ZIP 파일의 보안성 강화 기법※

이정필*, 박진홍**, 박상주**, 최윤성**, 이근우***
김승주**, 원동호**
*성균관대학교 수학과
**성균관대학교 정보통신공학부
***성균관대학교 대학원 컴퓨터공학과
e-mail:jplee@dosan.skku.ac.kr

Advanced security technique for ZIP file

Jungpil Lee*, Jinhong Park**, Sangjoo Park**
Yoonsung Choi**, Keunwoo Rhee**
Seungjoo Kim**, Dongho Won**
*Dept. of Mathematics, Sungkyunkwan University
**School of Information and Communication Engineering,
Sungkyunkwan University
***Dept. of Computer Engineering, Graduate School,
Sungkyunkwan University

요 약

본 논문에서는 ZIP 파일 형식을 이용하여 파일을 압축/암호화 할 때의 몇 가지 문제점에 대하여 알아보고 이를 해결하기 위한 방법을 논의한다. 이미 여러 논문에서 ZIP 파일 형식의 압축/암호화에 대한 문제점이 논의되어 왔지만, 그 중에서 본 논문은 압축/암호화된 ZIP 파일의 부분정보 노출과 로컬 파일의 변경 및 삭제를 이용한 수동적/능동적 공격 기법을 방지하기 위한 해결방법을 제안한다.

1. 서론

최근 여러 분야에서 가장 많이 사용되는 압축 파일 형식(format)중의 하나가 ZIP이다. 더불어 PKZIP이나 WinZIP과 같은 여러 종류의 ZIP 응용 프로그램에서는 ZIP 파일에 다양한 암호화 기법을 적용하여, 보안성이 강화된 압축 파일을 생성할 수 있게 해주고 있다. 따라서 사용자들은 쉽고 편하게 파일을 압축하기 위해서 뿐만 아니라, 자신의 데이터를 보호하기 위해 ZIP 파일을 사용하기도 한다. 이와 같이 ZIP 파일에 대한 사용이 증가함에 따라, ZIP 파일 형식이나 ZIP 응용 프로그램에 대한 문제점이 다양하게 보고되고 있다[1]. 본 논문에서는 대표적인 ZIP 응용 프로그램인 WinZIP을 이용하여, 현재 ZIP 파일 형식의 문제점을 알아보고, 그에 대한 해결방법을 제안한다.

2장에서는 일반적인 ZIP 파일 형식에 대해 설명하고, 3장에서는 압축/암호화된 ZIP 파일의 부분정보 노출에 대한 문제점과 로컬파일 변경 및 삭제에 대한 문제점을 분석한다. 4장에서는 3장에서 제기한 문제점에 대한 해결방법을 제안한다. 마지막으로 5장에서는 결론을 내린다.

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음

2. ZIP 파일 형식

일반적인 ZIP 파일 형식은 (그림 1)과 같이 크게 Local file header, File data, Central directory, End of central directory의 네 부분으로 구성되어 있다[5].

Local file header 1	로컬파일의 개수만큼 Header와 Data가 쌍으로 존재
local file header signature (4bytes) version needed to extract (2bytes) general purpose bit flag (2bytes) compression method (2bytes) **last mod file time (2bytes)** **last mod file date (2bytes)** crc-32 (4bytes) **compressed size (4bytes)** **uncompressed size (4bytes)** file name length (2bytes) extra field length (2bytes) **file name (variable size)** extra field (variable size)	
File data 1	
압축 또는 압축/암호화된 파일	
Local file header 2	
… 내용 생략 …	
File data 2	
압축 또는 압축/암호화된 파일	
…	
Central directory	로컬파일의 개수만큼 file header가 존재
File header 1	
central file header signature (4bytes) version made by (2bytes) ***version needed to extract ~ extra field length 까지*** ***Local file header 1과 동일*** file comment length (2bytes) disk number start (2bytes) internal file attributes (2bytes) external file attributes (4bytes) relative offset of local header (4bytes) ***file name과 extra field는*** ***Local file header 1과 동일*** file comment (variable size)	
File header 2	
…내용 생략…	
…	
End of central directory	
… 내용생략 …	

