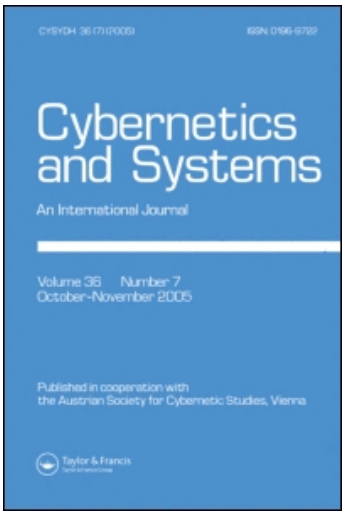


Cybernetics and Systems
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713722751

AN INTELLIGENT SECURITY AGENT FOR A RELIABLE CIPHER SYSTEM USING PINGPONG

Hoon Jae Lee [a]; Il Seok Ko (Franz Ko) [b]
[a] School of Computer and Information Engineering, Dongseo University, Busan, South Korea [b] Department of Computer Science, Dongguk University, Gyeongbuk, South Korea

Online Publication Date: 01 October 2008

To cite this Article Lee, Hoon Jae and Ko (Franz Ko), Il Seok(2008)'AN INTELLIGENT SECURITY AGENT FOR A RELIABLE CIPHER SYSTEM USING PINGPONG',Cybernetics and Systems,39:7,705 — 718
To link to this Article: DOI: 10.1080/01969720802257949
URL: http://dx.doi.org/10.1080/01969720802257949

Cybernetics and Systems: An International Journal, 39: 705–718

ISSN: 0196-9722 print/1087-6553 online
DOI: 10.1080/01969720802257949

AN INTELLIGENT SECURITY AGENT FOR A RELIABLE CIPHER SYSTEM USING PINGPONG

HOON JAE LEE[1] and IL SEOK KO (FRANZ KO)[2]

[1]School of Computer and Information Engineering, Dongseo University, Busan, South Korea
[2]Department of Computer Science, Dongguk University, Gyeongbuk, South Korea

An intelligent security agent for a reliable cipher system using a PingPong cipher is proposed. The system contains a security agent for security managements and services. The agent includes an intelligent selection of many security features: encryption processing speeds, encryption methods, encryption algorithms, synchronization methods, network interfaces, user authentications, security-bit levels, the size of synchronization patterns, etc. Furthermore, encryption algorithms with a PingPong-256, -192 and -128 for data confidentiality and a synchronization generator and detector with a SYNPAT-128, -112, -96, -80, and -64 for system stabilization are designed and analyzed.

***Keywords*: Security agent, Stream cipher, Synchronization, Pingpong, SYNPAT**

INTRODUCTION

Symmetric cryptosystems can be classified as either block ciphers or stream ciphers. A block cipher (EBC mode) divides plaintext into blocks and then encrypts each block independently, whereas a stream cipher

This research was supported by the University IT Research Center Project of Korea and by the Program for Training of Graduate Students in Regional Innovation.

Address correspondence to San 69-1 Jurye-2-Dong, Sasnag-Ku, Busan 617-716, South Korea.

encrypts plaintext on a bit-by-bit (or byte-by-byte) basis. Since a stream cipher is based on an exclusive-OR (XOR) operation, the encryption/decryption of bit-stream data bit-by-bit (or byte-by-byte) using a stream cipher is very efficient. Therefore, generally, a stream cipher is much simpler and faster than a block cipher (Schneier 1996). A block cipher is useful for software implementation; however, its performance can be degraded in a noisy channel due to its channel error propagation properties. Conversely, a stream cipher is good for high-speed enciphering and it can be implemented in noisy (wireless) channels because it does not produce error propagation. The major problem with a stream cipher is the difficulty in generating a long, unpredictable bit pattern (keystream). In the one-time pad in a stream cipher, the keystream is a sequence of randomly generated bits, and the probability that an individual bit will be 1, independent of the other bits, is equal to one-half. An ideal keystream in a one-time pad is purely random with an infinite length. Such a keystream can neither be generated by the receiving end nor distributed to the receiving end. Currently, pseudorandom bit generators are widely used to construct keystreams by generating a fixed-length pseudorandom noise. The ability to increase the length of a keystream while maintaining its randomness is crucial to the security of a stream cipher.

