VOL. 14, NO. 12, JUNE 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A NEW SYSTEM TO ENCRYPT SPEECH SIGNALS USING METAHEURISTICS

Mohamed Kaddouri[1], Zakaria Kaddouri[2], Driss Guerchi[2], Mohammed Bouhdadi[1] and Said Oukacha[3]
[1]LMPHE Laboratory Mohammed V University, Faculty of Sciences Rabat, RP, Rabat, Morocco
[2]Department of Computers Science, Mohammed V University Abu Dhabi, Abu Dhabi, United Arab Emirates
[3]LETS/Geomat Laboratory Mohammed V University, Faculty of Sciences Rabat, Rabat, Morocco
E-Mail: m.kaddouri2@gmail.com

ABSTRACT

We present in this paper a new symmetrical metaheuristic speech encryption for secure communication. Our approach consists of shuffling the samples of a speech signal using a metaheuristic generated key before transmission. The need for real-time encryption systems for audio communication is increasing due to the widespread of real-time voice applications, such as voice over IP. To minimize any delay, the speech signal is segmented into frames that are encrypted on the spot and transmitted to the receiver. Once received at the receiver side, the speech frame is decrypted and concatenated with the previous speech frames. The objective and subjective measures show that our technique outperforms the existing block encryption algorithms in terms of execution time and security performance.

Keywords: speech encryption, metaheuristics encryption, speech cryptography, speech communication.

INTRODUCTION

To date the transmission of voice over the Internet uses the IP protocol on a large part of the Internet networks. This method of transmission can be vulnerable to different types of attacks (Endler et al., 2006), such as denials of service (DoS), listening, intercepting traffic, and modifying traffic by an attacker (Sisalem *et al*., 2006).

Any solution to these problems must have several levels of availability, integrity and confidentiality that comply with security requirements including its legal aspects (Delfs, 2007). To meet the security requirements, it is necessary to take into consideration a set of aspects to be secured, in particular the security of the operating system, the application and the VoIP server.

Among the current solutions, we mention two protocols for voice security such as the Transport Layer Security (TLS) (Dierks, 2008) and Secure Real-Time Transport Protocol (SRTP) (Baugher, 2004) whose main purpose is to ensure the confidentiality of data by using the most popular block cipher methods such as the advanced encryption standard (AES) (Daemen *et al.*, 2013), Data encryption standard (DES) (Standard, 1999) and Triple Data Encryption Algorithm (TDEA) (Barker, 2017).

Cryptography is a technique that protects communications (Mao *et al*., 2006) and makes them incomprehensible to unauthorized persons (Stinson, 2005); it aims to provide many security services such as confidentiality, authentication and non-repudiation integrity (Delfs, 2007).

Because of the nature of cryptography, new algorithms emerge continuously and that existing algorithms are attacked without interruption (Kuo *et al.*, 1991). An algorithm being considered strong today may be vulnerable tomorrow. Given this, it is essential to think of other strategies for designing encryption systems. Considerations should also be given also to performance (Erkin, 2007) since many uses of IPsec are in environments where performance is a concern (Nadeem *et al*., 2005).

We have presented in a recent work a new method of image encryption, which is mainly based on Vigenère algorithm and meta-heuristics (Kaddouri *et al.*, 2017). This method integrates two techniques in the encryption phase to generate two different keys. The first technique uses the Vigenère algorithm to widen the intensity domain of the image, and the second technique uses meta-heuristics to maximize the disorders at the pixel level (Kaddouri, 2014). The two generated keys are used for both the encryption and decryption phases (Kaddouri *et al*., 2017).

We have also proposed in the same work a new evaluation function, related to the encryption phase of meta-heuristics (Dréo *et al.*, 2006), to generate keys that are completely random and totally independent of the content to be encrypted message (Kaddouri *et al*., 2017). This approach showed very satisfactory results in terms of security and execution time. Randomness aspect of the ciphering methods renders the cryptanalysis process very complicated (Kaddouri, 2014). All these strengths are behind our main motivation to adopt the metaheuristic encryption method in real-time voice cryptography.

