Secure Systems Design and Technology Development. Veliko Tarnovo, Bulgaria
Proceedings of the International Scientific Conference, 2026. Volume I, 363-367

Verifying Aspects an "Brute Force" Attack on 256 - AES Encrypted Voice Stream

Elitsa Petkova
Directorate "C4I Systems Development"
Bulgarian Defence Institute
Sofia, Bulgaria
ellydpetkova@gmail.com

Mario Angelov
Directorate "C4I Systems Development"
Bulgarian Defence Institute
Sofia, Bulgaria
m.angelov@di.mod.bg

***Abstract* — The rapid development of quantum technologies leads to new possibilities applying certain cryptographic attacks against symmetric algorithms. This paper discusses a verifying method an different approach to applying an "Brute force" attack on 256 - AES encrypted voice stream. Verification is performed after the correct decryption is selected from the set of decryptions. The purpose of verification is that if the selection module has accepted an incorrect decryption as correct, then the error is registered during verification. Unlike the classic application of a "Brute force" attack, which emphasizes computing power, here the focus is on signal processing.**

Keywords — AES, attacks, Brute force, voice streams, cryptanalysis.

I. INTRODUCTION

AES is a stream cipher approved by NIST [1]. The classic scenario for applying a "Brute force" attack on encrypted data with 256 – AES requires 2^{256} calculations. Providing a computational resource that provides the necessary calculations is only one of the tasks that need to be performed for the attack to be practical. The other significant problem is selecting the correct decryption from the set of such. From the point of view of transmitting encrypted voice streams in wireless networks, there is a practical approach for applying a "Brute force" attack, applied in a reduced form, developed in the Matlab 2019a – Simulink environment. It should be published. At present, there is no developed cryptanalytic approach, except for "Brute force", requiring only ciphertext. The developer of the "Twofish" algorithm - one of the most popular world-renowned cryptographers - Bruce Schneier, [2] claims that it is impossible to construct an attack that will lead to the deciphering of traffic. The attacks developed so far represent "academic cracking of the cipher". In fact, from the point of view of voice streams transmitted over a wireless network, a practical and feasible attack would be the one that requires only the ciphertext.

AES is resistant to classical differential cryptanalysis, but is susceptible to related key attacks. The published cryptanalysis methods applied to the AES block cipher require both ciphertext and a certain number of corresponding plaintexts, and also have requirements for the key [3-10]. A variant of differential cryptanalysis is presented in [11], in which the cipher is split into two parts, each part computed with partial knowledge of the key. In the middle, the attacker computes guesses for the key, checking for matches. A different but original attack is presented in .[12]. It operates on the principle of inducing random errors in the cipher state matrix. The induction leads to defective ciphertexts, which allows the full key to be deduced. From a practical point of view, a "Brute force" attack would be real and practical, since it only requires ciphertext. The disadvantage in this case is the need for computational resources, which depends on the development of technologies.

There are two typical ways of applying a "Brute force" attack on voice stream::

- classical - if we have a speech stream consisting of *"n"* consecutive encryptions, it follows that decrypted messages must be listened to, from which the correct one can be chosen. If we denote the listening time for one encryption by "t_n" then the listening time for the entire message would be $t_{full} = n.2^{256}$. This requirement makes it impossible to implement the attack in the classical scenario.
- by selection and subsequent verification – method presented in this publication. The main idea is to perform selection not on the entire encrypted voice message, but only on one encryption - 128 bits. The goal is to save both computational and time resources. Similarly, verification will also require one

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol1.223

encryption. If the first attempt at selection and verification generates a result, then the execution time of the attack will be $t_{perf} = t_{sel} + t_{ver}$, where t_{sel} denotes the time required for selection, and t_{ver} denotes the time required for verification. If "m" attempts are required to generate a result, then the time required for the attack will be $t_{full} = m.t_{perf}$. This attack variant is fully feasible if there is a technology that can perform 2^{256} calculations.

This article is structured as follows - in the section "functional model" the methodology for selection and subsequent verification is discussed, in the module "experimental part" some special cases of possible wrongly selected sequences are discussed, which confirm the need for verification. The module "results" gives a generalized assessment of the experiments performed, and in the "conclusion" the motives for conducting the present study are discussed.

