HMAC Modification Using New Random Key Generator

Dr. Shaimaa H. Shaker
Computer Engineering Department, University of Technology, Baghdad, Iraq
e-mail: shaimaa.sw@uotecnology.edu.iq

Received: 6/6/2013

Accepted: 20/2/2014

Abstract **–**Cryptographic hash functions have been very significant primitives to the cryptography. They have been utilized widely in cryptographic applications and the most important of which is their use in the composition of efficient Message Authentication Codes (MACs). Cryptanalysts cannot break the encryption code authorization message because they could not generate the message sets without key predictors.

This paper introduces anew way to achieve hash code authorization message to increase the strength of the code, reluctance to Birthday attack, and key's exhaustive search. Secret key with randomness propertyis generated with the assistance proposed algorithm called AKG (Algorithm of Key Generator) that will be hash function. AKG is used to generate a secret key with decimal digits instead of (0,1).The proposed AKG introduces a method that uses three inputs of (0..9994) states instead of (0,1) states.The output of AKG is a random key from key generator using 9995 tables. So the new method provides authentic and integral properties.

.

Keywords – Authentication, message integrity and confidentiality, hash Function, hash-based key derivation algorithms, random number generation

.

1. Introduction

The increasing use of messages increases the attack onthem, so sending private information without any measurement security or receiving messages not controlled by security policy is a big problem. The enhanced message security can give evidence that the received message is from the right originator and can check the message integrity [1]. This paper develops a secure algorithm based on authentication and hash function, which is used to enhance security features of messages.

A hash function is defined as a function which compresses an input sequence of variable lengths to an output sequence which is of fixed length. Cryptographic hash functions must possess properties like one-way and collision resistance. Hashing functions have always been significant primitives in cryptography and are commonly utilized in applications like digital signatures and e-commerce. One of the important areas of applications of cryptographic hashing functions is their utilization in the composition of proficient message authentication codes (MACs) [2], [3].

The design of HMAC specification was motivated by the existence of attacks on more trivial mechanisms for combing a key and a hash function. HMAC algorithm is mainly used to provide the message authentication and prevent the snooping attacks [4]. There are two fundamentally different strategies for generating random bits. One strategy is to produce bits non-deterministically, where every bit of output is based on a physical process that is unpredictable. This class of random bit generators (RBGs) is commonly known as non-deterministic random bit generators (NRBGs). The other strategy is to compute bits deterministically using an algorithm.This class of RBGs is known as Deterministic Random Bit Generators (DRBGs) [5]. The HMAC can be an implementation of any function like MD5, SHA-1. Naim et al., [6] presented a new algorithm that simplifies the process of generating and expanding cipher key. The algorithm generates a random pool of keys (long key size), these keys are sent to the authorized receiver. Murali and Palraj [7] proposed a new method for generating true random numbers based on image which generates 256 bits key or higher for key exchange algorithm. True random numbers are secured and good, compared to pseudo random numbers. Manish et al., [8] produceda method that encrypts the message, digests, and then again uses the previously available cryptographic algorithm and again encrypts the Message. The idea of confusion and diffusion makes message more secure. This produced MAC is more secure as it is encrypted by the DES method. Also, we can use any other cryptographic algorithm in place of the DES Algorithm in the proposed model.

This paper introduces a new algorithm of key generator that will be used in hash function as MAC. The paper proposed a key generator algorithm (AKG) to generate a key using it as a shared secret key and as secret string.

2. Hash Function

Messages may be exposed to many different attacks.Message authentication is a procedure to verify that the received messages has come from the alleged source and have not been altered. Hash function, shown in Figure (1), is one of the types of functions that may be used to produce an authentication. A hash function H transforms an arbitrarily long input string m to a fixed-size output sequence called hash value h== H(m) . It has the following major properties [9]:

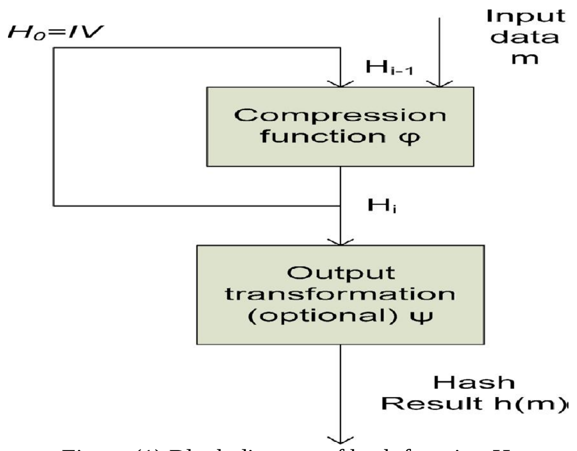

Figure (1) Block diagram of hash function H

1. The length of input string is variable,
2. The output string length is fixed,
3. It is quite easy to calculate H(m) for any known m,
4. H(m) is one-way,
5. H(m) is collision free.

