通信情報保護學會論文誌
第 8 卷 第 2 號, 1998. 6

자기 상관기를 이용한 개선된 키 수열 동기 방식

이 훈 재*

An Improved Keystream Synchronization using Autocorrelator

Hoon Jae Lee*

요 약

본 논문에서는 스트림 암호 구현 시 하드웨어 설계가 용이한 고속 자기 상관기와 이를 이용한 개선된 키 수열 동기 방식을 제안하였다. 제안 방식은 키 수열 동기시 잡음이 많은 무선 채널에서도 동기를 유지할 수 있는 고속, 고신뢰도 초기 키 수열 동기 방식이며, 기존 방식보다 복잡도를 크게 줄여서 하드웨어 구현이 용이하도록 하였다.

Abstract

In this paper we propose a high speed autocorrelator which is easily implementable and which gives an improved keystream synchronization method. The proposed method is a high-speed and a highly reliable initial synchronization technique in a noisy channel, and the hardware it uses is less complex than hardware currenthy being used.

Keyword: Correlator, 키 수열 동기, 동기 패턴 발생, 동기 패턴 검출

Ⅰ. 서 론

디지털 통신에서 송수신 동기를 위한 동기 패턴 검출기로는 자기 상관기(autocorrelator)를 일반적으로 많이 사용한다. 특히 확산 대역 통신에서 의사 난수 발생기(pseudo-random generator)를 초기화시키거나 또는 스트림 암호[4]-[6]에서 키 수열 동기(keystream synchronization)를 확립하기 위해서는 동기 패턴 검출 회로가 반드시 필요하다. Dixon[1]은 확산 대역 통신에서 의사 난수 발생기를 초기화하고자 자기 상관기를 이용하였다. Beker등[2]은

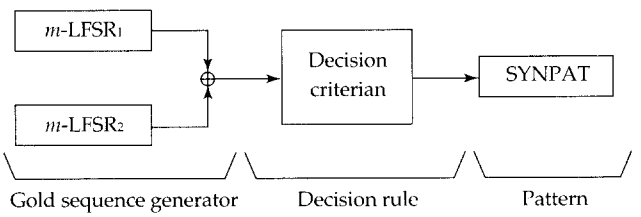
* 경운 대학교 컴퓨터공학과

동기식 스트림 암호(synchronous stream ciphers)의 수신단에서 동기 패턴을 검출하고자 자기 상관기를 이용한 동기 방식을 제안하였다. 그러나 이들 방식은 고속 및 고신뢰도 통신이 요구되는 최근의 통신 회로에서는 적용이 어렵다. 특히 잡음이 많은 무선 채널등에서 통신 신뢰성 강화를 위하여 자기 상관기의 크기(이동 레지스터 단수)를 늘릴 경우 하드웨어 복잡도가 크게 증가 되어 구현이 어렵게 된다. 왜냐하면 이동 래지스터의 단수(N)가 커질수록 구현 복잡도는 기하 급수적으로 증가하기 때문이다. 또한, Leibowitz[3]는 TDM 다중화된 자기 상관기(time-division multiplexed digital correlator)를 제안하여 k-배 만큼 속도를 개선시켰지만 고속 처리에만 치중하여 하드웨어 복잡도는 기존 방식보다도 k-배 이상 증가되었다.

