JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 6, pp. 73-79, June 2020
https://doi.org/10.9708/jksci.2020.25.06.073

A Study on the Security Processor Design based on Pseudo-Random Number in Web Streaming Environment

Seon-Keun Lee*

*Professor, Dept. of Electrical and Electronics Engineering, Woosuk University, Jeonbuk, Korea

[Abstract]

Nowadays, with the rapid spread of streaming services in the internet world, security vulnerabilities are also increasing rapidly. For streaming security, this paper proposes a PN(pseudo-random noise) distributed structure-based security processor for web streaming contents(SP-WSC). The proposed SP-WSC is basically a PN distributed code algorithm designed for web streaming characteristics, so it can secure various multimedia contents. The proposed SP-WSC is independent of the security vulnerability of the web server. Therefore, SP-WSC can work regardless of the vulnerability of the web server. That is, the SP-WSC protects the multimedia contents by increasing the defense against external unauthorized signals. Incidentally it also suggests way to reduce buffering due to traffic overload.

▸ **Key words**: Cryptographic algorithm, Adaptive, Processor, Security, Streaming, Web-Server

[요 약]

현재, 인터넷 세상은 스트리밍 서비스의 급격한 보급과 더불어 보안의 취약성 역시 매우 급속도로 증가되는 추세이다. 이러한 스트리밍 보안을 위하여, 본 논문은 웹 스트리밍 콘텐츠에 대한 PN 분산 구조 기반 보안프로세서 (SP-WSC)를 제안한다. 제안된 SP-WSC는 기본적으로 PN 분산 코드 알고리즘을 웹 스트리밍 특성에 맞게 설계하였기 때문에 다양한 멀티미디어 컨텐츠에 대한 보안을 수행할 수 있다. 제안된 SP-WSC는 웹 서버의 보안 취약성과 독립적이다. 그러므로 SP-WSC는 웹 서버의 취약성에 무관하게 동작할 수 있다. 즉, SP-WSC는 외부의 비인가 신호에 대한 방어력을 증대시켜 멀티미디어 컨텐츠를 보호한다. 또한 부수적으로 이것은 트래픽 과부하에 의한 버퍼링 현상을 감소시킬 수 있는 방안을 제시한다.

▸ **주제어**: 암호알고리즘, 적응형, 프로세서, 보안, 스트리밍, 웹서버

• First Author: Seon-Keun Lee, Corresponding Author: Seon-Keun Lee
*Seon-Keun Lee (caiserrisk@woosuk.ac.kr), Dept. of Electrical and Electronics Engineering, Woosuk University
• Received: 2020. 03. 30, Revised: 2020. 05. 30, Accepted: 2020. 06. 02.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

현대사회는 인터넷(유튜브, SNS 등)과 스마트폰, 다양한 서버 플랫폼, IoT(Internet of Things) 원/근거리 무선망 등의 발달로 인하여 장소, 시간 등에 구애받지 않고 다양한 멀티미디어를 활용할 수 있다는 문화적/기술적 격변기에 있다[1].

이러한 시점에서 보안에 대한 중요성은 더욱 심화된다. 그러나 현 시점에서 거의 대부분의 서버 및 클라이언트들은 비용 및 환경제약으로 인하여 소프트웨어적으로는 프로그래밍 기법을, 하드웨어적으로는 경량화 암호알고리즘을 이용하여 보안을 수행하고 있다. 이러한 프로그래밍 기법과 경량화 암호알고리즘을 이용한 보안기법은 OS 플랫폼과 응용프로그램의 기반에서 동작되거나 경량화 암호시스템을 하드웨어적으로 구현하여 동작시키기 때문에 시스템의 부하 및 처리속도 등의 한계를 태생적으로 가질 수 밖에 없다.

그러므로 본 논문에서는 서버와 클라이언트간의 실시간 스트리밍 및 멀티미디어 DB의 다양성을 안전하게 서비스할 수 있는 PN 분산 구조 기반의 보안 프로세서를 설계하였다. 제안된 SP-WSC(security processor for web streaming contents)는 기존 스트리밍 방식인 OS 플랫폼 및 응용프로그램 또는 경량화 암호시스템에서 수행되는 것이 아니라 별도의 프로세서로 동작하기 때문에 웹 서버의 보안과 무관하게 동작한다. 그러므로 웹 서버의 보안침해 및 과부하상태라도 독립적인 보안기능을 수행하기 때문에 멀티미디어 컨텐츠를 보호할 수 있을 뿐만 아니라 트래픽 과부하에 의한 버퍼링 현상을 미연에 방지할 수 있고 유료 사이트에 최적화 할 수 있다. 이러한 이유로 제안된 SP-WSC는 보다 안전한 컨텐츠, 안정적인 속도 등을 웹 스트리밍 서비스로 제공할 수 있다.

