Volume 6, Issue 7, July 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Implementation and Analysis of Sosemanuk Algorithm in Communication Data VOIP Networks Based Android

Desi Ramayanti*, Steven
Faculty of Compute Science, University of Mercu Buana,
Indonesia

Abstract*— Nowadays, along with the smart phones improvements, the development of information and communication technology has influenced the usage of VoIP. Although VoIP gives an affordable communication service to its users, it still has some weaknesses. One of the main concerns is the communication safety because VoIP uses computer network or internet as the medium which gives the chance to be eavesdropped. To overcome the problem, this research implements a method to encrypt voice data in Sipdroid, an Android based soft phone. Sosemanuk Algorithm was implemented to produce key stream before it was proceeded to the process of encryption or decryption. Waterfall method was also used to develop Sipdroid in this research. The encryption was processed inside a payload data from Sipdroid's RTP package. The implementation works well and Sipdroid's security service test based on Sosemanuk Algorithm successfully protects the voice data which was sent to the receiver or was received from the sender. In the future, research about the performance of Sipdroid and Sosemanuk Algorithm or Sipdroid and other encryption algorithm are expected.

***Keywords*— Sosemanuk, VOIP, Android, Encryption, Descryption**

I. INTRODUCTION

The information and communication technology is indeed developed fast nowadays, especially in the technology application and the communication networks. The infrastructure of these two is gradually becoming better in Indonesia as there are many telecommunication operators have applied 4G technology. The development of communication and computer technology has also influenced the development of VoIP technology or internet calls which allows voice communication through IP networks that include digitization and packeting voice signals [1]. A good and widespread signal allows VoIP to be implemented for a long distance thus communication service is cheaper. To use VoIP service, one of the tools needed is soft phone, software that can be connected to VoIP server and can be used to make and receive a call from VoIP networks.

The development in smart phones also gives positive impact to VoIP. Soft phone is well developed for the smart phone usage. One of free and open source soft phones in the internet is Siproid. Developed based on Android platform, Siproid application is currently available to download in Google Play, so it is easier for Android Smartphone users to communicate by VoiP.

One of VoIP advantages is to provide communication service at reasonable cost and easy to implement. However, one of its weaknesses is the security problem. VoIP communication which is carried out through IP networks or internet enables other irresponsible parties to eavesdrop. Therefore, there is a need to protect the communication in VoIP networks. One of them is to apply data encryption to the communication data produced by VoIP networks. Sosemanuk algorithm is one of the most suitable algorithms to encrypt and to decrypt types of data stream and software-oriented programs. [2]

Based on the background problems above, this research will formulate the research problems; the first one is how to implement Sosemanuk algorithm in Siproid soft phone to protect communication data in VoIP networks and the second one is how to test the result of Sosemanuk Algorithm implementation in Siproid soft phone.

The limitations of the research problems are as follows:

- Siproid used only suitable for Android version 3.0 or later.
- Encryption method to be implemented is only Sosemanuk Algorithm.
- The test bed is using WLAN which is built specifically for VoIP networks.

The purpose of this research is to implement Sosemanuk algorithm with Siproid soft phone to secure communication data. Meanwhile, the benefit of this research is to get a soft phone integrated with encryption module thus ensuring the security when communicating with VoIP.

Chapter I contains the research backgrounds, research limitations, research purpose, and research benefit. Chapter II contains theory-related research, such as Sosemanuk's explanations, and also Sipdroid. Chapter III contains the analysis and the design of the research. It analyzes the implementation of Sosemanuk algorithm in Sipdroid, especially during the process of encryption and decryption. Lastly, in Chapter IV, the conclusions and recommendations based on this research are provided.

II. THEORY AND METHODS

A. Related Research

This research is based on Rizal's finding when analysing Siproid's client about VoIP performance [3] and referring to [4]. In his writings, Siproid is integrated with encryption module of Java Cryptography Extension (JCE), by using AES, DES, and RC4 algorithm. Meanwhile in this research, Sosemanuk algorithm will be implemented as it is more suitable for data stream and software-oriented program. Finally, the result of the implementation will be tested.

