IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 8, 2020, accepted January 19, 2020, date of publication January 22, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968816

A New Method for Format Preserving Encryption in High-Data Rate Communications

ADRIÁN PÉREZ-RESA, MIGUEL GARCIA-BOSQUE, CARLOS SÁNCHEZ-AZQUETA, AND SANTIAGO CELMA

Electronic and Communications Engineering Department, University of Zaragoza, 50009 Zaragoza, Spain

Corresponding author: Adrián Pérez-Resa (aprz@unizar.es)

This work was supported in part by the Ministry of Economy and Competitiveness of Spain (MINECO)-European Regional Development Fund (FEDER) under Grant TEC2017-85867-R.

ABSTRACT In some encryption systems it is necessary to preserve the format and length of the encrypted data. This kind of encryption is called FPE (Format Preserving Encryption). Currently, only two AES (Advanced Encryption Standard) modes of operation recommended by the NIST (National Institute of Standards and Technology) are able to implement FPE algorithms, FF1 and FF3. These modes work in an electronic codebook fashion and can be configured to encrypt databases with an arbitrary format and length. However, there are no stream cipher proposals able to implement FPE encryption for high data rate information flows. The main novelty of this work is a new block cipher operation mode proposal to implement an FPE algorithm in a stream cipher fashion. It has been called CTR-MOD and it is based on a standard block cipher working in CTR (Counter) mode and a modulo operation. The confidentiality of this mode is analyzed in terms of its IND- CPA (Indistinguishability under Chosen Plaintext Attack) advantage of any adversary attacking it. Moreover, the encryption scheme has been implemented on an FPGA (Field Programmable Gate Array) and has been integrated in a Gigabit Ethernet interface to test an encrypted optical link with a real high data rate traffic flow.

INDEX TERMS FPE (format preserving encryption), stream cipher, FPGA (field programmable gate array), Ethernet.

I. INTRODUCTION

Format Preserving Encryption, FPE, is a kind of encryption used to cipher a plaintext preserving its original length and format [1]–[3]. In the past, some of the first FPE solutions [4], [5], were based mainly on the use of a standard binary block cipher working in a known operation mode. According to them, if the plaintext is in radix S, it must be added modulo-S to the block cipher output to produce the ciphertext. Although these techniques are based on standard modes of operation, also used to build stream ciphers, no argument for their security has been given. In addition, in some of them it is necessary to use an unbiasing operation when S is not a power of two [4].

There have been many other proposals for this type of encryption [6], but the only ones approved by the NIST (National Institute of Standards and Technology) are the modes FF1 and FF3 [7]. FF1, originally called FFX (Format-preserving Feistel-based Encryption), was proposed by Bellare *et al.* [8], whereas FF3 corresponds to the BPS-BC component proposed by Brier *et al.* [9]. Both operation modes are based on a non-binary Feistel structure, similar to that shown in Fig. 1, and they are able to encrypt blocks of data with an arbitrary format in an ECB (Electronic Code Book) fashion. In fact, although they are operation modes using AES as the underlying block cipher, they can be considered directly as a kind of FPE block ciphers.

Some application examples for FPE are the encryption of databases with an arbitrary format [6], [10]–[12] such as PANs (Primary Account Numbers) or SSNs (Social Security Numbers), which are not in binary format. Also, FPE can be used in communication systems when it is necessary to encrypt certain protocols, for example, in military or industrial environments [13], [14], or when encrypting some image formats [15].

Regarding the performance of FPE encryption methods, some studies have been done, however they are mainly related to software implementations [14], [16]–[18]. For performance benefits, a hardware implementation could be considered as in [13], [19] or [20].

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

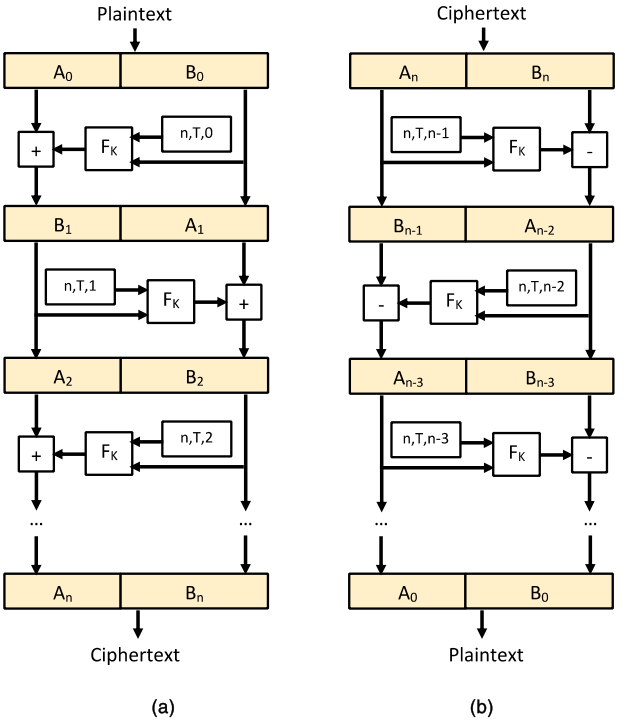

FIGURE 1. Generic cipher structure for format-preserving Feistel-based encryption: (a) ciphering (b) deciphering.

A hardware implementation of an FPE stream cipher could be advantageous for that cases where a high encryption rate is important and the plaintext format preservation is mandatory. A tentative solution for this cases could be also the usage of a standard stream cipher, however this would not be a valid solution. As an example, let us to imagine a plaintext formed by symbols in decimal radix. Each symbol will be represented with 4 bits. Then we could try to encrypt the plaintext thanks to a standard stream cipher generating a keystream formed by 4-bit words, which would produce 4-bit ciphertext symbols. As the keystream generator output could be considered random and uniformly distributed, then the XOR operation between the keystream and the plaintext would not guarantee a ciphertext also in decimal radix. For example, if the XOR operation between a symbol of the plaintext and the keystream produced a ciphertext value between 11 and 15, then the resulting value would not maintain the original format. If we needed to preserve the format (length and radix) of the plaintext, the described encryption mechanism with a standard stream cipher would not be valid.

An application example for the utility of FPE stream ciphers could be in [20] or [21], where a Gigabit Ethernet data flow must be encrypted at line rate preserving the 8b/10b encoding properties, which means preserving its format.

Some FPE stream ciphers have been proposed. For example, in the FF3 mode [9], the basic FPE block cipher component BPS-BC is proposed to be used in CBC (Cipher Block Chaining) mode, while in [20], it is proposed to be used in CTR (Counter) mode, as CTR can be considered the best and most modern way to achieve confidentiality-only

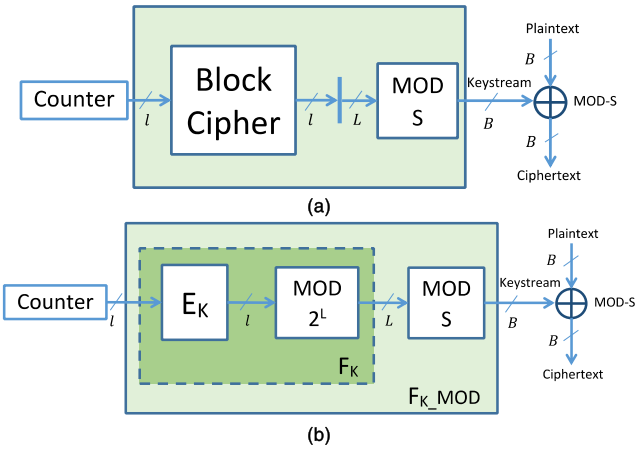

FIGURE 2. (a) Structure of the proposed keystream generator using a binary block cipher in CTR mode and a modulo-S operation. (b) Structure of PRF F_{K_MOD} decomposed in F_K and modulo-S operation. The output of F_K is the least significant L bits taken from the output of block cipher E_K which has a block size of l bits.

encryption [22]. In [21], another proposed solution consisted of using a conventional stream cipher whose output was subjected to a modulo-S operation to encrypt a plaintext in radix S. However, in this case the security is not clear as the bias introduced by the modulo-S operation is not analyzed.

The main novelty of this work is the proposal of an FPE stream cipher solution that reduces the hardware complexity of possible solutions based on FPE modes (FF1 and FF3) and is based on a recommended binary block cipher. Moreover, by means of a new operation mode that could use a standard block cipher, such as AES, it is possible to develop a formal security proof, in the same way that is usually done with traditional confidentiality-only operation modes, as in CTR or CBC. The formal security proof consists in the analysis of the IND-CPA (Indistinguishability under Chosen Plaintext Attack) advantage expression of any adversary attacking the proposed mode.

The proposed encryption mode in this work has been called CTR-MOD. To parametrize its resulting structure, a comparison of this mode with other taken as reference has been done in terms of their IND-CPA advantage expressions. The idea is to establish the condition under which CTR-MOD has at least the same or greater security than the reference mode when encrypting the same amount of information. Particularly, the mode used as reference has been CTR, since, as mentioned before, it can be considered the best to achieve confidentiality-only encryption [22].

CTR-MOD mode consists of a standard block cipher working in CTR mode plus a modulo-S operation applied to its output, which is added modulo-S to the plaintext. As shown in Fig. 2a, the keystream, plaintext and ciphertext will be also in radix S. In this figure the block size of the block cipher is l bits, from which L are taken and used as input for the modulo-S operator. The output values from the modulo-S operator will be in the range $\{0, \ldots, S-1\}$ and will be represented with B bits where $B = \lceil \log_2 S \rceil$.

Algorithm 1 $\{C, next\_ctr\} = CTR(M, ctr)$

$CNT_0 = ctr + 1$
Split M into m L-bit blocks $\{MB_0, MB_1, \ldots, MB_{m-1}\}$
$KB_i = F_K\,[CNT_i]$ *for* $i = 0, 1, \ldots, m-1$
$CNT_{i+1} = CNT_i + 1$ *for* $i = 0, 1, \ldots, m-2$
$CB_i = (MB_i \oplus KB_i)$ *for* $i = 0, 1, \ldots, m-1$
$next\_ctr = CNT_{m-1}$
$C = \{CB_0, CB_1, \ldots, CB_{m-1}\}$
Return $\{C, next\_ctr\}$

Algorithm 2 $\{C, next\_ctr\} = CTR - MOD(M, ctr)$

$CNT_0 = ctr + 1$
Split M into m symbols in radix S $\{MB_0, MB_1, \ldots, MB_{m-1}\}$
$KB_i = F_K\_MOD\,[CNT_i]$ *for* $i = 0, 1, \ldots, m-1$
$CNT_{i+1} = CNT_i + 1$ *for* $i = 0, 1, \ldots, m-2$
$CB_i = ((MB_i \oplus KB_i) mod S)$ *for* $i = 0, 1, \ldots, m-1$
$next\_ctr = CNT_{m-1}$
$C = \{CB_0, CB_1, \ldots, CB_{m-1}\}$
Return $\{C, next\_ctr\}$

The paper is divided in eight sections. In Section II both modes CTR and CTR-MOD are introduced, Section III details the IND-CPA advantage expression for CTR-MOD mode. Subsequently, Section IV makes a comparison between the security expression of CTR and CTR-MOD modes to parametrize the resulting structure of the proposed mode. In Section V the application case where the proposed mode has been applied, optical Gigabit Ethernet communication, is described. The implementation and some encryption results of CTR-MOD in the mentioned application case are shown in Sections VI and VII, respectively. Finally, in Section VIII conclusions are given.

