INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Internal Re-keying Based Modified AES

Amit Rathod[1], **Kajal Patel**[2]*

1 Ph.D. Student, Gujarat Technological University, Ahmedabad, Gujarat, India
2 Associate Professor, Vishwakarma Government Engineering College, Ahmedabad, Gujarat, India

OPEN ACCESS

Received: 03-08-2024

Accepted: 27-12-2024

Published: 16-01-2025

Citation: Rathod A, Patel K (2025) Internal Re-keying Based Modified AES. Indian Journal of Science and Technology 18(1): 85-94. https://doi.org/10.17485/IJST/v18i1.2012

* **Corresponding author**.

kajalpatel@vgecg.ac.in

Funding: None

Competing Interests: None

Copyright: © 2025 Rathod & Patel. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: Masking and re-keying are two major countermeasures against the Power Side Channel Analysis attacks. Re-keying has either secret sharing overhead or needs to be used in synchronization. The Advanced Encryption System (AES) has been modified in the proposed scheme and uses re-keying without the need for secret random sharing or need for synchronized communication. The research proposed a modified AES scheme and validated its effectiveness with the AES. **Methods:** This study proposes modifying AES and then implementing it as software encryption using Python. As the AES has been modified, the proposed scheme should also match its strength to be further used against the Power Analysis Attack. Avalanche parameter is used to check the proposed solution strength. A data set of 10000, 50000, and 100000 records were generated to test the avalanche effect. The avalanche of existing AES and modified AES are then compared. **Findings:** The results indicate that the avalanche effect for both AES and the modified AES remains equivalent for the supplied dataset. To analyze further, the avalanche distribution is analyzed and randomness is checked using the Shannon entropy and found that the modified AES provides 0.5% more randomness against the AES. Hence, the modified AES fulfills the benchmark criteria to further check its strength against the Power Side Channel Analysis attacks. **Novelty:** The algorithm uses a part of the plain text to generate the round key making the resultant key unique and as it can be re-generated from the ciphertext and original key schedule on the decryption side, the random sharing is not needed.

Keywords: Masking; Random Sharing Overhead; Differential Power Analysis (DPA); AES; Avalanche Effect; Re-keying

1 Introduction

The Power Side Channel Analysis(SCA) is an ongoing open problem. In Power SCA attacks, the adversaries with physical access to the hardware processing the encryption algorithm can use the probe to intercept the minute electrical signal that reflects the data being processed. Such attacks bypass the mathematical security of the cryptographic algorithms and hence pose a real threat. Simple Power Analysis (SPA) and Differential Power Analysis (DPA) are the most common attacks. For the symmetric encryption, the

DPA is the focus of the researchers. Masking and Re-keying are two main countermeasures against the DPA for software implementation. Against these countermeasures, using statistical methods, the attackers have successfully recovered the keys. Hence, the higher order masking has been applied but with a large processing overhead. The main drawback of the masking technique is that it uses the same key to encrypt a few thousand plain texts. Hence, even with masking, the higher order DPA attacks can recover the major part of the key eventually leading to full key recovery. Another mechanism is called rekeying where ideally we should use the key only once for an encryption which is a one-time pad. However, practically to achieve that, it has an overhead of sharing the key with the receiver which is an even bigger overhead. Internal rekeying is a mechanism in which an intermediate key is derived from the master key to protect the master key. The internal re-keying is based on the algorithmic changes during the key generation and hence, there is no need to share the key with the receiver as on the receiver side also, the same key can be generated. In the recent NIST workshop [1], the industry discussed the practical challenges and the need for new ciphers which is the motivation for the proposed research. The proposed work is a practical implementation of such a scheme for AES. The next part of this section discusses the recent literature in the field. The section 2 discusses the proposed method, section 3 is about validating the method and results discussion, and section 4 is the conclusion.

1.1 State of the Art

Till now, the researchers have broadly categorized the re-keying into internal and external re-keying. The mechanisms are classified also on the basis of application area e.g. cloud rekeying management, sensor network re-key management, encryption etc. The Table 1 contains all major re-keying schemes and methods.