B. Sosemanuk

Sosemanuk is a stream cipher algorithm which is software-oriented and synchronous [2]. Sosemanuk chiper uses two basic principles of SNOW 2.0 stream cipher and transformation of block cipher SERPENT. Sosemanuk means 'snow snake' from Cree Indian language. Sosemanuk algorithm has a key length from 128 up to 256 bits. Variable key length would still result in a security of 128 bits.

The flow of Sosemanuk Algorithm is as follow:

- LFSR has an initial value of s1 to s9, no value of s0. FSM initial values are R10 and R20.
- In the first step, R1, R2 and f1 are calculated from R10, R20, s2, s9 and s10 (s10 here represents the value that will be included in the shift register).
- The first step is resulted in the stored value of s1 and f1.
- In the first step, the feedback S11 is generated from s10, s4 and s1, and the values of the LFSR is updated, generating value of s2 to s11.
- The four first output value, z1, z2, z3 and z4 are generated from operating Serpent1 with (f4, f3, f2, f1) whose output is combined with XOR (s4, s3, s2, s1).

The initialization on Sosemanuk consists of two steps:

1. Key scheduling; the process is using the only secret key. This processing corresponds to the key scheduling of Serpent24.
2. Injection IV; processed by using the output of key schedule and IV; used to initialize the internal state of Sosemanuk. IV is worth 128 bits and is used as an input to the cipher block of Serpent24. Serpent24 has 24 rounds and output of lap 12, 18th and 24th are used.
   a. (Y312, Y212, Y112, Y012): the output from the 12th round.
   b. (Y318, Y218, Y118, Y018): output of round 18
   c. (Y324, Y224, Y124, Y024): output of round 24.

The values are then used to fill the internal state of Sosemanuk.

(S7, s8, s9, s10) = (Y312, Y212, Y112, Y012)
(S5, s6) = (Y118, Y318)
(S1, s2, s3, s4) = (Y324, Y224, Y124, Y024)
R10 = Y018
R20 = Y218

Fig 1 below is a scheme of Sosemanuk.

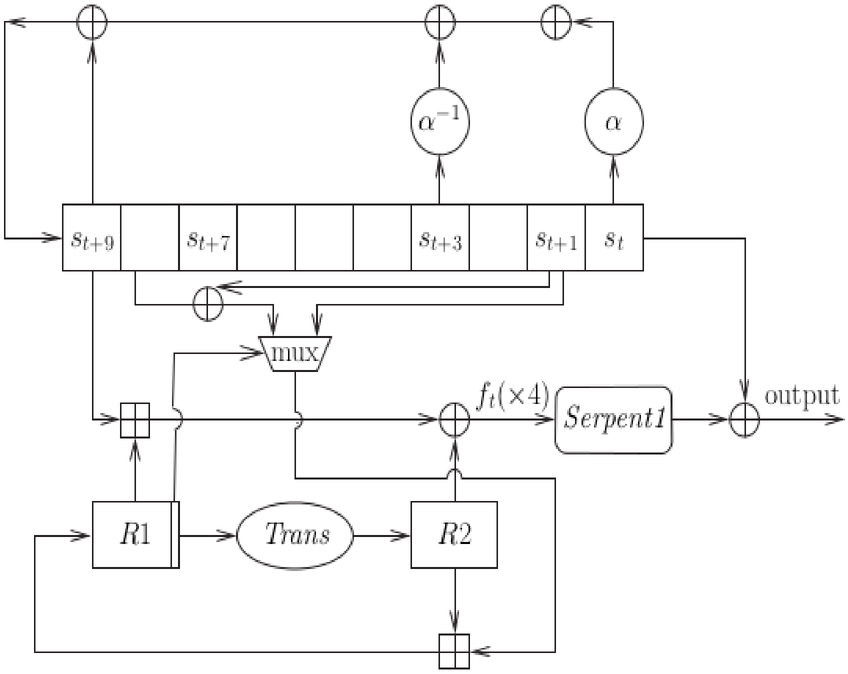

Fig 1- Scheme of Sosemanuk Algorithm [2]

Fig 2 below will describe the output transformation on four consecutive rounds of Sosemanuk.

