논문 13-38B-05-10
한국통신학회논문지 '13-05 Vol.38B No.05
http://dx.doi.org/10.7840/kics.2013.38B.5.401

Cloud Shredder에 대한 취약점 분석 및 개선 방안에 관한 연구

박 민 수◆, 강 성 훈*, 김 승 주°

Weakness and Improvement of Cloud Shredder

Minsu Park◆, Sunghoon Kang*, Seungjoo Kim°

요 약

최근 IT 인프라의 발전으로 대부분의 데이터가 디지털 형태로 저장 및 관리되고 있다. 이러한 디지털 데이터는 매우 작은 공간에 대량의 데이터를 저장할 수 있는 장점이 있지만 데이터가 저장된 디스크를 분실하거나 도난당할 경우 해당 디스크에 담긴 많은 양의 데이터가 유출될 수 있는 위험이 존재한다. 현재 이와 같은 디지털 데이터의 유출을 막기 위해 데이터 암호화와 같은 데이터 보호 방법을 이용하여 민감한 데이터를 보호하고 있다. 하지만 공격자의 시스템 성능이 점점 증가하고, 공격기법이 다양해지면서 보다 발전된 데이터 보호 수단이 필요하게 되었다. 이를 해결하기 위해 Cloud Shredder와 같이 사용자가 소유하고 있는 로컬 저장장치와 클라우드 서비스의 원격 저장장치에 데이터를 분산 저장하는 방법 등 다양한 연구가 진행되고 있다. 이에 본 논문에서는 Cloud Shredder의 문제점을 알아보고 해당 문제점을 해결할 수 있는 개선된 방법을 제안한다.

Key Words : Secret Sharing, Data Protection, Cloud, Cloud Shredder

ABSTRACT

Recently, almost all data has stored and managed in the shape of digital as development of IT infrastructure. Digital data is able to store the huge data in very small space. but if the disk should be stolen or lost, it would have many secure problems such as data leakage. Currently, digital data is protected by encryption method to prevent data leakage. However, the encryption method is not enough to protect data because the performance of attack system is higher and the attack methods is various. Therefore, there is a need for a new advanced data protection method. To solve secure problems, many research has been progressed like Cloud Shredder, which distributes data and then store. In this paper, We found out the problem of the Cloud Shredder and proposed an advanced method of digital data protection to solve those problem.

Ⅰ. 서 론

IT 인프라가 발전하면서 기존의 출력물로 관리되었던 많은 양의 데이터가 디지털 데이터로 변환되어 저장 및 관리되고 있다. 특히 음악 또는 동영상과 같은 일반적인 데이터뿐만 아니라 기업 기밀 자료, 고객들의 개인 정보 등의 민감한 정보들도 디지털 형태로 저장되고 있다. 이러한 디지털 데이터는

※"본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (NIPA-2013-H0301-13-1003)
◆ 주저자 : 고려대학교 정보보호대학원 정보보증 연구실, minsoon2@korea.ac.kr, 학생회원
° 교신저자 : 고려대학교 정보보호대학원 정보보증 연구실, skim71@korea.ac.kr, 종신회원
* 고려대학교 정보보호대학원 정보보증 연구실, korhoon@korea.ac.kr
논문번호 : KICS2013-04-185, 접수일자 : 2013년 4월 25일, 최종논문접수일자 : 2013년 5월 8일

기존의 아날로그 데이터보다 작은 공간을 사용하고, 쉽고 빠르게 생성 및 수정, 공유 할 수 있는 장점이 있다. 하지만 USB 메모리와 같이 매우 작은 저장장치에 수백GB의 자료를 저장할 수 있어, 중요한 정보가 저장된 장치를 분실하거나 도난당할 경우 매우 큰 피해를 입을 수 있다. 특히 데이터 유출이 발생하는 주요 경로에 노트북 또는 이동식 디스크와 같은 저장 장치의 도난 및 분실이 높은 비율을 차지하고 있다[1].

이와 같은 데이터 유출을 방지하기 위해 다양한 방법이 사용되고 있으며, 주로 데이터 암호화 방법을 사용한다. 하지만 데이터 암호화 방법은 해당 데이터의 유출 자체를 막는 것이 아니라 데이터가 유출되어도 해당 데이터의 내용을 획득할 수 없도록 하는 것을 목적으로 하는 한계가 있다. 이를 개선한 방법으로 Nan Zhang이 IEEE TrustCom2011에서 Cloud Shredder를 제안 하였다[7]. 본 논문에서는 Cloud Shredder의 특징과 문제점을 알아보고, 개선된 데이터 보호 방법을 제안한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 관련 연구로 Cloud Shredder와 Shamir의 비밀 분산 방식에 대해 알아보고, Ⅲ장에서 기존 Cloud Shredder 방식의 문제점에 대해 설명한다. Ⅳ장에서는 기존 Cloud Shredder 방식의 문제점을 해결할 수 있는 개선된 Cloud Shredder 방식에 대해 알아본다. Ⅴ장에서는 개선된 Cloud Shredder 방식과 기존 Cloud Shredder 방식을 비교하고, Ⅵ장에서 결론을 맺는다.