II. CTR AND CTR-MOD MODES

Concrete security analysis for CTR mode was originally established in [23]. This operation mode is a stateful (counter based and deterministic) encryption scheme.

Let us consider a family of PRF (Pseudo Random Function) functions F such that $F : \mathcal{K} \times \{0,1\}^l \rightarrow \{0,1\}^L$ where L is the block size, $\mathcal{K}$ is the keyspace and F_K is a PRF from this family configured with a random key K taken randomly from $\mathcal{K}$ ($K \epsilon \mathcal{K}$). The plaintext is formed by a group of m L-bit blocks $M = \{MB_0, MB_1, \ldots, MB_{m-1}\}$ and it is encrypted resulting in a ciphertext formed also by m L-bit blocks $C = \{CB_0, CB_1, \ldots, CB_{m-1}\}$ thanks to its encryption function $CTR(M, ctr)$, as described in Algorithm 1.

The inputs of $CTR(M, ctr)$ are the message M itself and the initial value of the counter ctr, which is considered the state of this algorithm. CNT_i and KB_i are the values of the counter and keystream block in each encryption step. The l-bit counter values are encrypted thanks to the underlying encryption function F_K giving rise to the L-bit keystream blocks. The last counter value CNT_{m-1} will be used as next initial ctr value for the next invocation of $CTR(M, ctr)$.

The new proposed structure for the mode CTR-MOD of Fig. 2a can be decomposed as shown in Fig. 2b, where the block cipher in Fig. 2a has been modeled as a PRF E_K. The least significant L output bits of E_K are taken as input of the modulo-S operation, which is equivalent to perform the modulo-2^L operation at the output of E_K. In this proposed mode we have called F_K and E_K_MOD to the functions such that $F_K(x) = E_K\_MOD(x) = \left(E_K(x) \bmod 2^L\right)$. In this case, as in $CTR(M,ctr)$ algorithm, F_K in Fig. 2b can be considered a PRF such that $F : \mathcal{K} \times \{0,1\}^l \rightarrow \{0,1\}^L$, which means that it maps the space of values $\{0, \ldots, 2^l - 1\}$ to the range $\{0, \ldots, 2^L - 1\}$. In Fig. 2b the whole module formed by F_K and the modulo-S operation has been called F_K_MOD, which means that $F_K\_MOD(x) = F_K(x) \bmod S$. We can also consider F_K_MOD a PRF such that $F_K\_MOD : \mathcal{K} \times \{0,1\}^l \rightarrow \{0, \ldots, S-1\}$, as it maps integer values from $\{0, \ldots, 2^l - 1\}$ to $\{0, \ldots, S-1\}$, where S is not necessarily a power of two.

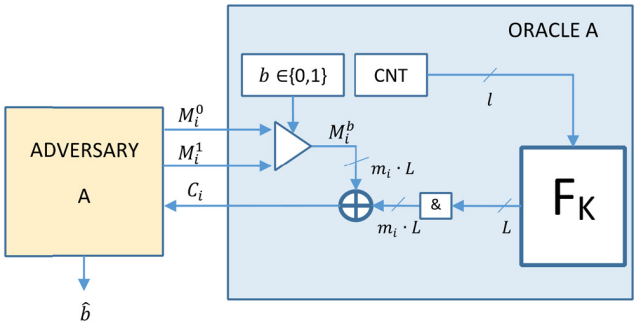

FIGURE 3. Scheme of the attack game between the adversary *A* and its oracle performing a CTR encryption scheme. Configuration bit *b* determines which message is encrypted during the game. After *s* queries the adversary outputs bit $\hat{b}$, meant as a guess at *b*.

By taking into account Fig. 2b and the definition of F_K_MOD, the encryption function of CTR-MOD scheme is described in Algorithm 2. This algorithm is similar to Algorithm 1 but using F_K_MOD function instead of F_K and the addition modulo-S instead of the XOR operation (addition modulo-2). Also, the plaintext, ciphertext and keystream are formed by m symbols in radix S instead of m L-bit blocks.

III. CTR-MOD IND-CPA SECURITY

Usually the security of traditional operation modes for only confidentiality is studied in the sense of IND-CPA security [24]. The attack model is a game between an active adversary A and an encryption oracle performing the encryption scheme $S\mathcal{E}$ configured with a key K and a configuration bit b.

During the game the adversary chooses a sequence of s pairs formed by two equal length messages $(M_1^0, M_1^1), \ldots, (M_s^0, M_s^1)$. For each pair of messages (M_i^0, M_i^1) the adversary receives from the oracle the ciphertext C_i corresponding to the message M_i^b. Finally the adversary must guess whether $(M_1^0, \ldots, M_s^0)$ or $(M_1^1, \ldots, M_s^1)$ were encrypted during the game. It means that the adversary has to guess the value of the configuration bit b after performing the s queries. Supposing that the $S\mathcal{E}$ is CTR, in Fig. 3 a scheme of the game is shown.

TABLE 1. Game definitions.

Game A: $E_{F'}^{nb}(A)$	Game B: $E_{F'}^{n}(A)$	Game C: $E_{F}^{nb}(A)$	Game D: $E_{F}^{n}(A)$
$K \xleftarrow{\$} \{0,1\}^k$	$K \xleftarrow{\$} \{0,1\}^k$	$K \xleftarrow{\$} \{0,1\}^k$	$K \xleftarrow{\$} \{0,1\}^k$
$f \xleftarrow{\$} Func(l,L)$	$f \xleftarrow{\$} Func(l,L)$	$f \xleftarrow{\$} Func(l,l)$	$f \xleftarrow{\$} Func(l,l)$
$Fun\_MOD \leftarrow f \bmod S$	$Func\_MOD \leftarrow f \bmod S$	$Func'\_MOD \leftarrow f \bmod 2^L$	$Func'\_MOD \leftarrow f \bmod 2^L$
$case\ (n)\ is$	$case\ (n)\ is$	$case\ (n)\ is$	$case\ (n)\ is$
$0: r \leftarrow Func\_MOD$	$0: g \leftarrow Func\_MOD$	$0: r \leftarrow Func'\_MOD$	$0: g \leftarrow Func'\_MOD$
$1: r \leftarrow F_K\_MOD$	$1: g \leftarrow F_K\_MOD$	$1: r \leftarrow E_K\_MOD$	$1: g \leftarrow E_K\_MOD$
$end\ case$	$end\ case$	$end\ case$	$end\ case$
$case\ (b)\ is$	$\hat{n} \leftarrow A(g)$	$case\ (b)\ is$	$\hat{n} \leftarrow A(g)$
$0: g \xleftarrow{\$} Func(l,T)$	Return $\hat{n}$	$0: g \xleftarrow{\$} Func(l,L)$	Return $\hat{n}$
$1: g \leftarrow r$		$1: g \leftarrow r$	
$end\ case$		$end\ case$	
$\hat{b} \leftarrow A(g)$		$\hat{b} \leftarrow A(g)$	
Return $\hat{b}$		Return $\hat{b}$	

In this figure, for each message M_i^b with a length of m_i L-bit blocks the oracle performs the CTR encryption as in Algorithm 1, generating m_i keystream blocks KB of length L bits. Symbol '&' represents the concatenation of the m_i keystream blocks that have to be XORed with M_i^b.

To measure the success of the adversary in breaking a symmetric encryption scheme S$\mathcal{E}$, the adversary advantage is defined in [25] as in the following equation:

$$ADV_{SE}^{IND-CPA}(A) = 2 \cdot \Pr\left(\hat{b} = b\right) - 1 \tag{1}$$

where the $ADV_{SE}^{IND-CPA}(A)$ is the IND-CPA advantage of the adversary A over the encryption scheme S$\mathcal{E}$, and $\Pr\left(\hat{b} = b\right)$ is the probability of the adversary A of guessing the correct value of configuration bit b. The advantage of A can be understood as the excess of this probability over 1/2. When the 'guess' probability is almost 1/2 and then the adversary advantage is negligible the encryption scheme S$\mathcal{E}$ can be considered secure.

IND-CPA advantage for CTR mode can be expressed as in Theorem 1, which is proven in [23].

Theorem 1: Let $F_K : \mathcal{K} \times \{0,1\}^l \rightarrow \{0,1\}^L$ be the underlying function of the encryption scheme S$\mathcal{E}$ that corresponds with CTR symmetric encryption mode. Let A be an adversary attacking the IND-CPA security of S$\mathcal{E}$ that asks at most s queries formed each one by a pair of messages (M_i^0, M_i^1) with a length of m_i blocks with L-bit length each one. The s message queries will produce a total number of qL-bit encrypted blocks which means that $q = \sum_{i=1}^{i=s} m_i$. Then an adversary B (attacking the PRF security of F_K and performing q queries) can be built thanks to A, such that:

$$ADV_{CTR}^{IND-CPA}(A) \leq 2 \cdot ADV_{F_K}^{PRF}(B) \tag{2}$$

where $ADV_{F_K}^{PRF}(B)$ is the prf-advantage [24] of any adversary B over F_K and $ADV_{CTR}^{IND-CPA}(A)$ is the IND-CPA advantage of A over the encryption scheme CTR.

Our purpose is to obtain the IND-CPA advantage for CTR-MOD to compare its security with the typical CTR scheme and in this way extract the necessary conditions to achieve at least the same or greater security. It can be proved that this advantage can be expressed as in the following theorem:

Theorem 2: Let $F_K\_MOD : \mathcal{K} \times \{0,1\}^l \rightarrow \{0,\ldots,S-1\}$ be the underlying function of the encryption scheme S$\mathcal{E}$ that corresponds with CTR-MOD symmetric encryption mode. Let A be an adversary attacking the IND-CPA security of S$\mathcal{E}$ that asks at most s queries formed each one for a pair of messages (M_i^0, M_i^1) with a length of m_i symbols of radix S. The s message queries will produce a total number of q encrypted symbols, which means that $q = \sum_{i=1}^{i=s} m_i$.

Then it is possible to build an adversary D (attacking the PRF security of E_K and performing q queries) such that:

$$ADV_{CTR-MOD}^{IND-CPA}(A) \leq 2 \cdot ADV_{E_K}^{PRF}(D) + \frac{q}{2^{I-1}} \tag{3}$$

where $ADV_{E_K}^{PRF}(D)$ is the prf-advantage of any adversary D over E_K as defined in [24], E_K corresponds to the block cipher that is part of the F_K_MOD function and I is the difference between L (the input bit length of modulo-S operation in F_K_MOD) and T, with $T = log_2 S$. The proof of this theorem is developed in the Appendix A thanks to the games described in Table 1 and the definition of the prf-advantage term $ADV_{F_K}^{PRF}$. The explanation of these games and the term $ADV_{F_K}^{PRF}$ are described in Appendix B.