Table 1. Re-keying schemes and related literature

Title	Description
EPREKM [2]	External re-keying scheme for key management with constant overhead and based on ElGamal scheme and uses the proxy re-encryption with public bulletin board for message management.
Encryption by block based on rekeying and inter-intra pixel permutation [3]	Internal re-keying scheme by using inter-intra blocks of the pixel permutation and as a result the adjacent pixel proximity property is dissipated in the encryption process.
Modified AES for unicode [4]	The modified AES algorithm to suit the UTF-16 cryptography with higher efficiency.
Counter Feedback Mode (CFB) mode of operation with external re-keying [5]	In normal CFB mode, a single initial key is used and proved vulnerable. The authors proposed multikey method to enhance security
Internal Re-keying CTR-ACPKM [6]	Internal re-keying method has entropy loss in successive repetitions is observed. Enhancing it is the need to use a sufficient timeline.
Multiple Physical Rounding for Efficient Fresh Re-Keying [7]	The authors have demonstrated that the multiple physical rounds of modular arithmetic for rekeying is more effective than the single round heuristic approach.
Rekeying in Real-Time Communication Systems [8]	The authors have proposed the algorithm in which the rekey data is piggybacked with the ciphertext from which the rekey can be generated.
Key mashing and key distribution combined [9]	For internal rekeying, an algorithm that combines the key mashing and key distribution both instead of using them as isolated techniques.
Rekeying-based Moving Target Defence Mechanism [10]	The re-keying scheme uses the key usage expiry mechanism based on the number of traces and it changes based on the profile information it gathers and uses the combination of masking and re-keying together. Also attack model is included to calculate dynamically to assess the number in key usage expiry
Tweakable functions in Re-keyings [11]	The authors have proposed a new scheme with random sharing and proved the earlier schemes with birthday and without birthday security and were generic and they did not prove them. so the authors proved them to be insufficient and proposed a new scheme with two tweakable functions . and where the r is applied to 2 different functions who do exor before and after by 1 function and tweaked by the other function.
Conditional Proxy Re-Encryption-Based Key Sharing Mechanism for Clustered Federated Learning [12]	The approach uses the tree based key mechanism for the group of users to perform re-encryption such that the privacy is preserved in a clustered federated learning environment.
Re-keying with Masking and Shuffling: Application to ISAP [13]	To thwart the single trace attack, the paper proposed the method for masking and shuffling instead of combining re-keying and masking and exhibits leakage resistance for some applications.

Continued on next page

Table 1 continued

Counter mode with related-key-based internal re-keying [14]	The scheme uses the CTR mode along with the related key based re-keying scheme and performs the attack to prove that the amount of effort required is far higher than required for the said case.

1.2 Literature and the Research Gap

The block cipher modes of operations [3] [5] [14] are the primary techniques of the internal rekeying mechanisms. Internal rekeying mechanisms reduce the number of randomness among the communicating parties. The limitations of the existing schemes are that as the new key is based on the earlier processed data, the encryption process must be synchronous in nature [5]. Hence, the defined problem here is, to develop an internal re-keying scheme that does not require sharing the randomness or key with the receiver and is not dependent on the earlier encryption. This paper focuses on overcoming the limitation of the internal re-keying of using synchronous encryption. Based on the attacks that are explored, it is inferred that there are requirements for combining multiple approaches to have a secured rekeying mechanism that uses internal as well as external rekeying [2,7–10,12,13]. For applications in Wireless Sensor Networks or IoT applications where power consumption is an important factor can focus more on internal rekeying and less on external rekeying. Multiple successive executions of internal rekeying schemes are causing the entropy loss [6] and hence it needs to have a continuous source of some reversible random number generated internally ideally for each encryption execution. In [4,11], the authors have discussed modifying the encryption and tweaking the key generation.

From the above literature discussion, there is a requirement of a rekeying scheme which is internal rekeying i.e. does not need to share any randomness or key with the recipient, and it uses the entropy generated internally but different for different execution, and the key generation for successive encryption is not sequential.

With this motivation, this paper contains the proposed approach which is an internal rekeying approach which is a plaintext dependent approach and does not repeat the key, hence a kind of one-time pad and which does not have a key transmission overhead as the rekeyed key can be recovered from the ciphertext that is implicitly embedded during the process. That brings the best of both worlds of internal and external rekeying. A possible threat in this case is related to the fact that the same plaintext renders the same output. To thwart that, a combination of this technique with another can be done which is beyond the scope of this work. This work focuses on the new technique which does not require random sharing either at the beginning, nor sending key data as piggyback.