Ⅱ. 관련연구

2.1. Cloud Shredder

Cloud Shredder는 2011 International Joint Conference of IEEE TrustCom에서 발표된 내용으로 기존의 데이터 보호 방법인 "Full Disk Encryption" (이하 FDE)의 문제점을 지적하고 이를 해결하기 위한 방법으로 제안되었다. FDE는 현재 널리 사용되는 데이터 보호 방법으로 FDE를 사용한 대표적인 제품으로는 Microsoft Windows에서 제공되는 BitLocker가 있다[2]. FDE에서 암호화 키의 길이는 "계산적 안전성(Computational Security)"에 의존하기 때문에 안전하다고 할 수 있다. 하지만 암호화된 데이터가 도난당했다는 사실에는 변함이 없고, 해당 데이터의 내용이 공격자에게 매우 의미 있는 정보일 경우 FDE storage key의 획득을 위해 전수조사 공격과 같은 offline Attack을 시간의 제한 없이 시도할 수 있다. 특히 BitLocker를 비롯한 대부분의 상용 암호화 제품의 경우 사용자가 입력한 패스워드로부터 암호화 키를 추출해내기 때문에 아주 우수한 성능의 컴퓨팅 환경이 갖추어진다면 전수조사 공격이 충분히 가능하다. Cloud Shredder는 공격자가 해당 데이터를 습득할 경우 유출된 데이터를 제어할 수 없는 문제점을 해결하기 위해 데이터가 유출되었을 때 해당 데이터를 쓸모없도록 하는 것을 목표로 한다[7].

이를 위해 Cloud Shredder는 XOR-method 또는 Ratio-method를 이용하여 데이터를 LS(Local Share)와 RS(Remote Share)로 분할된 후 LS는 사용자가 소지하고 RS는 클라우드 서비스에 저장한다. 이때 공격자가 LS와 RS 중 하나를 획득하더라도 나머지 하나를 획득하지 못한다면 LS와 RS로 나누기 전의 온전한 데이터를 획득할 수 없다. 이때 클라우드 서비스를 이용하여 RS를 저장하므로 클라우드 서비스를 이용하여 RS를 쉽게 전송할 수 있고, 로컬 장치의 도난으로부터 안전하게 보호할 수 있다. 만약 장치를 도난당해 LS가 유출된다면 사용자는 클라우드 서비스에 저장된 RS를 삭제하여 LS를 사용할 수 없도록 만들 수 있다. 반대로 클라우드 서비스에 LS가 유출 된다면 사용자가 소지한 RS를 삭제하여 유출된 LS를 쓸모 없는 데이터로 만들 수 있다. 이를 통해 FDE의 문제점이었던 데이터 유출시 해당 데이터에 대한 제어권이 공격자에게 넘어가는 것을 해결할 수 있다. 또 클라우드 서비스를 이용하여 네트워크에 접속할 수 있다면 RS에 접근할 수 있어 데이터 분할에 따른 불편함을 감소 시켰다[7]. Cloud Shredder의 전체 구성은 그림 1과 같다.

Cloud Shredder는 Console, Container, Safe Folder, Kernel Module, Cloud Interface로 구성되어 있고, Cloud Shredder의 동작은 다음과 같이 이루어진다. 먼저 Console을 통해 Amazon S3 Credential

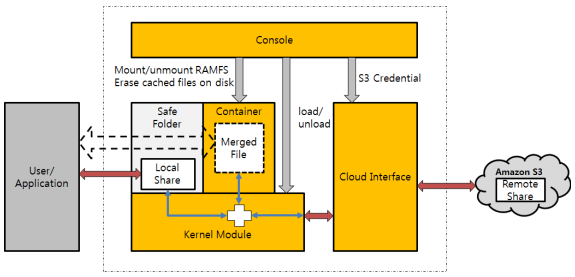


그림 1. Cloud Shredder의 구조[7]
Fig. 1. Architecture of the cloud shredder[7]

을 Cloud Interface에 전달하면 Cloud Interface는 Amazon S3에 접속하고, 해당 클라우드 서비스에 저장된 RS를 다운로드 한다. RS가 다운로드 되면 Kernel Module은 사용자가 Safe Folder에 저장한 LS와 Cloud Interface로부터 전달된 RS를 Container에 전달한다. Container는 전달된 LS와 RS를 조합하여 분할되기 전 원래의 파일을 생성한다. 이때 사용된 클라우드 서비스는 Amazon S3이다[7].

표 1은 Cloud Shredder에서 제공하는 XOR-method와 Ratio-method를 사용 하여 파일을 분할할 때 소요된 시간을 보여준다. 이때 Td는 파일 분할시 소요된 시간을 의미하고, Tm은 분할된 파일을 재조합 할 때 소요된 시간을 의미한다.

2.2. Shamir의 비밀 분산 방식

Shamir의 비밀 분산 방식은 1979년 Shamir에 의해 발표되었다[1]. Shamir는 다항식 보간법(Polynomial Interpolation)을 이용한 비밀 분산 방법을 제안하였고, 데이터 D를 n조각으로 나누면 k(k≤n)개 이상을 모을 경우 원래의 데이터 D를 복원할 수 있지만. k-1개 이하의 조각으로부터는 D에 대한 아무런 정보도 얻을 수 없다. 이와 같은 비밀 분산 방식을 (k, n) Threshold 방식이라고 한다. 해당 내용은 다음과 같다.

