Cryptography, and the Right To Privacy*, *Journal of Universal Computer Science (J.UCS),* Volume 2, No.3, 1996, p.113

[5] C. E. Shannon, *"Communication Theory of Secrecy Systems," Bell Systems Technical Journal*, vol. 128-4, pp. 656-715, 1949.

[6] A. Riahi, E. Natalizio, Y. Challal, N. Mitton, A. Iera, *"A systemic and cognitive approach for IoT security"*, *IEEE explore, ICNC*, pp. 183-188, 2014.

[7] H. Yoshida *"On the standardization of cryptographic application techniques for IoT devices in ITU techniques for IoT devices in ITU-T and ISO/IEC JTC 1 T and ISO/IEC JTC1*, https://www.ietf.org/proceedings/94/slides/slides-94-saag-2.pdf, 2015

[8] L. Zhang1, W. Wu, P. Wang, "Extended Models for Message Authentication," *LNCS, ICISC*, vol. 5461, pp. 286-301, 2008.

[9] P. Subpratatsavee, P. Kuacharoen, *"Transaction Authentication Using HMAC-Based One-Time Password and QR Code," Computer Science and its Applications*, vol. 330, pp.93-98.

[10] Lorenz. M., *"Authentication and Transaction Security in E-business ," Springer, The Future of Identity in the Information Society*, vol. 262, pp. 175-197, 2008.

[11] E. Fleischmann, C. Forler, S. Lucks, *"McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes"*, *LNCS, FSE*, Vol. 7549, pp. 196-215, 2012.

[12] F. Abed, S. Fluhrer, C. Forler, E. List, S. Lucks, D. McGrew, J. Wenzel, *"Pipelineable On-line Encryption"*, *LNCS, FSE*, Vol. 8540, pp. 205-223, 2015.

[13] D. G. B. Lectures., *A History of U.S. Communications Security,* National Security Agency (NSA), Volumes I, 1973, Volumes II 1981, partially released 2008, additional portions declassified October 14, 2015

[14] Encryption: The Threat, Applications, and Potential Solutions, declassified FBI, NSA, and DOJ, *https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-805-ethics-and-the-law-on-the-electronic-frontier-fall-2005/readings/read_tlp4/*, 1993.

[15] M. Abomhara, G. M. Kien, *"Security and privacy in the Internet of Things: Current status and open issues*, *IEEE explore, PRIMS*, pp. 1-8, 2014.

[16] H. K. Kim, T. H. Kim *"Design on Mobile Secure Electronic Transaction Protocol with Component Based Development," LNCS, ICCSA*, vol. 3043, pp. 461-470, 2004.

[17] L. C. Cao, *"Improving Security of SET Protocol Based on ECC," LNCS, WISM*, vol. 6987, pp. 234-241, 2011.

[18] G. Hanaoka, Y. Zheng, H. Imai, *"LITESET: A light-weight secure electronic transaction protocol," LNCS, Information Security and Privacy*, vol. 1438, pp. 215-226, 2006.

[19] P. Rogaway, *"Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC,LNCS, Asiacrypt*, Vol. 3329, pp. 16-31, 2004.

[20] K. Yasuda, *"A New Variant of PMAC: Beyond the Birthday Bound"*, *LNCS, Crypto*, Vol. 6841, pp. 596-609, 2011.

[21] Y. Naito, *"Full PRF-Secure Message Authentication Code Based on Tweakable Block Cipher"*, *LNCS, Provsec*, Vol. 9451, pp. 167-182, 2015.

[22] M. Bellare, P. Rogaway, *"The security of triple encryption and a framework for code-based game-playing proofs," LNCS, EUROCRYPT*, vol. 4004, pp. 409-426, 2006.

[23] X. Wang, X. Lai, D. Feng, H. Chen, X. Yu, *"Cryptanalysis of the Hash Functions MD4 and RIPEMD," LNCS, EUROCRYPT*, vol. 3494, pp. 1-18, 2005.

[24] X. Wang, X. Lai, X. Yu, *"Finding Collisions in the Full SHA-1.," CRYPTO*, vol. 3621, 2005.

[25] E. Fleischmann, C. Forler, S. Lucks, J. Wenzel, *"Weimar-DM: A Highly Secure Double-Length Compression Function," LNCS, ACISP*, vol. 7372, pp. 152-165, 2012.

[26] O. Ozen, M. Stam, *"Another Glance at Double-Length Hashing," LNCS, Cryptography and Coding*, vol. 5291, pp. 176-201, 2009.

