Peertechz

ENGINEERING GROUP

TRENDS IN Computer Science and Information Technology
OPEN ACCESS JOURNAL

ISSN: 2641-3086 DOI: https://dx.doi.org/10.17352/tcsit

Review Article

Counter-Offset mode: A new paradigm in resisting differential cryptanalysis

Muhammad Faisal Nawaz[1] and Yasir Nawaz[2]*

[1]University of Lahore, Pakistan

[2]Shanghai Jiao Tong University, China

Received: 19 February, 2024
Accepted: 29 March, 2024
Published: 30 March, 2024

***Corresponding author**: Yasir Nawaz, Shanghai Jiao Tong University, China, E-mail: my_nawaz@alumni.sjtu.edu.cn, my_nawaz@sjtu.edu.cn

ORCiD: https://orcid.org/0009-0001-3093-4475

Keywords: Block cipher modes; Symmetric encryption; Counter mode; Differential cryptanalysis; Concrete security; Pseudorandom function

https://www.engineegroup.us

Check for updates

Abstract

This study introduces the Counter-Offset mode, a novel advancement in block cipher encryption techniques designed to enhance the traditional Counter mode's resistance to differential cryptanalysis. By integrating a unique input transformation mechanism, the Counter-Offset mode significantly improves upon the security features of the conventional Counter mode without compromising its efficiency and ability to process data blocks in parallel. Through a rigorous security analysis, we demonstrate that this innovative mode not only maintains the essential advantages of its predecessor—including parallelizability and low overhead—but also offers increased protection against cryptanalytic attacks. Our findings suggest that the Counter-Offset mode presents a compelling solution for applications requiring high security without the expense of reduced performance. This work lays the groundwork for future investigations into its application across various cryptographic systems, highlighting its potential to address emerging security challenges in the digital age.

Introduction

Block cipher mode of operation is a scrutinized cryptographic primitive for secure encryption and decryption that ensures privacy, authenticity, and authenticated encryption [1,2]. In the last few years, researchers conducted a lot of research on block cipher modes, and it is believed that building an efficient and secure mode of operation for block cipher is now a big problem. For this purpose NIST recommends five modes of operation for the efficiency of block cipher, these operation modes are: Electronic Code Book (*ECB*), Cipher Block Chaining (*CBC*), Cipher Feedback (*CFB*), Output Feedback (*OFB*), Counter mode (*CTR*) and standardized in 2001 [3].

The cryptographic modes of operation have seen significant advancements, driven by the need for secure, efficient encryption methods suited to a variety of applications, from traditional computing environments to emerging technologies like the Internet of Things (IoT) and cloud computing [4]. The ECB and CBC have been foundational, with the ECB's simplicity being offset by its vulnerability to pattern analysis due to identical plaintext blocks producing identical ciphertext blocks [5]. CBC mode improved security by introducing data dependency through the XOR operation with the previous ciphertext block, although it also required careful management of initialization vectors (IVs) to prevent attacks [6]. The advent of CTR mode brought a paradigm shift with operations, highlighting the importance of non-repeating counters for maintaining security [7]. Recent research has also focused on authenticated encryption modes like GCM, which provide both encryption and integrity in a single operation, responding to the growing demand for data protection that encompasses both confidentiality and authentication [8].

As we move further into the decade, the cryptographic community has turned its attention to developing lightweight cryptographic solutions that cater to the constraints of IoT devices and ensuring encryption schemes can withstand the potential future threats posed by quantum computing [9]. The push towards quantum-resistant algorithms underscores a

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041. DOI: https://dx.doi.org/10.17352/tcsit.000078

proactive approach to cryptanalysis, with a keen eye on both current and future security landscapes [10]. Moreover, the increasing integration of encryption into everyday technologies has underscored the need for modes of operation that not only secure data against sophisticated attacks but do so with minimal impact on system performance and user experience [9]. This period has underscored the dynamic nature of cryptographic research, where innovation is not just about creating new encryption methods but also about adapting existing protocols to meet the evolving demands of technology and society [11].

Moreover, block cipher modes of operation have much attention lately and several other block cipher modes of operation suggested and analyzed [12]. Of many modes of operation, the *CTR* mode now a days are widely using mode of operation and has a number of desirable advantages than other modes. On the other hand, some authenticated encryption mode combined with *CTR* mode like *CCM* mode (*Counter with CBC-MAC)* [13] and that was designed as a non-patented alternative to *OCB* mode [14]. The authenticated encryption mode *EAX* [15] uses *CTR* mode for confidentiality and *OMAC* hash algorithm for authentication [16]. In such a way, *CWC* mode [17] combines CTR mode for a confidentiality with a Carter-Wegman universal hashing function over $2^{127}1$ field for authentication [18]. Let *E* be a block cipher with *n-bits* block length, let *ctr* be an *n-bit* counter, and message $m=(m_1,m_2...m_n)$ broken into *n-bit* blocks, the *CTR* mode work as follows. The keystream is $s=(s_1,s_2.....s_n)$ and ciphertext is $c=(c_1,c_2.....c_n)$.