To demonstrate it, we have used the AES algorithm. To validate the proposed work, it is shown that the strength of AES is not compromised. As discussed in [15], the avalanche effect is the baseline measurement to check the strength of the scheme, our proposed scheme has been validated with the AES with avalanche effect. Similar work of modifying AES has been discussed in [6] where the avalanche effect is also discussed. This algorithm can be checked against the side channel attacks also but it is beyond the scope of this paper.

2 Methodology

The proposed scheme has been implemented using the AES algorithm. The 10th round of the AES has been tweaked and the last round key has been re-keyed using earlier round ciphertext. As the proposed scheme has tweaked the AES algorithm, the scheme strength needs to be re-validated. To validate the proposed scheme, the Avalanche effect parameters have been used. To understand the enhanced effect of the proposed scheme, the avalanche distribution has been studied.

While discussing encryption, in all the literature the key and the lock analogy is used. However, in reality, the lock and key pairs are unique. The same key is not used to lock multiple locks and vice-versa. The property lies in the unique way of impression applied in the internal structure of the lock and the way the key is designed. In one way, the internal structure of the lock is imprinted on the key. If we use this property in our encryption, the one-time pad encryption can be achieved. This analogy has been used in the proposed scheme. To make the internal re-key for the particular plain text, the impression of the shuffled plaintext is used. By using this approach, the resultant key is not dependent on the previous cipher and is also a unique key. The linear and differential cryptanalysis focuses on the properties of the key and tries to uncover them. In the proposed re-key scheme, as the rekeying process includes the impression of the plaintext that can also be called the randomness that is used in external re-keying, the original key property is masked. In the proposed work, we have used the re-keying in the 11th round of key of AES, the 10th round ciphertext has been used as the impression, it is a reversible operation. Hence, on the decryption side, it can be recovered. As this randomness can be recovered from the ciphertext, it does not have the overhead of sharing the randomness. The same has been proven in the next section. Therefore, the proposed scheme gives the properties of both internal and external re-keying without the overhead of random sharing or performing the operation in a synchronous

way.

The proposed algorithm has chosen the last round after the 10th round ciphertext because the masking of the key by the ciphertext after the 10th round is done on the encryption side and unmasking is done on the decryption side. As we have seen in the literature, this scheme combined with other schemes or in the modes of operation would render the desired results. Additionally, for the other 9 rounds, masking techniques can also be used to thwart the side channel analysis attacks.

2.1 Modified AES

Proposed Modified AES is based on the following basic concepts:

- AES itself is an RNG algorithm where the aim is to make the plain and cipher text as unrelated as possible and it uses a message (plaintext) for key-schedule to generate the last round key. An intermediate ciphertext state is used in the key schedule to generate later round keys that are specific to the message (near one-time pad security).
- Use of masking and unmasking techniques for generation of message specific keys.
- There is a need for an invertible function. During the decryption phase, the intermediate cipher text used for key generation needs to be retrieved from the cipher text.
- The re-keying process in the proposed system is similar to masking. However, the masking in this case is applied on the key and uses earlier round ciphertext as a mask.
- The masking is applied during encryption on the round key. The mask removal is done during the decryption process. The following explanation and formula discuss the process.

2.1.1 Revisiting the basic concepts of AES key generation

Notations:

- Cipher state at the point of interest: (As shown in Figure 1) C_{10}
- Decipher state at the point of interest: (As shown in Figure 1) D_1
- D_1=C_{10}
- C10 divided into 8-bit states: $C_{10}(0,0)$ to $C_{10}(3,3)$
- + denotes XOR operation
- WB denotes word byte i.e. W_1B_0, W_1B_1, W_1B_2 and W_1B_3 meaning Word 1, Byte 0, 1, 2 and 3.
- . denotes concatenation i.e. $W_1B_0.W_1B_1$
- MW denotes masked keyword
- Operation S denotes SubWord function