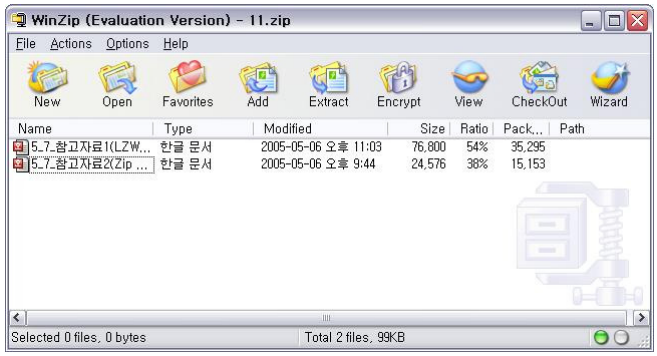
(그림 1) 일반적인 ZIP 압축 형식

Local file header는 File data에 대한 정보가 저장되고 File data는 파일의 실제 데이터가 압축 또는 압축/암호화되어 저장된다. Local file header의 내용은 Central directory에도 동일하게 존재하며 ZIP 파일에 대한 추가적인 정보가 Central directory에 저장된다. ZIP 응용 프로그램들은 Central directory에서 File data에 대한 정보를 읽어 들인다. ZIP 압축 형식의 가장 마지막 부분인 End of central directory는 Central directory에 대한 정보를 저장한다.

3. 압축/암호화된 ZIP 파일의 문제점 분석

(1) 부분정보 노출에 대한 문제점

그 동안 ZIP 파일 형식의 많은 업데이트에도 불구하고 해결되지 않고 있는 문제점들이 있다. 그 중 하나가 압축/암호화된 ZIP 파일의 로컬파일들에 대한 부분정보(Partial Information)가 그대로 노출되고 있다는 것이다. (그림 2)과 같이, 현재 ZIP 응용 프로그램들은 압축/암호화된 로컬파일들에 대한 '파일이름(Name)', '파일종류(Type)', '수정된 시간과 날짜(Modified)', '원본크기(Size)', '압축률(Ratio)', '압축된 크기(Packed Size)', '경로(Path)' 등의 부분정보를 그대로 보여주고 있다. 이는 Central directory의 내용을 보여주는 것으로 (그림 1)의 Local file header1에서 굵은 글씨로 되어 있는 필드에 해당한다. File data는 암호화 되지만 Central directory와 Local file header는 평문이기 때문에 압축/암호화된 ZIP 파일의 부분정보를 보는 것이 가능하다.



(그림 2) WinZIP에서 압축/암호화된 ZIP 파일을 열었을 때

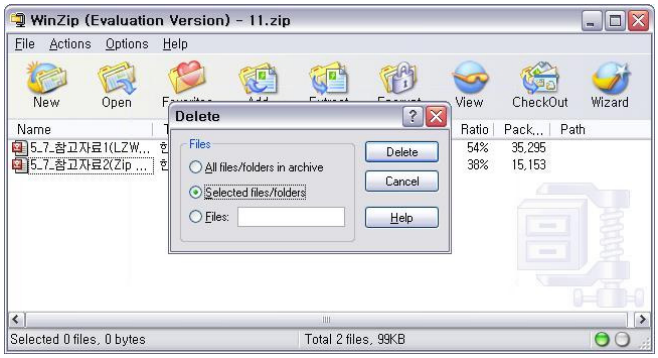
이러한 ZIP 파일 형식은 사용자에게 압축/암호화된 ZIP 파일을 복호화 하여 풀어 보지 않고도 각 로컬파일의 부분정보를 제공하여 사용자가 원하는 로컬파일을 풀어 볼 수 있도록 하려는 목적이 있다. 그러나 공격자에게도 로컬파일의 부분정보가 노출된다는 것이 문제점이다. 이는 암호화 기법에 대한 수동적 공격을 가능하게 한다. 또한 만약 사용자가

로컬파일의 이름에 포함된 숫자나 글자 등의 부분정보를 비밀번호로 사용하여 압축/암호화를 했을 경우 공격자가 비밀번호를 유추할 수 있어 공격자의 능동적 공격 역시 가능하게 만들 수 있다.

(2) 로컬파일 변경 및 삭제에 대한 문제점

(그림 3)에서 보는 바와 같이 공격자는 비밀번호 없이 압축/암호화된 ZIP 파일의 로컬파일들을 삭제할 수 있다.

이는 ZIP 파일 형식의 구조상 가능한 것으로 로컬파일은 Local file header와 File data로 구성되어 있고 (그림 1)에서 보는 바와 같이 각 로컬파일이 블록처럼 모여 하나의 ZIP 파일을 구성하고 있다. 각 로컬파일 사이에는 아무런 연관관계가 없다. 앞서 말한 바와 같이 Central directory에 Local file header의 내용이 중복되어 저장되기는 하지만 평문으로 저장되기 때문에 수정이 가능하다. 따라서 압축/암호화된 ZIP 파일이라 하더라도 ZIP 파일 형식의 구조상 로컬파일의 변경, 삭제가 가능한 것이다.



