Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

6th International Conference
on Electrical Engineering
ICEENG 2008

Security evaluation of VoIP cryptographic algorithms

By

Faiz Yousif Mohmmed* Alaa Eldin Rohiem** Ashraf Diaa Elbayoumy**

Abstract:

The communications world is moving toward VoIP but does not have the security expertise it needs in-house to meet the real world stress it will encounter. Unfortunately adding security has a negative effect on the voice quality of service (QoS). Many researches study the effects of adding security to VoIP using different cryptographic algorithms by comparing end to end delay, jitter and packet loss. Subjective and objective test methods are used to measure QoS [3,10]. But in our paper a new point of view is included, the goal of the paper is to compare the powerful of the securing algorithm by measuring the randomness of its encrypted output. In addition we measure execution time (delay) to be another factor of comparison. A C++ simulation program was written to simulate secure VoIP system. This paper organized as follows: 1. Introduction Section present the VoIP motivation, challenges, and security issues, section 2. Explain VOIP model, section 3. Describe statistical test suite and test bed environment, section 4. Demonstrate the statistical results, and the last section Express the conclusion.

Keywords:

VoIP, SVoIP, Cryptography, StsGui, ASCII Converter, Randomness measurement, StsGui, authentication.

---

* Sudanese Armed Forces
** Egyptian Armed Forces

1. Introduction:

Voice over IP (VoIP) can be briefly defined as the technology that allows the use of the IP protocol(s) to carry voice signalling and media traffic [1].
VoIP sound is sampled, quantified, digitized and compressed with an appropriate codec and streamed over traditional network architectures. Then, several coded speech frames are packetized to form the payload part of a packet (e.g. RTP packet). The headers (e.g. IP/UDP/RTP) are added to the payload and form a packet which is sent to IP networks. The packet may suffer different network impairments (e.g. packet loss, delay and jitter) in IP networks. It is and it behaves as normal IP data but at the same time has to obey to the rules imposed by classical telephony in terms of quality of service and availability [2,3] .
The main motivations for VoIP are: (bypass toll switches and save on call costs, Rich media conferencing combines voice, video, and data, mobility).VoIP service requirements are summarized in minimizing latency, enable bandwidth priority, ensuring reliability and ensuring security.
Confidentiality, Integrity, and Availability three key principles that should be guaranteed in any kind of secure system. This principle is applicable across the whole security spectrum. Confidentiality refers to mechanisms that ensure that only authorized individuals may access secure information. Cryptography and Encryption are examples of methods used to ensure confidentiality of data. Integrity means that information is unchanged as it moves between endpoints. Availability characterizes the operational state of the network [5].
VoIP data is transmitted in digital packet form. This means that the voice transmissions can be attacked, hacked, intercepted, manipulated, rerouted and degraded just as any data packet on the data network. Viruses, worms, trojan horses, denial of service attacks and hijacking are all possibilities on the VoIP network [4].
The implementation of various security measures can degrade QoS. These complications range from delaying or blocking of call setups by firewalls to encryption produced latency and delay variation (jitter) [5].
VOIP calls must achieve the 150 ms bound to successfully emulate the QoS that today's phones provide [6].
Adding security constraints significantly increases the bandwidth usage, causing more latency and jitter, thereby degrading the overall QoS of the network. In addition, these requirements do not explicitly take into account the heterogeneous data flow over the network. Since voice and data streams are sharing the same finite bandwidth, and data streams tend to contain much larger packets than VOIP, significant amounts of data can congest the network and prevent voice traffic from reaching its destination in a timely fashion. For this reason, most new hardware devices deployed on networks support QoS for VoIP [7].

