Applied Sig Process (1998)5:240–243

Applied
Signal
Processing

A Zero Suppression Algorithm for Synchronous Stream Cipher

Hoonjae Lee,[1] Bongjoo Park,[2] Byunghwa Chang[2] and Sangjae Moon[3]

[1]Department of Computer Engineering, Kyungwoon University, Korea; [2]Div. 5-4-2, Agency for Defense Development, Yuseung, Korea; [3]School of Electronic and Electrical Engineering, Kyungpook National University, Sankyuk-Dong, Korea

In synchronous digital communications it is important to recover the receiver clock, but this is difficult when received data contains long consecutive-zero sequences. We suggest a zero suppression (ZS) algorithm, which suppresses consecutive-zero sequences of more than k *zeros between successive ones in a ciphertext at the sender of a synchronous stream cipher system, and recovers the original message exactly at the receiver. The probability of* k *consecutive zeros in ciphertext at the sender is* 2^{-k} *in a synchronous stream cipher without ZS, but 0 with the suggested algorithm. The ZS algorithm does not affect cryptographic security when compared with a synchronous stream cipher without ZS. It is useful for systems which limit consecutive zeros, such as a T1-carrier system (k = 15).*

Keywords: Stream cipher; Zero suppression

1. Introduction

In synchronous digital communications, system performance depends on clock recovery accuracy at the receiver. In general, the receiver clock is recovered by a PLL (phase-locked loop) via 1-to-0 or 0-to-1 transition of the received data. But when there is no transition of consecutive data, that is, sequences of k consecutive zeros (000…000) or ones (111…111), it is more difficult for the receiver clock to be recovered exactly. For example, channel data (AMI coded) in T1-carrier systems are not permitted more than eight consecutive zeros by μ-law PCM coding, and 24-channel multiplexed trunk data are not permitted more than 15 consecutive zeros between successive ones by CCITT recommendation [1].

In a synchronous stream cipher, ciphertext bits are randomly distributed because they are the exclusive-ored value of a 0–1 balanced keystream with plaintext [2]. The probability of k consecutive zeros in ciphertext at the sender is thus 2^{-k} (Fig. 1(a)). In applying a synchronous stream cipher to point-to-point link encryption in a T1-carrier system, 15 or more consecutive zeros in ciphertext at the sender may be limited. Without limitation, the receiver clock may be unstable and the equipment then requires resynchronisation, since deciphering becomes impossible once synchronisation is lost. These resynchronisations, when frequent, can lead to significant deterioration of cryptographic security [3].

In this paper, we suggest a zero suppression (ZS) algorithm, which suppresses k or more zeros between successive ones in ciphertext at the sender and recovers the original message completely at the receiver of a synchronous stream cipher system. With the suggested algorithm, the probability of k consecutive zeros in ciphertext at the sender is reduced to 0 (Fig. 1(b)). The ZS algorithm does not affect cryptographic security when compared to a synchronous stream cipher without ZS. It is useful for systems which limit consecutive zeros, such as a T1-carrier system (with $k = 15$).

2. Zero Suppression Algorithm

Let

- $\mathbf{P}_i$ $(p_{in}, p_{in+1}, \cdots, p_{in+n-1})$ be the ith n-bit plaintext vector block,

Correspondence and offprint requests to: H. Lee, Department of Computer Engineering, Kyungwoon University, San 5-1 Indukri, Sandong-Myun, Kumi, Kyungbuk 730-850, Korea. E-mail: hjlee@kyungwoon.ac.kr