II. Preliminaries

1. Web multimedia playback technology

웹을 이용한 멀티미디어 재생 방식은 프로그레시브 다운로드(progressive download: PD) 방식과 스트리밍(streaming) 방식이 있다. PD의 경우, 파일의 일부분을 다운 받음과 동시에 재생할 수 있으며 동영상이 끊김없이 재생되기 위해서는 네트워크 속도가 동영상의 데이터 레이트 보다 높아야 한다. 이 방식은 내 컴퓨터로 파일을 다운로드 하기 때문에 보안에 문제가 있어 유료 비디오 서비스 분야에 한계가 있다. 또한 동영상의 극히 일부분을 보고 종료하여도 다운받은 파일의 용량에 따라 비용을 지불해야 한다. 이러한 단점을 없애고자 만들어진 방식이 스트리밍 방식이다.

스트리밍 방식은 파일을 전체 다운로드 받지 않고도 원하는 부분을 재생할 수 있으며, 서버는 인터넷 네트워크 속도의 가변성을 고려하여 파일을 작은 조각으로 나누고 압축함과 동시에 이에 대한 미디어 정보(manifest)를 만들어 클라이언트로 전송하여 끊김이나 해상도, 버퍼링 등의 문제를 해결하고 있다. 웹 멀티미디어 재생기술에 대한 대표기술은 표 1과 같다.

Table 1. Web multimedia playback technology

	description
Progressive download	Avoid expensive equipment (media server) and complex (such as the RTMP protocol)
HTTP Pseudo-Streaming	Click on the part that has not been downloaded -> Since you have metaframe information, you can play it starting from the desired part.
RTMP/RTSP Streaming	* Limitations of real-time relaying and security issues are complemented. * Live broadcasting is possible. * Bandwidth efficiency is increased because only necessary part is transmitted.
Adaptive HTTP Streaming	A combination of the advantages of Progressive Download and RTMP

표 1에서와 같이 웹 환경에서 멀티미디어 재생방식은 다양한 방식으로 발전하고 있다. PD의 단점을 보완하기 위하여 스트리밍 방식이 발전하고 있지만, 스트리밍 방식에도 플레이와 보안 사이에 트레이드 오프가 존재한다.

웹 환경에서 PD의 단점인 버퍼링 및 해상도에 대한 것을 해결한 방식이 적응형 스트리밍(adaptive streaming) 방식이다[2]. 이 방식은 고도화되는 웹 동영상 서비스의 버퍼링 없는 서비스를 제공하기 위하여 새로운 프로토콜 대신 기존 HTTP를 이용하여 최적의 상태를 유지토록 한 방식이다.

적응형 스트리밍 방식은 멀티미디어 콘텐츠를 다양한 해상도로 인코딩함과 동시에 하나의 콘텐츠로 저장하는 것이 아니라 여러 개의 분할된 형태로 저장해둔다. 사용자가 멀티미디어를 플레이할 때, 서버와 클라이언트 사이의 네트워크 상황에 따라서 적절한 전략으로 콘텐츠의 소스를 선택해 최적의 스트리밍 서비스를 제공한다. 다양한 소스로 인코딩 되어 있고 분할되어 있는 미디어 소스는 상황에 따라 선택하여 플레이 할 수 있기 때문에 다른 퀄리티로 쉽게 교체할 수 있다[3].

2. Web Streaming Security Methods

웹을 통한 다양한 스트리밍 기법들이 제공되면서 이에 적용 가능한 다양한 기법의 보안 기술이 발전되고 있다. 이러한 기법들은 대부분 소프트웨어 기법을 사용한다. 즉, OS 플랫폼을 기반으로 동작하는 메카니즘을 가지기 때문에 보안에 항상 취약할 수 밖에 없다[4]. M. A. Rajan 등[5]이 제안한 시스템은 Hierarchical Inner Product Encryption (HIPE) 기법을 사용하여 인가된 사용자만 보안을 수행하는 방법이다. 이 방법은 인가된 서비스만을 사용자가 이용할 수 있는 방법이므로 다수의 이용자들에게 적용하기 위해서는 더욱 복잡한 메카니즘이 필요할 수 밖에 없다. Venčkauskas, A. 등[6]이 제안한 방법은 Fog Node-End Device 계층을 위한 경량의 안전한 스트리밍 프로토콜을 제안한 내용이다. 이 방법은 다양한 보안 서비스를 제공하면서 에너지 효율성을 증대시키기 위하여 UDP(User Datagram Protocol) 패킷을 사용하여 인증 데이터를 스트리밍 데이터에 포함시킨 것이다. 이 방식은 다양한 서비스를 제공할 수 있지만, 분산된 사용자들 또는 센서들에 대한 보안기능은 전무하다.