The basic uses of the hash function are [10]:

1- Encryptsa message plus hash code.
   - Provides confidentiality only A and B share K.
   - H (M) is cryptographically protected.
2- Encrypts a hash code-shared secret key
   - Only hash code is encrypted, using symmetric encryption.
   - Reduces the process burden for those applications that do not require confidentiality.
3- Encrypt hash code sender's private key
   - Provides authentication and digital signature.

Message Authentication Code (MAC) is a sort of hashing function for which a secret key is used to assure authentication among the sender and the receiver. The security of the MAC generally depends on the bit length of the key. The weakness of the algorithm is the brute force attack [9].
Hash AuthenticationCode (HMAC) is a well-known cryptographic algorithm which is used for the assurance of data integrity and authentication as shown in Figure (2). HMAC is an explicit kind of MAC function (Message Authentication Code). It uses a hash function operating on an input string and the key. Presently, among the prime techniques in use for the certification, that message has not been distorted or tailored by any forgery during transmission over unprotected links [11].

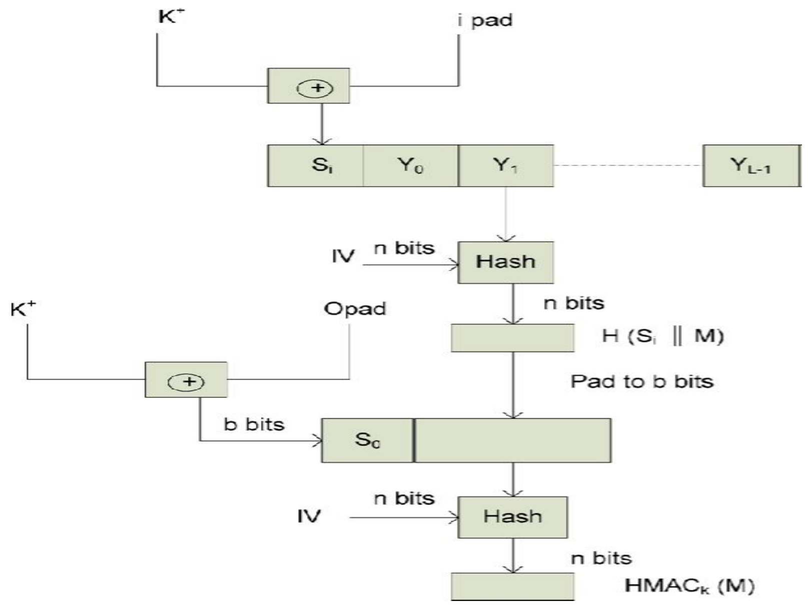

Figure (2) Hash Message Authentication Code (HMAC) function

3. Hashing Proposal for Message Authentication using AKG

The approach to the use of hashing for authentication is shown in Figure (3). In this scheme, nothing is encrypted. However, the sender appends a secret string S (known also to the receiver) to the message before computing its hash code.
Before checking the hash code of the received message for its authentication, the receiver appends the same secret string S to the message. Obviously, it would not be possible for anyone to alter such a message, even when they have access to both the original message and the overall hash code [10]. This paper modifies HMAC by using new Algorithm of Key Generator (AKG) which is used to generate asecret key with decimalsystem instead of (0,1),as shown in Figure (4).

