감시 카메라 영상의 보호를 위한 실시간 암호화 구현

이성연* · 김종남*

*부경대학교 컴퓨터공학과

Implementation of real-time encryption for forgery protection of surveillance camera

Seong-Yeon Lee* and Jong-Nam Kim*

*Department of Computer engineering, PuKyong National University

E-mail : sylee9997@pknu.ac.kr

요 약

감시 카메라는 보안, 감시 등의 목적으로 자주 사용된다. 최근의 감시 카메라는 자체적으로 인코딩 작업을 통해 H.264 영상을 생성하여 보내주기도 하는데, H.264는 높은 압축률으로 폭넓게 사용되는 인코딩 방법이다. 감시카메라는 주로 보안을 위해 사용하므로 위/변조에 강해야 하고 허가받지 않은 사용자가 봐서는 안된다. 대개의 멀티미디어 콘텐츠의 경우 별도의 보안 장치를 설정하기 힘들어 보호에 취약한 면이 있다. 본 논문에서는 이러한 피해를 줄이기 위하여 감시카메라 영상을 암호화 알고리즘을 이용하여 일부 또는 전부를 암호화하는 방법을 구현하였다. 감시카메라로 녹화된 영상을 H.264로 인코딩한 뒤 콘텐츠의 특성을 이용하여 암호화를 부분적으로 혹은 전체를 적용하면 암호화된 콘텐츠를 얻을 수 있다. 본 논문에서 제안하는 내용은 최근 많이 사용되는 감시 카메라 영상의 보호 뿐 아니라 그 외 DMB, 지상파 방송 등의 보편적 서비스에도 적용이 가능하다.

키워드

감시카메라, H.264, 암호화

Ⅰ. 서 론

최근 사회적 요구와 개인의 수요까지 맞물려 감시카메라의 보급이 활발히 이루어지고 있다. 길거리는 물론 교차로, 골목길, 가게 안, 건물 안팎에 이르기까지 감시카메라는 실생활 곳곳에서 찾아볼 수 있다. 이러한 감시카메라 영상은 여러 가지 목적에 맞게 사용된다. 감시 및 보안, 채증, 교통량 분석 등 여러 분야에서 사용되는데 이러한 감시카메라의 영상은 사회의 각 분야에서 유용하게 사용되고 있다.

그 중 가장 보편적으로 사용되는 것은 감시카메라 영상을 보안 및 감시의 목적으로 사용하는 것이다. 범죄 현장을 지킨다던지 사람이 접근하기 어려운 곳을 원거리에서 비춰 필요한 영상을 획득하는 등의 목적으로 사용된다. 최근의 감시카메라 활용 동향은 기존의 폐쇄식 카메라 시스템에서 벗어나 네트워크 등을 활용한 원거리에서 직접 접근 등이 가능하도록 발전하고 있다. 감시카메라 감시자는 카메라실이 아닌 다른 곳에서도 카메라 영상을 보길 원하고, 감시카메라 운영 주체 역시 자신의 위치에서 감시카메라 영상을 볼 수 있기를 바라기 때문이다. 최근 이러한 수요에 맞추어 원격지에서 인터넷 또는 휴대전화 네트워크를 이용한 CCTV 영상 수신 시스템이 활발히 보급되고 있는 것도 이러한 카메라의 발전 동향과 맞아 떨어지는 것이다.

그러나 이러한 원격지에서의 영상 획득은 네트워크 망의 보안 수준에 따라 크래킹(cracking), 스푸핑(spoofing) 등의 해킹 방법을 이용하여 네트워크에 침입 시 감시영상을 허가받지 않은 사용자가 받을 확률이 크다. 따라서 이러한 원격지 감시 시스템의 경우 폐쇄식 카메라 시스템에 비해 카메라 영상의 보안이 더욱 중요하다. 이러한 감시카메라 영상의 보안을 위하여 여러 가지 방법이 시도되고 있다. 워터마크를 삽입하거나 암호화 등의 방법이 사용되고 있다. 본 연구에서는 이러한 감시카메라 영상의 보안을 위하여 최근 광범위하게 사용되고 있는 H.264 방식 감시카메라 영상의 암호화 시스템을 제안하고 구현하였다. 방법

은 감시카메라에서 생성된 H.264 영상의 일부를 암호화하여 접근이 허락된 사용자에게만 영상을 볼 수 있도록 하는 것이다. 암호화 알고리즘은 AES를 사용하였으며 용도에 맞게 사용할 수 있도록 스트림 암호화 알고리즘도 사용하였다. 실험 결과 별도의 시간 소요 없이 실시간으로 암호화 된 영상의 획득이 가능함을 확인하였다.