$$c_i \leftarrow s_i \oplus m_i, \tag{1}$$

$$s_i \leftarrow E_k\left(ctr+i\right) \quad for\ i=1,\ldots,\ n, \tag{2}$$

$$ctr \leftarrow ctr+n. \tag{3}$$

Provable security is the standard security goal for modes of operation. The first two-formal notion of security (i.e. semantic security and polynomial security) for asymmetric encryption was first introduced by [19]. In the treating asymmetric setting given by Goldwasser, he says the symmetric case can be dealt with similarly, one ingredient missing in this view is a *CPA* model it is must be in symmetric setting. The four notion of security for symmetric encryption given by [20], and analyze the concrete security of different modes of operation under the attack assumptions of chosen-plaintext attack (*CPA*). These notions of security are following: *left-or-right* indistinguishability (*LR*), Real-or-Random indistinguishability (*RR*), Find-then-Guess security (*FTG*) and Semantic security (*SEM*). The security of cryptographic modes of operation, such as ECB, CBC, OFB, CFB, and Counter, is quantified through an advantage function. This function measures the maximum advantage an adversary could gain in compromising the mode's security. By establishing bounds on this advantage, cryptographers can assess and compare the relative security levels of different modes, ensuring that they remain robust against potential attacks. The variability of the boundaries of the advantage function directly impacts the perceived security of a cryptographic mode of operation. Tighter bounds (lower advantages) indicate stronger security, as they suggest minimal gain for an adversary attempting to breach the system. In setting these boundaries, assumptions about the adversary's capabilities, such as computational resources and access to plaintext-ciphertext pairs, are crucial. Restrictions often include limiting the adversary to polynomial-time computations and specifying the amount of data they can encrypt or decrypt. These assumptions and restrictions help in constructing a realistic security model, within which the cryptographic strength of different modes can be rigorously evaluated and compared.

W.Diffie and M. Hellman were first introduced to the counter mode (*CTR mode* [21]) and standardized by H.Lipmaa, P.Rogaway, and D. Wagner [22]. The *CTR* mode has significant efficiency advantages than existing modes of operation that recommended by NIST. Furthermore, it also give the better concrete security than other modes of operation [20]. On the other hand, *CTR* mode perceived disadvantages, its crucial for *CTR* mode that the counter value is not reuse in encryption. Inappropriately, if user reuses the counter, then all the security is loss. Usually there are small hamming difference in between successive *ctr* and *ctr+1*. Successive counter blocks are generated by a next-counter function, that is such a simple operation (i.e. integer increment). The next-counter function provides the uniqueness of the inputs of the underlying block cipher but cannot provide any security properties. These details could be important if the underlying block cipher has a crucial weakness, but they are not important when considering the underlying block to be secure (i.e. *AES*) [23,24]. The small hamming difference of successive counter blocks (*ctr*,*ctr*+1,... ,*ctr*+*n*) facilitate the differential cryptanalysis. So, details led to concern that the attacker can obtain many plaintext pairs with the known small plaintext difference.

In this paper we refine the *CTR* mode with a small additional overhead which is known as the *Counter-Off set* mode (*CTR-Off set*) that is very simple, fully parallelizable and efficient compared to conventional privacy-only *CTR* mode. The *CTR-Off set* mode achieves higher resistance against differential cryptanalysis than *CTR* mode and provides the concrete security as same as *CTR* mode. *CTR-Off set* Mode enhances unpredictability by using the block cipher to encrypt the counter values before they are used to generate the keystream. The use of XOR with these encrypted values and the original counter values adds another layer of randomness, thwarting potential cryptanalytic attacks that exploit predictability in the encryption process. This two-step process—encrypting the counter and then using XOR—transforms a predictable serial input into an unpredictable one, thereby enhancing the overall security of the cryptographic scheme.

This paper introduces the Counter-Offset mode, enhancing the traditional Counter mode's resistance to differential cryptanalysis. Section 1 outlines the evolution and significance of block cipher modes. Section 2 covers essential preliminaries and security analysis foundations. Section 3 elaborates on the Counter-Offset mode, its algorithm, and security benefits. Section 4 provides a detailed security analysis, demonstrating its superiority over the conventional Counter mode. Finally, Section 5 discusses the performance of the Counter-Offset

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041.
DOI: https://dx.doi.org/10.17352/tcsit.000078

mode and suggests future research directions, emphasizing its potential for securing cryptographic systems against advanced cryptanalytic techniques while retaining operational efficiency.

Preliminaries

In this section, we focus on some related definitions and their concrete security analysis. Our treatment follows the [1]. In [1], they described the security notion of symmetric encryption and analyzed the concrete security of three modes of operation (*XOR*,*CTR* and *CBC*) under *CPA* attack. Here we consider only *LR* under *CPA* attack, which gives the reduction among the other notions. If adversary *A* is a probabilistic algorithm, we define $d \leftarrow A(m_1,m_2)$ is the experiment of adversary *A* on m_1,m_2. A symmetric encryption scheme has tuples of algorithms *SE* = (*K*,*E*,*D*), where randomized key generation algorithm *K* returns a string *k*. The *encryption algorithm E* which might be randomized take a key $k \in key(SE)$ and a plaintext $M \in \{0,1\}^*$ to return ciphertext $C \in \{0,1\}^*$. The deterministic *decryption algorithm D* take a key $k \in key(SE)$ and a ciphertext $C \in \{0,1\}^*$ to return plaintext $M \in \{0,1\}^*$.