In general, a cipher system is required to automatically control various features, such as encryption processing speeds, encryption methods, encryption algorithms, synchronization methods, network interfaces, user authentications, security-bit levels and the size of synchronization patterns.

In this paper we propose an intelligent security agent for a reliable cipher system using a PingPong cipher (Lee and Chen 2007). This system contains a security agent for security management and services. The agent performs an intelligent selection of many security features: encryption processing speeds, encryption methods, encryption algorithms, synchronization methods, network interfaces, user authentications, security-bit levels, the size of synchronization patterns, etc. An intelligent security agent can select the features automatically: encryption processing speeds classified in low-, medium-, and high-speed encryption processing, encryption methods in block cipher and stream cipher, and encryption algorithms in PingPong-256/-192/-128, LILI-II, Dragon for stream cipher and AES, ARIA, SEED, and Camelia for block cipher. Security-bit levels classified in 128-bit key (low-level security), 192-bit key (medium-level security), and 256-bit key (high-level

security), and synchronization methods in continuous synchronization, initial synchronization, and absolute synchronization. Network interfaces are classified in ISDN, T1(DS1), E1, T3(DS3), OC-1, OC-3 and OC-12, and user authentication is in traditional password, biometrics, and hybrid. The size of synchronization patterns are classified in 128-, 112-, 96-, 80- and 64-bit SYNPAT.

Furthermore, encryption algorithms with a PingPong-256, -192 and -128 for data confidentiality and a synchronization generator and detector with a SYNPAT-128, -112, -96, -80 and -64 for system stabilization are all designed and analyzed.

CIPHER SYSTEM WITH A SECURITY AGENT

Stream ciphers can be classified into self-synchronous stream ciphers and synchronous stream ciphers. In a self-synchronous method, keystream synchronization is established autonomously based on the feedback of the ciphertext. However, one-bit errors can be propagated in the channel relative to the size of the shift register used. To eliminate this weakness, keystream synchronization in a synchronous stream cipher is reestablished by sync protocols when out-of-synchronization occurs. In this case, no bit-errors will be propagated in the channel, which is why the latter cipher is more generally used (Menezes et al. 1997). In a stream cipher, the pseudorandom binary sequences must be equal at both the transmitter and the receiver. The output keystreams should always coincide with each other, which establishes keystream synchronization. In general, keysteam synchronization methods can be classified into initial synchronization and continuous synchronization. In the initial synchronization the two keystreams are only synchronized initially at the transmitter and at the receiver, whereas in the continuous synchronization the keystreams are synchronized both initially and periodically. Keystream synchronization exchanges the synchronization patterns (SYNPAT) at the transmitter and at the receiver to initialize the starting point of the keystream cycle. The reliability of the keystream synchronization is critical to the overall system performance and communication efficiency.

The proposed synchronous stream cipher system is shown in Figures 1 and 2. In Figure 1, a network management system (NMS) communicates with each agent (cipher system) to set up the security features automatically. The function of each block in Figure 2 is as follows: block

