# Security proof of the canonical form of self-synchronizing stream ciphers

Brandon Dravie, Philippe Guillot, Gilles Millérioux

# Security Proof of the Canonical Form of Self-Synchronizing Stream Ciphers

Brandon Dravie[1], Philippe Guillot[2] and Gilles Millérioux[1]

[1] Université de Lorraine
Centre de Recherche en Automatique de Nancy (CRAN CNRS UMR 7039), France,
{brandon.dravie,gilles.millerioux}@univ-lorraine.fr
[2] Université Paris 8
Laboratoire Analyse, Géométrie et Applications (LAGA UMR 7539), France
philippe.guillot@univ-paris8.fr

**Abstract.** This paper studies the security level expected by the canonical form of the Self-Synchronizing Stream Cipher (SSSC). A SSSC can be viewed as the combination of a shift register together with a filtering function. The maximum security of such a cipher is reached when the filtering function is random. However, in practice, Pseudo Random Functions (PRF) are used as filtering functions. In this case, we show that the security against chosen ciphertext attacks (IND-CCA security) cannot be reached for the canonical form of the SSSC, but it is however secure against chosen plaintext attacks (IND-CPA secure). Then, a weaker property than pseudo-randomness is introduced in order to characterize the security of the canonical SSSC from its filtering function. A connection with the *left-or-right indistinguishability* (LOR-IND) is made. This property provides a necessary and sufficient condition to characterize the indistinguishablity of SSSC.

## 1 Introduction

Self-Synchronizing Stream Ciphers (SSSC) was patented in 1946. The basic principle of such ciphers is to encrypt every plaintext symbol with a transformation that only involves a fixed number of previous ciphertexts symbols. Therefore, every ciphertext symbol is correctly decrypted provided that previous symbols have been properly received. This self-synchronisation property has many advantages and is especially relevant to group communications. In this respect, since 1960, specific SSSC have been designed and are still used to provide bulk encryption (for hertzian line, RNIS link, ...) in military applications or governmental radio mobile network.

Regarding security, SSSC have intrinsic interesting properties due to the specific architecture. For example, as each plaintext symbol influences potentially all subsequent ciphertexts, they naturally have good diffusion properties and are efficient against attacks based on plaintext redundancy. Furthermore, they can prevent from traffic analysis as the information which is conveyed through the channel is encrypted whether there is traffic or not. In the early 90s, studies have been

performed [Mau91,DGV92] to propose secure design of SSSC. These works have been followed by effective constructions ([DGV92,Sar03,DK08]), but till now, all of these SSSC schemes have been broken([JM03,JM05,JM06,KRB+08,Klí05]), what may make believe that a secure SSSC is impossible to design. In this paper, we derive a result on the security of the canonical form of the Self-Synchronizing Stream Cipher . Let us recall that the canonical form of the Self-Synchronizing Stream Cipher (SSSC) is constituted by the combination of a shift register, which acts as a state register with the ciphertext as input, together with a filtering function that provides the running key stream. It had been shown in [BDM15] that the canonical form of the Self-Synchronizing Stream Cipher is not resistant against chosen ciphertext attack (IND-CCA security) but can reach the resistance against chosen plaintext attack (IND-CPA security) provided that the filtering function is pseudo random. In this paper, a weaker property than pseudo-randomness is introduced in order to characterize the security of the canonical SSSC from its filtering function. A connection with the *left-or-right indistinguishability* (LOR-IND) is made. This property provides a necessary and sufficient condition to characterize the indistinguishablity of SSSC. The technical developments used to establish the security proof follow similar lines than those used when dealing with block cipher symmetric encryption scheme.

The paper is organized as follows. In Section 2, we recall the characteristic of the canonical form of SSSC. Sections 3 present various criteria on the filtering function to guarantee the security of the SSSC. The notions of indistinguishability and security games, used to assess the security of the SSSC, are detailed in Section 4. The results on the security are established in Section 5. Finally, we end up with concluding remarks in Section 6.

## 2 Canonical form of the Self-Synchronization Stream Cipher

### 2.1 Generalities on SSSC

A conventional way for designing an SSSC is to resort to a shift-register-like architecture, giving the so-called canonical representation of the SSSC [MvOV96]. This canonical representation is depicted in Figure 1. It has also been studied in [Par12] and admits at the cipher and decipher sides the respective equations:

$$\text{cipher: } \begin{cases} z_t = h_\kappa(c_{t-1}, \ldots, c_{t-n}) \\ c_t = z_t \oplus m_t \end{cases} \qquad \text{decipher: } \begin{cases} \widehat{z}_t = h_\kappa(c_{t-1}, \ldots, c_{t-n}) \\ \widehat{m}_t = \widehat{z}_t \oplus c_t \end{cases} \tag{1}$$

