모바일 콘텐츠의 안전한 부분암호화 방법에 대한 연구

유경인○ †, 김민재†, 이진영‡, 조성제‡, 김준모‖
단국대학교 컴퓨터학과† 단국대학교 정보컴퓨터학부‡ 단국대학교 전자컴퓨터공학부‖
eras307@hotmail.com

A Study on Secure Partial Encryption for Mobile Contents

Kyung-In Ryu○† Min-Jae Kim†, Jin-Young Lee‡, Seong-je Cho‡, Jun-Mo Kim‖
Dept. of Computer, Dankook University† Dept of Computer Science, Dankook University‡
Dept. of Computer Engineering, Dankook University‖

요 약

모바일 인터넷 사용자가 급속히 늘어남에 따라 모바일 콘텐츠의 수요도 증가하고 있다. MP3, 온라인 게임, 비디오 클립 등 지적재산권이 있는 유료 콘텐츠를 보호하기 위해 일반적으로 모바일 DRM과 같은 암호화 방식이 적용된다. 하지만, 자원이 제한된 모바일 환경에서 AES 알고리즘 등으로 콘텐츠 전체를 암호화할 경우, 응답시간 지연과 전력소비 증가로 효율적 모바일 콘텐츠 서비스를 제공하기 어렵다. 이러한 문제를 해결하기 위해, 본 논문에서는 모바일 콘텐츠를 고정크기 분할(fragment)들로 나눈 다음 각 분할의 앞·뒤 부분만 암호화하는 효율적인 부분 암호화(partial encryption) 기법을 제안한다. 또한, 부분 암호화로 인한 안전성 감소 가능성을 보완하기 위하여 분할들에 대해 뒤섞기(shuffling)를 적용한다. 제안한 개념을 모바일 DRM 표준 블록 암호화 알고리즘인 AES를 사용하여 ARM 기반 임베디드 보드에서 구현하여 실험하였다.

1. 서 론

2008년 4월 8일 GSM 협회에 따르면 전 세계 모바일 브로드밴드(HSDPA) 가입자 수가 1년 새 10배 이상 증가하였으며, "초고속 모바일 인터넷의 효용을 경험한 사용자들이 모바일로 음악과 비디오를 내려받고 인터넷에 접속하는 일이 자연스러워졌다"[1]. 또한 사람들은 모바일 휴대기기를 사용하여 멀티미디어 음악, 동영상, 벨소리, 전자 서적과 같은 콘텐츠를 꾸준히 소비해 나가고 있으며 콘텐츠 소비량은 점차 증가할 전망이다[2]. 이에 따라 모바일 기기에서 디지털 콘텐츠의 지적 재산권을 관리하기 위한 모바일 디지털 저작권 관리(Mobile Digital Rights Management)가 중요해지고 있으며 OMA(Open Mobile Alliance)와 같은 표준화 단체가 결성되어 모바일 DRM 표준을 제시하고 있다.

디지털 저작권 관리 기술은 권한을 가진 합법적인 사용자만 디지털 콘텐츠를 사용할 수 있도록 하며, 이때 디지털 콘텐츠는 암호화되어 저장되거나 전송된다. 현재 디지털 저작권 관리는 콘텐츠를 대칭 키 기법으로 암호화하고 콘텐츠에 대한 권리를 명세한 권리 객체를 사용자의 공개키로 암호화하여 전달하는 방식을 사용하고 있다. 비 대칭키 암호화 기법은 매우 큰 계산량이 필요하기 때문에 멀티미디어 데이터와 같은 크기가 큰 콘텐츠를 암호화할 경우 응답 속도가 크게 느려질 뿐 아니라 전력의 소모도 크므로, 키 분배 및 디지털 서명 등의 용도로 사용되고 있다.