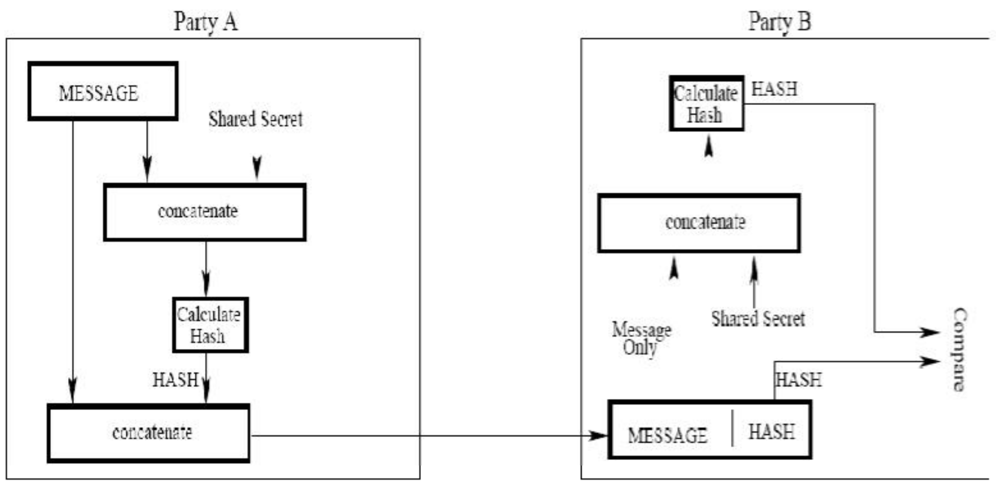

Figure (3) Hashing for message authentication

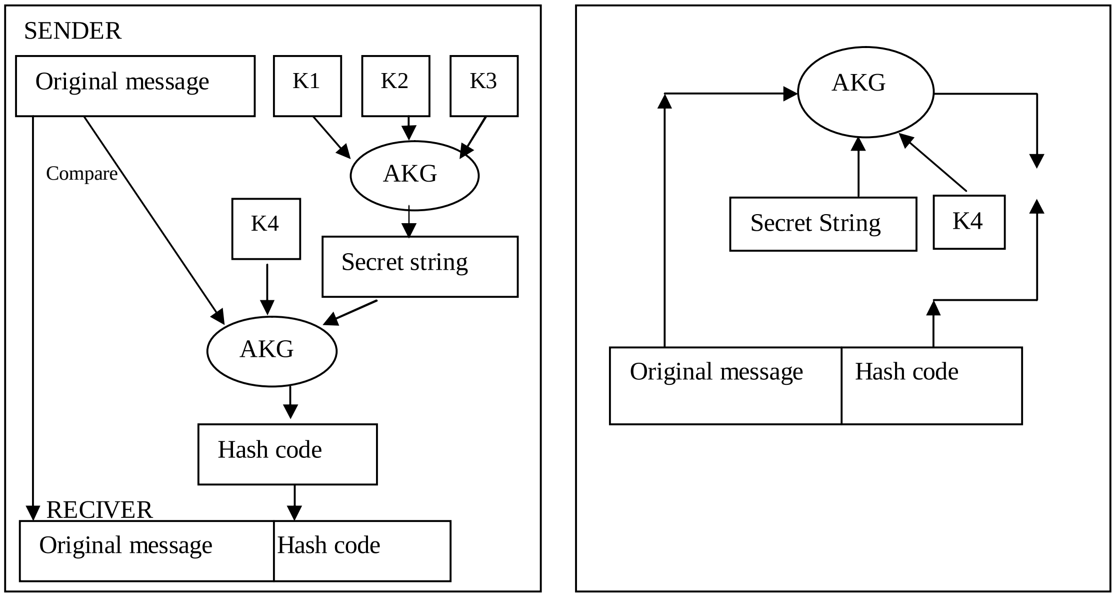

Figure (4) Hashing modification for message authentication using AKG

AKG uses some tables to detect the digits of random keys. These tables will be implemented using the algorithm shown as Algorithm (1). Thus, there are 9995 tables each one consists of 9995*9995 cells shown inFigure (5). Each table uses algorithm NRG (Number Random Generator) shown as Algorithm (2) to get the random numbers from 0..9994. AKG uses these tables to generate the value of random key that is used as a secret key (as shown in Algorithm (3).