(그림 3) 비밀번호 없이 로컬파일 삭제

이러한 ZIP 파일 형식의 블록구조는 사용자의 편의성을 위한 것이지만 암호통신에 압축/암호화된 ZIP 파일을 이용하기에는 부적합한 구조이다. 즉, 공격자가 이 문제점을 악용하여 두 통신자 사이의 중요문서를 삭제하거나 변경하는 등의 능동적 공격이 가능하게 된다.

4. 압축/암호화된 ZIP 파일의 문제점 해결방법

앞서 소개한 두 문제점을 해결하기 위한 가장 좋은 방법은 PKWARE사 '.ZIP file Format Specification' 기술문서의 'Strong Encryption Specification'을 따르는 것이다. 이 방법은 Central directory를 암호화하고 Local file header를 0값으로 채우는 방식으로 압축/암호화된 ZIP 파일의 보안 강도를 높여 주는 것이다. 하지만 기존의 ZIP 응용 프로그램과 호환이 되지 않는 문제가 있다[1]. 따라서 이 논문에서는 기존 ZIP 응용 프로그램과 호환을 이루면서도 압축/암호화된 ZIP 파일의 보안 강도를 높이는 방법을 제안하고자 한다.

(1) 부분정보 노출에 대한 해결방법

먼저, 사용자가 입력한 비밀번호를 해쉬(Hash) 함수에 입력하여 해쉬값을 구한다. 파일들을 ZIP 파일 형식으로 압축/암호화하고 앞서 구한 해쉬값을 문제가 되는 로컬파일의 부분정보 관련 필드값과 XOR 연산을 하여 각 필드에 저장한다.

이 방법은 기존의 ZIP 파일 형식은 유지하면서, 비밀번호를 모르는 공격자로부터 실제 로컬파일의 부분정보를 은폐시킬 수 있는 장점이 있다. 따라서 기존의 ZIP 응용 프로그램과 호환이 가능하다. 단지 보이는 부분정보만 실제 정보와 다를 뿐이다. 또한 ZIP 응용 프로그램을 수정하면 메뉴나 옵션 등을 통해 사용자가 자신의 비밀번호로 로컬파일의 실제 부분정보에 접근하는 것이 가능하다.

(2) 로컬파일 변경 및 삭제에 대한 해결방법

로컬파일 변경 및 삭제에 대해서는 두 가지 해결방법을 제안하고자 한다. 첫 번째 해결방법은 압축/암호화된 ZIP 파일의 로컬파일을 삭제할 수 있는 가능성을 확률적으로 낮추는 방법이고 두 번째 해결방법은 특정하게 계산된 값을 통해 로컬파일이 변경 또는 삭제되었는가를 알아내는 방법이다.

우선, 첫 번째 해결방법은 기존의 ZIP 압축 형식을 그대로 유지하면서 확률적으로 내부 파일을 보호할 수 있도록 더미(dummy) 파일을 이용하는 방법이다. 즉, ZIP 응용 프로그램에서 ZIP 파일을 생성하기 전에 보호하고자 하는 파일들 사이에 더미 파일을 생성하여 추가하고 압축/암호화하는 것이다. '.ZIP File Format Specification'에 따르면 Local file header에 필요에 따라 사용가능한 extra field가 존재하는데[5], 파일들을 압축/암호화할 때에 더미 파일의 extra filed에 더미 파일임을 알려주는 특정 값을 저장한다. ZIP 파일의 복호화 및 압축해제 시, 앞서 말한 특정 값을 읽어 더미 파일을 삭제하도록 하면 보호하고자 했던 파일들만 남게 된다.

공격자는 더미 파일이 추가된 ZIP 파일로부터 공격 대상(삭제 또는 변경하고자 하는 파일)을 찾기

어렵게 되며, 설사 공격자가 악의를 품고 아무 파일이나 삭제하려한다 해도 보호하고자 하는 파일들이 삭제될 확률을 줄여 줄 수 있다. 보호하고자 하는 파일이 m개, 더미 파일이 n개 있을 경우, 보호하고자 하는 파일이 삭제될 확률은 m/(m+n)이 된다.