The resources of the adversary *A* are parameters of concrete security. Let *t* be the running time of adversary *A*, q_e be the number of encryption oracle queries. The amount of the ciphertext of adversary *A* corresponding to the oracle queries q_e are μ_e. The adversary is allowed queries of the form (m_0,m_1) to an oracle that we call *LR*, where m_0, m_1 are equal-length messages. The oracle returns a ciphertext *C*. We Consider two possible ways in which ciphertext is computed by the oracle corresponding to two different games in which *A* lives (*left* and *right*). In the right world, the oracle given query m_0,m_1 and runs *E* with key *k* and takes input m_1 and return a ciphertext *C*. In the left world, the oracle, given m_0,m_1 and runs *E* with key *k* and takes input m_0 and return a ciphertext *C*. In formal definition, the *LR* oracle is defined by $E_k(LR(\cdot,\cdot,b))$, where $b \in \{0,1\}$ to take input (m_0,m_1) and do the following: If $b=0$, it computes $C \leftarrow E_k(m_0)$ and return *C*. If $b=1$, it computes $C \leftarrow E_k(m_1)$ and return *C*. Now we can define the *LR-CPA* as the following.

Definition: Let *SE* = (*K*,*E*,*D*) be a symmetric encryption scheme. Let A_{cpa} be an adversary give access to the Oracle $E_k(LR(\cdot,\cdot,b))$, Consider the following experiment:

$$Exp_{SE,\,A_{cpa}}^{LR-CPA-b}(k) \tag{4}$$

$$k \xleftarrow{\$} k \tag{5}$$

$$d \leftarrow A_{cpa}^{E_k(LR(\cdot,\cdot,b))}(k) \tag{6}$$

The advantage of the adversary can be figured out as follows:

$$Adv_{SE,\,A_{cpa}}^{LR-CPA}(k) = Pr[Exp_{SE,\,A_{cpa}}^{LR-CPA-1}(k) = 1] - Pr[Exp_{SE,\,A_{cpa}}^{LR-CPA-0}(k) = 1] \tag{7}$$

So, we can define the advantage function as follows.

$$Adv_{SE}^{LR-CPA}(k,\ t,\ q_e,\mu_e) = \max_{A_{cpa}} \{ Adv_{SE,\,A_{cpa}}^{LR-CPA}(k)\} \tag{8}$$

We consider an encryption scheme to be good if the advantage of a reasonable adversary closes to zero meaning the adversary is not doing a good job. The symmetric encryption schemes are based on pseudorandom permutations (*PRP*) or pseudorandom functions (*PRF*). Let $perm^l$ be the family of all permutations on $\{0,1\}^l$ and $rand^{l \to L}$ be the family of all functions $\{0,1\}^l \to \{0,1\}^L$. We will not define *PRP* and *PRF*, for detail see [25]. The concrete security of the symmetric encryption schemes (*i.e.XOR* mode, *CTR* mode, and *CBC* mode) using random functions (*RF*), random permutations (*RP*), *PRP* and *PRF* are describe below. Let *F* be a function family having key-length *k*, input-length l, and output-length *L*. To specify the function, we will use $f = F_k$. The followings are specified the *XOR*, *CTR* and *CBC* modes respectively. The message *m* to be encrypted is regarded as a sequence of $l-bits$ blocks $(m=m_1,m_2...m_n)$.

The concrete security of the *XOR* mode

SE = (*K*,*E*,*D*) be the symmetric encryption scheme corresponding to *XOR* mode. The key generation algorithm *K*, just outputs a random key *k* for the underlying *PRF* family *F*, and specifying $f = F_k$ of $l-bits$

to L – bits.

Encryption XOR	Decryption XOR
$r \leftarrow \{0,1\}^n$ for i=1,...,n do $c_i = f(r+i) \oplus m_i$ return r || c1, c2...cn	Parse z as $_{r\|\|}c_1,c_2...c_n$ for i=1,...,n do $m_i = f(r+i) \oplus c_i$ return m= m1,m2...mn

***XOR* lower bound insecurity using a *RF*:** Let $R = rand^{l \to L}$ be the random function, then for any t,q_e and μ_e such that $\mu_e q_e / L \le 2^l$

$$Adv_{XOR[R]}^{LR-CPA}(\cdot,\ t,\ q_e,\mu_e) \ge 0.316\,\frac{\mu_e.\,(q_e-1)}{L.2^l} \tag{9}$$

Proof: Proposition 9 [20]

***XOR* upper bound insecurity using a *RF* :** Let $R = rand^{l \to L}$ be the random function, then for any t,q_e and μ_e such that.

$$Adv_{XOR[R]}^{LR-CPA}(\cdot,\ t,\ q_e,\mu_e) \le \frac{\mu_e.\,(q_e-1)}{L.2^l} \tag{10}$$

