고속 스트림 암호 ASC16

김길호* · 송홍복** · 김종남* · 조경연*

*부경대학교 컴퓨터공학과

**동의대학교 전자공학과

Fast Stream Cipher ASC16

Gil-Ho Kim* · Hong-Bok Song** · Jong-Nam Kim* · Gyeong-Yeon Cho*

*Dept of Computer Engineering Pu-Kyong National University

**Dept of Electronics Engineering Dong-Eui University

E-mail : vnlqpcdd@hanmail.net

요 약

소프트웨어 구현을 위한 고속 스트림 암호 ASC16을 제안한다. ASC16은 ASR(Arithmetic Shift Register), NLF(Non-Linear Filter), NLB(Non-Linear Block)로 매우 간결한 구조를 이루고 있으며, 워드 단위로 연산을 수행하고, 비선형변환으로 S-박스를 사용하여 32비트 키 스트림을 만드는 무선 통신용 스트림 암호이다. Zhang, Carroll 그리고 Chan에 의해 개발된 32비트 출력 스트림 암호 SSC2와 수행 결과 비교에서 거의 동등한 결과를 보였고, 주기는 SSC2보다 더 길어 졌으며, 상관공격(Correlation attack)이 어려워 안전성은 더욱 향상 되었다. 제안한 ASC16은 무선통신 등과 같은 제한적인 환경에서 고속 암호 수행에 유용하게 사용될 수 있다.

ABSTRACT

We propose a fast stream cipher ASC16 for software implementation. ASC16 has a very simple structure with ASR(Arithmetic Shift Register), NLF(Non-Linear Filter), and NLB(Non-Linear Block), and is executed by a word. It is a stream cipher for wireless communication, which makes 32bit key streams using s-box with non-linear transformation. The processed result is almost same as SSC2, 32bit output stream cipher, developed by Zhang, Carroll, and Chan. The period is longer than SSC2, and it causes the difficulty of Correlation attack and raises security very much. The proposed ASC16 is efficiently used in the process of a fast cipher in the limited environment such as wireless communication.

키워드

SSC2, correlation attack, ASR, Non-Linear Filter, S-box

Ⅰ. 서 론

1990대 이후 암호화 기술이 일반화되고 인터넷을 통한 대용량의 멀티미디어 데이터 전송의 증가로 인한 빠른 암호 알고리즘의 개발이 필요했다. 일반적으로 스트림 암호는 블록 암호(Block Cipher)보다 5-10배 정도 빠르게 실행되는 장점을 가진다. 그리고 스트림 암호는 비트단위로 암호화하므로 에러 전파(error propagation) 현상이 없다. 최근에는 스트림 암호 또한 블록 암호와 마찬가지로 블록 단위로 키를 생성하여 암호화하는 방식이 널리 사용되고 있다. 1990년대 후반 소프트웨어 구현이 용이한 스트림 암호가 등장하기 시작 했고, 특히 2000년대에 유럽의 NESSIE(New European Schemes for Signatures, Integrity, and Encryption)[1], ecrypt Ⅱ[2], 일본의 CRYPTREC(Cryptography Research and Evaluation Committees)[3] 등의 국제적인 암호

공모사업으로 스트림 암호도 공모되어 여러 종류의 새로운 스트림 암호가 제안되었다.

Carroll, Chan 그리고 Zhang[4]가 제안한 32비트 스트림 암호 SSC2는 소프트웨어 지향의 무선 통신용 고속 스트림 암호이다. 그러나 SSC2는 Hawkes, Rose 그리고 Quick[5]의 논문에 의하면 LFG(Lagged Fibonacci Generator)[6]의 짧은 주기와 상관관계 분석(Correlation Analysis)[7]을 통해 현대 암호에서 필요로 하는 안전성을 만족시키지 못한다고 주장했다.

