모바일에서 압축분할과 다중 클라우드 스토리지 저장을 통한 보안 향상기법 연구

신형섭*, 양민수*, 송양의**
*동국대학교 정보보호학과
**동국대학교 컴퓨터공학과
e-mail:daem0n@naver.com, ddangggom@nate.com, redcroix@dongguk.edu

Study on The Security Enhancement Techniques through Mobile from Compression Splitting and Multi-cloud Storage to save

Hyung-Sup Shin*, Min-Su Yang*, Yang-Eui Song**
*Dept of Informaion Security, Dongguk University
**Dept of Computer Engineering, Dongguk University

요 약

모바일 디바이스의 사용이 활발해지면서 사용자들은 PC에서 사용하는 서비스를 모바일 디바이스로 사용하려고 한다. 모바일 디바이스는 PC에 비해 자원이 한정적이며, 이러한 문제를 클라우드 서비스를 통해 해결하고 있다. 특히 모바일 디바이스의 제한된 저장 공간을 클라우드 스토리지를 이용함으로서 저장공간에 대한 제한성을 해결한다. 하지만 중앙 집중형태로 저장되는 클라우드 스토리지의 보안문제는 아직도 현재 진행형이며, 네트워크상에 다양한 공격 위협으로부터 노출되어 있다.

본 연구에서 모바일 디바이스에서 클라우드 스토리지로 데이터를 저장 할 때 보안을 향상시킬 방법을 제안하고 일반 전송서버와 시간을 비교하였으며 실험을 통해 안정된 성능을 입증하였다.

제1장. 서론

최근 사용자들의 모바일 디바이스가 확장되고 플랫폼서비스가 다양해지면서 컴퓨터 시스템의 H/W자원을 가상화하여 여러 사용자에게 제공하는 클라우드 컴퓨팅(Cloud Computing)이 활발해지고 있다. 대다수 모바일 사용자들은 기존에 사용하던 데스크탑 컴퓨터에서 이동성과 개인 소유의 특징이 뛰어난 스마트폰에서 서비스를 이용하려고 있지만 모바일 디바이스는 데스크탑 컴퓨터 환경에 비해 자원(메모리, CPU, 화면 등)의 한계가 있다. 최근에는 이러한 모바일 디바이스 문제를 클라우드 서비스를 이용하여 해결하기 위해 모바일 클라우드 서비스가 각광 받고 있다.

클라우드 컴퓨팅은 가상화(virtualization), 즉시성(instance), 확장성(expandability)을 대변 한다[1]. 특히 이러한 클라우드 컴퓨팅 서비스 중 컴퓨터 시스템의 디스크 H/W 자원을 가상화하여 여러 사용자에게 제공하는 IaaS (Infrastructure as a Service)는 다양한 포털사이트에서 저장 공간을 제공함으로써 모바일 디바이스의 약점인 디스크 공간을 확장할 수 있게 되었다.

하지만 클라우드 컴퓨팅의 도입을 저해하는 주된 요인은 보안문제이다. 2011년 10월 한국인터넷진흥원(KISA)에서 발간된 “클라우드 서비스 정보보호 안내서” 에 따르면 모든 연산 및 서비스가 클라우드 서버에서 이루어지기 때문에 보안문제는 아직까지 현재 진행형이다[2]. 그 중에서 Iaas의 클라우드 스토리지 서비스는 중앙 집중형태로 데이터가 저장되기 때문에 보안 가이드라인의 부재, 접근제어관리, 가용성 및 무결성 보장 기술 등의 적용 부족으로 많은 위협에 노출되어 있다[3]. 이러한 위협은 기업 기밀정보 및 개인정보 유출 등으로 이어지며 보안 우려의 목소리가 크게 나타나고 있다.