- p : $p \geq n+1$ 인 큰 소수(단, n은 비밀분산에 참여하는 전체 참가자의 수)
- M : 분산하고자 하는 비밀, $M \in GF(p)$
- D : 각 참가자에게 부분정보(정보 조각, Share)를 전달하는 분배자
- P : 전체 참가자의 집합, $P = \{P_1, P_2, \ldots P_n\}$
- S_i : 참가자의 P_i에게 분배하는 분할정보

2.2.1. 비밀 분산 과정 :

① 정보 분배자는 $GF(p)$상에서 0이 아닌 n개의 원소 $x_1, x_2, \ldots, x_n$을 랜덤하게 선택한다.

② 정보 분배자는 $GF(p)$상에서 $a_1, a_2, \ldots, a_n$을 랜덤하게 선택하고 다음과 같이 $(k-1)$차 다항식을 생성한다.

$$f(x) = M + a_1 x + a_2 x^2 + \ldots + a_{m-1} x^{m-1} \quad (1)$$

③ 정보 분배자는 각 참가자 $P_1, P_2, \ldots, P_n$에게 분배할 부분정보 $S_i (1 \leq i \leq n)$을 다음과 같이 계산하여 $P_i (1 \leq i \leq n)$에게 안전하게 전송한다.

$$S_i = f(x_i) (1 \leq i \leq n) \quad (2)$$

④ $x_1, x_2, \ldots, x_n$는 공개하고, $a_1, a_2, \ldots, a_n$와 부분정보 $S_i (1 \leq i \leq n)$는 안전하게 보관한다.

2.2.2. 비밀복원 과정 :

비밀 복원에 참여하는 참가자의 집합을 $P = \{P_1, P_2, \ldots P_t\}$라 한다.

① k명의 참가자들은 Lagrange의 다항식 보간법에 의해 $f(x)$상의 k개의 서로 다른 점 $(x_i, s_i) (1 \leq i \leq k)$를 이용하여 다음과 같이 $f(x)$를 계산한다.

$$f(x) = \sum_{i=1}^{k} y_i \quad (3)$$

$$\prod_{i \leq j \leq k, j \neq i} \frac{x - x_j}{x_i - x_j} \quad (4)$$

② 복원하고자 하는 비밀 $M = f(0)$은 다음과 같이 계산될 수 있다.

표 1. Cloud Shredder를 이용한 파일 분할 및 재조합 시간 (시간 : 초)[7]
Table 1. File splitting and reconstruction time using Cloud Shredder (Time : sec)[7]

	500k		1M		5M	
	XOR-method	Ratio-method	XOR-method	Ratio-method	XOR-method	Ratio-method
Td	0.39	0.21	0.73	0.22	3.45	0.33
Tm	0.09	0.05	0.17	0.10	0.91	0.50

$$M = \sum_{i=1}^{k} c_j y_i \quad , (\text{단}, \ C_j = \sum_{i \le j \le k} \frac{x_j}{x_j - x_i}) \quad (5)$$

Shamir의 비밀분산 방식이 발표된 이후 다중 비밀 분산 등 다양한 비밀 분산 방법의 연구가 이루어지고 있다[5,6]. 본 논문에서 제안한 개선된 Cloud Shredder 방식은 Shamir의 비밀 분산 방식을 이용하여 파일을 분할하므로 비밀 분산 방식이 가지고 있는 높은 데이터 기밀성을 제공할 수 있다. 이때 크기가 큰 데이터를 분할한다면 매우 큰 p값이 필요 하다. 예를 들어 5Mbytes의 크기를 가지는 파일을 분할한다면 5M이상의 소수가 필요하며 이는 현실적으로 어렵다. 따라서 이러한 문제를 해결하기 위해 제안된 방법에서는 해당 데이터 내용을 1byte씩 읽어 들여 비밀분산 방식을 적용하였다.

Ⅲ. 기존 Cloud Shredder 방식의 문제점

Cloud Shredder의 경우에는 하나의 데이터를 LS와 RS로 분리하여 사용자의 로컬 저장장치와 클라우드 서비스의 원격 저장장치에 나누어 저장하는 방법을 사용한다. 원래의 데이터를 획득하기 위해서는 2개로 분리된 LS와 RS를 재조합하여 원래의 데이터로 생성해야 한다. 이때 LS는 데이터 소유자의 로컬 저장장치에 저장되므로 RS와 재조합을 하기 위해서는 항상 소지해야 한다. 이로 인해 클라우드 서비스를 사용할 때 가장 큰 장점인 데이터 가용성이 낮아지게 되는 문제가 발생한다. 예를 들어 사용자가 LS를 저장한 시스템을 사용하지 않거나 LS가 저장된 이동식 저장장치를 다른 곳에 둘 경우 클라우드 서비스에 접근하여 해당 클라우드 서비스에 저장된 RS를 다운로드 하더라도 LS와 재조합을 할 수 없기 때문에 원래의 데이터를 획득할 수 없다. 따라서 사용자는 해당 데이터를 사용하기 위해서 항상 LS를 소지해야 하므로 클라우드 서비스의 높은 가용성을 활용할 수 없다.

