Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Collaborating Integrated Diffie-Hellman Digital Signature with Smart Snort

Ritu Makani, Yogesh Chaba
Department of CSE GJUS&T
Haryana, India

Abstract— Cryptographic techniques and tools are playing an important role in designing emerging network security technologies. It is evident from the fact that world's most developed countries like U.S are considering cryptographic technology as the standard technology keeping in view the security aspect of the fast growing commerce, banking, military activities of the world and it is the need of the day that it should be standardised so that the whole world can benefit from it. It is the need of the day to collaborate varied security measures taken keeping in view varied aspects of security. Other than cryptographic techniques intrusion detection systems and intrusion prevention systems are there to detect and prevent occurrence of cyber financial as well as social crimes. Effort has been made in this paper to collaborate an integrated encryption and authentication algorithm to work in line with IDS like Snort which has been made "Smart Snort" by customizing it with improved rules, recrafted automata (rule tree structure) and also making snort to work as an intrusion prevention system. A conceptual model has been proposed with IDHDS and Smart Snort.

Keywords— Encryption, Automata, Pattern Matching, Digital signature, Smart Snort

I. INTRODUCTION

The explosive growth in computer systems and their interconnections via networks has increased the dependence of both organizations and individuals on the information stored and communicated using these systems. This, in turn, has led to a heightened awareness of the need to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks.Encryption, Authentication mechanisms, Intrusion Detection Systems, Security Management and Firewalls can be used to increase the security of computers in the Network. Combination of these techniques can be used in organizations to increase security of systems worldwide. The disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications to enforce network security. An intrusion detection system (IDS) is a well known security tool used by companies to prevent loss and harm of data. An open source intrusion detection system is a good option for companies and organizations which do not have the same amount of money as the larger companies and governmental organizations. It is difficult to know which intrusion detection system to choose without any previous knowledge.

Intrusion detection methods: The basic intrusion detection methods are the two complementary approaches for detecting intrusions with their inherent advantages and disadvantages

Anomaly detection or Behaviour analysis based methods: These methods use information about repetitive and usual behaviour on the systems they monitor, and this approach identifies events that deviate from expected usage patterns as malicious. Most anomaly detection approaches attempt to build some kind of a model over the normal data and then check to see how well new data fits into that model. In other words, anything that does not correspond to a previously learned behaviour is considered intrusive. Therefore, the intrusion detection system might not miss any attacks, but its accuracy is a difficult issue, since it can generate a lot of false alarms..

Anomaly detection can be either by *Unsupervised learning systems* which learn the normal behaviour of the traffic by observing the traffic for an extended period of time and building some model of the underlying process or by *Supervised Systems* in which the system has to be taught to detect certain anomalous events. The supervised anomaly detection approaches build predictive models provided labelled training data (normal or abnormal users' or applications' behaviour) are available. Thus the user of the system forms an opinion on what is considered abnormal for the system to signal a security violation.

Advantages of behaviour-based approaches: It detects new and unforeseen vulnerabilities
and is less dependent on operating system-specific mechanisms. It also detects 'abuse of privileges' types of attacks that do not actually involve exploiting any security vulnerability.

Disadvantages of behaviour-based approaches include the high false alarm rate is generally cited as the main drawback of behaviour-based techniques. Further the entire scope of the behaviour of an information system may not be covered during the learning phase and behaviour can change over time, introducing the need for periodic on- line retraining of the behaviour profile. Moreover the information system can undergo attacks at the same time the intrusion detection system is learning the behaviour. As a result, the behaviour profile contains intrusive behaviour, which is not detected as the

entire scope of the behaviour of an information system may not be phrased. Very few commercial tools today implement such an approach, leaving anomaly detection to research systems but still this is a requirement for IDS systems.

Knowledge-based or misuse based detection methods: Knowledge-based detection or misuse detection or signature detection methods use information about known security policy, known vulnerabilities, and known attacks on the systems they monitor. This approach compares network activity or system audit data to a database of known attack signatures or other misuse indicators, and pattern matches produce alarms of various sorts. All commercial systems use some form of knowledge-based approach. Thus, the effectiveness of current commercial IDS is based largely on the validity and expressiveness of their database of known attacks and misuse, and the efficiency of the matching engine that is used. It requires frequent updates to keep up with the new stream of vulnerabilities discovered, this situation being aggravated by the requirement to represent all possible facets of the attacks as signatures. This leads to an attack being represented by a number of signatures, at least one for each operating system to which the intrusion detection system has been ported. Examples for product prototypes are Discovery, IDES, Haystack and Bro.