[27] X. Lai, X. Massey, L. J., "Hash function based on block ciphers," *LNCS, EUROCRYPT*, vol. 658, pp. 55-70, 1993.

[28] J. Lee, D. Kwon, *"The Security of Abreast-DM in the Ideal Cipher Model," IEICE Transactions*, vol. 94-A(1), pp. 104-109, 2011.

[29] J. Lee, M. Stam, J. Steinberger, "The Collision Security of Tandem-DM in the Ideal Cipher Model," *LNCS, CRYPTO*, vol. 6841, pp. 561-577, 2011.

[30] E. Brier, T. Peyrin and J. Stern: *"BPS: a Format-Preserving Encryption Proposal"*, "`http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/bps/bps-spec.pdf`"

[31] Black, J.A., Rogaway, P.: *"Ciphers with Arbitrary Finite Domains.*, In: Preneel, B. (ed.) CT-RSA, vol. 2271, pp. 114-130. Springer, 2002

[32] Morris, B., Rogaway, P., Stegers, T.: *"How to Encipher Messages on a Small Domain: Deterministic Encryption and the Thorp Shuffle*, In: Halevi, S. (ed.) CRYPTO, LNCS, vol. 5677, pp. 286-302. Springer, 2009

[33] Ristenpart, T., Yilek, S.: *"The Mix-and-Cut Shuffle: Small-Domain Encryption Secure against N Queries*, In: Canetti, R., Garay, J.A. (eds.) CRYPTO, Part I. LNCS, vol. 8042, pp. 392-409. Springer, 2013

[34] Hoang, V.T., Morris, B., Rogaway, P.: *"An Enciphering Scheme Based on a Card Shuffle*, In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO, LNCS, vol. 7417, pp. 1-13. Springer, 2012

[35] B. Morris and P. Rogaway: *"Sometimes-Recurse Shuffle Almost-Random Permutations in Logarithmic Expected Time," LNCS, Eurocrypt*, vol. 8441, pp 311-326, 2014.

[36] S. Dara and S. Fluhrer: *"FNR: Arbitrary Length Small Domain Block Cipher Proposal,"* LNCS, SPACE, vol. 8804, pg. 146-154, 2014.

[37] J. Lee, K. Kapitanova, S. H. Son *"The price of security in wireless sensor networks," ELSEVIER, Computer Network*, vol. 54, no. 17, pp. 2967-2978, December 2010.

[38] J. Lee, M. Stam, *"MJH: A Faster Alternative to MDC-2," CT-RSA*, vol. 6558, 213-236, 2011.

[39] S. Hirose, *"Some Plausible Constructions of Double-Block-Length Hash Functions," LNCS, FSE*, vol. 4047, pp. 210-225, 2006.

[40] J. Y. Lee, Y. H. Huang, *"A lightweight authentication protocol for Internet of Things"*, *IEEE explore, ISNE*, pp. 1-2, 2014.

[41] G. Kenneth, P. A. Yau, "Padding Oracle Attacks on the ISO CBC Mode Encryption Standard," *LNCS, CT-RSA*, vol. 2964, pages 305-323, 2004.

[42] V. T. Hoang, R. Reyhanitabar, P. Rogaway, V. Damian, *"Online Authenticated-Encryption and its Nonce-Reuse Misuse-Resistance," LNCS, Crypto*, vol. 9215, pp. 493-517, 2015.

[43] T. Iwata , K. Minematsu, J. Guo, S. Morioka, *"CLOC: Authenticated Encryption for Short Input," LNCS, FSE*, vol. 8540, pp. 149-167, 2015.

[44] D. Che, M. Safran, Z. Peng, *"From Big Data to Big Data Mining: Challenges, Issues, and Opportunities," LNCS, DASFAA Workshops*, vol. 7827, pp. 1-15, 2013

[45] P. Rogaway , *"Evaluation of Some Blockcipher Modes of Operation," http://web.cs.ucdavis.edu/ rogaway/papers/modes.pdf*, 2011.

[46] T. Iwata , K. Minematsu, J. Guo, S. Morioka, E. Kobayashi *"SILC: SImple Lightweight CFB," DIAC Competitions*, *https://competitions.cr.yp.to/round2/silcv2.pdf.*

[47] K. Minematsu, *"Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions," LNCS, Eurocrypt*, vol. 8441, pp. 275-292, 2014.

[48] D. Chang, S. R. Manjunath, S. K. Sanadhya, *"PPAE: Practical Parazoa Authenticated Encryption Family," LNCS, Provsec*, vol. 9451, pp. 198-211, 2015.