다음 장에서는 이러한 Cloud Shredder의 한계를 해결하여 데이터의 높은 가용성을 유지한 채 데이터를 보호할 수 있는 개선된 Cloud Shredder 방식을 제안한다.

Ⅳ. 개선된 Cloud Shredder 방식

4.1. 개선된 Cloud Shredder 방식

본 논문에서 제안하는 개선된 Cloud Shredder 방식은 비밀분산과 클라우드 서비스를 이용한 방법으로 하나의 파일을 다수의 조각으로 분할하고, 분할된 조각들을 서로 다른 클라우드 서비스에 저장하여 기존의 데이터 보호 방법들이 제공하지 못했던 높은 데이터 가용성과 기밀성을 동시에 제공한다. 또한 보호하고자 하는 데이터가 사용자의 저장장치에 저장되지 않고, 클라우드 서비스에 저장되므로 저장장치의 분실 및 도난으로 인한 데이터 유출을 막을 수 있다. 또 제안된 방법은 Cloud Shredder뿐만 아니라 클라우드 서비스[11], 데이터 암호화 방법의 장점을 포함하고, 단점은 개선한다.

4.2. 구조

본 논문에서 제안하는 개선된 Cloud Shredder 방식의 전체 구조는 그림 2와 같다.

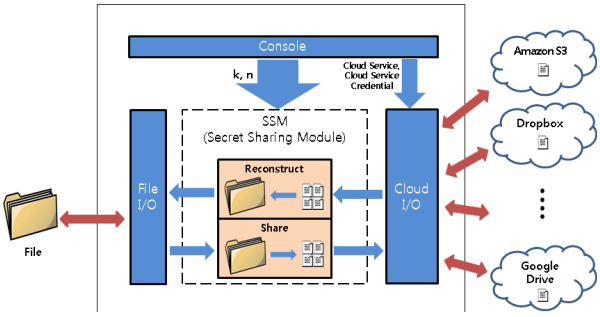


그림 2. 개선된 Cloud Shredder 방식의 구조
Fig. 2. Architecture of the Our Cloud Shredder

제안된 방법은 Console, File I/O, Secret Sharing Module, Cloud I/O로 구성되어 있으며, 각각의 기능은 아래와 같다.

Console

- 사용자로부터 동작(Upload/Download) 유형을 입력받는다.
- Upload : 사용자로부터 접근할 클라우드 서비스와 인증 정보를 입력 받는다. 파일 분할에 필요한 분할할 조각의 수(n)와 재조합에 필요한 조각의 수(k)를 입력받는다.
- Download : 사용자로부터 접근할 클라우드 서비스와 사용자 인증 정보를 입력받는다.

File I/O

- Upload : 사용자로부터 보호할 파일을 입력받고, 해당 파일을 분할하기 위해 Secret Sharing Module로 전달한다.
- Download : Secret Sharing Module로부터 전

달받은 파일을 시스템에 저장한다.

Secret Sharing Module

- Upload : File I/O로부터 전달받은 파일을 사용자가 입력한 k, n 값을 이용하여 n개의 조각으로 분할한다. 이때 분할된 조각은 k개 이상을 모을 경우 재결합할 수 있다.
- Download : Cloud I/O로부터 전달받은 다수의 조각을 재결합하여 온전한 하나의 파일을 생성한다. 이때 해당 조각은 k개 이상이어야 하며 k-1개 이하일 경우 파일 생성이 불가능하다.

Cloud I/O

- Upload : 각 클라우드 서비스의 인증 페이지에 연결시켜 주거나 사용자로부터 미리 입력된 인증 정보를 이용하여 로그인을 시도한다. 로그인이 완료되면 Secret Sharing Module로부터 분할된 조각들을 전달받아 다수의 Cloud 서비스에 업로드 한다.
- Download : 각 클라우드 서비스의 인증 페이지에 연결시켜 주거나 사용자로부터 미리 입력된 인증 정보를 이용하여 로그인을 시도한다. 로그인이 완료되면 분할된 조각들이 저장되어 있는 Cloud 서비스에 접근하고, 해당 조각들을 다운로드 하여 Secret Sharing Module로 전달한다.

4.3. 동작절차

제안한 방법은 클라우드 서비스를 사용하여 데이터를 저장한다. 따라서 데이터를 따로 소지할 필요 없이 네트워크에 연결된 장치를 통해 각각의 클라우드 서비스에 접근하여 분할된 조각들을 다운로드 하고, 각 조각들을 재조합하면 분할하기 전 원래의 데이터를 획득할 수 있다. 이러한 일련의 과정은 그림 3과 같이 파일 다운로드와 업로드로 나누어 진행된다.