In this work, we propose a new speech symmetrical metaheuristic encryption (SSME) for secure speech communication. The paper is organized as follows. Section 2 describes the paper main idea. Section 3 presents the experimental results. Conclusion and evaluative discussion are given in section 4.

Speech Symmetrical Metaheuristic Encryption

The SSME consists of using the Metaheuristics encryption algorithm as the main block of the speech encryption. Before encryption, the analog speech signal is converted to discrete-time format using the standard conversion criteria. At the receiver side, a copy of the original speech is extracted from the encrypted signal using metaheuristics decryption followed by discrete-time

to analog conversion. Figures 1 and2show the building blocks of the proposed speech encryption-decryption technique.

Encryption

Speech Pre-processing

To perform speech encryption, the analog speech signal *s(t)* is to be first converted to discrete-time format *s[n]*. This phase is accomplished using sampling, a process that is performed according to some specific signal processing criteria in order to reconstruct a high-quality speech signal at the receiver end. In this process the speech samples *s[n]* are selected from the analog speech *s(t)* at regular time intervals that are multiples of the sampling period (Quatieri, 2006).

The analog speech signal is to be sampled at a sampling rate at least greater than the Nyquist rate (double the speech bandwidth). Higher sampling rate produces better reconstructed analog signal, but requires higher coding rate and larger storage space. According to Shannon sampling theorem, the sampling rates F_s should be at least greater than twice the speech Bandwidth B (Proakis, 2003).

The Encryption phase

As mentioned before, the encryption phase is preceded by the conversion of the analog speech signal to discrete-time format. The speech signal is then segmented into L-milliseconds-duration frames; each frame S_j consists of N samples $s_j[n]$ (n = 0, …, N-1), where the frame length $N = Fs \times L$.

The speech frames S_j are encrypted separately. The encryption algorithm permutes randomly the locations of the N speech samples $s_j[n]$, hence producing an encrypted discrete-time speech signal ES_j with samples $es_j[n]$.

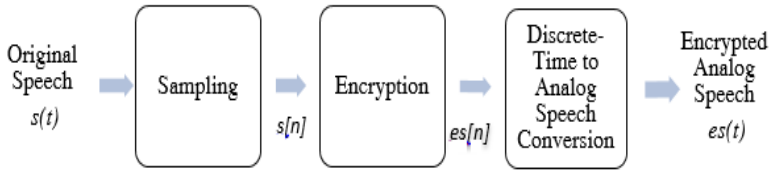

Figure-1. Speech Encryption Process.

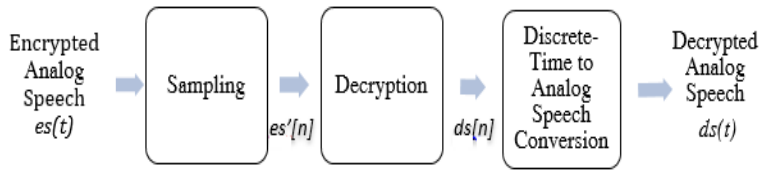

Figure-2. Speech Decryption Process.

The encryption key is generated and is used for J consecutive speech frames. To enhance the security level, the lifetime of each key is limited to J speech frames only. Once the encryption of the current speech frames finishes, the key is then updated for the next subsequent block of J speech frames.

The following algorithm gives the steps of the encryption phase:

Step 1: Convert the analog speech signal $s_j(t)$ to discrete-time format $s_j[n]$.

Step 2: Subdivide the discrete-time speech signal into blocks of J speech frames S_j.

Step 3: Generate a key K for each block of J consecutive speech frames. The key K consists of N samples $K(k)$, k = 0, …, N-1, representing the new locations for the current speech frame samples as illustrated in Figure 3.

Step 4: Apply consecutively the key K to each of the J speech frames.

- Shuffle the samples $s_j[n]$ of each speech frame S_j to their new positions: $es_j[n] = s_j[K(n)]$. This results in a new encrypted discrete-time speech frame ES_j .