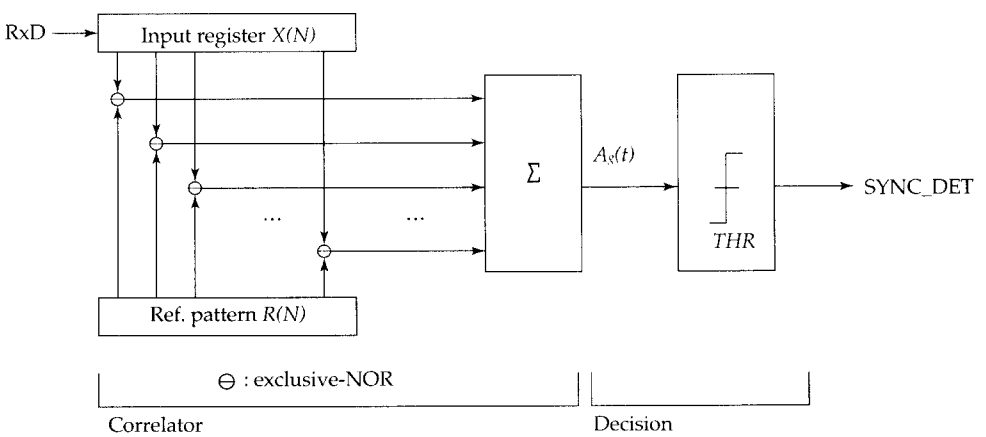
본 논문에서는 N의 증가에 대하여 하드웨어 복잡도가 거의 선형적으로 증가되는 새로운 자기 상관기를 제안한다. 즉, Beker 모델의 자기상관기를 개선하여 하드웨어 구현이 간단하면서도 고속, 고신뢰도 기본 요구사항을 만족시킬 수 있는 3단계 모델형을 제안한다. 그리고 이를 이용한 키 수열 동기 방식을 제안하여 그 특성을 분석한다.

II. 자기 상관기와 키 수열 동기

키 수열 동기부는 동기식 스트림 암호에서 송 수신 키 수열을 일치시키는 역할을 하는데, 일반화된 모델[7, 8]은 그림 1과 같다.



a) Synchronization pattern generator.



b) Synchronization pattern detector.

그림 1. 키 수열 동기

Fig. 1. Keystream synchronization.

동기 패턴에 대한 자기 상관값 $A(t)$, 문턱값 THR, N_T는 다음과 같다.

$$\begin{cases} A(t) = \dfrac{A_g(t) - D_g(t)}{N} \\ THR = N - N_T \end{cases}$$

여기서, $Ag(t) = \sum_{i=1}^{N} X(i) \ominus R(i)$는 일치 비트수

$Dg(t) = \sum_{i=1}^{N} X(i) \oplus R(i)$는 불일치 비트수

$$Ag(t) + Dg(t) = N$$

그림 1 b)의 동기 패턴 검출기는 합산 부분을 시스템 클럭에 맞추어 1 클럭만에 옮고 그 름을 계산해 내어야 하므로 이를 하드웨어로 구현하기에는 너무 복잡해진다. N이 커질수록 그 복잡도는 기하 급수적으로 커지며, 이를 구현하는 방법은 다음 3가지로 생각해 볼 수 있다.

첫째, N 배율 클럭 방법은 그림 2와 같이 수신 데이터(RxD)의 1 비트 입력에 대하여 시스템 클럭 ø(t) 보다 N배 더 빠른 클럭으로 $X(N)$과 $R(N)$을 회전시키면서 두 패턴을 비교 검사하는 방법이다. 이 방법은 고속 통신 클럭의 N배 클럭이 필요하므로 N값이 클수록 구현이 힘들다.

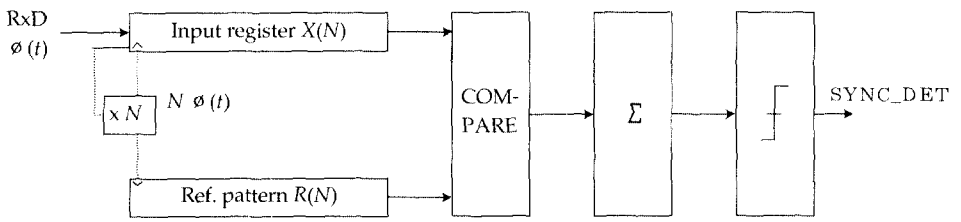


그림 2. N 배율 클럭 방법

Fig. 2. Method of N times system clock configuration.