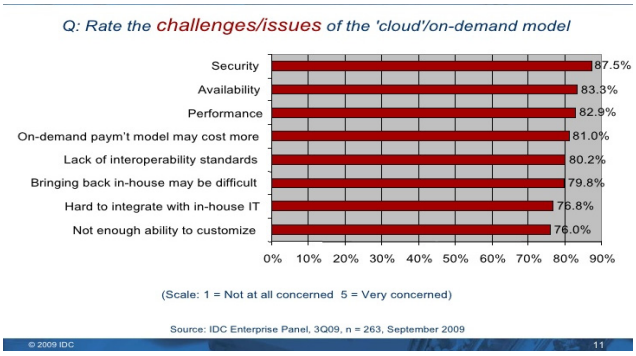
본 논문은 기존 클라우드 위협을 감소시키기 위해 클라우드 스토리지로 데이터를 저장 할 때 암호화 하기 위한 압축/분할 기법을 제시한다. 이는 스니핑 공격(Sniffing Attack)과 클라우드 스토리지의 가용성과 기밀성을 향상시키기 위해 다중 클라우드 스토리지 분할 전송 기법에 관한 것이다. 제2장에서는 관련연구를 소개하고, 제3장에서는 우리가 제안하는 구현알고리즘을 소개한다. 제4장에서는 실험 및 테스트이며 마지막으로 제5장에서 결론을 맺는다.

제2장. 관련연구

2.1 클라우드 컴퓨팅 보안이슈

클라우드 스토리지(Cloud Storage)는 직접 H/W를 소유하지 않은 개인 혹은 기업이 필요한 만큼 스토리지 자원을 사용하는 방식으로 보안은 필수적인 사항이다. (그림 1) 은 시장조사 기관인 IDC에서 244명의 IT 임원들에게 IT Cloud 서비스에 관하여 그들의 견해와 활용에 대하여 조사한 것이다[5]. 여기에서 클라우드를 위해 보안을 가장 먼

저 해결해야 한다고 답하고 있다[5]. 클라우드 컴퓨팅 서비스 위협은 [표 1] 과 같이 서비스 오남용, 데이터 손실 및 유출 등으로 분류된다[3].



(그림 1) Rate the challenges/issues of the cloud

특히 모바일 클라우드 서비스는 무선네트워크를 이용하여 대부분의 정보 처리와 저장을 클라우스 서비스를 이용하기 때문에 (표 1)과 같은 클라우드 서비스의 위협이 무선네트워크와 함께 복합적으로 발생 할 수 있다[3]. 무선네트워크의 위협으로 인해 인증되지 않는 무선 랜을 사용하거나 안전하지 않은 암호화방식 등 사용자는 스니핑 공격(Sniffing Attack)과 의인화(Impersonation) 공격으로부터 자유로울 수 없다.

(표 1) 클라우드 서비스의 위협

위협	요약	위험기술
서비스 오남용	클라우드 자원을 악의적인 목적으로 오용 및 남용	데이터암호화 보호 크랙
		서비스 과부하
		서버 모바일 봇넷화
		악성코드 호스팅
		CAPTCHA를 푸는 영역으로 활용
데이터 손실 및 유출	통합되고 공유된 자원이 악의적 접근에 노출되거나 데이터 사이 구분의 불명확함	불충분한 접근제어
		부적절한 권한부여
		Software 라이센스 크랙
		클라우드서버 정보 유출

2.2 Deflate 알고리즘

Deflate 알고리즘은 LZ77 Code와 Huffman Code가 결합된 대표적인 Zip 압축에서 사용하는 알고리즘이다[9]. 압축은 저장공간을 효율적으로 줄여주고 원본파일 헤더 은닉에도 사용된다.

LZ77 Code는 슬라이드 사전법 또는 LZ1법이라 불리운다[10]. 같은 문자열이 나타나면 그것을 앞에 나타난 문자열의 시작 위치, 문자열의 길이의 형식으로 식 (1.1)과 같이 바꾼다.

"ababbababaa" -> "ab(1,2)(2,3)(6,3)(10,1)" (2.1)

바뀐 문자열을 보내려면 문자 그대로인지 복사한 것인지를 나타내는 1비트의 플래그를 붙여서 <0,a><0,6><1,1,2>,<1,2,3>,<1,6,3>,<1,10,1> 의 형태로 나타낸다. 'a'와 'b' 등의 문자는 ASCII 부호의 8비트 그대로 나타내고 플래그 비트가 1일 때 <1,P,L>의 2번째 및 3번째 파라미터는 참조 문자열을 보관하는 버퍼 중의 일치하는 위치 P와 문자열의 길이 L을 나타낸다[6].

Huffman Code 알고리즘은 최적의 접두사 코드(prefix code)에 관한 아래의 두가지 관찰에 기초를 두고 있다.

Ⅰ. 최적의 코드에서 빈도수(또는 발생확률)가 많은 심벌(symbol)은 빈도수가 작은 심벌보다 짧은 코드를 갖는다.
Ⅱ. 최적의 코드에서 가장 빈도수가 작은 두 심벌의 코드의 길이는 같다.