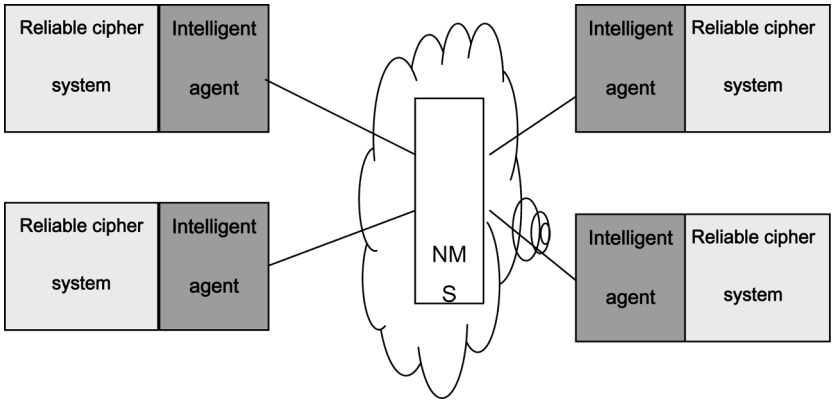
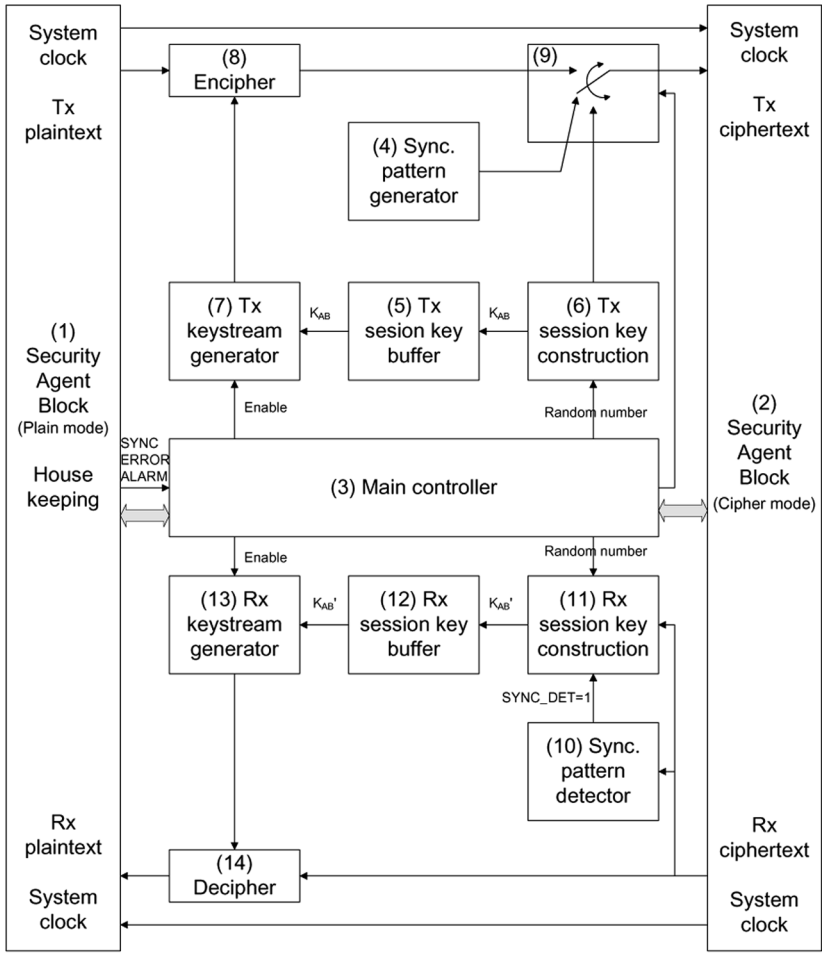


Figure 1. NMS and cipher system with agent.

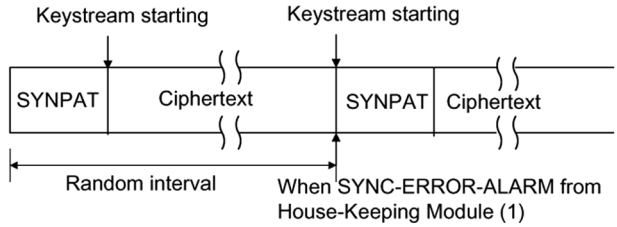
(1) is a security agent in plain mode; block (3) is the main controller for monitoring the system, and it contains the master CPU; block (4) is the synchronization pattern generator that matches the keystream sequences from the sender with those from the receiver for the initial synchronization; blocks (5) and (12) are the session-key buffers that initiate the keystream generators at the sender and receiver ends; blocks (6) and (11) are the session key constructors that distribute/construct a secure session key through a public channel from the sender to the receiver; blocks (7) and (13) are the same keystream generators with high security both at the sender and the receiver ends; blocks (8) and (14) are the enciphering process at the sender and the deciphering process at the receiver (zero suppression is required if necessary); switch (9) is the data selector for the synchronization pattern, session key, or ciphertext data; block (2) is a security agent in cipher mode; and block (10) is the synchronization pattern detector that identifies the sync pattern in the received data.

