제36회 한국정보처리학회 추계학술발표대회 논문집 제18권 제2호 (2011. 11)

암호화를 이용한 낸드 플래시 메모리에서의 데이터 보호를 위한 설계

류시광, 김강석, 예홍진
아주대학교 일반대학원 지식정보보안학과
e-mail: ryuu2@ajou.ac.kr, kangskim@ajou.ac.kr, hjyeh@ajou.ac.kr

Architectural Design for Protecting Data in NAND Flash Memory using Encryption

Sikwang Ryu, Kangseok Kim, Hongjin Yeh
Dept. of Knowledge Information Security, Graduate School of Ajou University

요 약

최근 낸드 플래시 메모리 기술의 발전으로 플래시 메모리의 용량이 증가함에 따라 다양한 장치에서 데이터 저장소로 사용되고 있으며, 하드디스크를 대체할 저장 매체로서 주목을 받고 있다. 하지만 낸드 플래시 메모리의 특성으로 인해 데이터를 삭제하더라도 일정 기간 삭제된 데이터가 메모리에 남아 있게 되며, 이러한 특성으로 사용자의 중요 데이터가 보호되지 않은 상태로 저장되어 외부에 노출될 수 있다. 따라서 이런 특성을 보완하는 방법이 필요하며 본 논문에서는 낸드 플래시 메모리의 단점을 해결하기 위하여 낸드 플래시 메모리를 위한 시스템 소프트웨어인 FTL(Flash Translation Layer) 계층에서 암호화 알고리즘을 사용하여 데이터를 노출하지 않게 하는 방법을 제안한다.

1. 서론

최근 스마트 폰, MP3 플레이어, PMP, USB 메모리 등과 같은 개인용 멀티미디어 및 저장용 이동기기의 사용이 보편화되면서 저전력, 빠른 속도, 큰 용량, 저렴한 가격 등을 가진 낸드 플래시 메모리가 이러한 이동기기들의 저장소로 많이 사용되고 있다. 따라서 낸드 플래시 메모리에는 이동기기 사용자의 다양한 데이터가 저장될 것이며, 데이터 중에는 사용자의 개인 정보 등을 비롯한 중요 데이터가 있을 수 있다. 사용자는 이러한 중요 정보를 단지 이동기기에서 삭제하는 것으로서 정보가 노출되지 않을 것이라고 생각한다.

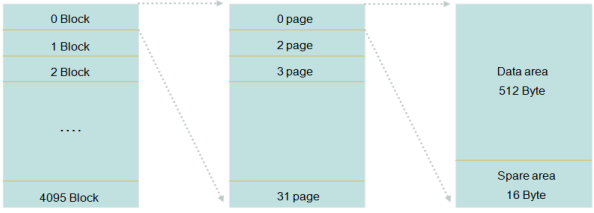
하지만 낸드 플래시 메모리의 특성으로 인해 저장된 데이터를 삭제하더라도 물리적으로는 데이터가 남아있게 된다[1]. 이러한 특성으로 인해 삭제된 데이터를 복구하는 것이 가능하며 개인의 중요 데이터가 저장될 수 있는 스마트 폰 및 휴대 기기의 경우 분실이나 도난으로 인해 이러한 데이터 노출의 위협이 있을 수 있다. 따라서 낸드 플래시 메모리를 저장소로 사용하는 기기에서 중요 데이터를 보호할 수 있는 방법이 요구된다.

이 논문에서는 삭제 연산 후에도 낸드 플래시 메모리상에 남아있게 되는 데이터를 노출하지 않기 위한 방법을 제안한다. 2장 관련 연구에서는 낸드 플래시 메모리의 구조 및 특성에 대해 알아보고, 데이터 보호를 위한 연구, 낸드 플래시 메모리 시스템 소프트웨어에 대해 알아본다. 3장 제안 방안에서는 암호화를 이용하여 낸드 플래시 메모리에서 중요 데이터를 보호하는 방법을 살펴볼 것이다. 그리고 마지막 결론에서는 진행상황과 보완되어야 할 점에 대해서 다룸으로써 마무리 할 것이다.