Huffman Code는 순서쌍(ordered pair) H = (S,P)로 정의되는 소스 H를 가정한다. S는 심벌 S = $\{S_1,S_2\cdots S_n\}$ 이고 각 심벌의 발생확률 P(Si)=Pi, 여기서 P_i-1 은 P_i 보다 작거나 같다. (표 2)는 주어진 H_1 에 대한 코드를 생성하는 과정의 예를 보여준다. H_i는 7개의 심벌 A. B, C, D, E, F, G로 구성되어 있고 각각의 발생확률은 43%, 25%, 15%, 7%, 6%, 3%, 1% 이다. 허프만 코드를 생성하기 위한 과정은 심벌을 발생 확률에 따라 내림차순으로 정렬한다. 허프만 트리에서 루트노드에 시작하여 각 노드의 좌측은 0, 우측은 1을 할당하면 각 심벌에 할당된 허프만 코드는 A = 0, B = 10, C = 110, D = 1110, E = 11110, F = 111110, G = 1111111이 된다[7].

(표 2) 주어진 H_1

과정	심벌(발생확률)
1	A B C D E F G 0.43 0.25 0.15 0.07 0.06 0.03 0.01
2	A B C D E S1 0.43 0.25 0.15 0.07 0.06 0.04
3	A B C S2 D 0.43 0.25 0.15 0.10 0.07
4	A B S3 C 0.43 0.25 0.17 0.15
5	A S4 B 0.43 0.32 0.25
6	S5 A 0.57 0.43

2.3 MD5 해시함수

MD5는 가변길이 입력으로부터 식 (2.2)을 이용한 512비트블록의 최종 결과 값인 128비트 난수를 생성한다. OTP를 이용한 암호화된 패스워드 생성에 사용되며 도치(Inversion), 충돌(Collision), 위조(Forgery)와 같은 방어책이 있어야 한다. 도치는 주어진 해시 값으로부터 메시지를 알아내는 것이고, 충돌은 두 개 이상의 서로 다른 메세지

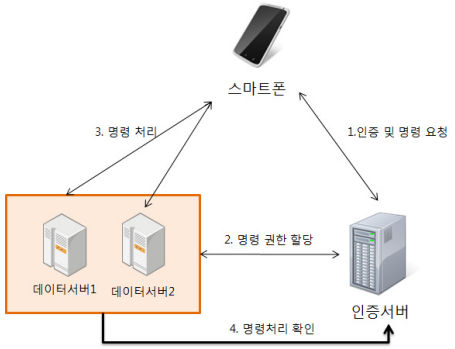
가 같은 해시 값을 갖는 것이다. 그리고 위조는 비밀키에 대한 지식 없이 MAC(Message Authentication Code)를 산출하는 것이다. 그리고 MD5는 IEEE 802.11의 무선 디바이스 인증 표준으로 사용되고 있다[8].

$$A \leftarrow B+((A+g(B,C,D)+X[k]+T[i])<<<s) \quad (2.2)$$

제3장. 제안 알고리즘

3.1 압축 / 분할

본 논문에서 제안하는 클라우스 스토리지 보안 기법은 (그림 2)와 같이 스마트폰, 인증서버, 클라우드 스토리지 개념의 N개의 데이터서버로 구성되어 있다. 클라우드 스토리지로 전송할 데이터 용량을 최소화하고 전송속도를 향상시키기 위해 Deflate 알고리즘의 Zip 압축 기법을 사용한다.

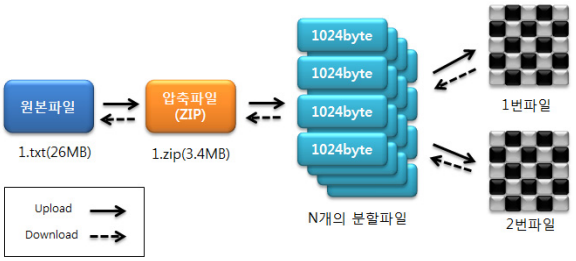


(그림 2) 클라우드 스토리지 보안기법

압축된 파일을 우선 순위가 높은 데이터서버의 2곳에 보내기 위해 전치암호화 방법으로 1024Byte 크기로 교차하며 데이터를 (그림 3)과 같이 2개의 파일로 분할한다.