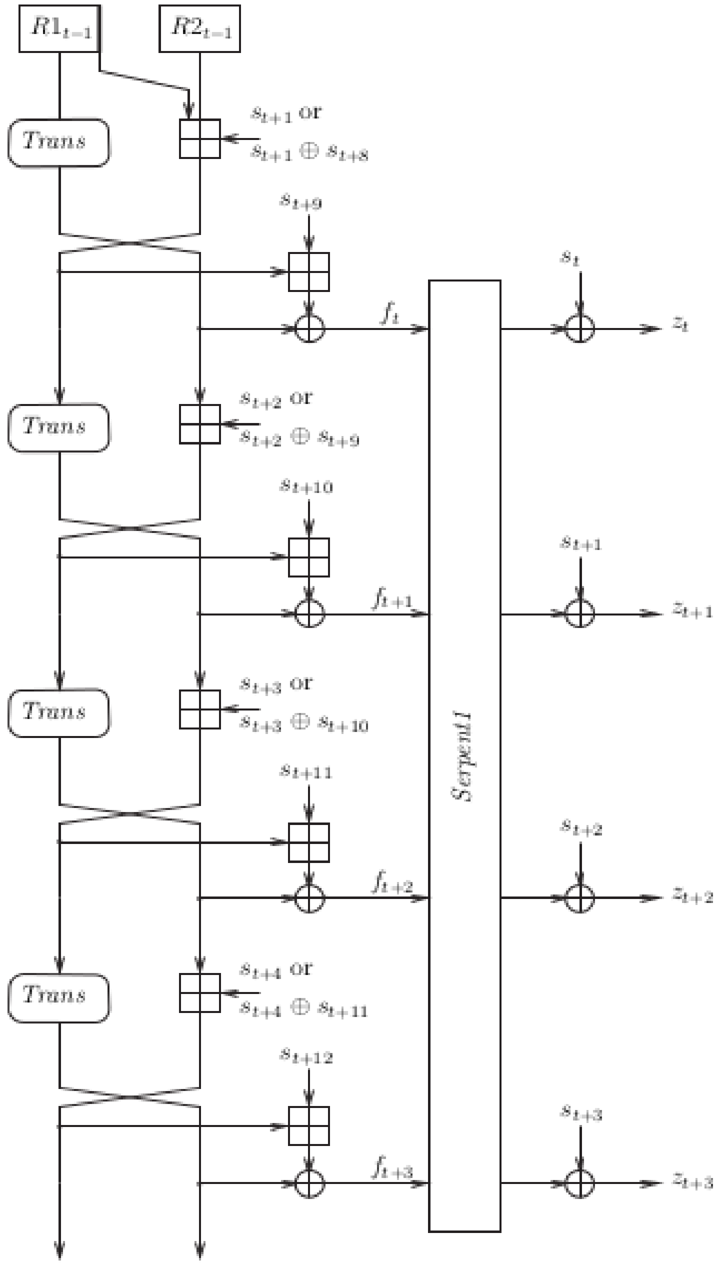

Fig 2 - Rounds in Sosemanuk Algorithm [2]

III. METHODOLOGY

At this stage, the steps required to implement the Sosemanuk algorithm into Sipdroid will be analyzed. In this research, a waterfall approach is used because the problem to be solved in this study has been well-defined [5]. Here are the steps undertaken by the waterfall approach.

