e-ISSN:2814-0990 DOI: https://dx.doi.org/10.53944/ojit-2206

OIRT Journal of Information Technology

Available at: https://journals.otuirt.com/index.php/OJIT

MINI REVIEW

A Review of Faults Attack on Symmetric Key Cipher and Appropriate Counter Measures

Sadiq Abdulkarim[1*], Samaila Kasimu Ahmad[2], Francis Binord[3]

[1]*Department of Computer Science, Faculty of Computing, Nigerian Army University Biu, Borno, Nigeria*
[2]*Department of Information Systems, Faculty of Computing, Nigerian Army University Biu, Borno, Nigeria*
[3]*Department of Cyber Security, Faculty of Computing, Nigerian Army University Biu, Borno, Nigeria*

Corresponding Author: Sadiq Abdulkarim, Department of Computer Science, Faculty of Computing, Nigerian Army University Biu, Borno, Nigeria. Email: sadiq.abdulkarim@naub.edu.ng

ABSTRACT

The world is evolving rapidly, industrialization, centralization, civilization, and revolution on the fore-go, the need for the day-to-day activities ranging from the usual informal conversation between family members and relatives, confidential information/data from government repository, teacher to student dialoguing, down to business activities, client-server requests and response etcetera, involves the use of mediums/channels/conduit to aid its feasibility, this supposed non-polar activities always tend to draw/attracts the attention/interest of various other external observers from a different frame of reference, the interest propels the observer into wanting to gain access, upper-arm or control of this activities/resource which in most occasions tend not to be favorable to the two standing parties, hence the need for a secure end-to-end communication regarding also and amidst the presence of the third parties also known synonymously as adversaries**.** This paper looked into various fault attacks on symmetric-key ciphers and came up with solutions to counteract the attacks.

Keywords: CIA, Cipher, Counter Measures, Fault Attack, Symmetric

Article History	Received: September 08, 2021	Revised: October 28, 2021	Accepted: January 02, 2022

INTRODUCTION

The revolution of cryptography and what it comes along with itself are the various endpoints that aid the three principles of security, namely, “confidentiality, integrity, and availability (CIA)” by preventing the four primary classes of attack, namely reconnaissance, access, denial of service and eventually malicious codes/scripts. Billions of people around the globe use cryptography daily to protect data and information, although most do not know that they are using it (Smirnoff & Turner, 2019). In other to be secured, info ought to be kept away from unauthorized or unprecedented access (that is, confidentiality), protected from unauthorized alteration and re-modification (which is integrity), and available only to an authorized mechanism or entity whenever it is needed for use (availability). The study and practice of encryption and decryption are called the science of cryptography.

Cryptologists are called cryptologists, who study different ways to protect and ensure the information's confidentiality, integrity, and authenticity. Cryptologists also engage in cryptanalysis to find ways to break encryption methods (Onwutalobi et al., 2011). This revolution drifts alongside the world's redshifts from the use of various security techniques such as the use of guards, doors, locks or mode of conveying of data and information by the use of human messengers, smokes, etc. this technique over the years was proved to be an insecure and inefficient way considering the rate of data consumption and availability index, cryptography to the rescue, with cryptography, information, and communicating channels can be protected from an external onlooker or adversary.

This pops up again the question of what cryptography is about. So far, many definitions have been proposed by experts. Cryptography can be defined as techniques that cipher data, depending on specific algorithms that make the data unreadable to the human eye unless decrypted by algorithms predefined by the sender (Qadir et al., 2019). At the course of this paper, we define cryptography as "*the techniques implored In making an intelligible piece of information called a plaintext looks un-intelligible or gibberish by the use of a key in a process known as encryption by converting it into a ciphertext such that an external onlooker upon intercepting it won't be able to make any sense out of it except for the person whom the message was meant for and in possession of the key which can be used for decrypting or reconverting from its gibberish form back to its intelligible form.*"

Security is key in ensuring that two endpoints/parties are confident enough to communicate without the fear of having their messages or information intercepted or sniffed by an attacker or receiving remarkably alarming threats from an unknown source; having that in mind, everyone's such a conscience about security and how secure and safe our information and resources are. This pops up the question, "*Making and ensuring that our information/data are secure in the information age*," to ensure that our data and personal resources are not compromised. As well, we provide that the data been sent from node A (ABU) intended for and Restricted to node B (BUBA) do not get intercepted by an adversary node C (Charles). The field of computing that deals with ensuring how data, either in digital, analog, or both formats, is made secure to prevent its compromises is called CRYPTOGRAPHY.