이처럼 모바일 단말기들은 데스크탑 환경의 컴퓨터보다 제한된 자원 및 전력 환경에서 동작하기 때문에 모바일 휴대기기에서 연산시간과 에너지 소모를 최대한 줄이면서 콘텐츠를 안전하게 보호할 수 있는 암호화 알고리즘이 요구된다. 계산 비용과 에너지 소모를 줄이면서도 디지털 저작권 관리의 사용 제어를 만족하기 위한 방법으로 선택적 암호화(Selective Encryption)기법이 연구되어 왔다. 초기 선택적 암호화는 미디어 데이터를 압축할 때 압축 알고리즘과 암호화 알고리즘을 적절히 결합하여 특정 부분만 암호화하여 계산량을 줄였다. 그러나 이 선택적 암호화는 데이터 압축 알고리즘을 완벽히 이해해야 한다는 단점이 있다.

이와 별도로 미디어 데이터의 경우, 일반 텍스트 데이터와 달리 서비스 품질 면과 연결시켜 데이터의 일부분만 암호화하는 부분 암호화 기법이 적용될 수 있다. 디지털 저작권 관리는 기밀성 보장과 더불어 사용 제어를

요구하기 때문에 일부 데이터의 내용이 노출되더라도 노출된 데이터의 양이나 품질의 사용에 지장을 주기만 하면 된다. 멀티미디어 데이터의 일부분만을 암호화하거나 변조시켜 정상적으로 복호화하지 않고 재생할 경우 멀티미디어 데이터의 품질을 청취·시청하기에 적합하지 않도록 만드는 기법이 부분 암호화 기법이다[9].

본 논문에서는 모바일 콘텐츠의 효율적인 암호화를 위해 모바일 콘텐츠 파일을 분할(fragment)[1]들로 분할하여 각 분할들의 일부분을 AES (Advanced Encryption Standard) 알고리즘으로 암호화하는 부분 암호화 기법을 제안한다. 각 분할의 크기는 64byte 크기로, 각 분할에서 앞 16byte와 뒤 16byte만을 AES로 암호화하고 중간의 32B는 암호화하기 않는다. 이렇게 하여 암호화하는 부분을 줄이고, 분할들 간의 연속성을 없애주어 유추를 방지하였다. 또한 안전성을 높이기 위해, 부분 암호화 후에 뒤섞기(shuffling) 방식을 추가하였다. 제안한 방법을 ARM 프로세서 기반 임베디드 보드에서 구현하여 성능을 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 선택적 암호화 기법과 블록암호화 방식인 AES의 특징을 살펴보고, 3장에서는 모바일 환경 콘텐츠 암호화에서 응답속도와 에너지 효율을 높이기 위한 AES에서의 부분적 암호화와 이를 보완하기 위한 뒤섞기 방법을 기술한다. 4장에서는 실제 임베디드 보드에서의 암호화 방식에 대한 성능 측정을 하였으며, 5장에서는 결과를 토대로 결론과 향후 계획을 기술한다.

2. 관련 연구

2.1 블록 암호화

일반적으로 블록 암호화 기법으로 암호화 표준 알고리즘인 AES(Advanced Encryption Standard)가 강한 보안성과 적절한 암호화 속도를 가져 콘텐츠 암호화에 사용되고 있다. 모바일 휴대기기에서도 디지털 저작권을 관리하기 위하여 AES를 콘텐츠 암호화 알고리즘으로 사용하고 있다. 하지만 모바일 휴대기기와 같은 제한된 환경에서 AES를 이용하여 콘텐츠 전체를 암호화 하는 것은 많은 전력소비를 가져온다.

DES를 비롯한 대부분의 대칭키 암호 시스템들은 Feistel[7] 구조의 라운드 변환을 기반으로 하는데 비해 AES 알고리즘은 Feistel구조를 채택하지 않으며, 4개의 독립된 역변환 가능한 라운드 변환으로 구성된다. AES 알고리즘은 블록 길이를 128비트로 고정하고, 3가지 키 길이 128, 192, 256비트를 사용한다. 암호에 필요한 라운드 수는 키 길이(Nk)에 따라 라운드 수(Nr)는 10, 12, 14로 다른 값을 갖게 된다. AES 대칭키 암호 알고리즘의 연산 처리과정은 그림 1과 같이 초키 라운드 키 가산(AddRound-Key)후에 (Nr-1)번의 반복 라운드를 수행한 후 MixColumn 변환이 제외된 최종라운드 순으로 진행된다.