Basic Notions

In this paper we propose a highly reliable initial keystream synchronization method with a generator and a receiver. A generator part (block 4) and a detector part (block 10) of the keystream synchronization, shown in Figure 2, are used to match the keystream sequences at both the sender and the receiver ends. In the proposed scheme, the keystream synchronization exchanges the synchronization patterns (SYNPAT) to



(a) Proposed cipher system with agent.



(b) Absolute synchronization method.

Figure 2. Proposed cipher system with agent and absolute synchronization.

initialize the starting point of the two keystream cycle sequences at the sender and receiver ends.

The statistical properties are selected based on the following decision criteria (Tilborg 2000):

1. A good autocorrelation property is required.
2. The same rate must be achieved on "0" and "1" in the pattern.
3. A short run should be larger than a long run.

Accordingly, the pattern is determined based on the above criterion using a Gold sequence generator, as illustrated in Figure 3. In this figure the primitive polynomials of a 31-stage LFSR1 and LFSR2 were selected, then the N-bit pattern (here, $N = 128$, 112, 96, 80 or 64-bit SYNPAT) was generated and checked using the decision criteria. The two selected primitive polynomials and generated patterns are as follows:

- $h_1(x) = x^{31} + x^{11} + x^2 + x + 1$ (31-stage LFSR_1)
- $h_2(x) = x^{31} + x^9 + x^3 + x + 1$ (31-stage LFSR_2)
- SYNPAT-128 = 6DDA 5191 7C90 726C 7941 AD04 6ABC 8F5D (hexadecimal) if the channel bit error rate (BER) approached about 10^{-1}
- SYNPAT-112 = 6B84 E64D 6362 E90A 58A6 FDF8 C17E (hexadecimal) if the channel BER approached about 10^{-2}
- SYNPAT-96 = 321F 8D48 276E 707A 6ED7 A82F (hexadecimal) if the channel BER approached about 10^{-3}
- SYNPAT-80 = 22D6 8E48 A6B7 357E F816 (hexadecimal) if the channel BER approached about 10^{-4}

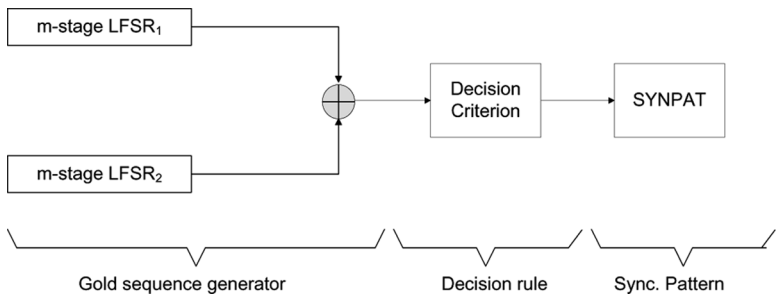


Figure 3. Synchronization pattern generator in sender ($N = 128$, 112, 96, 80 or 64).

- SYNPAT-64 = 2630 FB21 D5AD 783A (hexadecimal) if the channel BER approached about 10^{-5} to 10^{-12}

Synchronization Methods

In the OFB mode of block cipher and synchronization stream cipher, cipher synchronization plays a transmission and reception synchronization role.

Autocorrelation value of synchronization pattern is denoted as $A(t)$ and the threshold is denoted as THR. The N_T can be found in Beker et al. (1985). Autocorrelation, threshold, the number of agreements, and the number of disagreements are follows:

$$A(t) = \frac{A_g(t) - D_g(t)}{N} \tag{1}$$

$$THR = N - N_T$$

where $A_g(t) = \sum_{i=1}^{N} X(i)\Theta R(i)$: the number of agreements in bits, $D_g(t) = \sum_{i=1}^{N} X(i) \oplus R(i)$: the number of disagreements in bits, and

$$A_g(t) + D_g(t) = N,$$

wherein Θ denotes an exclusive-NOR operation, and $\oplus$ denotes an exclusive-OR operation.