둘째, 테이블 읽기 방법은 그림 3과 같이 수신 데이터(RxD)가 1 비트 입력될 때마다 레지스터 X(N) 값으로 테이블 번지를 지정하여 동기 패턴 또는 유사 패턴이면 "1"을, 아니면 "0"을 읽도록 하는 방법이다. 이 방법은 가능한 모든 메모리 번지에 대하여 동기 패턴 여부를 판단할 수 있도록 테이블을 미리 생성해야 하며, N이 클수록 메모리 번지 수는 2의 지수승으로 증가하기 때문에 이에 적합한 메모리를 구성하기 어렵다.

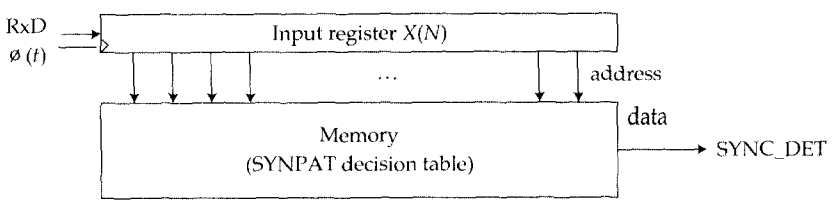


그림 3. 테이블 읽기 방법

Fig. 3. Method of table-read configuration.

셋째, 조합 논리(combinatorial logic) 구성 방법은 그림 4와 같이 수신 데이터가 1 비트 입력될 때마다 조합 논리 회로를 이용하여 입력 레지스터의 값이 동기 패턴인지 여부를 판단하는 방법이다. 이 방법 역시 N이 클수록 (예로써, N=128) 하드웨어 복잡도는 기하 급수적으로 증가된다.

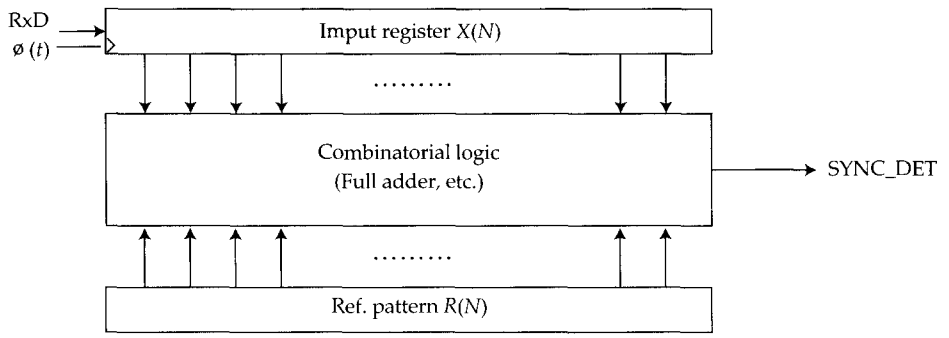


그림 4. 조합 논리 구성 방법

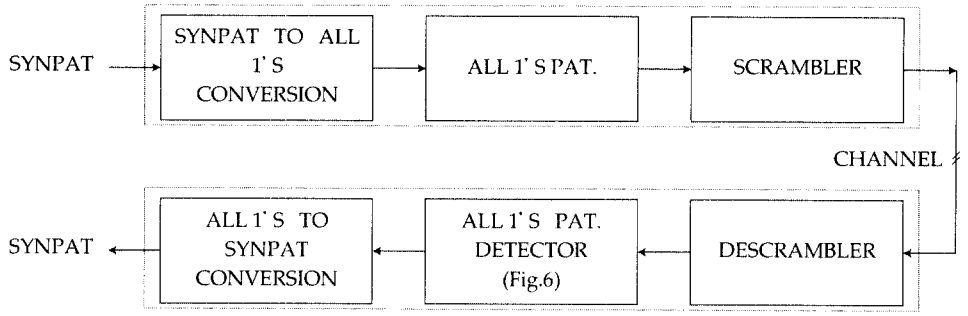
Fig. 4. Method of combinatorial logic configuration.

이상에서 살펴본 바와 같이 기존의 자기 상관기를 이용한 키 수열 동기방식(패턴 발생기, 송신기 및 검출기 등)에서는 통신 신뢰성을 높이고자 동기 패턴의 길이(N)를 증가시키면 하드웨어의 복잡도가 기하급수적으로 증가되는 문제점을 안고 있다(구현 복잡도 $O(N^2)$ 정도로 예상됨).

