휴대폰 벨소리 보호 기법에 관한 연구

오현수*, 최종천*, 조성제*
*단국대학교 정보 컴퓨터과학과
e-mail : pyxis3@hanafos.com, {godofslp, sjcho}@dankook.ac.kr

A Study on a Protection Scheme of Ring-tone on Mobile Phone

Oh Hyunsoo*, Choi Jong-chun*, Cho Seong-je*
*Div. of Information & Computer Science, Dankook Univ.

요 약

최근 무선 인터넷이 가능하고 엄청난 성능을 발휘하는 핸드폰의 보급으로 인해 이제는 다양한 모바일 콘텐츠가 필요로 되고 있다. 그래서 통신사는 게임, 음악, GPS 등을 이용한 많은 콘텐츠를 개발하여 유료로 공급하고 있다. 하지만 요즘 사용자들은 여러 해킹 툴을 사용하여 유료 콘텐츠를 무단으로 배포하므로 콘텐츠 개발업체와 이동통신사에 손해를 입히고 있다. 본 논문에서는 콘텐츠 중 가장 많이 사용되고 있는 벨소리를 보호하는 기법을 제시하여 안전한 모바일 DRM 시스템이 갖추어지기 전 단계에서 그로 인한 피해를 효과적으로 줄이는 것을 기대할 수 있다.

1. 서론

최근 1 인 1 폰 시대가 열리고 그 핸드폰들은 멀티미디어 처리가 충분히 지원 되는 성능을 발휘하면서 사용자들의 관심사는 모바일 콘텐츠로 옮겨가기 시작했다. 처음에는 벨소리 서비스부터 시작하여 게임, 음악, e-book, 정보 등의 다양한 콘텐츠들이 등장했고 최근에는 DMB 서비스가 등장하여 언제 어디서건 방송을 볼 수 있게 까지 되었다. 이러한 콘텐츠는 거의 대부분 유료이기 때문에 이동통신사와 콘텐츠 제공업체의 저작권이 매우 중요하다. 하지만 최근에는 핸드폰 해킹도구와 제조업체에서 제공한 도구 등을 이용하여 벨소리, 게임 등의 콘텐츠들이 무단으로 유포가 되고 있는 실정이다. 그 결과 이동통신사와 콘텐츠 제공업체는 막대한 손실을 입고 있다.

본 논문에서는 핸드폰에서 사용하는 콘텐츠 중 가장 기본적이고 많이 사용하는 벨소리를 보호하는 기법을 제안한다.

논문의 구성은 2 장에서는 관련연구 및 최근 동향을 알아보고, 3 장에서는 보호 기법 소개와 제안한 파일 포맷을 설명하고, 4 장에서는 구현 및 성능평가를 보여주며, 5 장에서는 결론과 향후 연구에 대해 기술하였다.

2. 관련연구

2.1 OMA DRM

OMA(Open Mobile Alliance)에서 제정한 mobile DRM 규격이다. 3GPP 와 OMA 등의 여러 표준화 기구가 있으나 현재 OMA DRM v2.0 이 가장 강력한 mobile DRM 의 표준으로 여겨지고 있다. OMA DRM v2.0 의 시스템은 [그림 1]과 같다.

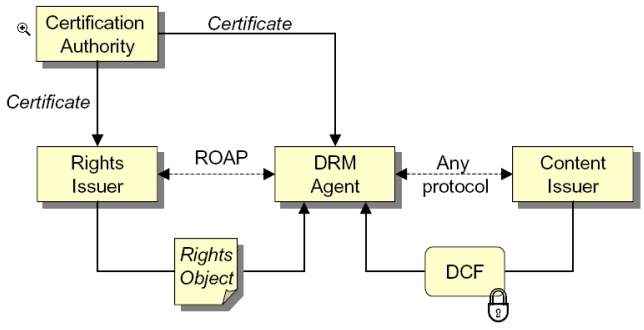

그림 1 - OMA DRM v 2.0

OMA DRM v 2.0 은 Contest Object 와 Right Object 가 따로 있다는 특징이 있으며 Right Issuer 과 DRM Agent 사

이는 ROAP 통신을 한다.