더미 파일이 많을수록 보호하고자 하는 파일이 삭제될 확률은 낮아지지만 반대로 총 파일의 개수와 크기가 늘어나므로 압축률이 떨어질 것으로 생각할 수 있는데, 더미 파일의 크기는 최소로 하여 생성/추가하도록 하되 3장에서 제안한 부분정보 노출 문제 해결방법을 함께 사용한다면, 더미 파일의 '원본 크기(Size)'와 '압축된 크기(Packed Size)'도 숨길 수 있으므로 공격자는 공격 대상을 찾기가 더욱 어려워지게 된다. 이 방법을 사용할 경우 사용자가 원하는 더미 파일의 크기와 추가하고자 하는 더미 파일의 개수를 설정할 수 있도록 ZIP 응용 프로그램에서 옵션이나 메뉴 등을 통해 지원하는 것이 좋을 것이다.

두 번째 해결방법은 파일이 공격자에 의해 삭제 또는 변경되었을 경우 그것을 알 수 있도록 Central directory의 마지막 extra field에 특정하게 계산된 값을 추가하는 것이다.

특정하게 계산된 값이란 랜덤(Random)한 순열에 따라 파일의 각 부분을 4byte씩을 XOR한 값을 의미한다. 조금 더 구체적으로 설명하면 파일을 암호화하는데 사용되었던 사용자의 비밀번호를 해쉬 함수에 입력하여 해쉬값을 얻어낸다. 이 값을 시드(seed)값으로 이용하여 랜덤한 순열을 생성한 다음 순열의 순서대로 파일을 따라가며 4byte씩 XOR하여 값을 계산하는 것이다. 이렇게 계산된 값을 Central directory의 제일 마지막 extra field에 저장한다. 이렇게 저장된 값을 압축해제 시, 앞서 설명한 계산과정을 통해 재계산된 값과 비교한다. 이 때 비교한 값이 서로 다르다면 파일이 중간에 변경되거나 삭제되었다는 것을 알 수 있다.

공격자가 앞서 계산된 값을 자신의 의도에 맞게 수정하기 위해서는 랜덤한 순열을 찾아내야 한다. 하지만, 사용자의 비밀번호를 해쉬 함수에 입력하여 시드값을 얻어내기 때문에 공격자가 랜덤한 순열을 찾아내는 것은 꽤 어렵다고 볼 수 있다.

5. 결론

본 논문에서는 ZIP 파일 형식의 알려진 문제점 중 압축/암호화된 ZIP 파일의 부분정보 노출과 로컬 파일 변경 및 삭제를 해결하기 위한 방안에 대하여 논의하였다. ZIP 파일은 현재 여러 분야에서 가장 많이 사용되어지는 압축 파일 형식 중 하나이다. 하지만, 사용자의 편의성에 중점을 두고 개발되었기 때문에 강력한 암호화 기법을 적용했음에도 불구하고 WinZIP과 같은 상용 프로그램에서 보호해야할 로컬파일의 부분정보를 제공하고 사용자의 비밀번호 없이 로컬파일을 자유롭게 변경/삭제할 수 있는 문제점을 가지게 되었다.

일반 사용자들은 개인적인 용도에 있어서 이런 문제는 크게 중요해 보이지 않을 수도 있다. 하지만 중요한 데이터를 백업하거나 전송하기 위해서 압축/암호화된 ZIP 파일을 사용한다면 앞선 문제점들은 간과해서는 안 될 큰 문제가 될 수 있다. 또한, 이러한 문제점을 해결하기 위해서 PKWARE사의 'Strong Encryption' 이란 기술이 사용되기는 하지만 기존 ZIP 응용 프로그램과의 호환성과 사용자의 편의성은 크게 떨어지는 것이 사실이다. 그래서 이 논문에서 제시한 세 가지 방법이 ZIP 파일에 대한 사용자의 편의성을 유지하며 보안 강도를 높이는데 유용하리라 생각된다.

참고문헌

[1] Tadayoshi Kohno, "Analysis of the WinZip encryption method", IACR ePrint Archive 2004/078. http://eprint.iacr.org/

[2] Eli Biham, Paul C. Kocher, Mikel Stay , "Yet another plaintext attack to ZIP encryption scheme", Mailing Lists Discussion of security issues, 2003-February Archives

[3] Mikel Stay, "ZIP attacks with reduced known plaintext", 8th Fast Software Encryption Workshop, pp. 125-134. LNCS 2355, Springer-Verlag, 2002.

[4] "64 Cracking ZIP file's password", http://www.guideme.itgo.com/atozofc/ch64.pdf

[5] "ZIP File Format Specification appnote", http://www.pkware.com/company/standards/appnote/appnote.txt

[6] "AES encryption information: Encryption specication AE-2", WinZip Computing Inc. Jan. 2004. Version 1.02, available at http://www.winzip.com/aes_info.htm

[7] B. Kaliski, "Password-Based Cryptography Specification Version 2.0", Network Working Group RFC 2898, Category: Informational ,September 2000