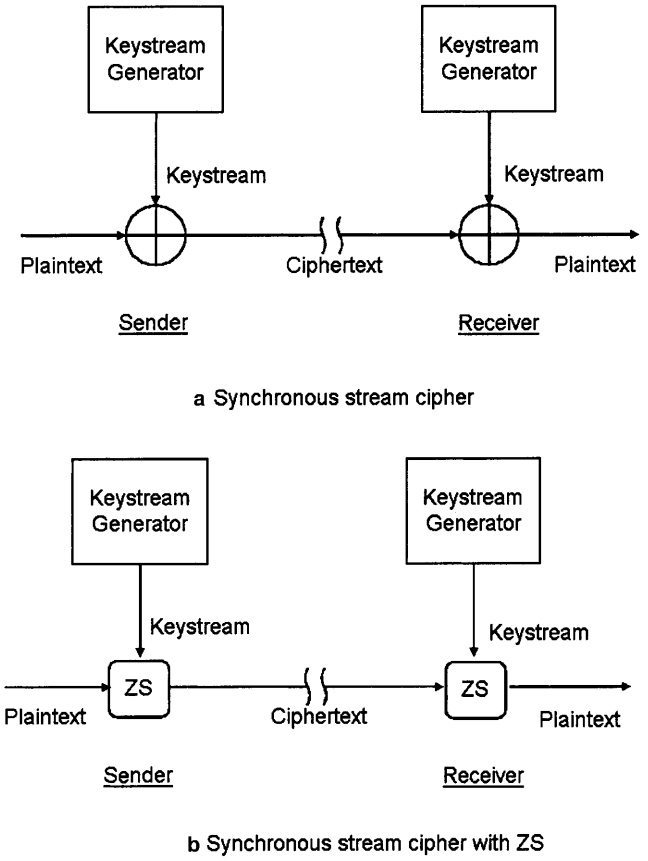

Fig. 1. Insertion of zero suppression algorithm.

- K_i $(k_{in}, k_{in+1}, \cdots, k_{in+n-1})$ the ith n-bit keystream block,
- $\mathbf{C}_i$ $(c_{in}, c_{in+1}, \cdots, c_{in+n-1})$ the ith n-bit ciphertext block,
- $\mathbf{Q}_i$ $(q_{in}, q_{in+1}, \cdots, q_{in+n-1})$ the ith n-bit recovered plaintext block at the receiver,
- $\mathbf{0}$ $(0, 0, \ldots, 0)$ the n-bit 0 vector $(i > 0)$.

The block size is $n = [(k+1)/2]$, where $[x]$ denotes the maximum integer which is not over x.

The ZS algorithm consists of a detection part which detects n-bit zeros (blocks of n consecutive zeros), and the substitution part which substitutes a nonzero block for n-bit zeros.

2.1. Assumptions

1. A redundant bit insertion or deletion at the sender is not permitted. (It is difficult problem to control clock rate in an intermediated synchronous stream cipher system).
2. For all i, $\mathbf{P}_i \neq \mathbf{0}$, i.e. P_i is a n-bit nonzero vector $(i \geq 0$, checking in block interval $n)$.
3. Cryptographically strong keystream sequences must be used in the system.

2.2. ZS Algorithm

Sender:

1. Put $\mathbf{P}_i \oplus \mathbf{K}_i$ and $\mathbf{K}_i$ to two n-stage shift registers respectively.
2. Check $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$.
3. If $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$, the output is $\mathbf{C}_i = \mathbf{K}_i$ (keystream block). Otherwise, the output is $\mathbf{C}_i = \mathbf{P}_i \oplus \mathbf{K}_i$ (ciphertext block).

Receiver:

1. Put $\mathbf{C}_i \oplus \mathbf{K}_i$ and $\mathbf{K}_i$ to two n-stage shift registers respectively.
2. Check $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{0}$.
3. If $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{0}$, the output is $\mathbf{Q}_i = \mathbf{K}_i$. Otherwise, the output is $\mathbf{Q}_i = \mathbf{C}_i \oplus \mathbf{K}_i$.

Theorem 1. Under the condition that for a plaintext block $\mathbf{P}_i \neq \mathbf{0}$ for all i in a synchronous stream cipher, the ZS algorithm limits the output of consecutive $2n - 1 (= k$ or $k - 1)$-bit zeros at the sender and (excluding channel error) recovers plaintext exactly at the receiver.

Proof.

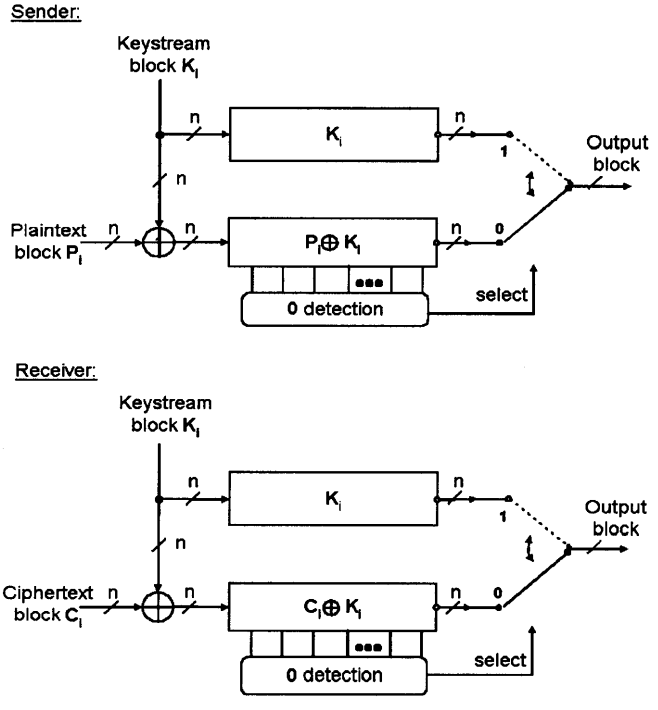

Fig. 2. Zero suppression algorithm.