최근 국내 DRM 업체도 OMA DRM 규격에 맞추어 개발하고 있으며 SKtelecom 은 다양한 OMA 규격을 활용하여 서비스하는 중이다. [1, 2]

2.2 ESN (Electronic Serial Number)

ESN 은 단말기 제조업체가 단말기에 부여하는 고유번호이다. 단말기를 켜거나 전화를 걸 때 기지국을 통해 사업자에게 전달되며, 사업자는 이를 활용하여 사용자 식별, 통화, 인증 및 과금 등에 사용한다. 원래 AMPS 에 처음 사용되었으나 현재는 ANSI-41 표준을 사용하는 TDMA, CDMA 방식의 전세계 모든 휴대전화에서 사용하고 있다. 한편, 국제 Roming 을 할 경우 세계적으로 고유한 번호여야하기 때문에 미국의 TIA 에 그 관리와 배포를 위임하고 있다. ESN 의 구성는 아래 [그림 1]과 같다. Manufacture Code 는 8bit 와 14bit 가 있는데 이것은 제조회사 번호를 나타낸다. [3]

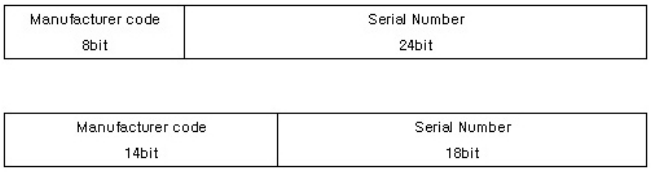

그림 2 - ESN 의 구성

2.3 ESN 의 안정성

ESN 은 핸드폰을 구분할 수 있는 가장 중요한 값이다. 예로 최근 ESN 을 복제하는 브릿지라고 부르는 휴대폰 복제를 비롯하여 유료 콘텐츠 다운로드, 도청까지도 가능하다. 작년 인도와 영국에서는 국내 휴대폰의 ESN 값을 해킹하는 프로그램을 고가에 판매 하는 기현상도 있었다.

그래서 제조업체들은 최근 출시된 핸드폰일수록 ESN 에 대한 보호를 더욱 확실히 하고 있다. ESN 은 저장 시에 쉽게 변경할 수 없도록 저장 번지를 암호화 하거나 한 바이트씩 띄어서 저장한다. [4, 5]

2.4 SMAF (Synthetic music Mobile Application Format)

SMAF 는 YAMAHA 에서 개발한 멀티미디어 데이터 형식이다. 그리고 그 형식은 모바일 기기에서 재생시킬 수 있는 멀티미디어 파일의 형식을 정의한 것이다. SMAF(.mmf) 파일은 SMF(.mid) 파일 등에 비해서 파일 용량이 작고 표현력이 우수하다는 장점이 있다. 이 파일은 YAMAHA 에서 제조하는 칩 시리즈에서 작동하는데 우리나라 핸드폰들은 모두 이 칩을 사용하고 있다.

현재 SMAF 는 주로 휴대폰 벨소리 제작에 사용되고 있지만, SMAF 의 확장 사양으로 텍스트나 그래픽 표시도 할 수 있다. 이 그래픽에 움직임이나 변화 등의 효과를 추가하면 모바일 기기상에서 벨소리와 그래픽을 동기화 재생이 되는 역동적인 콘텐츠 제작이 가능해진다. [6]

3. 벨소리 보호기법

본 절에서는 핸드폰에 다운되는 SMAF(.mmf) 벨소리 파일을 새로운 파일 형식 HSF(.hsf)로 저장하여 다른 사용자들과 공유 할 수 없도록 하는 방법을 소개한다.

3.1 개요

현재 벨소리는 서버에서 다운로드 후에 핸드폰의 호처리시 마다 호출되어 플레이가 된다. 하지만 여기서는 제안하는 방법은 다운로드 시 HSF 파일로 저장하여 벨소리 플레이 시 마다 새로운 플레이어로 플레이 한다. 그리고 HSF 파일을 다른 핸드폰으로 옮기면 플레이가 되지 않는 시스템을 보여준다.

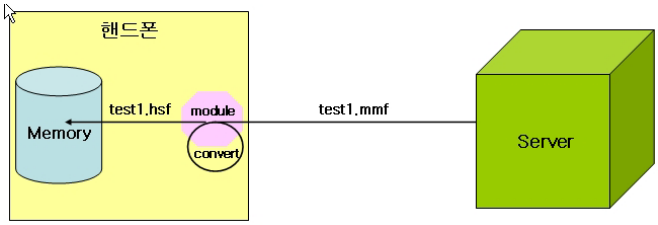

그림 3 - 시스템 개요

위 [그림 3]은 벨소리 다운로드 서버에서 test1.mmf 파일을 다운로드하고 중간 모듈에서 test1.hsf 로 변환하여 메모리에 저장되는 과정을 보여주고 있다.

