Resynchronization Interval of Self-synchronizing Modes of Block Ciphers

Karel Burda

The Faculty of Electrical Engineering and Communication
Brno University of Technology, Brno, Czech Republic

Summary

In some communication systems, some of the transmitted bits or bit groups are lost or, on the contrary, repeated. Lost or repeated bits or bit groups are called slips. In the case of cipher transmission, these slips produce a loss of the synchronization between transmitter and receiver, which is accompanied by the loss of the information being transmitted. Self-synchronizing modes of block ciphers are used for the elimination of this phenomenon. An important characteristic of these mode types is the average number of bits that were transmitted between the end of the slip and the moment of synchronization recovery - so-called resynchronization interval. In this paper, the probability distribution of the occurrence of a synchronization sequence in the cryptogram is derived first. On this basis, formulas for computing the resynchronization interval of OCFB and modified SCFB statistical self-synchronizing modes of block ciphers are then derived. The formulas obtained are compared with the formula for the resynchronization interval of the CFB mode. The results obtained may be used to choose a suitable self-synchronizing mode of a block cipher.

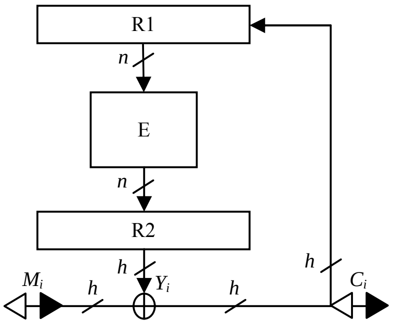
Key words:
self-synchronizing mode, block cipher mode, CFB mode, OCFB mode, SCFB mode

1. Introduction

In the paper, we assume that cryptograms are encrypted or decrypted in bit groups with length $h \geq 1$ bit. We will refer to these h-tuples as bytes. In communication systems, cryptogram bytes are transmitted per certain transmission units (e.g. bits or octets) with length $l \geq 1$ bit. In some communication systems, however, some of the transmitted l-tuples are lost or, on the contrary, repeated. Lost or repeated bits or bit groups are called slips. In the case of cipher transmission, these slips produce a loss of the synchronization between transmitter and receiver, which is accompanied by the loss of the information being transmitted. Self-synchronizing modes of block ciphers are used for eliminating the above effect of slips. The condition for self-synchronism is the requirement that l is an integer multiple of byte length h. In this paper, we assume that the above requirement is fulfilled.

Currently, three types of self-synchronizing modes of block ciphers are published – CFB, OCFB and SCFB modes. The block cipher used in these modes operates with blocks of n bits in length, whereas the block cipher length is an integer multiple N of the byte length, i.e. it is valid that $n = N \cdot h$. A message is encrypted or decrypted per single bytes in CFB, OCFB and SCFB modes. Let us denote the i-th byte of a message, cryptogram and keystream by symbols M_i, C_i and Y_i, respectively. For encrypting, it is valid that $C_i = M_i \oplus Y_i$, where the symbol $\oplus$ denotes bit-by-bit modulo-two addition of bytes. For decrypting, it is valid that $M_i = C_i \oplus Y_i$. Particular modes differ in the by manner of deriving keystream bytes Y_i.

The CFB mode [1] is shown in Fig. 1. The previous bytes C_{i-1} to C_{i-N} of the cryptogram are stored in the shift register R1. This n-bit block is enciphered by block cipher E and the result is stored in output register R2. After this, the first byte from R2 is used as the current keystream byte Y_i. The procedure described is repeated for encrypting or decrypting further bytes, with a pre-arranged initialization vector IV being used for generating the first keystream byte Y_1. Depending on the byte size h, the CFB mode is denoted as an h-CFB mode, with the most used variants being the modes with h = 1 or 8 in practice. The variant with h = 1 allows resynchronization after slips of arbitrary length while the variant with h = 8 permits resynchronization only after slips whose length is an integer multiple of eight bits. In all variants of this mode, synchronism is restored after N bytes, i.e. after one n-bit block. A disadvantage of the CFB mode is the low utilization of the block cipher, because only one byte from N possible keystream bytes is used.



Legend: ▶ = for encrypting,
◁ = for decrypting.

Fig. 1: The CFB mode.