In Figure 4 the synchronization pattern detector is used to calculate the truth and false within one clock as adjusting summation part in system clock. The synchronization pattern detector at the receiver consists

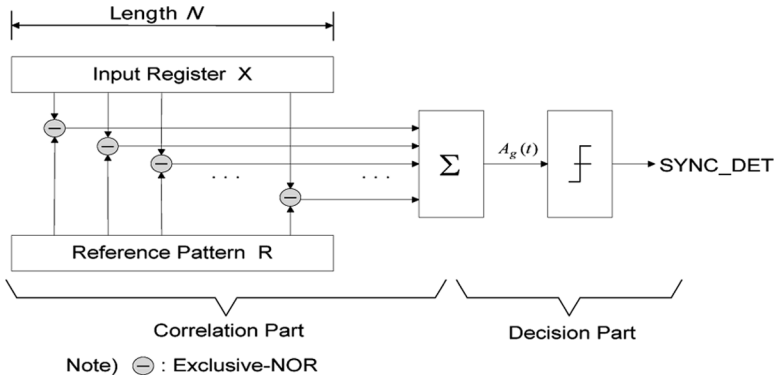


Figure 4. Synchronization pattern detector in receiver ($N = 128$, 112, 96, 80 or 64).

of a correlation part and a decision part, as shown in Figure 4. The correlation part computes the number of agreements in bits, $Ag(t)$, between an input pattern and the reference pattern (*SYNPAT*). The decisional part then compares the number of agreements in bits with a threshold, *THR*, and displays one of the following: if the number of agreements in bits is larger than the threshold, "synchronization detected ($SYNC\_DET = 1$)," otherwise, "synchronization failed ($SYNC\_DET = 0$)."

We have designed five different length SYNPATs for five reliability-level approaches. In design principles, an intelligent security agent can select five SYNPATs for an adaptive reliability-level of the communication network.

Encryption Algorithm

Encryption is divided into a stream cipher, a block cipher, and a public key algorithm. There are four modes, ECB mode (electronic codebook), CFB mode (cipher feedback), CBC mode (cipher block chaining) and OFB mode (output feedback), in a block cipher. In this chapter we consider a new design and selection problem in an encryption algorithm, which is already applied in a security agent. We have two approaches: one is the OFB mode application way—it changes a Block cipher to an alternative algorithm in order to meet the requirement in a wireless environment—and another is to apply the encryption algorithm in a Stream cipher.

Stream Cipher Algorithm. We propose the PingPong-256 (Lee and Chen 2007) and Dragon (Chen et al. 2004) for a high-level security encryption algorithm, PingPong-192 for a medium-level security algorithm, and PingPong-128, LILI-II (Clark et al. 2002), and Parallel LM (Lee and Moon 2002) for a low-level security algorithm. We have designed three PingPong algorithms in Figure 5 for three security-level approaches. In design principles, an intelligent security agent can select three algorithms for adaptive security levels of the communication network.

Block Cipher Algorithm with OFB Mode. It is desirable to apply the OFB mode of the Block cipher in a noisy, wireless network, because the channel quality would be low in the case of using an ECB mode.