IV. SECURITY ANALYSIS: CTR-MOD VS CTR

Although usually block ciphers are analyzed as PRFs, PRPs (Pseudo Random Permutation) are what best models them. Thanks to the PRF-PRP switching lemma [23] it is possible to relate the PRF and PRP advantages of an adversary against a block cipher as shown in (4).

$$ADV_{E_K}^{PRF}(A) \leq ADV_{E_K}^{PRP}(A) + \frac{q^2}{2^{l+1}} \tag{4}$$

TABLE 2. IND-CPA Advantage comparison of different modes.

Encryption Mode $\mathcal{SE}$	IND-CPA Advantage expression[1] $ADV^{IND-CPA}_{\mathcal{SE}}(A)$
CTR-MOD	$2 \cdot ADV^{PRP}_{E}(B) + \frac{\mu^2}{T^2 2^l} + \frac{\mu}{T \cdot 2^{l-1}}$
CTR	$2 \cdot ADV^{PRP}_{E}(B) + \frac{\mu^2}{l^2 2^l}$
CTR$	$2 \cdot ADV^{PRP}_{E}(B) + \frac{2\mu^2}{l^2 2^l}$
CBC	$2 \cdot ADV^{PRP}_{E}(B) + \frac{2\mu^2}{l^2 2^l}$
CFB[2]	$2 \cdot ADV^{PRF}_{F}(B) + \frac{\mu^2}{m^2 2^{l+1}}$

[1]In each expression l is the block size, μ the number of encrypted bits and T the bits mapped per symbol in CTR-MOD mode.

[2]The term ADV^{PRF}_{F}refers to the prf-advantage where F_K is the function $select(E_K(\cdot)).E_K(\cdot)$ is the block cipher with blocksize l and $select(\cdot)$ is a function that outputs m fixed bits from its input.

where $ADV^{PRF}_{E_K}(A)$ and $ADV^{PRP}_{E_K}(A)$ are the prf-advantage and prp-advantage of adversary A against block cipher E_K, respectively. The block size is l and the number of encryption queries performed by the adversary during the prf-advantage game is q.

According to (4), equations (2) and (3) of Theorems 1 and 2 can be rewritten as:

$$ADV^{IND-CPA}_{CTR}(A) \leq 2 \cdot ADV^{PRP}_{E_K}(B) + \frac{q^2_{CTR}}{2^{l_{CTR}}} \tag{5}$$

$$ADV^{IND-CPA}_{CTR-MOD}(A) \leq 2 \cdot ADV^{PRP}_{E_K}(D) + \frac{q^2_{CTR-MOD}}{2^{l_{CTR-MOD}}} + \frac{q_{CTR-MOD}}{2^{I-1}} \tag{6}$$

where E_K is the underlying block cipher used in both modes, q_{CTR} and $q_{CTR-MOD}$ are the number of encrypted blocks and symbols during the IND-CPA games of each mode, and l_{CTR} and $l_{CTR-MOD}$ are the block sizes of E_K in CTR and CTR-MOD, respectively. By each encrypted block CTR mode encrypts l_{CTR} information bits, while by each encrypted symbol CTR-MOD encrypts T information bits ($T = log_2 S$). Therefore the total number of encrypted bits in each mode during the IND-CPA game is $\mu_{CTR} = q_{CTR} \cdot l_{CTR}$ and $\mu_{CTR-MOD} = q_{CTR-MOD} \cdot T$, respectively.

According to this, it is possible to express the advantages of equations (5) and (6) in terms of the encrypted bits and compare them with the expressions of other well-known operation modes, as shown in Table 2.

As mentioned in Section I, we want to parametrize CTR-MOD and establish under what condition it has at least the same security as the classical CTR scheme when encrypting the same amount of information, with $\mu_{CTR} = \mu_{CTR-MOD} = \mu$. It means that we want to know the constraints needed to get the following condition:

$$ADV^{IND-CPA}_{CTR-MOD}(A) \leq ADV^{IND-CPA}_{CTR}(A) \tag{7}$$

If we assume that E_K is in both modes a secure and a recommended cipher we can consider that it is a good PRP and has a great prp-security, which means that the term $ADV^{PRP}_{E_K}$ is negligible for both expressions (5) and (6). Then it is only needed to compare the second terms of both expressions to meet (7), as shown in the following equation:

$$\frac{q^2_{CTR-MOD}}{2^{l_{CTR-MOD}}} + \frac{q_{CTR-MOD}}{2^{I-1}} \leq \frac{q^2_{CTR}}{2^{l_{CTR}}} \tag{8}$$

As $q_{CTR} = \mu_{CTR}/l_{CTR}$, $q_{CTR-MOD} = \mu_{CTR-MOD}/T$, and $\mu_{CTR} = \mu_{CTR-MOD} = \mu$ then (8) can be rewritten as:

$$\frac{\mu^2}{T^2 \cdot 2^{l_{CTR-MOD}}} + \frac{\mu}{T \cdot 2^{I-1}} \leq \frac{\mu^2}{l^2_{CTR} \cdot 2^{l_{CTR}}} \tag{9}$$

It is possible to rewrite (9) as:

$$\frac{1}{\mu} \leq T \cdot 2^{I-1} \cdot \left(\frac{1}{l^2_{CTR} \cdot 2^{l_{CTR}}} - \frac{1}{T^2 \cdot 2^{l_{CTR-MOD}}} \right) \tag{10}$$

In (10) $I = L - T$, where L is the input bit length of modulo-S operation in CTR-MOD mode. If we define the difference in bits between the output of the block cipher E_K and the input to the modulo-S operation as $P = l_{CTR-MOD} - L$, it is possible to rewrite (10) as:

$$L \geq T + 1 + \log_2 \left(\frac{l^2_{CTR} \cdot 2^{l_{CTR}}}{T} \cdot \left(\frac{1}{\mu} + \frac{1}{T \cdot 2^{P+T+1}} \right) \right) \tag{11}$$

As $\mu \geq 1$ and $P \geq 0$, then the lowest bound for L that always meets (11) is:

$$L \geq T + 1 + \log_2 \left(\frac{l^2_{CTR} \cdot 2^{l_{CTR}}}{T} \cdot \left(1 + \frac{1}{T \cdot 2^{T+1}} \right) \right) \tag{12}$$

It is possible to conclude that if the underlying block ciphers used in CTR and CTR-MOD modes have the same prp-security, and the block size $l_{CTR-MOD} \geq L$, then CTR-MOD scheme can have equal or better IND-CPA security than CTR when encrypting the same amount of information. It is important to notice that the block size $l_{CTR-MOD}$ used in CTR-MOD will depend on the block size l_{CTR} of the CTR mode used as reference and the radix S of the plaintext, as $T = log_2 S$. Expression in (12) will be the constraint necessary to achieve condition (7).

V. APPLICATION CASE: ETHERNET 1000BASE-X

As we have mentioned in Section I, as far as the authors are concerned, there are no standardized solutions for FPE stream ciphers and its usage could be relevant in the cases where a high encryption rate is necessary. For example, in the case of the encryption in 1000Base-X standard for Gigabit Ethernet optical links [20] .

The encryption in a layered communication model such as TCP/IP can be performed at different levels of the communication, such as in layers 2 or 3 with MACsec or IPsec standards, respectively. Although encryption in physical layer (layer 1) is less usual than in other layers some proposals

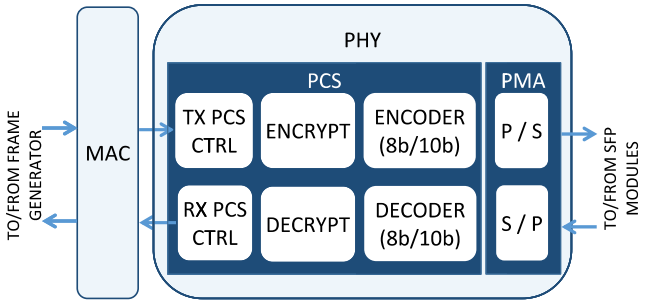

FIGURE 4. Scheme of the Ethernet Interface formed by the PHY and MAC (Medium Access Control) modules. MAC layer builds the Ethernet packets transmitted to the PHY, which includes the PCS and PMA (Physical Medium Attachment) sublayers. ENCRYPT and DECRYPT modules perform the format preserving encryption/decryption of 8b/10b symbols at the PCS sublayer. P/S and S/P modules are Parallel to Serial and Serial to Parallel modules, that transmit and receive the bitstream from the optical link.

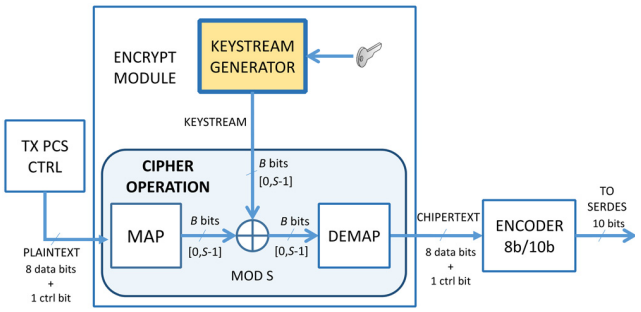

FIGURE 5. Location and generic structure of a stream cipher in a physical layer with 8b/10b line encoding. 8b/10b symbols are formed by eight data bits and one control bit. These symbols are mapped in CIPHER_OPERATION block thanks to MAP and DEMAP modules. The mapped symbols are represented with $B = \lceil \log_2 S \rceil$ bits.

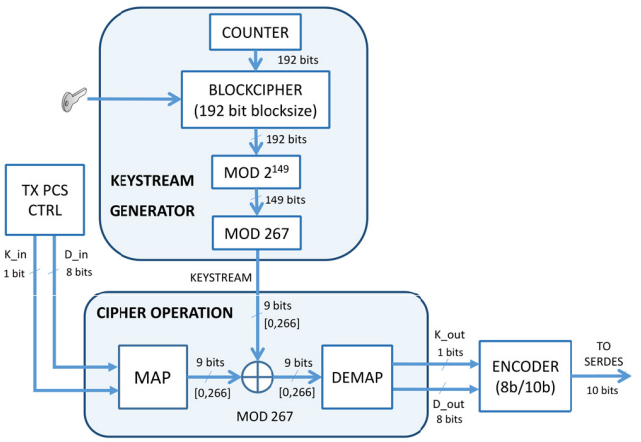

FIGURE 6. Overall structure for the streaming encryption system in a physical layer with 8b/10b encoding using CTR-MOD mode. Decryption will be as encryption but using a modulo-267 subtraction instead of an addition. 8b/10b symbols are formed by 8 data bits, D_in, and one control bit K_in. The symbols are mapped to 9-bit values (B = 9) in the range [0, 266] as S = 267.

have been made, for example related with photonic [26]–[29] or radio [30], [31] technologies or with the physical layer protocols [20], [21], [32].

One of the key benefits of performing the encryption at layer 1 is the possibility of masking the data traffic pattern, then achieving additional privacy, which permits to hide the existence of data transmission as in [26] with the optical steganography or in [20] with the encryption at encoding sublayer.

Our application case is the standard 1000Base-X, as in [20], [21]. In both cases the encryption is performed in one of the sublayers of physical layer, where the 8b/10b encoding is performed. This sublayer is called PCS (Physical Coding Sublayer). The 8b/10b encoding at PCS is used to provide certain properties to the bitstream that is transmitted through the Gigabit Ethernet optical link, such as DC balance, high transition density and short run length. Ciphering at PCS level must be performed in a way that the 8b/10b encoding properties are preserved, which means that the encryption method must preserve the same format in the plaintext and the ciphertext.