본 논문의 구성은 다음과 같다. 2장의 관련 연구에서는 현재 사용되고 있는 보안 방식과 H.264 압축 방법의 특징에 대해 설명한다. 3장의 구현 사항에서는 암호화에 대한 구현 사항에 대해 설명하고 4장 실험 및 결과에서는 구현된 시스템의 암호화 결과에 대해 기술한다. 마지막으로 5장에서 결론을 맺는다.

II. 관련 연구

감시카메라 시스템은 두 가지 종류로 구분할 수 있다. 일반적으로 설치되어 사용되는 카메라와 TV 장치가 폐쇄된 회로로 연결되어 다른 곳에서 확인하기 어려운 시스템과 웹, 네트워크 등을 이용하여 TV 수상기, PC, PDA, 휴대폰 등 다른 기기를 이용하여 원격지에서 관측 가능한 시스템이 있다. 두 방식의 차이를 그림 1에 나타내었다.

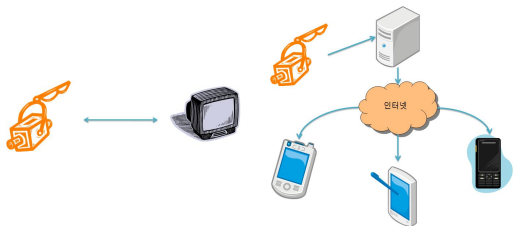


그림 1. 감시카메라 시스템의 두 가지 종류

이러한 시스템은 영상 저장 서버에서 네트워크 등을 통해 영상을 확인할 수 있다는 장점이 있다. 그러나 이러한 환경의 경우 보안성이 떨어진다는 단점이 있다. 폐쇄식 시스템은 외부에서 접근하기 어려워 물리적 접근을 하여야 영상을 확인할 수 있어 보안성이 뛰어나지만 네트워크를 통해 영상이 전송되므로 네트워크 후킹이나 봇 등을 이용한 해킹 범죄에 노출될 경우 영상의 획득이 가능하여 보안성이 높지 않다. 이러한 범죄를 사전에 예방하고자 영상에 보안을 설정하는 방법이 제시되고 있다. 그 방법을 몇가지 소개하자면 첫째, 워터마킹 방법이 있다. 어떤 보이지 않는 신호나 잡음을 영상에 숨겨 이를 이용하는 방법이다 [1]. 이러한 워터마킹 방법은 영상의 보안 보다는 저작권 보호 등에 사용되는 방법이다. 워터마크를 폭넓게 응용하면 제어비트를 섞는 방법으로 보안이 이루어질 수 있으나 근본적으로 좋은 방법은 아니다. 다른 방법은 암호화 방법인데, 암호화는 영상의 일부나 전부를 암호화하여 전송하고 수신단에서는 암호화 된 영상을 복호화하여 재생하는 방법이다. 암호화는 처음 개발될 때부터 보안을 염두에 두고 설계가 되기 때문에 워터마크 방법보다 더 적당한 방법이라 생각된다.