where $n$ is the dimension of the shift register, $h_\kappa : \{0,1\}^n \to \{0,1\}$ denotes the filtering function parametrized by the secret key $\kappa$ and for all instant $t$, $m_t \in \{0,1\}$ is the plaintext symbol, $c_t \in \{0,1\}$ is the ciphertext symbol, $z_t \in \{0,1\}$ is the key stream symbol. The quantity $\widehat{z}_t \in \{0,1\}$ is the key stream symbol on the deciphering side and $\widehat{m}_t \in \{0,1\}$ is the recovered plaintext symbol. NB: it is

worth pointing out that all the results are established here in the Boolean case but still hold when considering any other finite alphabet.

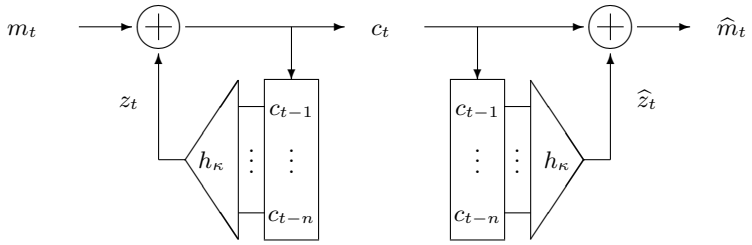


**Fig. 1.** Canonical form of a SSSC. Left: the ciphering. Right: the deciphering.

The secret key $\kappa$ is some suitable parameters that select the function $h$ among a family of filtering functions. The dimension of the shift register is given by the integer $n$. The key stream symbol $z_t$ is the output of the filtering function. It only depends on the secret key $\kappa$ shared by the cipher and the decipher and on $n$ past values of the ciphertext. The ciphertext $c_t$ is worked out from an exclusive or of the plaintext symbol $m_t$ and of the key stream symbol $z_t$. It is conveyed through the public channel. Since the generator function $h$ shares, at the transmitter and receiver sides, the same values, namely the $n$ past ciphertexts, then the receiver recovers the plaintext when he properly received the last $n$ ciphertext symbols. Indeed $\widehat{m}_t = m_t$ whenever $\widehat{z}_t = z_t$. Finally, the vector $x_t = (c_{t-1}, \ldots, c_{t-n})$ stands for the internal state. Then, it is clear that the generators synchronize automatically after a finite transient time of length $n$. As a result, the dimension $n$ of the shift register defines the synchronization delay of the decipher.

The SSSC model defined by equations (1) and shown on Figure 1 remains a conceptual model that can be implemented by different architectures resulting from different design approaches. An overview has been given in [Dae95] and more recently in the survey paper [MG10]. The so called Cipher Feedback (CFB) mode of operation consists in building a closed-loop architecture involving both a shift register and a block cipher primitive. The mostly adopted method to implement a binary SSSC is to resort to a block cipher (AES for instance) in 1-bit CFB mode [oS80]. Nevertheless, this mode is quite inefficient in terms of encryption speed since one block cipher operation is required for encrypting a single plaintext bit. The state transition function is described by the shift register. It is very simple and is secret key independent. Therefore, the security relies entirely on the security of the filtering function. Maurer's approach is suggested in [Mau91] as an alternative. The most important concept is the use of several finite state automata in parallel and serial combinations. However, it is shown in [Dae95] that this method is not really relevant. In order to guarantee the self-synchronization in finite-time, the state transitions functions proposed in [Dae95] are the $T$–functions ($T$ for Triangle), which are functions that prop-

agate dependencies in one direction only. More general architectures having the self-synchronization property have been proposed in [MG10][PGM11]. The constructions were based on advanced control theoretical concepts such as flatness.

### 2.2 Encryption/decryption mechanisms of SSSC

In order to assess the security level reached by the canonical SSSC, it is necessary to formally define the encryption and the decryption mechanisms.

*Setup* A random secret key $\kappa$ is randomly chosen in the key space for both the cipher and the decipher.

*Encryption* For encrypting a message $m$ consisting in $\ell$ binary symbols, the steps are the following:

1. Choose randomly an $n$-dimensional binary vector as the initial state $x_0$ of the shift register.
2. Choose randomly $n$ binary symbols to build the synchronization sequence and concatenate it with the message to be encrypted yielding a binary sequence $m_0, \ldots, m_{n+\ell-1}$. The first $n$ symbols are those of the synchronization sequence, and the other ones are those of the message to be encrypted.
3. For each symbol at instant $t$,
   (a) Compute the key stream symbol as $z_t = h_\kappa(x_t)$.
   (b) Compute the ciphertext symbol as $c_t = m_t \oplus z_t$.
   (c) Update the shift register state by shifting its components and feeding it by the computed ciphertext symbol $c_t$ to obtain the next state $x_{t+1}$.