Let
First input= 9099 2135 4543 5557 0008 1098 **9994** 2000 9098 **0000** 3213 8789 7987 **0000** 4010 0016 1912 0009 6098 8110 1111 **9994** 1125 0621
Second input=7002 1221 9009 3112 0007 3324 **0002** 2635 3347 **0006** 0000 6322 9212 **0004** 3347 7659 0000 0002 1128 4458 5564 **0003** 0983 9891
Third input= 0002 6008 8993 5000 0022 9004 **0007** 3213 1998 **0005** 0095 2326 9874 **0008** 0043 0032 0000 8004 4320 9991 9123 **0009** 8743 0920
The output of AKG is: 3125 0334 0334 0533 0334 6933 **0035** 1033 0005 **0030** 5002 1875 8331 **0000** 2982 8333 2449 5552 4002 3439 5421 **0042** 1330 1565

```
Algorithm name: Tables implementation
Input : input no. of  tables, no. of rows, no. of columns
Output : Tables of random values
Begin
For i:=0 to no. of tables do
Begin  /* assign the values of each table using random function */
      For j:=0 to no. of rows do
        For k:=0 to no. of columns do
          Begin  /* Generate the random value from 0 to 9994 */
              Call the NRG algorithm to generate the random value.
               Cell[j,k]:=GR
 End; /* Now the implementation of table is done each cell is random no.from 0..9994*/
 End
 End.
```

Algorithm (1): The table's implementation of AKG

```
Algorithm name: Number Random generator (NRG)
Input : input number from 0..4992
Output :  random values from 0..9994
Begin
            /* Generate the random value from 0 to 9994 */
               Detect a number from 0 to 4992
               Add 7 to this number
               Multiply the result by 2 then subtract 4 from result
               GR=The result from the previous step/*represent as 4 digit*/
End.
```

Algorithm (2): The Numbers Random Generator

```
Algorithm name: AKG
Input : 3 inputs of  n digits
Output : key of (4n) digits
Begin
   1-  Enter the 1st input that detect the number of table.
   2-  Enter the 2nd input that detect the number of row in the table.
   3-  Enter the 3rd input that detect the number of column in the table.
   4-  For all digits of three inputs , the cross point of row and column in the table is
        the output(each digit is represent of 4digits).
   5-  The (4n) digits of output represents the key.
End
```

Algorithm (3): Proposed AKG algorithm

Algorithm (4) states the main steps that the user used it to find the origin number corresponding to the random number.

```
Algorithm name: find the origin no.
Input : input the random no.
Output : the origin no. generate form hash function.
Begin
               Take the value of  GR
               Divided it by 2
               Subtract 5 from the result
               Origin=The result from the previous step
End.
```

Algorithm(4): Detection the origin number from the random number

```
Algorithm name: Convert Message (CM).
Input : the character of a message
Output : stream of number (0..8204).
Begin
        1-Convert each character in a message into 8 bits of (0,1).
        2-Divided the stream of (0,1) into blocks. Each block is 13 digits length.
        3- The value of each block is represent in (0..8204)states.
End.
```

Algorithm (5) Converting the message into 0..8204 states

4. Proposal of the secret key generator to calculate the shared secret string

In order to enhance the strength of Hash MessageAuthentication Code (HMAC), this paper generated the Secret key used in calculation of MAC by using the proposed algorithm.

The proposed method needs three keys to calculate the secret string S that will beused in this scheme. K1, K2 and K3 are the names of the three keys.Each one is a stream of numbers (0..9994) with any length generated by using the proposed algorithm called NRG (see Algorithm (2)).These keys are the inputs of the proposed algorithm of key generated (AKG).

The steps of secret string generation using AKG are as follows:

1- The digits of K1 detect the number of table that will be used.
2- The digits of K2 detect the number of rows included in the table that will be used.
3- The digits of K3 detect the number of columns included in the table used.
4- The cross point in the table that will be used detect the output of secret string.

The contents of tables are detected previously before the generation of secret string begins.These contents are generated randomly to get the values of cells; only the sender and receiver know the random key, as shown in Figure (6). Now this secret string is concatenate to the original message.

5. Proposal of the secret key generator to calculate the hash code

Now we used another key k4 which the receiver knows.K4 is considered to be the first input to AKG as shown in Figure (7). K4 is generated by using the proposed NRG algorithm. The original message is considered to be the second input to AKG. The original message is converted to 0..8204 states by using the converted message (CM) algorithm (as shown in Algorithm (5)). The shared secret string applied from the previous section is considered to be the third input to AKG.

Now the output of AKG is considered to be a hash code. The original message is concatenate with this hash code as shown in Figure (4).

Figure (8) shows the details and requirements of the proposed method to generate the secret string and hash code.

6. The Implementation and Discussion

The implementation of the proposed system is done by using visual Basic6 programming language with many pages to operate the proposal algorithms as mentioned inthe previous sections.