A modification of the CFB mode is a mode called

Manuscript received October 5, 2007
Manuscript revised October 20, 2007

Optimized CFB (OCFB) mode [2]. The principle of the *h*-OCFB mode is illustrated in Fig. 2. At the beginning, the shift register R1 is filled with initialization vector *IV*. This vector is encrypted, the result is stored in register R2 and bytes from this register are used as keystream bytes Y_i. Meanwhile, cryptogram bytes C_i are successively stored in shift register R1. After the *N*-th cryptogram byte is processed, a new *n*-tuple of bits is contained in register R1. The new content of R1 is then encrypted and another *N* bytes of keystream are obtained. The action described is periodically repeated excepting a situation when a pre-arranged sequence of *s* bits occurs in the left part (i.e. at the end) of register R1. This sequence is called the synchronization sequence SYN. When this sequence is detected by the appropriate detector DET, then a new encryption is performed in spite of the fact that all *N* keystream bytes have not been used yet. When the receiver receives errorless bits and simultaneously no slips occur after sequence SYN, the same *n*-bit blocks occur in registers R1 of the transmitter and receiver. Thus, in both the transmitter and the receiver, the same bytes of the keystream are generated and these bytes are used for encrypting or decrypting the same cryptogram bytes C_i. In this way, synchronization is restored and maintained in the OCFB mode. The average resynchronization interval depends on the probability of the occurrence of the synchronization sequence. A better utilization of the block cipher is an advantage of the mode described because as many as *N* cryptogram bytes are obtained with any operation E compared with one byte in the case of the CFB mode.

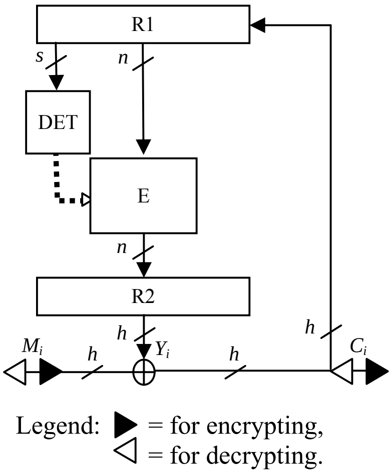


Fig. 2: The OCFB mode.

The statistical CFB mode (SCFB mode) is another self-synchronizing mode of block ciphers [3, 4]. In principle, the SCFB mode is a hybrid between the OFB (Output Feedback Mode [1]) and the OCFB modes. Usually, the OFB mode is used and at the same time the sequence of cryptogram bits is continuously observed. When the SYN sequence is detected in the cryptogram, the resynchronization is performed. The principle of the SCFB mode is shown in Fig. 3. At the beginning, initialization vector *IV* is inserted into register R1, which is then encrypted and the result of encryption is stored in register R2. Subsequently, bytes from this register are successively used as *N* keystream bytes Y_i. Cryptogram bytes C_i are being stored in shift register R3 of both the transmitter and the receiver. Usually, the content of the R2 register is replicated into R1 after using up all keystream bytes, the new content of the R1 register is encrypted and the result of encryption is used as a new keystream. The operation described is periodically repeated (OFB mode practically). An exception is the situation when a synchronization sequence occurs in the cryptogram. When this sequence is detected at the end of the R3 register, the content of R3 is encrypted and the result of encryption in R2 is used as a keystream (OCFB mode practically). Subsequently, the system returns to the OFB mode until a new synchronization sequence occurs at the end of register R3. The principle of restoring and maintaining the synchronization in the SCFB mode is the same as in the OCFB mode. The other characteristics are the same too. The average resynchronization interval for the SCFB mode depends on the occurrence probability of the synchronization sequence and the SCFB mode makes a better use of the block cipher than the CFB mode does.

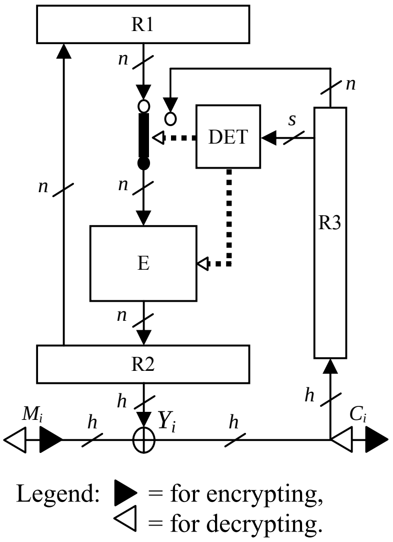


Fig. 3: The SCFB mode.