본 논문에서는 ASR(Arithmetic Shift Register)[8], NLF(Non-Linear Filter), NLB(Non-Linear Block)로 구성된 32비트 출력의 새로운 스트림 암호 ASC16를 제안한다. ASC16은 소프트웨어 및 하드웨어 구현이 쉽게 디자인된 스트림 암호 알고리즘이다. 특히 계산능력이 제한된 무선 통신장비에서 빠르게 수행할 수 있도록 개발되었다. ASC16은 다양한 길이(8-32바이트)의 키를 지원하고 있으며, 워드 단위로 연산을 수행한다. ASC16는 매우 간결한 구조를 가지고 있으며 선형 궤환 순서기(Linear Feedback Sequencer)로 ASR을 적용하였고, 비선형 순서기(Nonlinear sequencer)로 NLB와 비선형 순서 발생기는 NLF로 구성되어 있는 결합 함수(combining function)[9] 스트림 암호이다. 그리고 8비트, 16비트, 32비트 프로세스에서 쉽게 구현이 가능하다.

제안한 스트림 암호 ASC16는 SSC2와 수행시간 테스트에서 거의 동등한 결과를 보여주고 있으며, SSC2보다 주기는 더 긴 약 2^{57}의 주기를 갖고 있고, 상관공격(Correlation attack)이 어려워 안전성은 더욱 향상되었다.

본 논문의 구성은 2장에서 ASC16을 구조적으로 구성된 부분들을 상세히 설명하고, 3장에서 ASC16의 안전성에 대해 설명하고 4장에서 ASC16을 소프트웨어 구현과 수행 결과를 설명한 후 결론으로 끝맺는다.

II. ASC16 상세 설명

스트림 암호 ASC16은 선형 궤환 순서기 ASR과 비선형 순서기 NLB, 그리고 비선형 순서 발생기(Non-Linear Filter)로 구성한다. ASR은 전체 151비트로 5개의 워드로 구성되어 있으며, 마지막 워드는 23비트만 사용한다. NLB는 16개의 워드로 구성되어 있으며 PG(Point Generator)에 의해 발생된 2개의 포인터 변수를 이용하여 NLF에 의해 순차적으로 워드단위로 업데이트 된다. 마지막으로 비선형 필터는 ASR의 32비트 3개의 워드와 NLB의 15개의 워드를 이용하여 32비트 출력을 생성한다. (그림 1)은 ASC16의 키 생성과정의 전체적인 흐름을 그림으로 표현한 것이다. ASC16의 최종적인 키 스트림의 출력은 NLF의 32비트 출력과 ASR의 출력을 XOR연산을 통해서 최종적인 32비트 키 스트림이 생성된다.

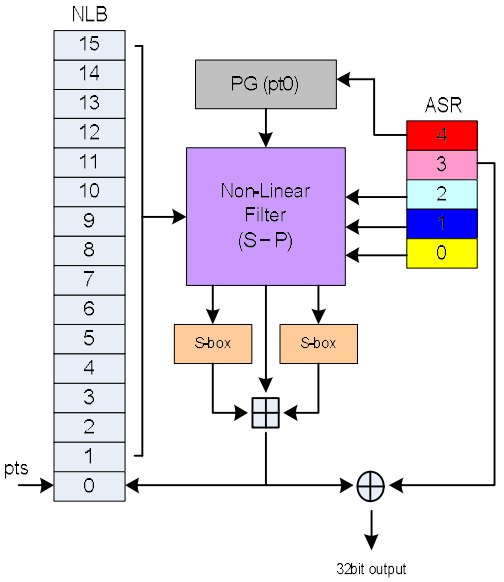

그림 1. ASC16의 키 스트림 생성도

2.1 초기화

ASC16은 16-32바이트의 가변적인 키를 지원하고 사용자가 입력한 시드 키(Seed key)는 해시 알고리즘을 사용하여 128비트 또는 256비트 마스터 키와 64비트 또는 128비트 초기벡터(IV)를 가지고 ASR의 5개의 워드와 NLB의 16개의 워드를 초기화 시킨다. 초기화 과정은 128비트 마스터 키와 64비트 IV를 가지고 설명한다.