III. The Proposed SP-WSC algorithm

적응형 스트리밍 방식을 포함하는 웹 스트리밍 환경에서 가장 중요한 파라미터는 데이터 레이트이다[7-9]. 속도에 의하여 미디어 품질이 결정되기 때문이다. 이러한 단점들을 없애고자 미디어 플랫폼에서 PD와 다른 방식으로 스트리밍 방식 등의 다양한 기법들이 발전되고 있다[10].

웹 서비스의 수요가 가중되고 있는 시점에서 웹 서비스에 대한 보안의 필요성은 절대적이다. 그러므로 본 논문은 PN 분할 구조를 갖는 SP-WSC 알고리즘을 웹 서비스의 데이터 레이트 및 사용자 수에 무관하게 안정적인 서비스를 수행할 수 있도록 적응형 스트리밍 기법에 적용하였다.

멀티미디어 서비스를 위한 웹 서비스는 대역폭, 데이터 레이트 등의 성능향상을 위하여 스트리밍 방식을 많이 적용한다. 적응형 스트리밍을 제공하는 곳들은 애플의 Apple-HLS(HTTP Live Streaming)[9], MPEG(Moving Picture Experts Group)의 표준안인 DASH(Dynamic Adaptive Streaming over HTTP)[8] 등이 있다. 이러한 스트리밍 기법들은 HTMLx(Hyper Text Transfer Protocol x)에서 무난하게 동작되기 때문에 별도의 부가모듈이 특정되지는 않는다. 이러한 스트리밍 방식의 웹 서비스의 보안을 위하여 적용되는 암호알고리즘은 경량화 암호알고리즘을 적용하거나 스트림 암호알고리즘을 적용하게 된다.

특히 웹 멀티미디어는 근거리 통신망 및 인터넷 서비스 등의 특징 때문에 블록 암호알고리즘보다 고속 동작이 가능한 스트림 암호기법이 많이 이용되고 있다. 스트림 암호기법은 PN 자체만으로 안전성을 제공하지 못하므로, 높은 주기성과 높은 통계적 성질을 PN에 결합하여 우수한 암호 알고리즘이 설계된다. 그러나 PN에 의해 발생된 수열은 큰 주기 및 높은 통계성을 갖지만, 출력 수열로부터 쉽게 예측이 가능하다. 일반적인 PN은 단지 한 사이클에 하나의 난수만을 생성하지만, 대부분의 어플리케이션이 다양화되면서 다중 비트의 난수열이 요구되다. 이를 위해 다중 LFSR(Multiple Linear Feedback Shift Register: PN을 구성하는 하부조직) 구조[11]가 주로 사용되었으나, 구현상의 문제점과 해석의 용이성이라는 단점을 갖는다. 이와 같은 단점을 보완하기 위해 Leap-ahead 구조를 갖는 LFSR이 제안되었다. Leap-ahead 구조는 하나의 LFSR을 이용하여 다중 비트 출력을 얻을 수 있는 구조이기 때문에 구현상의 문제점은 감소되지만, LFSR의 크기와 출력 단계 수의 상관관계에 따라 생성되는 난수들의 주기가 크게 변화된다[12].

제안된 SP-WSC는 적응형 스트리밍 방식의 장점을 유지하면서 암호화 기능을 증대시키기 위하여, 미디어 데이터 다중 분할과 이에 해당하는 정보를 가진 미디어 정보를 Leap-ahead 구조를 갖는 LFSR에 적용하여 다중 난수열을 발생시켜 실시간 보안을 유지하고 구현상의 문제점도 감소시킬 수 있는 PRNG(pseudo-random number generator) 구조이다.

분할된 미디어들과 미디어 정보 그리고 분할 구조를 가지는 PRNG로 구성된 SP-WSC는 기존 적응형 스트리밍 방식을 사용하는 과정에서 하드웨어 기반 암호화 과정을 거치기 때문에 실시간 처리가 가능하고 적응형 스트리밍 방식의 장점을 유지함과 동시에 보다 복잡한 네트워크 환경에 최적의 서비스를 제공할 것으로 생각된다.

그림 1은 SP-WSC 구조를 스트리밍 방식에 적용한 경우이다. 웹 서버 내부에 위치하는 SP-WSC는 클라이언트의 미디어 요청신호에서 클라이언트에 대한 요청정보(ID, URI(Uniform Resource Identifier) 등)와 서버의 미디어 정보 등을 이용하여 PRNG의 IV(Initialization Vector)로 사용한다. 그리고 클라이언트는 암호화된 미디어와 미디어 정보 그리고 자체 정보를 이용하여 복호화 과정을 거치게 된다.