Proof: Lemma 10 [20]

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041. DOI: https://dx.doi.org/10.17352/tcsit.000078

***XOR* security using a *PRF*:** Let F be a *PRF* family with $l-bit$ input-length and L - *bit* output-length. Then, for any t,q_e and $\mu_e = q' L$,

$$Adv^{LR-CPA}_{XOR[F]}(\cdot,\ t,\ q_e,\mu_e) \le 2.\ Adv^{\text{pdf}}_{F}(t,q') + \frac{\mu_e.(q_e-1)}{L.2^{l}} \tag{11}$$

Proof: Theorem 11 [20]

The CTR mode achieves better security than that of the *XOR*. The adversary has no advantage in the ideal case.

The concrete security of the *CTR* mode

$SE = (K,E,D)$ be the symmetric encryption scheme corresponding to *CTR* mode. The key generation algorithm K, just outputs a random key k for the underlying *PRF* family F and specifying $f = F_k$ of $l-bits$ to $L-bits$.

Encryption CTR	Decryption CTR
for $i=1,...,n$ do $c_i = f(ctr+i) \oplus m_i$ $ctr \leftarrow ctr+n$ return $(ctr, ctr \| c_1 c_2 ... c_n)$	Parse z as $ctr \| c_1, c_2 ... c_n$ for $i=1,...,n$ do $m_i = f(ctr+i) \oplus c_i$ return $m = m_1, m_2 ... m_n$

***CTR* security using a *RF*:** Let $R = rand^{l\to L}$ be the random function, then for any t,q_e and $\mu_e \le L2^{l}$

$$Adv^{LR-CPA}_{CTR[R]}(\cdot,\ t,\ q_e,\mu_e) = 0 \tag{12}$$

Proof: Lemma 12 [20]

***CTR* security using a *RF*:** Let F be a *PRF* family with $l-bit$ input-length and $L-bit$ output-length. Then, for any t,q_e and $\mu_e = \min(q' L,\ L2^{l})$,

$$Adv^{LR-CPA}_{CTR[F]}(\cdot,\ t,\ q_e,\mu_e) \le 2.\ Adv^{\text{prf}}_{F}(t,q) \tag{13}$$

Proof: Theorem 13 [20]

Although in the CBC mode, $l = L$ is required, and each F_k should be a permutation, [1] consider the F is a *PRF* family $(l = L)$. Also, we will see the case that F is a *PRP*.

The concrete security of the *CBC* mode: $SE = (K,E,D)$ be the symmetric encryption scheme corresponding to *CBC* mode. The key generation algorithm k is the same as *XOR* mode just outputs a random key K for the underlying permutation family F(*CBC* mode required that $l = L$).

Encryption CBC	Decryption CBC
$c_0 \leftarrow \{0,1\}^{l}$	Parse z as $c_0 \| C_1, C_2 ... C_n$
for i=1,...,n do $c_i = f(c_{i-1}) \oplus m_i$ return $(c_0 \| c_1, c_2 ... c_n)$	for $i=1,...,n$ do $m_i = f^{-1}(c_i) \oplus c_{i-1}$ return $m = m_1, m_2 ... m_n$

Let us see the concrete security of the *XOR*, *CBC* and *CTR* modes. We first summarize the security of the *XOR* mode.

CBC lower bound insecurity using a RP: Let $RP = perm^{l}$ be the random permutation, then for $q_e = \frac{\mu_e}{l}$, and $\mu_e \le l.2^{\frac{l}{2}}$

$$Adv^{LR-CPA}_{CBC[RP]}(\cdot,\ t,\ q_e,\mu_e) \ge 0.316\left(\frac{\mu_e^{\,2}}{l^2} - \frac{\mu_e}{l}\right).\frac{1}{2^{l}} \tag{14}$$

Proof: Proposition 15 [20]

***CBC* upper bound insecurity using a *RF*:** Let $R = rand^{l\to L}$ be the random function, then for any t,q_e, and μ_e,

$$Adv^{LR-CPA}_{CBC[R]}(\cdot,\ t,\ q_e,\mu_e) \le \left(\frac{\mu_e^{\,2}}{l^2} - \frac{\mu_e}{l}\right).\frac{1}{2^{l}} \tag{15}$$

Proof: Lemma 16 [20]

***CBC* Security using a *PRP*:** Let F be a *PRP* family with length l. Then, for t,q_e, and $\mu_e = ql$,

$$Adv^{LR-CPA}_{CBC[F]}(\cdot,\ t,\ q_e,\mu_e) \le 2.\ Adv^{\text{prp}}_{F}(t,q) + q^2 2^{-l-1} + \left(\frac{\mu_e^{\,2}}{l^2} - \frac{\mu_e}{l}\right).\frac{1}{2^{l}} \tag{16}$$

Proof: Theorem 17 [20]