Step 5: Convert the discrete-time speech frame ES_j into analog format, and then concatenate it with the remaining speech frames before transmission or storage.

Step 6: Repeat steps 3 to 4 until reaching the last speech block.

Figure-4 summarizes the encryption process for one speech frame S_j using the metaheuristic algorithm in the main phase of the encryption process. As mentioned in the above algorithm, the resulting encrypted speech will be subject to a reverse decryption process at the receiver side.

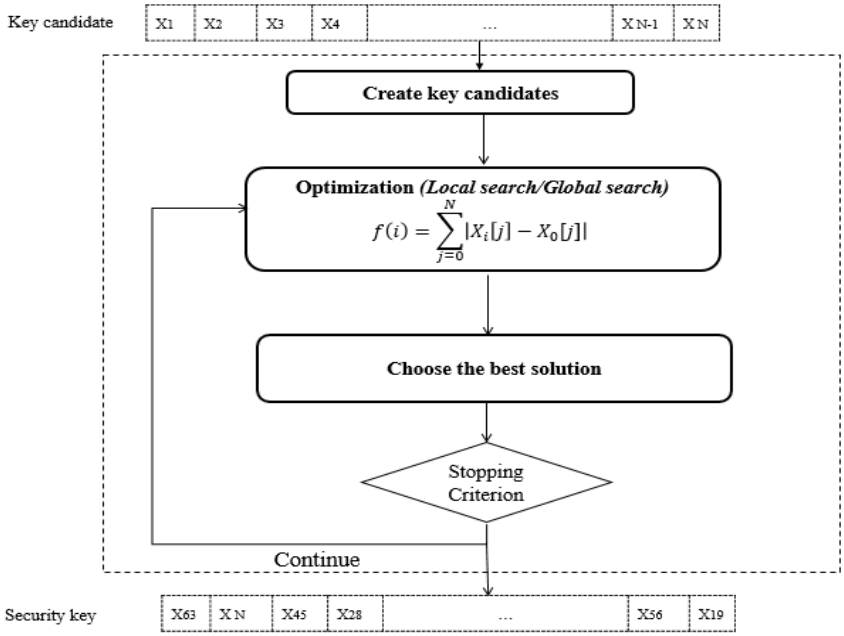

Figure-3. Encryption key Generation.

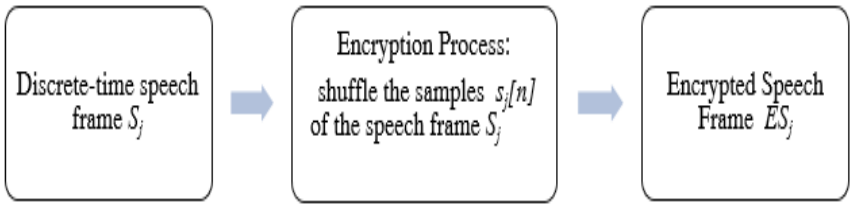

Figure-4. Encryption process for one speech frame S_j.

The Decryption phase

At the receiver side, the analog encrypted speech signal is first converted to discreet-time format using the same sampling frequency F_s. The discrete-time encrypted signal is then subdivided into blocks of J frames each. An encryption key is used for J consecutive speech frames only, but applied to each frame of the block sequentially.

The following algorithm illustrates the decryption phase steps:

Step 1: Convert the analog speech signal $es_j(t)$ to discrete-time format $es_j[n]$.

Step 2: Subdivide the speech signal into blocks of J speech frames ES_j.

Step 3: Retrieve the key K for each block of J consecutive speech frames.

Step 4: Apply consecutively the key K to each of the J encrypted speech frames ES_j

- Use the key samples $K(k)$ to relocate each speech samples $es_j[n]$ to its initial position. Hence generating the decrypted speech frame DS_j with samples $ds_j[n]$.
- Convert the discrete-time speech frame DS_j into analog format, and then concatenate it with the other speech frames.