*Decryption* For decrypting the cryptogram consisting in $n + \ell$ binary symbols, the steps are the following:

1. Initialize the shift register state to any arbitrary value, for example the $n$-dimensional zero vector $\widehat{x}_0 = 0$.
2. For each symbol at instant $t$,
   (a) Compute the key stream symbol as $\widehat{z}_t = h_\kappa(\widehat{x}_t)$.
   (b) Compute the plaintext symbol as $\widehat{m}_t = c_t \oplus \widehat{z}_t$.
   (c) Update the shift register state by shifting its components and feeding it by the computed ciphertext symbol $c_t$ to obtain the next state $\hat{x}_{t+1}$.
3. The first $n$ decrypted symbols correspond to the synchronization sequence and are ignored. The decrypted message consists of the $\ell$ last decrypted symbols.

## 3 Security criteria on the filtering function

In this section, we first recall what a *pseudo random function* is, and then we define a new security criterion on the filtering function that characterizes the security of the Self-Synchronizing Stream Cipher.

### 3.1 Pseudo Random Functions

A Pseudo Random Function is an element randomly chosen in a family of Pseudo Random Functions. A family of functions is said to be Pseudo Random if it is computationally indistinguishable from the set of all the functions. This means that for polynomial complexity adversaries, a Pseudo Random Function behaves as if it was a true randomly chosen function.

The Pseudo Random property is assessed by the so called PRF-game that involves an adversary $\mathcal{B}$, challenged to guess whether a given function is either a true random function, randomly chosen in the set of all the functions, or is an element of the family. The entries of the algorithm $\mathcal{B}$ are an integer $n$ together with an oracle that, on request for an $n$-dimensional vector, returns the value of the function.

*The PRF-game.* The element of the game for the parameter $n$ is a family of functions $f_\kappa : \{0,1\}^n \to \{0,1\}$. The steps of the game are the followings:

1. The oracle chooses a random bit $b \in \{\mathrm{rf}, \mathrm{prf}\}$. If $b = \mathrm{prf}$ (pseudo random world) then it randomly chooses a function $f = f_\kappa$ in the family of pseudo random functions. If $b = \mathrm{rf}$ (random world), it randomly chooses a Boolean function $f$ in the set of all $2^{2^n}$ Boolean functions $\{0,1\}^n \to \{0,1\}$.
2. The challenger asks the oracle for the values of $f(x_i)$ for inputs $x_1, \ldots, x_q$. The number $q$ of queries the challenger is allowed to perform is bounded by a polynomial in $n$.
3. After the $q$ queries, the challenger must answer a binary value $\widehat{b}$ that means that the challenger guesses that the chosen function was pseudo random ($\widehat{b} = \mathrm{prf}$) or was purely random ($\widehat{b} = \mathrm{rf}$). If $\widehat{b} = b$ then the challenger wins.

The adversary $\mathcal{B}$ does not know in which world it plays: pseudo random or random world? Let $\Pr_{\mathrm{prf}}(\mathcal{B} = \mathrm{prf})$ be the probability that $\mathcal{B}$ answers prf given it plays in the pseudo random world and let $\Pr_{\mathrm{rf}}(\mathcal{B} = \mathrm{prf})$ be the probability that $\mathcal{B}$ answers prf given it plays in the random world. In the first case, the answer of $\mathcal{B}$ is correct, and in the second case, it is wrong. The advantage of $\mathcal{B}$ in the PRF-game is by definition:

$$\mathrm{Adv}_{\mathrm{prf}}(\mathcal{B}) = \left|\Pr_{\mathrm{prf}}(\mathcal{B} = \mathrm{prf}) - \Pr_{\mathrm{rf}}(\mathcal{B} = \mathrm{prf})\right|$$

A family of Boolean functions is said to be pseudo random (PRF) if the maximum advantage of any adversary is negligible.

*N.B.* For a family of functions to be PRF, it is necessary that its number of elements increases faster than any polynomial in the security parameter $n$. If not, the exhaustive search algorithm has polynomial complexity. In cryptographic context, the key secret selects an element of the family. So, the key size must be greater than the logarithm of any power of $n$. In practice, an $n$-bit long key is admissible.

### 3.2 Weak Pseudo Random Functions

In this section, we define a new family of functions with a weaker property than pseudo random functions defined in the previous section and that is more suitable to assess the security of Self Synchronizing Stream Ciphers.