2개의 파일로 교차 분할된 데이터는 업로드/다운로드 시 네트워크 중간자공격 등을 방지 할 수 있고, 데이터서버에 업로드 했을 때 기밀성을 향상시킬 수 있다.

압축/분할된 파일은 인증서버가 지정한 우선순위가 높은 2개의 데이터서버로 전송하기 위하여 인증서버에 데이터 업로드를 요청한다.



(그림 3) 압축/분할 방법

3.2 다중 스토리지 저장

클라이언트(모바일 디바이스 등)에서 클라우드 스토리지에 저장하기 위해 (그림4)와 같은 방법으로 업로드/다운로드 요청을 시도한다.

1.모바일 단말기는 인증서버에게 인증 데이터 (AuthData) (Filename, size, ID/PWD) 및 업로드 요청을 한다.

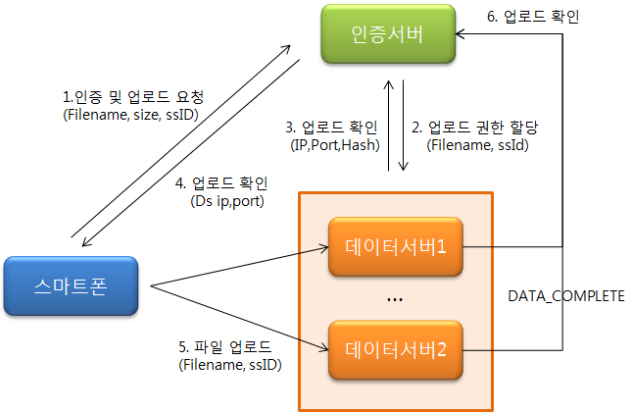
2.요청을 받은 인증서버는 인증처리 후 데이터서버에 인증데이터와 SessionID를 우선순위가 높은 2개의 독립된 데이터서버에 전송한다.

3.인증서버로부터 요청을 받은 데이터서버는 SessionID를 확인 후, 자신의 데이터서버 IP, Port 와 File을 구분짓는 MD5 방식의 랜덤해쉬 값을 인증서버로 보내게 된다.

4. 인증서버는 데이터서버로부터 받은 랜덤 해시값과 파일명을 DB에 저장하고 파일리스트로 관리하고 데이터서버 IP, Port를 스마트폰에게 전달한다.

5. 스마트폰은 인증서버로 부터 전달받은 Data Server IP, Port로 데이터서버에 데이터 인증정보(File name, sessionID)와 파일전송 요청을 한다. 데이터서버는 sessionID을 확인하고 파일전송을 시작한다.

6. 데이터서버 2곳에 파일전송이 끝나면 클라이언트에게 완료 메세지를 전송하고 데이터전송이 완료된다.



(그림 4) 파일 업로드

제4장. 실험 및 분석

테스트를 위한 환경으로 통신은 WIFI 무선랜의 WEP 방식으로 94Mb/s의 전송 속도에 최대한 맞췄다. 모바일 디바이스는 듀얼코어 1.4GHz CPU, 1GB 메모리의 스마트폰이며, 인증서버/데이터서버는 쿼드코어 CPU, 4GB 메모리로 구성했다.

스마트폰에서 자주 사용되는 용량과 파일형식으로 5MB의 텍스트 파일, 10MB의 JPG 이미지 파일, 50MB의 apk 파일 및 100MB 동영상 멀티미디어 파일을 전송을 위한 테스트 파일로 하여 각각 10회 테스트 하였다.

실험은 테스트 파일을 제안한 기법인 '압축/분할 및 데

이터 전송' 시간을 체크했으며, 시간 비교를 위하여 압축/분할이 없이 원본 테스트파일을 전송하는 일반 '비압축/비분할 테스트 파일'을 전송하는 시간을 실험하였다.