II. PREVIOUS WORK DONE

Jim Omura has proposed alternatives to RSA: using Diffie-Hellman with DSS [1], P. C. van Oorschot and M. J. Wiener worked on Diffie-Hellman key agreement with short exponents[2]. A. J. Menezes, P. C.Van Oorshot, and S. A Vanstone, in their book titled Handbook of Applied Cryptography explain the basics of applied cryptography and its applications[3]. Aho A.V; Corasick M.J. worked on efficient string matching as an aid to bibliiographic search[4]. Zhou Zhimin *et al* worked on the study on network intrusion detection system of Snort[5]. Chintan C. Kacha *et al* worked on improved Snort intrusion detection system using modified pattern matching technique[6]. Tongaonkar A *et al* threw some light on fast packet classification for Snort by native compilation of rules[9]. Jiqiang Zhai *et al* worked on network intrusion prevention system based on Snort. Deepak Dembla et.al have discussed [10] about modelling and analysis of intelligent AODV routing protocol based on request retransmission strategy in MANETS. Yogesh Chaba et.al in their work [11] have discussed about performance analysis of disable IP broadcast technique for prevention of flooding-based DDoS attack in MANETS. Further Yudhvir Singh et.al have described about information theory tests based performance evaluation of cryptographic technique [12].. But none of the above talked about collaborating the goods of hybrid cryptographic algorithms(IDHDS) with open source intrusion detection systems working smartly(smart Snort) to benefit the today's commercial scenarios where there can be two party or multiparty secret communication involving finance or secret data or information.

III. PROPOSED WORK

Proposed work includes the study and design of the integrated Diffie-Hellman Digital Signature algorithm and smart Snort and then combining them to hybrid both to achieve the better security techniques.

Integrated DH-DS Algorithm (IDHDS): The integrated Diffie-Hellman Digital Signature algorithm begins with an assumed prime number. Function primitive is used to compute the primitive root of the given prime number. There can be more than one primitive roots of the given prime number. Given coprimes of prime number g^nmodp where n varies from 1 to (p-1). If the remainders are unique then g is the primitive root of p. Given x as the secret key of user A compute g^xmodp. The user B selects the secret key y. y is then used to compute B to g^ymodp. User A transmits A to user B and user B transmits B to A. With this received value A computes k1=B^xmodp. Then user B uses A to compute k2=A^ymodp. The computation of K1 comes the same as computation of K2 i.e the key computed for application in the encryption process. Once the key is generated we begin to digitally sign the message. Digital signature if applied to a document the sender cannot nonrepudiate. Signature generation begins by selection of public key component p such that $2^{L-1}<p<2^L$(512<=L<=1024), L is a multiple of 64 bits. Then select q such that it is prime divisor of (p-1). Initialize i to 2. Start the while loop to check if remainder of (p-1)/q is 0 then set q=1 else increment i. Choose h such that it is an integer in the range 1 and (p-1). Compute t such that $h^{(p-1)/q}$%p.

Choose x1 as pseudorandom number between 0 and q and compute y1 as (t^{x1}%p). Then choose a pseudorandom value for k between 0 and q. To compute r component of digital signature we apply (t^k%p)%q. The value of r is independent of either the input message or hash of the message. It is a function of k,p,q and t. The value computed for r is secret since it involves the use of k which is a one time secret number between 0 and q. Then we input the integer or alphabetic message in msg. The msg in textual form is liable to spoofing so hash of the msg(digest1) is computed using md5_hex(msg). This function accepts as input variable size message and generates a hash value of 128 bits. This hash value is then utilized in the computation of s value of digital signature as S1=[k^{-1}*digest1+x1*r]%q where x1 ia a random number between 0 and q. The value of S1 computed is then transmitted to the receiver. Thereafter sha1_hex(msg) is applied to generate hash of the message using sha1(digest2). This function takes as input a message of any length less than 2^{64} and produces as output hash value of 160 bits. The variable digest2 is then used to compute S2 as [k^{-1}*digest2+x1*r]%q. After this, sha256_hex(msg) is used. This function takes as input message of length less than 2^{64} and produces hash value of 256 bits. This generates hash value digest3 which is used to comput S3 as [k^{-1}*digest3+x1*r]%q. Then sha384_hex(msg) is applied. This function accepts a message of any length less than 2^{128} and produces a hash value of length 384 bits. sha384_hex(msg) operates on 64 bit words which computes digest4 to be used for computation of S4 as [k^{-1}*digest4+x1*r]%q. Lastly we apply sha512_hex. This function takes as input a message of length less than 2^{128} and produces a hash value of length 512 bits. The hash produced is digest5 to be further used in S5 computation as [k^{-1}*digest5+x1*r]%q. Blowfish algorithm is then applied to encrypt the message. After the r and s components are computed, this forms the digital signature generation, then the digital signatures are to verified. The value of w1 is