A *Weak Pseudo Random Function*, wprf for short, is an element of a family of weak pseudo random functions. Such a family is computationally indistinguishable from the family of all function. The wprf property is defined by the so called wprf-game defined further.

Given a prf-family, the wprf-game challenger is challenged to guess whether a given function presented to it is either an element of the family or a random function. In order to answer the challenge, the challenger can ask for help from an oracle. The difference with prf-families lies in the form of the requests to the oracle.

*The WPRF-game.* The element of the game for the parameter $n$ is a family of functions $f_\kappa : \{0,1\}^n \to \{0,1\}$. The steps of the game are the following.

1. The oracle chooses a random bit $b \in \{\mathrm{rf}, \mathrm{wprf}\}$. If $b = \mathrm{rf}$, then it randomly chooses a function in the set of all $2^{2^n}$ functions $\{0,1\}^n \to \{0,1\}$. Otherwise, if $b = \mathrm{wprf}$, then it chooses a random parameter $\kappa$ that defines a random element $f = f_\kappa$ of the WPRF-family. The oracle keeps the chosen bit secret and the goal of the challenger is to guess it.
2. The challenger asks the oracle for responses to requests. The input of the request is a polynomial length string of bits $m_1 \cdots m_k$. The response of the oracle is the sequences of couples $\big(x_i, f(x_i)\big)_{i\in\{0,\ldots k\}}$, where:

$$\begin{aligned} &x_0 \text{ is chosen at random by the oracle} \\ &\text{for } i = 1 \text{ to } k,\ x_i = (x_{i-1} \ll 1) + \big(m_i \oplus f(x_i)\big), \end{aligned}$$

   and $x \ll 1$ denotes the left shift of the binary vector $x$ by one.
3. After a polynomial number $q$ of queries, the challenger answers a binary value $\widehat{b} \in \{\mathrm{rf}, \mathrm{wprf}\}$ which means that the function $f$ chosen by the oracle is either random or pseudo random.

*Remark 1.* The sequence of couples transmitted by the oracle at the step 2 corresponds exactly to the parameters and the values of the canonical SSSC filtering function during the encryption process of the message $m$. Given this sequence, the cryptogram is deduced by:

$$c_i = m_i \oplus f(x_i).$$

Conversely, knowing the cryptogram of a given message, it is possible to infer the sequence transmitted by the oracle. The parameter $x_i$ consists in the vector whose components are the latter cryptogram symbols and the value is given by:

$$f(x_i) = c_i \oplus m_i.$$

*Remark 2.* As the initial value is randomly chosen by the oracle, the challenger cannot predict which parameters $x_i$ will be returned by the oracle.

The advantage of a challenger $\mathcal{B}$ in a wprf game is defined in the same manner as in the prf game:

$$\mathrm{Adv}_{\mathrm{wprf}}(\mathcal{B}) = \Big|\mathrm{Pr}_{\mathrm{wprf}}(\mathcal{B} = \mathrm{wprf}) - \mathrm{Pr}_{\mathrm{rf}}(\mathcal{B} = \mathrm{prf})\Big|.$$

### 3.3 Pseudo random functions are weak pseudo random functions

**Proposition 1.** *If a family of functions is pseudo random, then it is weak pseudo random.*

*Proof.* Given a family of functions $\mathcal{F}$, one has to prove that if a challenger against the weak pseudo random property of $\mathcal{F}$ exists, then a challenger against the pseudo random property can be derived.

Let $\mathcal{B}$ a challenger against the weak pseudo random property. We construct a challenger $\mathcal{A}$ against the pseudo random property, that will act for $\mathcal{B}$ as a weak pseudo random oracle. The precise definition of $\mathcal{A}$ is:

1. On request of a length $k$ sequence $m_1 \cdots m_k$ from $\mathcal{B}$, the challenger $\mathcal{A}$ chooses a random value $x_0 \in \{0,1\}^n$, computes $x_1, \ldots, x_k$ as $x_i = (x_{i-1} \ll 1) + m_i$, asks the oracle for the values $f(x_i)$ and transmits them to $\mathcal{B}$.
2. After a polynomial number $q$ of such queries, the challenger $\mathcal{B}$ answers. If $\mathcal{B}$ answers rf, then $\mathcal{A}$ answers rf, and if $\mathcal{B}$ answers wprf, then $\mathcal{A}$ answers prf.