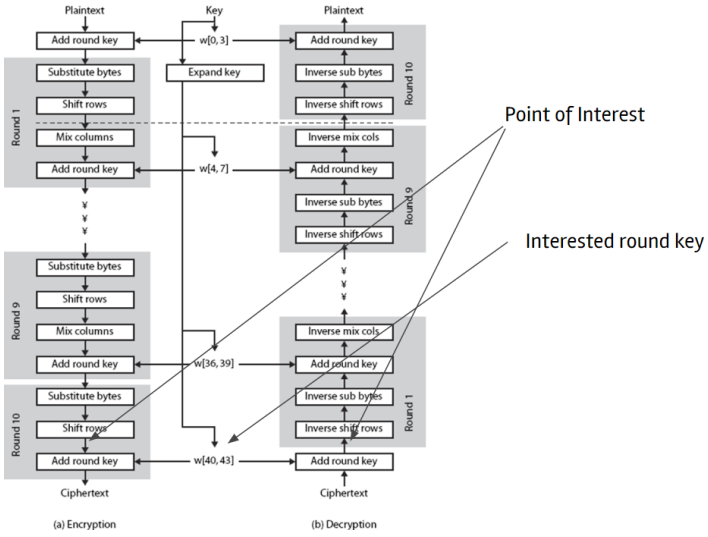

Fig 1. Proposed Algorithm change (1) Points of Interest to modify and (2) Proposed change for 11th round key modification

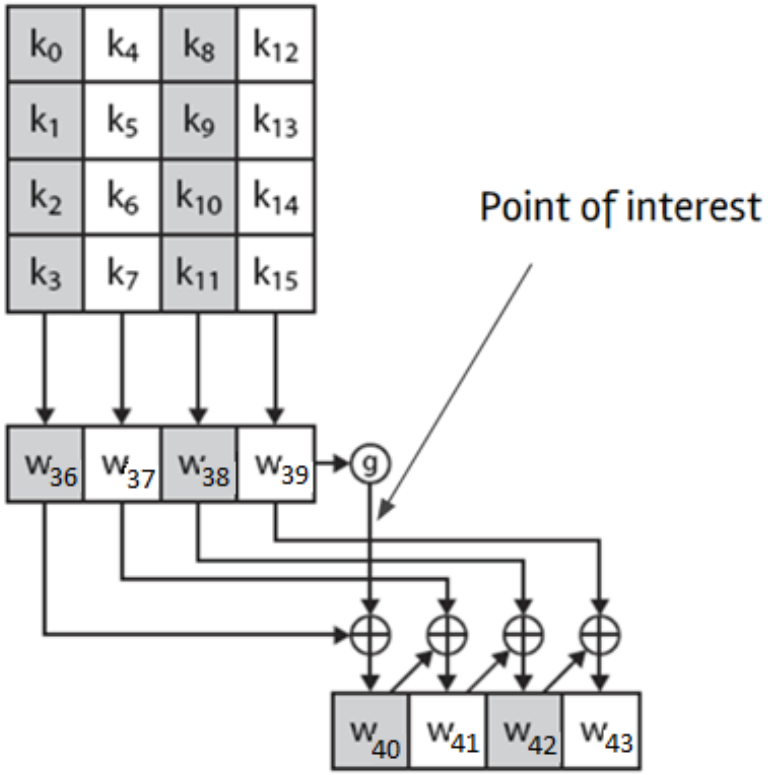

Fig 2. The point of modification in the 11th round key generation

Some Formula from key generation: (from Figure 2)
$W_{40}=W_{36}+SubWord(RotWord(W_{39})+36q$
$W_{41}=W_{37}+W_{40}$
$W_{42}=W_{38}+W_{41}$
$W_{43}=W_{39}+W_{42}$
As W_{41} to W_{43} depends on W_{40} directly or indirectly, hence the point of interest is W_{40} . as shown in Figure 1 and Figure 2.
Round key addition:
S(0,0), S(1,0), S(2,0), S(3,0) are points of interest (Figure 2) as they will be XORed with Wi as shown in Figure 3.