In this connection it is necessary to note that, so far, we have described a modified version of the SCFB mode. The original version of the SCFB mode [3, 4] encrypts per bits only (i.e. not per bytes) and when statistical resynchronization has taken place any potential statistical resynchronization is blocked for *n* bits. Further, the length of the R3 register is not *n* but (*n*+*s*) bits in the original variant. The last *n* bits from the R3 register are used for encryption when a synchronization sequence is detected. The modification described in this paper allows comparing the SCFB mode with the other self-synchronizing modes and, moreover, it also reduces the average resynchronization interval of the SCFB mode. It is the minimization of this interval that has been the aim of our research.

2. State of the art

The basic performance criterion of self-synchronizing modes of block ciphers is resynchronization interval D, which is the average number of bytes between the last byte of a slip and the nearest synchronously processed byte. The resynchronization interval of the CFB mode is equal to the block length of the cipher, i.e. $D_{\mathrm{CFB}} = n$. This is due to the fact that when the last byte of the slip leaves the R1 register (i.e. byte $C_{i\text{-}N\text{-}1}$), then identical bytes $C_{i\text{-}N}$ to $C_{i\text{-}1}$ occur in registers R1 of both the transmitter and the receiver. (Note: It is assumed that bytes $C_{i\text{-}N}$ to $C_{i\text{-}1}$ are errorless bytes and that no slips have occurred after slip $C_{i\text{-}N\text{-}1}$ yet.) In this way, the same keystream is obtained and thus synchronism is restored beginning with the currently transmitted byte C_i.

The situation is more complicated with statistical self-synchronization modes, because the resynchronization interval is a random quantity. An approximation of $D = 2^s$ bytes is introduced in [2] on the assumption that the length of synchronization sequence $s \leq h$. This condition means that the whole synchronization sequence SYN must occur in a single byte. We will call this byte the synchronization byte. The occurrence probability of this byte in a cryptogram is obviously equal to the value $p = 2^{-s}$. With respect to the assumption of the randomness and mutual independence of bits in a cryptogram, the randomness and mutual independence of bytes hold too. Then, the probability that the synchronization byte is exactly (k-1) non-synchronization bytes away from some randomly selected byte (from the last byte of the slip in our case), can be expressed by the geometric distribution:

$$P(k) = (1-p)^{k-1} \cdot p\,,\; k = 1, 2, 3..., \tag{1}$$

The mean value of this random variable is equal to $E = 1/p = 2^s$. Then the average interval from the end of the slip to the end of the synchronization byte $d = 2^s$ [byte] = $h{\cdot}2^s$ [bit]. In [2], this value is given as a resynchronization interval D. We will see later that this statement is not quite exact.

For the original SCFB mode (i.e. when $h = 1$), quite a complex approximation of D is derived in [3], which for big values of s can be simplified to the formula $D \approx 2^s$ bits. For this derivation, the author used the assumption that the probability distribution of binary synchronization s-tuples is given by the shifted geometric distribution:

$$P(k) = (1-p)^{k} \cdot p\,,\; k = 0, 1, 2, 3 ..., \tag{2}$$

where again $p = 2^{-s}$. However, this assumption is not correct, because the random variable k for this distribution is not in units of bits, but in units of bit s-tuples. The following considerations thus lead to approximate results only. The OCFB and SCFB modes are mutually compared in [5], but the respective values of D were obtained via simulation.

The above description of the current state leads to the conclusion that only approximations of the resynchronization interval are known for the published OCFB and SCFB modes while for the modified SCFB mode (i.e. when $h \geq 1$) even an estimate of this interval is unknown. In this paper, the probability distribution of intervals d between a randomly selected cryptogram byte (the last byte of a slip in our case) and the terminal byte of so-called closed synchronization sequence (see bellow) is derived first. Formulas for the resynchronization intervals of the OCFB mode and the modified SCFB mode are then derived on this basis.