먼저 파일 다운로드 시 사용자는 분할된 조각이 저장되어 있는 클라우드 서비스를 선택하고, 해당 클라우드 서비스의 ID와 Password를 입력한다. 올바른 ID와 Password가 입력되면 클라우드 서비스에 접근할 수 있다. 이후 클라우드 서비스에 파일 조각이 저장되어 있는지 확인하고, 해당 파일 조각을 다운로드한다. 이때 파일 조각이 저장되어 있지 않다면 에러 메시지를 출력하고 종료한다. 이러한 과정

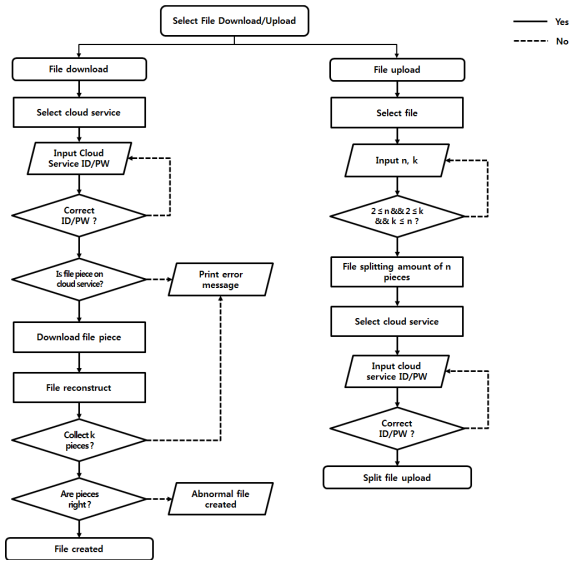


그림 3. 개선된 Cloud Shredder 방식의 동작 절차
Fig. 3. Flowchart of the Our Cloud Shredder

은 각 클라우드 서비스에서 반복적으로 이루어지고, 모든 조각이 다운로드 되면 파일 재조합을 시작한다. 이때 k개의 조각이 모여 있지 않다면 에러 메시지가 출력되고, 종료된다. k개의 조각이 모여 있다면 분할되기 전 원래의 파일을 생성한다. 하지만 파일 재조합에 사용된 조각 중 하나라도 올바르지 않은 조각이 있다면 비정상적인 파일이 생성된다.

파일 업로드를 선택할 경우에 사용자는 분할하려는 파일을 선택하고, 해당 파일을 분할하기 위해 n값과 k값을 입력한다. 이때 n과 k의 값이 $2 \le n, 2 \le k, k \le n$을 만족하지 않는다면 올바른 n, k값을 다시 입력받는다. n, k의 값이 올바르게 입력되었다면 n개의 조각으로 파일을 분할한다. 파일 분할이 완료되면 각 파일 조각들을 업로드 할 서로 다른 n개의 클라우드 서비스를 선택한다. 사용자는 선택된 클라우드 서비스들의 ID와 Password를 입력하고, ID와 Password가 올바르다면 해당 클라우드 서비스에 접근한 후 파일 조각들을 업로드 한다.

4.4. 데이터 관리

본 논문에서 제안한 개선된 Cloud Shredder 방식을 사용하면 데이터를 안전하게 보호할 수 있고, 유출된 데이터의 처리도 해결할 수 있다. 유출된 데이터의 처리란 데이터 조각 중 일부가 유출 되었을 경우 해당 조각을 쓸모없도록 처리하여 사용 할 수 없게 만드는 것을 말한다.

4.4.1. 데이터 보호.

개선된 Cloud Shredder 방식은 데이터 유출이 발생할 수 있는 여러 가지 상황에서 기존의 데이터 보호 방법과 Cloud Shredder가 제공하는 데이터 보호 수준을 모두 만족하고 있고, 데이터 암호화와 Cloud Shredder에서 제공하지 못하는 높은 데이터 가용성도 제공한다. 표 2는 데이터 유출이 발생할 수 있는 여러 가지 상황에서 현재 일반적으로 사용되는 클라우드 서비스, 데이터 암호화와 새롭게 제안된 Cloud Shredder, 본 논문에서 제안하는 개선된 Cloud Shredder 방식의 데이터 보호 수준 및 데이터 가용성을 비교한 결과이다.

이때 각 공격 및 데이터 유출 경로에 대한 보안성을 High(H), Medium(M), Low(L)로 평가하였다. 이때 해당 공격 및 경로로는 데이터가 유출 될 수 없거나 유출된 데이터를 쓸모없도록 처리할 수 있을 경우 H 값을 갖는다. ID, Password와 같이 많은 시간이 소요 되지만 전수 조사 공격이 가능하다면 M 값을 갖고, 해당 공격 및 경로를 통해 데이터 내용이 유출 된다면 L 값을 갖는다. 데이터 가용성의 경우 데이터를 항상 소지해야 한다면 낮은 데이터 가용성을 가지므로 L 값을 가지고, 데이터를 항상 소지하지 않아도 필요할 때 해당 데이터에 접근할 수 있다면 H 값을 가진다.

먼저 클라우드 서비스는 데이터를 보호하기 위한 수단으로 ID와 Password를 사용한다. 따라서 공격자가 클라우드 서비스의 ID와 Password를 획득할 경우 해당 서비스에 저장된 모든 데이터가 유출될 수 있다. 특히 사용자가 단순하고 짧은 Password를 사용할 경우 전수조사 공격에 의한 데이터 유출 가능성이 매우 높아진다. 또 데이터를 저장한 클라우드 서비스에서 보안 사고가 발생하여 인증 시스템이 무력화 될 경우 대량의 데이터가 유출될 수 있다.

