스마트카드 Atomic operation의 최적화 방안

전은아* 이정엽* 지재덕* 정석원**

*고려대학교 정보보호 대학원
**국립 목포대학교 정보공학부

Optimizing method of smart card atomic operation

Eun-A Jun*, Jung-Youp Lee*, Jae-Deok Ji*, Seok-Won Jung**

*Graduate School of Information Security, Korea University
**Department of Information Security, Mokpo National University

요 약

스마트카드의 EEPROM은 갱신, 삭제가 가능한 프로그램 및 데이터가 저장되는 저장 장치로서, 호스트 환경(PC환경)의 하드디스크와 같은 역할을 한다. 스마트카드의 EEPROM에 데이터를 저장하는 과정은 먼저 EEPROM의 데이터를 지우고, 새로운 데이터를 쓰는 두 단계로 이루어져 있기 때문에 중요 데이터에 대한 무결성을 보장하기 위해서 atomic operation은 하드웨어로서 지원하지 못할 경우 반드시 소프트웨어적으로 지원되어져야한다. 스마트카드 운영체제의 Atomic operation이 수행되는 과정에서 EEPROM의 버퍼 구조의 설계는 스마트카드의 수명과 밀접한 관계가 있으며, 파일에 접근하여 데이터를 처리하는 시간에 대하여 의존도가 매우 높다. 이에 본 논문에서는 스마트카드의 atomic operation 메커니즘에 대하여 알아보고, atomic operations 메커니즘을 지원하는 EEPROM의 Capabilities 증가 구조 제안과 효율적으로 파일의 접근 속도를 최소화하는 구조를 제안 한다.

I. 서론

스마트카드의 EEPROM은 갱신, 삭제가 가능한 프로그램 및 데이터가 저장되는 저장 장치로, 호스트 환경(PC환경)의 하드디스크와 같은 역할을 한다[1].

하지만 EEPROM은 RAM이나 하드디스크와는 달리 읽기 시간과 쓰기 시간의 차이가 크며, 쓰기 횟수의 제한이 있다. 삼성전자의 S3C89V5/C89V8 칩의 경우 EEPROM의 데이터를 읽는 시간은 수 μs이지만, 데이터를 지우거나 쓰는 시간은 4ms이상의 시간이 필요하다[10]. 또한, EEPROM을 쓰는 횟수를 15만회 까지만 보증한다. 따라서 스마트카드의 운영체제(COS) 설계에 있어서 EEPROM 동작의 최적화는 필수적인 요소이다.

스마트카드의 EEPROM에 데이터를 저장하는 과정은 먼저 EEPROM의 데이터를 지우고, 새로운 데이터를 쓰는 두 단계로 이루어져 있기 때문에 중요 데이터에 대한 무결성을 보장하기 위해서 atomic operation은 반드시 지원되어져야한다.

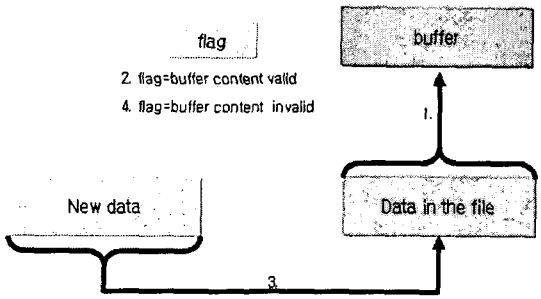
이에 본 논문에서는 먼저 스마트카드의 atomic operation 메커니즘에 대하여 설명하고, 현재 적용되고 있는 atomic operation의 문제점을 살펴본다. 그리고 이러한 문제점을 해결할 수 있는 효율적인 atomic operation을 할 수 있는 최적화 방안에 대하여 제안한다.

II. 스마트카드의 Atomic operation

Atomic operation이란 완전하게 수행되거나 그렇지 않으면 전혀 수행되지 않는 동작을 의미한다[1].

스마트카드 EEPROM에 데이터를 저장하는 과정은 두 단계로 이루어져 있기 때문에 전원 공급이 갑자기 중단되거나 인터페이스로부터

카드가 이탈하였을 경우 데이터가 부분적으로만 기록될 수 있다[1]. 따라서 그림 1과 같은 atomic operation이 수행되는 것이 일반적이다.



step 1) 파일에 있는 기존의 데이터 및 그 주소와 길이를 버퍼로 백업한다.
step 2) 버퍼 플래그는 'data in buffer valid'로 설정한다.
step 3) COS는 새로운 데이터를 파일에 저장한다.
step 4) 버퍼 플래그에 'data in buffer not valid'로 다시 설정한다.
* 운영체제가 시작 될 때 마다 버퍼 플래그를 확인하여 만약, 플래그가 'data in buffer valid'로 설정되어 있을 경우, 버퍼에 있는 데이터를 자동으로 파일로 복원한다.