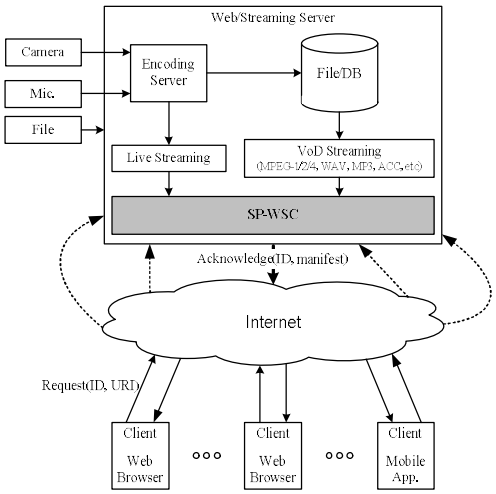


Fig. 1. Applied SP-WSC architecture in web adaptive streaming

MSE(Media Source Extension)[13]에서 세그먼트는 인코딩된 동영상 데이터의 작은 조각이다. 이 조각에 대한 정보(코덱, 타임스탬프 등)는 DASH[8]나 HLS[9]를 통해 얻어지며 사용자가 동영상을 볼 수 있도록 플레이어에 전달한다. SP-WSC에서 세그먼트는 기존 세그먼트와 동일하게 두 가지 종류의 세그먼트가 있다. 초기화 세그먼트(Initialization Segment: IS_{ij}, $1 \leq i, j \leq m, n$)와 미디어 세그먼트(Media Segment: M_{ij})이다. IS_{ij}는 실제 동영상 정보를 담고 있는 미디어 세그먼트의 시퀀스를 디코딩하는데 필요한 정보를 담고 있으며 코덱 초기화 데이터, 트랙 ID, 타임스탬프 오프셋 등의 정보를 포함한다. M_{ij}는 패킷화상태이며 자신이 플레이되어야 할 미디어 타임라인상의 타임스탬프 정보가 포함된 실제 동영상 데이터다. M_{ij}는 IS_{ij}의 정보를 토대로 자신의 위치를 알기 때문에 M_{ij}를 순차적으로 플레이어에 제공하지 않아도 플레이되어야 할 위치에서 플레이 되게 된다. 만약 IS_{ij} 정보가 없을 경우, M_{ij}를 플레이어에 제공해도 정상적으로 플레이 되지 않는다. 또한 MSE, EME (Encrypted Media Extensions)[13] 등에서 보안을 수행하고 있지만, HTML 상에서 이루어지기 때문에 복잡한 형태의 네트워크 환경 및 실시간 처리가 어렵다는 단점이 있다. 그러므로 SP-WSC는 M_{ij}가 아닌 IS_{ij}를 이용하여 암호화를 수행한다. 하나의 서버에 미디어 세그먼트를 $m \times n$(세그먼트가 m개, 세그먼트 세부 분할이 n개)으로 분할하였다고 가정한다.

그림 2는 SP-WSC가 적용된 웹 적응형 스트리밍 구조이다. IS_{ij}에 피보나치 형(fibinacci type) LFSR을 적용하여 암호화 값을 도출하고 이를 미디어 정보(manifest)에 첨부하여 외부로 송출한다. 송출된 미디어 정보는 다양한 형태의 클라이언트의 스트리밍 정보로 공급된다.

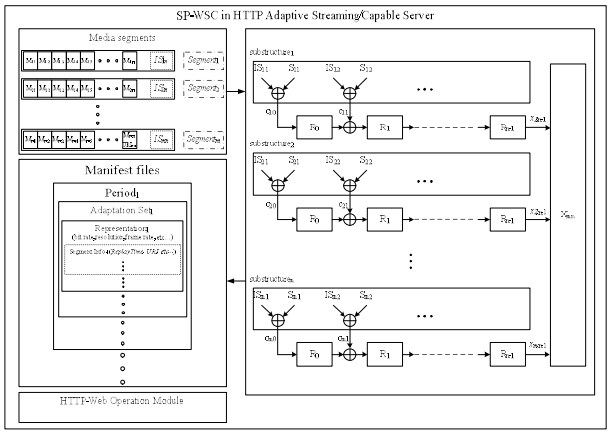


Fig. 2. Proposed SP-WSC architecture

IS_{ij}와 요청 클라이언트 정보인 S_{ij}는 PRNG IV로 사용되기 위해 식 (1)과 같은 과정을 수행한다. 식 (1)은 PRNG LFSR의 시드(seed)로 사용하고, 이를 스트림 암호에 적용하게 된다.

$$c_{ij} = IS_{ij} \oplus S_{ij} \tag{1}$$

IS_{ij}의 수를 $m \times n$개로 고정한 경우, 시스템 전체 값은 식 (2)와 같이 고정된 길이의 값을 가진다.