On the other hand, encryption and decryption modules must be located in the 1000Base-X datapath as shown in Fig. 4, where an optical Ethernet interface is shown.

In order to preserve the coding properties, the encryption of an 8b/10b symbol must give as result another valid 8b/10b symbol, which means to perform an FPE encryption. Ciphered symbols must be within the alphabet of symbols supported by the encoding standard. For this reason, the generic structure of the ENCRYPT module is built as shown in Fig. 5.

In Fig. 5, since S is the possible number of valid 8b/10b symbols, these are mapped to an integer value in the range $\{0, \ldots, S-1\}$, giving rise to a plaintext in radix S. Then a modulo-S addition is performed between the mapped symbols and a keystream also represented with values in the range $\{0, \ldots, S-1\}$. After that, the resulting ciphered values are reverse-mapped to its corresponding 8b/10b symbols which are finally encoded to 10-bit values and sent to the serializer.

In the 1000Base-X standard only 267 possible symbols are valid in the 8b/10b encoding, which means that S =267.

In this work, CTR-MOD operation mode has been used to perform the keystream generation and the modulo-S addition of Fig. 5, in the same way as Algorithm 2 of Section II.

VI. SYSTEM IMPLEMENTATION

We assume that we want to achieve at least the same or better IND-CPA security than a recommended block cipher working in CTR mode. Let be the block and key size of this reference block cipher a standard length of 128 bits. According to (12) if $l_{CTR} = 128$ bits, and $T = log_2 S \cong 8.06$ bits, then $l_{CTR-MOD}$ must be $l_{CTR-MOD} \geq L \geq 149$ bits to achieve the security that meets condition (7). By taking into account these parameters, proposed CTR-MOD structure has been adapted to the generic scheme of Fig. 5. The resulting encryption scheme is shown in Fig. 6.

As the underlying block cipher must have a block size greater than 128 bits because $l_{CTR-MOD} \geq 149$, the well-known Rijndael [33] cipher has been used. The main difference between Rijndael and AES (Advance Encryption Standard) is the range of configuration values for the block

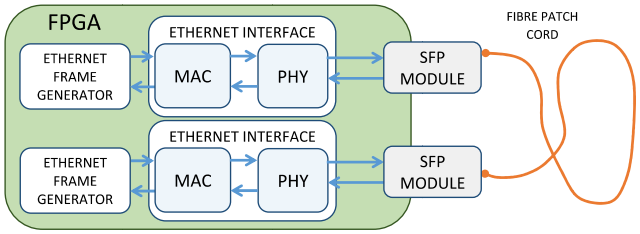

FIGURE 7. Test set-up scheme.

TABLE 3. Comparison with other hardware solutions.

Resource type	FF1 [13]	FF3 [13]	FPE [20]	SC[2] [21]	This Work
Slice Registers	11285	5592	11127	6097	8807
Slice LUTs	7426	3587	16978	13391	10974
18K Block RAMs	343	170	77	0	78
DSP cells	0	0	0	144	0
Slices[1]	3268	1596	5636	4110	3844
Encryption Rate (Mbps)	41.1	109.6	1000	1000	1000
Encryption Rate/Slice (Kbps/Slice)	12.57	68.7	177.4	243.3	260.1

[1]Slices are estimated from the number of register and LUTs, assuming they are not packed together.
[2]For the particular case of [21] the amount of resources has been calculated supposing the same input width in the modulo-267 operation.

size and the key length, in fact AES is a subset of Rijndael. While in AES the block size is fixed to 128 bits, in Rijndael it can take three values, 128, 192 and 256 bits. In this work Rijndael has been configured with 192 bit block size and a 128 bit key length, which means $l_{CTR-MOD} = 192$.

The block MOD_2^{149} of Fig. 6 is simply to take the 149 least significant bits of the Rijndael output. However, the second modulo operation, the MOD_267 module, takes more resources as 267 is not a power of two. Its implementation has been based on [34], which presents a high-speed hardware structure for a generic operation '*x mod z*'.

The final structure shown in Fig. 6 has been synthetized using a Xilinx Virtex 7 FPGA. Regarding the hardware resources used, they have been compared in Table 3 with other solutions based in the mentioned FF1 and FF3 modes for FPE [13], [20], and an ad-hoc stream cipher [21]. In this table, the amount of registers, LUTs (Look Up Tables) and BRAMs (Block RAMs) are shown. According to these results it is possible to conclude that the solution in this work achieves a better figure in *Encryption_Rate/Slice* than the others.

Finally, to test the encryption mechanism with real traffic, the KEYSTREAM_GENERATOR and CIPHER OPERATION modules of Fig. 6 have been integrated in the ENCRYPT and DECRYPT modules of the Ethernet Interface in Fig. 4. Two Ethernet Interfaces have been implemented in the Xilinx FPGA platform as shown in Fig.7. The PHY sides of both interfaces have been connected to optical SFP (Small Form Factor Pluggable) modules able to transmit at 1 Gbps data rate through a fiber link. Also two Ethernet Frame Generators have been connected to the MAC sides of both Ethernets Interfaces to test the encrypted link with real traffic. With these generators it is possible to produce Ethernet frame flows configured with different frame size, payload and interframe gap.

Note that in Table 3 we have only compared the solution in this work with other FPE solutions. A comparison of the proposed system with other well-known binary stream cipher implementations could be done. However, as mentioned in Section I this kind of ciphers cannot be a valid solution to preserve the format of the plaintext. In addition, although stream ciphers are suitable for high speed applications, their cryptanalysis and design criteria are less understood than block ciphers [35]. Indeed, a stream cipher application can be implemented easily thanks to a secure block cipher such as AES working in CTR mode, considered secure thanks to its formal security proof [23]. In the same way CTR-MOD mode can be considered enough safe as an FPE stream cipher structure.

VII. ENCRYPTION RESULTS

As mentioned in previous Section, one of the key benefits of performing the encryption at layer 1 is the possibility of masking the data traffic pattern, achieving additional privacy, as possible passive attackers are not able to detect the presence of current communications.

In the particular case of 1000Base-X, the transmission of 8b/10b symbols is carried out constantly, including in the case no frame is being transmitted or during the gap between frames. In these situations, the PHY always transmits idle sets of 8b/10b symbols, whose purpose is to maintain the synchronization between remote terminals.

To check the masking capability of the encryption, the SE (Shannon Entropy) in (13) has been measured for different encrypted and non-encrypted frame traffic patterns. The 8b/10b symbol stream for each traffic pattern, mapped between 0 and $S-1$, has been grouped in tuples of t symbols called β_t, and the probability for each tuple, $P(\beta_t)$, has been calculated. Particularly, SE has been measured for values of t equals to 1 and 2.

$$SE = -\frac{1}{t} \cdot \sum_{0 \le \beta_t < S^t} P(\beta_t) \cdot \log_2 P(\beta_t) \qquad (13)$$

Ideally, if every t-tuple (β_t) is equally likely with probability $P(\beta_t) = p = S^{-t}$ the value of Shannon Entropy for every t should be equals to $SE = \log_2 S = \log_2 267 \cong 8.0606$.

In Fig. 8 the SE measured for different traffic patterns is shown. Pattern A corresponds with no frame transmission, where only idle sets are transmitted over the link. Patterns B, C and D correspond to continuous frame transmission of 1024-bytes length with random payload at rates of 10.2%, 50% and 91% of the maximum Gigabit line rate. Pattern E corresponds to continuous frame transmission of random

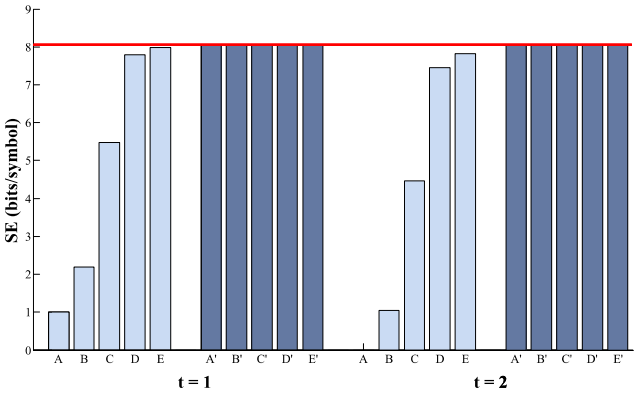

FIGURE 8. Shannon Entropy measured for different traffic flows of 8b/10b symbols grouped in tuples of t symbols with t from 1 to 2. Ethernet traffic patterns are called A, B, C, D and E and their encrypted versions, A', B', C', D' and E'. The red line marks the maximum entropy achievable with an 8b/10b data flow.

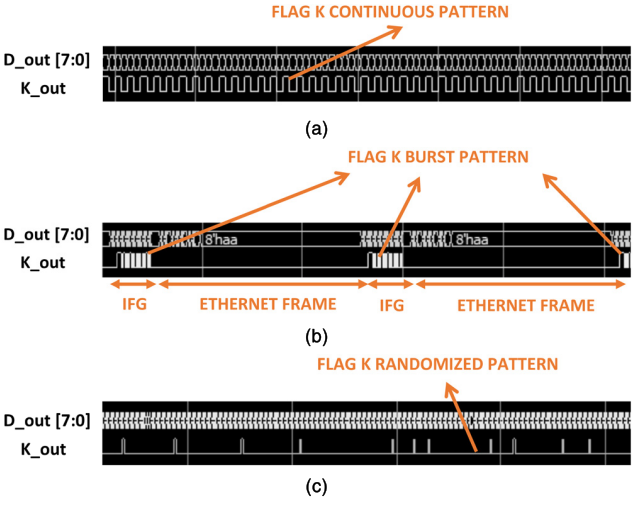

FIGURE 9. (a) K_out flag pattern without encryption when no Ethernet frame is transmitted; (b) K_out flag pattern without encryption when transmitting an Ethernet frame burst; (c) K_out flag pattern after encryption regardless of the transmission or non-transmission of Ethernet frames.

length and payload and minimum gap between frames. The non-encrypted versions of these patterns have different *SE*, however every encrypted version has the maximum possible *SE*, which makes them indistinguishable from each other and therefore proves the masking property, as shown in Fig. 8.

Moreover, to appreciate graphically this masking property is interesting to show the signal waveforms of 8b/10b flows after the encryption module. As shown in Fig. 6, 8b/10b symbols are formed by one control bit and eight data bits. In Fig. 6, after the encryption operation, these are called *K_out* and *D_out*, respectively, and they are used by the 8b/10b encoder to generate 10 bits. Each 8b/10b symbol is a control or data one depending whether its *K_out* flag is '1' or '0', respectively. When no frames are transmitted (pattern A) and encryption is disabled, Ethernet physical layer always transmit continuously idle sets composed by two consecutive symbols, one control symbol (with *K_out* equals to '1') and a data symbol (with *K_out* equals to '0'). It means that the *K_out* pattern in this situation is a signal that switches continuously between '0' and '1', as shown in Fig. 9a. When frames are transmitted (patterns B, C, D, E), the *K_out* flag will behave as a blast signal, as the idle sets are only transmitted in the space between frames. During frame transmission only 8b/10b data symbols are transmitted which will make *K_out* flag remain to '0' as shown in Fig. 9b.

