ISSN 2047-3338

# Fast and Secure Generating and Exchanging a Symmetric Key for TVWS Protocol

Mubark M. A. Elmubark
Department of MIS Blue Nile University, Sudan
MubarkElmubark@gmail.com

*Abstract*– **In network security, symmetric key encryption methods are commonly used in order to generate and exchange a secret key between the sender and the receiver. The main drawbacks of this method are that, the attackers might access the transmitted data and use it to obtain the key and even generate new keys. Also, the process of generating the key is inefficient and time consuming. In this paper a new key generation and exchange protocol is proposed to overcome the aforementioned problems in the conventional symmetric key encryption methods. In the proposed protocol, the results show that the protocol is saved.**

*Index Terms*– **Security, Key Generation and Key Exchange**

## I. INTRODUCTION

KEY generation and exchange is considered as one of the most critical issues in network security. In network security asymmetric or symmetric key is used for authentication purposes. Asymmetric key uses both public and private key schemes, which makes it reliable and secure. However, the process of encryption and decryption requires high time consumption. Therefore, this method is impractical for some applications especially real time applications. In case of the symmetric key, the sender and the receiver share a single key. Thus, it is important to handle and exchange this security key carefully.

## II. BACKGROUND STUDY

Due to moving from analog TV transmission to digital transmission, there will be free frequencies called TV White Space (TVWS). These frequencies can be reused in broadband communication. TVWS can be licensed by auction or freely unlicensed, which is preferred by many parties around the world. There are two types of TVWS access, sensing by using cognitive radio and geolocation database [1], TVWS unlicensed can use Cognitive Radio (CR) techniques for sharing the spectrum. Many researches and standard efforts has been given to TVWS techniques and other related issues like security, frequency allocation, interference, database management, throughput, etc. the main problem in symmetric key method is how to exchange the security key. So many security protocols can be used in TVWS with pros and cons.

This paper concentrated on security issues in spectrum database access and studies these protocols and proposed new methods for key generation and exchange.

*PKMv1:* WiMAX security, as specified by IEEE 802.16 standard [2] is provided by a security sublayer resided in the MAC layer. A protocol called PKM has been adopted in the security sublayer of WiMAX to provide authorization, authentication and keys exchange and distribution between Base Stations (BSs) and Subscriber Stations (SSs).

The standard has approved two versions of PKM protocol. PKMv1 which is approved in IEEE 802.16-2004 standards provides one way authentication; however it was vulnerable to attacks such as replay attack, Man-In- The-Middle (MITM) attack, and Denial-Of-Service (DoS) attacks.
PKMv1 has been modified by adding a random number called nonce to the messages exchanged between BS and SS to prevent such attacks, nevertheless possibility of attacks still exists. The Authentication and Authorization procedure is shown in Fig. 1 [3].

**1、Auth Info**

***Message1: SS→ BS: Cert(Manufacturer)***

**2、Auth Request**

***Message2: SS→ BS: C ert(SS) | Capabilities |SAID***

**3、Auth Reply**

***Message3: BS→ SS: RSA-Encrypt(AK)PubKey(SS) | Lifetim e | AK_SN | SAIDList***

Fig. 1. PKMv1 Authentication & Authorization Procedure

*PKMv2:* To overcome the problems in PKMv1 the (IEEE 802.16-2005, 2009) standards have approved PKMv2 which uses either RSA based or EAP based authentication modes [3]. One of the key advantages over its predecessor is it provides mutual authentication, where the identity of BS and SS can be authenticated by each other.

Following the authentication phase, authenticated Mobile Subscriber (MS) will get Pairwise Master Key (PMK) from Authenticated Server (AS), which is used to generate Pairwise Transient Key (PTK) [6]. This PTK is used to install a key at the MC and AS, and is also used to generate the Group Temporal Key (GTK). The GTK will be used by both supplicant (MS) and Authenticator (AP) to encrypt all messages exchanged between each other or between other MS’s.