VoIP deployments generally operate over various signalling and transport protocols which have a vital role in the management and transmission of the data packets, and also represent weak points and opportunities for malicious activities.
VOIP adds a number of complications to existing network technology, and these problems are magnified by security considerations. The packet switching nature of data networks allows multiple connections to share the same transport medium. Therefore, unlike telephones in circuit switched networks, an IP terminal endpoint can receive and potentially participate in multiple calls at once. Thus, an endpoint can be used to amplify attacks. On VoIP networks, resources such as bandwidth must be allocated efficiently and fairly to accommodate the maximum number of callers. This property can be violated by attackers who aggressively and abusively obtain an unnecessarily large amount of resources. Alternatively, the attacker simply can flood the network with large number of packets so that resources are unavailable to all other callers [5].
Theft of services and information is also problematic on VoIP networks. These threats are almost always due to active attack. Many of these attacks can be thwarted by implementing additional security controls at layer 2.This includes layer 2 security features such as DHCP Snooping [4].
Users may defer transitioning to IP Telephony if they believe it will reduce overall network security by creating new vulnerabilities that could be used to compromise non-VoIP systems and services within the same network.
Firewalls, network and system intrusion detection, authentication systems, antivirus scanners, and other security controls, which should already be in place, are required to counter attacks that might debilitate any or all IP-based services (including VoIP services).
The most comprehensive list of VoIP threats is maintained by VOIPSA at 8].

2. VoIP Simulation Program Model (FCHAT).

FCHAT is visual c++ socket programming package consists of two sides. The server side is play as a controller, authenticator and a communicator between the other clients. The client side is able to record, encrypt and send voice message to one or all clients. At the same time the client can receive, decrypted and replay the real time voice message. Fig.1 describes how the server receive the connection request and if he accept the request then the “NEW user Address” is encrypted and broadcasted to all other clients. Fig.2 describes the real time conversation between two clients through the server and how they can record, encrypt and send the messages in the same time that they can receive, decrypt and play the incoming messages. Fig.3 represents how client record, encrypt and send the messages to **all** other clients that are decrypt and replay the voice messages (conference).