We now prove that the challenger $\mathcal{A}$ so constructed has the same advantage as the challenger $\mathcal{B}$. This follows from the fact that the challenger $\mathcal{A}$ gives the same answer as $\mathcal{B}$. Thus $\mathcal{A}$ wins at the same time as $\mathcal{B}$. And yet as $\mathcal{A}$ acts for $\mathcal{B}$ as a wprf oracle, one has $\mathrm{Pr}_{\mathrm{prf}}(\mathcal{A} = \mathrm{prf}) = \mathrm{Pr}_{\mathrm{wprf}}(\mathcal{B} = \mathrm{wprf})$ and $\mathrm{Pr}_{\mathrm{rf}}(\mathcal{A} = \mathrm{rf}) = \mathrm{Pr}_{\mathrm{rf}}(\mathcal{B} = \mathrm{rf})$. It follows that $\mathrm{Adv}_{\mathrm{prf}}(\mathcal{A}) = \mathrm{Adv}_{\mathrm{wprf}}(\mathcal{B})$, and then, if $\mathcal{B}$ has a non negligible wprf-advantage, then $\mathcal{A}$ has a non negligible prf-advantage too.

## 4 Indistinguishability and security games

The framework used here to assess the security of SSSC is the one defined in [BR05] for symmetric encryption schemes. The notions of security games is used within this framework. They are based on security concepts introduced by Goldwasser and Micali [GM82,GM84]. The first concept is the semantic security, which is a computational analogue to the Shannon perfect secrecy. It means that an adversary cannot recover any bit of information of the plaintext from the ciphertext. However, the semantic security does not allow to fluently deal with security proof. This leads the authors in [GM82,GM84] to define another concept known as indistinguishability which better facilitates proving the security of practical ciphers.

In order to characterize the security of the SSSC, we need a variant of the indistinguishability property known as the *left-or-right indistinguishability* which imply the indistinguishability property against an adaptive chosen plaintext attack.

### 4.1 Indistinguishability and IND-game

A cryptographic scheme is said to be secure, in the sense of indistinguishability, if a polynomial complexity algorithm has a negligible advantage in distinguishing from which of two messages $m_0$ and $m_1$ it has provided, comes the cryptogram presented to it. This is formalized by the so called IND-game. This game involves two players: an algorithm $\mathcal{A}$ called adversary or challenger, and an oracle. The game consists of the following steps:

1. The challenger chooses two messages $m_0$ and $m_1$ and provides the oracle with them.
2. The oracle randomly chooses a binary digit $b \in \{0, 1\}$, encrypts the message $m_b$ and returns the cryptogram to the challenger.
3. The challenger is challenged to guess the value of $b$, that is which of the two messages has been encrypted. The answer of the challenger is a binary value $\widehat{b}$. The challenger wins if $b = \widehat{b}$.

Before giving its answer, the challenger can be supported by the oracle to decrypt ciphertexts the challenger chooses (Chosen Ciphertext Attack, CCA) or to encrypt plaintexts the challenger chooses (Chosen plaintext Attack, CPA). Of course, in a CCA attack, the challenger is not allowed to request the decryption of the challenge it has received.

For any $b$ and $\widehat{b}$ in the set $\{0, 1\}$, let $\Pr_{m_b}(\mathcal{A} = \widehat{b})$ be the probability that the challenger $\mathcal{A}$ answers $\widehat{b}$ given the ciphertext which corresponds to the encryption of the message $m_b$. The advantage of $\mathcal{A}$ in this IND-game is by definition:

$$\mathrm{Adv}(\mathcal{A}) = \Big|\Pr_{m_0}(\mathcal{A} = 0) - \Pr_{m_1}(\mathcal{A} = 0)\Big|.$$

As $m_0$ and $m_1$ are randomly chosen with the same probability $1/2$, it holds that

$$\mathrm{Adv}(\mathcal{A}) = \Big|2\Pr(b = \widehat{b}) - 1\Big|,$$

where $\Pr(b = \widehat{b})$ denotes the probability that $\mathcal{A}$ wins.

If the challenger answers at random, then its advantage is null. If it always wins, or always looses, then its advantage equals 1.

A cryptographic scheme is defined by a security parameter, which is for example the key size. Such a scheme is said to be IND-secure if the advantage of the challenger is negligible. A function of the real number $x$ is said to be negligible if it decreases faster that the inverse of any polynomial in $x$ as $x$ grows to infinity. A cryptographic algorithm is said to be IND-CCA if it is IND-secure during a CCA attack. It is said to be IND-CPA if it is IND-secure during a CPA attack.

For the canonical SSSC, the security parameters are the size of the shift register which equals the number of inputs of the filtering function and the size of the secret key.