(i) Since $\mathbf{P}_i \neq \mathbf{0}$ for all i, the output of ZS at the sender cannot be permitted to contain consecutive $(2n - 1)$-bit zeros.

(ii) When $\mathbf{P}_i \oplus \mathbf{K}_i = \mathbf{0}$ is detected, $\mathbf{C}_i = \mathbf{K}_i$ is transmitted to the receiver, and $\mathbf{P}_i = \mathbf{K}_i$. The receiver detects $\mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{K}_i \oplus \mathbf{K}_i = \mathbf{0}$ and then outputs $\mathbf{Q}_i = \mathbf{K}_i = \mathbf{P}_i$. So plaintext is completely recovered.

(iii) Whenever $\mathbf{P}_i \oplus \mathbf{K}_i \neq \mathbf{0}$, $\mathbf{C}_i = \mathbf{P}_i \oplus \mathbf{K}_i$ is transmitted to the receiver, the receiver deciphers $\mathbf{C}_i \oplus \mathbf{K}_i = (\mathbf{P}_i \oplus \mathbf{K}_i) \oplus \mathbf{K}_i = \mathbf{P}_i \neq \mathbf{0}$ and then outputs $\mathbf{Q}_i = \mathbf{C}_i \oplus \mathbf{K}_i = \mathbf{P}_i$. So plaintext is completely recovered.

In case of ciphertext-only attack on a stream cipher with the proposed ZS algorithm, an eavesdropper cannot find the substituted block (= keystream block) at the sender, because it is impossible to find more information from the ciphertext even though it equals plaintext itself whenever keystream sequences are k-bit zeros. In case of known-plaintext attack or chosen-plaintext attack, the main factor in maintaining cryptographic security is the strength of keystream generator, not the ZS algorithm. So the ZS algorithm does not affect cryptographic security when compared to the system without ZS.

In a synchronous stream cipher, ciphertext bits are randomly distributed, because they are the exclusive-ored value of a 0–1 balanced keystream with plaintext. The probability of k consecutive zeros in ciphertext at the sender is thus 2^{-k}. In applying a synchronous stream cipher to point-to-point link encryption in a T1-carrier system, 15 or more consecutive zeros in ciphertext at the sender may be limited. Without ZS, the receiver clock may be unstable and lose keystream synchronisation. The equipment then requires resynchronisation, since deciphering becomes impossible once synchronisation is lost. These resynchronisations, when frequent, can lead to significant deterioration of cryptographic security [3].

Applying the ZS to a T1-carrier system for link encryption with $k = 15$, $n = 8$, we can confirm improved system performance for clock recovery. As shown in Fig. 3, mean numbers of keystream synchronisation error in the system without the proposed ZS algorithm are rapidly increased by increasing the numbers of transmitted data and by decreasing k-parameter ($F = 2^{N-k}$). But the numbers are 0 by applying the proposed ZS algorithm which increases the system performances. Therefore, it seems that ZS prevents a serious deterioration of performance of keystream synchronisation.