[49] R. Mazumder, A. Miyaji and C. Su, "A Simple Authentication Encryption Scheme," Concurrency and Computation: Practice and Experience, Wiley Publishers, DOI: 10.1002/cpe.4058, pp. 1-10, 2016

[50] R. Mazumder, A. Miyaji, C. Su: "A Blockcipher based Authentication Encryption," 4th International Cross-Domain Conference on Availability, Reliability and Security in Information Systems (CD-ARES), LNCS, vol. 9817, pp.106-123, 2016

[51] R. Mazumder, A. Miyaji and C. Su, "A Simple Construction of Encryption for a Tiny Domain Message," 51st Annual Conference on Information Sciences and Systems (CISS), IEEE, Accepted, pp , 2017

[52] R. Mazumder, A. Miyaji and C. Su, "Probably Secure Keyed-Function based Authenticated Encryption Schemes for Big Data," submitted to Special Issue in International Journal of Foundation of Computer Science (February 2017), Accepted.

[53] A. Miyaji, R. Mazumder, *"A new (n, 2n) Double Block Length Hash Function based on Single Key Scheduling," IEEE explore, AINA*, pp. 564-570, 2015.

[54] R. Mazumder, A. Miyaji, *"A New Scheme of Blockcipher Hash"*, *IEICE Transactions*, Vol. 99-D (4), 2016.

[55] A. Miyaji, R. Mazumder, T. Sawada *"A New (n, n) Blockcipher Hash Function: Apposite for Short Messages"*, *IEEE Explore, AsiaJCIS*, pp. 56-63, 2014.

[56] R. Mazumder, A. Miyaji, *"A Single Key Scheduling based Compression Function"*, *LNCS, CRiSIS*, pp. 207-222, vol. 9572, 2015.

[57] J. Chen, R. Mazumder, A. Miyaji and C. Su, "Variable message encryption through blockcipher compression function," Concurrency and Computation: Practice and Experience, Wiley Publishers, DOI: 10.1002/cpe.3956, pp. 1-10, 2016

[58] F. Armknecht, E. Fleischmann, M. Krause, J. Lee, M. Stam, J. Steinberger, *"The Preimage Security of Double-Block-Length Compression Functions," LNCS, ASIACRYPT*, vol. 7073, pp. 233-251, 2011.

[59] B. Mennink, *"Optimal Collision Security in Double Block Length Hashing with Single Length Key," LNCS, ASIACRYPT*, vol. 7658, pp. 526-543, 2012.

[60] J. A. Black, P. Rogaway, T. Shrimpton, *"Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV," LNCS, CRYPTO*, vol. 2442, pp. 320-335, 2002.

[61] J. A. Black, P. Rogaway, T. Shrimpton, M. Stam, *"An Analysis of the Block cipher-Based Hash Functions from PGV," LNCS, J.CRYPTOL*, vol. 23, pp. 519-545, 2010.

[62] S. Hirose, H. Kuwakado., *"Collision Resistance of Hash Functions in a Weak Ideal Cipher Model," IEICE Transactions*, vol. 95 A(1), pp. 251-255, 2012.

[63] M. Liscov, *"Constructing an ideal hash function from weak ideal compression function," LNCS, SAC*, vol. 4356, pp. 358-375, 2006.

[64] M. Nandi, W. Lee, K. Sakurai, S. Lee, *"Security Analysis of a 2/3-Rate Double Length Compression Function in the Black-Box Model," LNCS, FSE*, vol. 3557, pp. 243-254, 2005.

[65] J. Lee, S. Hong, J. Sung, H. Park, *"A New Double-Block-Length Hash Function Using Feistel Structure," LNCS, ISA*, vol. 5576, pp. 11-20, 2009.

[66] F. Abed, C. Forler, E. List, S. Lucks, J. Weznel *"Counter-b DM: A Provably Secure Family of Multi-Block-Length Compression Functions," LNCS*, vol. 8469, pp. 440-458, 2014.

[67] D. Yevgeniy, P. Prashant, *"On the Relation Between the Ideal Cipher and the Random Oracle Models," LNCS, Theory of Cryptography*, vol. 3876, pp. 184-206, 2006.

[68] H. Kuwakado, S. Hirose, *"Hashing Mode Using a Lightweight Blockcipher*, *LNCS, Cryptography and Coding"*, vol. 8308, pp. 213-231, 2013.