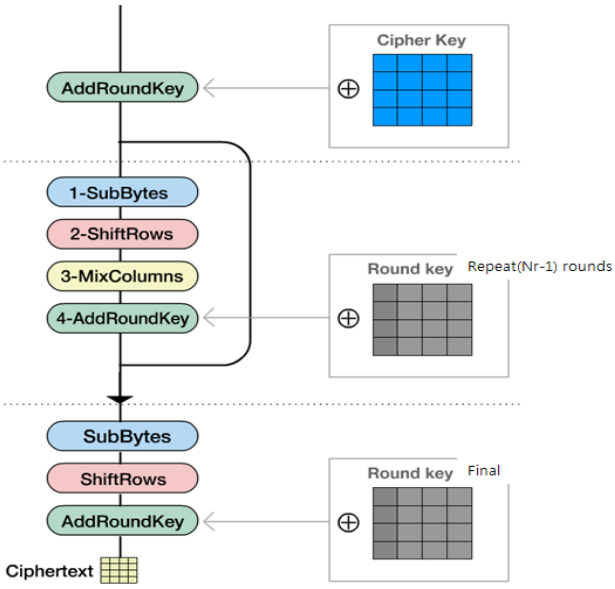

그림 1. AES 암호 알고리즘의 연산 처리과정[8]

2.2 선택적 암호화 및 부분적 암호화 기법

모바일 단말기와 같은 자원이 제한된 임베디드 시스템에서 대용량 멀티미디어 데이터를 전체 암호화를 적용할 경우 계산 비용과 에너지 소모를 무시할 수 없다. 이를 해결하기 위한 방법으로 선택적 암호화 기법이 연구되어 왔다. 즉, 데이터 압축(또는 인코딩) 시에 특정 부분을 선택하여 암호화 알고리즘을 적용하여 계산 량을 줄이는 것이다.

데이터 압축에 사용하는 허프만 테이블을 여러 개 사용하고 사용한 테이블을 암호화하는 기법[4], 동영상 압축의 변환계수를 블록 혹은 세그먼트로 나눈 다음 일부 비트를 뒤섞거나 블록들의 순서 변경, 회전을 통해 정상적인 영상이 출력되지 않게 하는 기법[5] 등 다양한 선택적 암호화 기법이 제안되었으며 이들은 모두 적은 비용으로 멀티미디어 데이터를 암호화할 수 있다. 선택적 암호화의 경우, 압축(인코딩) 알고리즘을 완벽히 이해해야

1) AES 블록 암호 알고리즘에서의 블록과 혼동을 줄 수 있으므로 균등하게 나누어진 각 데이터 부분을 '분할'(fragment)이라고 한다.

한다는 단점이 있고, 또 전체 파일의 10%만 암호화하였지만 속도 개선은 전체 암호화의 50%에 그치는 등의 비효율성이 문제가 된다[6].

부분 암호화의 경우, 미디어 데이터가 인코딩된 후에 일부분만 암호화하여 배포하고, 미디어 재생 시에 해당 부분만 복호화하는 방식으로 압축 알고리즘을 알지 못해도 된다는 장점이 있고, 부분 암호화로 인해 성능이 개선되었다. 하지만, 암호화되지 않은 평문 부분으로부터 암호화 된 부분의 데이터를 추리해 낼 수 있는 경우가 많아 보안성이 충분하지 않다는 단점이 있다.

3. 안전한 부분 암호화 방식

블록 암호화 알고리즘인 AES는 안전하지만, 구조적 특성으로 인해 긴 연산시간과 많은 전력소모로 인하여 모바일 휴대 기기에서 대용량 파일을 전체 암호화하기에 적합하지 않다. 본 논문에서는 지적재산권에 민감한 대용량 모바일 DRM 콘텐츠에 대해 효율적이면서 안전한 부분 암호화와 뒤섞기 방식을 결합하여 적용한다.

3.1 부분 암호화

보통 부분 암호화를 사용할 때, 파일들을 분할(fragmentation)한 후 홀수 번째 분할들만 암호화하고 짝수 번째 분할들은 암호화하지 않는 방식을 사용한다(그림 3.a). 또는, 파일의 헤더 부분만을 암호화하는 방식을 사용한다. 이러한 경우 암호화되지 않은 평문 분할을 이용하여 이웃한 분할의 정보를 유추할 수 있고, 암호화된 헤더부분을 암호화되지 않은 다른 파일의 헤더로 바꿔치기 했을 때 정상적으로 실행되는 경우도 있다.