데이터 암호화의 경우에는 암호화 Key의 안전성에 의존한다. 하지만 데이터가 로컬 장치에 저장되고, 자체적인 데이터 유출 방지 기능이 없으므로 데이터가 저장된 장치를 분실하거나 도난당할 경우 암호화된 데이터가 쉽게 유출될 수 있다. 그리고 데이터 유출이 발생할 경우 유출된 데이터의 처리를 위한 방법이 제공되지 않는다. 따라서 암호화된 데이터를 공격자가 획득하게 된다면 전수조사 공격과 같은 복호화된 데이터 획득을 위한 시도를 아무런 시간제한 없이 실행할 수 있으므로 데이터가 유출될 수 있는 위험을 가지고 있다. 더욱이 대부분의 상용 암호화 제품의 경우 사용자가 입력한 패스워드로부터 암호화 키를 추출해내기 때문에 아주 우수한 성능의 컴퓨팅 환경이 갖추어진다면 전수조사 공격이 충분히 가능하다. BitLocker의 경우 잘못된 Password를 반복적으로 입력하여도 CAPTCHA를 요구하거나 일정시간 Password를 입력하지 못하도록 하는 등 전수조사 공격을 방어하기 위한 추가적인 수단을 제공하지 않는다. 따라서 공격자는 직접 CAPTCHA를 입력하거나, Password 입력 불가능 시간이 지난 후 다시 Password를 입력하는 등의 추가적인 노력 없이 전수조사 공격을 시도할 수 있다. 그리고 해당 데이터를 사용하기 위해서는 데이터가 저장된 이동식 저장장치 등을 항상 소지해야 하므로 낮은 데이터 가용성을 가진다.

Cloud Shredder의 경우에는 데이터를 2개의 LS와 RS로 분할하고, 분할된 조각을 각각 클라우드 서비스와 로컬 장치에 저장하는 방법을 사용한다.

표 2. 개선된 Cloud Shredder 방식과 다른 데이터 보호방법 비교 (데이터 보호 수준을 상,중,하로 표현)
Table 2. Comparison with Our Cloud Shredder and other data protection methods (Data protection level : High(H), Medium(M), Low(L))

	Cloud Service	Data Encryption	Cloud Shredder	Our Cloud Shredder
Brute-Force (ID/PW/Key)	M (Depends on ID/PW)	M (Depends on Key)	H	H
Storage device theft or loss	H (N/A)	M (Encryption Data leakage)	H (Requires RS)	H (N/A)
The cloud service provider's security incident	L (Data leakage)	M (Encryption Data leakage)	H (Requires LS)	H (Require other pieces of the data)
Availability of data	H	L (Must have File)	L (Must have LS)	H

따라서 LS가 저장된 장치를 분실하거나 도난당하더라도 클라우드 서비스에 저장된 RS를 획득하지 못한다면 LS로부터 아무런 정보도 획득할 수 없다. 반대로 클라우드 서비스의 ID와 Password가 노출되거나 해당 클라우드 서비스에 보안 사고가 발생하여 RS가 유출 되더라도 LS를 구하지 못한다면 RS로부터 정보를 획득할 수 없으므로 저장장치의 도난 및 분실, 클라우드 서비스 제공자의 보안사고로부터 안전하다고 할 수 있다. 하지만 Cloud Shredder의 경우 데이터를 사용하기 위해서는 LS와 RS가 모두 필요하다. 따라서 LS를 항상 소지해야하므로 데이터 가용성 낮은 문제가 있다.

개선된 Cloud Shredder 방식을 사용하면 Cloud Shredder와 동일한 수준으로 데이터를 보호할 수 있고, 높은 데이터 가용성도 제공받을 수 있다. 예를 들어 사용자가 다수의 클라우드 서비스의 ID와 Password를 다르게 설정한다면 공격자가 하나의 클라우드 서비스에 대한 ID와 Password를 획득하더라도 하나의 데이터 조각만 획득할 수 있다. 하지만 온전한 파일을 획득하기 위해서는 다른 클라우드 서비스에 저장된 k개의 조각들이 필요하다. 따라서 공격자는 각 클라우드 서비스에 접근할 수 있는 ID와 Password를 모두 획득해야 하므로 각 클라우드 서비스에 대한 추가적인 공격이 필요하다. 이는 공격자로 하여금 많은 시간과 노력을 소요하게 만든다. 더욱이 데이터 소유자가 Password를 주기적으로 변경한다면 Password가 변경되기 전까지 제한된 시간 내에 모든 클라우드 서비스의 ID와 Password를 획득해야 하는 어려움이 있다. 특히 데이터를 로컬 디스크에 저장하지 않고, 클라우드 서비스에 저장하기 때문에 노트북, USB 메모리와 같은 데이터가 저장된 기기나 저장장치를 분실하여도 데이터를 안전하게 보호할 수 있다. 또 네트워크에 연결만 할 수 있다면 데이터 조각들을 다운로드하여 원래의 파일을 생성할 수 있기 때문에 Cloud Shredder가 제공하지 못했던 높은 데이터 가용성도 제공받을 수 있다.