III. 자기 상관기 개선

일반적인 동기 패턴은 랜덤한 값으로 구성되어 있기 때문에 동기 패턴 검출기는 N이 커질수록 구현이 더 복잡해진다. 그러므로 랜덤한 동기 패턴을 단순 패턴으로 변환시켰다가 다시 역변환시키면 하드웨어 복잡도를 줄일 수 있음을 알 수 있다. 즉, 임의의 동기 패턴에 대하여 모든 비트가 "1"(all 1's pattern)이나 "0"(all 0's pattern), "10" 반복 또는 "1100" 반복 등과 같은 단순 패턴으로 변환한 후 일치 비트 수(number of agreement bits)를 구하면 의외로 하드웨어가 간단해진다. 다만 단순 패턴 그 자체는 자기 상관성이 낮아서 동기 패턴으로 적합치 않으므로 자기 상관성이 우수한 동기 패턴을 생성한 후 단순 패턴으로 변환시키는 과정이 별도로 필요하다. 또한 단순 패턴은 송신시 선로 부호화상에서 문제(연속 "0" 또는 연속 "1"로 인한 모뎀 클럭 복구 문제등)가 될 수 있기 때문에 이를 방지코저 간단한 스크램블러와 디스크램블러를 추가하여야 한다. 이를 도시한 것이 그림 5이다.

그림에서 송신단의 스크램블러는 전 비트 "1" 패턴을 랜덤한 패턴으로 변경시켜 주고 수신단의 디스크램블러는 그 역 과정에 해당하며, 이들은 선로 특성에 맞게 간단히 구현될 수 있다. 또한 수신단에서의 전 비트 "1" 패턴 검출기(all 1's pattern detector)를 자세히 나타



SYNPAT = 6DDA 5191 7C90 726C 7941 AD04 6ABC 8F5D (hexa)

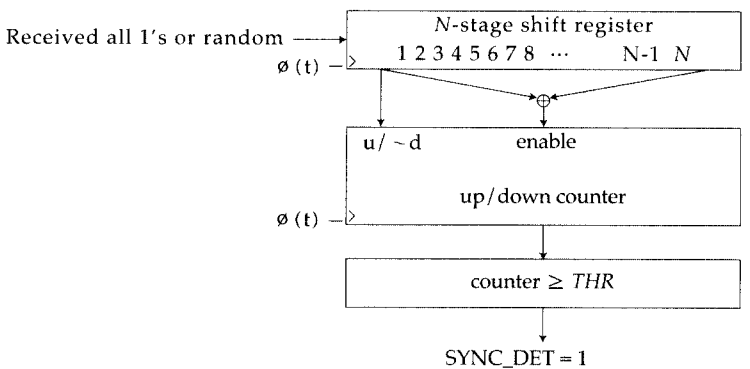
그림 5. 제안된 단순 패턴을 이용한 키수열 동기 방식

Fig. 5. Proposed synchronization using simple pattern.

낸 것이 그림 6이다. 그림에서 전 비트 "1" 패턴 또는 랜덤 패턴이 N단 레지스터에 입력되면 업-다운 카운터는 레지스터가 보유한 "1"의 갯수를 카운트한다. 즉, 이동 레지스터의 첫째 단에 "1"이 입력되면 카운터는 증가되고, 최종 단에서 "1"이 출력되면 카운터는 감소되기 때문에 이 카운터는 이동 레지스터에 포함되어 있는 "1"의 갯수를 항상 보유하게 된다. 출력 단에서는 카운터 값을 문턱 값(N_T)과 비교하여 그 보다 크면 동기 검출 사실(SYNC-DET=1)을 알려준다.

IV. 시뮬레이션 및 분석

선택된 동기 패턴(그림 5의 128 비트 SYNPAT)에 대하여 자기 상관 특성, "0"-"1" 분포 특성, run 특성 및 동기 확률을 계산하면 다음과 같다.



shift register		counter
1	N	
0	0	disable
0	1	down(enable)
1	0	up(enable)
1	1	disable

그림 6. 전 비트 "1" 패턴 검출기

Fig. 6. All 1's pattern detector.