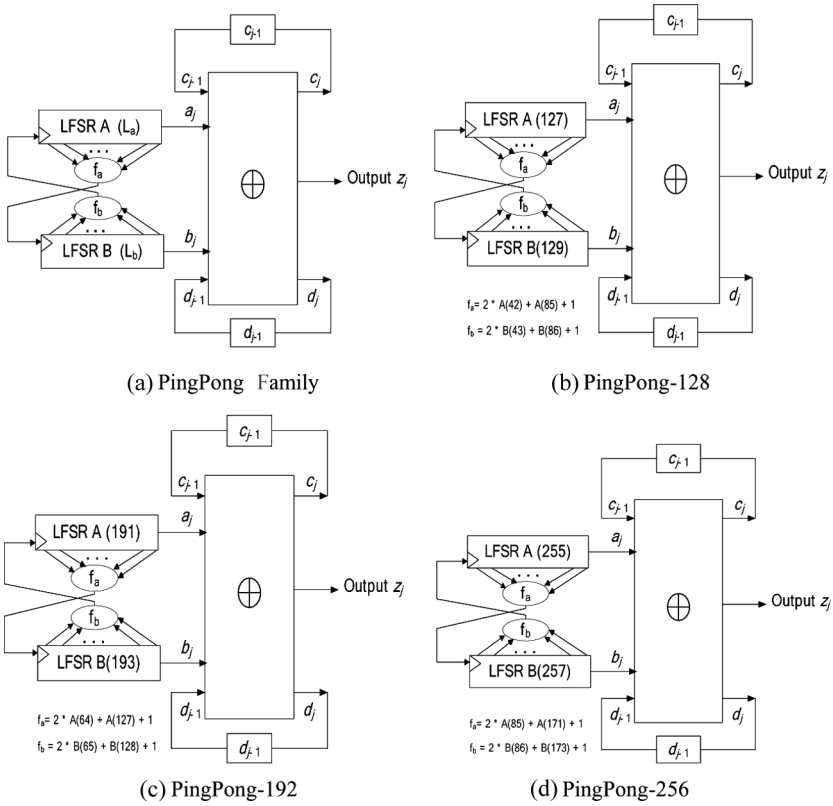


Figure 5. Pingpong family and Pingpong-128, -192, -256 for data confidentiality.

On top of that, most of the block algorithm could apply to the OFB mode. In this paper, we consider the algorithm which is to make up for the encryption weakness of the DES algorithm which has been verified internationally. As the result of that, it is possible for an FIPS-197 encryption AES algorithm (NIST 2001), a Triple-DES algorithm, an ARIA algorithm, and a SEED algorithm (KISA 1998) to be applied as shown in Table 1.

To apply the wireless communication method, we need to use the OFB mode. The applied OFB mode changes the encryption algorithm to a PN-generator and uses it, then the output block is returned to the input registry and a PN-series is generated. Finally, the encryption by plaintext and bit-by-bit XOR operation is produced.

Table 1. Algorithm proposal for application in an enhanced encryption system

Previous algorithm		Improved algorithm	
Confidentiality algorithm	Identification algorithm	Confidentiality algorithm	Identification algorithm
DES	MD5	1) Stream cipher: PingPong-256,192,128 LILI-II, Dragon, Parallel LM 2) Block cipher: AES , ARIA, SEED , T-DES,IDEA, Camelia	SHA-160, SHA-1, MD5, SEED-CBC, AES-CBC, T-DES-CBC, ARIA-CBC

An Intelligent Security Agent

In general, a cipher system is required to control various features automatically, such as encryption speed, encryption method, encryption algorithm, synchronization method, network interfaces, user authentication, security-bit level, and the size of synchronization patterns.

Table 2. Automatic selection parameters of the intelligent security agent with a reliable cipher system (Lee and Chen 2007)

Items	High-level	Medium-level	Low-level
Encryption processing speed	High-speed	Medium-speed	Low-speed
Encryption method	Block cipher	Stream cipher	
Encryption algorithm	Pingpong-256, Dragon, AES, ARIA, SEED, Camelia	Pingpong-192	Pingpong-128, LILI-II
Security-bit level	256-bit key (high-level security)	192-bit key (medium-level security)	128-bit key (low-level security)
Synchronization method	Absolute synchronization	Initial synchronization	Continuous synchronization
Network interface	OC-1, OC-3, OC-12	T1(DS1), E1, T3(DS3)	ISDN
User authentication	Hybrid	Biometrics	Traditional password
The size of synchronization patterns	128- or 112-bit	96- or 80-bit	64-bit

Table 3. Securities or performances for the intelligent security agent with a reliable cipher system (Lee and Chen 2007) (Lee 1997)