3. Mathematical model

In this paper, we assume that the transmission channel does not cause any errors and that the values of the cryptogram bits are mutually independent random variables. At the same time, the occurrence probability of zero bits in the cryptogram is equal to the occurrence probability of one bits.

Finally, we assume that the so-called closed synchronization sequence is used for resynchronization. A closed sequence is a sequence of symbols in which none of its prefixes is simultaneously its suffix. An example of a closed sequence is the bit string 10000. On the contrary, the sequence 10010 is not a closed sequence, because its prefix 10 (the first two bits) is simultaneously its suffix (the last two bits). It is evident that closed sequences do not overlap in the cryptogram and therefore their minimal mutual distance is equal to the sequence length s. Thus, a minimal interval between any two adjacent statistical resynchronizations is s bits, which is advantageous from the viewpoint of block cipher utilization.

In this part, we derive the probability distribution of intervals d, where d is the distance from the last byte termination of the slip in a cryptogram to the last byte termination of the nearest synchronization sequence SYN. For the purpose of better intelligibility, we will call the SYN synchronization sequence the SYN string for short or SYN only. The byte length of the SYN string $S = \lceil s/h \rceil$, where the value S of function $\lceil s/h \rceil$ expresses the nearest integer for which it is valid $S \geq s/h$. We assume that the beginning of the SYN string is identical to the beginning of the respective byte. At the end of the last (i.e. the S-th) byte of the SYN string, there are $r = (S{\cdot}h - s)$ bits, which are not part of SYN and which therefore may take an arbitrary value.

The above quantities are illustrated by an example in Fig. 4, where it holds that $S = 2$ and $r = 1$ for $h = 3$ and $s = 5$. The random variable k is the number of bytes from the last byte termination of the slip to the last byte termination of the SYN string. We will number these bytes from the last byte of SYN backwards to the slip. In Fig. 4, the situation is shown when $k = 4$. Note that the above numbering has the meaning of a distance and not of the order in time. It means that in our example, byte number 1 is the last byte of the observed sequence and byte number 4 is the first byte of this sequence.

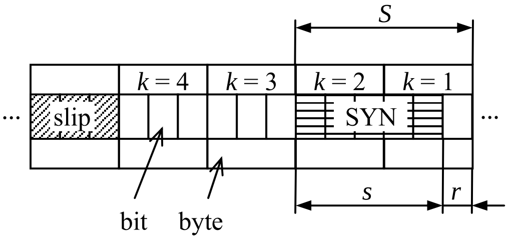


Fig. 4: Parameters of the SYN string.

For the derivation of the distribution sought, we define the terms *terminal sequence* and *bilateral sequence*. The terminal sequence is a sequence that contains a SYN in its termination only. On the contrary, the bilateral sequence is a sequence that contains a SYN at both its beginning and its termination. The number of terminal sequences of length k bytes is denoted $T(k)$ and the number of bilateral sequences of length k bytes is $U(k)$.

Due to the length of SYN, the SYN string cannot be contained in the last k bytes, when $k < S$. Evidently, it is valid that $T(k) = 0$ in this case. For $k = S$, the SYN string can be contained in the last k bytes, with the last r bits taking arbitrary values. The number of these terminal sequences is obviously 2^r and therefore $T(S) = 2^r$. The number of terminal sequences for $k > S$ can be computed iteratively on the basis of the following, self-evident property of terminal sequences: adding any byte in front of the terminal sequence with the length (k-1) bytes, gives rise to a sequence with the length k bytes which is either terminal or bilateral. Therefore, it obviously holds:

$$T(k\text{-}1)\cdot 2^h = T(k) + U(k), \qquad (3)$$

where 2^h represents the number of all possible values of the byte added.