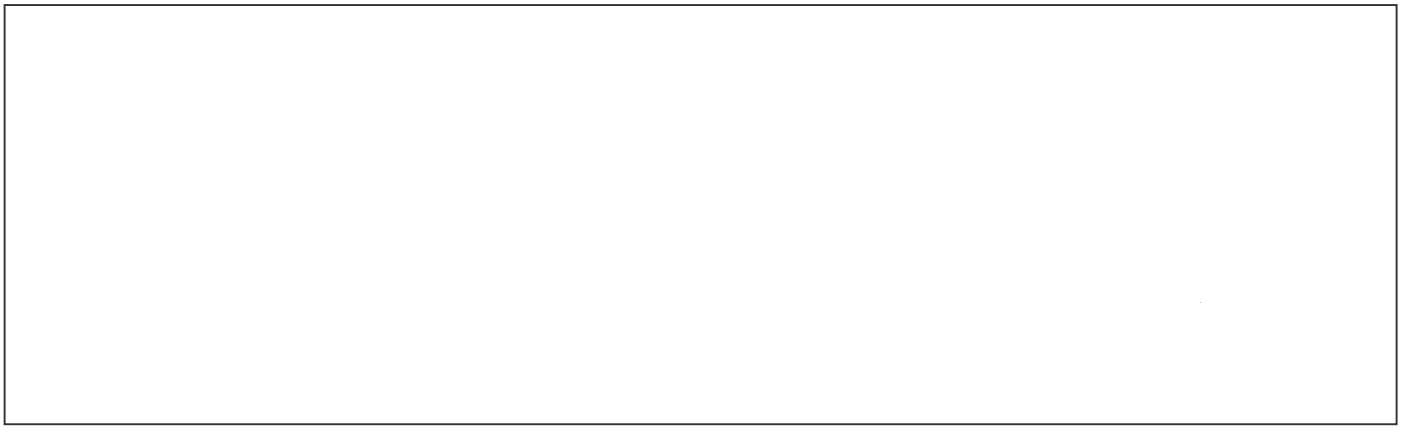

Figure (1): Broadcast authorized address

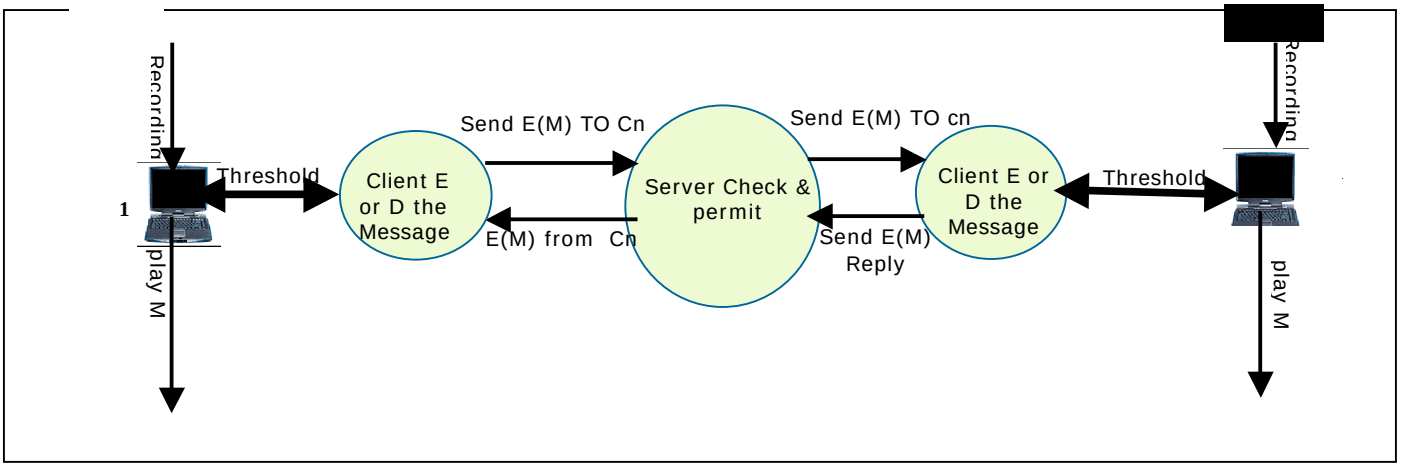

Figure (2): Real Time Conversation

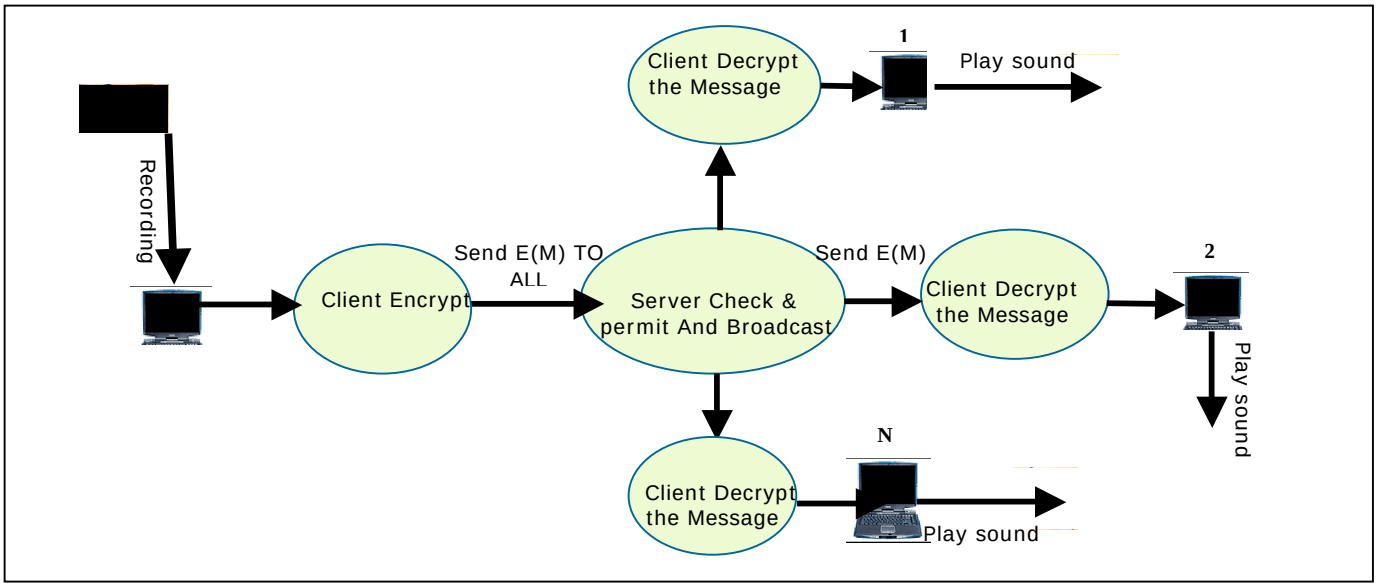

Figure (3): VoIP Conference

3.1 StsGui (NIST Statistical suite)

The National Institute of Standards and Technology (NIST) develop StsGui Test Suite as a statistical package consisting of 16 tests that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based Cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non randomness that could exist in a sequence. Some tests are decomposable into a variety of subtests. The 16 tests are mention in table 1.The test code was developed using a SUN workstation under the Solaris operating system. No guarantee is made regarding the compilation and execution of the PRNG implementations on other platforms. For this reason, a switch has been incorporated into the source codes to disable the inclusion of the PRNGs [9].
The objectives during the development of the NIST statistical test suite included:

- Platform Independence: The source code was written in ANSI C. However, some modification may have to be made, depending on the target platform and the compiler.
- Flexibility: The user may freely introduce their own math software routines.
- Extensibility: New statistical tests can easily be incorporated.
- Versatility: The test suite is useful in performing tests for PRNGs, RNGs and cipher algorithms. Portability: With minor modifications, source code may be ported to different platforms. The NIST source code was ported onto the SGI Origin, and a 200 MHz PC using the Microsoft Visual C++ 6.0 development environment.
- Orthogonality: A diverse set of tests is provided.
- Efficiency: Linear time or space algorithms were utilized whenever possible.

The majority of the tests in the test suite either examine the distribution of zeroes and ones in some fashion, study the harmonics of the bit stream utilizing spectral methods, or attempt to detect patterns via some generalized pattern matching technique on the basis of probability theory or information theory.

In practice, any number of problems can arise if the user executes this software in unchartered domains. It is plausible that sequence lengths well beyond the testing procedure (i.e., on the order of 10) may be chosen. If memory is available, there should not be any reason why the software should fail. However, in many instances, user defined limits are prescribed for data structures and workspace [9].

- The parameter ALPHA denotes the significance level that determines the region of acceptance and rejection. NIST recommends that ALPHA be in the range (0.001,0.01).

Table (1): StsGui tests(From: [9])

Test N0	Description
1.	The Frequency (Monobit) Test,
2.	Frequency Test within a Block,
3.	The Runs Test,
4.	Test for the Longest-Run-of-Ones in a Block,
5.	The Binary Matrix Rank Test,
6.	The Discrete Fourier Transform (Spectral) Test,
7.	The Non-overlapping Template Matching Test,
8.	The Overlapping Template Matching Test,
9.	Maurer's "Universal Statistical" Test,
10.	The Lempel-Ziv Compression Test,
11.	The Linear Complexity Test,
12.	The Serial Test,
13.	The Approximate Entropy Test,
14.	The Cumulative Sums (Cusums) Test,
15.	The Random Excursions Test, and
16.	The Random Excursions Variant Test.

3.1.1 Modification and adaptation

After the first compilation of the program suite the modification mentioned above appear to be needed. The system couldn't create the results folders and files, many library package were not include, need some adaptation with the operating system .Then we change the system setting to debugging mode then we begin to trace every code line and it's function then detect how the system calculate every parameter value(s) and discover many codes tricks and found the critical values for every test and by the way maintain the require adaptations. After this step and as the user manual advise [9] we run the tester using model file so as to be sure that modification does not change the logic of the program and the result is that :(the model pass all tests with P-value =1 (max) for every test ; the model file name is data.e and it's report in appendix A.

3.2 Test bed environment

Three PC computers.(1.8 GHz) and one Lap Top (2 GHz) are used during the Experiments **,** the duration of one Experiment is approximately 30 Minutes, By using the model described in above section we were recorded a dialog in append mode file until the file size be greater than 13.2 MB. (Mandatory required). We save the recorded file as "DATAIN.TXT" which is used forty* times to be encrypted by Different Cryptographic Algorithms, and the outputs were saved in the files (d1.txt..d40.txt) which are converted to binary format so as to be enter one by one to StsGui (NIST Statistical Suite) package which was used to measure the randomness of each file. The output reports are summarized in the table 2. including the execution time of Encryption & Decryption processes in seconds. Table 3. reorganized table 2 in a new form (so as to display the Experiments results and facts). Appendix A. summarized the P-values of sample tests.

4.Results and Discussions

- From table 3. Observed that all algorithms used ECB chain mode are fail to pass any randomness test. The BlowFish algorithm also doesn't pass all tests.
- XOR256BLOCK fail in 4 tests (fail in one test is enough to say it is not random).
- AES(CFB), TEA, and XOR256STREAM success in all tests, and to get which the optimum one special table of compression is constructed containing execution time (table 4). And for more clarify a graph representation for the three algorithms applied to (16 randomness tests) is constructed into Fig.4 (use table 5. P-values).
- From Table 4. and Fig. 4. it is clear that TEA algorithm is the best one for VoIP encryption.