Step 5: Repeat step 2 to step 3 until reaching the last speech block.

Figure-5 shows the building blocks of the decryption process.

Experimental Tests

Performance Measures

To measure the performance of our technique, we use various assessment tools. A part of the security robustness, we need to measure the similarity between the original speech and the decrypted speech at the receiver side. For this purpose, we used both objective and subjective speech quality assessment measures.

The objective measure consists of the average signal-to-noise-ratio (SNR). The SNR is the average of all the frames SNR_j. For a speech signal of M speech frames, the SNR is defined by:

$$SNR = \frac{1}{M}\sum_{j=0}^{M-1} SNR_j \quad (1)$$

while $SNR_j$is the signal-to-noise ratio for speech frame S_j.

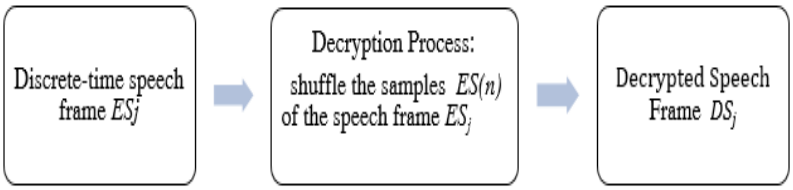

Figure-5. Decryption process for one speech frame S_j.

Comparison Original vs Encrypted Speech

To measure the discrepancy between the original and encrypted speech, we used informal listening as subjective measure and Signal-to-Noise Ratio (*SNR*) as objective measure.

The SNR_j for a speech frame j is defined as the ratio between the power of the original speech $s_j[n]$ and the power of the noise, $s_j[n]$- $es_j[n]$, generated by the encryption process.

$$SNR_j = \frac{\sum_{n=1}^{N} s_j^2[n]}{\sum_{n=1}^{N} \left(s_j[n] - es_j[n]\right)^2} \quad (2)$$

In the experimental tests we used a database of ten clean speech signals: five male and five female speech messages. In this work, we used a sampling frequency of 8 KHz with a speech frame length N varying from 5 milliseconds to 80 milliseconds. It is worth mentioning that the most common frame length in speech communication is 10 milliseconds or 80 samples for a sampling frequency of 8 kHz.

Table-1 shows the average *SNRs* for five female and five male speech signals for a fixed speech frame of 80 samples.

Table-1. The planning and control components.

	SNR
Female speech	**0.5230**
Male speech	0.5125
Average *SNR*	0.51775

Figure-6 displays an original speech signal and its encrypted version. The above *SNRs* supports the informal listening results that confirm that all the encrypted speech signals are unintelligible. It is worth noting the SNR between a signal and its exact copy is infinite. So reducing the *SNR* from infinity to a value close to zero proves the success of our encryption technique in producing a totally unintelligible speech signal.

In the second phase of our experiments, we have run 100 times the same encryption code using the same parameters: same speech file with the same speech frame length *N*. We performed this test to show the randomness of the generated encrypted speech after each code execution, hence proving the randomness of the encryption key. As shown in Figure-7, for a given speech signal, the *SNR* between the original speech and the encrypted speech vary after each run of the encryption code. The encryption index in the figure represents a sequential code execution order. The test is repeated for four different speech files.