2. 관련 연구

2-1. 낸드 플래시 메모리의 구조 및 특성

낸드 플래시 메모리의 구조는 아래 (그림1)과 같으며 하나의 블록은 32개의 페이지를 가지며 각 페이지는 512 바이트의 data area와 추가 정보를 기록하는 16바이트의 spare area으로 나눌 수 있다.



(그림1) 낸드 플래시 메모리 구조

낸드 플래시 메모리의 특성으로 메모리에 쓰기 횟수 제한이 있어 대략 10만 번 정도 쓰게 되면 BAD 블록으로 변해 해당 블록을 사용할 수 없게 된다. 따라서 모든 메모리 블록들을 균등하게 사용하도록 하는 기술이 필요하다. 또한 같은 위치에 쓰기(in-place-update)가 불가능하여 이미 데이터가 기록된 위치에 다시 기록하기 위해서는

쓰기 연산 전에 삭제 연산을 하고 쓰기 연산을 수행해야만 한다. 따라서 낸드 플래시 메모리를 운영하는 파일 시스템이나 드라이버에서는 기존 데이터가 갱신될 시에 기존 자료를 무효화 처리하고 갱신된 데이터를 빈 페이지에 작성하는 방식을 사용한다. 그리고 쓰기 연산은 페이지를 단위로 하는 반면 삭제 연산의 경우 블록단위로 처리하기 때문에 삭제가 많아질수록 유효 페이지를 옮겨야 하는 오버헤드가 더 크게 발생하게 된다.

2-2. 데이터 보호를 위한 연구

데이터 보호를 위한 연구로 임의의 패턴으로 데이터를 여러 번 덮어쓰기 하여 알아볼 수 없게 함으로써 데이터를 완전 삭제하는 방법이 있다. 자기 디스크를 저장소로 사용할 시 데이터를 삭제하더라도 자기장의 잔상이 남게 되고, 이러한 잔상은 특수한 장비를 통해 검출이 가능하다. 따라서 임의의 패턴으로 여러 번 덮어쓰기를 통해 자기장의 잔상을 알아볼 수 없도록 하는 것이다. 이와 관련하여 어떤 패턴으로 몇 번 덮어써야 하는가에 관한 gutmann의 논문이 있다[2]. 하지만 낸드 플래시 메모리의 덮어쓰기 제한과 쓰기 횟수 제한에 의해 이러한 방식은 낸드 플래시 메모리의 수명을 줄일 수 있다.

다른 방법으로는 데이터를 암호화한 후 데이터를 암호화할 때 사용된 암호화 키를 완전히 삭제하는 방법이다[3]. 암호화 된 데이터를 키가 없이 복호화 하는 것은 상당히 어려운 것임이 증명되어 있으며, 내용을 알 수 없게 됨으로써 데이터가 보호된다고 볼 수 있다.

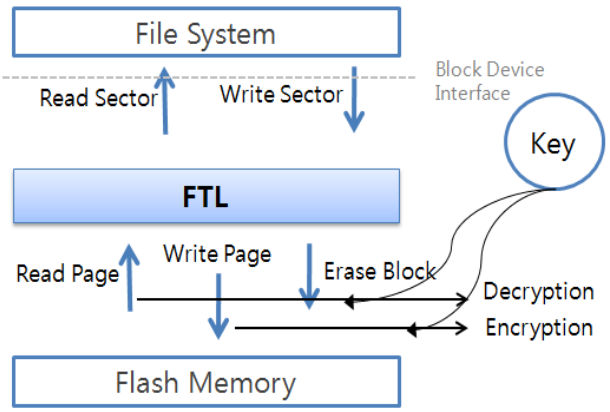
2-3. 시스템 소프트웨어

낸드 플래시 메모리의 특성으로 인해 플래시 메모리의 특성을 고려하여 메모리를 동작시키는 시스템 소프트웨어가 요구된다. 이러한 기능을 가진 시스템 소프트웨어에는 낸드 플래시 전용 파일 시스템인 YAFFS(Yet Another Flash File System)[4]와 기존 파일 시스템을 유지하면서 그 하위 계층에서 논리 주소를 물리 주소로 변환해주는 FTL[5][6]이 있다. 전용 파일 시스템인 YAFFS는 FTL에 비해 성능이 더 좋지만 운영체제에 따라 코드를 수정해야 하는 단점을 가지고 있다. 반면 FTL은 낸드 플래시 메모리만을 위한 파일 시스템인 YAFFS에 비해 성능이 뒤처지지만 기존의 파일 시스템을 그대로 사용하면서 낸드 플래시 메모리를 제어할 수 있는 유연성을 제공한다.