calculated as (1/S1')%q where S1' is the received value of S1. The value of digest1' is used to calculate u1 as (digest1' *w1)%q where digest1' is the received value if digest1. Next use r' value to compute u2 as (r'*****w1)%q where r' is the received component of r. After u1 and u2 gets computed we finally calculate v as ((g^{u1}*$y1^{u2}$)%p)%q). If the calculated v value comes equal to r' value then the signature is verified.

ALGORITHM **:**

```
Begin
        Prime←p
        int g=sub primitive(p)
        x← secret key of user A
        A←g^x%p
        Y←secret key of user B
        B←g^y%p
        k1←B^x%p
        k2←A^y%p
        k1=k2=key for encryption(k)
        i=2
        while(i<p-2)
                start
                        if((p-1)%i==0) then
                            q=i
                        else
                    i=i+1
                end
        q←prime divisor of (p-1)
        h←integer between 1 and p-1
        t←h^((p-1)/q)%p
        x1←random number between 0 and q
        y1 ←(t^x1)%p
        k←random number with 0<k<q
        r←((t^k)%q)%q
        msg←input message
        digest1←md5_hex(msg)
        S1←( k -1 * digest 1 + x1*r) % q
        digest 2←sha1_hex(msg)
        S2←( k -1 * digest 2 + x1*r) % q
        digest 3← sha256_hex(msg)
        S3←( k -1 * digest 3 + x1*r) % q
        digest 4←sha384_hex(msg)
        S4←( k -1 * digest 4 + x1*r) % q
        digest 5←sha512_hex(msg)
        S5←( k -1 * digest 5 + x1*r) % q
        E=blowfish(msg);
        W←(1/S1') % q
        u1←(digest1'*w) % q
        u2←(r'*w) % q
        v←(((g^u1)*(y1^u2) % p)%q
        v= r'{ signature verified}
end
sub primitive(p)
        for n=1 to p-1
        do
                y=gn % p
                if  y unique
                return y
        done
end sub
```

Smart Snort:

As far as working with the open source IDS snort is concerned its pattern matching scheme can be modified by working on the rule tree structure(automata). Changes can be made in the order of processing of rules. Rule application order by default in the passive mode with standard rule set called VRT rules is: activation->dynamic->pass->drop->sdrop->reject->alert->log. However this can be changed to economise the pattern matching effort of snort and lead to faster packet

processing. NIDSs often cannot decipher the packets they capture. In addition, in the absence of something like a corporate key, no IDS can decipher encrypted information. Snort was configured in the inline mode with more refined rules and its auutomata also worked so it can work as smart Snort.

Automata file :

```
#include <stdio.h>
#include <string.h>
#include <ahocorasick.h>
char* sample_patterns[] = {
   "woodcock",
   "WOODCOCK",
   "trueman",
   "truman",
   "unix",
   "UNIX",
};
#define PATTERN_COUNT (sizeof(sample_patterns))
char* input_text1 = {"woodcock cannot be same as WOODCOCK"};
char* input_text2 = {"trueman is not same as truman"};
AC_ALPHABET_t * input_text3 = {"unix is not equal to UNIX"};

int main (int argc, char ** argv)
{
   unsigned int i;
   AC_AUTOMATA_t   *atm;
   AC_PATTERN_t    tmp_pattern;
   AC_TEXT_t       tmp_text;
   atm = ac_automata_init ();
   for (i=0; i<PATTERN_COUNT; i++)
   {
      tmp_pattern.astring = sample_patterns[i];
      tmp_pattern.rep.number = i+1; // optional
      tmp_pattern.length = strlen(tmp_pattern.astring);
      ac_automata_add (atm, &tmp_pattern);
   }
   ac_automata_finalize (atm);
   ac_automata_display (atm, 'n');
   printf ("Searching: \"%s\"\n", input_text1);
   tmp_text.astring = input_text1;
   tmp_text.length = strlen(tmp_text.astring);
   ac_automata_settext (atm, &tmp_text, 0);
   AC_MATCH_t * matchp;
   while ((matchp = ac_automata_findnext(atm)))
{
      unsigned int j;
      printf ("@ %2ld: ", matchp->position);
      for (j=0; j < matchp->match_num; j++)
        printf("#%ld (%s), ", matchp->patterns[j].rep.number, matchp->patterns[j].astring);
      printf ("\n");
   }

  ac_automata_release (atm);
   return 0;
}
```