1. 자기 상관 특성

동기패턴에 대한 자기 상관값은 식 (1)과 같이 계산하며, 그 결과를 요약하면 표 1 및 그림 7과 같다. 선택된 동기 패턴의 일치비트 수 $Ag(t)$의 1차 피크 값과 2차 피크값의 차이가 60(= 128-68)이 됨으로서 문턱값 THR은 98(= 128-60/2) 보다 큰 값 중에서 선택하면 된다. 그러므로 $N = 128$이고, $N_T = 25$일 때 $THR = 128-25 = 103$을 선택하였다.

〈표 1〉 동기 패턴의 일치 비트 수에 대한 분포

〈Table 1〉 Distribution of no. of agreement bits in SYNPAT.

No. of agreement bits Ag(t)	Distribution	Remarks
54	1/64	minimum
58	3/64	
60	5/64	
62	35/128	
64	3/16	
66	3/8	
68	1/64	2nd peak
128	1/128	1st peak

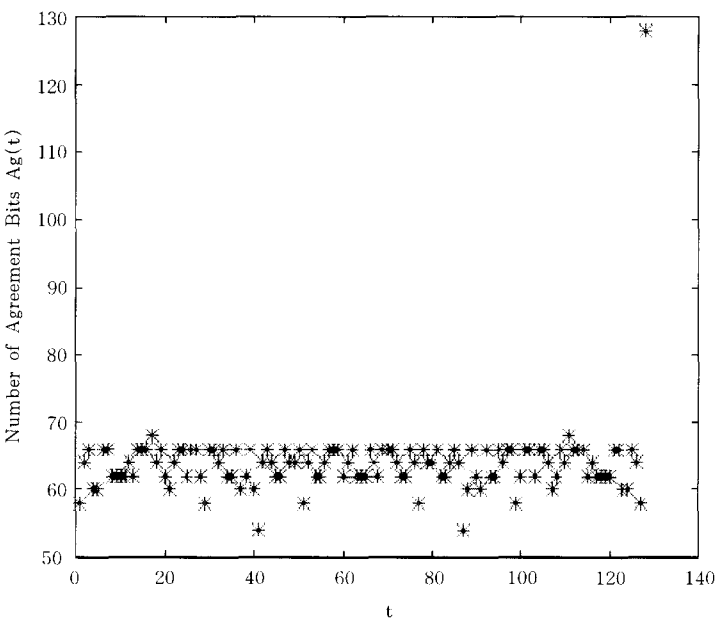


그림 7. 동기 패턴의 일치 비트 수에 대한 분포

Fig. 7. Distribution of no. of agreement bits in SYNPAT.

2. "0"-"1" 분포 특성

"0"의 확률분포 p(0)와 "1"의 확률분포 p(1)을 계산하면, p(0) ≒ p(1) ≒ 0.5 이므로 선택된 동기 패턴의 "0"-"1" 분포특성이 양호함을 알 수 있다.

3. Run 특성

연속된 "1"을 run이라 하며, 연속된 "1"의 개수를 run 크기라 한다. 총 run의 수 = 35이고, run 크기 i에 따른 확률분포 $p_{run}(i)$ ≒ $1/2^i$, $(1 \le i \le 6)$ 이므로 선택된 동기 패턴의 run 특성이 양호함을 알 수 있다.

4. 동기 확률

검출 window N, 채널 비트 오류율 B, 전송속도 R bps하에서 N비트 동기 신호 송출시 문턱 값 $N_T(0 \le N_T \le N)$라 하면, 에러 개수 i에 대한 동기 검출 확률 밀도 함수 p_{Di}와 동기 검출 확률 P_D, 그리고 미검출 확률 P_M은 다음과 같다[2, 7].