총 21개 워드의 메모리를 할당받고 그 중 16개 워드를 0, 5, 10, 15번째 워드는 고정된 상수로 각각 16진수 0x30fb40d4, 0x9fa0ff0b, 0x6beccd2f, 0x3f258c7a로 셋팅한다. 그리고 1, 2, 3, 4번째 워드는 128비트 마스터 키 값이 순서대로 적용되고, 6, 7, 8, 9번째 워드에는 128비트 마스터 키를 비트별 NOT연산한 값이 순서대로 적용되고 11, 12번째 워드는 IV값, 13, 14번째는 IV값을 비트별 NOT연산한 값이 순서대로 적용된다. 초기화의 진행과정은 4개의 워드 가지고 간단한 회전(Rotation)연산과 워드 덧셈 연산을 4회 반복 수행한 후 ASR의 5개 워드에 순서대로 적용하고 다시 5회 반복 수행 후 NLB의 16개 워드를 초기화시킨다.

2.2 선형 궤환 순서기(ASR)

의사난수발생기로 사용할 수 있는 산술 쉬프트 레지스터(Arithmetic Shift Register)는 GF(2^n)상에서 0이 아닌 초기값에 0 또는 1이 아닌 임의의 수 D를 곱하는 수열로 정의한다. ASR의 i번째 값(상태) ASR^i는 $ASR^0 \cdot D^i$가 된다.

$D^k = 1$이 되는 t가 $t = 2^n-1$로 유일하게 되는 비복원 다항식(irreducible polynomial)이 ASR의 특성다항식(Characteristic Polynomial)이며, ASR

의 주기는 2^n-1로 최대 주기를 가진다. 그리고 ASR의 선형 복잡도(Linear Complexity)는 기존의 LFSR(Linear Feedback Shift Register)의 선형 복잡도 보다 높아서 안전도가 높다.

본 논문에서는 GF(2^{151})상에서 특성다항식은 16진수로 '0x00800000 0x00000001 0x00000001 0x00000004 0x00000025', D = 2^{23}을 적용한다. ASR의 동작을 소프트웨어로 작성하면 다음과 같다.

```
w0 = ASR[4];
ASR[4]=ASR[3] >> 9;
ASR[3]=((ASR[3]<<23) | (ASR[2]>>9)) ^ w0;
ASR[2]=((ASR[2]<<23) | (ASR[1]>>9)) ^ w0;
ASR[1]=((ASR[1]<<23) | (ASR[0]>>9)) ^ (w0<<2);
ASR[0]=(ASR[0]<<23) ^ (w0<<5) ^ (w0<<2) ^ w0;
```

2.4 포인터 생성기(PG)

ASC16에서는 2개의 포인터가 사용된다. 순차적으로 1씩 증가는 포인터 pts와 예측할 수 없는 가변적인 포인터 pt0가 있다. pts는 초기에 NLB의 첫 번째 워드(NLB_0)를 가리키고 있으며, pt0은 ASR의 마지막 워드(ASR_4)의 LSB(least signification bit) 4비트와 1을 OR연산을 수행한 후 pts와 XOR연산을 수행한 값을 포인터로 사용한다. 이는 pts와 pt0이 절대 NLB의 같은 워드를 가리킬 수 없다.

2.4 비선형 순서기(NLB)

포인터 변수 pts는 초기에 NLB_0을 참조하고, NLB의 업데이터는 워드 단위로 이루어진다. NLF의 출력 32비트와 포인터 변수 pt0과 6, 10과 XOR연산을 수행한 후 pt0이 가리키는 NLB의 워드 각각을 2개의 8비트에서 32비트로 비선형 변환을 하는 S-box를 통과한 값을 모두 더하여 포인터 변수 pts가 가리키는 NLB에 저장하고, 포인터 변수 pts는 1증가한다. pts가 16이 되면 pts는 다시 0으로 셋팅 된다.