3. 제안 방안

낸드 플래시 메모리의 데이터 보호를 위한 기존의 연구 방식은 데이터를 암호화한 후 암호화 키를 한 곳에 모아 삭제하는 것이다[6]. 하지만 낸드 플래시 메모리의 특성으로 무효화된 페이지가 많은 블록을 소거하여 빈 블록을 생성하는 가비지 컬렉션(Garbage Collection)의 수행 전까지 암호화된 페이지와 키가 같이 낸드 플래시 메모리에 존재하게 된다. 따라서 가비지 컬렉션이 수행되지 않은 블록의 경우 따로 키 관리가 되지 않은 이상 데이터가 노출될 수 있다.

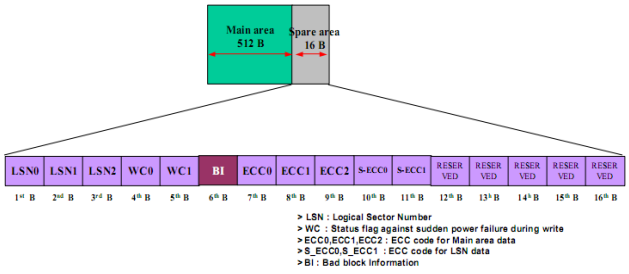
이 논문에서는 운영체제에 따라 코드를 수정하거나 작성해야 하는 YAFFS와 달리 기존의 파일 시스템을 사용할 수 있는 FTL 계층에서 데이터를 낸드 플래시 메모리에 저장, 읽기를 할 때 암호화 및 복호화를 수행함으로써 낸드 플래시 메모리의 블록에 저장되는 데이터들을 보호하는 방법을 제안한다. (그림2)는 제안하는 방식의 전체적인 흐름을 나타내고 있다.



(그림2) 제안 방식

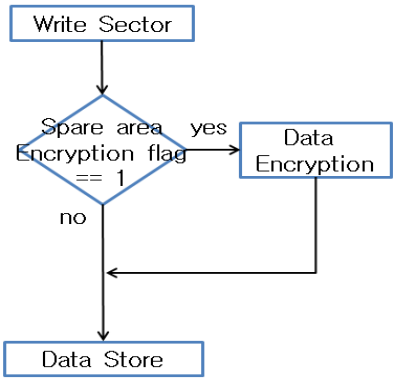
상위 파일 시스템에서 데이터를 낸드 플래시 메모리에 저장하거나 읽어들이기 위해 read sector, write sector를 호출하게 되고 FTL에서는 이러한 논리 주소를 물리 주소로 변환해주는 주소 매핑 테이블에서 해당 위치를 찾아 실제로 낸드 플래시 메모리에 저장하거나 저장된 데이터를 읽어오는 구간에서 미리 생성된 암호화 키를 이용하여 암호화 및 복호화를 수행함으로써 데이터가 노출되더라도 알아볼 수 없도록 하는 방식이다.

하지만 모든 입력, 출력 데이터를 암호화하게 되면 상당한 오버헤드가 발생하게 될 것이다. 따라서 오버헤드를 줄이기 위해 중요한 데이터만 암호화하며, 이러한 데이터의 암복호화 여부를 구분할 수 있는 표시가 필요하게 된다. 이를 해결하기 위해 낸드 플래시 메모리의 spare area을 사용한다. 읽고 쓰는 단위인 페이지는 데이터를 저장할 수 있는 512바이트의 data area와 16바이트의 spare area가 있다. 이 spare area 중 사용되지 않는 공간을 암호화여부를 알 수 있는 flag로서 사용하는 것이다.



(그림3) 삼성 낸드 플래시 메모리 spare area

(그림3)은 spare area의 사용 공간을 나타낸다. 16바이트 중 사용되지 않는 한 바이트를 암호화 판단 flag로 사용하여 1은 암호화한 데이터로, 0은 암호화되지 않은 일반 데이터로 구분하여 기록하는 것이다. 데이터의 저장에 관한 과정은 (그림4)와 같다.