RELATED WORK

Fault attacks are among the well-researched issues in the area of cryptography. The attacks create an influential tool to recover the secret key used in the encryption process. The fault attacks are situations created to force a device/system to work under non-ideal environmental conditions (such as high temperature) or external disturbances (like a glitch in the power supply) while performing a cryptographic operation (Baksi et al., 2020).

With the emergence of the IoT and artificial intelligence in the modern world, small-scale computing systems are becoming more universal. However, these computing systems usually perform cryptographic operations, which are vulnerable to system-specific attacks (Baksi et al., 2020). This is why device vendors invest a significant design effort when executing computationally intensive cryptographic algorithms onto constrained embedded computing systems to match the computational demands of the algorithms with the stringent area, power, and energy budgets of the platforms (Karaklajić et al., 2013).

A study come with two major categories of countermeasures (Dobraunig et al., 2018). The number one category covers sensor-based countermeasures that focus on spotting the physical process of the fault induction. Given likely examples, protected implementations may include light, voltage, and temperature sensors to ascertain fault inductions and number two as fault induction on a cryptographic algorithm (Dobraunig et al., 2018).

(Malkin et al., 2006) in their paper describing the study, they took in other to show some current countermeasures implemented in the fault attacks are inadequate, citing duplication and repetition as a security weakness. Finally, Malkin et al. suggest some design improvements for countermeasures, such as error detection techniques using space redundancies. (Malkin et al., 2006).

TYPES OF CRYPTOGRAPHY

Cryptography has been invariably subdivided into two endpoints. The logic behind this sub-sectional division is centred upon the technique used in the encryption and decryption process, their significance and how they affect our day-to-day activities, their respective impacts on globalization and commercial workflow. Picking and implementing one of the supposed types of cryptography lies in the business needs, organizational resources based on its technical workforce, and deliverables with its critical security index and data/information sensibility. The two types of cryptography are - (1) symmetric; (2) asymmetric.

Symmetric Cryptography:

Symmetric cryptography, otherwise known as private key cryptography, is a type of encryption in which a single key is used for both the encryption and the decryption process. The logic behind the symmetrically-centred kind of cryptography is simple and self-coherent (Dobraunig et al., 2018), the sender BUBA willing to send a message M to the receiver ABU self-generates an encryption key K (from some cryptographic algorithm or function) of which he BUBA uses to encrypt or encode the message. BUBA generates this key, uses it to encrypt the message, and sends them (both the key and message) to ABU. ABU uses the obtained key to decrypt or decode the ciphertext (encrypted message) into its original format when receiving the message and key. Within the heart of this technique reside a few glitches and faults, which we will discuss in this paper.

Asymmetric Cryptography:

Asymmetric cryptography, otherwise known as public-key cryptography, is similar to its symmetric counterpart except that in this case, we are using two identical but different keys for its encryption and decryption processes. This scheme calls for each party to generate its bi-keys. The working mechanism for this type of encryption follows that for a sender ABU intending to send a secret message to a receiving end, say BUBA requires both parties to generate two or a pair of keys (Daniel, 2021). From the onset, the deliverables of the mathematical algorithm or function are to generate two identical keys; the algorithm on itself has no idea which of the keys will be used for the encryption or instead which of the key is inscribed with the status of PUBLIC or PRIVATE. The logic follows that if one of the keys (probably chosen from some sets of arbitrary rules) is used for the encryption, its counterpart will be decrypted. For ABU and BUBA having generated their own sets of keys, assuming ABU the sender intends encrypting a message M and sending it over to the receiver BUBA, all ABU ought to do is to use the public key of the receiver BUBA, which of course will be stored in a public repository/registry known and available to everyone else to encrypt the message, BUBA upon receiving the message uses his private key which is residing in his private repository to decrypt the received ciphertext. The principle of the supposed public key cipher follows that it should be practically infeasible for an attacker, say OSCAR, to be able to decipher the ciphertext with a limited obtained information such as the encrypting key, knowledge of the encryption and decryption algorithm, and also the communicating channel (Daniel, 2021).

Principles of Algorithmic Encryption Scheme:

It follows that anyone could quickly develop and implement a cryptographic algorithm. But, on the other hand, the effects of misguided use of any random cryptographic algorithm could be catastrophic. Hence the need arises for a standard to be set. Two principles were propounded such that any developed cryptographic algorithm is required to yield/succumb to this principle.

The overall economic cost of breaking an encrypted message should relatively exceed the value of the information/data. Therefore, the overall time required for successful cryptanalysis (attacks cryptography) should span beyond the useful lifespan of the information/data. Therefore, the overall time needed for successful cryptanalysis (attacks cryptography) should span beyond the useful lifespan of the information/data (Afzal et al., 2020).