***CBC* Lower Bound Insecurity using a *RF*: [26]** Let $R = rand^{l\to L}$ be the random function, then for any t,q_e and $\mu_e \le l2^{\frac{l}{2}}$,

$$Adv^{LR-CPA}_{CBC[R]}(\cdot,\ t,\ q_e,\mu_e) \ge 0.316\left(1-\frac{2}{2^{\frac{l}{2}}}\right).\left(\frac{\mu_e^{\,2}}{l^2} - \frac{\mu_e}{l}\right).\frac{1}{2^{l}} \tag{17}$$

***CBC* Upper Bound Insecurity using a *RP*: [26]** Let $RP = perm^{l}$, Then, for any t,q_e and μ_e

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041.
DOI: https://dx.doi.org/10.17352/tcsit.000078

$$Adv^{LR-CPA}_{CBC[RP]}(\cdot, t, q_e, \mu_e) \leq \left(\frac{\mu_e^2}{l^2} - \frac{\mu_e}{l}\right) \cdot \frac{1}{2^l} \quad (18)$$

***CBC* security using a PRF: [26]** Let F be a *PRF* family with $l-bit$ input-length and L - *bit* output-length. Then, for any t, q_e and $\mu_e = ql$.

$$Adv^{LR-CPA}_{CBC[F]}(\cdot, t, q_e, \mu_e) \leq 2.\left(Adv^{\text{prf}}_{F}(t, q) + \left(\frac{\mu_e^2}{l^2} - \frac{\mu_e}{l}\right) \cdot \frac{1}{2^l}\right) \quad (19)$$

In the above, we can see that the *CTR* mode gives better security in a *RF* as compared to the other modes. There is no collision in the input strings underlying the function f, since the function f is in a random function. The adversary cannot distinguish in the *LR* sense. However, the *XOR*, *CBC*, mode and NIST recommended modes of operation may have collision on input strings of the underlying function f by the birthday paradox, which can leak some information to distinguish.

The *CTR-off set* mode: CTR-Offset Mode represents a significant evolution in counter-based encryption strategies, specifically designed to fortify cryptographic systems against sophisticated forms of cryptanalysis like differential and linear cryptanalysis [27]. Its innovation lies in the manner it manipulates the counter values to enhance the security provided by the block cipher, even when the cipher itself may have weaknesses.

Understanding the risks in traditional CTR Mode

In conventional Counter (CTR) mode, encryption proceeds by combining the plaintext with a keystream generated by encrypting a sequence of counter values. These counters are typically incremented by one, leading to a scenario where successive counter values have a small Hamming difference — meaning only a few bits change between one counter and the next. This small difference is systematic and predictable, which can be exploited by adversaries using differential cryptanalysis, especially if the underlying block cipher is not robust against such attacks. Similarly, linear cryptanalysis can take advantage of predictable relationships between the plaintext, ciphertext, and the key.

The genesis of CTR-offset mode

CTR-Offset Mode innovates by injecting a layer of unpredictability into the counter values before they are used to generate the keystream, depicted in Figure 1. This unpredictability is achieved by first encrypting the counter value and then XORing it with the counter itself to create a new, less predictable value. The result of this XOR operation, the offset-modified counter, is encrypted once more to produce the keystream.

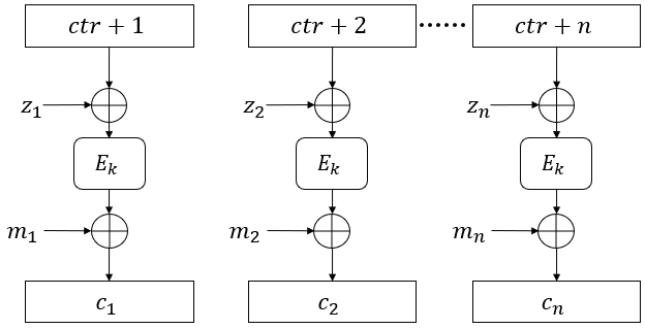

Figure 1: CTR-Offset Mode.

The dual encryption process serves a dual purpose: not only does it introduce more complexity into the keystream generation, making it harder for attackers to find useful correlations, but it also leverages the security of the block cipher to its fullest. By ensuring the input to the block cipher is unpredictable, CTR-Offset Mode makes each block's encryption independent of the others, significantly mitigating the risks posed by differential and linear cryptanalysis.

Encryption Algorithm for CTR-Offset Mode

Input:
- Plaintext divided into blocks: $p = m_1, m_2, \ldots m_n$
- Key: k, the secret key for the block cipher
- Initial Counter (Nonce): *ctr*, a unique value for each encryption session

Output:
- Ciphertext: $c = c_1, c_2, \ldots c_n$

Procedure:
1. For each plaintext block m_1, from 1 to n:
   - Compute $z_i = E_k(ctr+i)$, where **i** is the counter offset for the current block.
   - Compute $s_i = z_i \oplus (ctr+i)$.
   - Encrypt s_i using the block cipher with key k to get the keystream: $k_i = E_k(s_i)$.
   - XOR the keystream k_i with the plaintext block m_i to get the ciphertext block: $c_i = k_i \oplus m_i$.
2. Concatenate all c_i to form the ciphertext c.