[69] L. R. Knudsen, F. Mendel, C. Rechberger, S. S. Thomsen, *"Cryptanalysis of MDC-2"*, *LNCS, Eurocrypt*, Vol. 5479, pp. 106-120, 2009.

[70] E. Fleischmann, C. Forler, and S. Lucks *"The Collision Security of MDC-4*, *LNCS, Africacrypt*, vol. 7374, pp. 252-269, 2012.

[71] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, *"Internet of Things for Smart Cities*, *IEEE Internet of Things Journal*, volume-1, issue 1, pp. 22 - 32, 2014.

[72] L. D. Xu, W. He, S. Li, *"Internet of Things in Industries: A Survey*, *IEEE Transactions on Industrial Informatics*, volume-10, issue 4, pp. 2233 - 2243, 2014.

[73] J. S. Coron, Y. Dodis, E. List, S. Lucks, J. Weznel, *"Merkle-Damgard revisited: How to construct a hash function," LNCS, Crypto*, vol. 3621, pp. 430-448, 2005.

[74] D. Joan, R. Vincent, *"The Design of Rijndael, AES-The Advanced Encryption Standard"*, ISBN 978-3-662-04722-4, Springer Press, 2002.

[75] A. K. L. Yau, K. G. Paterson, C. J. Mitchell, "Padding Oracle Attacks on CBC-Mode Encryption with Secret and Random IVs," *LNCS, FSE*, vol. 3557, pp. 299-317, 2005.

[76] T. Lee, J. Kim, C. Lee, J. Sung, S. Lee, D. Hong, "Padding Oracle Attacks on Multiple Modes of Operation," *LNCS, ICISC*, vol. 3506, pages 343-351, 2004.

[77] E. Andreeva, A. Bogdanov, A. Luykx, B. Mennink, E. Tischhauser, K. Yasuda, *"Parallelizable and Authenticated Online Ciphers"*, *LNCS, Asiacrypt*, vol. 8269, pp. 424-443, 2013.

[78] E. Andreeva, B. Bilgin, A. Bogdanov, A. Luykx, B. Mennink, N. Mouha, K. Yasuda, *"APE: Authenticated Permutation-Based Encryption for Lightweight Cryptography"*, *LNCS, FSE*, vol. 8540, pp. 168-186, 2014.

[79] J. W. Bos, O. Ozen, M. Stam, *"Efficient Hashing Using the AES Instruction Set"*, *LNCS, CHES*, vol. 6917, pp. 507-522, 2011.

[80] S. Hirose, K. Ideguchi, H. Kuwakado, T. Owada, B. Preneel, H. Yoshida, *"A Lightweight 256-Bit Hash Function for Hardware and Low-End Devices: Lesamnta-LW"*, *LNCS, ICISC*, vol. 6829, pp. 151-168, 2010.

[81] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, T. Iwata *"The 128-bit Blockcipher CLEFIA"*, *IACR archive, Extended Abstract*, https://www.iacr.org/archive/fse2007/45930182/45930182.pdf

[82] L. Barreto, A. Celesti, M. Villari, M. Fazio, A. Puliafito, *"An Authentication Model for IoT Clouds"*, *IEEE explore, ASONAM*, pp. 1032-1035, 2015.

[83] D. V. Bailey, J. Brainard, S. Rohde, C. Paar, *"Wireless Authentication and Transaction-Confirmation Token*, *Springer, ICETE*, vol. CCIS 130, pp. 186-198, 2011.

[84] L. Atzori, A. Iera, G. Morabito, *"The Internet of Things: A survey*, *ELSEVIER, Computer Networks*, vol. 54, issue. 15, pp. 27872805, 2010

[85] Z. Zhou1, K. F. Tsang, Z. Zhao, W. Gaalou, "Data intelligence on the Internet of Things," *Springer, Pers Ubiquit Comput*, DOI 10.1007/s00779-016-0912-1, 2016

[86] P. Coppola, V. D. Mea, L. D. Gaspero, R. Lomuscio, D. Mischis, S. Mizzaro, E. Nazzi, I. Scagnetto, L. Vassena, *"AI Techniques in a Context-Aware Ubiquitous Environment," Springer, Computer Communications and Networks*, pp 157-180, 2009.

[87] K. Zhao, L. Ge, *"A Survey on the Internet of Things Security," IEEE explore, 9th CIS*, 978-1-4799-2548-3, pp. 663-667, 2013.

[88] B. Mennink, *"Embedded Security for Internet of Things ," IEEE explore, 2nd NCETACS*,978-1-4244-9578-8, pp. 1-6, 2011.