2.5 비선형 순서 발생기(NLF)

NLF는 ASR의 3개의 워드 ASR_0, ASR_1, ASR_2와 포인터 변수 pts, pt0이 가리키는 NLB의 2개의 워드를 가지고 비선형 변환 8비트 S-박스를 2번 통과한 후 최종적으로 32비트 출력을 얻는다. (그림 2)는 NLF의 전체적인 진행을 그림으로 표현한 것이다. (그림 2)에서 W는 NLB의 pts와 pt0이 참조하는 2개의 워드를 더한 값이고, 이 W와 ASR_0을 XOR연산을 한 후 8비트 S-박스를 통과하고 확산(Permutation)을 수행한다. 그리고 다시 ASR_1과 XOR연산 후 2단계 S-박스를 수행하고 확산과정을 거쳐 마지막 ASR_2와 XOR연산을 수행 후 32비트 출력을 얻는다.

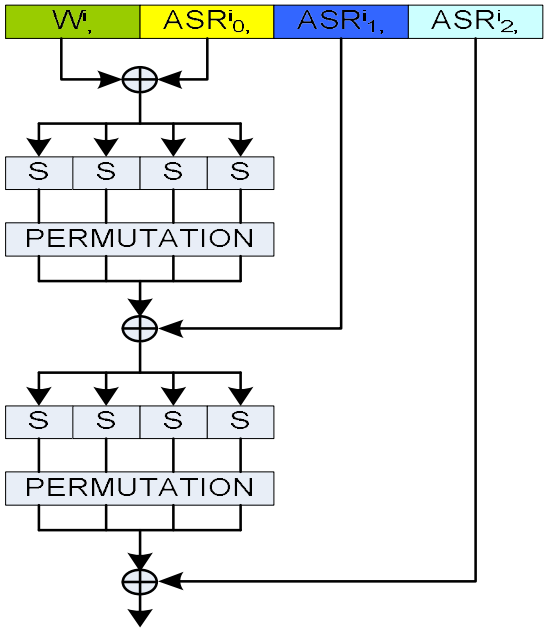

그림 2. NLF 흐름도

III. 안 전 성 분 석

3.1 ASR의 분석

ASR은 참고문헌[8]에 의해 최대 주기 수열을 생성한다. 본 논문에서는 ASR에 의해 생성된 수열에서 ASR_3의 32비트를 사용하므로 ASR_3을 추측할 수 있는 확률은 2^{-151} * 32이므로 2^{-146}의 확률을 얻을 수 있다. 그리고 ASR은 (그림 3)과 같이 ASR의 각 워드는 최소 23비트 이상 값의 변화를 주고 있으며 이는 ARS의 출력과 각각의 워드간의 상관관계를 복잡하게 한다.

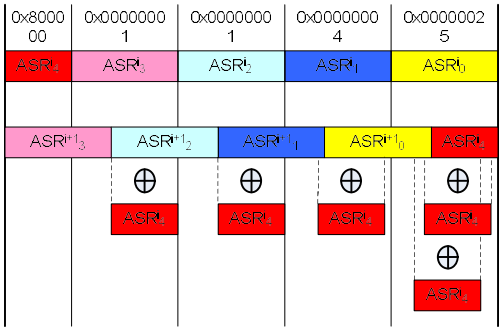

그림 3. ASR 진행과정

3.2 NLF의 분석

비선형 변환 2단계 S-박스를 사용하고, NLF의 출력을 다시 8비트에서 32비트로 변화하는 2개의 S-박스를 통과시켜 NLB의 주기를 향상시켰다. 먼저 NLF의 내부 S-박스와 확산은 AES[10]를 그대

로 사용하였다. S-박스와 확산의 처음과 끝 그리고 중간의 XOR연산은 화이트닝(Whitening)단계로 분석은 S-P-S-P과정으로 생각하면 된다. S-박스의 최대 차분 특성이 2^{-6}이고, 확산은 MDS(Maximum Distance Separated)코드 이므로 최종 5개의 S-박스가 활성화된다. 그래서 NLF의 출력은 $(2^{-6})^5 = 2^{-30}$이 된다. NLF의 출력 후 32비트 2개의 S-박스를 한 번 더 통과하므로 $2^{-36} + 2^{-36} + 2^{-30}$은 약 2^{-37}이 된다.