Finally, by enabling encryption, 8b/10b symbols are ciphered and *K_out* flag and *D_out* are randomized (in every flow A, B, C, D or E) making indistinguishable which pattern is being transmitted, as shown in Fig. 9c.

VIII. CONCLUSION

In this work the authors have presented a new block cipher operation mode able to preserve the format of the plaintext when encrypting a high speed data stream. A security analysis has been made proving that it is possible to get at least the same or better security than a traditional CTR mode working with a standard 128-bits block cipher. The proposed solution permits the use of classical block ciphers instead of ad-hoc stream ciphers whose cryptanalysis and design criteria are less understood, or FPE structures whose architecture is more complex and entails larger hardware resources.

A parametrization and implementation of this FPE proposal has been carried out for the specific case of optical Gigabit Ethernet communications, achieving an *Encryption_Rate/Slice* better than in other existing FPE solutions, including FF1 and FF3 modes.

Finally, the masking property of the encryption at physical layer in 1000Base-X standard has been tested with the proposed operation mode, checking that different data patterns can be made indistinguishable from each other, including from the situation of no frame transmission.

APPENDICES

APPENDIX A

PROOF OF THEOREM 2

According to [23], and taking into account that CTR-MOD is a CTR-like mode but using a modulo-S addition (instead of an XOR operation) and an underlying PRF function F_K_MOD that maps integer values from $\{0,\ldots,2^l-1\}$ to $\{0,\ldots,S-1\}$ instead from $\{0,\ldots,2^l-1\}$ to $\{0,\ldots,2^L-1\}$ as F_K, it is possible to express the IND-CPA security of CTR-MOD in a similar way as in (2) such that:

$$ADV^{IND-CPA}_{CTR-MOD}(A) \leq 2 \cdot ADV^{PRF}_{F_K\_MOD}(B) \qquad (14)$$

where $ADV^{PRF}_{F_K\_MOD}(B)$ is the prf-advantage of an adversary B over F_K_MOD. Therefore, by obtaining this advantage it will be possible to get the expression for the IND-CPA advantage over the proposed scheme. The generic prf-advantage $ADV^{PRF}_{F_K}$ of any adversary over a PRF F_K is defined in [24] and an explanation about it is detailed in the Appendix B.

Thanks to the games defined in Table 1 it is possible to express the term $ADV^{PRF}_{F_K\_MOD}(A)$ according the following lemma:

Lemma 1: Let be F_K and E_K PRFs taken from the families of functions $F : \mathcal{K} \times \{0,1\}^l \rightarrow \{0,1\}^L$ and $E : \mathcal{K} \times \{0,1\}^l \rightarrow \{0,1\}^l$. Let be *Func_MOD* and *Func′_MOD* two functions defined as in Table 1. On the one hand, $Func\_MOD(x) = f(x) \, mod S$, where f is a random function taken from the set of functions $Func(l, L) : \{0,1\}^l \rightarrow \{0,1\}^L$. On the other hand $Func'\_MOD(x) = f(x) \, mod 2^L$, where f is a random function taken from the set of functions $Func(l, l) : \{0,1\}^l \rightarrow \{0,1\}^l$. Then their prf-advantages can be related as in the following equation:

$$ADV^{PRF}_{F_K\_MOD}(A) = ADV^{PRF}_{E_K}(D) + ADV^{PRF}_{Func'\_MOD}(A) + ADV^{PRF}_{Func\_MOD}(A) \quad (15)$$

In addition, it is possible to express the prf-advantages of any adversary A over the functions *Func_MOD* and *Func′_MOD* according to the Lemmas 2 and 3, respectively:

Lemma 2: Let be *Func_MOD* a function defined as in Lemma 1. Then any adversary A making q oracle queries when attacking the prf-security of *Func_MOD* will obtain an advantage bounded by the following expression:

$$ADV^{PRF}_{Func\_MOD}(A) \leq \frac{q}{2^I} \quad (16)$$

where $I = L - T$ and $T = log_2 S$.

Lemma 3: Let be *Func′_MOD* a function defined as in Lemma 1. Then any adversary A making q oracle queries when attacking the prf-security of *Func′_MOD* will obtain an advantage such that:

$$ADV^{PRF}_{Func'\_MOD}(A) = 0 \quad (17)$$

By taking into account Lemma 2 and Lemma 3 it is possible to derive the following expression from equation (15):

$$ADV^{PRF}_{F_K\_MOD}(A) \leq ADV^{PRF}_{E_K}(D) + \frac{q}{2^I} \quad (18)$$

Finally, by substituting the expression of the term $ADV^{PRF}_{F_K\_MOD}(A)$ in equation (14) it is possible to derive (3) which proves Theorem 2. For clarity purposes, proofs of Lemmas 1, 2 and 3 are given in the Appendices C, D and E using the games definitions of Table 1. Game definitions are explained in Appendix B.

APPENDIX B
GAME DEFINITIONS

In this Section the definition of term $ADV^{PRF}_{F_K}$ and the explanation of games in Table 1 are described. These are necessary to develop the proof of Theorem 2 and Lemmas 1, 2 and 3.

A. PRF ADVANTAGE $ADV^{PRF}_{F_K}$

The generic prf-advantage $ADV^{PRF}_{F_K}(A)$ of any adversary A over a PRF F_K is defined in [24] according to the following definition:

Definition 1: Let $F : \mathcal{K} \times \{0,1\}^l \rightarrow \{0,1\}^L$ be a family of functions, and let adversary A be an algorithm that takes from an oracle a function $g: \{0,1\}^l \rightarrow \{0,1\}^L$, that can be configured as a random function or a PRF F_K depending on a bit b, and tries to distinguish which function has been taken. The function g is passed to the adversary as an argument and it returns a result bit $\hat{b}$ ($\hat{b} \leftarrow A(g)$) meant as a guess at b. According to the game EX^b_F in Table 4 the prf-advantage of A over F_K is defined as:

$$ADV^{PRF}_{F_K}(A) = P\left[EX^1_F(A) = 1\right] - P\left[EX^0_F(A) = 1\right] \quad (19)$$

where $P\left[EX^b_F(A) = 1\right]$ is the probability that the result of the game $EX^b_F(A)$ is $\hat{b} = 1$.

TABLE 4. Game EX^b_F.

Game $EX^b_F(A)$
$K \xleftarrow{\$} \{0,1\}^k$; $f \xleftarrow{\$} Func(l, L)$
case (b) *is*
0: $g \leftarrow f$
1: $g \leftarrow F_K$
end case
$\hat{b} \leftarrow A(g)$; Return $\hat{b}$

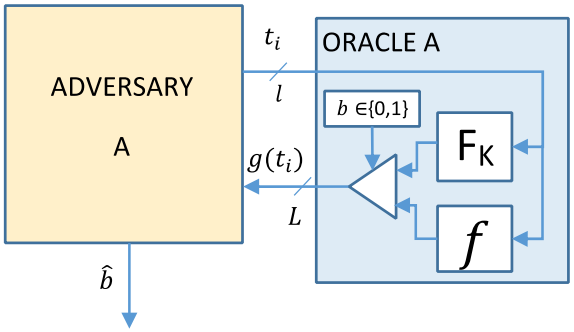

FIGURE 10. Scheme of the game $EX^b_F(a)$ between the adversary A and its oracle. Configuration bit b determines which function is being implemented by the oracle, F_K or f. The outputs of the resulting function g are analyzed by A to return $\hat{b}$, meant as a guess at b.

The game $EX^b_F(A)$ is configured with the bit b. If $b = 1$ a key of length k bits is selected randomly from the keyspace $\mathcal{K}$. It is expressed as $K \xleftarrow{\$} \{0,1\}^k$, where the operator $\xleftarrow{\$}$ means a 'random selection'. In addition, the argument g for the adversary A is selected to be the PRF F_K configured with key K. On the other hand, when the game $EX^b_F(A)$ is configured with $b = 0$ the argument for A is a random function f, taken randomly from the whole set of functions $Func(l, L) : \{0,1\}^l \rightarrow \{0,1\}^L$, which is expressed as $f \xleftarrow{\$} Func(l, L)$. The value of $ADV^{PRF}_{F_K}(A)$ measures how well A is doing when distinguishing between f and F_K. Game EX^b_F can be represented graphically as in Fig. 10, where the oracle implements the function g, that can be F_K or f depending on the value of b. The adversary A interacts with g by performing q queries of l-bit values t_i, $\{t_0, t_1, \ldots, t_{q-1}\}$, and the oracle responds to the adversary queries with qL-bit output values $g(t_i)$. Finally A returns $\hat{b}$ meant as guess at b.

In order to obtain the expression for the $ADV^{PRF}_{F_K\_MOD}(A)$ we have defined a set of games shown in Table 1. In these games, as in game EX^b_F, the operator $\xleftarrow{\$}$ means a 'random selection', and $Func(l, L)$, $Func(l, l)$

and $Func(l, T)$ represent sets of functions such that $Func(l, L) : \{0, 1\}^l \rightarrow \{0, 1\}^L$, $Func(l, l) : \{0, 1\}^l \rightarrow \{0, 1\}^l$ and $Func(l, T) : \{0, 1\}^l \rightarrow \{0, \ldots, S-1\}$ with $T = log_2 S$.

In the same way as in game EX_F^b, games in Table 1 can measure the ability for an adversary A to distinguish between two functions depending on the configuration of the bits b and n. Moreover, in each game A performs q queries of values t_i, $\{t_0, t_1, \ldots, t_{q-1}\}$, to an oracle implementing a function g. Adversary A receives from the oracle q output values $g(t_i)$ and returns a result bit as in EX_F^b.

B. GAMES B AND D

Games $E_{F'}^n(A)$ and $E_F^n(A)$ are games B and D respectively in Table 1, they are only configured with the configuration bit n, and at the beginning of both a random key K is obtained, such that $K \xleftarrow{\$} \{0, 1\}^k$.

In game B, $E_{F'}^n(A)$, adversary A tries to distinguish between functions F_K_MOD and $Func\_MOD$. F_K_MOD is defined as in Section II, where $F_K\_MOD(x) = F_K(x) \, mod \, S$, and F_K is the PRF configured with the random key K. $Func\_MOD$ is a function obtained after applying modulo-S operation to the output of the function f, such that $Func\_MOD(x) = f(x) \, mod \, S$. Function f is selected randomly from the set of functions $Func(l, L)$.

In game D, $E_F^n(A)$, adversary A tries to distinguish between functions E_K_MOD and $Func'\_MOD$. E_K_MOD is defined as in Section II, where $E_K\_MOD(x) = \left(E_K(x) \, mod 2^L\right)$, and E_K is the PRF function that represents the block cipher of Fig. 2b configured with the random key K. $Func'\_MOD$ is a function obtained after applying a modulo-2^L operation to the output of a function f selected randomly from the set of functions $Func(l, l)$.

As summary, in $E_{F'}^n(A)$, adversary A tries to distinguish between a PRF and a random function with different input and output bit lengths, both followed by a modulo-S operation, while in $E_F^n(A)$ adversary A tries to distinguish between a PRF and a random function with the same input and output bit lengths, both followed by a modulo-2^L operation.