3.2 보호기법

본 논문에서는 DES 와 RC4 두 가지의 암호 알고리즘을 사용하여 현재의 방식에 영향을 주지 않고 보호하는 기법을 제시한다..

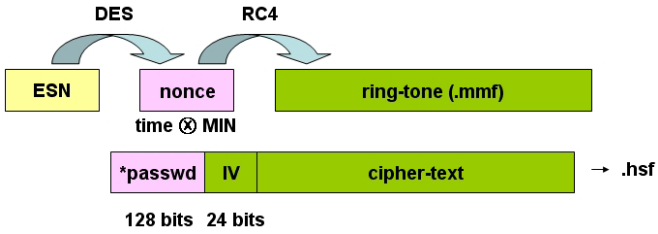

그림 4 – SMAF(.mmf)에서 HSF(.hsf)로 변화 과정

[그림 4]를 보면 현재 시간과 핸드폰 고유의 MIN(Mobile Identification Number)값을 이용하여 128bit 의 nonce 를 생성한다. 생성된 nonce 값의 첫 104bit 를 키로 사용하여 다운 받은 벨소리 파일(.mmf)을 RC4 로 암호화 하고 다시 nonce 는 DES 를 사용하여 암호화한 후 그 암호화된 값(*passwd)을 암호화된 벨소리(cipher-text)에 헤더로써 붙이고 파일(.hsf)로 저장한다. 그 결과 처음 SMAF 파일에 *passwd(128 bits)와 RC4 암호화 시 생성되는 IV(24 bits)가 더해져 처음보다 152bits 가 큰 HSF 파일이 생성된다.

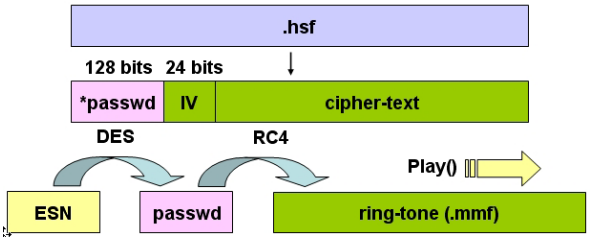

그림 5 - HSF(.hsf) 플레이 과정

호 처리 시에는 [그림 5]처럼, 먼저 ESN 을 키로 HSF 파일의 첫 128bit 를 DES 로 복호화 하여 passwd 를 얻는다. 그리고 앞의 104bit 를 이용하여 나머지 부분을 RC4 로 복호화 하고 클리어한 파일(.mmf)을 얻어 플레이 한다. (※ passwd 는 RC4 의 키이고 *passwd 는 passwd 가 DES 로 암호화된 후를 나타낸다.)

3.3 효과

DRM 시스템은 모든 agent 를 신뢰한다는 가정하에 공개키 기반 구조로 되어있다. 그래서 현재의 핸드폰에는 아직 적용하기 힘든 점이 적지 않다.

하지만 이 시스템에서는 인증서가 없는 환경에서 대칭키 암호를 사용하고 모든 agent 를 신뢰한다는 가정없이도 훌륭하게 적용될 수 있다.

이 시스템의 장점을 살펴보면 먼저, 핸드폰 마다 ESN 이 다르기 때문에 헤더부분이 핸드폰 동기화 되어 다른 핸드폰으로 옮겼을 경우에 정상적인 키를 얻을 수 없으므로 쓸모 없는 파일이 된다.

또, 파일 자체가 클리어하지 않게 저장되기 때문에 필요한 부분을 뽑아 낸다 하여도 재생이 불가능하게 된다.

그러므로 ESN 을 알고 있는 핸드폰의 파일을 복호화 하고 자신의 단말기의 ESN 을 이용해 다시 암호화 해야만 벨소리를 사용할 수 있게 된다.

하지만 이 방법은 사실상 매우 힘들다. ESN 은 현재 일반 사용자가 정상적으로 알 수 있는 방법이 존재하지 않기 때문이다

4. 구현 및 성능 평가

본 절에서는 3 절에서 제안한 시스템을 구현해본 환경을 살펴보고 구현결과의 성능을 평가 해보도록 한다.

4.1 구현환경

Operating System	Windows XP SP2
Emulator	Aroma WIPI 1.1.1.8
Dev. Language	JAVA
Dev. tool	Eclipse 3.0
JAVA version	Java SDK 5
Crypto Algorithm	DES, RC4
CPU	Pentium 4 1.7GHz
main memory	512MB

그림 6 - 구현환경

[그림 6]은 구현환경을 소개한다. 그리고 Aroma-WIPI Emulator 에서 HEAPSIZE 는 1024 로 설정하고 실행하였다.