Encryption in CTR-Offset Mode

Here is how the encryption process in CTR-Offset Mode works in more detail:

1. A counter value (typically starting from zero and incrementing) is prepared for each block of plaintext data that needs to be encrypted.
2. The counter value is encrypted using the block cipher (denoted as E_k) to create a temporary value.
3. This temporary value is then XORed with the counter value to produce a modified counter, which is substantially different from the original counter, thereby increasing unpredictability.
4. The modified counter is encrypted again with the same block cipher, producing the keystream.
5. The keystream is then XORed with the plaintext block to produce the ciphertext block.

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041. DOI: https://dx.doi.org/10.17352/tcsit.000078

This process is repeated for each block of data, with the counter incrementing each time as shown in the encryption algorithm.

Decryption Algorithm for CTR-Offset Mode

Input:

- Ciphertext divided into blocks: $c = c_1, c_2, \ldots c_n$
- Key: k, the same secret key used for encryption
- Initial Counter (Nonce): ctr the same unique value used during encryption.

Output:

- Plaintext: $p = m_1, m_2, \ldots m_n$

Procedure:

1. For each ciphertext block c_i, from 1 to n:
   - Compute $z_i = E_k(ctr+i)$, identical to the encryption process.
   - Compute $s_i = z_i \oplus (ctr+i)$.
   - Encrypt s_i using the block cipher with key k to get the keystream: $k_i = E_k(s_i)$.
   - XOR the keystream k_i with the ciphertext block c_i to recover the plaintext block: $m_i = k_i \oplus C_i$
2. Concatenate all m_i to form the plaintext p.

Decryption in CTR-Offset Mode

The same incremented counter values used during encryption are processed through the same steps to reproduce the keystream used for each block. The ciphertext block is XORed with the corresponding keystream to retrieve the original plaintext as shown in the decryption algorithm.

By encrypting the counter value before it is used to create the keystream, CTR-Offset Mode disrupts the pattern that might be exploited in differential or linear cryptanalysis. Each block is encrypted with a keystream based on a counter value that is no longer predictable after being passed through the block cipher and XOR operation. This makes it much more challenging for an attacker to deduce the key or find a systemic relationship within the encrypted data, even if they have access to multiple plaintext-ciphertext pairs.

Furthermore, the use of the same encryption function twice in generating the keystream does not compromise security but rather enhances it. The first encryption of the counter generates a temporary value that is entirely unrelated to the actual keystream. This temporary value, when XORed with the counter, produces a modified counter that bears no obvious relation to its original form. The second encryption of this modified counter then generates the actual keystream. The strength of this approach lies in its unpredictability — any patterns or predictability from the original counters are obscured through this process.

CTR-Offset Mode is a robust response to the vulnerabilities exposed in traditional CTR mode. It adds a layer of unpredictability that preserves the operational advantages of CTR such as the ability to encrypt blocks in parallel and the independence of each block's encryption while significantly bolstering its resistance to cryptanalysis. For environments where the underlying block cipher may have potential weaknesses, the CTR-Offset Mode offers a heightened level of security, making it a prudent choice for modern cryptographic applications.

Security analysis

We use the notion of security as the same as in section 2. In this analysis we take RF instead of RP, so F is with input length l, the output length L, and key length k. If the underlying block cipher is a secure PRF has an advantage value $\in'$ for resources $t'q$, then the advantage value of $CTR - off\ set$ mode is at most $2\in'$ for resources t = t', $\mu = q'l$ and any q. On the other hand, if the underlying block cipher under the assumption is ideal (meaning$\in'$ = 0), it is possible for the adversary to attack other existing modes (like CBC mode) and derive some advantage. This is not true for CTR mode and $CTR - off\ set$ mode.

The following theorem gives the concrete security of $CTR - off\ set$ mode.

Theorem: *CTR-Offset* security using a *RF*:

Let $R = rand^{l \to L}$ be the random function, then for any t, q_e and $\mu_e \leq L2^l$

Proof: Let (M_i, N_i) be the oracle queries of the adversary A, where I =1,...q denote $(M_1, N_1), \ldots (M_q, N_q)$. Each query consists of a pair of equal-length messages. Let n_i be the number of blocks in the i'th query. Let $ctr+i$ be the counter value associated to (M_i, N_i) as chosen at random by the oracle, for i=1,....q. In answering the i'th query, the oracle applies the underlying function f to the n_i string $ctr+\oplus z_i$. These strings called as $i'th\ sequence$ and $ctr+i$ is the $i'th$ point in this sequence, i=1,...n. Let D be the event defined for either game: $ctr+i \neq ctr+j$ whenever $i \neq j$ where i=1,...,n and j=1,...,n. That is D is the event that there are no overlapping sequences (no collision occurs) in input strings to the random function among all the queries. We define two probabilities of an event for either game: the Pr_0 in game 0 and Pr_1 in game 1.

$$\textbf{\textit{Claim}}\ 1: \ Pr_0\left[\bar{D}\right] = Pr_1\left[\bar{D}\right] \ for\ \mu_e \leq L2^l$$