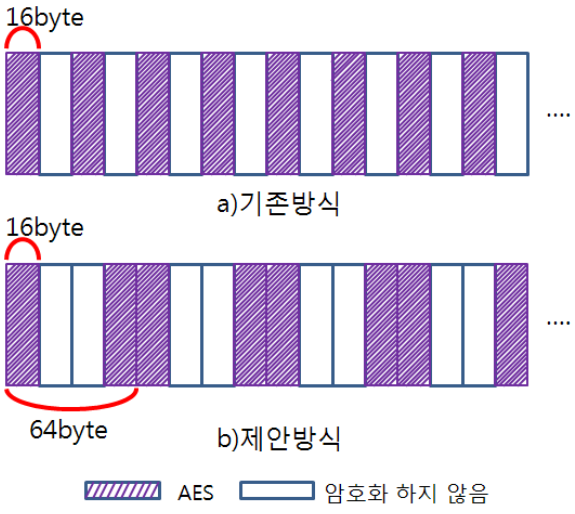

그림 2 선택적 부분 암호화 방법

본 논문에서는 그림 3.b와 같이 64byte를 하나의 분할 크기로 정하고 그중 앞뒤 각각 16byte 크기만큼만 AES 암호화 알고리즘으로 암호화시킨다. 이로써 이웃한 세그먼트를 유추하기 힘들어지게 했으며, 그대로 재생할 경우 서비스 품질이 저하되게 된다.

부분 암호화 방식에서, 암호화되지 않은 부분의 정보를 이용하여, 암호화된 부분을 추론하는 것이 가능하지만 현재까지 AES는 안전하다고 알려져 있다. 또한, 본 논문에서는 추가적으로 부분 암호화 후에 뒤섞기를 수행한다.

3.2 부분 암호화 및 뒤섞기의 결합

멀티미디어 파일을 부분적으로 암호화하여 전체 암호화 한 것보다 속도를 향상시킬 수 있다. 여기에 더불어 보안성을 강화시키기 위해 뒤섞기(shuffling) 방법을 결합 사용하였다[9, 10]. 뒤섞기가 되는 단위는 AES 블록길이(128bit)를 고려하여 64byte(분할 단위)로 하였으며, 뒤섞기 단위는 융통성 있게 적용 가능하다.

뒤섞기 방법에서는 모바일 단말기마다 서로 다른 방법을 적용하는 것이 중요하다. 한 단말기에서 뒤섞기 알고리즘이 분석되었다 하더라도 다른 단말기에서는 뒤섞기 알고리즘이 유지되게 하는 것이 필요하다. 휴대폰에서 적용 가능한 뒤섞기 알고리즘에 사용될 정보로는ESN 정보나 각 가입자의 휴대폰 번호 등이며, 안전을 위해서는 각 모바일 단말기마다 해당 정보를 서버에서 생성하여 암호화하여 전달하는 것도 가능하다.

뒤섞기 방법의 적용 예가 그림3에 나타나 있다. 부분 암호화 된 파일을 64byte 단위로 분할하여 (n+1)개로 균등하여 나눈다. 그 다음, 부분 암호화를 적용한 후, (n+1)번째 분할 F_{n+1} (마지막 분할)을 제외한 분할들을 규칙에 맞게 섞어준다[10]. 뒤섞기에 대한 자세한 방법은 [9, 10]을 참조 바란다.