C. GAMES A AND C

Games $E_{F'}^{nb}(A)$ and $E_F^{nb}(A)$ are games A and C respectively, they are configured with the mode bit n and the configuration bit b, and at the beginning of both a random key K is obtained, such that $K \xleftarrow{\$} \{0, 1\}^k$.

In $E_{F'}^{nb}(A)$ adversary A will try to distinguish between a random function selected from the set of functions $Func(l, T)$ and a function r that depends on the value of the mode bit n. When $n = 1$, this function is configured as $r = F_K\_MOD$ as defined in Section II, while with $n = 0$, $r = Func\_MOD$ that is defined as in the previous subsection.

In $E_F^{nb}(A)$ adversary A will try to distinguish between a random function selected from the set of functions $Func(l, L)$ and a function r that depends on the value of the mode bit n. With $n = 1$, $r = E_K\_MOD$ as defined in Section II, and when $n = 0$, r is configured as $r = Func'\_MOD$ as defined in the previous subsection.

APPENDIX C
PROOF OF LEMMA 1

According to game A in Table 1 and the definition of the prf-advantage in (19), we can define the following prf-advantages of adversary A over F_K_MOD and $Func\_MOD$:

$$ADV^{PRF}_{F_K\_MOD}(A) = P\left[E^{11}_{F'}(A) = 1\right] - P\left[E^{10}_{F'}(A) = 1\right] \tag{20}$$

$$ADV^{PRF}_{Func\_MOD}(A) = P\left[E^{01}_{F'}(A) = 1\right] - P\left[E^{00}_{F'}(A) = 1\right] \tag{21}$$

where $P\left[E^{nb}_{F'}(A) = 1\right]$ is the probability that the result of the game $E^{nb}_{F'}(A)$ is $\hat{b} = 1$. In the case of (20) we are measuring the advantage of A over F_K_MOD as we are performing the game $E^{1b}_{F'}(A)$, and in this case A tries to distinguish between the PRF F_K_MOD and a random function taken from the set of functions $Func(l, T)$. Both F_K_MOD and $Func(l, T)$ map values between $\{0, 1\}^l$ and $\{0, \ldots, S-1\}$. In the case of (21) we are measuring the advantage of A over $Func\_MOD$ as we are performing game $E^{0b}_{F'}(A)$ where A tries to distinguish between $Func\_MOD$ and a random function taken from $Func(l, T)$.

According to Table 1, if $b = 0$ the input argument for adversary A is always the same in games $E^{00}_{F'}(A)$ and $E^{10}_{F'}(A)$, therefore $P\left[E^{00}_{F'}(A) = 1\right] = P\left[E^{10}_{F'}(A) = 1\right]$. Then it is possible to derive the following expression from (20) and (21):

$$\begin{aligned} &ADV^{PRF}_{F_K\_MOD}(A) - ADV^{PRF}_{Func\_MOD}(A) \\ &\quad = P\left[E^{11}_{F'}(A) = 1\right] - P\left[E^{01}_{F'}(A) = 1\right] \end{aligned} \tag{22}$$

As the inputs for adversary A in games $E^{n1}_{F'}(A)$ and $E^{n}_{F'}(A)$ are the same then $P\left[E^{n}_{F'}(A) = 1\right] = P\left[E^{n1}_{F'}(A) = 1\right]$, which means that:

$$\begin{aligned} &ADV^{PRF}_{F_K\_MOD}(A) - ADV^{PRF}_{Func\_MOD}(A) \\ &\quad = P\left[E^{1}_{F'}(A) = 1\right] - P\left[E^{0}_{F'}(A) = 1\right] \end{aligned} \tag{23}$$

In Fig. 11a, a graphical diagram for game $E^n_{F'}(A)$ is shown. Adversary A makes q queries to the oracle that implements the function F_K_MOD or $Func\_MOD$ depending on the value of configuration bit n. The diagram in Fig. 11a can be represented also as in Fig. 11b. As both schemes are equivalent from the point of view of adversary A we can conclude that it is possible to build an adversary C from A that produces thesame result bit as A. Moreover, as the role of adversary C in Fig. 11b is the same as the role of adversary A in Fig. 10 when performing game $EX^b_F(A)$, (23) can be rewritten as:

$$\begin{aligned} &ADV^{PRF}_{F_K\_MOD}(A) - ADV^{PRF}_{Func\_MOD}(A) \\ &= P\left[E^{1}_{F'}(A) = 1\right] - P\left[E^{0}_{F'}(A) = 1\right] \\ &= P\left[EX^{1}_{F}(C) = 1\right] - P\left[EX^{0}_{F}(C) = 1\right] \\ &= ADV^{PRF}_{F_K}(C) \end{aligned} \tag{24}$$

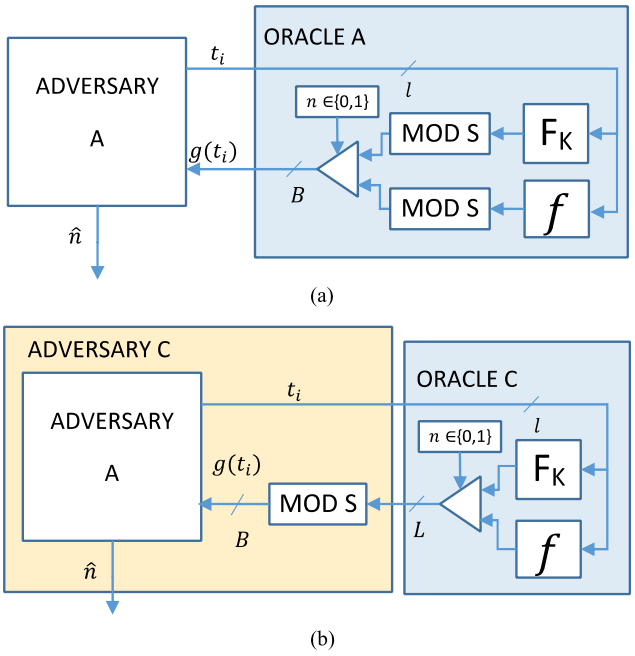

FIGURE 11. Diagram of (a) experiment $E^n_{F'}(A)$ and (b) its equivalent using the adversary C. In both cases adversary A generates the same amount of queries to the oracles. In each query A sends ti word represented with l bits and receives g(ti) symbol, represented with $B = \lceil \log_2 S \rceil$ bits. At the end, adversary A produces the result bit $\hat{n}$.

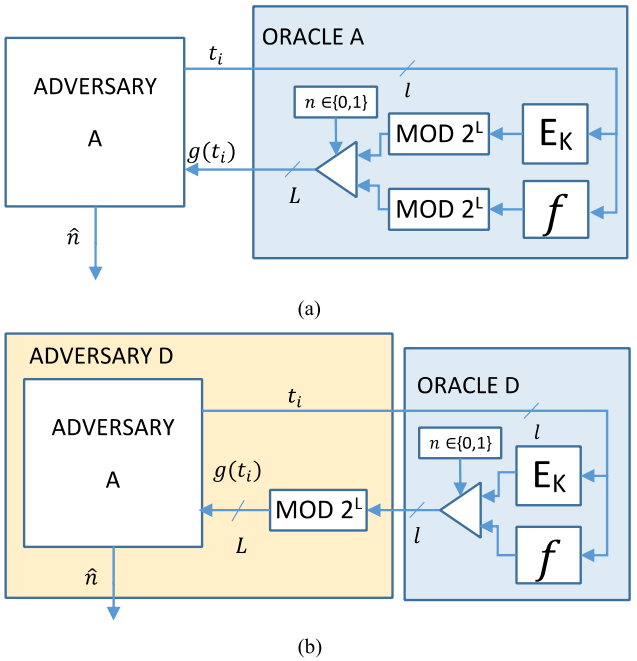

FIGURE 12. Diagram of (a) experiment $E^n_F(A)$ and (b) its equivalent using the adversary D. In both cases adversary A generates the same amount of queries to the oracles. In each query A sends t_i word and receives $g(t_i)$ symbol. At the end, adversary A tries to guess the configuration bit $\hat{n}$.

As defined in Section II, $F_K(x) = E_K\_MOD(x)$, which means that:

$$ADV^{PRF}_{F_K\_MOD}(A) = ADV^{PRF}_{E_K\_MOD}(C) + ADV^{PRF}_{Func\_MOD}(A) \tag{25}$$

On the other hand, thanks to game C in Table 1, it is possible to define the following advantages:

$$ADV^{PRF}_{E_K\_MOD}(A) = P\left[E^{11}_F(A) = 1\right] - P\left[E^{10}_F(A) = 1\right] \tag{26}$$

$$ADV^{PRF}_{Func'\_MOD}(A) = P\left[E^{01}_F(A) = 1\right] - P\left[E^{00}_F(A) = 1\right] \tag{27}$$

where $P\left[E^{nb}_F(A) = 1\right]$ is the probability that the result of the game $E^{nb}_F(A)$ is $\hat{b} = 1$. In the case of (26) we are measuring advantage of A over E_K_MOD, as we are performing game $E^{1b}_F(A)$, and in this case adversary A tries to distinguish between E_K_MOD and a random function taken from the set of function $Func(l, L)$. In the case of (27), as we are performing game $E^{0b}_F(A)$ the adversary A tries to distinguish between $Func'\_MOD$ and a random function taken from $Func(l, L)$, which is equivalent to measure the advantage over $Func'\_MOD$.

According to Table 1, if $b = 0$ the input argument for adversary A is always the same in games $E^{00}_F(A)$ and $E^{10}_F(A)$, therefore $P\left[E^{00}_F(A) = 1\right] = P\left[E^{10}_F(A) = 1\right]$. Then, as $E^{00}_F(A) = E^{10}_F(A)$ and $E^{n1}_F(A) = E^n_F(A)$ it is possible to derive (28) from (26) and (27).

$$\begin{aligned} &ADV^{PRF}_{E_K\_MOD}(A) - ADV^{PRF}_{Func'\_MOD}(A) \\ &\quad = P\left[E^{11}_F(A) = 1\right] - P\left[E^{01}_F(A) = 1\right] \\ &\quad = P\left[E^{1}_F(A) = 1\right] - P\left[E^{0}_F(A) = 1\right] \end{aligned} \tag{28}$$

In Fig. 12a, game $E^n_F(A)$ is shown graphically. As Fig. 12a and Fig. 12b are equivalent from the point of view of adversary A, it is possible to build an adversary D from A that produces the same return bit than A. In addition, the role of adversary D in Fig. 12b is the same as the role of adversary A in Fig. 10 when performing game $EX^b_F(A)$. Then equation (28) results as:

$$\begin{aligned} &ADV^{PRF}_{E_K\_MOD}(A) - ADV^{PRF}_{Func'\_MOD}(A) \\ &\quad = P\left[E^{1}_F(A) = 1\right] - P\left[E^{0}_F(A) = 1\right] \\ &\quad = P\left[EX^{1}_F(D) = 1\right] - P\left[EX^{0}_F(D) = 1\right] \\ &\quad = ADV^{PRF}_{E_K}(D) \end{aligned} \tag{29}$$

Therefore:

$$ADV^{PRF}_{E_K\_MOD}(A) = ADV^{PRF}_{E_K}(D) + ADV^{PRF}_{Func'\_MOD}(A) \tag{30}$$

According to (30), equation (25) can be written as:

$$\begin{aligned} ADV^{PRF}_{F_K\_MOD}(A) = ADV^{PRF}_{E_K}(D) + ADV^{PRF}_{Func'\_MOD}(A) \\ + ADV^{PRF}_{Func\_MOD}(A) \end{aligned} \tag{31}$$

which is the same expression as in equation (15) and then proves Lemma 1.