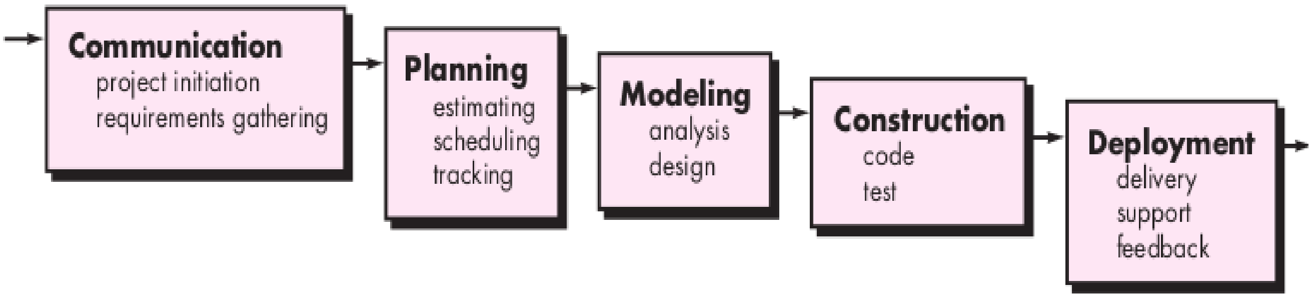

Fig 3- Steps of Waterfall approach

For more details, a flowchart that describes the methodology used to complete this study is shown in Fig 4 below.

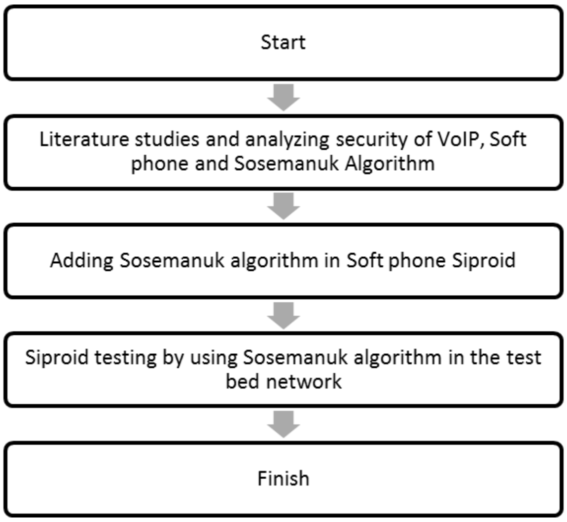
Start
Literature studies and analyzing security of VoIP, Soft phone and Sosemanuk Algorithm
Adding Sosemanuk algorithm in Soft phone Siproid
Siproid testing by using Sosemanuk algorithm in the test bed network
Finish

Fig 4 – Methods used in the research

a. Collecting Materials

This stage is used to gather requirements that are used to implement the algorithm Sosemanuk into Sipdroid.The first step is to gather the literatures related to the research as stated in Chapter II. The next step is to find program codes associated with this research. Program code sought is Sipdroid program code [6], and references to make Sosemanuk program code. Some hardwares such as wireless routers and smart phones are also needed.

b. Planning

This stage plans Sosemanuk algorithm implementation with Sipdroid. The plan is scheduled to perform this implementation in two months with the following description:

- Month 1.
  Analyzing Sipdroid program code and build Sosemanuk algorithms.
- Month 2
  Implementing Sosemanuk algorithm into Sipdroid and testing it.

c. Analyzing the implementation of Sosemanuk Algorithm in Siproid

Analysis conducted in implementing Sosemanuk algorithm has been done by Rizal [3], thus in this research the flow of encryption and decryption is modified by Sosemanuk algorithm.The pattern made for encryption process before sending voice data is described in Fig 5 below.

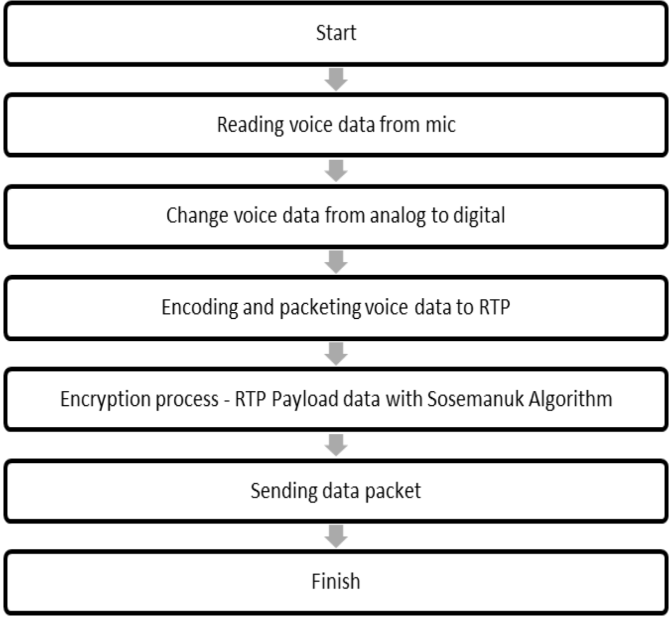
Start
Reading voice data from mic
Change voice data from analog to digital
Encoding and packeting voice data to RTP
Encryption process - RTP Payload data with Sosemanuk Algorithm
Sending data packet
Finish