4.4.2. 유출된 데이터의 처리

표 3은 각 데이터 관리 및 보호 방법들의 유출된 데이터의 처리 가능 여부를 보여준다.

클라우드 서비스와 데이터 암호화의 경우 유출된 데이터를 더 이상 사용할 수 없도록 하는 데이터 처리 기능을 제공하지 않는다. 즉 클라우드 서비스에 저장된 데이터가 유출된다면 해당 데이터에 대한 제어권을 공격자가 획득하게 되고, 해당 데이터가 암호화가 되었을 경우 시간의 제한 없이 데이터 획득을 위한 여러 가지 방법을 시도할 수 있다.

표 3. 유출된 데이터의 처리 가능 여부
Table 3. Processing of the leakage data

	Cloud Service	Data Encryption	Cloud Shredder	Our Cloud Shredder
Processing of the leakage data	unavailable	unavailable	available	available

Cloud Shredder의 경우 LS가 유출된다면 클라우드 서비스에 저장된 RS를 삭제하여 유출된 LS를 더 이상 사용할 수 없도록 할 수 있다. 반대로 RS가 유출된다면 사용자가 소지한 LS를 삭제하여 유출된 RS를 사용할 수 없도록 할 수 있다.

개선된 Cloud Shredder 방식의 경우에도 데이터 처리가 가능하다. 예를 들어 데이터 조각을 저장한 다수의 클라우드 서비스들 중 일부 클라우드 서비스의 ID와 Passwor가 노출되거나 보안 사고가 발생하여 데이터 조각이 유출될 경우 다른 클라우드 서비스에 저장된 조각을 삭제한다면 추가적인 데이터 조각의 유출을 막을 수 있고, 유출된 데이터 조각은 더 이상 쓸모가 없게 된다.

V. 구현결과

본 논문에서 제안한 개선된 Cloud Shredder 방식은 C와 Python으로 작성 되었다. Console과 File I/O, Secret Sharing Module은 C로 작성되어 파일 분할 및 재조합 시 빠른 속도를 제공하고, Cloud I/O는 각 클라우드 서비스에서 제공하는 Python API로 작성되었다. 그림 4는 개선된 Cloud Shredder 방식을 이용하여 Amazon S3와 Dropbox에 접근하는 모습을 보여준다. 이때 사용자 인증은 Amazon S3는 미리 입력된 인증 정보를 사용하였고, Dropbox는 해당 서비스에서 제공하는 사용자 인증 페이지를 이용하였다.

표 4는 Cloud Shredder와 개선된 Cloud Shredder 방식의 파일 재조합(Tm) 속도를 비교한 결과이다. 테스트에 사용된 시스템은 Core 2 Duo 2.5Ghz, 4GB 800Hz DDR2 RAM, Windows 7 운영체제가 사용되었고, 파일 분할에 사용된 n, k의 값은 n=2, k=2이다. 기존 Cloud Shredder 방식의 경우 XOR와

표 4. 기존 Cloud Shredder와 개선된 Cloud Shredder 방식의 속도 비교 (크기 : bytes, 시간 : 초)
Table 4. Compare the speed of the Cloud Shredder and Our Cloud Shredder (Size : bytes, Time : sec)

	500k		1M		5M	
Tm	Cloud Shredder	Our Cloud Shredder	Cloud Shredder	Our Cloud Shredder	Cloud Shredder	Our Cloud Shredder
	0.09(XOR) 0.05(Ratio)	0.04	0.17(XOR) 0.10(Ratio)	0.10	0.91(XOR) 0.50(Ratio)	0.45

Ratio-Method 방식의 데이터 분할 및 재조합을 지원하고, 개선된 Cloud Shredder 방식은 Shamir의 비밀 분산 방식을 이용한다. 기존 Cloud Shredder 방식과 비교 결과 XOR, Ratio-Method 방식과 상관없이 본 논문에서 제안한 개선된 Cloud Shredder 방식이 비슷하거나 더 빠른 것을 확인할 수 있다.

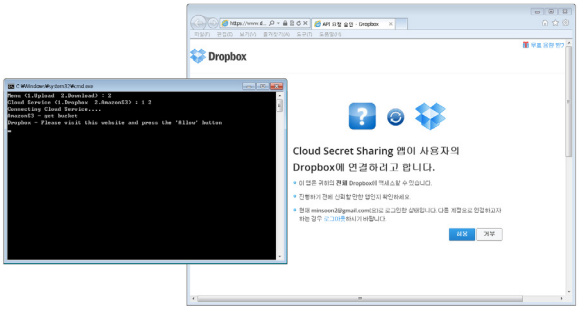


그림 4. 개선된 Cloud Shredder 방식의 실행
Fig. 4. Our Cloud Shredder execution

표 5는 실제 클라우드 서비스에서 개선된 Cloud Shredder 방식을 사용하여 2개의 조각을 다운로드하고 재조합하여 원래의 파일이 생성될 때 까지 소요된 시간을 보여준다. 이때 사용된 클라우드 서비스는 Dropbox, Amazon S3이다[9,10]. 이때 클라우드 서비스는 네트워크를 통해 제공되므로 사용자의 네트워크 환경에 따라 결과가 달라질 수 있다. 실험 결과 개선된 Cloud Shredder 방식을 이용하면 2개의 클라우드 서비스에 접근한 후 해당 클라우드 서비스에 저장된 조각을 다운로드하고 재조합하여 원래의 파일을 생성하는데 평균 24.61초가 소요되었다.