The message authentication functions in a way that the hardness of forging anauthenticated message can be related to the cryptographic strength of the underlying hash function. The security strength provided by the HMAC algorithm depends on the security strength of the HMAC key, the underlying hash algorithm, and the length of hash code. The proposed AKG introduces a method that uses three inputs of (0..9994) states instead of (0,1) states.The output of AKG is a random key from key generator using 9995 tables. The number of operations required to break the AKG depends upon the number of bits that the AKG uses to output the key.

AKG uses inputs each one of which is of (0..9994) state with n length. Each digit in these inputs is processed according key generator to output the random n digit of (0..9994). This paper proposes to randomly generate three inputs of (0..9994) states with n length to generate the shared secret string. Then, proposes to use again three inputs of (0..9994) states with n length (K4 (generated randomly by using NRG), secret string, the original message (converted to 0..8204 state)) to generate the hash code that concatenate to the original message. So, the receiver must know the secret string and the key (K4) to calculate the hash code to be able to read the original message and be sure of authenticity. Using random function to generate the value of key, the permuted values according to the 9995-tables each time makes more complexity of the key, even by using smaller block size (at least 256 bit). So the proposed method offers the authentic manner.

.

Table (0)	0	1	2	3	4	5	6	7	8	9	10	….	9994
0	0500	1222	3112	6100	6233	2798	0084	6546	4823	7875	1098		4567
1	4101	1101	0822	1700	1422	1332	0885	3377	5235	1109	4532		4320
2	8434	7781	4033	0202	1174	5822	5499	6655	9333	5154	1111		1200
3	3460	9423	5944	0004	0138	4111	9003	0001	2731	1243	1098		4567
4	0022	3045	5855	6003	9879	0911	8122	9230	**0000**	0873	4532		4320
5	6500	5390	0967	5600	0914	1333	7223	0998	9022	9209	1111		1200
6	0001	8732	6740	4799	2870	**0030**	4933	2091	2320	4490	1098		4567
7	3417	1433	3253	2445	1600	8433	9122	1765	2083	6543	4532		4320
8	0910	3222	6432	8244	4454	9055	6245	6658	0052	4531	1111		1200
9	4555	9033	8043	3833	5233	4423	8123	6899	7398	8743	1098		4567
10	5432	2345	1234	2111	4567	2322	4323	6543	7684	6543	4532		4320
. ..													
9994	0023	3210	1232	0983	3456	1323	1234	5678	1000	2310	1969		3210

.

Table (9994)	0	1	2	3	4	5	6	7	8	9	10	…	9994
0	1402	0000	8130	2145	1673	9211	2381	4492	3312	1000	1234		1234
1	2587	2520	6156	4166	2463	5112	8441	4481	4405	9088	5432		1344
2	4732	4370	2167	2700	3122	9221	0445	**0035**	4451	5422	1334		3121
3	5923	6674	2765	6500	4126	6323	2331	4447	4490	**0042**	5566		1313
4	7330	7551	6030	3055	4330	6332	8336	1771	3308	3352	0000		1313
5	8223	8160	7566	5144	5456	5342	9332	0984	1139	3360	7655		1133
6	0224	8478	8154	6687	1664	6448	8227	5005	2226	7000	4222		8886
7	1226	9266	7333	0303	3772	3444	2222	2003	3356	1887	0000		4436
8	2811	1734	9221	7272	1888	5333	2229	4003	5577	1933	2334		0980
9	1300	2133	1734	8439	990	7232	1213	5008	7726	1235	0775		5555
10	0000	2345	5455	2123	5432	1200	1238	7685	3242	2854	2341		5511
. . .													
9994	0019	2300	4376	8765	5432	8700	4400	4322	0098	2678	0986		1267