$$\begin{aligned} f(x)_{mn} &= x_{1,n-1} \,|\, x_{2,n-1} \,|\, \cdots \,|\, x_{m,n-1} \\ &= x_{1,n-1} \,\&\, x_{2,n-1} \,\&\, \cdots \,\&\, x_{m,n-1} \end{aligned} \tag{2}$$

여기에서 $i = 1, 2, \cdots m$, $j = 1, 2, \cdots n$이고, &는 단순 결합(junction)이다. 또한 분산 블록인 하부구조(substructure_1^m)는 식 (3)과 같이 미디어 정보와 클라이언트 정보를 이용하여 시드값을 생성한다. 이때 IS_{ij}와 클라이언트 정보값을 1:1로 LFSR 원시다항식으로 매핑을 수행하기 위하여 식 (3)과 같이 LFSR에 대한 분산 블록(segmentation)을 수행한다.

$GF(q^m)$를 기초체(base field) $GF(q)$상의 확대체(extension field)라 하면, $GF(q^m)$는 $GF(q)$상의 m차 기약다항식의 잉여집합(residue set)이다. 또한, $GF(q^m)$의 모든 원소에 대한 최소다항식은 항상 존재하고 유일하다[14].

$$\begin{aligned} x_{1,n-1} &= c_{10} + c_{11}x + c_{12}x^2 + \ldots + x^{n-1} \\ &= \begin{bmatrix} IS_{11} \\ \oplus \\ S_{11} \end{bmatrix} + \begin{bmatrix} IS_{12} \\ \oplus \\ S_{12} \end{bmatrix} x + \ldots + \begin{bmatrix} IS_{1n} \\ \oplus \\ S_{1n} \end{bmatrix} x^{n-1} \\ &\vdots \\ x_{m,n-1} &= c_{m0} + c_{m1}x + c_{m2}x^2 + \ldots + x^{n-1} \\ &= \begin{bmatrix} IS_{m1} \\ \oplus \\ S_{m1} \end{bmatrix} + \begin{bmatrix} IS_{m2} \\ \oplus \\ S_{m2} \end{bmatrix} x + \ldots + \begin{bmatrix} IS_{mn} \\ \oplus \\ S_{mn} \end{bmatrix} x^{n-1} \end{aligned} \tag{3}$$

확대체 $GF(q^n)$의 원시 원소를 근으로 하는 최소다항식은 m차 원시다항식으로 귀결된다. α가 확대체 $GF(q^m)$의 원시원소일 경우, $\gcd(k, q^m-1) = 1$($\gcd$: greatest common divisor)인 모든 정수 k에 대해 α^k는 $GF(q^m)$의 원시원소이고, 이 원소의 최소다항식 $g_k(x) = x^m + \sum_{i=0}^{m-1} c_i x^i$은 m차 원시다항식이다. 여기서, c_i는 $GF(q)$의 원소이다[14].

그러므로 위 정의[14]에 의하여 원시다항식 $g_k(x)$는 식 (4)를 만족한다.

$$g_k(\alpha^k) = \alpha^{km} + c_{m-1}\alpha^{k(m-1)} + c_{m-2}\alpha^{k(m-2)} + \ldots + c_0 = 0 \quad (4)$$

이때 α^h는 식 (5)와 같이 표시할 수 있다.

$$\alpha^h = a_{m-1}^{[h]} x^{m-1} + a_{m-2}^{[h]} x^{m-2} + \ldots + a_0^{[h]} \quad (5)$$

여기서 $0 \leqq h \leqq q^n-1$이다. $a_i^{[h]} \in GF(q)$는 α^h $(i = 0, 1, 2, \ldots, m-1)$의 i번째 성분이다.

식 (5)를 식 (4)에 대입하고 α^{km}의 각 성분에 대하여 정리하면 m개의 미지수를 갖는 식 (6)과 같은 m개 선형방정식이 만들어진다.

$$\begin{aligned} a_0^{[km]} &= -c_{m-1}a_0^{[k(m-1)]} - c_{m-2}a_0^{[k(m-2)]} - \ldots - c_0 a_0^{[0]} \\ a_1^{[km]} &= -c_{m-1}a_1^{[k(m-1)]} - c_{m-2}a_1^{[k(m-2)]} - \ldots - c_0 a_1^{[0]} \\ &\cdots\cdots\cdots\cdots \\ a_{m-1}^{[km]} &= -c_{m-1}a_{m-1}^{[k(m-1)]} - c_{m-2}a_{m-1}^{[k(m-2)]} - \ldots - c_0 a_{m-1}^{[0]} \end{aligned} \quad (6)$$

α를 근으로 하는 원시다항식으로부터 $\alpha^k, \alpha^{2k}, \ldots, \alpha^{mk}$를 각각 계산하여 계수 $a_i^{[h]}(i = 0, 1, 2, \ldots, m-1)$, $(h = 0, k, 2k, \ldots, mk)$을 얻는다. 계수 $a_i^{[h]}$를 식 (6)에 대입하여 $g_k(x)$의 계수 $c_0, c_1, \ldots, c_{m-1}$를 구할 수 있다. 이때 계수값을 구하는 방법은 matrix inversion과 A. D. Porto가 제안한 방법[15]이 있다.