3.3 NLB의 분석

NLF의 2^{-37}의 확률을 가지는 워드가 16개있고, pts가 참조하고 있는 워드를 제외한 나머지 워드를 PG의 알고리즘으로 선택하는 최종적인 확률은 $2^{37} * 2^4 * 2^{16}$-1로 2^{57}이 된다. 이는 SSC2의 2^{52}보다 길다.

Ⅳ. 구 현 및 실 행 결 과

소프트웨어 구현 시 고속처리를 위해 NLF의 S-박스와 확산을 미리 계산한 후 메모리에 저장해 놓고 참조하는 방식을 사용했다. 소요되는 메모리는 약 6KB 정도이고 이는 초기화 과정에서 미리 계산한다. 제안한 ASC16의 소프트웨어 구현은 Visual Studio 2005 C 컴파일러를 사용하였고 실행환경은 Windows Vista, Intel Core(TM)2 Duo CPU 2.26Ghz, 2.27Ghz, 2GB RAM의 환경에서 SSC2와 제안한 알고리즘의 수행 시간을 테스트했다. 결과는 (표 1)과 같다.

표 1. 수행시간 테스트 결과

시간 / 알고리즘	키 스트림 생성 시간
ASC16	1.5GB / 30초
SSC2	1.5GB / 29초

(표 1)의 결과는 제안한 ASC16과 SSC2의 수행시간테스트는 거의 차이가 없고, 안전성은 향상되었다.

Ⅴ. 결 론

소프트웨어 구현으로 무선 통신 장비에 적용할 고속 스트림 암호 ASC16을 제안한다. ASC16은 ASR(Arithmetic Shift Register), NLF(Non-Linear Filter), NLB(Non-Linear Block)로 구성된 32비트 출력의 고속 스트림 암호이다. ASC16은 기존의 SSC2보다 더 긴 2^{57}의 주기를 갖고 ASR을 적용하여 상관공격을 더욱 어렵게 하여 SSC2보다 안전성이 향상되었으며, 수행 속도는 SSC2와 거의 차이가 없다.

ASC16은 무선통신 등과 같은 제한적인 환경에서 고속 암호 수행에 유용하게 사용될 수 있다.

감사의 글

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신 인력양성사업, 중소기업청의 산학연공동기술 개발 지원 사업(선도형)의 지원으로 수행되었음.

참 고 문 헌

[1] “New European Schemes for Signatures. Integrity. and Encryption(NESSIE)” http://cryptonessie.org/.

[2] http://www.ecrypt.eu.org/

[3] “Cryptography Research and Evaluation Committees(CRYPTREC)” http://www.cryptrec.go.jp/.

[4] C. Carroll, A. Chan, and M. Zhang "The software-oriented stream cipher SSC-Ⅱ" FSE 2000 LNCS Vol.1978 p.p 39-56 2000.

[5] P. Hawkes, F. Quick, and G. Roes "A practical cryptanalysis of SSC2" Selected Areas in Cryptography 2001 LNCS Vol. 2259 p.p 27-37 2001.

[6] D. E. Knuth "The Art of Computer programming. Volume 2 : Seminumerical Algorithms" 3rd Edi tion Addison-Wesley 1997.

[7] P. Hawkes, and G. Rose "Correlation cryptanalysis of SSC2" Presented at the Rump Session of CRYPTO 2000.

[8] 박창수, 조경연 “갈로이 선형 궤환 레지스터의 일반화” 전자공학회논문지 제43권 C1편 제1호 2006.1.

[9] L. Brynielsson "On the linear complexity of combined shift register sequences" In F. Pichler editor Advances in Cryptology - Eurocrypt ‘85 p.p 156-166 Berlin Springer-Verlag 1986.

[10] J. Daemen, and V. Rijmen, "AES Proposal : Rijndael" http://www.csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf, 1999.