Figure (5) The 9995 tables used in proposed method

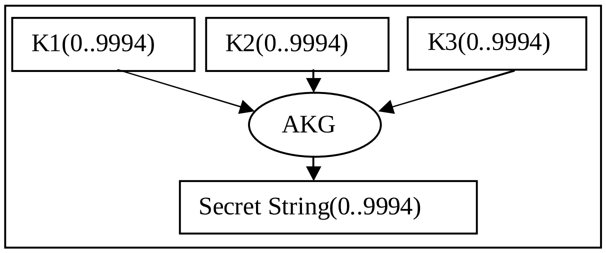

Figure (6) Block diagram to explain the steps of shared secret string generation

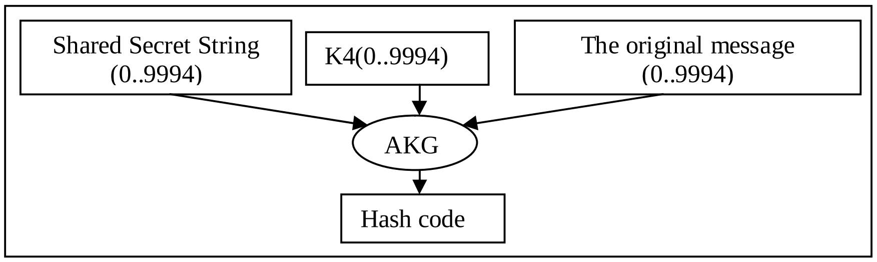
Shared Secret String (0..9994)
K4(0..9994)
The original message (0..9994)
AKG
Hash code

Figure (7) Block diagram to explain the steps of hash code generation

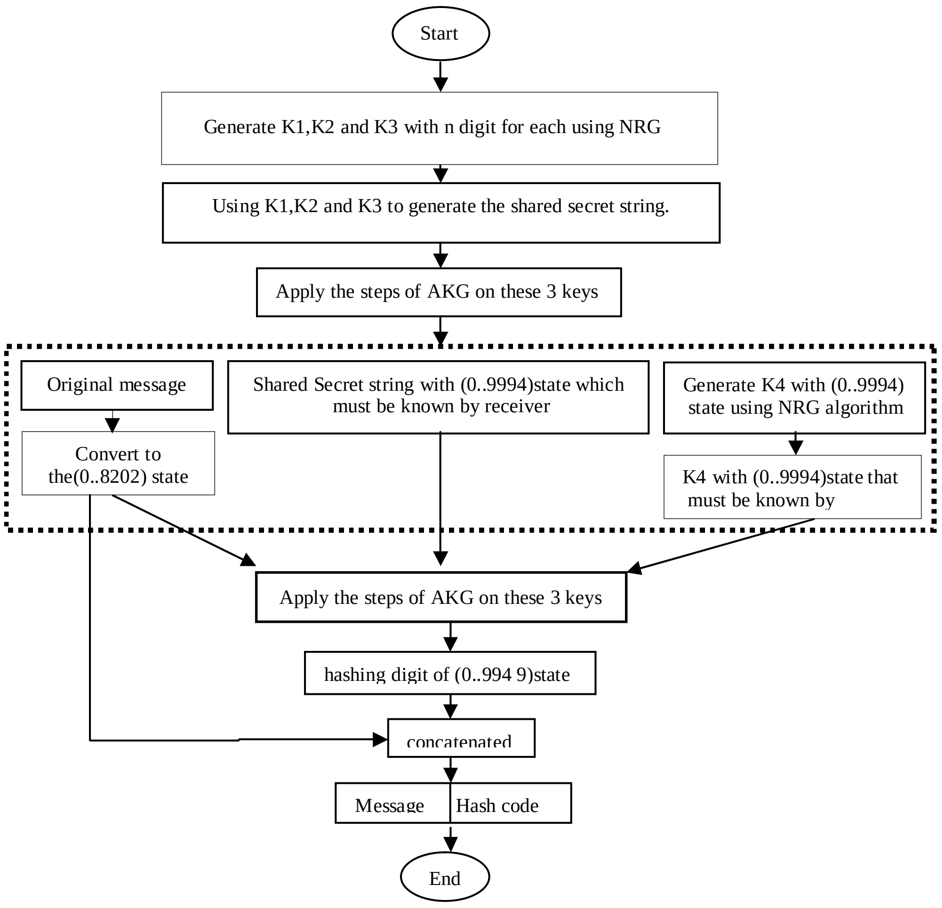
Start
Generate K1,K2 and K3 with n digit for each using NRG
Using K1,K2 and K3 to generate the shared secret string.
Apply the steps of AKG on these 3 keys
Original message
Convert to the(0..8202) state
Shared Secret string with (0..9994)state which must be known by receiver
Generate K4 with (0..9994) state using NRG algorithm
K4 with (0..9994)state that must be known by
Apply the steps of AKG on these 3 keys
hashing digit of (0..994 9)state
concatenated
Message
Hash code
End

Figure (8) Explanation of the proposed method in sender part

The security of AKG (P) depends on the number of keys (N), length of each key (L), number of Tables (T), number of cells of each table (C), and number of possible states of each cell in each table(S). The possible state of each cell is a number from 0 to 9994 according to the proposed function by using a number from 0 to 4992.