The time complexity of the algorithm is of O(f(k)+f(q)) in which f(k) is function of length of the pattern and f(q) is the function of the length of the text.

IV. RESULTS

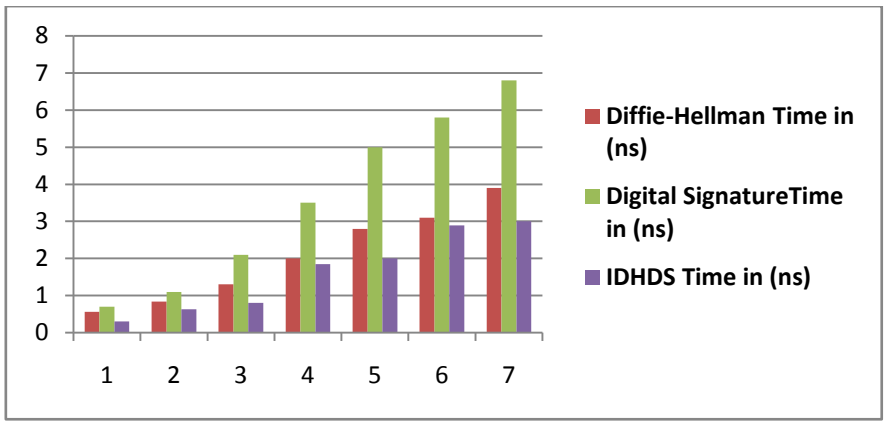
Before discussing the results received in this section let us discuss some parameters on which the performance evaluation of the implemented scheme/mode on which work has been done can be checked for any change in the existing scheme of things.Execution time, Packet processing time, time complexity are some of these parameters. Further rule modifications in respect of the above mentioned parameters are also one of them.

(A) Execution Time:

Table 1 Execution Time of the D-H, DS, IDHDS

Execution Time in nano seconds			
No of Rounds	**Diffie-Hellman Time in (ns)**	**Digital Signature Time in (ns)**	**IDHDS Time in (ns)**
1	0.56	0.7	0.3
5	0.84	1.1	0.63
10	1.3	2.1	0.8
15	2	3.5	1.85
20	2.8	5	2
25	3.1	5.8	2.89
30	3.9	6.8	3

Table 1 shows the execution time of Diffie -Hellman, Digital Signature Algorithm and IDHDS for different number of rounds.



Graph 1 Comparative Execution Time of the DH, DS, IDHDS

Graph 1 shows the comparative execution times for 30 rounds of the proposed IDHDS algorithm, D-H and DS. Execution time of the proposed algorithm is much less as compared to other two algorithms.

(B) Packet Processing Time for inside TCP Dump Data Set

Total packets	snort	smart snort
1011149	5.741	4.386
1563069	11.293	10.912
1362422	9.153	8.374
1753377	11.456	10.753
1362422	8.936	7.591

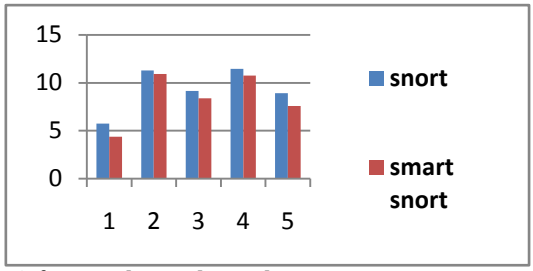


Figure 2 Packet Processing time (sec) for inside tcpdump dataset

Figure 2 shows the runtime for packet processing for inside tcpdump dataset from DARPA.Here we analyze that the smart snort depicts lesser execution time than snort for the DARPA dataset taken from DARPA Intrusion Detection system Evaluation Datasets. The data was taken for inside tcpdump format and given to snort and smart snort.