4.2 구현 클래스

[그림 7]은 구현한 주요 클래스들을 간단히 소개하고 있다.

먼저, 에뮬레이터는 벨소리 다운로드 시에 서버로부터 파일을 받으면 FileEx 클래스가 동작하며 DES 클래스와 RC4 클래스를 사용하여 SMAF 파일을 HSF 파일로 변환하여 저장한다.

HS_player class	호 처리시에 벨소리를 플레이 하는 클래스
FileEx class	SMAF 파일을 HSF로 만들어주는 클래스
RC4 class	RC4 encrypt/dec
DES class	DES encrypt/dec

그림 7 - 구현된 클래스

또 에뮬레이터는 호 처리시 Player 클래스의 play() 대신 HS_player 클래스의 play()를 사용한다. HS_player 의 play() 메쏘드는 DES 클래스와 RC4 클래스를 사용하여 벨소리 파일을 재생이 가능한 SMAF 포맷으로 변환하여 플레이 하게 된다.

4.3 성능 평가

실험결과는 SMAF 파일과 HSF 파일의 플레이 시작부분과 플레이가 끝나는 부분까지의 시간을 측정하여 비교하여 나타내었다. 그리고 각각 3 개의 다른 크기의 파일 {(test1, test2, test3)(mmf, hsf)}을 사용하였으며 각 3 회씩 시간을 측정하였다.

SMAF 파일에 비해 HSF 파일은 DES 복호화와 RC4 복호화 과정이 추가되어 수행이 된다.

	SMAF(.mmf) milli sec.			HSF(.hsf) milli sec.			GAP sec.
test1 13.7KB	141	142	140	1688	1704	1672	
	141			1688			1.5
test2 54.1KB	172	218	187	7093	6125	6188	
	192.3			6468.6			6.3
test3 83.4KB	188	187	172	9563	9578	10532	
	182.3			9891			9.7

그림 8 - SMAF 와 HSF 의 실행 속도 비교

[그림 8]을 보면, 먼저 SMAF 파일은 파일의 크기에 상관 없이 실행시간의 차이가 크게 나지는 않는다. 하지만 HSF 파일은 크기에 따라 실행시간에 엄청난 차이를 보인다.

또, 표를 보면 같은 크기의 SMAF 와 HSF 의 실행시간의 차이가 파일 크기에 따라 크게 차이가 난다.

가장 작은 파일인 test1(13.7KB)파일은 실행 시간이 1.5sec 증가했으며, 중간 크기인 test2(54.1KB)파일은 6.3sec 로 증가했고, 가장 큰 파일인 test3(83.4KB)파일은 9.7sec 나 증가했다. 이 결과는 파일이 크면 핸드폰의 제한된 메모리로 인해 메모리 접근 횟수의 증가로 실행시간이 늘어나는 것으로 생각된다.

5. 결론 및 향후 연구

본 논문에서는 핸드폰의 주요 콘텐츠 중 벨소리가 무단으로 배포되는 것을 막기 위해 암호화 기법을 추가한 새로운 형식의 파일을 제안하고 구현 하였다. 하지만 실제 성능 평가를 하였을 때 원본을 플레이 한 것과 현격한 성능 차이가 발행하였다. 따라서 현재 복호화 과정을 플레이 방식이 아닌 플레이 도구에서 실행하면서 플레이를 한다면 보다 향상된 성능과 안정성을 보장할 수 있을 것이라고 생각한다..

그래서 본 논문에서는 이러한 점을 고려하여 실제 플레이 되는 부분을 암호화 할 때는 스트림 암호 방식인 RC4 를 사용하여 제안하였고 그에 대한 연구를 진행할 것이다. 또 OMA DRM 규격과 연동한 콘텐츠 보호기법에 대해서도 연구를 진행할 것이다.

참고문헌

[1] http://www.inews24.com/php/news_view.php?g_serial=169247&g_menu=020300

[2] Daniel Thull, Roberto Sannino, “Performance Considerations For an Embedded Implementation of OMA DRM 2” IEEE

[3]http://www.tta.or.kr/Home2003/library/weeklyNews View.jsp?news_id=473

[4] http://www.dt.co.kr/content/200409060201102561800 3.html

[5] 재미있는 CDMA 단말기(II) – SK Telecom 중앙연구원

[6] http://smaf-yamaha.com/kr/what/about.html