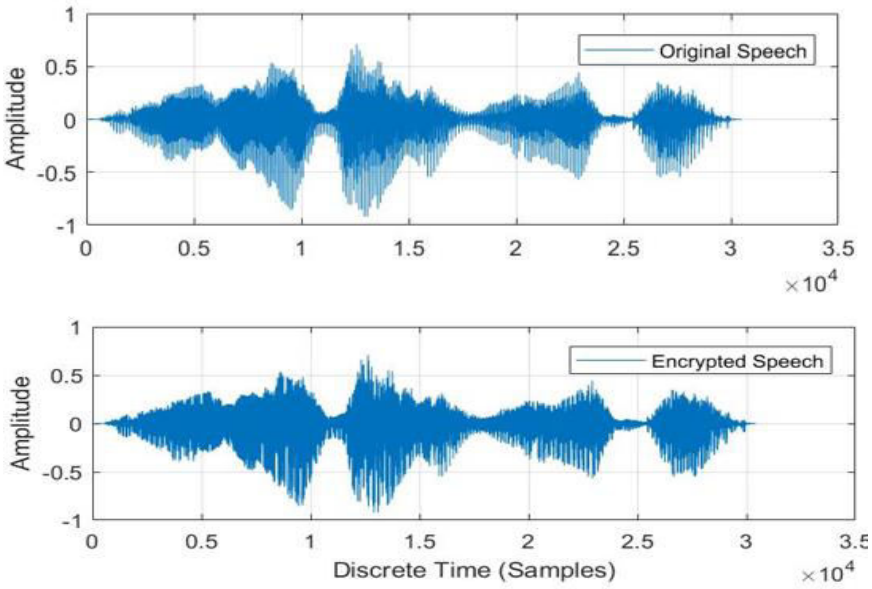

Figure-6. Comparison between the original and encrypted speech waveforms.

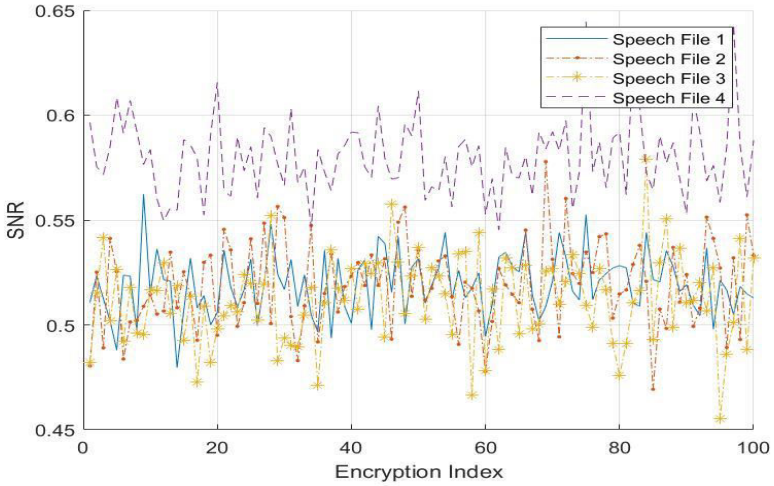

Figure-7. Variation of the SNR with the encryption index.

The *SNRs* between the speech file number four and its 100 encrypted versions are relatively higher than those for the three other speech files. A closer look at the shape of speech file 4 shows that this file contains mostly voiced speech with a quasi-periodic shape. Because of this periodicity, the permutation of the speech samples results in an encrypted speech with relatively higher correlation compared to that produced by the encryption of an aperiodic speech signal

In terms of security, not only does our encryption technique produce an unintelligible encrypted speech signal, but repetitive encryptions of the same speech with the same encryption parameters generate different unintelligible encrypted speech signals (different *SNRs*), hence rendering the cryptanalysis more difficult (there is no repeated pattern that can help in the cryptanalysis process).

In the third phase of our experiments, we carried out other testing to study the impact of the length speech frame on the overall *SNR*. We applied five times the same encryption code on the same speech file but with different speech frame lengths in each code execution. These lengths are 40, 80,160, 320, and 640.

Figure-8 gives the *SNR* as a function of the frame length for four speech files.

For a given speech file, there is an abrupt drop of the *SNR* between the case $N = 40$ and other speech frame lengths. This can be explained by the fact that within a small speech frame, the permutations are applied to speech samples that are still highly correlated. For this reason, one should select carefully the speech frame length to avoid such kind of weaknesses.

For N greater than 40, the average *SNR* vary slowly with the speech frame length.

Comparison Original vs Decrypted Speech

While the first set of experiments tests the dissimilarities between the original and encrypted speech,

the objective of the decryption process is to reproduce a decrypted speech that is comparable to the original speech signal.