If the added byte does not cause the creation of the SYN string at the beginning of the sequence, then this new sequence remains terminal. In opposite case, a bilateral sequence arises. Note that the SYN string fills s bits in a sequence S bytes long and that the remaining r bits can take an arbitrary value. It follows that a bilateral sequence k bytes long arises by adding one sequence from 2^r different sequences S bytes long with the SYN string in front of an arbitrary terminal sequence (k-S) bytes long. Then, for the number of bilateral sequences $U(k)$ it is valid:

$$U(k) = 2^r\cdot T(k\text{-}S). \qquad (4)$$

By substituting this formula in (3) and by elementary rearrangements, we obtain the formula for iteratively computing the number of terminal sequences $k > S$ bytes long:

$$T(k) = 2^h\cdot T(k\text{-}1) - 2^r\cdot T(k\text{-}S). \qquad (5)$$

Now, we can write the resultant formula for computing the number of terminal sequences:

$$T(k) = \begin{cases} 0, \text{for } k = 1,2,\ldots S-1, \\ 2^r, \text{for } k = S, \\ 2^h\cdot T(k-1) - 2^r\cdot T(k-S), \text{otherwise.} \end{cases} \qquad (6)$$

We denote $P(k)$ the occurrence probability of the terminal sequence k bytes long. Obviously, it is valid that:

$$P(k) = \frac{T(k)}{2^{h\cdot k}}, \qquad (7)$$

where $2^{h\cdot k}$ is the number of all binary sequences k bytes long. This probability is simultaneously the probability distribution of intervals between the last byte termination of the SYN sequence and the last byte termination of the slip. After substituting (6) in (7) and after some elementary rearrangements, we obtain the following formula for $P(k)$:

$$P(k) = \begin{cases} 0, \text{for } k = 1,2,\ldots S-1, \\ 2^{-s}, \text{for } k = S, \\ P(k-1) - P(S)\cdot P(k-S), \text{otherwise.} \end{cases} \qquad (8)$$

An example of the above distribution is shown in Fig. 5 for $h = 1$ and $s = 5$ (i.e. $S = 5$). It is evident from the figure that a characteristic feature of the $P(k)$ distribution is the fact that the first (S-1) probabilities are zero, the next S probabilities are identical and have the value 2^{-s}, and then the remaining probabilities fall monotonously. The mean of $P(k)$ is (see the Appendix):

$$\mathrm{E(k)} = 2^s. \qquad (9)$$

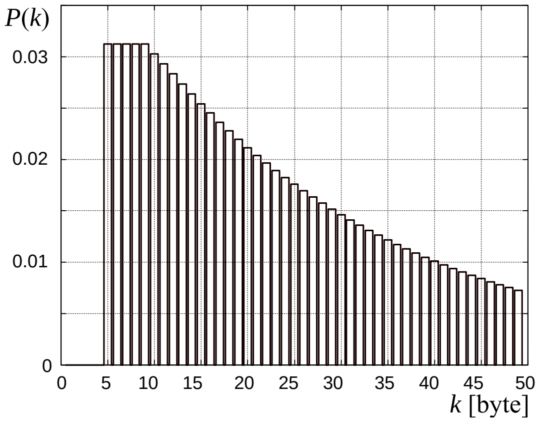


Fig. 5: The probability distribution $P(k)$ for $h = 1$ and $s = 5$.

We can see from Fig. 5 that the approximation of $P(k)$ by the geometric distribution is not very good, especially for greater values of S. However, an interesting feature both of the $P(k)$ distribution and of the geometric distribution is the identical means.

The mean $\mathrm{E}(k)$ is in fact the mean d of intervals between the last byte termination of the SYN sequence and the last byte termination of the slip. Thus, we can write:

$$d = \mathrm{E}(k) = 2^s \text{ [byte]} = h\cdot 2^s \text{ [bit]}. \qquad (10)$$

On this basis, we can now express the average resynchronization interval D of the OCFB and the modified SCFB modes.

4. Results

In the case of the OCFB mode, resynchronization happens when the SYN sequence occurs in the last S bytes of the R1 register. The last byte of the slip has already left

R1 and, meanwhile, $d + (N\text{-}S)$ bytes have been processed on average. The first addend d represents the average number of cryptogram bytes processed between the last byte termination of the slip and the last byte termination of the SYN sequence. The second addend $(N\text{-}S)$ represents the number of bytes needed to fill the rest of the R1 register in order that the SYN sequence should occupy the last S bytes of R1. Thus, the average resynchronization interval of the OCFB mode is equal to:

$D_{\mathrm{OCFB}} = d + (N\text{-}S) = 2^s + (N\text{-}S)$ [byte]. (11) In [2], it is stated that for the OCFB mode $D = 2^s$. We can see that this formula is not exact, because it neglects the influence of the R1 register length.