### 4.2 Left-or-right indistinguishability and LOR-IND-game

Likewise the IND-game, the LOR-IND-game, namely *left-or-right indistinguishability* is described in [BDJR97]. It involves two algorithms: the challenger $\mathcal{A}$ and an oracle. The game consists in the following steps:

1. First, the oracle randomly chooses a bit $b \in \{0, 1\}$.
2. The challenger sends to the oracle a polynomial number of queries in the form of couples of messages $(m_0^i, m_1^i)$. The oracle encrypts either the message $m_0^i$ or the message $m_1^i$ depending on the bit $b$ chosen during the step 1. It returns to the challenger the cryptogram $c^i$ of the message $m_b^i$.
3. After a polynomial number of queries, the challenger is challenged to guess the value of the bit $b$ chosen by the oracle. The response of the challenger is a binary value $\widehat{b} \in \{0, 1\}$. The challenger wins if $b = \widehat{b}$.

Likewise the IND-game, the advantage of a challenger $\mathcal{A}$ in the LOR-IND-game is defined by:

$$\begin{aligned} \mathrm{Adv}(\mathcal{A}) &= \left|\Pr_{b=0}(\mathcal{A} = 0) - \Pr_{b=1}(\mathcal{A} = 0)\right| \\ &= \left|2\Pr(b = \widehat{b}) - 1\right| \end{aligned}$$

A ciphering scheme is said to have the LOR-IND property if any polynomial adversary has a negligible advantage in the LOR-IND-game.

The following proposition states that the left-or-right indistinguishability introduced above is a stronger property that the indistinguishability against a chosen plaintext attack.

**Proposition 2.** *If a ciphering scheme reaches the LOR-IND-CPA property, then it reaches the IND-CPA property.*

*Proof.* By contraposition, assume that a challenger $\mathcal{A}$ against the IND-CPA property exists. A challenger $\mathcal{B}$ against the LOR-IND property is constructed from $\mathcal{A}$:

1. The challenger $\mathcal{A}$ chooses a first couple of messages $(m_0^0, m_1^0)$ that it sends to $\mathcal{B}$, and $\mathcal{B}$ sends directly theses messages to the LOR-IND oracle.
2. When $\mathcal{A}$ requires for the cryptograms of a chosen plaintext message $m^i$, the algorithm $\mathcal{B}$ sends to the LOR-IND-oracle the couple with twice the same message : $(m^i, m^i)$. The cryptogram returned by the oracle thus corresponds to $m^i$ whatever the random bit $b$ chosen by the oracle. This cryptogram is returned to $\mathcal{A}$.
3. After a polynomial number of queries, the challenger $\mathcal{A}$ returns a guess $\widehat{b}$ for $b$, and $\mathcal{B}$ returns the same value.

The algorithm $\mathcal{B}$ so defined simulates for $\mathcal{A}$ an IND-CPA-oracle, and the advantage of $B$ equals the advantage of $\mathcal{A}$. Thus, if an adversary against the IND-CPA property exists, then an adversary against the LOR-IND property exists too. By contraposition, if the ciphering scheme is LOR-IND-CPA secure, then it is IND-CPA secure.

In [BDJR97], it has been shown that LOR-IND security is in fact equivalent to IND security, under the condition to increase the number of requests to the oracle.

# 5 Security proof of the canonical SSSC

## 5.1 IND-CCA security of the canonical SSSC

The ideal case for a canonical SSSC is when the filtering function is randomly chosen among all the $2^{2^n}$ Boolean functions. This ideal case is not realistic as it would imply an exponential key size. However, even in this situation, the following property holds:

**Proposition 3.** *The canonical SSSC cannot reach the IND-CCA security*

*Proof.* Consider the special situation when the challenger provides $m_0 = 0 \cdots 0$, the message that only involves symbols 0, and $m_1 = 1 \cdots 1$, the message that only involves symbols 1. It receives a cryptogram $c$. It modifies only the last symbol of the cryptogram and asks the oracle to decrypt. If the answer of the oracle starts with a sequence of 0, then $c$ is the encryption of $m_0$ and if it starts with a sequence of 1, then $c$ is the encryption of $m_1$. Following this strategy, the challenger always wins. That suffices to complete the proof.

Actually, this is due to the fact that the cryptograms produced by an SSSC are malleable. Indeed, they can be modified at their end without effect on the beginning of the plaintext.

The only security level that can be expected for the canonical SSSC is the IND-CPA as shown in next section.

## 5.2 IND-CPA security of the canonical SSSC

The main result of this section is to prove that the wprf property of the filtering function characterizes the IND-CPA security of the canonical SSSC.