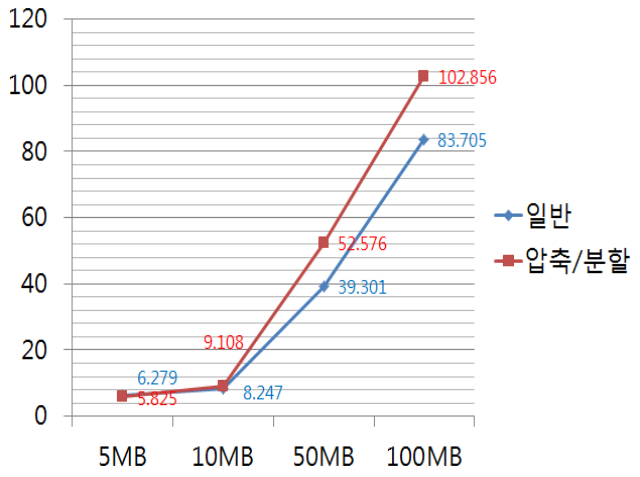
(표 3) 실험 결과

단위 : 초(s)

방법	데이터 크기			
	5MB	10MB	50MB	100MB
비압축/비분할 방법 평균	5.92	8.17	39.21	83.67
압축/분할 방법 평균	5.91	9.54	51.94	103.69

10회 실험 후 평균 시간을 보여주는 (표 3)을 보면 파일의 크기가 작을수록 압축/분할하는 속도가 빠르고 분할된 전송속도가 우수하였다. 제안하는 압축/분할 전송방법은 5MB의 텍스트 파일의 실험에서 일반 비압축/비분할 전송방식보다 1s의 시간 향상이 있었으며, 10MB에서는 1.37s, 50MB에서는 12.73s, 100MB에서는 20.02s 정도로 압축 및 분할의 시간이 포함되어 있음에도 불구하고, 전체적으로 전송시간에 영향을 줄 만큼 큰 차이는 없었다.

전송 데이터의 용량이 커질수록 차이가 있었지만 압축/분할 전송방법은 일반적인 비압축/비분할 전송방법의 전송시간보다 평균적으로 1.25배 정도 차이 정도 였다. (그림 5)의 그래프는 제안한 압축/분할 전송방법과 비압축/비분할 전송 방법의 실험 결과(1~5회)를 전송 데이터 크기에 따른 전송 시간의 비교표로 나타낸 것이다.



(그림 5) 전송시간 비교

제5장. 결 론

본 연구에서는 모바일 환경에서 클라우드 스토리지 보안 기법을 언급하였다. 본 논문에서 제안하는 방법은 스마트폰에서 클라우드 스토리지에 전송하기 전에 압축/분할 방법으로 중간 스니핑 공격을 방지하고, 다수의 클라우드 스토리지에 교차 분할 저장함으로써 데이터 자체의 기밀성을 향상시키고자 하는 보안 향상기법이다.

제안한 방법의 우수성을 검증하기 위해 일반적인 비압축/비분할 전송과 비교하여 테스트하였다. 테스트 결과는 파일이 작을수록 좀 더 빠른 속도를 나타냈으며, 시간적인 측면에서는 일반 비압축/비분할 전송과 비교하여 압축/분할 전송방법이 전체적으로 1.25배 정도 밖에 차이가 없었다.

본 논문의 제안에서 전송 시간의 단축에 향상이 있었지만, 100MB 이상의 데이터인 경우 압축/분할에 시간과 메모리가 많이 소모되는 어려움이 있다. 보안을 위하여 압축/분할에 소요되는 시간의 소모를 줄이기 위한 알고리즘을 개선은 추후 과제로 남겨둔다.

참고문헌

[1] 이태규 외 1명, "클라우드 컴퓨팅과 스마트 컴퓨팅의 정보 상호작용", 한국 정보기술 학회지 제10권, 2012.3

[2] 손태식 외 1명, "Cloud Computing에서의 IoT 보안 동향", 정보보호학회지, 2012, 2

[3] 장은영 외 4명, "모바일 클라우드 서비스의 보안위협 대응방안 연구", 정보보호학회, 2011, 2

[4] Cloud Computing Survey An IDC Update , 2010

[5] 은성경 "클라우드 컴퓨팅 보안 기술 동향", 한국정보보호학회, 2010, 4

[6] 박지환 "멀티미디어 압축 기술", 한국멀티미디어 학회지, 1997, 12

[7] 박상호, 허프만 코드의 선택적 암호화에 관한 연구, 정보보안 논문지, 2007, 6

[8] 이철승, MANET 기반 MD5 보안 라우팅에 관한 연구 한국 전자통신학회 논문지, 2012, 6,

[9] RFC 1951, DEFLATE Compressed Data Format Specification version 1.3

[10] Jacob Ziv and Abraham Lempel; A Universal Algorithm for Sequential Data Compression, IEEE Transactions on Information Theory, 23(3), pp. 337 - 343, May 1977.