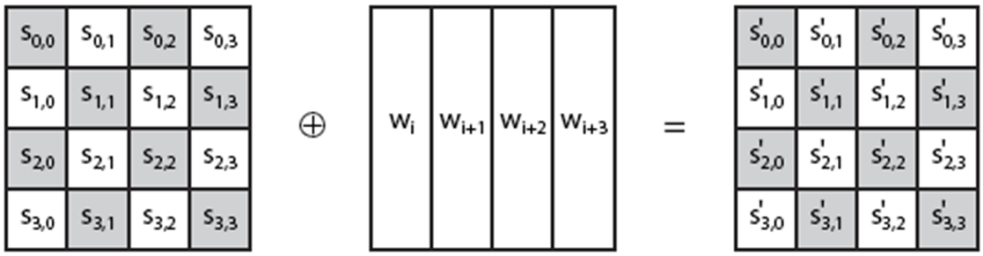

Fig 3. Round key addition process

As we're recomputing 10th round key, and W40 is the main word from which we get others, the mask would be from the first column i.e. $C_{10}(0,0)$, $C_{10}(1,0)$, $C_{10}(2,0)$ and $C_{10}(3,0)$.

W40 without mask

$W_{40}=W_{36}+SubWord(RotWord(W_{39})+36$

Solving the above, we get the following.
W_{40}=W_{36}+S($W_{39}B_1$).S($W_{39}B_2$).S($W_{39}B_3$).S($W_{39}B_0$)+36
As the above formula applicable to all 4 bytes, we try to work at byte level as follows:

$$W_{40}B_0 = W_{36}B_0 + S(W_{39}B_1) + 36 \tag{1}$$

$$W_{40}B_1 = W_{36}B_1 + S(W_{39}B_2) \tag{2}$$

$$W_{40}B_2 = W_{36}B_2 + S(W_{39}B_3) \tag{3}$$

$$W_{40}B_3 = W_{36}B_3 + S(W_{39}B_0) \tag{4}$$

2.1.2 Proposed Scheme Masking and Encryption steps

Masked W40

$$MW_{40}B_0 = W_{36}B_0 + C_{10}(1,0) + S(W_{39}B_1) + 36 \tag{5}$$

$$MW_{40}B_1 = W_{36}B_1 + C_{10}(2,0) + S(W_{39}B_2) \tag{6}$$

$$MW_{40}B_2 = W_{36}B_2 + C_{10}(3,0) + S(W_{39}B_3) \tag{7}$$

$$MW_{40}B_3 = W_{36}B_3 + S(W_{39}B_0) \tag{8}$$

Masked key Encryption - 11th key generation steps:

- C11(0,0) =C10(0,0) +$MW_{40}B_0$

2. Expanding the above function using Equation (5), we get

$$C_{11}(0,0) = C_{10}(0,0) + W_{36}B_0 + C_{10}(1,0) + S(W_{39}B_1 + 36) \tag{9}$$

Similarly, we get final cipher texts as

$$C_{11}(1,0) = C_{10}(1,0) + W_{36}B_1 + C_{10}(2,0) + S(W_{39}B_2) \tag{10}$$

$$C_{11}(2,0) = C_{10}(2,0) + W_{36}B_2 + C_{10}(3,0) + S(W_{39}B_3) \tag{11}$$

$$C_{11}(3,0) = C_{10}(3,0) + W_{36}B_3 + S(W_{39}B_0) \tag{12}$$

The C_{11} is sent over the network and on the destination, it needs to be encrypted. A 24-bit mask is applied on the 32-bit word. This is also called rekeying for the 10th round key. As W_{40} is being used for the construction of W_{41-43}, the same masking will be propagated to other key parts. C_{10}(1,0), C_{10}(2,0) and C_{10}(3,0) are the key masks. This would need to be calculated before decryption.

2.1.3 Masked Key Recovery on Receiver Side and Masked Key Decryption Steps

Masked key Recovery: C_{11} and 11th round key W_{40}-W_{43} are available at the decryption side. To decrypt the 11th round, the MW_{40}-MW_{43} are required. For that, recovering $C_{10}(1,0)$, $C_{10}(2,0)$ and $C_{10}(3,0)$ from C_{11} and W_{40-43} are important. Now XORing C_{11} (Equations (9), (10), (11) and (12)) with W_{40-43}, we get as mentioned below:

$C_{11}(0,0)+W_{40}B_0=C_{10}(0,0)+W_{36}B_0+C_{10}(1,0)+S(W_{39}B_1)+36+W_{40}B_0$

Using Equation (1) to expand $W_{40}B_0$ on the left side, we get

$C_{11}(0,0)+W_{40}B_0=C_{10}(0,0)+W_{36}B_0+C_{10}(1,0)+S(W_{39}B_1)+36+W_{36}B_0+S(W_{39}B_1)+36$

$$C_{11}(0,0)+W_{40}B_0=C_{10}(0,0)+C_{10}(1,0) \tag{13}$$

Similarly,

$$C_{11}(1,0)+W_{40}B_1=C_{10}(1,0)+C_{10}(2,0) \tag{14}$$

$$C_{11}(2,0)+W_{40}B_2=C_{10}(2,0)+C_{10}(3,0) \tag{15}$$

$$C_{11}(3,0)+W_{40}B_3=C_{10}(3,0) \tag{16}$$

Equation (16) is $C_{10}(3,0)$. Now XORing Equation (16) with Equation (15), we get the $C_{10}(2,0)$ Then, XORing Equation (15) with Equation (14), we get the value of $C_{10}(1,0)$.

2.1.4 Masked Key Decryption

Now with all 24-bit masks $C_{10}(1,0)$, $C_{10}(2,0)$ and $C_{10}(3,0)$ recovered, we can use Equations (5), (6), (7) and (8) to calculate the MW_{40}-MW_{43} and then decrypt the C_{11} to get C_{10}. The remaining decryption for other rounds remains the same.

2.2 Experiment Setup and example use case

The above modified AES algorithm was implemented in the Google co-lab with Python language and Linux machine with 16GB RAM and the following is the sample use case. Following Python libraries have been used: numpy, pandas, math, matplotlib, and seaborn. This study focuses only on the validity of the algorithm and future work will focus on hardware implementation and other combining techniques to decide the effectiveness against the side channel attacks.

Table 2. Encryption side

Plain text:	3243F6A8885A308D313198A2E0370734
Key:	2B7E151628AED2A6ABF7158809CF4F3C
11th Round Key	3925841D02DC09FBDC118597196A0B32
C10	EB40F21E592E38848BA113E71BC342D2
11th Round Modified Key	E1694CA8F8939089D042B9C8871EB9A6
Modified Ciphertext:	0858311D33A1BCFBED6C30972817BE32

Table 3. Decryption side

Modified Ciphertext:	0858311D33A1BCFBED6C30972817BE32
Decryption 1st round key (Reverse of Encryption)	3925841D02DC09FBDC118597196A0B32
Derived Modified Decryption 1st round key (Reverse of Encryption)	E1694CA8F8939089D042B9C8871EB9A6
Derived D1	EB40F21E592E38848BA113E71BC342D2

2.2.1 Parameters

The aim of the research is to propose a new scheme that provides re-keying without random sharing. To validate the proposed algorithm and to further analyze it for the side channel analysis, its strength needs to be checked against the existing AES algorithm. Hence, Avalanche is calculated and the result is compared.

Avalanche Effect= (Number of Changed bit in ciphertext) / (Number of bits in ciphertext)

3 Results and Discussion

3.1 Avalanche Effect

The experimentation was conducted in the Google co-lab with samples of 10000, 50000, and 100000 plaintexts with a change of 1 bit to measure the avalanche among them. A random synthetic data has been generated to evaluate the results. Multiple data points yield a similar pattern. The first data point is taken as the base point and others are calculated as avalanche and the average avalanche has been calculated and proved to be similarly effective as AES as mentioned in Table 4.

Table 4. Average Avalanche Comparison

Number of samples	Average Avalanche -AES (bits)	Average Avalanche - Proposed Modified AES (bits)
10000	63.06	62.98
50000	63.12	63.14
100000	63.04	63.04

However, based on some of the samples, it was observed that the avalanche was more irregular for the proposed method than the AES. Hence, it is important to use the Probability Density Function (PDF) of the AES to further analyze the effects.