$$P_{Di} = {}_NC_i\, B^i\,(1-B)^{N-i},\ i=0,1,\cdots,N \quad (2)$$

$$P_D = \sum_{i=0}^{N_T} P_{Di} = \sum_{i=0}^{N_T} ({}_NC_i\, B^i (1-B)^{N-i}) \quad (3)$$

$$P_M = 1-P_D \quad (4)$$

한편 동기 신호를 전송하지 않아도 채널에서의 랜덤 잡음에 의해서 동기신호로 오인될 수 있으므로 이를 오검출(false detection)이라 하며, 에러 수 i에 대한 오검출 확률 밀도 함수 p_{Fi}와 오검출 확률 P_F, 그리고 평균 오검출 시간 T_F는 아래와 같다.

$$P_{Fi} = {}_NC_i\, 0.5^i\,(1-0.5)^{N-i} = {}_NC_i\, 2^{-N} \quad (5)$$

$$P_F = 2^{-N} \sum_{i=0}^{N_T} {}_NC_i \quad (6)$$

$$T_F = \frac{1}{P_F \cdot R} \quad (7)$$

N_T 가변에 따른 동기 확률 밀도 및 동기 확률을 계산하면 표 2, 표 3 및 그림 8과 같고, BER 가변에 따라 계산하면 표 4와 같다. 표 3에서는 $BER = 0.1$에서의 동기확률이 $P_F = 0.97 \times 10^{-12}$, $P_D = 0.9996387$, $P_M = 0.36 \times 10^{-3}$ 이며, 표 4에서는 $BER = 0.01$에서의 동기검출확률이 $P_D = 1-10^{-13}$ 이상이 됨에 따라 동기 신뢰성이 충분히 높음을 알 수 있다.

5. 하드웨어 속도

제안 방식을 구현시 하드웨어 속도에 민감한 부분은 그림 6이며, 그림 6에서 동기식 하드웨어(시스템 클럭에 맞추어 하드웨어가 동작되도록 설계)로 구현

〈표 2〉 N_T가변에 따른 동기 확률 밀도(BER = 10^{-1})
〈Table 2〉 Sync. probability density for variable N_T(BER = 10^{-1}).

N_T	P_F	P_D
0	2.938735877055719e-39	1.390084229174165e-06
1	3.761581922631320e-37	1.977008714225174e-05
2	2.388604520870888e-35	1.394889504798287e-04
3	1.003213898765773e-33	6.509484463501870e-04
4	3.135043433643041e-32	2.260237698360556e-03
5	7.774907715434741e-31	6.228210649713053e-03
10	6.666409318632301e-25	9.043728470698699e-02
15	3.884042269407742e-20	8.923330428107028e-02
20	3.516394320263388e-16	1.368132268217058e-02
21	1.808431364706885e-15	7.817898804965787e-03
22	8.795552546528941e-15	4.224824171623192e-03
23	4.053602477965512e-14	2.163436568148469e-03
24	1.773451084109911e-13	1.051670571373409e-03
25	7.377756509897232e-13	4.861055165943051e-04
26	2.922647386613134e-12	2.139695258042562e-04
27	1.104111234942740e-11	8.981437034315019e-05
28	3.982686954614882e-11	3.599702997956143e-05
29	1.373340329177545e-10	1.379196573776196e-05
30	4.532023086285900e-10	5.057054187574915e-06

〈표 3〉 N_T 가변에 따른 동기 확률($BER = 10^{-1}$)

〈Table 3〉 Sync. probability for variable N_T($BER = 10^{-1}$).