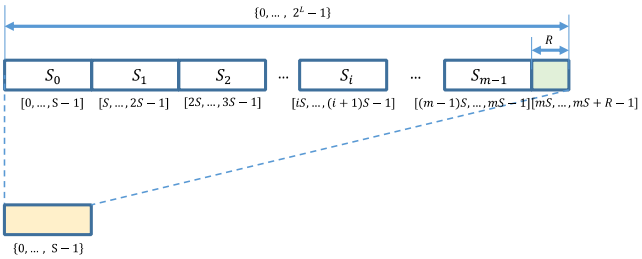

FIGURE 13. Domain $\{0, \ldots, 2^L - 1\}$ is mapped into interval $\{0, \ldots, S-1\}$. Each interval Si is mapped in the range $\{0, \ldots, S-1\}$. As 2^L is not a multiple of S the last green interval is called S_{last} and it contains only R values that are mapped to the range $\{0, \ldots, R-1\}$.

APPENDIX D
PROOF OF LEMMA 2

To prove the Lemma 2 it is necessary to obtain the term $ADV^{PRF}_{Func\_MOD}(A)$, that depends on game $E^{01}_{F'}(A)$ and $E^{00}_{F'}(A)$ as shown in (21). This term measures the capability of A for distinguishing the function $Func\_MOD(x) = f(x)\, mod S$, where f is a random function taken from the set of functions $Func(l, L)$, from a random function from the set $Func(l, T)$, where $T = log_2 S$ and S is not a power of two. The function f maps values from the domain $\{0, \ldots, 2^l - 1\}$ to $\{0, \ldots, 2^L - 1\}$. By applying modulo-S operation to the outputs of function f, values in the domain $\{0, \ldots, 2^L - 1\}$ are mapped to domain $\{0, \ldots, S-1\}$. Therefore function $Func\_MOD$ will map values from the domain $\{0, \ldots, 2^l - 1\}$ to $\{0, \ldots, S-1\}$, as random functions in the set $Func(l, T)$ do. In Fig. 13, the way of mapping values from $\{0, \ldots, 2^L - 1\}$ to $\{0, \ldots, S-1\}$ thanks to modulo-S operation is shown. As S is not a power of two, 2^L is not a multiple of S. Then the remainder R of the division between 2^L and S can be written as:

$$R = 2^L - m \cdot S \tag{32}$$

where m is the quotient of the division.

According to (32) the range $\{0, \ldots, 2^L - 1\}$ can be divided in m blocks with S values each one and one last with only R values, as shown in Fig. 13. The m blocks are called S_i, with $0 \le i \le m-1$ and the last one with R values is called S_{last}. After modulo-S operation, values in each range S_i will be equally distributed in destination range $\{0, \ldots, S-1\}$, however the last range S_{last} includes the last R values in the range $\{m \cdot S, \ldots, m \cdot S + R - 1\}$ and they only are mapped in the first R values of the destination range, $\{0, \ldots, R-1\}$. It will make that each value in the destination range $\{0, \ldots, R-1\}$ can have one more occurrence than values in the range $\{R, \ldots, S-1\}$. Therefore, if each input of modulo-S operation were uniformly distributed in the range $\{0, \ldots, 2^L - 1\}$ it will be possible to consider that its output will not be uniformly distributed, as R is not zero. It means that the output of modulo-S operation would have the following probability distribution:

$$P(x) = \begin{cases} \dfrac{m+1}{2^L} & for\ 0 \le x \le R-1 \\ \dfrac{m}{2^L} & for\ R \le x \le S-1 \end{cases} \tag{33}$$

where $P(x)$ is the probability of occurrence of value x at the output of the modulo-S operation.

During game $E^{0b}_{F'}(A)$, the adversary A makes q queries and they are not repeated. The queries are made to an oracle performing function g, that can be configured thanks to b value as $g = Func\_MOD$ or taken randomly from the set $Func(l, T)$.

In $E^{0b}_{F'}(A)$, $f \in Func(l, L)$ and it is a random function. According to [24] if the inputs of a random function f are not repeated we can consider that each output of f is randomly and uniformly distributed in its output range, which in this case is $\{0, \ldots, 2^L - 1\}$, independently of anything else. The same assumption can be done with the case of a random function from the set $Func(l, T)$ we can consider that each of its output is random and uniformly distributed in the range $\{0, \ldots, S-1\}$.

In the case of game $E^{00}_{F'}(A)$, as g is a random function taken from the set $Func(l, T)$, the final output from the oracle can be considered random and uniformly distributed in $\{0, \ldots, S-1\}$.

However, in game $E^{01}_{F'}(A)$, the oracle performs as function g the function $Func\_MOD(x) = f(x)\, mod S$. As f is a random function with a uniform distributed output in the range $\{0, \ldots, 2^L - 1\}$, the output of $Func\_MOD$ will have a probability distribution as in (33) owing to modulo-S operation.

Let us consider two situations when performing game $E^{01}_{F'}(A)$. The first of them is when during the q queries made by adversary A, at the output of $f \in Func(l, L)$ no value falls in the range S_{last}, but in any of the remainng S_i intervals. In this case the output of modulo-S could be considered randomly and uniformly distributed in the range $\{0, \ldots, S-1\}$, which is a situation indistinguishable from the game $E^{00}_{F'}(A)$, where adversary analyzes the outputs from $g \in Func(l, T)$. Let us name this situation 'Good interval' or *Gint*.

The second case is the opposite, when during the q queries performed in $E^{01}_{F'}(A)$, some output of f falls in the range S_{last}, then we can consider that this result could make us perceive the behavior of $Func\_MOD$ no to be like a random function taken from $Func(l, T)$. Due to that it is possible to consider that, at least, a good adversary could have a chance to know that we are running $E^{01}_{F'}(A)$ instead of $E^{00}_{F'}(A)$. Let's name this situation 'Bad interval' or *Bint*.

If we call $w1$ to the event $E^{01}_{F'}(A) = 1$ and $w0$ to the event $E^{00}_{F'}(A) = 1$ then:

$$\begin{aligned} \mathrm{P}[w1] = {} & \mathrm{P}[w1|Gint] \cdot \mathrm{P}[Gint] \\ & + \mathrm{P}[w1|Bint] \cdot \mathrm{P}[Bint] \end{aligned} \tag{34}$$

where $P[Gint]$ and $P[Bint]$ are the probabilities of *Gint* and *Bint* to occur during the q queries made by the adversary in game $E^{01}_{F'}(A)$, respectively.

As the adversary cannot distinguish the situation *Gint* during game $E^{01}_{F'}(A)$ from game $E^{00}_{F'}(A)$ because the output from the oracle in both situations can be considered random and uniformly distributed, then:

$$P\left[E^{00}_{F'}(A)=1\right]=P\left[E^{01}_{F'}(A)=1|Gint\right]\rightarrow P[w0]=P[w1|Gint] \quad (35)$$

In addition, by taking into account (34) and (35) we can rewrite (21) as:

$$\begin{aligned}ADV^{PRF}_{Func\_MOD}(A)&=P\left[E^{01}_{F'}(A)=1\right]-P\left[E^{00}_{F'}(A)=1\right]\\&=P[w1]-P[w0]=P[w1]-P[w1|Gint]=\\&P[w1|Gint]\cdot(P[Gint]-1)+P[w1|Bint]\cdot P[Bint]\end{aligned} \quad (36)$$

Moreover, as $P[Gint]=1-P[Bint]$ then (36) can be rewritten as:

$$ADV^{PRF}_{Func\_MOD}(A)=(\mathrm{P}[w1|Bint]-\mathrm{P}[w1|Gint])\cdot \mathrm{P}[Bint] \quad (37)$$

To get an upper bound for $ADV^{PRF}_{Func\_MOD}(A)$ we assume the worst case where a perfect adversary is able to always detect the function *Func_MOD* when running game $E^{01}_{F'}(A)$ and situation *Bint* is produced, which means that it always gives as result $E^{01}_{F'}(A)=1$. It means that with this perfect adversary $P[w1|Bint]=1$. Therefore (37) can be rewritten as:

$$ADV^{PRF}_{Func\_MOD}(A)=(1-\mathrm{P}[w1|Gint])\cdot \mathrm{P}[Bint]\leq \mathrm{P}[Bint] \quad (38)$$

The probability for an output value of the random function $f\in Func(l,L)$ of falling into the range S_{last}, is equal to $R/2^L$, as this range has R values among the 2^L possible. As q queries are performed by the adversary A during game $E^{01}_{F'}(A)$, the probability for any of the q oracle outputs of falling in S_{last} is $\mathrm{P}[Bint]=q\cdot R/2^L$. Therefore (38) can be expressed as:

$$ADV^{PRF}_{Func\_MOD}(A)\leq q\cdot R/2^L \quad (39)$$

By taking into account equation (32), as $I=L-T$, $T=log_2 S$ and $m=\lfloor 2^L/S\rfloor$, R can be expressed as:

$$\begin{aligned}R&=2^L-m\cdot S=\\&2^L-\left\lfloor\tfrac{2^L}{2^T}\right\rfloor\cdot 2^T=2^L\cdot\left(1-\left\lfloor\tfrac{2^L}{2^T}\right\rfloor\cdot\tfrac{2^T}{2^L}\right)=\\&2^L\cdot\left(1-\tfrac{\lfloor 2^I\rfloor}{2^I}\right)\end{aligned} \quad (40)$$

Therefore (39) can be rewritten as:

$$\begin{aligned}ADV^{PRF}_{Func\_MOD}(A)&\leq q\cdot R/2^L=q\cdot\left(1-\frac{\lfloor 2^I\rfloor}{2^I}\right)\\&\leq q\cdot\left(1-\frac{2^I-1}{2^I}\right)=\frac{q}{2^I}\end{aligned} \quad (41)$$

which corresponds with the equation (16) and finishes the proof of Lemma 2.

APPENDIX E
PROOF OF LEMMA 3

Regarding the term $ADV^{PRF}_{Func'\_MOD}(A)$, it is possible to make a similar reasoning to that for $ADV^{PRF}_{Func\_MOD}(A)$. In *Func'_MOD* the output of the random function $f\in Func(l,l)$ is subjected to a modulo-2^L operation. Thanks to this operation the output range of f, $\{0,\ldots,2^l-1\}$, is mapped to the space $\{0,\ldots,2^L-1\}$. This case differs from the situation of Lemma 2, when analyzing *Func_MOD*, where the space $\{0,\ldots,2^L-1\}$ is mapped to the range $\{0,\ldots,S-1\}$.

In the proof of Lemma 2, the output interval of $f\in Func(l,L)$ is $\{0,\ldots,2^L-1\}$. It is divided in m subintervals S_i with S values each one, and one last subinterval S_{last} with the R remaining values. Each subinterval S_i is mapped to $\{0,\ldots,S-1\}$ after modulo S operation while S_{last} is mapped to $\{0,\ldots,R-1\}$. This concludes in equation (39) obtaining $ADV^{PRF}_{Func\_MOD}(A)$ as a function of R, $ADV^{PRF}_{Func\_MOD}(A)\leq q\cdot R/2^L$.