On the other hand, there is occasionally an n-bit

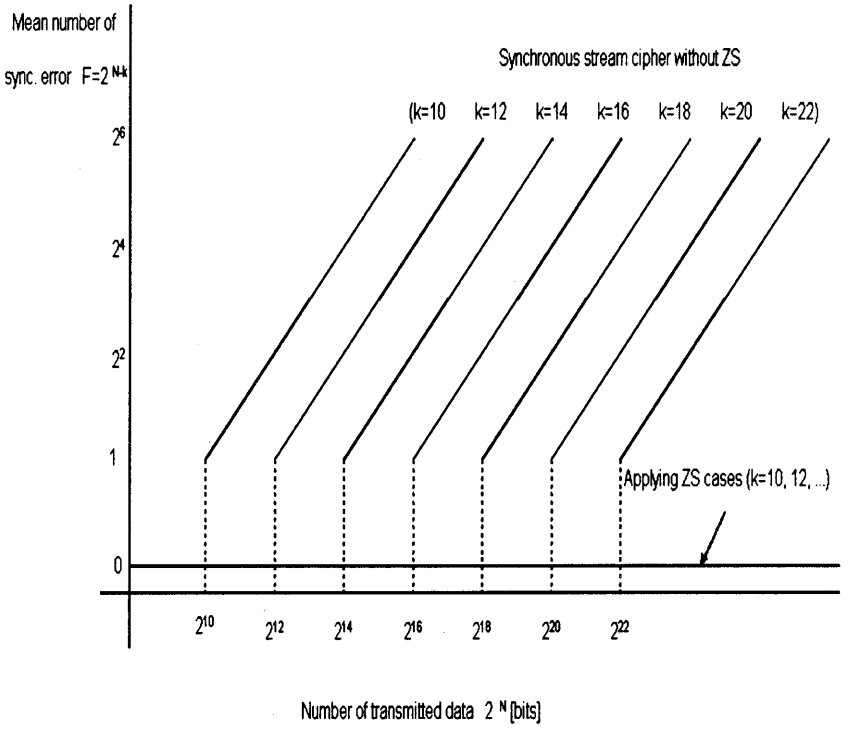

Fig. 3. Examples of mean numbers of keystream synchronisation error.

Table 1. Total error rate of ZS (by BER or n).

BER	Total error rate of ZS (k = 15, n = 8)	n	Total error rate of ZS (BER = 10^{-5})
10^{-1}	1.1779790×10^{-1}	6	1.5632498×10^{-5}
10^{-2}	1.2414228×10^{-2}	7	1.3833208×10^{-5}
10^{-3}	1.2491268×10^{-3}	8	1.2499912×10^{-5}
10^{-4}	1.2499125×10^{-4}	9	1.1584116×10^{-5}
10^{-5}	1.2499912×10^{-5}	10	1.0977844×10^{-5}
10^{-6}	1.2499991×10^{-6}	11	1.0591593×10^{-5}
10^{-7}	1.2499999×10^{-7}	15	1.0068753×10^{-5}
10^{-8}	1.2500000×10^{-8}	20	1.0003819×10^{-5}
10^{-9}	1.2500000×10^{-9}	25	1.0000186×10^{-5}
10^{-10}	$1.2500000 \times 10^{-10}$	31	1.0000004×10^{-5}

error propagation due to channel bit error of the substituted block. In computer simulations (Table 1, results on left) the total error rate with ZS,

$$P_E = 2^{-n}\,[1 - (1 - B)^n]n + B$$

where B is BER (bit error rate B in channel), is increased to approximately 1.25 times BER for n = 8, but the increase is bearable or acceptable for large n (right-hand side of Table 1; Fig. 4). Therefore, ZS is an available algorithm.

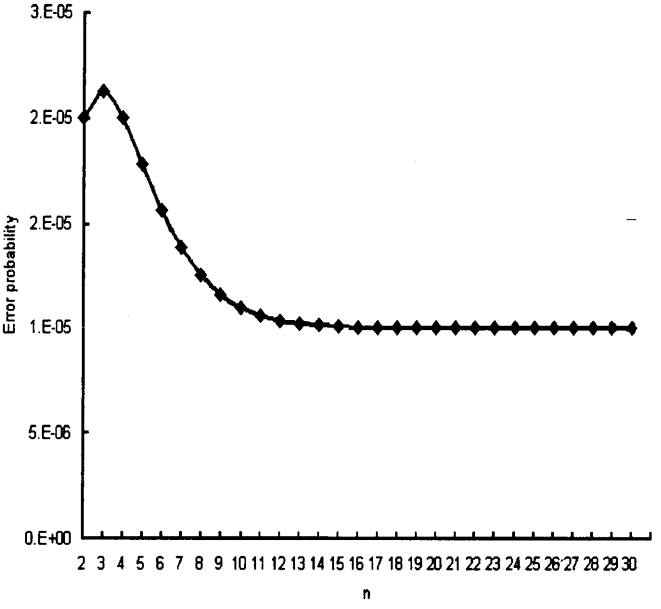

Fig. 4. Bit error propagation of ZS algorithms for variable n (BER = 10^{-5}).

3. Conclusion

The proposed ZS algorithm suppresses blocks of k or more consecutive zeros of ciphertext at the sender and recovers the original message exactly at the receiver. It is very useful algorithm for solving the difficulty of receiver clock recovery in a synchronous stream cipher which may occur due to excess zeros. By applying the ZS algorithm to a T1-carrier system for k = 15, n = 8, we can confirm improved system performance for clock recovery. The ZS algorithm does not affect cryptographic security when compared to a synchronous stream cipher without ZS.

References

1. CCITT Recommendation. Physical/electrical characteristics of hierarchical digital interface. In: CCITT red book, 1985, vol III, Rec. G.703
2. Beker HJ, Piper FC. Cipher systems: the protection of communications. Northwood Books, London, 1982
3. Daemen J, Govaerts R, Vandewalle J. Resynchronization Weaknesses in Synchronous Stream Ciphers. In: Advances in cryptology – Eurocrypto '93, Lecture Notes in Computer Science, No. 765. Springer-Verlag, 1994, pp 159–167