암호화 알고리즘에는 여러 방법이 있다. 암호화 하는 방법에 따라 블록 기반 알고리즘과 스트림 암호화 알고리즘으로 나눌 수 있으며, 키의 공개 유무에 따라 대칭키 알고리즘과 공개키 알고리즘으로 나눌 수 있다[2]. 블록 기반 대칭키 알고리즘에는 DES, AES, SEED 등이 있고 공개키 알고리즘에는 RSA, ECC 등이 있다. 블록 기반 알고리즘은 암호화 평문을 블록 단위로 쪼개어 암호화를 진행한다. DES는 56비트의 키와 64비트의 평문 블록으로 동작하며, AES, SEED는 128비트 키와 128비트 크기의 평문 블록을 형성하여 암호화 및 복호화가 이루어지는데, 블록 크기는 변경하지 못하고 고정된다. 이와 반대로 스트림 암호화는 한 번에 암호화 하는 평문이 고정 길이가 아닌 가변 길이이다. 블록 기반 알고리즘에 비하여 스트림 암호화 알고리즘이 더욱 고속으로 동작한다.

블록 기반 암호화 알고리즘 중 DES는 1997년 해독 방식이 알려지고 고성능 컴퓨터의 발전에 의한 전수 공격에도 안전하지 않음이 알려지면서 현재는 강력한 암호 알고리즘으로 보기 힘들다. 이를 대체하기 위하여 NIST(National Institute of Standard and Technology)에서는 2001년 Rjindael의 알고리즘을 AES로 선정하였다[2]. SEED 알고리즘은 한국의 정보보호진흥원(Korea Information Security Agency, KISA)에서 개발한 128비트 블록 기반 알고리즘이다[3]. 강인하고 신뢰성이 높을 뿐 아니라 로열티도 없다. 그러나 속도는 AES에 비해 뒤쳐진다.

스트림 암호화 알고리즘은 Dragon, F-FCSR, HC-128, Rabbit, MICKEY, Trivinum 등이 있다[4]. 더욱 고속이지만 아직 검증된 암호화 방법은 아니다.

감시카메라의 영상은 압축되지 않은 NTSC 신호로 출력된다. 이는 제한된 대역폭의 디지털 네트워크를 통해 전달하기 어렵다. 특히 대역폭이 매우 좁은 휴대전화용 무선통신망을 통하여 전송할 경우 압축되지 않은 카메라 신호를 그대로 전송할 수 없다. 따라서 별도의 코덱을 사용하여 압축한 뒤 전송하게 된다. 이 때 사용되는 코덱은 MPEG-2, MPEG-4, H.264 등 여러 가지가 있다. 그 중 H.264는 높은 압축률과 네트워크 전송 또는 스트리밍에 적합한 데이터 형태로 인해 널리 사용되고 있다. H.264의 데이터 구조를 아래 그림 2에 나타내었다.

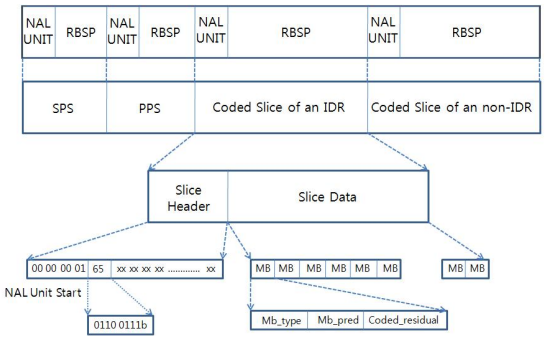


그림 2. H.264 스트림의 구조

III. 구현 사항

시스템을 구현하기 위해서는 우선 H.264 영상의 특성에 대한 이해가 선행되어야 한다. 사용할 영상은 H.264 인코딩 시스템에 따라 다르게 설정된다. 실험에 사용될 H.264 스트림의 경우 30프레임을 하나의 GOP(Group of Pictures)로 설정하였고 각 GOP는 하나의 I-프레임과 29개의 P-프레임으로 구성된다. 이러한 구조는 네트워크 전송 시 잡음이나 신호 손실 등에 따른 패킷 손실의 가능성에 대비하기 위한 것이다. 네트워크 등을 통해 패킷이 손실될 경우 뒤따르는 P-프레임은 데이터가 손상되기 쉽다. 최악의 경우 첫 번째 I-프레임이 손실될 경우 뒤따르는 P-프레임이 모두 손실될 수 있다. 이러한 경우를 예방하고자 여러개의 GOP로 나누어 각 I-프레임이 손실되더라도 하나의 GOP만 손실되도록 하였다. I-프레임의 손상에 따른 P-프레임의 손실은 암호화에도 적용할 수 있다. 의도적으로 I-프레임을 손상시킬 경우 뒤따르는 P-프레임도 손상시킬 수 있기 때문이다.