*TEK exchange procedure in PKMv2:* The creation of Traffic Encryption Keys (TEK) is similar to the steps in PKMv1 in PKMv2. They are also used in similar fashion for encrypting traffic [4 p.279]. Key Encryption Key is used for securing the exchange of TEK, GTEK and GKEK. This is derived from AK using same method as corresponding MAC keys as shown in figure 2. KEK is a 128-bit long key if SA is using ciphersuite which uses 128 bits as a basic block size, otherwise the length of a KEK is 64-bits. For encrypting multicast message GTEKs there are Group Key Encryption Keys (GKEK) which are randomly generated at the BS and sent to SS after encrypting it with KEK. For each Group SAs there is a single GKEK.

When the SS achieves authorization, a separate TEK state machine is started for each Security Association IDs identified in the Authorization Reply or SA-TEK Response [4p.274]. TEK state machine (one or more) is started if data encryption is provisioned for one or more service flows. Each TEK state machine manages the keying material associated with its respective SAID by sending Key Request messages to BS. The TEK key exchange is shown in figure 3. BS uses random or pseudo-random number generator for generating TEK keys [4 p.300]. SS is responsible for keeping its copies of SAID's TEKs synchronized with the BS it is connected to. TEK state machine remains active as long as SS has a valid AK and the BS provides continuously fresh keying material when requested [4 p.274, 275].For each SAID there are two sets of keying material which have overlapping lifetimes [4p.274]. Similarly to AKs, every TEK has a certain lifetime and configurable Grace Time, which is known only by SS [4 p.292]. The new Key Request is scheduled at the beginning of Grace Time period, before the latest TEK is scheduled to expire. The Grace Time period provides to the SS sufficiently long period of time to successfully complete the traffic keying material exchange process in spite of system delays [4 p.292]. The TEK grace time can vary from 5 minutes to 3.5 days and the default value is 1 hour [4 p.641].

*IPSec:* The IPSec system is a set of protocols that facilitate the creation and maintenance of secure IP channels called Security Associations (SAs). This is optional for IPv4, but it is an integral component of IPv6. IPSec consists of three main protocols; Authentication Header (AH), Encapsulating Security Protocol (ESP), and Internet Key Exchange (IKE). Consequently, high speed key exchange is a fundamental requirement in order to support IPSec for applications requiring high speed connections [5]. IPSec is based on a key exchange protocol to make an automatic establishment IKEv2 of security associations (SA). Each SA is maintained between two or more entities which describe the algorithms, keys and other security parameters to be used. To maintain a SA, two phases are required by IKEv2. Phase 1 performs mutual authentication between two parts and establishes an IKE_SA, whereas Phase 2 executes the creation of IPSec_SA between the same pairs. This presents challenges for the implementation of IKEv2 on wireless environments by considering the processor cost and bandwidth limitation. So, there is need to develop a lightweight IKE which can be easily deployed in the target network while maintaining security properties [4].

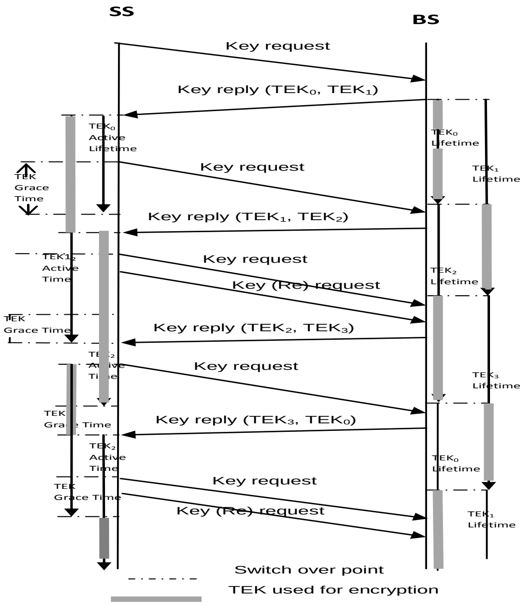


Fig. 2: TEK key management [2]

## III. RELATED WORK

### A) Problem Statement

*Problem1:* The data sends - during the authentication protocol- has a relation with the key generation (like nonce), and the attackers can use this information to generate the key. In [7], Hasan et al., proposed a new method, they used the self-organizing map (SOM) to generate common secret keys in both sides of a communication channel instead of exchanging them over such a public communication channels. They consider this method allows changing the key frequently, easily and safely. They presented an application of the self organizing map and the Kohonen algorithm to solve the problem of exchange cryptography keys securely. This method generates a large numbers of keys which are the same number of neurons in the map. They discussed two methods to generate a key in both sides of the communication channel; the fake key and the key index methods. And they specify that using the second method - key index methods-leads to reduce the length of the message several bits. Lastly this paper also introduced a mask function allowing the association of several neurons within the key computation. The mask enhances the security of the keys and allows the use of the same map with different mask function to exchange messages between several communicators.