Fig 5 – Encryption process in Siproid

The encryption process in Sipdroid begins with receiving voice data in analog form through the microphone. Then the voice analog data is converted into digital data by PCM in Sipdroid. The voice data soon is encoded and is converted to the form of RTP. Before being sent to the recipient, RTP packet is encrypted to payload the data, so the data carried by RTP was safe because it has been encrypted with Sosemanuk algorithm. Soon after that, the new data is sent to the receiver.

The decryption process when a data packet is received is described in Fig6 below.

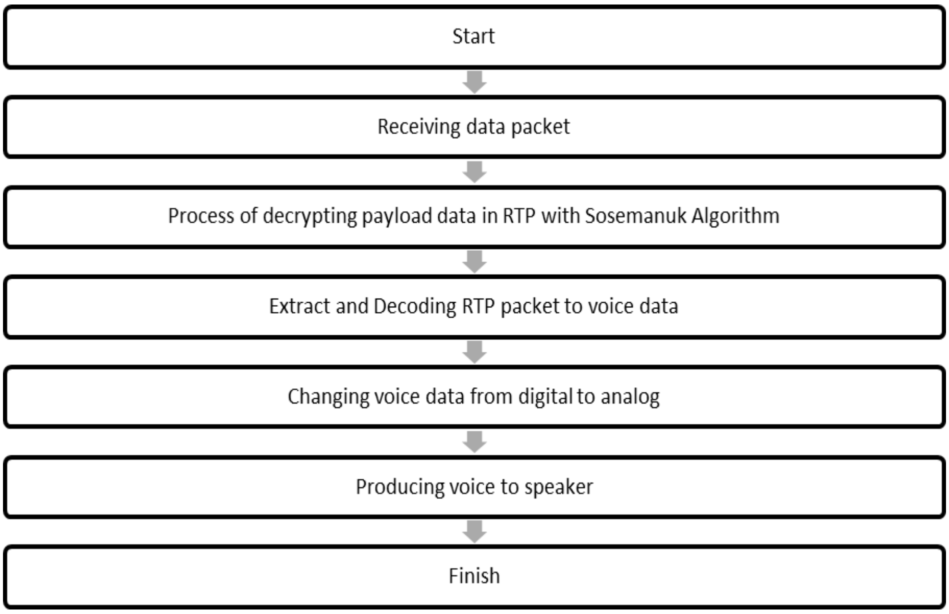
Start
Receiving data packet
Process of decrypting payload data in RTP with Sosemanuk Algorithm
Extract and Decoding RTP packet to voice data
Changing voice data from digital to analog
Producing voice to speaker
Finish

Fig 6 – Process of decrypting in Siproid

The decryption process performed in Sipdroid is conducted after Sipdroid receives data packets in RTP formats. Before the extracting process of Sipdroid packet, the decryption process for the payload data contained in RTP is performed using Sosemanuk algorithm. After that, the process of converting voice data from digital form to analog form is done so that the sound can be produced by the speaker.

d. Coding and Testing

Coding Stage

Encoding stages are stages to implement Sosemanuk algorithm into Sipdroid and also to change the look of Sipdroid in order to deal with the use encryption and decryption by using Sosemanuk Algorithm.