암호화를 프레임 전체에 적용할 경우 암호화에 소요되는 시간이 오래 걸리게 된다. 따라서 암호화에 소요되는 시간을 줄이기 위해 암호화가 적용되는 부분을 줄일 필요가 있다. 따라서 각 프레임의 가장 첫 번째 데이터만을 암호화한다. 128비트 블록 암호 알고리즘의 단위 크기만큼인 128비트, 16바이트를 암호화하였다. 암호화 하는 데이터는 영상의 일부에 지나지 않는다. 하지만 이것만으로도 전체 프레임을 정상적으로 출력되지 않게 하는 것은 충분하다. 첫 부분의 작은 데이터 변화는 디코딩시의 허프만 코딩, RLE, DCT, Quantization이 역으로 적용되는 디코딩 과정에서 큰 왜곡을 발생시키기 때문이다 [5].

그림 3은 암호화를 적용하는 부분을 나타낸다. 암호화를 적용하기 위하여 각 프레임의 시작 부분을 찾아내는 것이 필요하다. 이는 스트림의 분석을 통하여 알아낼 수 있는데 그 방법은 다음과 같다. 우선 H.264 스트림에서 헤더 정보를 제외한 매크로블록(MacroBlock, MB) 부분만 암호화가 적용되므로 매크로블록을 찾아야 한다. 매크로블록은 Slice Data가 포함된 RBSP(Raw Byte Sequence Payload)에 위치하므로 우선 NAL Unit의 시작패턴을 찾고 뒤이어 나오는 nal_unit_type 정보를 이용하여 RBSP에 담긴 데이터가 어떠한 슬라이스인지 알 수 있다. 이를 이용하여 첫 번째 MB데이터를 찾아 암호화를 적용한다. I/P 프레임에 암호화를 적용하는 경우 동일한 방법을 이용하여 각 첫 번째 매크로블록 데이터로부터 16바이트를 찾아서 암호화한다.

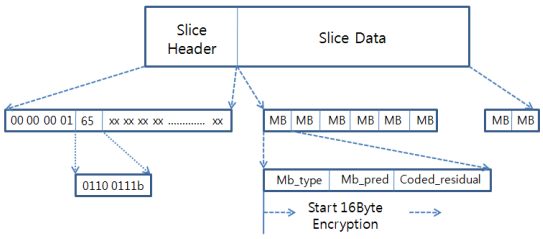


그림 3. 암호화를 적용하는 부분

사용한 알고리즘은 AES로 강건성이 뛰어나고 암호화와 복호화에 소요되는 시간이 적으며, 로열티가 없어 암호화 시스템에 폭넓게 적용되고 있다. 전체 암호화에는 스트림 암호화 알고리즘을 사용하였는데 스트림 암호화 알고리즘은 블록 암호 대비 3배 가량 빠른 성능을 보인다. 이러한 고속 성능을 살려 스트림 암호화 알고리즘은 더 많은 부분을 암호화한다. 블록 암호와 마찬가지로 I-프레임 뿐 아니라 P-프레임의 데이터도 암호화를 적용하며 암호화 하는 부분은 블록 암호화 알고리즘의 128비트보다 더 많은 1024비트(128바이트)를 암호화하도록 하였다. 컴퓨터 성능이 충분하다면 스트림 전체를 암호화 할 수도 있을 것이다. 사용된 스트림 암호화 알고리즘은 F-FCSR으로 개발 그룹인 eStream Project에서 개발 코드 및 사용법을 공개하여 편리하게 사용할 수 있다.