***** Forty times due to change of the algorithm method or its situations (Padding, Chain).

Table (2): StsGui Experiments results

FILE NO	ALGO	Padding	Chain	Frequency	Number of Tests pass	E.Time **
B1	AES(32)	ZEROS	ECB	0.000000	Zero	11
B2	AES(32)	BLANKS	ECB	0.000000	Zero	11
B3	AES(32)	PKCS7	ECB	0.000000	Zero	10
B4	AES(16)	ZEROS	ECB	0.000000	Zero	8
B5	AES(24)	ZEROS	ECB	0.000000	Zero	10
B6	AES(32)	PKCS7	CBC	0.249284	(15) *RUNS	9
B7	AES(32)	ZEROS	CBC	0.249284	(15) *RUNS	9
B8	AES(32)	BLANKS	CBC	0.249284	(15) *RUNS	10
B9	AES(32)	PKCS7	CFB	0.236810	(16)	9
B10	AES(32)	ZEROS	CFB	0.236810	(16)	9
B11	AES(32)	BLANKS	CFB	0.236810	(16)	9
B12	B.Fish	PKCS7	ECB	0.000000	Zero	14
B13	B.Fish	ZEROS	ECB	0.000000	Zero	14
B14	B.Fish	BLANKS	ECB	0.000000	Zero	14
B15	B.Fish	PKCS7	CBC	0.000000	Zero	14
B16	B.Fish	ZEROS	CFB	0.000000	Zero	14
B17	B.Fish	BLANKS	CFB	0.000000	Zero	16
B18	B.Fish	PKCS7	CFB	0.000000	Zero	15
B19	TEA	ZEROS	ECB	0.000000	Zero	10
B20	TEA	BLANKS	ECB	0.000000	Zero	10
B21	TEA	PKCS7	ECB	0.000000	Zero	10
B22	TEA	ZEROS	CBC	0.657933	(16)	10
B23	TEA	BLANKS	CBC	0.657933	(16)	10
B24	TEA	PKCS7	CBC	0.657933	(16)	11
B25	TEA	ZEROS	CFB	0.699313	(16)	10
B26	TEA	BLANKS	CFB	0.699313	(16)	10
B27	TEA	PKCS7	CFB	0.699313	(16)	10
B28	Xor256B	ZEROS	ECB	0.000000	Zero	14
B29	Xor256B	BLANKS	ECB	0.000000	Zero	15
B30	Xor256B	PKCS7	ECB	0.000000	Zero	15
B31	Xor256B	ZEROS	CBC	0.066882	(11) *C-sum,Fft,Apen,REV,Serial	15
B32	Xor256B	BLANKS	CBC	0.066882	(11) *C-sum,Fft,Apen,REV,Serial	15
B33	Xor256B	PKCS7	CBC	0.066882	(11) *C-sum,Fft,Apen,REV,Serial	15
B34	Xor256B	ZEROS	CFB	0.719747	(13) *Runs,Fft,Serial	14
B35	Xor256B	BLANKS	CFB	0.719747	(13) *Runs,Fft,Serial	14
B36	Xor256B	PKCS7	CFB	0.719747	(13) *Runs,Fft,Serial	13
B37	Xor256S			0.085587	(16)	21

** E.Time: Execution Time in seconds consumed to encrypt and decrypt (13.2 MByte) File
* :Fail in the following tests

Table (3): Facts Summery

FACT N0	EXPERMENTS	ALGORTHIM	CHAIN	FACT
1	B1..B3	AES	ECB	Pass "0" TESTS
	B12..B14	BLOWFISH		
	B19..B21	TEA		
	B28..B30	XOR256BLOCK		
2	B12..B14	BLOWFISH	ECB	Pass "0" TESTS
	B15		CBC	
	B16..B18		CFB	
3	B28..B30	XOR256BLOCK	ECB	Pass "12" TESTS
	B31.. B33		CBC	
	B34.. B36		CFB	
4	B6..B8	AES	CBC	Pass "15" TESTS
5	B9..B11	AES	CFB	Pass "all" TESTS
6	B22..B24	TEA	CBC	Pass "all" TESTS
	B25..B27		CFB	
7	B37	XOR256STREAM		Pass "all" TESTS