[그림 2] Atomic operation의 Process Control

III. Atomic operation의 문제점 및 기존의 해결방안

앞 장에서 설명한 일반적인 atomic operation 방법을 적용하면, "자료들이 부분적으로 EEPROM에 저장되어서는 안 된다"는 기본적인 전제를 충족시킬 수 있다. 그러나 이 과정에서 두 가지 위협적인 결함이 존재하게 된다[1].

첫 번째 취약점은 데이터를 적을 때 마다 EEPROM의 특정 영역 즉, 버퍼에 쓰기와 지우기로 매번 수행 되므로 EEPROM 영역 중 버퍼 영역이 제일 먼저 쓰기 및 읽기 횟수에 대한 제한을 받게 된다. 따라서 스마트카드의 전체 수명에 영향을 주게 되는 원인이 된다[1].

두 번째 취약점은 atomic operation이 수행되는 동안, 기존의 데이터를 버퍼에 백업하고 플래그를 갱신하는 과정이 추가 되므로, atomic operation을 적용하지 않고 EEPROM에 직접 데이터를 기록하는 시간 보다 최대 3배 까지 증가하게 된다[1].

3.1 카드 수명 제한에 대한 기존의 해결방안

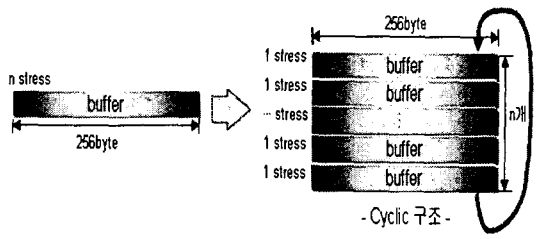
스마트카드의 EEPROM 특정영역에 무리한 읽기와 쓰기로 인한 스마트카드의 수명이 제한되는 점을 보안하기 위하여 현재 제시되고 있는 해결방법은 여러 개의 버퍼를 cyclic 구조로 설계하는 방법이다. 그림 2는 cyclic 버퍼로 설계 하였을 경우의 구조를 보여준다. 그림 2에서 처음으로 들어오는 데이터는 첫 번째 버퍼를 사용하고 다음 들어오는 데이터는 두 번째 버퍼를 사용하는 과정을 순환 반복하고 있기 때문에 메모리의 특정영역에 부담을 주는 것을 완화시켜준다[1].

그러나 이 방법은 버퍼를 위한 메모리 공간이 늘어나기 때문에 상대적으로 사용자 메모리 공간이 줄어든다.

APDU(Application Protocol Data Unit) 명령어로 들어올 수 있는 최대 데이터가 256byte[7]이므로 atomic operation을 위한 버퍼의 크기는 256byte가 되어야 한다.

기존의 방법으로 버퍼를 위한 메모리 공간을 4배만 늘인다고 가정하면 1kbyte의 공간을 필요로 한다. 그러나 저가형으로 사용되는 스마트카드는 보통의 경우 8kbyte의 EEPROM 영역을 가지고 있으므로 1kbyte의 버퍼 메모리 공간은 상당히 큰 부담이 된다.

결과적으로 제안되었던 방법은 버퍼의 특정영역을 반복적으로 사용하지 않는 장점은 있지만, 버퍼 메모리의 확장 방법을 사용하고 있기 때문에 효율적인 메모리 할당이 이루어지지 않는 결함을 갖고 있다.



[그림 3] 스마트카드 수명에 대해 고려한 저장 공간 확장 방법 적용 구조

3.2 접근 시간에 대한 기존의 해결방안

EEPROM에 접근하는 모든 데이터를 버퍼링 하는 것은 스마트카드 전체 동작 시간이 크게 증가될 수 있기 때문에 중요 데이터에 대해서만 atomic operation을 적용하는 절충안이 사용되는 경우가 일반적이다[1].