Proof: We know that in the event D for either game, the input string does not have the same value for each query, since the counter values are different. In the input string corresponding to each block are associated with counter values and unpredictable value z. Thus, the input string does not repeat until 2^l block. So, the probability of each game is $Pr_0\left[\bar{D}\right] = Pr_1\left[\bar{D}\right] = 0 \ for\ \mu_e \leq L2^l$.

$$\textbf{\textit{Claim}}\ 2: \ Pr_0\left[A = 1 \middle| D\right] = Pr_1\left[A = 1 \middle| D\right]$$

Proof: In either game, the given event D, we have that the underlying function f evaluate at a new point each time. Therefore, the output of the underlying function f is randomly and uniformly. The consequence of this is that each block cipher has a distribution that is independent of any previous block cipher. So, we have $Pr_0\left[A = 1 \middle| D\right] = Pr_1\left[A = 1 \middle| D\right]$.

The advantage of the adversary A we compute is as follows.

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041. DOI: https://dx.doi.org/10.17352/tcsit.000078

$$Adv^{LR-CPA}_{CTR-Offset[R]}(\cdot,\ t,\ q_e, \mu_e) = Pr_1\left[A=1\right] - Pr_0\left[A=1\right]$$

$$Pr_1\left[A=1 \middle| D\right]\cdot Pr_1\left[D\right] + Pr_1\left[A=1 \middle| \bar{D}\right]. Pr_1\left[\bar{D}\right] -$$

$$Pr_0\left[A=1 \middle| D\right]\cdot Pr_0\left[D\right] + Pr_0\left[A=1 \middle| \bar{D}\right]\cdot Pr_0\left[\bar{D}\right]$$

Using claim 1 and claim 2, we have,

$$Adv^{LR-CPA}_{CTR-Offset[R]}(\cdot,\ t,\ q_e, \mu_e) = 0 \quad (20)$$

***CTR-Offset* security using a *PRF*:** Let *F* be a *PRF* family with $l-bit$ input-length and $L-bit$ output-length. Then, for any t,q_e, and μ_e = min (q' L, L2^l).,

$$Adv^{LR-CPA}_{CTR-Offset[F]}(\cdot,\ t,\ q_e, \mu_e) \le 2.\ Adv^{prf}_{F}(t, q') \quad (21)$$

Proof: The proof is achieved similar way to the theorem 13 [20]. The addition here we assume *F* to be a *PRP* family instead of once we get the security assuming *F* to be a *PRF* family. [1] assume the encryption function of counter mode is a *PRF*, not a *PRP* like *AES*. To apply the *PRP*, its necessary to apply proposition 8 from [20,24,28,29].

$$Adv\left[PRF\right]\left(t,\ q\right) \le Adv\left[PRP\right]\left(t,\ q\right) + q^2 2^{-l-1} \quad (22)$$

Performance and future work

Comparative security analysis: To strengthen the claims about the security and efficacy of CTR-Offset Mode, a comprehensive evaluation against existing modes of operation and potential baseline algorithms in the field is essential. This evaluation must encompass both theoretical analysis and practical performance metrics. The comparative security analysis will delve into the resistance of CTR-Offset Mode against differential and linear cryptanalysis compared to modes like ECB, CBC, CFB, OFB, and standard CTR. It will also include statistical tests to assess the randomness and unpredictability of its ciphertext, alongside establishing theoretical security bounds to prove that CTR-Offset Mode maintains, or exceeds, the security level of standard CTR mode without introducing new weaknesses.

Performance analysis: The performance analysis section will benchmark the throughput and latency of CTR-Offset Mode against other modes, considering the additional encryption steps it entails. It will also assess the computational resources required, such as CPU cycles, memory usage, and power consumption, and examine the mode's support for parallel processing. This analysis is critical for quantifying the impact of CTR-Offset Mode's security enhancements on its performance, especially in different environments like software, hardware, and cloud.

Implementation considerations: Implementing CTR-Offset Mode in real-world applications demands careful attention to several critical considerations to ensure the system's security and performance. This includes robust key management practices like secure key storage, regular key rotation policies, and proper nonce management to prevent nonce reuse. Performance considerations will address the mode's impact on encryption and decryption operations and strategies for mitigating performance overhead through hardware acceleration and parallel processing. Additionally, ensuring the unpredictability of the keystream, secure implementation practices to resist side-channel attacks, and adherence to cryptographic standards and regulatory compliance are essential for the successful deployment of CTR-Offset Mode.

Recommendations for future work: The paper will conclude with recommendations for future work, highlighting areas for further research and development. This may involve exploring more efficient implementations of CTR-Offset Mode, investigating its security in the context of quantum computing advances, or developing enhanced strategies for nonce generation. These recommendations will be based on the findings from the comparative security analysis, performance analysis, and implementation considerations, aiming to guide future efforts in advancing cryptographic practices.

By covering these aspects, the paper aims to provide a holistic view of CTR-Offset Mode's place within cryptographic practice, offering insights into its strengths, potential weaknesses, and practical considerations for implementation.