II. FUNCTIONAL MODEL

In order to design a properly functioning selection algorithm, multiple decryptions of voice streams have been considered and analyzed. Within a decryption consisting of 16 integer values corresponding to 128 bits, we mark a given current integer value with *"n"* and the next one with *"n + 1"*. Then the distance between them - Δ_n represents the sum or difference or absolute value of the difference of two adjacent values or of their absolute value.

As a result of experimental data analysis, the highest value of Δ_n in decryptions with a "wrong" key is Δ_n =7808, while in those with a correct key it is of Δ_n = 1170, encountered once in a single decryption. If we set an upper bound for delta Δ_n, we can implement an algorithm based on the following postulate - if for a given decrypted sequence consisting of sixty integer values, there is more than one 1300 > Δ_n >= 1200, then the corresponding sequence was not decrypted with a correct key. Experimental results for calculating all Δ_n values when applying a reduced "Brute force" attack for one encryption and six parallel decryptions are presented in Table 1. A reduced "Brute force" attack involves six parallel decryptions, the first five decrypted with a pseudo-random key, the sixth decryption with a real key.

The principle of operation of the verification module is the same as the selection module If the selection module misses an incorrect decryption, the probability of the same thing happening during verification - another sequence decrypted with the same key - is negligible. Both modules are executed using the following program code:

```
provX = 0;
for i = 1:15
      if (InX(i+1)<=0 && InX(i) >=0)
            minX(i) = abs(InX(i+1)) +
InX(i);
            if minX(i) > 1300
                provX = 2;
            end
            if minX(i) >= 1200
                provX = provX + 1;
            end
      elseif (InX(i+1)>=0 && InX(i)
<=0)
            minX(i) = InX(i+1) +
abs(InX(i));
            if minX(i) > 1300
                provX = 2;
            end
            if minX(i) >= 1200
                provX = provX + 1;
            end
      elseif (InX(i+1)<=0 && InX(i)
<=0)
            minX(i) = abs(abs(InX(i+1))
- abs(InX(i)));
            if minX(i) > 1300
                provX = 2;
            end
            if minX(i) >= 1200
                provX = provX + 1;
            end
      elseif (InX(i+1)>=0 && InX(i)
>=0)
            minX(i) = abs(InX(i+1) -
InX(i));
            if minX(i) > 1300
                provX = 2;
            end
            if minX(i) >= 1200
                provX = provX + 1;
            end
      end
 end
if provX <= 1
        Out1 = Out(X:X+15,1);
else
        Out1                          =
int16([0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
]);
end
```

Software implementation of the selection method

TABLE 1 REDUCED "BRUTE FORCE" ATTACK FOR ONE KRANDOM ENCRYPTION

Output sequence	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	Δ_8	Δ_9	Δ_{10}	Δ_{11}	Δ_{12}	Δ_{13}	Δ_{14}	Δ_{15}
1	30	3579	5408	1875	74	127	1636	416	2350	274	1468	1643	10	131	1252
2	1726	1726	3456	3648	2110	3102	2983	38	68	34	93	546	880	503	281
3	1299	1263	107	208	2026	2157	12	3845	3877	75	51	26	245	368	900
4	1109	29	65	53	3322	2384	1033	227	220	447	1991	1656	1208	3200	1812
5	1558	1196	146	230	580	180	43	191	756	714	42	86	394	128	408
6	0	24	30	28	23	14	36	38	45	32	26	32	24	32	24

The algorithm of the "Brute force" attack functionality, including selection and verification blocks, is presented below in fig. 1. There is a counter with a zero initial value, which increases the value by one upon selection. After selecting a potential decrypted sequence, a verification process follows. If the potential sequence turns out to be real, the result is visualized on a display, otherwise the counter is reset and the attempts continue.