(C) Packet Processing Time for outside TCP Dump Data Set

Total Packets	Snort	smart snort
1337777	10.47271	9.4581
1535894	10.8945	9.3961
888139	4.791	3.9513
1412645	9.23988	8.6329
1252412	8.21449	7.75313

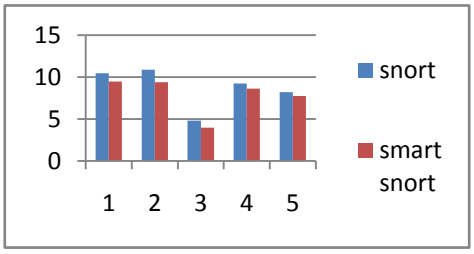


Figure 3 Packet Processing time (sec) for outside tcpdump dataset

Figure 3 shows the runtime for packet processing in seconds for outside tcpdump dataset. It is found that Smart Snort shows less execution time for packet processing for outside tcpdump dataset from DARPA Intrusion detection system Evaluation dataset.

Performance Analysis of IDHDS

The time difference required for execution of integrated DHDS algorithm textual messages and integer messages is about 5-6% . Time complexity of the algorithm is O(log k M(n)) where M(n) is complexity of multiplying two n bit integers Signature generation requires one modular exponentiation so its complexity is $O(\log^3 n)$ and signature verification requires two modular exponentiation so its complexity is $2O(\log^3 n)$ since here k=O(n) and time complexity of multiplying two n bit integers is $O(\log^2 n)$.

(D) Proposed Hypothetical Security Model

In the end hypothetical security model has been proposed by combining the smart snort and the IDHDS. It incorporates the implementation of the proposed encryption and authentication algorithm IDHDS with smart Snort which has been discussed in this and the previous chapter. Basic block diagram of this proposed model is shown in the Figure 2:

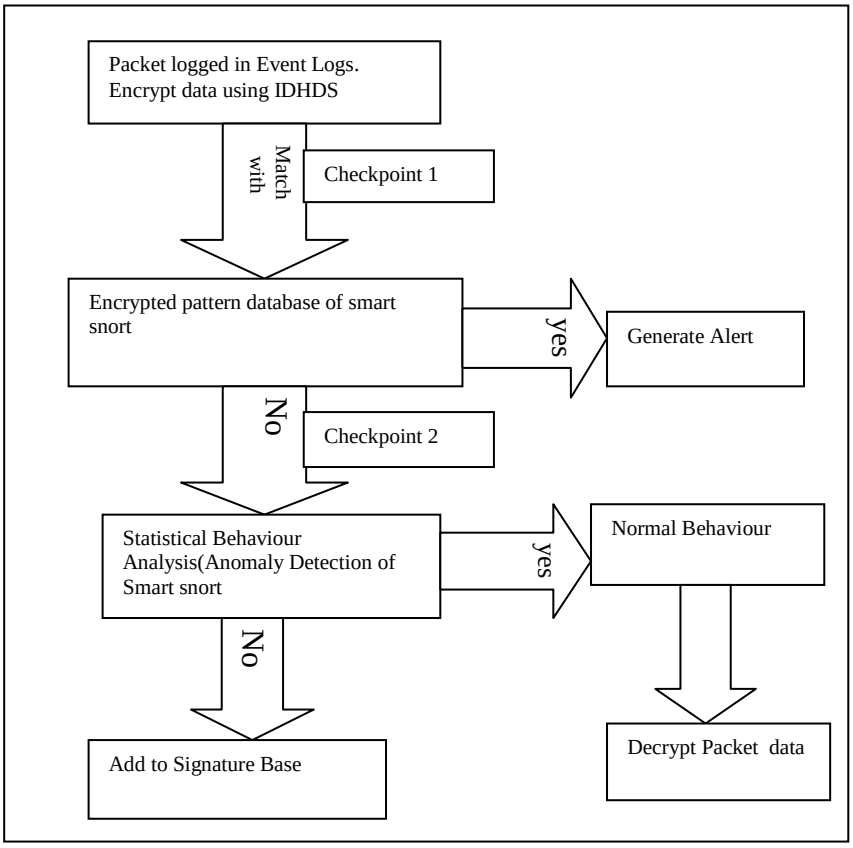


Fig 4 Proposed security model using Smart Snort and IDHDS

1. Suppose a packet arrives and is logged in event logs. It is encrypted using IDHDS algorithm and its pattern is matched with encrypted pattern database (signature database) already stored in the system. Signature database is specifically known patterns of unauthorized behaviour. If it matches with the signature database that means it is an attack packet so an alert is generated.
2. If the pattern does not match it is sent for a second stage check Anomaly Detection which is statistical behaviour analysis. If it is not the normal behaviour the signature database is updated to include this new signature.
3. If it shows a normal behaviour then packet data is decrypted and if it does not match then system alerts for deny access.
4. This way evasion of IDS may be reduced to some extent at important checkpoints. The model include IDHDS algorithm for Encryption and smart snort for pattern matching.

V. CONCLUSIONS

From the block diagram of the proposed conceptual security model it is assumed that IDS will itself have the inherent flexibility of decryption and encryption for encrypted network data packets and will overcome the drawback of an IDS. Cryptographic algorithms clubbed with an open source IDS snort which has been made to work smartly and will serve the purpose of an IPS also, will provide a unique substitute for a robust security system. Collaborating the integrated DiffieHellman-Digital Signature(IDHDS) with Smart Snort can be a novel way to key exchange problems between two parties along with the watch dog activity of smart snort. Side by side problem of slowing down of the performance of the

system is also addressed by faster packet processing of smart snort. Proposed hybrid model can be a base for the design of some simulation products on network security where crypt functions can be used with smart Snort.

REFERENCES

[1] Jim Omura "*Alternatives to RSA: Using Diffie-Hellman with DSS"* CYUNK Resource Library White Papers

[2] P. C. van Oorschot and M. J. Wiener, On "*Diffie-Hellman Key Agreement with Short Exponents. EUROCRYPT'96*, LNCS 1070, Springer-Verlag, 1996, pp. 332–343.

[3] A. J. Menezes, P. C. van Oorshot, and S. A Vanstone, *Handbook of Applied Cryptography*. CRC Press, New York, New York, 1997.

[4] Aho A.V.; Corasick M.J.(1975): Efficient String Matching: An Aid to Bibliiographic Search, Bell Laboratories, Communication of the ACM, pp.333-340.

[5] 2010 International Conference on Networking and Digital Society The Study on Network Intrusion Detection System of Snort Zhou Zhimin, Chen Zhongwen , Zhou Ti echeng, Guan Xiaohui Department of Computer Science Zhejiang Water Conservancy And Hydropoeer College Hangzhou, China

[6] Chintan C. Kacha *et al* "Improved Snort Intrusion Detection System Using Modified Pattern Matching Technique" IJTEAE(ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 7, July 2013

[7] Roesch M. (1999): Snort;Lightweight intrusion detection for networks, In Proceedings of the 1999 USENIX LISA Systems Administration Conference, pp.229-238.

[8] Comparison of Different Intrusion Detection And Prevention Systems, Abhishek Chauhan

[9] Tongaonkar A; Vasudevan S; Sekar R(2008): Fast Packet Classification for Snort by Native Compilation of Rules, Stony Brook University, 22nd Large Installation System Administration Conference, pp.159-165.

[10] http://sourceforge.net/projects/multifast/

[11] 2011 The 6th International Forum on Strategic Technology Researh on Network Intrusion Prevention System Based on Snort Jiqiang Zhai, Yining Xie Computer Science & Technology College,Harbin University of Science and Technology Harbin,China

[12] Deepak Dembla, Yogesh Chaba, "Modeling and Analysis of an intelligent AODV Routing Protocol based on Route Request Retransmission Strategy in MANETs" International Journal of Computer Applications (0975-8887), Vol. 30, Issue 11, pp. 6-13 (2011)

[13] Yogesh Chaba, Yudhvir Singh, Preeti Aneja, "Performance Analysis of Disable IP Broadcast Technique for Prevention of Flooding-Based DDoS Attack in MANET*" Journal of Networks*, Vol 4, Issue 3, pp. 178-183 (2009)

[14] Yudhvir Singh, Yogesh Chaba, "Information theory tests based performance evaluation of cryptographic techniques*" International Journal of Information Technology & Knowledge Management*, Vol. 1, Issue 2, pp.475-483 (2008).

[15] William Stallings, "Cryptography and Network Security", *Pearson Education* (2003)