RESULTS AND DISCUSSIONS

Brute forced attack:

All cryptographic algorithms are vulnerable to brute force attacks (Hoffman, 2013). This pops up the question, what is a brute force attack? A brute force attack, also known as the all-out attack, is a guessing technique. The logic behind a brute force attack is to make all available combination of passwords candidates, hash it and compare against the hashed value or ciphertext of the intended message or plaintext. How does the fault in symmetric key cipher give way to the authoritative ship of the passage and resiliency of the supposed majestic brute force attack? THE LENGTH OF THE KEY of symmetric-key cipher is the out-weighted fault. Attackers knowing the key-independent nature of brute force attack and the corresponding fault in key-length could use a brute force attack and eventually break it anyway.

Countermeasures against Brute Force Attack:

Even with the famous and assuring nature of brute-force attacks, there are possible methods that will render the technique of brute force futile when harnessed and put in place. In as much as a brute force attack could break any algorithmic-enhanced ciphertext, it still totally depends on various contingencies such as the processor speed of the working machine, efficiency, and fault tolerance of the machine. With these outlined constraints, the mechanism of the supposed brute-force attack depends solely on how fast the machine could make those combinations. Having that in mind, it is logical to verify that the number of possible combinations tied to its success depends on the key length of the key used and its corresponding ciphertext with ultimate reference to the time it will take to attain that success. This brings us to ask, what if we use a reasonable key length? What if we increase the length of the key used in the encryption? It is reasonable to note that this countermeasure will not entirely stop the success of a brute force. Still, it will make it almost practically infeasible as the machine will have to make some rather infinitely-inclined combinations that will correspondingly take a sufficiently infinite amount of time and workforce. The table below will give us an analytic calculation showing the relationship between an encryption key's size and the corresponding time required for a successful attack.

Table 1: Brute Force key Length-Time Relationship

Key length (bits)	Alternative keys (n)	Time required at one decoding per micro secs	Time required at 10^6 decoding per microseconds
32	2^{32}=4.3x10^{9}	2^{31}ms=35.8minutes	2.15 milliseconds
128	2^{128}=3.4x10^{38}	2^{127}ms=5.4x10^{24} years	5.4x10^{18} years
168	2^{168}=3.7x10^{50}	2^{167}ms=5.9x10^{36}years	5.9x10^{30} years

Remark: From the above table's meta-data, we can see that increasing the size of a key to some reasonable amount will not entirely make brute force impossible, but it is sufficient to make it pragmatically infeasible.

Key-exchange problem:

Another fault associated with the symmetric key cipher is the key distribution problem (Stapko, 2008). To understand this problem, let take a careful look at the working mechanism of the symmetric cipher. ABU, on the verge of sending BUBA a secret message, will generate a key K and the intended message M. ABU will use the generated key to encrypt the message M into a ciphertext C, and now ABU will send the message over to BUBA. But for BUBA to understand the sent gibberish (ciphertext), he will need the generated key. Since it is ABU's intention for BUBA to read the encrypted message, he will have to send the KEY K also over to BUBA. This pops up the question, how will ABU send this key over the established channel without having a third party say Charlie intercepting it? Now, this is a fault in the system of an asymmetric cipher as it is vulnerable to MAN-IN-THE-MIDDLE-ATTACK. However, we have a way to rectify this inconsistent glitch. Let's see how.

Countermeasures in the Key-Exchange Problem:

In the symmetric cipher scheme, we care less about anyone intercepting the encrypted message as we, for some reason, trust our cryptographic algorithm. We believe that should the encrypted message be intercepted by CHARLIE; CHARLIE won't be able to decipher it without possessing the key. Two approaches could be used to rectify this fault.

The integration of symmetric cipher scheme and asymmetric by integrating alongside the symmetric mode of encryption the supposed popular Diffie-Hellman key exchange scheme. In this composite schema system, we use the symmetric key cipher scheme to encrypt the message and then use the asymmetric way of resolving key issues. One is likely to ask, why don't we use the asymmetric mode for all our encryption processes since it is secure and the key exchange problem has been eliminated from its end and save ourselves the stress of having to incorporate two different schemes? The answer to this non-trivial question is simple. The symmetric mode of encryption is considerably easy to implement and incredibly fast as it is recommended for bulky data due to its processing speed; asymmetric is never recommended when it comes to bulky data as it uses long ranges of key-length values and hence takes high computation power of several magnitudes, but its good part is eliminating the key-exchange problem. A problem that has fretted the whole computing community.