SP-WSC는 n개와 m개로 분할된 IS_{ij} 정보를 이용하여 m개의 $segment_i$를 생성하고, 생성된 이 값과 요청 클라이언트 정보를 이용하여 m개의 정보를 생성한다. 그러므로 SP-WSC는 2-레벨로 구성된다. 2-레벨에 대하여 α를 상위계층($segment_i$)으로, a를 하위계층($structure_i$)으로 구성할 수 있다.

식 (5)는 m개로 구성된 a에 대한 선형방정식이다. 그러므로 이때 n개의 LFSR로 구성된 정보가 존재할 경우, 식 (3), 식 (5)를 그림 2와 같이 구성하면, 식 (3)과 식 (5)는 식 (7)과 같이 변형된 표현이 가능하다.

$$\begin{aligned} \alpha^h &= a_0^{[h]} + \ldots + a_{n-2}^{[h]} x^{n-2} + a_{n-1}^{[h]} x^{n-1} \\ &= x_{1,n-1} \mid x_{2,n-1} \mid \cdots \mid x_{m,n-1} \end{aligned} \quad (7)$$

여기서 $0 \leqq h \leqq q^n-1$이다. $a_i^{[h]} \in GF(q)$는 α^h $(i = 0, 1, 2, \ldots, n-1)$의 i번째 성분이다.

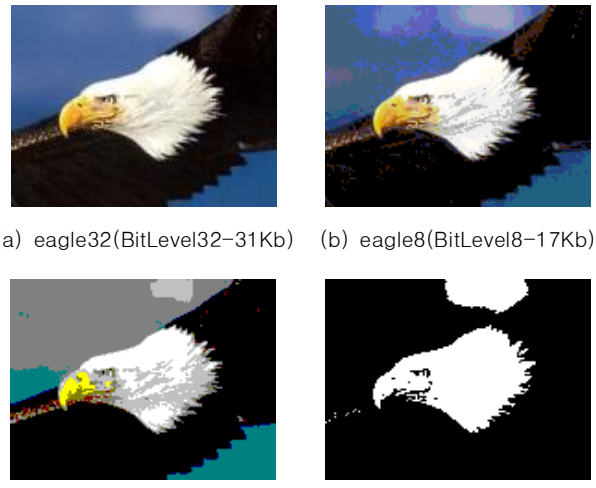
그러므로 식 (2)와 식 (7)은 동일한 표현이다.

이러한 표현은 SP-WSC를 이용하면 스트리밍 기법을 사용하는 미디어 정보와 스트림 방식의 암호화 기법을 융합시킬 수 있다는 것을 증명한다. 그러므로 SP-WSC를 스트리밍 미디어 전송기법에 적용하게 될 경우, 스트리밍 기법의 장점을 유지하면서 보다 안전한 보안을 수행할 수 있고 SNI(server name indicator)[16]에 대한 방어책으로 사용할 수 있다.

IV. Design and simulation of the proposed SP-WSC

SP-WSC를 설계하기 위하여 사용된 LFSR 구조는 피보나치(Fibonacci) 타입이다. 이미지 크기는 144x110이고, 비트 깊이는 8이다. 사용된 이미지는 그림 3과 같다.

그림 3에서, 사용된 이미지들은 동일한 이미지에 대하여, 전송채널 환경의 변화를 위한 모델로 비트레벨을 다르게 설정(4가지)하였다. 이때 사용된 각각의 이미지들에 대한 데이터들은 표 2와 같다. 표 2는 각 이미지들에 대한 IS_{ij} 값들로 사용하기 위한 값들이다. 각각의 이미지 파일들에서 이미지 정보를 추출하고, 추출된 정보를 이용하여 IS_{ij}를 만들고, 요청 클라이언트 정보를 이용하여 S_{ij}를 만들어 사용하였다.



(a) eagle32(BitLevel32-31Kb) (b) eagle8(BitLevel8-17Kb)

(c) eagle4(BitLevel4-8Kb) (d) eagle1(BitLevel1-3Kb)

Fig. 3. SP-WSC simulation images

IS_{ij}와 S_{ij} 정보를 이용하여 SP-WSC 암호화를 수행한 후, 결과 값을 미디어 정보에 피드백 시킨 후 요청 클라이언트에 전송하게 된다.