IV. 실험 및 결과

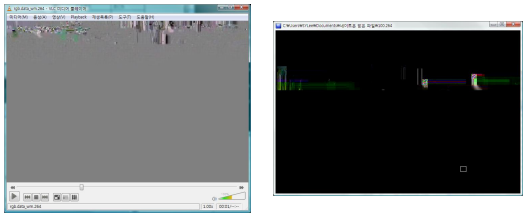
구현 및 실험에 사용된 시스템은 Intel Pentium4 2.8Ghz, 2GB Ram, Windows XP 사양이며 Microsoft Visual Studio 2008을 이용하여 프로그램을 작성하였다. H.264 영상을 인코딩하는데 VideoLan의 X264를 이용하였다. X264 영상의 재생을 위하여 Elecard 의 EStreamEYE와 VideoLan의 VLC Player를 이용하였다.

감시카메라 영상의 획득을 위하여 삼성테크윈의 SNC-550 카메라를 사용하였으며 H.264 압축 및 네트워크 전송용 서버 시스템을 구축하였다. 서버에서 영상 캡쳐를 하기 위해 USB타입 캡쳐

보드를 사용하였다.

아래 그림 4는 원본 영상을 VLC 플레이어로 재생한 모습을 나타내며 그림 5는 암호화 결과로 불법으로 유출된 파일을 재생할 경우를 재현하였다.

그림 4. 원본 파일 재생



(a) 블록 암호화 (b) 스트림 암호화

그림 5. 암호화 결과

암호화를 적용할 경우 위 그림과 같이 손상됨을 확인할 수 있다. 또한 암호화와 복호화가 고속으로 동작하므로 실시간 처리가 가능하다. 암호화에 소요되는 시간의 경우 128비트 암호화를 할 때 3회 평균 소요시간의 경우 AES는 6.87μs가 소요되었고 F-FCSR의 경우 1024비트 암호화에 16.82μs가 소요되었다.

현재 실험은 녹화된 파일을 이용하여 암호화 실험을 진행하였는데 실제 스트리밍 환경에 적용하지 못하였다. 이러한 점은 추후 연구 내용에 반영할 예정이다.

Ⅴ. 결 론

본 논문에서는 감시카메라 영상의 보안을 위하여 영상을 암호화 하는 방법을 제안하고 구현하였다. 그 방법은 H.264로 압축된 영상 데이터에서 각 프레임의 데이터 일부를 암호화하여 영상을 보호하는 것이다. 실험 결과 암호화가 적용된 영상을 전송 도중 불법으로 취득하여 재생할 경우 암호화가 적용되어 복호화 없이는 재생이 불가능함을 확인할 수 있었고 암호화에 소요되는 시간도 짧아 실시간으로 적용 가능함을 확인할 수 있었다. 본 논문에서 제안한 내용은 감시카메라 영상의 보안 뿐 아니라 유료 케이블 방송, 위성 영상 등 기타 보안이 필요한 영상 서비스에 적용할 수 있을것이다.

감사의 글

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신 인력양성사업, 중소기업청의 산학연 공동기술개발지원사업(선도형)의 지원으로 수행되었음.

참고문헌

[1] I. Cox, M. Miller, and J. Bloom, "Digital watermarking," Press of Morgan Kaufmann, SanFrancisco, Oct .2001.

[2] W. Stalling, Cryptography and network security : principles and practice 3/E, Prentice Hall, New Jersey, 2002.

[3] “128비트 블록 암호알고리즘(SEED) 개발 및 분석 보고서,” 한국정보보호진흥원, 2003.

[4] The eSTREAM Project http://www.ecrypt.eu.org/stream/

[5] 김건희, 신동규, 신동일, “효율적인 MPEG-4 비디오 파일의 암호화에 관한 연구,” 한국정보과학회 추계학술발표논문집, 제31권, 제2호, pp. 631-633, 2004.

[6] W. Zeng, “Format-Compliant Selective Scrambling for Multimedia Access Control," in Proceedings of IEEE ICASSP, pp. 77-80, 2002.

[7] Prasertsatid, N. “Implementation conditional access system for pay TV based on Java card,” Proceedings of IEEE. ICCEA, pp 533 - 536 2004

[8] Noore, A, "A secure conditional access system using digital signature and encryption," Consumer Electronics, 2003. ICCE. 2003 IEEE International Conference on pp. 220 - 221 2003.