**Proposition 4.** *A canonical SSSC reaches the LOR-IND security if and only if the filtering functions is wprf.*

*Proof.* We first prove that if the filtering function of a canonical SSSC is wprf, then the SSSC reaches the LOR-IND security. In order to prove this property, we prove that if a polynomial adversary $\mathcal{A}$ exists against the SSSC, then we can deduce a challenger $\mathcal{B}$ against the wprf property of the filtering function. Let $\mathcal{B}$ be defined as follow:

1. $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$.
2. For each received couple of messages $(m_0^i, m_1^i)$, the simulator $\mathcal{B}$ requests for the wprf oracle the sequence of bits $m_b^i$ and receives form the oracle the sequence $\left(x_k^i, f(x_k^i)\right)$. From this sequence, $\mathcal{B}$ computes a cryptogram $c$ it sends to $\mathcal{A}$.

3. After a polynomial times of queries, the adversary $A$ answers a bit $\widehat{b} \in \{0,1\}$. If $b = \widehat{b}$ then $\mathcal{B}$ answers wprf, and if $b \neq \widehat{b}$ then it answers rf.

It must be shown that if the advantage of $\mathcal{A}$ is non negligible, then the advantage of $\mathcal{B}$ is non negligible too. From the definition of $\mathcal{B}$, its advantage is:

$$\mathrm{Adv}(\mathcal{B}) = \left|\mathrm{Pr}_{\mathrm{wprf}}(b = \widehat{b}) - \mathrm{Pr}_{\mathrm{rf}}(b = \widehat{b})\right|.$$

In the wprf world, the challenger $\mathcal{B}$ acts for $\mathcal{A}$ as a true ciphering oracle. Thus, the advantage of $\mathcal{A}$ is:

$$\begin{aligned}\mathrm{Adv}(\mathcal{A}) &= \left|2\mathrm{Pr}_{\mathrm{wprf}}(b = \widehat{b}) - 1\right| \\ &\leq \left|2\mathrm{Pr}_{\mathrm{wprf}}(b = \widehat{b}) - 2\mathrm{Pr}_{\mathrm{rf}}(b = \widehat{b})\right| + \left|2\mathrm{Pr}_{\mathrm{rf}}(b = \widehat{b}) - 1\right| \qquad (2)\end{aligned}$$

The first term is twice the advantage of $\mathcal{B}$. Moreover, as in the random world, the answers of the wprf oracle are random, the answers of $\mathcal{B}$ to $\mathcal{A}$ are random too. In consequence, the answer of $\mathcal{B}$ is random and the second term in the second member of inequality (2) is null. It follows that:

$$\mathrm{Adv}(\mathcal{A}) \leq 2\mathrm{Adv}(\mathcal{B}),$$

which proves the expected result.

It remains to prove the reciprocal: if the canonical SSSC reaches the LOR-IND-CPA security, then the filtering function achieves the wprf property. By contraposition, it is equivalent to prove that if a wprf challenger exists against the filtering function, then a LOR-IND adversary can be constructed against the canonical SSSC.

Given a wprf challenger $\mathcal{B}$, then a LOR-IND adversary is constructed as follow:

1. On each query $m^i$ from challenger $\mathcal{B}$, the adversary $\mathcal{A}$ generates a random binary sequence $a^i$ and sends to the LOR-IND oracle the couple $(m^i, a^i)$.
2. The oracle encrypts either $m^i$ or $a^i$ according to the value of a random bit $b \in \{0,1\}$ chosen once for all. The oracle returns to $\mathcal{A}$ the corresponding cryptogram $c^i$.
3. From the received cryptogram $c^i$, the adversary $\mathcal{A}$ computes the values of the filtering functions used by the oracle, assuming that the oracle has encrypted the message $m^i$, and transmits the sequence of couples $\left(x_k^i, f(x_k^i)\right)$ to $\mathcal{B}$.
4. After a polynomial number of queries, the challenger $\mathcal{B}$ returns an answer: rf or wprf. If $\mathcal{B}$ answers wprf, then $\mathcal{A}$ returns $\widehat{b} = 0$ and if $\mathcal{B}$ answers rf, then $\mathcal{A}$ returns $\widehat{b} = 1$.

If the LOR-IND oracle chooses $b = 0$, then its answers to $\mathcal{B}$ correspond to the encryption of the binary sequences provided by $\mathcal{A}$, and $\mathcal{B}$ simulates to $\mathcal{A}$ an oracle in the wprf world. Conversely, if the LOR-IND oracle chooses $b = 1$, and as the sequences $a^i$ are random, $\mathcal{B}$ simulates to $A$ an oracle in the rf world. In consequence, the advantage of $\mathcal{B}$ equals exactly the advantage of $\mathcal{A}$ and that achieves the proof.