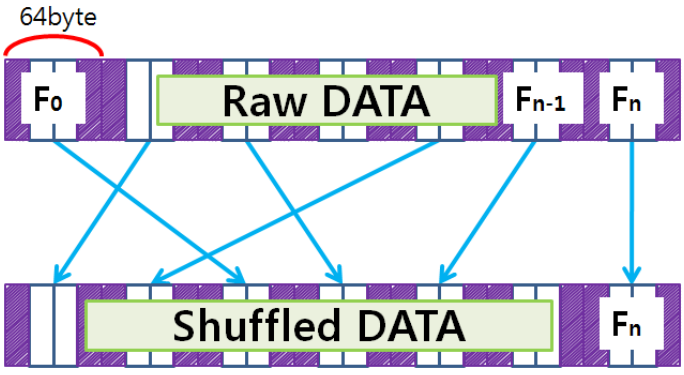

그림 3 뒤섞기 방법

이렇게 부분 암호화 후에 뒤섞기를 적용함으로써 단순한 뒤섞기 방식에서보다 원래 분할들의 순서를 알아내는 공격에 효율적으로 방어할 수 있다. 즉, 부분 암호화된 n개의 분할이 뒤섞여 있을 때, 공격자가 이를 완전히 해독하기 위해서는 먼저 n! 개의 경우의 수를 조사하는 것이 필요하다.

제안한 방식은 홀수 번째 분할만을 암호화하여 뒤섞기를 수행한 방식에 비해, 매 분할마다 분할의 앞, 뒤가 부분 암호화되어있어 분할들의 연속적 위치를 확인하기 어렵게 함으로써 각 분할의 안전도가 좀 더 뛰어나다고 볼 수 있다.

4. 실험 및 결과

본 논문에서 제안한 방법을 Intel Xscal PXA255 400MHz CPU, SDRAM 128Mbyte, Flash ROM 32Mbyte, Embedded Linux(커널 2.4.19)에서 구현하여 실험하였다.

4.1 뒤섞기의 유무에 따른 성능 분석

콘텐츠를 AES로 전체 암호화하여, 뒤섞기를 수행한 것과 뒤섞기를 적용하지 않은 것을 콘텐츠의 크기에 따라 시간을 측정하였다. 그 결과 그림 4와 같이 뒤섞기를 수행할 때와 안할 때의 암·복호화시간이 거의 비슷하게 나왔다. 즉, 뒤섞기는 무시할 수 있을 정도로 수행시간이 빠르지만, 공격자의 입장에서는 뒤섞기를 적용하지 않았을 때보다 뒤섞기를 적용한 것이 콘텐츠를 재생하기 훨씬 더 어렵다.

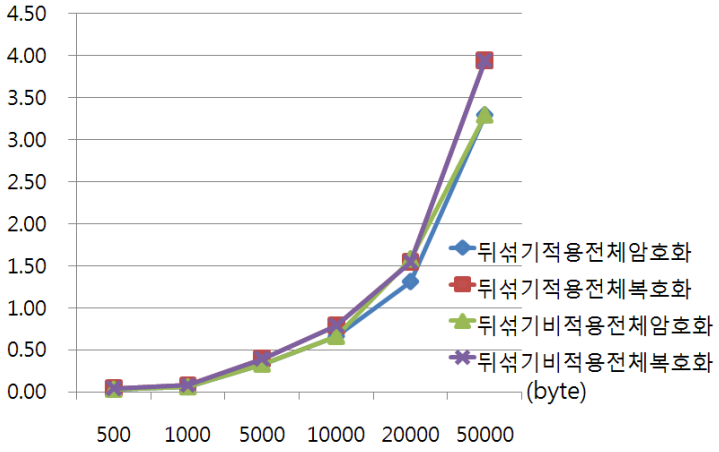

그림 4 전체 암호화 후 뒤섞기에 따른 성능차이

그림 4에서, "뒤섞기비적용전체암호화"는 암호화만 적용한 시간(뒤섞기 연산이 포함되지 않은 시간)을 나타내며, "뒤섞기적용전체암호화"는 암호화 후에 뒤섞기를 적용한 시간을 나타낸다.

4.2. 부분 암호화 + 뒤섞기 방법의 성능분석

뒤섞기 방법이 실행 시간에는 크게 영향을 주지 않으면서 보안에는 효과적이다. 본 절에서는 암호화 후에 뒤섞기 방식을 적용하되, 전체 암호화 후 뒤섞기를 수행한 것과 부분 암호화 후 뒤섞기를 수행한 것을 비교 평가하였다. 그림 5와 나타난 결과처럼, 전체 AES 암호화와 뒤섞기를 적용했을 때 보다, 부분 AES 암호화와 뒤섞기를 적용했을 때 성능이 약 두 배 정도 향상되었다. 결론적으로, 제안한 방법이 임베디드 기기에 효율적으로 적용될 수 있음을 알 수 있다.