Conclusion

In this work, we presented the Counter-Offset mode, a novel adaptation of the traditional Counter mode, designed to significantly enhance resistance against differential cryptanalysis without forsaking the efficiency and parallelizability that are hallmarks of the original mode. Through meticulous analysis and comparison with established modes of operation, we have demonstrated that Counter-Offset mode not only retains the advantageous features of Counter mode but also introduces an additional layer of security by incorporating unpredictability into the encryption process. This enhancement addresses critical vulnerabilities, particularly in environments where the underlying block cipher might be susceptible to cryptanalytic attacks. Our findings affirm that the Counter-Offset mode stands as a robust, efficient, and secure mode of operation that aligns with the evolving landscape of cryptographic needs. Future work will focus on exploring the integration of Counter-Offset mode in real-world applications, optimizing its implementation, and further evaluating its performance and security in diverse scenarios. The continuous advancement of cryptographic methods, as exemplified by the development of the Counter-Offset mode, remains imperative in the pursuit of safeguarding digital information against increasingly sophisticated threats.

References

1. Rogaway P. Japan, Evaluation of some blockcipher modes of operation. 2011.
2. Katz J, Lindell Y. Introduction to modern cryptography. 2014. CRC press.
3. Dworkin M. Recommendation for block cipher modes of operation. Methods and techniques. National Inst of Standards and Technology Gaithersburg MD Computer security Div. 2001.

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041. DOI: https://dx.doi.org/10.17352/tcsit.000078

4. Mehmood A. Advances and Vulnerabilities in Modern Cryptographic Techniques: A Comprehensive Survey on Cybersecurity in the Domain of Machine/Deep Learning and Quantum Techniques. 2024; 12: 27530-27555.

5. Gava J. Assessment of Radiation-Induced Soft Errors on Lightweight Cryptography Algorithms Running on a Resource-constrained Device. 2023.

6. Karimov MM. Encryption Methods and Algorithms Based on Domestic Standards in Open-Source Operating Systems. 2023; 20: 42-49.

7. Usman H. Access Control and Privacy Preservation of Medical Records with Enhanced Rivest-Shamir-Adleman Algorithm Using Counter Mode Encryption. 2023.

8. Alkhyeli M. Secure Chat Room Application Using AES-GCM Encryption and SHA-256. In 2023 15th International Conference on Innovations in Information Technology (IIT). 2023. IEEE.

9. Thabit F. A comprehensive literature survey of cryptography algorithms for improving the iot security. 2023; 100759.

10. Dam DT. A survey of post-quantum cryptography: Start of a new race. 2023; 7(3): 40.

11. Salami Y. Cryptographic Algorithms: A Review of the Literature, Weaknesses and Open Challenges. 2023; 16(2): 46-56.

12. Stallings W. Cryptography and Network Security, 4/E. Pearson Education India. 2006.

13. Dworkin M. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. National Institute of Standards and Technology. 2004.

14. Rogaway P, Bellare M, Black J. OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Transactions on Information and System Security (TISSEC), 2003; 6(3): 365-403.

15. Bellare M, Rogaway P, Wagner D. A conventional authenticated-encryption mode. Manuscript. 2003.

16. Iwata T, Kurosawa K. OMAC: One-Key CBC MAC—Addendum. 2003.

17. Kohno T, Viega J, Whiting D. The CWC authenticated encryption (associated data) mode. ePrint Archives. 2003.

18. Wegman MN, Carter JL. New hash functions and their use in authentication and set equality. Journal of computer and system sciences. 1981; 22(3): 265-279.

19. Goldwasser S, Micali S. Probabilistic encryption. 1984; 28(2): 270-299.

20. Bellare M. A concrete security treatment of symmetric encryption. In Proceedings 38th Annual Symposium on Foundations of Computer Science. 1997. IEEE.

21. Diffie W, Hellman ME. Privacy and authentication: An introduction to cryptography. 1979; 67(3): 397-427.

22. Lipmaa H, Wagner D, Rogaway P. Comments to NIST concerning AES modes of operation: CTR-mode encryption. 2000.

23. Rijmen V, Daemen J. National Institute of Standards, and Technology. Advanced encryption standard. 2001; 19-22.

24. McGrew DA. Counter mode security: Analysis and recommendations. Cisco Systems, Inc 2002; 2: 4.

25. Bellare M, Kilian J, Rogaway P. The security of the cipher block chaining message authentication code. In Advances in Cryptology–CRYPTO. 1994.

26. Sung J. Concrete security analysis of CTR-OFB and CTR-CFB modes of operation. In International Conference on Information Security and Cryptology. 2001.

27. Wallén J. Design principles of the KASUMI block cipher. In Proceedings of the Helsinki University of Technology Seminar on Network Security. 2000.

28. Xian L, Tingthanathikul W. Advanced Encryption Standard (AES) in Counter Mode. ECE.

29. Sibleyras F. Cryptanalysis of the Counter mode of operation. 2017.

Citation: Nawaz MF, Nawaz Y (2024) Counter-Offset mode: A new paradigm in resisting differential cryptanalysis. Trends Comput Sci Inf Technol 9(1): 033-041.
DOI: https://dx.doi.org/10.17352/tcsit.000078