In the case of the modified SCFB mode, resynchronization happens when the SYN sequence occurs in the last S bytes of the R3 register. The lengths of R3 for the modified SCFB mode and R1 for the OCFB mode are identical and both these registers are filled by the immediately preceding cryptogram bytes. Thus, this situation is analogous to the resynchronization situation for the OCFB mode and therefore we can, by analogy, write that the average resynchronization interval for the SCFB mode is:

$D_{\mathrm{SCFB}} = d + (N\text{-}S) = 2^s + (N\text{-}S)$ [byte]. (12)

It follows from the two previous formulas that the OCFB and the modified SCFB modes are entirely equivalent from the viewpoint of resynchronization intervals. In this connection, note that the original SCFB mode blocks any possibility of statistical resynchronization in the interval of the following n bits when the statistical resynchronization has been realized. As a result, the average resynchronization interval becomes extended compared with the modified SCFB mode. This is to mean that the original SCFB mode has in no case a better average resynchronization interval than the OCFB and the modified SCFB modes. This fact is validated in [5] by a simulation comparison of the original SCFB mode and the OCFB mode. The resynchronization interval of the modified SCFB mode is a lower boundary of D for any variant of the SCFB principle.

The formulas for the average resynchronization intervals of all modes examined are clearly arranged in Table 1.

Table 1: Formulas for the average resynchronization interval of self-synchronizing modes.

Type of mode	D [byte]	D [bit]
CFB	N	$h \cdot N = n$
OCFB	$2^s + (N\text{-}S)$	$h \cdot [2^s + (N\text{-}S)]$
modified SCFB	$2^s + (N\text{-}S)$	$h \cdot [2^s + (N\text{-}S)]$

We can see that the value of the average resynchronization interval of statistical modes depends on the synchronization sequence length s, on the byte length h and on the length n of the feedback register (i.e. on the cipher block length). In Fig. 6, the dependence of the average resynchronization interval D on the synchronization sequence length s is shown for byte lengths $h = 1$ and 8. This dependence is valid for a block cipher with block length $n = 128$ bits. It is evident from the figure that the CFB mode is the optimal mode from the viewpoint of resynchronization interval because offers the shortest resynchronization interval D. In the OCFB and the modified SCFB modes the interval D is always worse and grows approximately exponentially with the growing synchronization sequence length s. From the figure, the great effect of byte length h on the resultant value of the resynchronization interval is also evident. Thus for a practical application of the self-synchronization modes, the conclusion follows that the choice of the statistical resynchronization is correct only in the case when the resynchronization delay is not a critical requirement. A further conclusion is that the minimal resynchronization delay requires a minimal value of byte length h, a minimal value of block cipher length n, and, in particular, a minimal value of synchronization sequence length s.

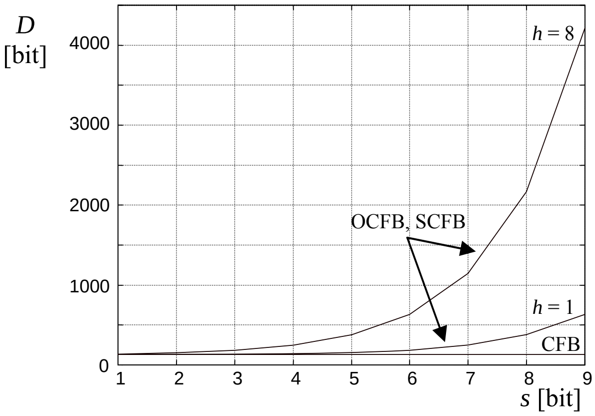


Fig. 6: The dependence of the average resynchronization interval D on the synchronization sequence length s for $n = 128$ bits.

5. Conclusion

The theoretical contribution of this paper is a mathematical representation of the occurrence of closed synchronization sequences in a cryptogram. This representation is given by formula (8) and holds for the general case when the transmission is realized both per bits (i.e. $h = 1$) and per bytes (i.e. $h > 1$), and the synchronization is performed by a binary sequence, which may be contained even in more than one byte in the case of byte transmission. In this way, particular cases are simultaneously described, when both the transmission and the synchronization are of the byte character or, on the contrary, of the bit character.