SP-WSC 설계환경에서 회로합성은 QuartusII v.13.0, 모델링 및 모의실험은 ModelSim10.1d이며, 사용된 디바이스는 CycloneIII-EP3C40F324C6이다.

그림 4는 SP-WSC 프로세서를 설계한 후 수행한 모의실험 결과이다.

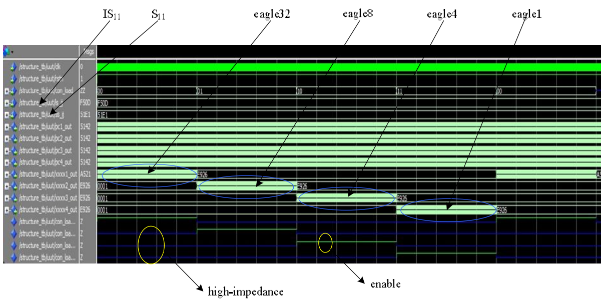


Fig. 4. SP-WSC processor simulation results

미디어는 전송채널에 적합하게 해상도를 4가지로 분류한 후, 미디어 정보와 요청 클라이언트 정보를 이용하여 암호화를 수행하고 이를 다양한 채널로 전송할 수 있는지를 확인하였다.

제어신호 con_load의 값에 따라 미디어 데이터를 전송하고, 그렇지 않은 경우, 하이 임피던스를 가지도록 하여 전기적으로 분리할 수 있도록 설계하였다.

모의실험시, @100MHz에서 동작을 검증하였으며 각 데이터들의 흐름을 관측하기 위하여 PC(program counter)를 이용하였다.

V. Conclusions

1인 미디어 및 인터넷상의 미디어 정보에 대한 데이터량 급증은 개인정보에 대한 노출이 심화될 수 있으며 정보의 위/변조들에 대한 심각한 문제점을 노출시킬 수 밖에 없다. 또한 정보의 전송 도중 발생하는 네트워크 환경 변화(클라이언트들 중 일부의 오류발생 등)에 의하여 원하는 정보를 전송하기 힘들기 때문에 다양한 방법이 발전하고 있지만 이에 맞는 보안기법은 아직은 제자리 수준이다.

그러므로 본 연구에서 제시한 하드웨어 기반 SP-WSC 프로세싱은 서버의 리소스를 점유하지 않기 때문에 트리픽 과부하로 인한 버퍼링 현상을 감소시킬 수 있으며, 네트워크 환경에 유동적으로 미디어 정보를 암호화하여 전송함과 동시에 서버와 클라이언트와의 관계가 아닌 그룹핑된 분산된 클라이언트들간의 보안을 수행할 수 있다는 특징을 가진다. 이러한 이유 때문에 SP-WSC는 네트워크 환경에서 개인별 통신에 대한 보안을 확보하며 웹상에서 유료회원 관리가 용이함과 동시에 인터넷 강좌, 비화통신, 분산되어 처리되는 센서 시스템들에 매우 적합할 것으로 생각된다.

Table 2. Used image data for SP-WSC simulation

	description
eagle32	{{{71,102,157},{72,105,159},{71,104,158},{70,103,157},{71,104,158},{71,102,159},{71,103,160},{71,103,160},{72,104,161},{71,103,160},{72,104,161},{74,106,163},{73,105,162},{73,106,159},{73,106,159},{74,107,160},{74,107,160},{73,106,159},{74,107,160},{76,109,162},{75,108,161},{76,107,161},{76,109,162},{77,110,163},{78,111,164},{79,110,165},{79,110,165},{80,111,166},{80,111,166},{82,111,167},{82,111,167},{82,112,166},{83,113,167},{84,114,166},{84,114,166},{83,113,165},{83,113,165},{86,113,166},{86,113,166},{87,114,167},{87,114,167},{87,114,167},{86,113,166},{86,113,166},{86,113,166},{87,112,166},{87,114,167},{87,114,167},{86,113,166},{85,112,165},{86,113,166},{85,112,167},{83,110,165},{80,107,162},{81,108,163},<<35>>,{145,153,189},{146,154,190},{147,155,191},{146,154,190},{146,154,191},{143,151,188},{141,149,186},{143,151,188},{142,152,188},{137,146,185},{135,144,183},{130,143,187},{128,141,185},{124,137,181},{120,133,177},{113,130,174},{106,127,174},{98,122,170},{90,116,165},{84,111,166},{80,111,165},{77,109,160},{78,105,158},{75,104,160},{71,105,166},{71,105,166},{70,104,165},{78,101,153},{82,99,143},{91,101,136},{84,80,103},{51,38,55},{29,17,21},{56,44,44},{82,73,66},{18,11,5},{13,6,0},{8,4,1},{3,1,2},{4,2,5},{7,2,6},{11,5,9},{14,5,6},{11,2,3},{12,4,2},{16,8,6},{21,11,10},{15,5,4},{15,7,5},{15,7,5},{8,0,0},{17,9,6},{17,9,6},{14,6,4}},<<109>>}
eagle8	{{{64,96,128},{64,96,128},{64,96,128},{64,96,128},{64,96,128},{64,96,128},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,128},{64,96,128},{64,96,192},{64,96,192},{64,96,128},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,192},{96,96,192},{96,96,192},{96,96,192},{96,96,192},<<82>>,{64,96,192},{64,96,192},{64,96,192},{64,96,192},{64,96,128},{96,96,128},{96,96,128},{96,96,128},{64,32,64},{0,0,0},{64,32,64},{96,64,64},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0}},<<108>>,{<<1>>}}
eagle4	{{{128,128,128},<<88>>,{128,128,128},{128,128,128},{128,128,128},{128,128,128},{128,128,128},{0,0,0},{0,0,0},{0,0,0},{128,128,128},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0}},<<108>>,{<<1>>}}
eagle1	{{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},<<108>>,{0,0}}