The stages of coding in this study are as follows:

1. Adding settings menu to set up the use of the algorithm and the key used for encryption and decryption.
2. Adding a special packet in Sipdroid to accommodate the class associated with Sosemanuk algorithm.
3. Implementing Sosemanuk program code.
4. Processing the encryption and decryption of data packets with Sosemanuk program code that has been made.

The explanation of the encryption process is as follows:

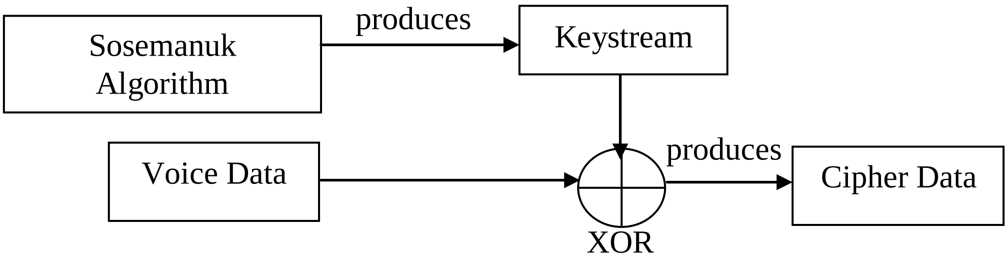

Fig 7 – Process of implementing encryption on data voice with Sosemanuk Algorythm

The encryption process in Fig 7 shows that Sosemanuk algorithm in this implementation is used to generate key stream. Key stream generated will be synchronized with the voice data so it will become a cipher data. This cipher data will be put back into the payload data from RTP which is ready to be sent to the recipient.

The implementation of decryption process is graphically described in Fig8 below:

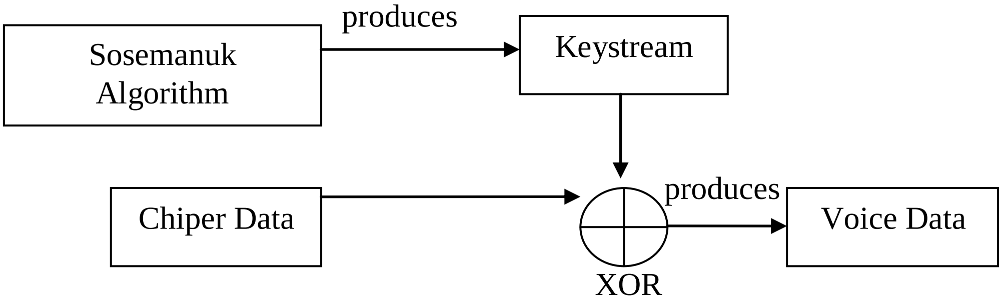

Fig 8 – Implementation process of decrypting voice data with Sosemanuk Algorithm

The decryption process performed in Fig 8 is using almost the same concept with the encryption process. The only difference is in the process of decrypting cipher data, which is synchronized with the key stream generated from Sosemanuk algorithm.

Testing Stage

Testing stage for the result of implementing Sosemanuk algorithm in Siproid will use the security services [7]. The security services are divided into six categories; Authentication, Access Control, Confidentiality, Integrity, Non-repudiation, and Availability.

The tests that use the security services in this research only use three parameters; the first one is the confidentiality data for privacy mechanisms or encryption; the second one is the integrity data for the incorporation of security or encryption mechanisms; and the third one is the availability of all resources. The uses of these three parameters refer to [8]. Other parameters are not used because the Authentication parameter is not examined in this research, which can also affect the parameters of non-repudiation and parameters of Access Control.