Items	Securities or Performances
Period of PingPong-256 by computing the same method in Lee and Chen (2007)	Period (PingPong-256) $\geq 2^{256} \approx 10^{77}$
Period of PingPong-192 by computing the same method in Lee and Chen (2007)	Period (PingPong-192) $\geq 2^{192} \approx 10^{58}$
Period of PingPong-128 in Lee and Chen (2007)	Period (PingPong-128) $\geq 2^{128} \approx 10^{38}$
Linear complexity of PingPong-256 by computing the same method in Lee and Chen (2007)	LC(PingPong-256) $\geq 2^{256} \approx 10^{77}$
Linear complexity of PingPong-192 by computing the same method in Lee and Chen (2007)	LC(PingPong-192) $\geq 2^{192} \approx 10^{58}$
Linear complexity of PingPong-128 in Lee and Chen (2007)	LC(PingPong-128) $\geq 2^{128} \approx 10^{38}$
SYNPAT-128, detection [probability by computing the same method in Lee (1997)	$PD = 0.9996387$
SYNPAT-128, missing probability by computing the same method in Lee (1997)	$PM = 0.36 \times 10^{-3}$
SYNPAT-128, false alarm probability by computing the same method in Lee (1997)	$PF = 0.96667 \times 10^{-12}$
Intelligent security agent	Automatically select the security features
Adaptable network interfaces	ISDN, T1(DS1), E1, T3(DS3), OC-1, OC-3, OC-12

We propose an intelligent security agent for a reliable cipher system using the PingPong cipher. The system contains a security agent for security managements and services. A network management system (NMS) communicates with each of the agents (cipher system) to set up the security features automatically. The agent includes an intelligent selection of many security features: encryption processing speed, encryption methods, encryption algorithms, synchronization method, network interfaces, user authentications, security-bit level, the size of synchronization pattern, etc.

An intelligent security agent can select the features in an automatic manner.

- Encryption processing speed classified in low-, medium, and high-speed processing
- Encryption method classified in block cipher and stream cipher
- Encryption algorithm is classified in PingPong-256/-192/-128, LILI-II, or Dragon for a stream cipher, or AES, ARIA, SEED, or Camelia for a block cipher
- Security-bit level in classified in 128-bit key (low-level security), 192-bit key (medium-level security), and 256-bit key (high-level security)
- Synchronization method is classified in continuous synchronization, initial synchronization, and absolute synchronization
- Network interface is classified in ISDN, T1(DS1), E1, T3(DS3), OC-1, OC-3, and OC-12.
- User authentication is classified in traditional password, biometrics, and hybrid.
- The size of synchronization patterns is classified in 128-, 112-, 96-, 80- and 64-bit SYNPAT.

SYSTEM PERFORMANCES

For an extremely noisy channel at $BER = 0.1$, the window size of $N = 128$, threshold $N_T = 25$, and $THR = 128 - 25 = 103$ were designed/selected as the performance parameters. A probability of false-detection in the sync. pattern $P_F = 0.96667 \times 10^{-12}$, a probability of detection $P_D = 0.9996387$, and a probability of missing $P_M = 0.36 \times 10^{-3}$ were computed using the same method as in Lee (1997). Although $BER = 0.01$ is a very noisy channel, $P_D = 1\text{–}10^{-15}$ was estimated using the same method as in Lee (1997). Therefore, the proposed system would appear to produce a highly reliable synchronization performance, even in a noisy channel, and be automatically selectable by an intelligent security agent.

Therefore, the proposed system possesses a highly secure keystream generator within a period of about 10^{77}, good randomness, an appropriate maximal linear complexity of about 10^{77}, and a maximal order correlation immunity of 1. Its processing speed is available for fast approaching DS1 class (1.544 Mbps, 647ns/1-bit interval of the system clock) through OC-12 class (622 Mbps, m-parallel cipher) for 500 MHz system clock intervals.