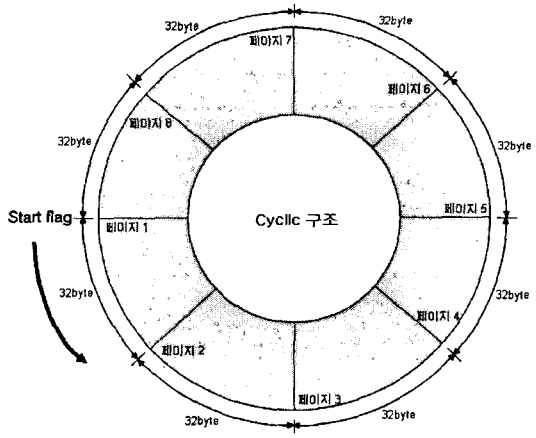
IV. 제안하는 Atomic operation의 최적화 구조

4.1 Atomic operation 메커니즘을 지원하는 EEPROM의 Capabilities 증가 구조 제안

EEPROM은 32byte 또는 64byte를 하나의 페이지 단위로 사용하며 쓰기 또는 지우기는 한 페이지 단위로 수행된다. 한 페이지를 32byte로 가정한다면 앞 장에서 언급한 256byte의 한 개의 버퍼는 8페이지로 구성이 된다.

실제로 입력되는 APDU 데이터의 크기를 분석해 보면 64byte 이하의 데이터가 입력되는 것이 대부분이다. 따라서 하나의 버퍼에서 앞의 1~2페이지만 주로 사용되고 나머지 페이지들은 거의 사용되지 않게 된다.

이에 본 논문에서는 하나의 버퍼를 위해 8개의 페이지를 cyclic 구조로 설계함으로서 버퍼를 위한 공간을 최소화 할 수 있는 방법을 제안한다.



[그림 4] 제안 cyclic 구조 최적화 메모리 할당

예를 들어 그림 3에서 입력되는 APDU 데이터의 크기가 31byte일 경우 기존 파일에 있는 데이터를 버퍼의 페이지 1로 백업하고, 다음에 입력되는 APDU 데이터의 크기가 33byte일 경우 페이지 2와 3으로 백업 한다. 이러한 방법으로 백업을 수행하면 8개의 페이지만을 사용하면서도 모든 페이지를 균등하게 사용할 수 있게 된다.

기존의 해결방안이 버퍼의 수명을 n배 늘이기 위해서 n배의 메모리 공간을 필요로 했다면 본 논문에서 제안한 방법은 추가적인 메모리 공간 없이도 버퍼의 수명을 최대 8배로 증가시킬 수 있다. 따라서 저용량의 EEPROM을 사용하는 스마트카드에 대하여 더욱 효율적으로 사용할 수 있는 구조이다.

4.2 Atomic operation 메커니즘을 지원하는 접근 시간 해결방법 제안

기존의 방법은 스마트카드 전체 동작 시간이 크게 증가될 수 있기 때문에 중요 데이터에 대해서만 atomic operation을 적용하는 절충안이 사용되었다.

그러나 전자화폐에서 사용되는 로그 파일과 같은 중요한 데이터는 반드시 atomic operation을 지원하여야 하므로 동작시간의 증가가 필연적이다. 로그 파일은 일반적으로 cyclic 파일 구조로 설계되며[1], 로그 데이터는 append 명령어로 스마트카드에 저장이 된다. append 명령어의 로그 데이터를 파일에 저장할 때 기존의 atomic operation 방법을 적용하게 되면 atomic operation을 적용하지 않은 방법에 비하여 그 수행시간이 최대 3배가 증가되는 문제가 있다.

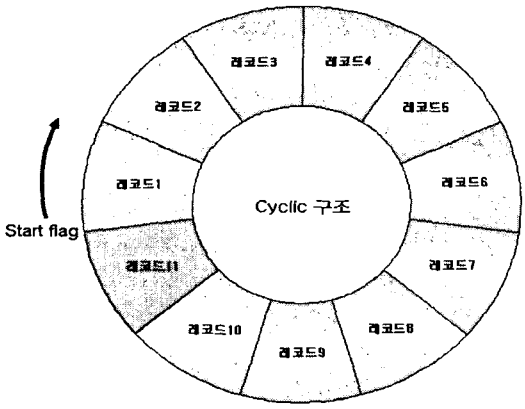
특히, 전원 공급이 불안정한 RF환경에서는 수행 시간의 증가는 전원 공급의 오류가 발생할 확률을 증가시키게 된다.