The SNR_j for each frame is defined as the ratio between the power of the original speech and power of the noise generated by the encryption-decryption processes.

$$SNR_j = \frac{\sum_{n=1}^{N} s_j^2[n]}{\sum_{n=1}^{N}\left(s_j[n] - ds_j[n]\right)^2} \quad (3)$$

Table-2. The planning and control components.

	SNR
Female speech	∞
Male speech	∞

The objective measures show that regardless of the type of speech file and frame length, the average *SNR* has always an infinite value. This result was expected since using the same key at the encryption and decryption phases will generate, as show in Figure-9, a decrypted speech that is an exact copy of the original speech.

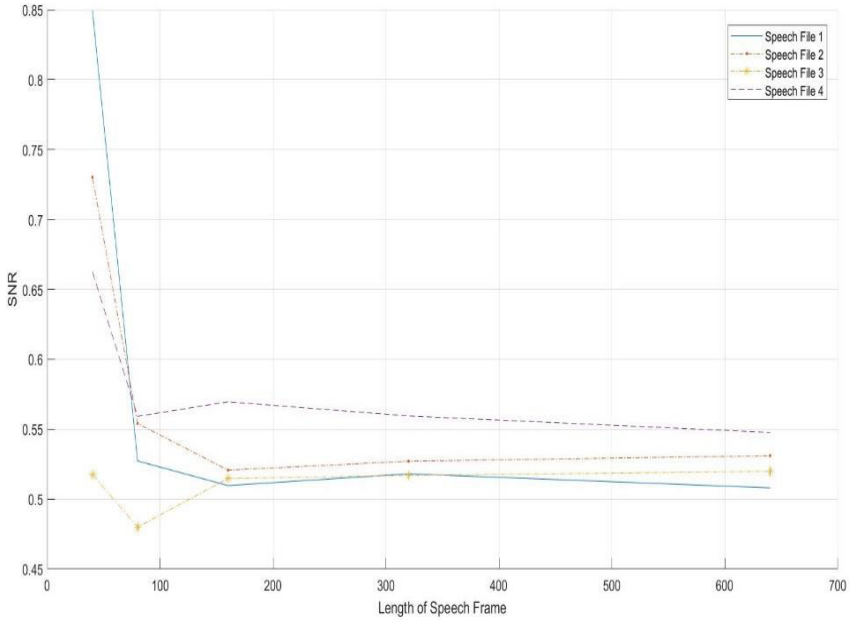

Figure-8. Variation of the SNR with the speech frame length.