3.2 PDF of Avalanche

The PDF function for the AES when plotted exhibited a skewed pattern towards several values and their frequencies also seem to be higher as shown in Figure 4 and for the proposed algorithm, the PDF function looked quite sparse than the AES with more data points and fewer frequencies as shown. To reduce the key space, attackers use skewed values to reduce the key space. As the proposed distribution is less skewed, the key space for the avalanche spreads over multiple similar values. To examine this pattern further, it was decided to check the randomness of the distribution and the Shannon entropy value is calculated for both the algorithms. Shannon Entropy is defined as:

$$H = -\sum(i = 1\ to\ n)\ xi \cdot \log xi$$

The higher the entropy, the higher the randomness in the distribution. The Shannon Entropy Avalanche Distribution (S.E.A.D.) was calculated and the results indicated that the proposed algorithm distribution has higher randomness. Higher randomness for encryption methods indicates a higher strength of the algorithm. The results for the Shannon entropy are given in Table 5.

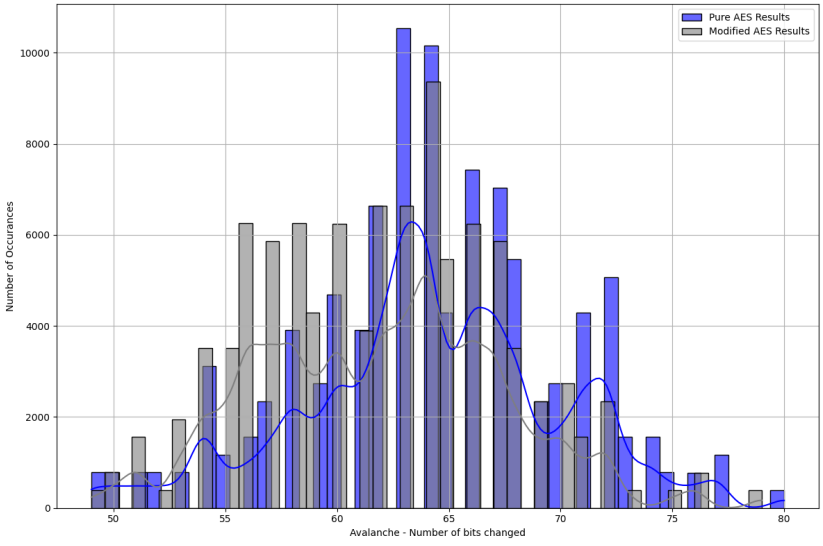

Fig 4. Probability Distribution of Avalanche of AES

Table 5. Shannon Entropy of Avalanche Distribution (SEAD)

Number of samples	S.E.A.D. – AES	S.E.A.D - Proposed Modified AES
10000	4.239899537652175	4.308705417207129
50000	4.427264567129531	4.4669034577913065
100000	4.427085395940455	4.466050935867275

4 Conclusion

Re-keying is used to thwart statistical cryptanalysis. There are mainly two categories of re-keying: external and internal. The external re-keying uses the random value 'r' to be shared between the sender and receiver and carries its overhead. On the other hand, the internal re-keying does not need to share the randomness and uses the synchronized algorithms to generate the subsequent values. The external re-keying carries the overhead while internal re-keying needs to be sufficiently resilient against attacks. Previous studies have suggested the combined approaches to increase the effectiveness and efficiency of the approaches. The proposed work in this paper is a novel approach in which the randomness of the intermediate ciphertext is used as randomness to generate a partial key that will be used in the last round of AES and is sent implicitly along with the ciphertext. The randomness based partial key can be re-generated from the ciphertext and the original key and then can be used to decrypt the message on the receiver side. Hence, it is a re-keying technique without random sharing. The scope of this paper is only to present the technique and measure its strength with reference to the benchmark. The results of the avalanche indicate that it is equivalently strong and provides slightly more (0.5%) avalanche distribution than the AES. As the avalanche randomness increases, this technique combined with masking, rekeying, or any other technique could render better protection against the statistical analysis and power analysis attacks. In future work, the technique can be combined with masking and calculate its effectiveness against the Simple Power Analysis (SPA) and Differential Power Analysis (DPA) attacks. The proposed internal rekeying method leverages the benefits of external re-keying without random sharing overhead.

Acknowledgements

The authors would like to acknowledge the Vishwakarma Government Engineering College, Chandkheda for providing the computing and library resource for conducting the study.