Another way of countering the key-exchange problem is using a responsible third party to generate and distribute the keys to participating parties. The essence of this technique is to leave the channel likely to entertain a man-in-the-middle-attack idle, thereby leaving the attacker hanging and in suspense. In addition, the third-party key generator is expected to use an entirely different channel for distributing the keys.

Timing Attack:

The timing attack exploits the co-linear relationship between different inputs and the corresponding time it would take for the encryption mode to spit out something. This technique uses timing response under discrete but different input. Technically we can show a mathematical but direct proportional relationship between the size of input or message and the size of the corresponding time length. Using this flaw, an attacker will be able to observe the system under different inputs and will be able to draw credible inferences of the possible pattern or nature of the key used in encryption. In his attempt, this attack was designed and proven to be efficient by the cryptographer Paul Kocher; he systematically obtained the private key used by RSA encryption without explicitly breaking the encryption.

Countermeasure to Timing Attacks

(I) Constant Response Time:

If we modify our system only to respond (T), T is the time for response, and subscript c is a constant entity. What if we disregard the magnitude of the input? The continuous-time response suggests that if the system only spits out an output of all firms in a specific predefined time, it will thwart the attacker's analysis based on timings.

Random Delay:

Does this countermeasure technique lie in the idea that a random delay is added to the system's output response in a sporadically inclined fashion? Instead, this technique attempts to bridge and falsify the mathematical relationship between the variable-length inputs and their corresponding response time.

Statistical Frequency Attacks:

Statistical frequency attacks try to analyse the patterns of the elements in ciphertext with the sole aim of picking out the elements that appear frequently. This technique checks the frequency of various characters in the ciphertext and tries to associate it with corresponding plaintext characters that stand as possible candidates as correspondence. In this technique, the attacker is usually interested in the patterns related to the syntactic nature of a given language and how often a given character has been used. For example, in the English language, we tend to notice some characters usually appear more frequently than others in words pattern or sentences, characters patterns such as a, e, you, I tend to occur most often than characters such as z, x, w, q, etc. equipped with this knowledge, an attacker by merely observing the ranges of ciphertext word pattern of the individual characters can skilfully associate this pattern with their corresponding counterparts in the plaintext format.

Countermeasures to Frequency Attacks:

Two possible techniques are usually used to thwart crypt-analytical techniques associated with frequency attacks. Claude Shannon first propounded this technique in 1945 in his research paper known as A mathematical theory of cryptography. They are the famous diffusion confusion technique.

(i) Diffusion:

In the supposed diffusion technique, we are concerned about making the relationship and similarities (transfer of info-byte) between the plaintext and the ciphertext as obscure as possible to thwart a frequency analysis attack. We are mostly less concerned about the mathematical nature of this technique as it uses a transposition algorithm as we can implement it by performing some letter permutation integrated with some mathematical function. In the diffusion method, the statistical structure of M, which leads to its redundancy, is "dissipated" into long-range statistics — i.e., into statistical structure involving long combinations of letters in the cryptogram. The effect here is that the enemy must intercept a tremendous amount of material to tie down this structure since the structure is evident only in blocks of minimal individual probability (Wu, 2019). This technique uses a mathematical function to clear-cut a massive distinction between the plaintext and the ciphertext. It functions because a single change in the plaintext pattern drastically affects the design of the ciphertext with a magnitude of almost half. This could also go the other way round. This will indeed frustrate an attacker attempting to use the frequency attack.

(ii) Confusion:

Like in the case of the supposed diffusion technique in thwarting frequency attacks, the confusion counter-analytical measures aim to make the relationship of any sort between an encryption key with its corresponding ciphertext. Technically, by merely observing the structures of ciphertext and their relationship with one another, an intelligent cryptanalytic attacker could, from standard inferences, successfully attempt and guess correctly the actual key used in the encryption. We could use a complicated mathematical substitution technique or algorithm to achieve this. However, we would be making a lot of progress if half of the entire structure of a ciphertext would be tweaked by merely altering the encryption key by a single bit. This is off-course similar but different from the famous avalanche effect.

System Attack:

In this type of attack, attackers are not interested in implementing the cryptographic algorithm or its mathematical nature; attackers tend to be more concerned with the systems this algorithm resides or is burnt. Technically the efficiency of a cryptographic algorithm depends solely on several constraints such as the magnitude of trans positional process, the complexity of substitution algorithm, the intensity of permutational structure, the computational power of the machine, randomization prowess of random number generator algorithm, etc. it is then the flaws found on the systems of which our experimental algorithm resides that has been exploited, For example, taking a critical look at several implementations of random numbers in an algorithmic term known as PSEUDO-RANDOM NUMBER GENERATOR. The essence of these systems is to output non-deterministic, un-predictable random numbers, but research has shown this system to be flawed as they depend on certain predictable constraints. An attacker could easily use these flaws, especially when a block cipher has been used, and the need for randomizing the block space arises.