The practical contribution of this paper is formulas (11) and (12) for finding accurately the resynchronization interval of statistical modes of block ciphers. These formulas can be used in the design phase of the block cipher mode, allowing an optimization of the selection of the type and parameters of the mode.

References

[1] Dworkin, M.: Recommendation for Block Cipher Modes of Operation. National Institute of Standards and Technology, Gaithersburg 2001.

[2] Alkassar, A. - Geraldy, A. - Pfitzmann, B. – Sadeghi, A.: Optimized Self-Synchronizing Mode of Operation. In Proceedings of the 8th International Workshop on Fast Software Encryption, Yokohama, April 2001, Proceedings in LNCS 2355, Springer-Verlag.

[3] Heys, H.: An Analysis of the Statistical Self-Synchronization of Stream Ciphers. In Proceedings of IEEE INFOCOM 2001, 22-26 April 2001, Anchorage, Alaska, USA, Volume 2, pp. 897-904.

[4] Heys, H.: Analysis of the Statistical Cipher Feedback Mode of Block Ciphers. IEEE Transactions on Computers, vol. 52, no. 1, Jan. 2003, pp. 77-92.

[5] Yang, F. - Heys, H.: Comparison of Two Self-Synchronizing Cipher Modes. In Proceedings of Queen's 22nd Biennial Symposium on Communications, Kingston, Ontario, Jun. 2004.

Appendix

The goal of this Appendix is to prove that the mean of probability distribution (8) is equal to the value 2^s. Note that the value $P(k) = 0$ for $k = 1, 2, .., S$-1. The following equalities are a consequence the above property:

$$\sum_{k=S}^{\infty} P(k) = \sum_{k=1}^{\infty} P(k) = 1,$$

$$\sum_{k=S}^{\infty} k \cdot P(k) = \sum_{k=1}^{\infty} k \cdot P(k) = E(k).$$

These equalities are used in the following derivation of the mean $E(k)$:

$$\begin{aligned}
E(k) &= \sum_{k=1}^{\infty} k \cdot P(k) = \sum_{k=1}^{S-1} k \cdot 0 + S \cdot P(S) + \\
&+ \sum_{k=S+1}^{\infty} k \cdot \left[P(k-1) - P(S) \cdot P(k-S)\right] = \\
&= 0 + S \cdot P(S) + \sum_{k=S+1}^{\infty} k \cdot P(k-1) - \\
&- P(S) \cdot \sum_{k=S+1}^{\infty} k \cdot P(k-S) = S \cdot P(S) + \\
&+ \sum_{i=S}^{\infty} (i+1) \cdot P(i) - P(S) \cdot \sum_{i=1}^{\infty} (i+S) \cdot P(i) = \\
&= S \cdot P(S) + \sum_{i=S}^{\infty} i \cdot P(i) + \sum_{i=S}^{\infty} P(i) - \\
&- P(S) \cdot \sum_{i=1}^{\infty} i \cdot P(i) - S \cdot P(S) \cdot \sum_{i=1}^{\infty} P(i) = \\
&= S \cdot P(S) + \sum_{i=1}^{\infty} i \cdot P(i) + \sum_{i=1}^{\infty} P(i) - \\
&- P(S) \cdot \sum_{i=1}^{\infty} i \cdot P(i) - S \cdot P(S) \cdot \sum_{i=1}^{\infty} P(i) = \\
&= S \cdot P(S) + E(k) + 1 - P(S) \cdot E(k) - S \cdot P(S) \cdot 1 = \\
&= E(k) + 1 - P(S) \cdot E(k).
\end{aligned}$$

Thus, we now have the equality:

$$E(k) = E(k) + 1 - P(S) \cdot E(k)\,.$$

By simple rearrangement, we obtain from this equality the formula:

$$E(k) = \frac{1}{P(S)} = 2^s \cdot$$

This is the formula which was to be proved.

Karel Burda received the M.S. and PhD. degrees in Electrical Engineering from the Liptovsky Mikulas Military Academy in 1981 and 1988, respectively. During 1988-2004, he was a lecturer at two military academies. At present, he works at Brno University of Technology. His current research interests include the security of information systems and cryptology.