Table 4: AES,TEA,XOR256S

Algorithm	Frequency Test Result	Execution Time 13.2Mb	Execution Time for 500byte	Notes
AES(CFB)	0.236810	11 Sec	0.8 m sec	
TEA	0.699313	10 Sec	0.72 m sec	The Best
XOR256S	0.085587	21 Sec	1.52 m sec	

Table (5): XOR,AES,TEA Tests P-values of all tests

Test No	XOR256S	AES	TEA
1	0.08856	0.23681	0.69931
2	0.89776	0.11654	0.04011
3	0.12962	0.30413	0.35049
4	0.75976	0.77919	0.7749
5	0.63712	0.86769	0.83292
6	0.30413	0.01791	0.69931
7	0.85138	0.35049	0.53415
8	0.09658	0.79814	0.21331
9	0.85138	0.83431	0.12962
10	0.75976	0.22482	0.43727
11	0.00576	0.77919	0.88317
12	0.58521	0.31154	0.23276
13	0.68902	0.00316	0.60246
14	0.69931	0.24928	0.59555
15	0.18156	0.28967	0.59555

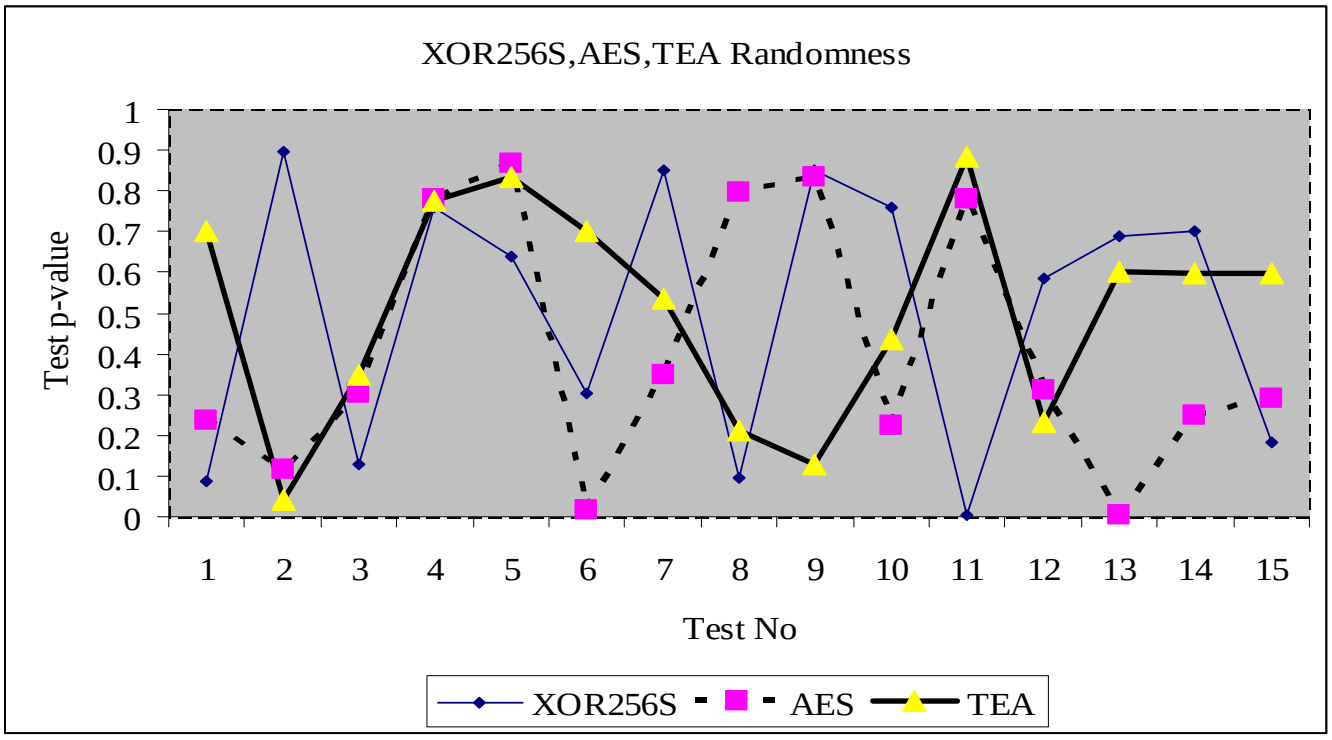

Figure (4): XOR,AES,TEA Tests

6. Conclusions:

This paper conclude that the TEA algorithm is more suitable to encrypt VoIP due to its encryption strength (more random) and it's smallest execution time which minimize the total end to end delay and has the least negative effect on QoS.