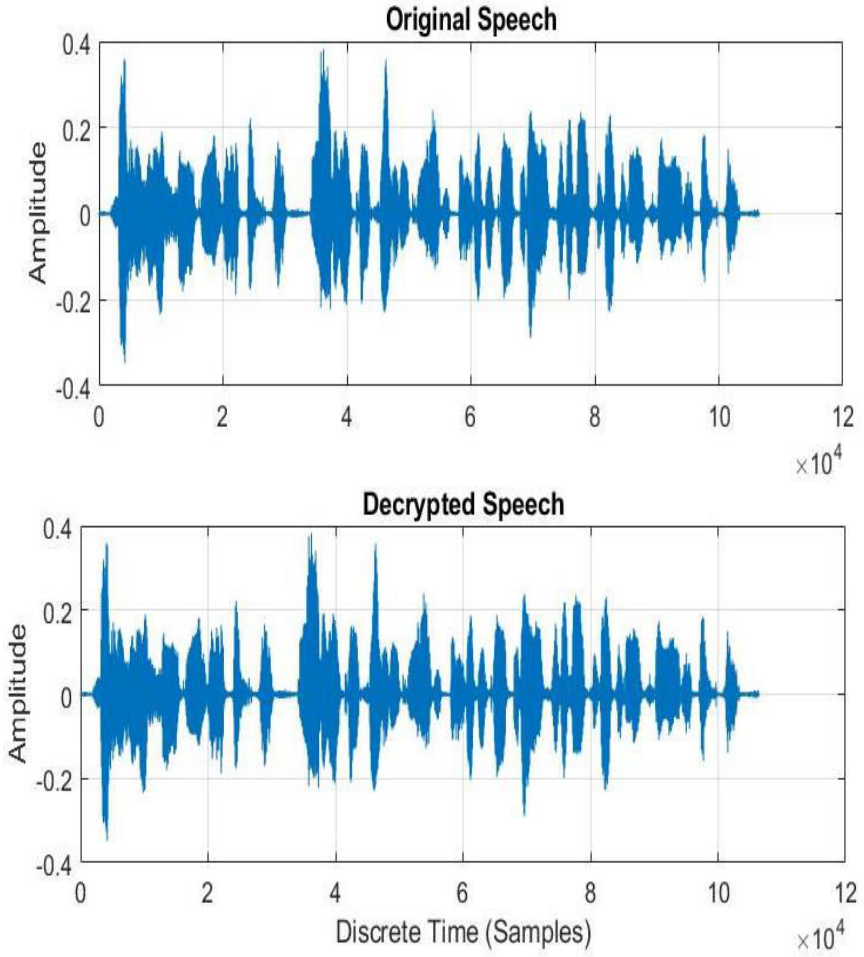

Figure-9. Comparison between the original and decrypted speech waveforms.

Execution time

Table-3 compares the execution time (in milliseconds) of SSME with that of the 3DES and AES encryption systems.

Table-3. Execution Time.

Block size (bytes)	DES	3DES	AES	SSME
20480	0.88	3.08	1.76	0.53
46080	1.6	5.6	3.2	1.21
69632	2.31	8.085	4.62	1.06
158720	5.44	19.04	10.88	3.87
191488	6.52	22.82	13.04	4.22

The results show that our encryption method requires less execution time for the different block sizes as compared to the other encryption standards.

If we take the case of DES, which is the most efficient in terms of execution time compared to the 3DES and AES, our cryptosystem outperformed even this system.

These results lead to the statement that the cryptosystem based on the metaheuristics has proved a second time its paramount importance for the encryption of the IP voice.

Security key

The protection provided by a symmetric encryption algorithm is related to the length of the key that is expressed in bits. In fact, the length of the key quantifies the maximum number of operations needed to find the right key. It is therefore an essential point for the security of the system. As mentioned before, our algorithm generates keys with limited duration. To encrypt a phone conversation of duration t seconds, our algorithm generates $K_t$keys. Table-4 gives the number of keys generated for different speech durations and frame lengths.

Table-4. Variation of the generated number of keys with the speech duration and frame lengths.

	Speech Duration *t*				
Frame Length *N* in samples	**1s**	**10s**	**1 min**	**10 min**	**30 min**
40	20	200	1200	12000	36000
80	10	100	600	6000	18000
160	5	50	300	3000	9000
320	2	25	150	1500	4500
640	1	12	75	750	2250

The results in Table-4 are obtained using the following general formula that relates the three parameters: the number of keys K_t, the speech duration t (in seconds), and the speech frame length N (in samples).

$$K_t = \text{rounddown}\left(\frac{800}{N}t\right) \quad (4)$$

The function rounddown was used to render $K_t$an integer.

Let's denote by $K_N$the maximum number of different keys generated by our algorithm, P_A the probability to find the right key, and by the brute-force attack (BFA).

The probability to find the right key $P_A$is related to the maximum number of keys K_N by the expression:

$$P_A = \frac{1}{K_N} \quad (5)$$

Since in our algorithm each speech sample is allocated 8 bits and each speech frame of length N samples is encrypted by one key, the key size is defined by

$$\text{Key Size} = 8N \text{ bits} \quad (6)$$

Table 5 gives some examples using the two above formulas.

Table-5. Recapitulative presentation of the results of the keys size generated by our algorithm and the complexity of the brute-force attack.