이에 본 논문에서는 위의 문제점을 해결하기 위하여 접근 시간을 증가시키지 않으면서 atomic operation을 구현할 수 있는 방법을 제안한다.

사용하는 로그 파일을 10개의 레코드를 가진 cyclic 파일 구조라고 가정 한다면, 본 논문에서 제안하는 방법은 그림 4와 같이 버퍼로 사용할 레코드 공간 하나를 더 추가하여 11개의 레코드 공간을 위한 cyclic 파일 구조로 설

계한다. 특정 영역의 메모리 버퍼는 사용하지 않는다.

예를 들어 처음으로 입력되는 로그 데이터는 레코드 1에 저장하고 파일의 헤더에 레코드 1이 시작 레코드임을 갱신 한다. 동일한 방법으로 다음 9개의 로그 데이터를 저장하면, 레코드 1에서 10까지의 공간이 사용되고 있으며 현재의 시작 레코드는 10이다.



[그림 5] 제안 접근 시간 향상 cyclic 구조

11번째 로그 데이터를 저장하는 경우 일반적인 cyclic 파일이라면 레코드 1에 겹쳐 쓰여지지만 제안한 방법은 레코드 11에 로그 데이터를 저장하고 시작 레코드를 11로 갱신한다. 로그 데이터가 완전하게 쓰여진 경우, 레코드 11부터 레코드 2까지를 정상적인 레코드로 인식한다. 그러나 완전하게 쓰여지지 않은 경우, 레코드 11의 데이터는 오류가 발생하게 된다. 따라서 레코드 10부터 레코드 1까지를 정상적인 레코드로 인식하게 된다.

레코드 11의 오류 판단 시간은 메모리 읽기 동작이므로 수행시간이 매우 짧다. 따라서 백업을 위한 EEPROM 쓰기 시간 없이 atomic operation을 구현함으로 기존의 atomic operation을 지원하지 않는 방법에 비해 EEPROM 접근 시간이 추가로 발생 하지 않는다.

결과적으로 atomic operation을 지원하는 방법과 비교하여 볼 때 백업을 하기 위해 버퍼에 접근하는 시간이 줄어들게 되어 속도를 향상 시킬 수 있다.

V. 결론

본 논문에서는 스마트카드 atomic operation을 지원하고 범용으로 사용 가능한 EEPROM 메모리 공간의 효율적 사용을 위한 구조를 제안하였다. 또한 로그파일에서 많이 사용하는 cyclic 구조에서 버퍼의 메모리 공간을 따로 만들지 않고 레코드를 추가하여 명령의 처리 속도를 줄이면서 atomic operation을 보장하는 구조를 제안하였다. 이는 실제 스마트카드 운영체제를 구현하여 적용함에 있어 EEPROM 영역을 효율적으로 사용이 가능하며, 속도 개선에 있어서 매우 효과적인 방법으로 기대된다.

[참고문헌]

[1] W. Rankl, W. Effing Smart Card Handbook, Third Edition, John Wiley & Sons, Ltd, 2004.
[2] ISO/IEC 7810:1995, Identification Cards-Physical Characteristics.
[3] ISO/IEC 7813:1995, Identification cards-Financial transaction cards.
[4] ISO/IEC 7816-1:1998, Identification cards-Integrated circuit(s) cards with contacts-Part 1:Physical characteristics.
[5] ISO/IEC 7816-2:1999(E), Information Technology-Identification Cards-Integrated Circuit(s) Cards with Contacts-Part 2: Dimensions and Location of the Contacts.
[6] ISO/IEC 7816-3:1997, Identification cards-Integrated circuit(s) cards with contacts -Part 3:Electronic signals and transmission protocols.
[7] ISO/IEC 7816-4:1995, Identification cards-Integrated circuit(s) cards with contacts -Part 4: Interindustry commands for interchange.
[8] ISO/IEC 14443-3:2001, Identification cards. Contactless integrated circuit(s) cards. Proximity cards. Part 3: Initialization and anticollision.
[9] ISO/IEC 14443-3:2001, Identification cards. Contactless integrated circuit(s) cards. Proximity cards. Part 4: Transmission protocol.
[10] Samsung Electronics, S3C89V5/C89V8 User's Manual, 2003.
[11] Samsung Electronics, S3CC9RB User's Manual, 2003.
[12] 금융결제원, K-Cash 품질인증 표준규격, Available at http://www.kftc.or.kr