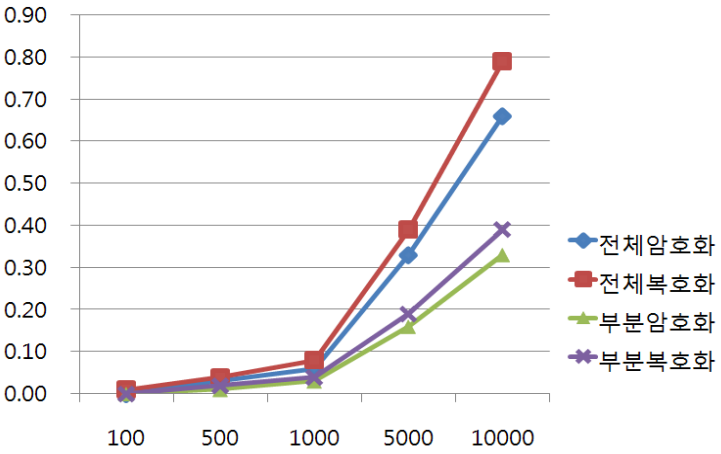

그림 5 암호화와 뒤섞기가 결합되었을 때 성능

5. 결론 및 향후 계획

유비쿼터스 컴퓨팅 시대로 들어서게 되면서, 여러 멀티미디어 콘텐츠를 사용할 수 있는 휴대용 장치들의 이용이 증가하게 되어 모바일 콘텐츠의 수요가 급증하고 있다. AES 블록 암호화의 경우 그 안전성이 인정되어 현재 널리 쓰이고 있지만, 자원이 제한된 임베디드 시스템에서 대용량 멀티미디어 데이터를 암호화할 경우 긴 암호화 연산시간과 많은 배터리 소모 문제를 해결하는 것이 필요하다.

본 논문에서는 임베디드 기기에서AES 암호화 방법을 효율적으로 사용하기 위해서 부분AES 암호화와 뒤섞기

방법을 같이 적용하여 암호화의 안전성은 유지하면서 속도 및 전력사용 효율을 향상시킬 수 있는 방법을 제안하고 실험하였다. 현재는 부분 암호화 연산으로 감소된 계산량으로 인해 전력소모가 얼마나 줄었는지를 실험하고 있다.

향후, 스트림 암호화 방식인 RC4 알고리즘 기반의 부분 암호화의 효과, AES로 일부 암호화와 RC4로 나머지 암호화하는 방식의 결합하는 방식에 대해 연구하고 이에 따른 전력소모 량에 대해 측정해볼 계획이다.

참 고 문 헌

[1] 류현정, 모바일 인터넷 인구, 1년새 10배 증가, 전자신문 20080409.

[2] N. Dufft, A. Stiehler, D. Vogeley, and T. Wichmann, "Digital Music Usage and DRM", results from and European Consumer Survey, 2005

[3] 한정규, "유사 랜덤생성기와 순열 풀을 결합한 효율적인 모바일 멀티미디어 데이터 암호화 기법, 서울대학교 대학원 석사학위논문, 2007. 2

[4] C. P. Wu and J. Kuo, "Design of integrated multimedia compression and encryption systems", IEEE Transactions on Multimedia Vol.7, No.5, 2005

[5] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video", IEEE Transactions on Multimedia, Vol.5, 2003

[6] X. Liu and A. M. Eskicoioglu, "Selective encryption of multimedia content in distribution networks: Challenges and new directions", 2nd IASTED International Conference on Communications, Internet, and Information Technology, 2003

[7] http://en.wikipedia.org/wiki/Feistel_cipher

[8] http://www.cs.bc.edu/~straubin/cs381-05/blockciphers/rijndael_ingles2004.swf

[9] 윤성환 "Rijndael 암호 알고리즘을 이용한 부분 암호화 기법에 관한 연구" 대전대학교 대학원 석사학위논문, 2005. 2

[10] 오현수 "휴대폰 벨소리의 접근제어 및 사용제어에 대한 연구" 단국대학교 대학원 석사학위논문, 2006. 6