Frame Length in Samples	K_N	P_A	**Key Size (in bit)**	**Complexity of BFA**
40	40!	$1.23\ e^{-48}$	320	2^{320}
80	80!	$1.40\ e^{-119}$	640	2^{640}
160	160!	$2.12\ e^{-285}$	1280	2^{1280}
320	320!	$4.73\ e^{-665}$	2560	2^{2560}
640	640!	$1.55\ e^{-1520}$	5120	2^{5120}

With the modern computing systems, we are able to test several thousands or millions of keys per second. Any algorithm generating a small key will not be able to withstand such an attack for a long time.

It is important to note that the smallest key generated by our algorithm (320 bits) is greater than the current standard AES key 256-bit, which is impossible to break using the brute force attack and the current computing technologies (Daemen *et al,* 2013). However, we have to mention that the rapid evolution of computing systems may render a current encrypted message easy to attack in the near future (Li *et al*., 2008). To overcome this vulnerability, we advise the potential users of this algorithm to choose large-size keys.

Conclusions and Future Work

We introduced in this paper a new technique to encrypt speech signals. Not only is our technique security-robust, but it also requires a short execution time while producing an encrypted speech signal that is unintelligible.

In future works, we plan to study the noise performance of our technique under different transmission environments. It is worth mentioning that the transmission noise has no impact on the security robustness of our approach. The encrypted speech will still remain unintelligible. However, it is expected that the decrypted speech at the receiver will be slightly different from the original speech, but still intelligible. One of the techniques that we propose in the future work to combat the transmission noise is to combine speech encryption with channel coding.

REFERENCES

Barker E. 2017. SP 800-67 Rev. 2, Recommendation for Triple Data Encryption Algorithm (TDEA) Block Cipher. NIST special publication. 800, 67.

Baugher M., McGrew D., Naslund M., Carrara E. & Norrman K. 2004. The secure real-time transport protocol (SRTP) (No. RFC 3711).

Daemen J. & Rijmen V. 2013. The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media.

Delfs H. and H. Knebl. March 2007. Introduction to Cryptography: Principles and Applications. Springer.

Dierks T. 2008. The transport layer security (TLS) protocol version 1.2.

Dréo J., Pétrowski A., Siarry P. & Taillard E. 2006. Metaheuristics for hard optimization: methods and case studies. Springer Science & Business Media.

Endler D. & Collier M. 2006. Hacking exposed VoIP: voice over IP security secrets & solutions. McGraw-Hill, Inc..

Erkin Z., Piva A., Katzenbeisser S., Lagendijk R. L., Shokrollahi J., Neven G. & Barni M. 2007. Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing. EURASIP Journal on Information Security. 2007, 17.

Kaddouri Z. 2014. Design of new techniques to computer security based on evolutionary algorithms and hash functions.

Kaddouri Z., Hyaya M. A. & Kaddouri M. 2017. A New Cryptosystem using Vigenere and Metaheuristics for RGB Pixel Shuffling. Coordinates. 25(4): 255.

Kuo C. J. & Chen M. S. 1991, October. A new signal encryption technique and its attack study. In Security Technology, 1991. Proceedings. 25th Annual 1991 IEEE International Carnahan Conference on (pp. 149-153). IEEE.

Mao Y. & Wu M. 2006. A joint signal processing and cryptographic approach to multimedia encryption. IEEE Transactions on Image Processing. 15(7): 2061-2075.

Nadeem A., & Javed M. Y. 2005, August. A performance comparison of data encryption algorithms. In Information and communication technologies, 2005. ICICT 2005. First international conference on (pp. 84-89). IEEE.

Proakis J. G. 2003. Companders. Wiley Encyclopedia of Telecommunications.

Quatieri T. F. 2006. Discrete-time speech signal processing: principles and practice. Pearson Education India.

Sisalem D., Kuthan J. & Ehlert S. 2006. Denial of service attacks targeting a SIP VoIP infrastructure: attack scenarios and prevention mechanisms. IEEE Network. 20(5): 26-31.

Standard D. E. 1999. Data encryption standard. Federal Information Processing Standards Publication.

Stinson D. R. 2005. Cryptography: theory and practice. CRC press.