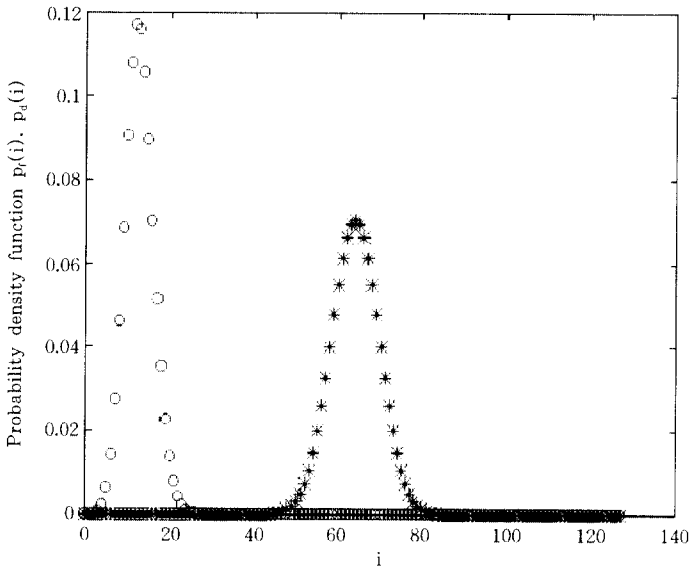
N_T	P_F	P_D	P_M
0	2.938735877055719e-39	1.390084229174165e-06	9.999986099157708e-01
1	3.790969281401877e-37	2.116017137142591e-05	9.999788398286286e-01
2	2.426514213684907e-35	1.606491218512547e-04	9.998393508781488e-01
3	1.027479040902622e-33	8.115975682014417e-04	9.991884024317985e-01
4	3.237791337733303e-32	3.071835266561997e-03	9.969281647334380e-01
5	8.098686849208071e-31	9.300045916275050e-03	9.906999540837249e-01
10	7.271572314915841e-25	2.559625999178171e-01	7.440374000821830e-01
15	4.465076540551982e-20	7.912163018922382e-01	2.087836981077618e-01
20	4.294311477937454e-16	9.838947814335587e-01	1.610521856644131e-02
21	2.237862512500631e-15	9.917126802385244e-01	8.287319761475564e-03
22	1.103341505902957e-14	9.959375044101476e-01	4.062495589852388e-03
23	5.156943983868469e-14	9.981009409782960e-01	1.899059021703953e-03
24	2.289145482496759e-13	9.991526115496695e-01	8.473884503304996e-04
25	9.666701992393990e-13	9.996387170662638e-01	3.612829337361623e-04
26	3.889317585852534e-12	9.998526865920681e-01	1.473134079319482e-04
27	1.493042993527993e-11	9.999425009624112e-01	5.749903758878183e-05
28	5.475729948142875e-11	9.999784979923908e-01	2.150200760919763e-05
29	1.920913323991833e-10	9.999922899581286e-01	7.710041871389350e-06
30	6.452936410277733e-10	9.999973470123162e-01	2.652987683759989e-06

〈표 4〉 BER 가변에 따른 동기 확률($N_T = 25$)

〈Table 4〉 Sync. probability for variable BER($N_T = 25$).

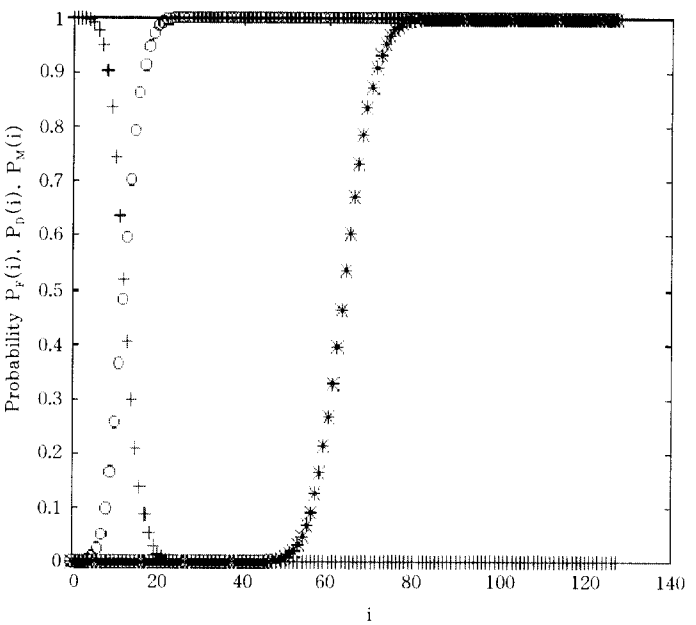
BER	P_F	P_D	P_M
10^{-1}	$9.666701992 \times 10^{-11}$	0.9996387171	$3.612829337 \times 10^{-2}$
10^{-2}	$9.666701992 \times 10^{-11}$	1.0000000000	0.000000000
10^{-3}	$9.666701992 \times 10^{-11}$	1.0000000000	0.000000000
10^{-4}	$9.666701992 \times 10^{-11}$	1.0000000000	0.000000000
10^{-5}	$9.666701992 \times 10^{-11}$	1.0000000000	0.000000000