For testing this research, a test bed is constructed with VoIP networks built inside. The test bed network is created by using Wireless Local Area Network (WLAN) and the device used is a wireless router. A laptop or a PC is also used to be a VoIP server. Then two smart phones are used for communication or act as a VoIP client. Data are collected within 10 experiments and the talking time for each communication session takes 30 seconds. The test bed network built is illustrated in Fig 9 below:

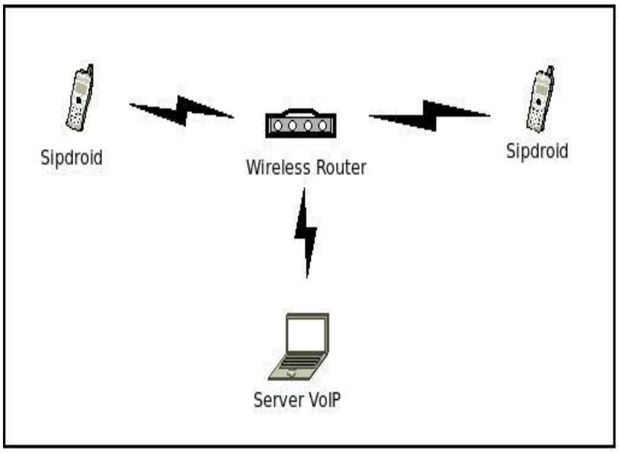

Fig 9 – The test bed used in the research

The explanations of each parameter tested by the security services are as follows:

1. Testing with Data Confidentiality Parameters

Testing with Data Confidentiality Parameters is intended to avoid the attack of passive attacker. Testing can be done with the help of the network protocol analyzer, for example, Wireshark. Wireshark can decode RTP streams captured, when passing through VoIP server, into a conversation audio record. The test results with the data confidentiality parameter can be seen in Table I below:

Table I Test Result With Data Confidentiality Parameters

No.	Types of Siproid	RTP Stream Decodes
1	Normal Siproid	Successful decoding
2	Sosemanuk Siproid	Unsuccessful decoding

The results of a successful RTP decode is done by using Wireshark to Sipdroid Normal shown in Fig 10 below:

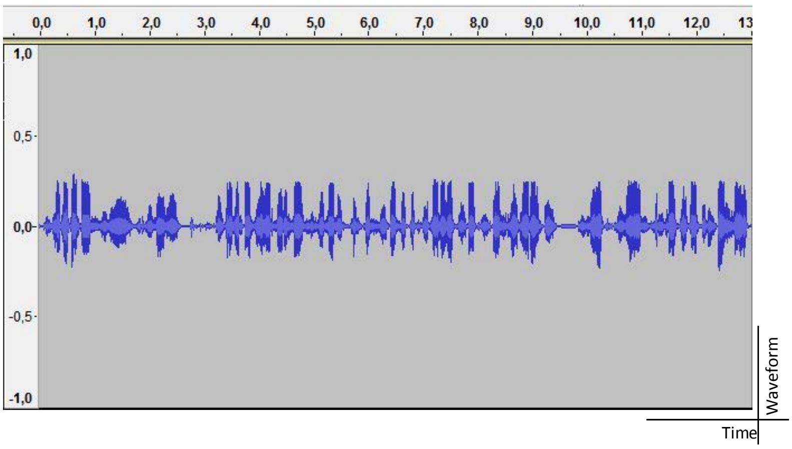

Fig 10 – The result of RTP Stream decoding for normal Siproid

Fig10 shows the results of the RTP stream decoding without any additional encryption. Spectrum of sound wave is visibly patterned when communication occurs. The resulting sound can be heard well. However, the results of decoding which are not successfully performed using Wireshark to Sipdroid with Sosemanuk are shown in Fig 11:

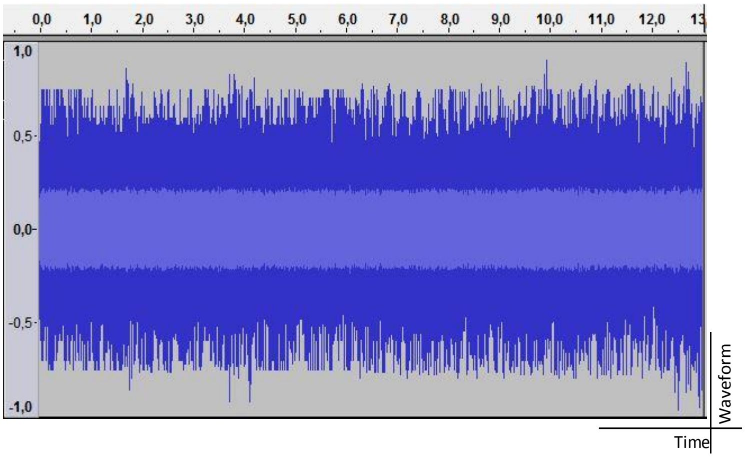

Fig 11 – The result of decoding RTP stream for Sipdroid with Sosemanuk algorithm

In Fig 11, the spectrum of sound waves looks irregular. The sound emitted is also irregular, only audible noise can be heard continuously.