From the above propositions, the following result holds:

**Corollary 1.** *If the filtering function of a canonical SSSC is prf, then it achieves the IND-CPA security.*

*Proof.* Assume that the filtering function of a canonical SSSC is prf. From Proposition 1, it is wprf. From Proposition 4, the canonical SSSC also achieves the LOR-IND security, and finally, from Proposition 2, it achieves the IND-CPA security too. This is summarized by the following implications:

$$\text{prf} \Rightarrow \text{wprf} \Leftrightarrow \text{LOR-IND-CPA} \Rightarrow \text{IND-CPA}$$

## 6 Concluding remarks

The cryptological complexity of the canonical form of the Self-Synchronizing Stream Cipher lies in the filtering function. The maximum security is achieved when this function is a pure random function. It has been shown in this paper that, even in this ideal case, this kind of cipher cannot reach the IND-CCA security. This is due to the fact that the ciphertexts are malleable. Thus, the maximum expected security is IND-CPA. In realistic implementation, the filtering is a pseudo random function, for example a single output of a block cipher primitive. We have proved that for such practical ciphers, the IND-CPA security can be guaranteed. The interest of the result lies in that it guarantees the existence of SSSC that can be IND-CPA secure although till now, the SSSC proposed in the open literature had be broken against IND-CPA attacks.

Finally, a weaker property than pseudo-randomness has been proposed. It had been called wprf. It allows to assess a stronger security level than indistinguishability, mainly left-or-right indistinguishability, against chosen plaintext attacks. Insofar as the IND-CPA of the canonical SSSC was an open problem for a long times, we think that the result of this paper will be a further motivation to consider new searches of secure designs of SSSC.

**Acknowledgement**
This work was supported by Research Grants ANR-13-INSE-0005-01 from the Agence Nationale de la Recherche. We thank Duong-Hieu PHAN for providing helpful discussions in the establishing of the security proof.

## References

[BDJR97] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *In Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 394–403. IEEE Computer Society Press, 1997.

[BDM15] Philippe Guillot Brandon Dravie and Gilles Millérioux. Security proof of the canonical form of self-synchronizing stream ciphers. In *Proceedings of Workshop on Cryptography and Coding, WCC 2015, Paris, France*, 2015.

[BR05] Mihir Bellare and Phillip Rogaway. Introduction to modern cryptography. In *UCSD CSE 207 Course Notes*, page 207, 2005.

[Dae95] Joan Daemen. *Cipher and Hash function design, strategies based on linear and differential cryptanalysis.* PhD thesis, Katholieke Universiteit Leuven, 1995.

[DGV92] Joan Daemen, Ren Govaerts, and Joos Vandewalle. A practical approach to the design of high speed self-synchronizing stream ciphers. In *SINGAPORE ICCS/ISITA 92*, pages 279–293. IEEE, 1992.

[DK08] Joan Daemen and Paris Kitsos. The self-synchronizing stream cipher moustique. In *New Stream Cipher Designs - The eSTREAM Finalists*, pages 210–223. 2008.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 365–377, 1982.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[JM03] Antoine Joux and Frédéric Muller. Loosening the KNOT. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, pages 87–99, 2003.

[JM05] Antoine Joux and Frédéric Muller. Two attacks against the HBB stream cipher. In *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*, pages 330–341, 2005.

[JM06] Antoine Joux and Frédéric Muller. Chosen-ciphertext attacks against MOSQUITO. In *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, pages 390–404, 2006.

[Klí05] Vlastimil Klíma. Cryptanalysis of hiji-bij-bij (HBB). *IACR Cryptology ePrint Archive*, 2005:3, 2005.

[KRB+08] Emilia Käsper, Vincent Rijmen, Tor E. Bjørstad, Christian Rechberger, Matthew J. B. Robshaw, and Gautham Sekar. Correlated keystreams in moustique. In *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings*, pages 246–257, 2008.

[Mau91] Ueli M. Maurer. New approaches to the design of self-synchronizing stream ciphers. In *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, pages 458–471, 1991.

[MG10] Gilles Millerioux and Philippe Guillot. Self-synchronizing stream ciphers and dynamical systems: State of the art and open issues. *I. J. Bifurcation and Chaos*, 20(9):2979–2991, 2010.

[MvOV96] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[oS80] National Bureau of Standards. Des mode of operations. Technical report, Institute for Computer Sciences and Technology, National Bureau of Standards, Springfield, VA, Decembre 1980.

[Par12] J. Parriaux. *Control, synchronization and encryption.* PhD thesis, Université de Lorraine, 2012.

[PGM11] J. Parriaux, P. Guillot, and G. Millérioux. Towards a spectral approach for the design of self-synchronizing stream ciphers. *Cryptography and Communications*, 3:259–274, 2011. 10.1007/s12095-011-0046-2.

[Sar03] Palash Sarkar. Hiji-bij-bij: A new stream cipher with a self-synchronizing mode of operation. *IACR Cryptology ePrint Archive*, 2003:14, 2003.