할 경우 이동 레지스터나 업-다운 카운터의 처리 속도는 10~20ns이면 시간 여유가 충분하다. 그러므로 이 회로는 T1 속도(1.544 MHz 클럭, 648ns)에서 데이터 처리가 가능할 뿐만 아니라 50 MHz 클럭(20ns) 속도에서도 충분히 적용할 수 있을 만큼 고속 처리가 가능하다.



* : $p_F(i)$, o : : $p_d(i)$

a) 동기확률 pdf 분포



* : $P_F(i)$, o : $P_D(i)$, + : $P_M(i)$

b) 동기 확률

그림 8. 동기 확률 밀도 및 동기 확률

Fig. 8. Sync. probability density and sync. probability.

VI. 결 론

본 논문에서는 N의 증가에 대해서 하드웨어 복잡도가 거의 선형적으로 증가하는 새로운 자기 상관기 구현 방법과 이를 이용한 개선된 키 수열 동기방식을 제안하였다. 제안된 동기방식은 랜덤한 동기 패턴을 단순 패턴으로 변환시켜 수신단에서 동기 검출기(자기상관기)를 간단히 구현한 다음 검출 신호를 다시 원래 신호로 역변환시켜 원하는 성능을 얻도록 하였으며, 단순 패턴의 채널 적응력을 키우기 위하여 간단한 스크램블 및 디스크램블 회로를 송수신단에 추가시킨 3단계 모델이다. 제안된 동기방식을 분석한 결과 하드웨어로 간단히 구현 가능하면서 고속, 고신뢰도 기본 요구사항을 만족시킬 수 있음을 알 수 있었다. 즉, 하드웨어 구현시 1.544 Mbps급 이상에서도 고속 통신을 위한 구현이 가능하며 이 때 회로가 비교적 단순해진다. 또한 BER=0.1에서도 동기 확률이 0.9996 이상이고 오검출 확률이 10-12이하인 고신뢰도 통신 동기가 가능하다.

참 고 문 헌

[1] R. C. Dixon, Spread Spectrum Systems, Wiley, New York, 1976.

[2] H. J. Beker and F. C. Piper, Secure Speech Communications, Academic Press, London, 1985.

[3] L. M. Leibowitz, "Multiplexing Techniques for Digital Correlator Speed Improvement," IEEE Trans. on Comm. Vol. COM-33, No. 6, pp. 579-588, June 1985.

[4] B. Schneier, Applied Cryptography : Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc., New York, USA, 1994.

[5] D. R. Stinson, Cryptography: Theory and Practice, CRC Press, Inc., N.W., Boca Raton, Florida, 1995.

[6] G. J. Simmons(Ed.), Contemporary Cryptology: The science of Information Integrity, IEEE Press, Inc., New York, 1992.

[7] 이훈재, 문상재, "고신뢰도 동기식 스트림 암호 시스템," 한국정보보호학회논문지, 1998년 3월호 게재.

[8] 이훈재, "링크 암호에 적합한 개선된 동기식 스트림 암호 시스템," 경북대학교 박사학위논문, 1997년 12월.

□ 著者紹介

이 훈 재

1985년 2월 경북대학교 공과대학 전자공학과(전자공학, 공학사)
1987년 2월 경북대학교 대학원 전자공학과(통신공학, 공학석사)
1987년 2월 - 1998년 1월 국방과학연구소 선임연구원
1993년 3월 - 1998년 2월 경북대학교 대학원(정보통신, 공학박사)
1993년 3월 - 현재 경운대학교 컴퓨터공학과(전임강사)

※ 주관심분야 : 정보보호기술, 디지털 통신, 정보통신망