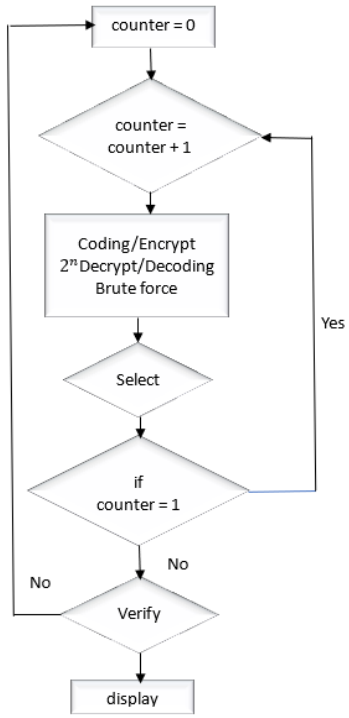

Fig. 1. Functional model of application of brute force attack with selection and subsequent data verification

III. EXPERIMENTAL PART

A "Brute force" attack is implemented in a reduced form in a Matlab 2019a – Simulink environment and should be published. The input sequence is a voice stream in the form of integer values. The output sequence after the overall signal processing also represents integer values close to or equal to the input. The small differences are due to the encoding and decoding processes.

The principle of operation of the verification module is the same as the selection module. The selection module is designed so that there is a possibility that it will select a sequence decrypted with the wrong key, i.e. it is possible to make an error. The assumption is that if decryption with an incorrect key produces an integer result that passes the selection module successfully, then the probability of this happening again when decrypting the next 128-bit sequence with the same key during verification is negligible. The model is designed so that the module only generates a result if the result is verified. The purpose of the experiment is to assess the effectiveness of the selection process and the need for a verification process.

We will look at several scenarios that confirm the need for verification

Hypothesis 1 – We perform a process of encryption and decryption of a given 128 bits sequence with the same key. The input and output sequences match. After the decryption process, we invert the least significant bit of the sequence in order to simulate a random result of applying a "Brute force" attack, similar to the real one. Fig. 2 shows a graph of the input and modified output sequence. It successfully passes the selection module.

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol1.223

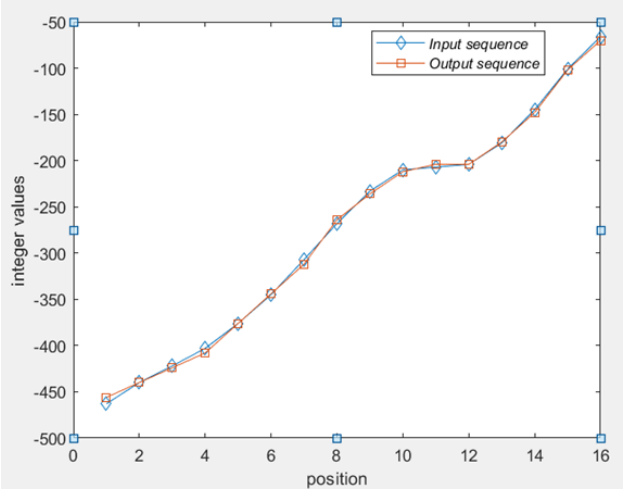

Fig. 2 Input and selected output sequence according to hypothesis 1

Hypothesis 2 – we repeat the scenario as for each integer value out of a total of 16 such for an encryption corresponding to 128 bits, we invert the least significant bit. The implementation of the hypothesis is presented with the following algorithm in the form of program code:

```
nn = 8;
        while nn <=128
              y1(nn, 1) = ~y1(nn, 1);
              nn = nn+8;
        end
```

Software implementation of a hypothesis 2

Fig. 3 presents a graph of the input and modified output sequence. Analogous to hypothesis 1, despite the difference in the number of inverted bits, the output sequence successfully passes through the selection module.

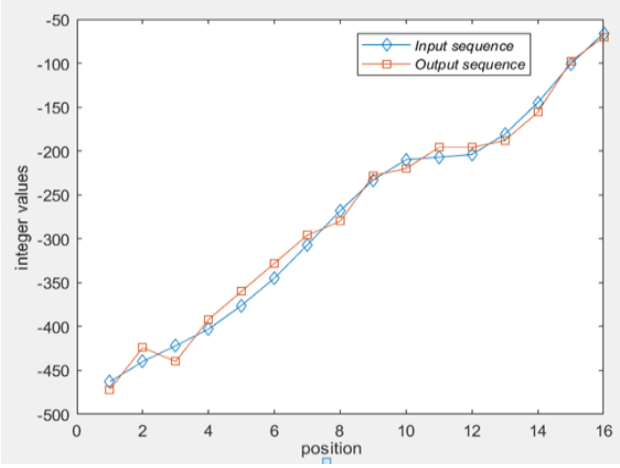

Fig. 3 Input and selected output sequence according to hypothesis 2

Hypothesis 3 – for each integer value out of a total of 16 such for an encryption corresponding to 128 bits, we invert the most significant bit. The result is presented in Fig. 4

The huge difference in the values between the input and the selected output sequence is due to the G.711 codec. In the experimental model, it performs the functions of encoding and decoding. In it, the most significant bit of a given binary equivalent of an integer value is determined by the sign in front of the value. In this case, all the most significant bits of the entire binary sequence are inverted. The input and output sequences are mirrored with respect to the "X" axis. The output sequence passes successfully through the selection module.