표 5. 개선된 Cloud Shredder의 파일 생성 시간 (크기 : bytes, 시간 : 초)
Table 5. File creation time using the Our Cloud Shredder (Size : bytes, Time : sec)

	100k	1M	5M
File creation time	23.37	24.59	25.87

Ⅵ. 결 론

본 논문에서는 기존의 파일 보호 방법들과 새롭게 제안된 Cloud Shredder의 문제점을 알아보고, 이를 해결하기 위해 새로운 데이터 보호 방법인 개선된 Cloud Shredder 방식을 제안하였다. 제안된 방법은 클라우드 환경의 장점인 데이터 가용성을 보장하면서 높은 데이터 기밀성을 제공하였다. 또 데이터 보호뿐만 아니라 기존의 데이터 보호 방법에서 제공하지 못했던 데이터 폐기 문제도 해결할 수 있었다.

향후 더 높은 보안성 제공을 위해 보다 많은 클라우드 서비스와의 호환성을 제공하여 분할 가능한 조각의 수를 늘리고, 이를 통해 조각의 일부가 삭제되거나 손상되어도 나머지 조각만을 이용하여 원래의 데이터를 획득할 수 있는 데이터 안전성을 제공하도록 할 것이다. 또 Ramdisk를 활용한 파일 분할 및 재조합을 구현하여 로컬 시스템에 데이터의 흔적이 남지 않도록 하고, GUI(Graphic User Interface)를 제공하여 사용자의 편의성을 높일 예정이다.

References

[1] U. Berger, *CSI/FBI Computer Crime and Security Survey. 2011-2012*, Retrieved April, 28, 2011, from http://gocsi.com.

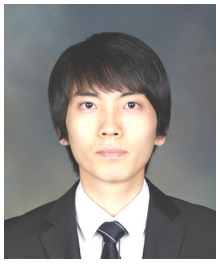
[2] Microsoft, *Windows bitlocker drive encryption frequently asked questions*, Retrieved April, 30, 2009, from http://technet.microsoft.com/enus/library/cc766200(WS.10).aspx

[3] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and Grid computing 360-degree compared," in *Proc. Grid Comput. Environment Workshop(GCE'08)*, pp. 1-10, Austin, U.S.A., Nov. 2008.

[4] NBS, "Data Encryption Standard. Federal

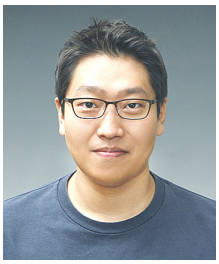
Information Processing Standard Pub," 1977
[5] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, pp. 612-613, Nov. 1979.
[6] Shamir's Secret Sharing Scheme, [Online], Available: http://point-at-infinity.org/ssss,
[7] N. Zhang, J. Jing, and P. Liu, "CLOUD SHREDDER: Removing the laptop on-road data disclosure threat in the cloud computing era," in *Proc. IEEE Int. Conf. Security and Privacy Comput. Commun.*, pp. 16-18, Changsha, China, Nov. 2011.
[8] A. Zonenberg, "Distributed hash cracker: A cross-platform GPU-accelerated password recovery system," *Rensselaer Polytechnic Institute*, Apr. 2009.
[9] Dropbox, [Online], Available: https://www.dropbox.com,
[10] Amazon, [Online], Available: http://www.amazon.com,
[11] Y. Cho, Y. Seo, and Y. Kim, "The cloud computing service and the advanced network," in *Proc. KICS Fall Conf.*, pp. 158~159, Seoul, Korea, Nov. 2010.

박 민 수 (Minsu Park)



2010년 2월 신라대학교 컴퓨터네트워크학과 졸업
2013년 2월 고려대학교 정보보호대학원 석사
2013년 3월~현재 고려대학교 정보보호대학원 박사과정
<관심분야> 정보보증, 정보보호제품 보안성 평가, 디지털 포렌식, Usable Security

강 성 훈 (Sunghoon Kang)



2010년 2월 서원대학교 컴퓨터공학과 졸업
2013년 2월 고려대학교 정보보호대학원 석사
2013년 3월~현재 고려대학교 정보보호대학원 박사과정
<관심분야> 정보보증, 정보보호제품 보안성 평가, 보안공학, Usable Security

김 승 주 (Seungjoo Kim)

1994년 2월 성균관대학교 정보공학과 졸업
1996년 2월 성균관대학교 정보공학과 석사
1999년 2월 성균관대학교 정보공학과 박사
1998년 12월~2004년 2월 KISA(舊한국정보보호진흥원) 팀장
2004년 3월~2011년 2월 성균관대학교 정보통신공학부 조교수, 부교수
2011년 3월~현재 고려대학교 사이버국방학과.정보보호대학원 정교수
2012년 3월~2012년 6월 선관위 디도스 특별검사팀 자문위원
<관심분야> 보안공학, 암호이론, 정보보증, 정보보호제품 보안성 평가, Usable Security