CONCLUSIONS

This paper proposes an intelligent security agent for a reliable cipher system using PingPong cipher. The system contains a security agent for security management and services. The agent performs an intelligent selection of many security features: encryption processing speeds, encryption methods, encryption algorithms, synchronization methods, network interfaces, user authentications, security-bit levels, the size of synchronization patterns, etc. Furthermore, encryption algorithms with a PingPong-256, -192 and -128 for data confidentiality, and a synchronization generator and detector with a SYNPAT-128, -112, -96, -80 and -64 for system stabilization are all designed and analyzed. This paper also proposed an improved initial synchronization method (referred to as *absolute synchronization*) that can be applied to noisy channels (wireless) and produces a high-performance on the probability of synchronization. In addition, a stream cipher system was proposed for the absolute synchronization of keystream synchronization, at a specified design of PingPong-256, -192 and -128 for data confidentiality. In summary, the proposed system possesses a highly secure keystream generator within a period of about 10^{77}, good randomness, an appropriate maximal linear complexity of about 10^{77}, and a maximal order correlation immunity of 1. Its processing speed is available for fast approaching DS1 class (1.544 Mbps, 647ns/1-bit interval of the system clock) through OC-12 class (622 Mbps, m-parallel cipher) for 500 MHz system clock intervals.

REFERENCES

Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., and Tokita, T. 2000. Camellia: A 128-bit block cipher suitable for multiple platforms—design and analysis. *Selected Areas in Cryptography* 39–56.

ARIA. http://www.nsri.re.kr.

Beker, H. J. and Piper, F. C. 1985. *Secure speech communications*. London: Academic Press.

Chen, K., Henrickson, M., Millan, W., Fuller, J., Simpson, A., Dawson, E., Lee, H., and Moon, S. 2004. Dragon: A fast word based stream cipher. *LNCS* 3506 (ICISC'2004): 414–431.

Clark, A., Dawson, E., Fuller, J., Golic, J., Lee, H., Millan, W., Moon, S., and Simpson, L. 2002. The LILI-II keystream generator. *LNCS* 2384 (ACISP'2002): 25–39.

Diffie, W. and Hellman, M. E. 1976. New directions in cryptography. *IEEE Trans. On Infor. Theory* IT-22(6): 644–654.

KISA. 1998. Development of SEED, the 128-bit block cipher standard, http://www.kisa.or.kr.

Lee, H. 1997. An improved synchronous stream cipher system for a link encryption. PhD diss., KyungPook National University, Daegu, Korea.

Lee, H. and Chen, K. 2007. PingPong-128, A new stream cipher for ubiquitous application. *IEEE CS (ICCIT 2007)* 1893–1899.

Lee, H. and Moon, S. 2002. Parallel stream cipher for secure high-speed communications. *Signal Processing* 82(2): 259–265.

Lee, H. and Moon, S. 1998. A Zero-suppression algorithm for the synchronous stream cipher. *Applied Signal Processing* 5(4): 240–243.

Menezes, A., Oorschot, P., and Van Vanstone, S. 1997. *Handbook of applied cryptography.* New York: CRC Press.

Moon, S. and Lee, P. 1990. A proposal of a key distribution protocol. Paper presented at the Proceedings of the Korean Workshop on Information Security and Cryptography-WISC'90, 117–124.

NIST. 2001. Announcing the advanced encryption standard (AES). FIPS-197.

Park, B., Choi, H., Chang, T., and Kang, K. 1993. Period of sequences of primitive polynomials. *Electronics Letters* 29(4): 390–391.

Proakis, J. 1995. *Digital Communications* (3rd Ed.). New York: McGraw-Hill, Inc.

Rueppel, R. A. 1986. *Analysis and design of stream ciphers*. Springer-Verlag: Berlin.

Schneier, S. 1996. *Applied cryptography: protocols, algorithms, and source code in C* (2nd Ed.). New York: John Wiley and Sons, Inc.

Tatebayashi, M., Matsuzaki, N., and Newman, D. B. 1990. A cryptosystem using digital signal processors for mobile communication. *ICASSP'90* 37.1.1–37.1.4.

Tilborg, H. C. A. 2000. *Fundamentals of cryptology.* Kluwer Academic Publishers. Boston.