However, this method highly depends on the selected type of SOM as well as the training method additionally; the proposed method requires time consuming hardware implementation.

*Problem2:* The second problem is the time to generate the key is too long.

Three-party Encrypted Key Exchange Protocol -(3PEKE) this protocol provided two main indexes for describing the performance of the key exchanging system. One is 'transmission round' and the second one is 'computation complexity' [8], [9], [10]. A transmission round includes all independent steps that must be done in sending and receiving the data during the same time interval. The computation complexity represents how heavy cryptographic operations such as symmetric encryption or one-way hash function are adopted in the protocol [6], [8], [9], [10]. Therefore, a 3PEKE can be operated in less time with the help of transmitting the messages in parallel or adopting fewer cryptographic operations [6], [8], [9], [10]. As the result of the 3PEKE protocol was simplicity and convenience, so that so many investigations [11]-[15] have been focused on the 3PEKE research.

Thus, Lin *et al.* [9], [10] proposed an improved 3PEKE protocol (LSH-3PEKE) using server's public key to prevent the attackers. Nevertheless, the public key technologies need to take more computation times if it is applied to 3PEKE protocol.

Subsequently, in 2004, Chang et al. [11] proposed the three-party encrypted key exchange (ECC-3PEKE) protocol which is quite different from the protocols [9, 16]. Their proposed scheme was without using the server's public keys. However, they claimed that their proposed ECC-3PEKE scheme is secure, efficient, and practical.

Hsing-Chung Chen el [17] proposed an improved approach which is applied to the time bound in the 3PEKE protocol. By using time bound approach, they can effectively protect all clients' secret passwords and stored it in the server –client's passwords. More importantly, they enhanced the security of the traditional 3PEKE schemes [11-15] but not affecting the protocol efficiency. But this enhancement method in the password protocol and using public key takes much time to generate the password which is consider as a delay.

The authors of [21](modified) NIST P-256 curve and key generation algorithm by using split exponents for fast exponential and implementation to speed up and increase the randomness of key generation. Algorithm key agreement scheme is employed with smaller key sizes resulting in faster computations. Fast Exponentiation is achieved by using split exponents. Split exponents are used for more efficient implementation of cryptographic models based on discrete logarithm. Splitting of exponents leads to a higher order of randomness while generating the keys. The increased randomness of the keys makes it more difficult for the attacker to obtain the secret key [22]. However, this method depends on the public key generation and the key generation and encryption/decryption consumes a huge time. Additionally, this method also uses small size key which will be less secure. Patrick Moore [20] this study presents a scalable and flexible multi-core SOC architecture for high-speed key exchange for emerging IP security systems. Novel approaches are proposed for HMAC authentication block parallelization, distributed key handling and a pipelined block cipher design that allows feedback encryption modes. This improves upon previous state-of-the-art designs for IPSec, creating architecture suitable for delivering emerging secure high-speed streaming applications via the Internet.

As a packet enters the system, the keys are retrieved from the central Security Association Database and are tagged with a unique identifier for the packet before being sent to a local look-up for each cryptographic block. Because the keys for each cryptographic operation are different, and there may be more than one packet from each stream in the system at any time, it would be inefficient in terms of hardware resources to associate copies of the keys with every packet. To let the system to be more efficient the used the method of allow the relevant keys to be available only as required, rather than unnecessarily duplicating them.

*Problem3:* The third problem is when the attacker knows the key, they may be able to generate more keys once the life time of the existing key is expired.

Most of the security protocols consider the security to be vulnerable when the key is broken consequently, that particular key will be disabled. However, it is important to ensure that the attackers are unable to use that key to generate more keys. T. Subashri *et al.* [18] describes a method to generate a numbers of random keys, but they didn't explain a method to change the key in case of the attacker break the key.

## IV. PROPOSED KEY GENERATION AND EXCHANGE PROTOCOL

To overcome the above mentioned problems this section presents the proposed key generation protocol. The proposed protocol aims to generate a reliable keys based on mathematical calculations and in two stages.