REFERENCES

[1] C. Maple, "Security and Privacy in the Internet of Things", Journal of Cyber Policy, vol. 2, no. 2, pp. 155-184, 2017. DOI:10.1080/23738871.2017.1366536

[2] O. Oyman and S. Singh, "Quality of Experience for HTTP Adaptive Streaming Services", IEEE Communications Magazine, Vol. 50, No. 4, pp. 20-27, Apr. 2012. DOI: 10.1109/MCOM.2012.6178830

[3] Microsoft, Smooth Streaming, [Online]. Available: *http:// www.iis.net/downloads/smooth-streaming/*

[4] William Diehl, Farnoud Farahmand, Panasayya Yalla, Jens- Peter Kaps, Kris Gaj, "Comparison of hardware and software implementations of selected lightweight block ciphers", 27th International Conference on Field Programmable Logic and Applications (FPL), Sept. 2017. DOI: 10.23919/FPL.2017.8056808

[5] M. A. Rajan, Ashley Varghese, N. Narendra, Meena Singh, V. L. Shivraj, Girish Chandra, Balamuralidhar P., "Security and Privacy for Real Time Video Streaming Using Hierarchical Inner Product Encryption Based Publish-Subscribe Architecture", 30th International Conference on Advanced Information Networking and Applications Work shops (WAINA), March 2016. DOI: 10.1109/WAINA.2016.101

[6] Venčkauskas, A., Morkevicius, N., Bagdonas, K., Damaševičius, R., Maskeliūnas, R., "A Lightweight Protocol for Secure Video Streaming", Sensors 18(5), 2018. DOI: 10.3390/s18051554

[7] Adobe, HTTP Dynamic Streaming, [Online]. Available: *http:// www.adobe.com/products/httpdyna-micstreaming/*

[8] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP- Standards and Design Principles", Proc. of the ACM Conference on Multimedia Systems, pp. 133-144, Feb. 2011. DOI: 10.1145/1943552.1943572

[9] Apple, HTTP Live Streaming, [Online]. Available: *http:// developer.apple.com/rescources/http-streaming/*

[10] C. Wang, A. Rizk, and M. Zink, "SQUAD: A Spectrum-based Quality Adaptation for Dynamic Adaptive Streaming over HTTP," Proc. of the International Conference on Multimedia Systems, May 2016. DOI: 10.1145/2910017.2910593

[11] Rainer A. Rueppel, "Analysis and design of stream ciphers", Springer-Verlag, NewYork, 1986.

[12] X. Gu and M. Zhang, "Uniform random number generator using Leap-Ahead LFSR architecture", 2009 Int'l Conf. on Computers and Communication Security, pp. 150-154, 2009. DOI: 10.1109/ICCCS.2009.11

[13] Encrypted Media Extensions, Available From: *https://www.w3.org/TR/encrypted-media/*

[14] R. Lidl and H. Niederreiter, "Introduction to Finite Fields and Their Application", Cambridge University Press, Cambridge, 1986.

[15] A. D. Porto, F. Guida and E. Montolivo, "Fast Algorithm for Finding Primitive Polynomials over GF(q)", Elect. Lett., B28. (2), pp. 118-120, 1992. DOI: 10.1049/el:19920073

[16] SNI, *https://en.wikipedia.org/wiki/Server_Name_Indication*

Authors

Seon-Keun Lee received the B.S., M.S. and Ph.D. degrees in Electronics Engineering from Wonkwang University, Korea, in 1995, 1997 and 2003, respectively. He is currently a Professor in the Department of Electrical and Electronics, Woosuk University. He is interested in IoT, embedded system, H/W cryptographic system design, and various processor designs.