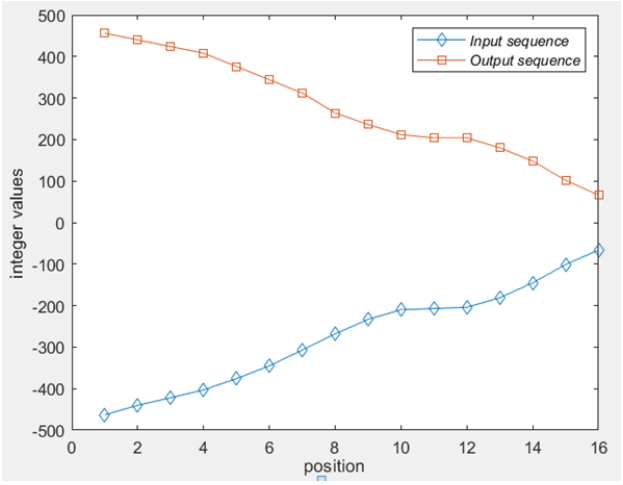

Fig. 4 Input and selected output sequence according to hypothesis 2

It is impossible to predict which of all 2^{256} the key options when applying a brute force attack would generate output sequences, according to the scenarios described above and many others. Fig. 5 only visualizes what two consecutive sequences would look like - the first selected and the second rejected during verification. It is visible that the first 16 values change smoothly, which is why they successfully pass through the selection module. It is visible that the first 16 values change smoothly, which is why they successfully pass the selection module. The second 16 values are chaotic. They should not be allowed to pass during verification.

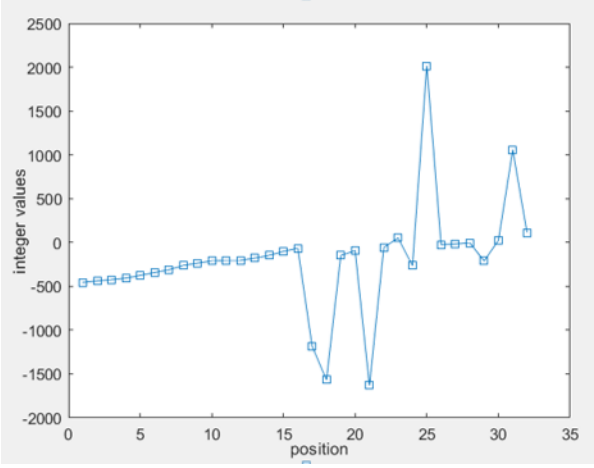

Fig. 5 Potential selected and rejected sequences during verification

IV. RESULTS

The experiments were conducted with the same input sequence and the same key. Since the number of possible decryptions in the MATLAB 2019a – Simulink environment is very limited, the conducted experiment aims to evaluate the effectiveness of the selection module and the need for a verification module. The results show that a certain number of sequences pass through the selection module, which, in the format of integer values, are sequential, not chaotic. With a limited number of experiments, the verification module works effectively.

V. CONCLUSION

This article presents certain aspects of designing selection and verification modules for applying a reduced brute force attack on voice streams. This need arises from the fact that the classical application of the attack does not carry information about which is the correct decryption from the set of such. The interest in "Brute force" attack stems from the fact that it does not depend on the mathematical construction of the algorithm, but only on the length of the key and the availability of computational resources. Technological progress in quantum technologies leads to the fact that compared to symmetric algorithms including AES, Grover's algorithm [13],[14] increases the probability of effective application of brute force attack. It provides quadratic acceleration [15], therefore, with 256 - AES it would require $O(2^{128})$ attempts to implement the attack, comparable to $O(2^{256})$ attempts applied on a classical computer system. Despite technological progress, at the moment 256-bit AES remains computationally unbreakable. It is a matter of the near future for this to happen.