*Stage one the pre-establish protocol*

The key is a combination of binary bits; therefore, in this stage the Master and the Database exchanges a message (string of bits) of length (m bits) suppose 2 Mega bits, and then agree about the length (L) of the key (like the pre-share key) and as an optional agreement they can fix the key life time and the grace time. This stage must be run outside of the protocol.

*Stage two key generation and exchange*

When the Master wants to generate a key he has to follow these steps:

1- Generate a random number (N) with length size less than the message size.

2- Setup the length of the key (L) and the key life time (T) and the grace time G.

3- Generate a changing message request ChngMessReq (N, L, T and G).

4- Send the ChngMessReq (N, L, T, and G) to the Database server.

5- Before the grace time is finished the Master should go back to step1 or he will need new authentication to start this protocol.

When the Database server receives this message starts the following process:

1- Count the N bits from the message (m) and then,

2- Starts calculate the L numbers of bits to be the key.

3- Conceder T to be the key life time and G to be the grace time.

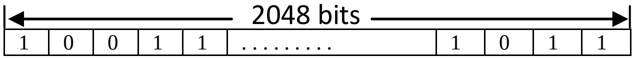
In this protocol the key generation will not take time because the process is very simple and quick.

### A) An example of this protocol

*Stage one pre-establish protocol*

Suppose that the Master and the Database server generate and exchange the message (m) with the length is equal 2 mega bits, and they agreed that the key size is equal 64 bits and the key life time T is 12 hours and the grace time is 30 minutes. Like this

m =



K= 64 bits, T = 12 hours, G = 30 minutes

*Stage tow*

1-The Master generate a random number (N) suppose that N=155 (less than 2048)

2- Fixed length of the key (L = 64) and the key life time (T = 12 hours = 720 minutes) and the grace time G = 30 minutes.

3- Generate an exchanging message request ExchngMessReq (155, 64, 720 and 30).

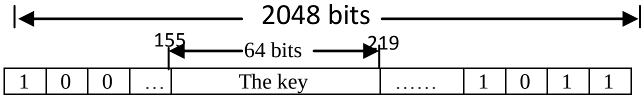
4- Send the ExchngMessReq to the Database server.

When the Database receive the ExchngMessReq (155, 64, 720,30)

1- Shift 155 bit from the left side of the message (m) and the then starts calculate the next 64 bits to be the Key.

2- Conceder the key life time as 720 minutes and the key life time is 30 minutes.

Before the grace time is finish the master must generate a new ExchngMessReq request and generate and exchange a new key. If the grace time is finish then the Master must re-authenticate himself to the Database server which is out scope of this protocol.



### B) Protocol analysis

This protocol can be broken under one difficult condition, when the attackers can get both the message m and the ExchngMessReq and this very difficult because the message m will never transmit in the network or uses during the protocol, so the attacker need to get inside the database server or the master data to get this message. But the protection of the data inside the devices is outside of the protocol scope. If the attacker can break the ExchngMessReq he can't drive the (m) message because there is no relation or formula to calculate m.

Now suppose the attacker can break the key (by luck or any way) he can use this key if and only if the life time is not finished yet and the master is not working during this period of time. In the first case (life time not finished) the attacker won't know the life time and won't know how long he can use this key. And even if he knew the key life time he can't do any things about this and he cannot generate another key. For the second case (the master is not working) when the master is working the database server will receive tow messages in the same time with different properties and the server will suspect there is an attacker , so the database server will response with changing- key- request message and will conceder this key is not valid.

## V. CONCLUSION

Generate and exchange a key between the sender and receiver is the most critical issue in network security especially in TVWS database. In this paper we specify three problems, the sender always sends an information (data) during the authentication procedure and the attacker can use it to get the key, the second problem is the time to generate the key is very long, and lastly when the attackers get the key they can generate a new key after the recent key's life time is expired and continue working. This study proposed a designed and analysis a new protocol to generate the key and exchange it and a void these problems. The HLPSL simulation results show that this protocol is saved.

## REFERENCES

[1]. Mubark et. al., "High secure Mutual Authentication Protocol for TVWS Database", IJISET, Vol. 5, Issue 4, April 2018.