The security of AKG (P)=N*L*T*C*S.

The proposed AKG is applied to any size of data and produces a fixed –length output to pad the output to the message. So, the proposed method makes it relatively easy to compute any given string. Key is generated with less computation and used.

Notice thatTable (1) shows the relation of message size (digits of (0..9994) and total time of the proposed method measured by nanosecond.Table(2) summarizes the randomness of the key generated for different lengths. So the generated key passed the randomness tests.

Use the following form to get the similarity percentage of the sender and receiver message:

*Similarity = ((total message size-difference)/total message size)*100)*

Difference=$\sum_{message\ size} \sqrt[2]{(x_2 - x_1)^2}$

Where, x is the bits of messages.

Table (3) shows the value of the similarity of sender and receiver messages.

Table (1) Execution Time(inns)of the proposal method

Message size	Sender Speed(ns)	Receiver Speed(ns)	Total Time(ns)
32bit	3.304	3.304	6.608
64bit	5.119	5.119	10.238
128bit	11.001	11.000	22.001

Table (2) Randomness test value of the proposal method

Key size (bit)	Freq. Test(<=3. 84)	Serial Test(<=5. 99)	Poker Test(<=5. 99)	Run test(<-22.362		Correlation test					
				T0	T1	Shift1	Shift2	Shift3	Shift4	Shift5	..
32	0.175	1.875	2.4	3.749	3.011	0.134	0.345	0.563	0.865	0.376	..
64	0.069	0.179	0.784	7.341	11.23	0.196	1.4e-02	3.2e-02	0.134	6.8e-02	..
128	0.675	0.832	4.187	13.43	9.543	4.3e-02	1.4e-03	1.06e-02	1.65e-02	0.432	..

Table (3) Integrity Test of the messages

Message size(bit)	Similarity percentage
32	100%
64	100%
128	99.6%

7. Conclusions

The authentication, snooping attacks and replay prevention are essential in secured communications. Some ofthe conclusions of the proposed method are:

a) Checking for the message integrity: the receiver is able to identify that the message is coming from a valid source and that it is not modified.
b) The proposed method to design a HMAC algorithm provides authenticity by using key generator function which increases the randomness of the used secret key by using multi-state (0..9994) instead of (0,1) state of the original key through the tables generator and random function which gives a randomness as a new secret key and secret string with randomness property.
c) The new approach of random key generator is developed to increase the complexity degree against the attacker task by consuming more time to achieve analytical process depending on the state and number of keys used.
d) The proposed method uses only hash function without certification.

References

[1] Y. Dodis, R. Thomas and Sh. Thomas," Advances in Cryptology-ROCRYPT'09, *Vol.5479, pp.371-388, Springer-Verlag, 2009.*

[2] C. Y. Christopher,"Security Proofs for the MD6 hash function Mode of operation", Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 2008.

[3] William Stalling, "Cryptography and Network Security Principle and Practices", Fourth Edition, Prentice Hall, 2005.

[4] K.B., S. V.," Efficient HMAC Based Message Authentication System for Mobile Environment ", *International Journal of Advanced Engineering Sciences and Technologies Vol. No.11, Issue No.1, 208 – 212.*

[5] Quynh Dang, "Recommendation for Applications Using Approved Hash Algorithms", Computer Security Division, Information Technology Laboratory Computer Security,August 2012.

[6] Naim A., Asim E. and Abdullah A., "A new approach in keygeneration and expansion in Rijndael algorithm", *International Arab Journal of Information Technology, Vol.3, No.1, January 2006.*

[7] P. Murali and R. Palraj, "True Random number generator method based on image for key exchange algorithm", *International Symposium on Computing, Communication, an Control of CSIT vol.1 (2011),Singapore*

[8] Manish K., Ashish A., and Gaurav M.," Advancing the Cryptographic Hash-Based Message authentication Code", *IACSIT International Journal of Engineering and Technology, Vol.3, No.3, June 2011*

[9] Vilas S.B., A. Dhotre, "Information Security", 1st Edition, published by Technical Publication Pune, 2009.

[10] Avi Kak, "Computer and Network Security", lecture note, March 11, 2011

[11] Syeda Iffat , Adeel Akram, "Pseudo-random Key Generation for Secure HMAC-MDS", Faculty of Telecom & Information Engineering, UET Taxila, Pakistan,2010.