[89] D. Burak *"Parallelization of a Block Cipher Based on Chaotic Neural Networks"*, *LNAI, ICAISC*, pp. 192-201, 2015.

[90] L. Adrienne, *World War I, Espionage Information: Encyclopedia of Espionage, Intelligence, and Security*, *Advameg, Inc.* Retrieved 2015.

[91] Cohen, Fred. *A Short History of Cryptography.*, *http://all.net/edu/curr/ip/Chap2-1.html*, 1995.

[92] E. Andreeva, A. Luykx, B. Mennink, K. Yasuda, *"COBRA: A Parallelizable Authenticated Online Cipher Without Block Cipher Inverse"*, *LNCS, FSE*, vol. 8540, pp. 187-204, 2014.

[93] N. Datta, M. Nandi, *"ELmE: A Misuse Resistant Parallel Authenticated Encryption"*, *ACISP, LNCS*, Volume: 8544, pp. 306-321, 2014.

[94] M. Bellare, P. Rogaway, and D. Wagner, *"The EAX Mode of Operation"*, *LNCS, FSE*, vol. 3017, pp. 389-407, 2004.

[95] D. Gligoroski, H. Mihajloska, S. Samardjiska, H. Jacobsen, R. E. Jensen, and M. El-Hadedy, *"π Cipher: Authenticated Encryption for Big Data"*, *LNCS, NordSec*, vol. 8788, pp. 110-128, 2014.

[96] M. Naor, O. Reingold, *On Construction of Pseudorandom Permutations: LubyRackoff Revisited*, *J, CRYPTOLOGY*, Vol.12, pg.29-66, 1999.

[97] Maurer, U., Pietrzak, K.: *The security of many-round Luby-Rackoff pseudo-random permutations*, *Eurocrypt*, Vol.2656, pp.544-561, 2003.

Publications

LIST OF INTERNATIONAL JOURNALS

[1] R. Mazumder, A. Miyaji: "A New Scheme of Blockcipher Hash," IEICE Trans., Information and Systems. Vol. E99-D, No.4, pp. 796-804 (2016).

[2] J. Chen, R. Mazumder, A. Miyaji and C. Su: "Variable Message Encryption through Blockcipher Compression Function," Concurrency and Computation: Practice and Experience, Wiley Publishers, DOI: 10.1002/cpe.3956, Vol. 29, Issue. 7, pp. 1-10 (2016).

[3] R. Mazumder, A. Miyaji and C. Su: "Probably Secure Keyed-Function based Authenticated Encryption Schemes for Big Data," Submitted to Special Issue in International Journal of Foundation of Computer Science, World Scientific Publishers, (February 2017) [Accepted].

[4] R. Mazumder, A. Miyaji and C. Su: "A Simple Authentication Encryption Scheme," Concurrency and Computation: Practice and Experience, Wiley Publishers, DOI: 10.1002/cpe.4058, pp. 1-10 (2017).

LIST OF INTERNATIONAL CONFERENCES

[5] R. Mazumder, A. Miyaji: "A new (n, 2n) Double Block Length Hash Function based on Single Key Scheduling," 29th IEEE International Conference on Advanced Information Networking and Applications (AINA), IEEE, pp. 564-570 (2015). [Gwangju, South Korea]

[6] J. Chen, R. Mazumder, A. Miyaji: "A Single Key Scheduling Based Compression Function," 10th International Conference on Risks and Security of Internet and Systems (CRiSIS), Lecture Notes in Computer Science, 9572, Springer-Verlag, pp. 207-222 (2015). [Lesvos Island, Greece]

[7] R. Mazumder, A. Miyaji, C. Su: "An Efficient Construction of a Compression Function for Cryptographic Hash," 4th International Cross-Domain Conference on Availability, Reliability and Security in Information Systems (CD-ARES).,Lecture Notes in Computer Science, 9817, Springer-Verlag, pp. 124-140 (2016). [Salzburg, Austria]

[8] R. Mazumder, A. Miyaji and C. Su: "A Simple Construction of Encryption for a Tiny Domain Message," 51st Annual Conference on Information Sciences and

Systems (CISS), IEEE, DOI: 10.1109/CISS.2017.7926080, pp. 1-6, (2017). [John Hopkins University, Baltimore, USA]

[9] R. Mazumder, A. Miyaji and C. Su: "A Re-visited Construction of Nonce and Associated-data based Authenticated Encryption," US-Japan Workshop on Collaborative Global Research on Applying Information Technology under 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017), Accepted, (2017). [Atlanta, GA, USA]