[2]. IEEE 802.16-2004. IEEE standard for local and metropolitan area networks Part l6: air interface for fixed broadband wireless access systems. IEEE Press, 2004.

[3]. Fan Yang, "Comparative Analysis on TEK Exchange between PKMv1 and PKMv2 for WiMAX", IEEE-201.

[4]. IEEE 802.16-2005e Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile.

[5]. Arto Katajasalo and Jussi Laakkonen, "Key management and authorization protocols in WiMAX", Master Thesis, Lappeenranta University of Technology 2013.

[6]. Mohamed KASRAOUI et. al., "Formal Verification of Wireless Sensor Key Exchange Protocol using AVISPA", 2014 International Symposium on Computer, Consumer and Control, IEEE 2014.

[7]. Hasan Abdulkader, Daniel Roviras, "Generating Cryptography Keys Using Self-Organizing Maps", IEEE, 2012.

[8]. T.H. Chen, W.B. Lee, and H.B. Chen, "A round-and computation-efficient three-party authenticated key exchange protocol", Journal of Systems and Software, Vol. 81, Issue 9, pp. 1581-1590, Sep 2008.

[9]. H.S. Kim and J.Y. Choi, "Enhanced password-based simple three-party key exchange protocol," *Computers & Electrical Engineering*, Vol. 35, Issue 1, pp. 107-114, Jan 2009.

[10]. T.F. Lee, T. Hwang, and C.L. Lin, "Enhanced three-party encrypted key exchange without server public keys", Computers & Security, Vol. 23, Issue 7, pp. 571-577, Oct. 2004.

[11]. S.B. Wilson and A. Menezes, "Authenticated Diffie-Hellman key agreement protocols", *Proceeding of* the 5th Annual Workshop on Selected Areas in Cryptography, of LNCS, Vol. 1556, pp. 339-361, 1998.

[12]. H.C. Chen and H.Y. Chuang, "A Three-party Encrypted Key Exchange Protocol with Protected Password Authentication", The IET International Conference on Frontier Computing – Theory, Technologies and Applications, Taichung City, Taiwan, Aug. 4-6,2010.

[13]. C.C. Chang and Y.F. Chang, "A novel three-party encrypted key exchange protocol", *Computer Standards & Interfaces*, Vol. 26, Issue 5, pp. 471-476, Sep 2004.

[14]. H.B. Chen, T.H. Chen, W.B. Lee, and C.C. Chang, "Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks", *Computer Standards & Interfaces*, Vol. 30, Issue 1-2, pp. 95-99, Jan 2008.

[15]. T.H. Chen, W.B. Lee, and H.B. Chen, "A round-and computation-efficient three-party authenticated key exchange protocol", *Journal of Systems and Software*, Vol. 81, Issue 9, pp.1581-1590, Sep 2008.

[16]. H.R. Chung and W.C. Ku, "Three weaknesses in a simple three-party key exchange protocol", *Information Sciences*, Vol. 178, Issue 1, pp. 220-229, Jan 2008.

[17]. H. Guo, Z. Li, Y. Mu, and X. Zhang, "Cryptanalysis of simple three-party key exchange protocol", *Computers & Security*, Vol. 27, Issue 1-2, pp. 16-21, Mar 2008.

[18]. C.L. Lin, H.M. Sun, and T. Hwang, "Three-party encrypted key exchange: attacks and a solution", *ACM Operating Systems*, Vol. 34, pp.12-20, 2000.

[19]. Hsing-Chung Chen et. al., "A Time constraint Three-party Encrypted Key Exchange Protocol", International Conference on Broadband and Wireless Computing, Communication and Applications, 2011.

[20]. Patrick Moore et. al., "A High-Speed Key Exchange Multi-Core SoC Architecture for IPSec Real-Time Internet Traffic", globcom workshop no multimedia communication and service IEEE 2010.

[21]. T. Subashri et. al., "Real time implementation of elliptic curve cryptography over an open source VOIP server", 5th ICCCNT - July 11-13, 2014, Hefei, China.

[22]. Jung Hee Cheon, Stanislaw Jarecki, Taekyoung Kwon, and Mun-Kyu Lee, "Fast Exponentiation Using Split Exponents", IEEE transactions on information theory.