References:

[1] Praphul Chandra andLide, Wi-Fi TelephonyChallenges and Solutionsfor Voice over WLANs, AMSTERDAM • BOSTON • HEIDELBERG • LONDON, 2007.

[2] Amarandei-Stavila Mihai Voice over IP Security A layered approach, Amarandei-Stavila Mihai mihaia@gmail.com , XMCO consultants.

[3] L. Sun, SPEECH QUALITY PREDICTION FOR VOICE OVER INTERNET PROTOCOL NETWORKS, Ph.D. January 2004.

[4] An ISS Whitepaper, VoIP: The Evolving Solution and the Evolving Threat, MC-VOIPWP-1204,2004.

[5] Thomas Porter, PracticalVoIP Security, Syngress, 2007.

[6] Juniper Networks, Inc., Voice Over IP 101, www.juniper.net, Part Number: 200087-002 May 2007.

[7] Steven Sullivan,Securing a Converged Network

[8] D. Richard Kuhn, Thomas J. Walsh, Steffen Fries , Security Considerations for Voice Over IP Systems ,www.voipsa.com/Activities/taxonomy.php,NISTSP 800-58, January 2005 .

[9] Andrew Rukhin, Juan Soto, James Nechvatal,Miles Smid, Elaine Barker, Stefan Leigh,Mark Levenson, Mark Vangel, David Banks,Alan Heckert, James Dray, San Vo, A STATIISTIICAL TEST SUIITE FOR RANDOM AND PSEUDORANDOM NUMBER GENERATORS FOR CRYPTOGRAPHIIC APPLIICATIIONS, NIST Special Publication 800-22(with revision dated December 2000).

[10] A. Elbayoumy,S. Shepherd , A high grade secure VoIP system using the tiny encryption algorithm, Proceeding of 7 th Annual International Symposium on advanced Radio Technologies Boulder, Colerader,1-3 March 2005 .

Appendix A. summarized the P-values of sample tests

STATISTICAL TEST	P-VALUE model file	P-VALUE fail file	P-VALUE B6	P-VALUE B9	P-VALUE B22	P-VALUE B25	P-VALUE B31	P-VALUE B34	P-VALUE B37
frequency	1	0	0.249284	0.23681	0.65793	0.69931	0.06688	0.71975	0.08856
block-frequency	1	0	0.032923	0.11654	0.36692	0.04011	0.53415	0.65793	0.89776
cumulative-sums	1	0	0.678686	0.30413	0.15376	0.35049	0.24928	0.43727	0.12962
runs	1	0	0.051942	0.77919	0.5749	0.7749	0.23681	0.08559	0.75976
longest-run	1	0	0.911413	0.86769	0.41902	0.83292	0.22482	0.0156	0.63712
rank	1	0	0.911413	0.01791	0.20227	0.69931	0.88317	0.75976	0.30413
fft	1	0	0.202268	0.35049	0.61631	0.53415	0.00056	3.7E-05	0.85138
nonperiodic-templates	1	0	0.798139	0.79814	0.61631	0.21331	0.23681	0.61631	0.09658
overlapping-templates	1	0	0.816537	0.83431	0.22482	0.12962	0.53415	0.38383	0.85138
universal	1	0	0.090936	0.22482	0.12962	0.43727	0.61631	0.21331	0.75976
apen	1	0	0.55442	0.77919	0.88317	0.88317	0.14533	0.28967	0.00576
random-excursions	1	0.392456	0.77276	0.31154	0.63712	0.23276	0.63712	0.4528	0.58521
random-excursions-variant	1	1.392456	0.090936	0.00316	0.4686	0.60246	0.04281	0.31154	0.68902
serial	1	0	0.595549	0.24928	0.5749	0.59555	0.31908	0.22482	0.69931
linear-complexity	1	0	0.678686	0.28967	0.17187	0.59555	0.97807	0.9717	0.18156