2. Testing with Data Integrity Parameter

Testing with data integrity is intended to confirm whether the voice data transmitted through RTP packets can be well integrated with Sosemanuk encryption module being made. The test results with data integrity parameters can be seen in Table II.

Table II Test Result With Data Integrity Parameters

No.	Siproid’s partner	Communication Result
1	Normal Siproid with Normal Siproid	Successful communication
2	Normal Siproid with Sosemanuk Siproid	Unsuccessful communication
3.	Sosemanuk Siproid with Sosemanuk Siproid	Successful communication

Table II shows that Sosemanuk algorithm implementation with Sipdroid is successful by the fact that there is no successful communication between normal Sipdroid and Sosemanuk Sipdroid but a good communication occurs between Sipdroid that uses Sosemanuk encryption.

3. Tests with Availability Parameters

The test with the availability parameter is intended so that all existing resources and new resources added can be accessed by authenticated system users. The testing results with availability parameters are shown in Table III below:

Tabel III Test Result With Availability Parameter

R / **S**	Normal Siproid	Sosemanuk Sipdroid
Normal Sipdroid	Successful communication	Unsuccessful communication
Sosemanuk Sipdroid	Unsuccessful communication	Successful communication

*R = Receiver; S=Sender

The results of the test showed that Sipdroid with normal Sipdroid encryption module cannot communicate successfully.

IV. CONCLUSIONS

The conclusions that can be drawn from this research are as follows:

1. The implementation of Sosemanuk algorithm with Sipdroid has been successfully done by modifying the data payload of the RTP packet to be sent.
2. The tests that use security services on confidentiality data parameter indicate that the RTP packet with Sosemanuk Sipdroid cannot be decoded with wire shark, which ensures the privacy of transmitted data.
3. The tests that use the security service on the data integrity parameters show that Sosemanuk algorithm is successfully integrated well in Sipdroid.
4. The tests that use the security service on availability parameters indicate that the results of Siproid which is integrated with Sosemanuk algorithm can only communicate with other Sosemanuk Siproid.

ACKNOWLEDGMENT

For developing this research with adding another algorithm encryption that is suitable for the data stream types and Comparing the performance of the Sipdroid integrated with Sosemanuk algorithm with other Sipdroid integrated with different algorithm.

REFERENCES

[1] IP Telephony Voice Over IP (VoIP). http://www.cisco.com/c/en/us/tech/voice/ip-telephony-voice-over-ip-voip/index.html
[2] C. Berbain, et al. "Sosemanuk, A Fast Software-Oriented stream Cipher".
[3] R. Bahaweres, M. Alaydrus, A. Wahab. "Analisis Kinerja VoiP Client Sipdroid Dengan Modul Enkripsi Terintegrasi". Paper pada Seminar Nasional Aplikasi Teknologi Informasi 2012, Yogyakarta, Indonesia.
[4] Amin, A.H.M. "VoIP Performance Measurement Using QoS Parameter". Paper pada International Conference on IIT, Dubai UEA.
[5] R. Pressman. "Software Engineering: A Practicioner's Approach 7e".
[6] Sipdroid. https://github.com/i-p-tel/sipdroid
[7] ITU Rec. X.800. "Security Architecture For Open Systems Interconnection For CCITT Applications". Geneva, 1991.
[8] S. McGann, D.C. Sicker. "An Analysis of Security Threats and Tools in SIP-Based VoIP Systems". Paper pada VoIP Security Workshop 200s5, Washington DC, USA.