REFERENCES

[1] J. Daemen, V. Rijmen, "The Design of Rijndael: AES - The Advanced Encryption Standard", Springer, 2002, Available: https://link.springer.com/book/10.1007/978-3-662-04722-4 [Accessed April 27, 2026]

[2] B. Schneier, "Schneier on Security". [Online]. Available: https://www.schneier.com/crypto-gram/archives/2000/1015.html [Accessed June 8, 2026]

[3] A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, A. Shamir, "Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds", In EUROCRYPT'10, vol. 6110 of Lecture Notes in Computer Science, pp. 299–319. Springer, 2010.

[4] A. Biryukov, D. Khovratovich, "Related-Key Cryptanalysis of the Full AES-192 and AES-256", In ASIACRYPT'09, vol. 5912 of Lecture Notes in Computer Science, pp. 1–18. Springer, 2009.

[5] A. Biryukov, D. Khovratovich, I. Nikoli´c, "Distinguisher and related-key attack on the full AES-256", In CRYPTO'09, vol. 5677 of Lecture Notes in Computer Science, pp. 231–249. Springer, 2009

[6] A. Biryukov, I. Nikoli´c, "Automatic Search for Related-Key Differential Characteristics in Byte - Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others", In EUROCRYPT'10, vol. 6110 of Lecture Notes in Computer Science, pp. 322–344. Springer, 2010.

[7] A. Bogdanov, D. Khovratovich, Ch. Rechberger, "Biclique Cryptanalysis of the Full AES", LectureNotes in Computer Science, pp. 344–371. Springer, 2011, Available: https://www.researchgate.net/publication/221326929_Biclique_Cryptanalysis_of_the_Full_AES [Accessed April 27, 2026]

[8] Ch. Boura, P. Derbez, M. Funk, "Related – Key Differential Analysis of the AES", IACR Transactions on Symmetric Cryptology, vol. 2023, no. 4, pp. 215 – 243, Available: https://tosc.iacr.org/index.php/ToSC/article/view/11286/10818 [Accessed June 8, 2026], https://doi.org/10.46586/tosc.v2023.i4.215-243

[9] B. Tao, H. Wu, "Improving the Biclique Cryptanalysis of AES", Information Security and Privacy. Lecture Notes in Computer Science, vol. 9144, pp. 39- 56, Springer, Available: https://link.springer.com/chapter/10.1007/978-3-319-19962-7_3 [Accessed June 8, 2026]

[10] A. Bogdanov, D. Chang, M. Ghosh., S.K. Sanadhya, "Bicliques with minimal data and time complexity for AES" In: Lee, J., Kim, J. (eds.) Information Security and Cryptology-ICISC 2014. Lecture Notes in Computer Science, pp. 160–174. Springer, 2014

[11] Ch. Boura, N. David, P. Derbez, G. Leander, M. Naya-Plasencia "Differential meet-in-the-middle cryptanalysis" , In CRYPTO 2023, Part III, vol. 14083 of Lecture Notes in Computer Science, pp. 240–272. Springer, 2023.

[12] Dh. Saha, D. Mukhopadhyay, D. RoyChowdhury. "A Diagonal Fault Attack on the Advanced Encryption Standard", Available: https://eprint.iacr.org/2009/581.pdf, [Accessed June 8, 2026]

[13] L. Grover, "From Schrödinger's equation to the quantum search algorithm", American Journal of Physics pp. 769-777, 2001. Available: https://arxiv.org/pdf/quant-ph/0109116, [Accessed June 8, 2026]

[14] C. H. Bennett, E. Bernstein, G. Brassard, U. Varirani, "The strength and weaknesses of quantum computation". SIAM Journal on Computing, 26(5), pp. 1510-1523, 1996. Available: https://arxiv.org/pdf/quant-ph/9701001, [Accessed June 8, 2026]

[15] S. Rao, D. Mahto, D. Yadav, Danish, D. Khan., "The AES-256 Cryptosystem Resist Quantum Attacks", International Journal of Advanced Research in Computer Science, 2017, 8(3) p. 404-408 Available:https://www.researchgate.net/publication/316284124_The_AES-256_Cryptosystem_Resists_Quantum_Attacks,[Accessed June 8, 2026]