Countermeasures of System Attacks:

The most apparent countermeasure to this type of attack concerning be a flaw mentioned above in the random number generator is by using what is known as THE TRUE RANDOM NUMBER GENERATOR, which can be attained by the use of non-deterministic sources in outputting randomness. The mode of operation is by the measuring of non-predictable disperse processes ranging from the motion of micro-processor, pulse emitters, ionizing radiations, rotational movement of disk-drives, amplitudes of the frequency from working capacitors, etc. with this set in place, the randomness produced won't and can't be predicted.

CONCLUSION

The field of cryptography has not only come to stay. Still, it is willing to stay, in as much as the world and its inhabitants depend solely on the digital way of life, hence the need for the resources and property of individuals to be protected. On the verge of making sure human reliance on the digital mechanism has not been exploited and compromised in any way, cryptographic researchers will keep on working tirelessly in building, improving, etc., security standards which in the dead-end converges to the field of cryptography. Regarding the above, it is logical to yield to the reasoning that the field of cryptography would be incomplete if researchers focus only on building cryptographic schemes and paying imperceptible or null attention to cryptanalysis (a technique involving an effort to break cryptographic schemes/algorithms). The efforts put in place by cryptanalysts make cryptography an outstanding and reliable discipline in the whole sample space of all forms of security disciplines. It is important to note that what the world refers to as a "strong" cryptographic algorithm could be regarded as a "weak" cryptographic algorithm tomorrow. The above discussion shows how brute force attacks can be counteracted with strong and long key encryption. Time timing attacks can also be avoided by the constant response time, random delay, and statistical frequency attacks.

Conflict of Interest: None

Funding/Support: None

REFERENCES

Afzal, S., Yousaf, M., Afzal, H., Alharbe, N., Mufti, M.R. (2020). *Cryptographic Strength Evaluation of Key Schedule Algorithms. Security and Communication Network.* ID 3189601. https://doi.org/10.1155/2020/3189601

Baksi, A., Bhasin, S., Breier, J., Jap, D., and Saha, D., (2020). *Fault attacks in symmetric key cryptosystems systemization of knowledge,* eprint IACR: Available from: https://eprint.iacr.org/2020/1267.pdf [Accessed on August 20, 2021].

Daniel, B., (2021). *Symmetric vs. Asymmetric Encryption: What's the Difference*, Trenton Systems, https://www.trentonsystems.com/blog/symmetric-vs-asymmetric-encryption [Accessed on August 20, 2021].

Dobraunig, C., Eichlseder, M., Korak, T., Mangard, S., Mendel, F., & Primas, R. (2018). SIFA: Exploiting Ineffective Fault Inductions on Symmetric Cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, *2018*(3), 547–572. https://doi.org/10.13154/tches.v2018.i3.547-572

Hoffman, C., (2013). *Brute-force attacks explained: how all encryption is vulnerable*, How-To-Geek. Available from: www.howtogeek.com [Accessed on June 01, 2021].

Karaklajić, D., Schmidt, J., Verbauwhede, I., (2013). *Hardware Designer's Guide to Fault Attacks, IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(12), 2295-2306, DOI: 10.1109/TVLSI.2012.2231707.

Malkin, T.G., Standaert, F.X., and Yung, M., (2006). *A comparative cost/security analysis of fault attack countermeasures*, FDTC'06: Proceedings of the Third International Conference on Fault Diagnosis and Tolerance in Cryptography; 159–172. Doi: https://doi.org/10.1007/11889700_15

Onwutalobi, A.C., and Lahden, M., (2011), *Overview of Cryptography*, SSRN Electronic Journal, 10.2139/ssrn.2741776.

Qadir, A.M, and Varol, N., (2019). *A review paper on cryptography*, 7th International Symposium on Digital Forensics and Security (ISDFS), Barcelos, Portugal, 1-6, DOI: 10.1109/ISDFS.2019.8757514.

Smirnoff, P. and Turner, D.M., (2019). *Cryptomathic. Symmetric key encryption -why, where, and how it is used in banking.* Available from: https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking [Accessed on August 20, 2021].

Stapko, T., (2008). *Practical embedded security, building secure resource-constrained systems.* DOI: https://doi.org/10.1016/B978-0-7506-8215-2.X5001-0

Wu, W., (2019). *Confusion and diffusion* CISSP/ISSMP/ISSAP/ISSEP,CCSP,CSSLP,CISM,PMP,CBAP. Available from: www.wentzwu.com [Accessed on August 20, 2021].