(그림4) 데이터 저장 순서도

(그림4)와 같이 암호화를 판별하여 데이터를 저장할 시에 데이터의 암복호화를 위해서 키 생성 및 데이터 암호화에 사용된 키에 대한 관리가 필요하게 된다.

기존의 논문[8]에서 암호화 키 관리는 사용자로부터 입력받은 마스터 키로 암호화에 사용된 키를 다시 한 번 암호화하여 낸드 플래시 메모리에 데이터와 같이 저장하는 방식을 사용한다. 이 경우 사용자가 마스터 키를 입력해야 하며 만약 사용자가 마스터 키를 잊어버릴 경우 데이터를 정상적으로 복호화할 수 없다. 또 일정시간이 지나면 다시 마스터 키를 입력해야 하는 번거로움이 있다.

제안 방식에서는 데이터 암호화를 위해 암호화 flag를 판별하여 암호화를 수행할 데이터일 경우 시스템의 시간 정보를 이용하여 생성한다. 암호화 알고리즘으로 대칭 암호화 방식인 DES(Data Encryption Standard)을 사용하고 키의 길이는 64bit이다. 그리고 암호화에 사용된 키 관리는 현재 아주대학교 지식정보보안학과 홈네트워크보안연구실에서 연구하고 있는 방식을 사용한다. 암호화에 쓰인 키를 커널 영역에 공간을 할당하여 저장함으로써 쉽게 접근할 수 없도록 안전하게 저장하는 방식을 사용하여 키를 관리한다[9].

4. 결론

이 논문에서는 낸드 플래시 메모리의 특성으로 인해 삭제된 데이터가 메모리에 남아있게 되는 문제점을 해결하기 위한 방법을 제안하였다. 전용 파일 시스템을 사용하지 않고 FTL 계층에서의 데이터 암복호화를 제안함으로써 전용 파일 시스템보다 유연성을 더했으며 선택적인 암복호화를 통해 오버헤드를 최소로 줄였다. 하지만 아직 암호화한 키를 관리하는 측면에서 더 구체적으로 보강되어야 하며, 섹터 매핑, 블록 매핑, 혼합 매핑과 같은 다른 주소 변환 알고리즘에서 제안 방식을 적용했을 경우의 비용 소모 비교와 같은 실험이 보강되어야 한다.

앞으로 낸드 플래시 메모리는 더 발전할 것이며. 더 많은 분야에서 다양하게 사용될 것이다. 하지만 낸드 플래시 메모리의 특성으로 인해 중요한 데이터가 노출될 수 있다면 보안 측면에서 심각한 위협이 될 수 있다. 따라서 이를 해결하기 위한 더 많은 연구가 요구된다.

참고문헌

[1] 선경문, 최종무, 이동희, 노삼혁 “플래시 메모리 파일 시스템을 위한 안전한 파일 삭제 기법”, 정보과학회논문지: 컴퓨팅의 실제 및 레터 제13권 제6호(2007.11)

[2] P. Gutmann, "Secure deletion of data from magnetic and solid-state memory", in Proceedings of the sixth USENIX Security Symposium, 1996, pp.77-90

[3] K. C. Kung, "Secure file erasure", U.S.Patent 5,265,159, June 1992

[4] Aleph One Ltd, Embedded Debian. Yaffs: A NAND-Flash Filesystem. http://www.aleph1.co.uk/ yaffs/.

[5] Intel Corporation, "Understanding the Flash Translation Layer(FTL) specification", http://www.intel.com, 1998

[6] T. Chung, D. Park, S. Park, D. Lee, S. Lee, and H. Song, "System Software for Flash Memory: A Survey", In Proceedings of the International Conference on Embedded and Ubiquitous Computing, pp.394-404, August 2006

[7] 이재홍, 오진하 외 5명 “플래시 파일 시스템을 위한 안전 삭제기법”, 정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제3호 (2008.5)

[8] 김석현 “플래시 메모리를 위한 암호화 파일 시스템 및 안전 삭제”, 서울대 공학석사학위논문 2008년 2월

[9] 신용명, 김강석, 예홍진 “RAM Core Dump Exploitation을 방지하기 위한 RAM Encryption”, 제 35회 한국정보처리학회 춘계학술발표대회 논문집 제 18권 1호(2011.5)