References

1) Kampanakis P, Campagna M, Crocket E, Petcher A, Gueron S. Practical Challenges with AES-GCM and the need for a new cipher. 2024;p. 1–12. Available from: https://csrc.nist.gov/csrc/media/Events/2023/third-workshop-on-block-cipher-modes-of-operation/documents/accepted-papers/Practical%20Challenges%20with%20AES-GCM.pdf.
2) Sharma P, R PB. EPREKM: ElGamal proxy re-encryption-based key management scheme with constant rekeying cost and linear public bulletin size. *Concurr Comput*. 2024;36(13). Available from: https://doi.org/10.1002/cpe.8078.
3) Rimani R, Pacha A, Said N. Encryption by block based on rekeying and inter-intra pixel permutation. *International Journal of Information and Computer Security*. 2024;24:117–136. Available from: https://doi.org/10.1504/IJICS.2024.140209.
4) Rose SJ, Ozkaya U, Yasmin S, Jabin S, Hasan R, Kabir E. An Efficient Unicode encoded in UTF-16 text cryptography based on the AES algorithm. *Cryptology ePrint Archive*. 2023;p. 1–10. Available from: https://eprint.iacr.org/2023/1104.pdf.
5) Tsypyschev V, Morgasov I. Provable security of CFB mode of operation with external re-keying. *Cryptology ePrint Archive*. 2022;p. 1–15. Available from: https://eprint.iacr.org/2022/291.pdf.
6) Dunkelman O, Ghosh S, Lambooij E. Attacking the IETF/ISO Standard for Internal Re-keying CTR-ACPKM. *Cryptology ePrint Archive*. 2023;p. 1–30. Available from: https://eprint.iacr.org/2023/230.pdf.
7) Duval S, Méaux P, Momin C, Standaert FX. Exploring Crypto-Physical Dark Matter and Learning with Physical Rounding: Towards Secure and Efficient Fresh Re-Keying. *ACR Transactions on Cryptographic Hardware and Embedded Systems*. 2020;2021(1):373–401. Available from: https://doi.org/10.46586/tches.v2021.i1.373-401.
8) Bühler H, Walz A, Sikora A. Mechanism for Seamless Cryptographic Rekeying in Real-Time Communication Systems. In: and others, editor. 17th IEEE International Conference on Factory Communication Systems (WFCS). IEEE. 2021;p. 53–58. Available from: https://ieeexplore.ieee.org/document/9483613.
9) Akhmetzyanova L, Alekseev E, Smyshlyaev S, Oshkin I. On Internal Re-keying. In: and others, editor. International Conference on Research in Security Standardisation;vol. 12529. Springer, Cham. 2020;p. 23–45. Available from: https://doi.org/10.1007/978-3-030-64357-7_2.
10) Vuppala S, Mady AED, Kuenzi A. A Rekeying-based Moving Target Defence Mechanism for Side-Channel Attacks. In: and others, editor. Global IoT Summit (GIoTS), 17-21 June 2019, Aarhus, Denmark. IEEE. 2019;p. 1–5. Available from: https://ieeexplore.ieee.org/document/8766426.
11) Komano Y, Hirose S. Re-Keying Scheme Revisited: Security Model and Instantiations. *Applied Sciences*. 2019;9(5). Available from: https://www.mdpi.com/2076-3417/9/5/1002.
12) Zhang Y, Zhang Z, Ji S, Wang S, Huang S. Conditional Proxy Re-Encryption-Based Key Sharing Mechanism for Clustered Federated Learning. *Electronics*. 2024;13(5). Available from: https://doi.org/10.3390/electronics13050848.

13) Udvarhelyi B, Bronchain O, Standaert FX. Security Analysis of Deterministic Re-keying with Masking and Shuffling: Application to ISAP. In: and others, editor. International Workshop on Constructive Side-Channel Analysis and Secure Design;vol. 12910. Springer. 2021;p. 168–183. Available from: https://link.springer.com/chapter/10.1007/978-3-030-89915-8_8.
14) Alekseev EK, Goncharenko KS, Marshalko GB. Provably secure counter mode with related-key-based internal re-keying. *Journal of Computer Virology and Hacking Techniques*. 2020;16:285–294. Available from: https://link.springer.com/article/10.1007/s11416-020-00357-9.
15) Verma R, Sharma AK. Cryptography: Avalanche effect of AES and RSA. *International Journal of Scientific and Research Publications*. 2020;10(4):119–125. Available from: https://www.ijsrp.org/research-paper-0420/ijsrp-p10013.pdf.