In the case of *Func'_MOD* we could make the same reasoning as for Lemma 2, it is possible to divide the output range of $f\in Func(l,l)$, $\{0,\ldots,2^l-1\}$, into m intervals S_i with 2^L values each one and one last subinterval S_{last} with the R' remaining values, where R' is the remainder of the division between 2^l and 2^L. Then it is possible to derive an expression similar to (39) for $ADV^{PRF}_{Func'\_MOD}(A)$:

$$ADV^{PRF}_{Func'\_MOD}(A)\leq q\cdot R'/2^l \quad (42)$$

In Lemma 2, S is not a power of two and 2^L is not a multiple of S, then it was possible to deduce that $R=2^L-m\cdot S$, however in this case as 2^l is a multiple of 2^L the remainder of its division is zero, that is $R'=0$.

It means that $ADV^{PRF}_{Func'\_MOD}(A)=0$, which is the same expression as (17) and therefore it proves Lemma 3.

REFERENCES

[1] A. Lenk, P. Marcus, and I. Povoa, "GeoFPE: Format preserving encryption of geospatial data for the Internet of Things," in *Proc. IEEE Int. Congr. Internet Things (ICIOT)*, Jul. 2018, pp. 172–175.

[2] K. Kim and S.-S. Lee, "Encoding of Korean characters with less radix in format-preserving encryption," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2015, pp. 1075–1077.

[3] S. Liang, Y. Zhang, J. Guo, C. Dong, Z. Liu, and C. Jia, "Efficient format-preserving encryption mode for integer," in *Proc. 22017 IEEE Int. Conf. Comput. Sci. Eng. (CSE) IEEE Int. Conf. Embedded Ubiquitous Computing (EUC)*, Jul. 2017, pp. 96–102.

[4] *Guidelines for implementing and using the NBS Data Encryption Standad*, National Bureau of Standards USA. Standard FIPS PUB 74, 1981.

[5] M. Brightwell and H. Smith, "Using datatype-preserving encryption to enhance data warehouse security," in *Proc. 20th Nat. Inf. Syst. Secur. Conf.(NISSC)*, 1997, pp. 141—149.

[6] P. Rogaway, "A synopsis of format-preserving encryption," Voltage Secur., Cupertino, CA, USA, Mar. 2010. [Online]. Available: https://web.cs.ucdavis.edu/~rogaway/papers/synopsis.pdf

[7] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*. Gaithersburg, MD, USA: NIST, 2016.

[8] M. Bellare, P. Rogaway, and T. Spies. (Sep. 2010). *Addendum to' The FFX Mode of Operation for Format-Preserving Encryption*. [Online]. Available: https://csrc.nist.gov/projects/block-cipher-techniques/bcm

[9] E. Brier, T. Peyrin, and J. Stern. *BPS: A Format-Preserving Encryption Proposal*. Accessed: Jan. 1, 2020. [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/bps/bps-spec.pdf
[10] P. Chandrashekar, S. Dara, and V. N. Muralidhara, "Efficient format preserving encrypted databases," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2015, pp. 1–4.
[11] B. Cui, B. Zhang, and K. Wang, "A data masking scheme for sensitive big data based on format-preserving encryption," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE) IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, Jul. 2017, pp. 518–524.
[12] P. Wang, H. Luo, and J. Liu, "Format-preserving encryption for Excel," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, May 2016, pp. 1–2.
[13] R. Agbeyibor, J. Butts, M. Grimaila, and R. Mills, "Evaluation of format-preserving encryption algorithms for critical infrastructure protection," in *Critical Infrastructure Protection VIII*. Heidelberg, Germany: Springer, 2014, pp. 245–261.
[14] I. Oh, T. Kim, K. Yim, and S.-Y. Lee, "A novel message-preserving scheme with format-preserving encryption for connected cars in multi-access edge computing," *Sensors*, vol. 19, no. 18, p. 3869, Sep. 2019.
[15] H. Sun, H. Luo, and Y. Sun, "Data hiding for ensuring the quality of the host image and the security of the message," *IEEE Access*, vol. 7, pp. 64767–64777, 2019.
[16] K. Kim and K.-Y. Chang, "Performance analysis of format-preserving encryption based on unbalanced-feistel structure," in *Advances in Computer Science and Ubiquitous Computing* (Lecture Notes in Electrical Engineering), vol. 373. Springer: Singapore, 2015, pp. 425–430.
[17] K. Mallaiah, S. Ramachandram, and S. Gorantala, "Performance analysis of Format Preserving Encryption (FIPS PUBS 74-8) over block ciphers for numeric data," in *Proc. 4th Int. Conf. Comput. Commun. Technol.(ICCCT)*, Sep. 2013, pp. 193–198.
[18] M. Li, Z. Liu, J. Li, and C. Jia, "Format-preserving encryption for character data," *J. Netw.*, vol. 7, no. 8, pp. 1239–1244, 2012.
[19] T. W. Arnold, M. Check, E. A. Dames, J. Dayka, S. Dragone, D. Evans, W. S. Fernandez, M. D. Hocker, R. Kisley, T. E. Morris, J. Petreshock, and K. Werner, "The next generation of highly reliable and secure encryption for the IBM z13," *IBM J. Res. Dev.*, vol. 59, no. 4/5, pp. 6:1–6:13, Jul. 2015.
[20] A. Perez-Resa, M. Garcia-Bosque, C. Sanchez-Azqueta, and S. Celma, "Physical layer encryption for industrial ethernet in gigabit optical links," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3287–3295, Apr. 2019.
[21] A. Perez-Resa, M. Garcia-Bosque, C. Sanchez-Azqueta, and S. Celma, "Chaotic encryption applied to optical ethernet in industrial control systems," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 12, pp. 4876–4886, Dec. 2019.
[22] P. Rogaway, "Evaluation of some blockcipher modes of operation," Cryptogr. Res. Eval. Committees, Government Japan, Tokyo, Japan, 2011. [Online]. Available: https://crossbowerbt.github.io/docs/crypto/rogaway_modes.pdf
[23] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in *Proc. 38th Annu. Symp. Found. Comput. Sci.*, Oct. 1997, pp. 394—403.
[24] M. Bellware and P. Rogaway. (May 2005). (Introduction to Modern Cryptography). [Online]. Available: http://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf
[25] M. Bellware and P. Rogaway, "Indistinguishably under chosen-plaintext attack," in *Introduction to Modern Cryptography*. May 2005, ch. 5–4. [Online]. Available: https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf
[26] M. P. Fok, Z. Wang, Y. Deng, and P. R. Prucnal, "Optical layer security in fiber-optic networks," *IEEE Trans. Inf. Forensic Security*, vol. 6, no. 3, pp. 725–736, Sep. 2011.
[27] J. Ji, G. Zhang, W. Li, L. Sun, K. Wang, and M. Xu, "Performance analysis of physical-layer security in an OCDMA-based wiretap channel," *J. Opt. Commun. Netw.*, vol. 9, no. 10, p. 813, Oct. 2017.
[28] J. Hizanidis, S. Deligiannidis, A. Bogris, and D. Syvridis, "Enhancement of chaos encryption potential by combining all-optical and electrooptical chaos generators," *IEEE J. Quantum Electron.*, vol. 46, no. 11, pp. 1642–1649, Nov. 2010.
[29] A. Sultan, X. Yang, A. A. E. Hajomer, S. B. Hussain, and W. Hu, "Dynamic QAM mapping for physical-layer security using digital chaos," *IEEE Access*, vol. 6, pp. 47199–47205, 2018.
[30] Y. Gao, S. Hu, W. Tang, Y. Li, Y. Sun, D. Huang, S. Cheng, and X. Li, "Physical layer security in 5g based large scale social networks: Opportunities and challenges," *IEEE Access*, vol. 6, pp. 26350–26357, 2018.
[31] H. Rahbari and M. Krunz, "Full frame encryption and modulation obfuscation using channel-independent preamble identifier," *IEEE Trans.Inf. Forensic Security*, vol. 11, no. 12, pp. 2732–2747, Dec. 2016.
[32] K. Guan, J. Kakande, and J. Cho, "On deploying encryption solutions to provide secure transport-as-a-service (TaaS) in core and metro networks," in *Proc. 42nd Eur. Conf. Opt. Commun.*, Düsseldorf, Germany, Sep. 2016, pp. 1–3.
[33] J. Daemen and V. Rijmen, *The Design of Rijndael, AES—The Advanced Encryption Standard*. Berlin, Germany: Springer-Verlag, 2002.
[34] J. T. Butler and T. Sasao, "Fast hardware computation of x mod z," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops Phd Forum*, Shanghai, China, May 2011, pp. 294–297.
[35] A. Klein, *Stream Ciphers*, London, U.K.: Springer-Verlag, 2013.

ADRIÁN PÉREZ-RESA was born in San Sebastián, Spain. He received the M.Sc. degree in telecommunications engineering from the University of Zaragoza, Zaragoza, Spain, in 2005. He is currently pursuing the Ph.D. degree with the Group of Electronic Design, Aragón Institute of Engineering Research, University of Zaragoza.

He was an Research and Development Engineer with Telecommunications industry for more than ten years. He is a member of the Group of Electronic Design, Aragón Institute of Engineering Research, University of Zaragoza. His research interests include high-speed communications and cryptography applications.

MIGUEL GARCIA-BOSQUE was born in Zaragoza, Spain. He received the B.Sc. and M.Sc. degrees in physics from the University of Zaragoza, Zaragoza, in 2014 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Group of Electronic Design, Aragón Institute of Engineering Research, University of Zaragoza.

He is a member of the Group of Electronic Design, Aragón Institute of Engineering Research, University of Zaragoza. His research interests include chaos theory and cryptography algorithms.

CARLOS SÁNCHEZ-AZQUETA was born in Zaragoza, Spain. He received the B.Sc. degree in physics from the University of Zaragoza, Zaragoza, in 2006, the Dipl.-Ing. degree in electronic engineering from the Complutense University of Madrid, Madrid, Spain, in 2009, and the M.Sc. and Ph.D. degrees in physics from the University of Zaragoza, in 2010 and 2012, respectively.

He is currently a member of the Group of Electronic Design, Aragón Institute of Engineering Research, University of Zaragoza. His research interests include mixed-signal integrated circuits, high-frequency analog communications, and cryptography applications.

SANTIAGO CELMA was born in Zaragoza, Spain. He received the B.Sc., M.Sc., and Ph.D. degrees in physics from the University of Zaragoza, Zaragoza, in 1987, 1989, and 1993, respectively.

He is currently a Full Professor with the Group of Electronic Design, Aragón Institute of Engineering Research, University of Zaragoza. He has coauthored more than 100 technical articles and 300 international conference contributions. He is also a coauthor of four technical books. He holds four patents. He appears as the Principal Investigator in more than 30 national and international research projects. His research interests include circuit theory, mixed-signal integrated circuits, high-frequency communication circuits, wireless sensor networks, and cryptography for secure communications.