OTP 기반의 USB 디바이스 인증 프로토콜

정윤수* · 김용태** · 박길철***

USB Device Authentication Protocol based on OTP

Yoon-Su Jeong* · Yong-Tae Kim** · Gil-Cheol Park***

본 연구는 지식경제부 지역혁신센터사업인 민군겸용보안공학연구센터 지원으로 수행되었음.

요 약

최근 대용량의 USB 장치가 휴대하기 편리해지면서 USB 장치의 기능성 또한 빠르게 발전하고 있다. 그러나, USB 장치에 저장되어 있는 개인정보는 별도의 사용자 인증 과정없이 악의적인 목적으로 사용되어 개인정보가 노출될 수 있는 문제가 있다. 이 논문에서는 USB에 저장되어 있는 사용자의 개인정보를 추가 인증 정보 없이 안전하게 보호하기 위한 OTP(One-Time Password)기반의 USB 인증 프로토콜을 제안한다. 제안된 OTP 기반의 USB 인증 프로토콜은 일방향 해쉬 함수를 사용하여 단순한 동작을 수행하기 때문에 낮은 계산을 요구할 뿐만 아니라 서로 다른 네트워크에서 USB 장치의 불리적인 접근을 예방하고 사용자의 불필요한 서비스 액세스를 허용하지 않기 때문에 통신 오버헤드 및 서비스 지연이 향상되었다. 실험에서 제안 프로토콜은 패킷 인증 지연시간과 USB 수에 따른 인증서버의 처리량을 단순파일 저장매체(USB driver)와 자체 연산 가능한 매체(USB Token) 등과 비교 평가하였으며, 실험 결과 인증 지연시간에서 평균 12.5% 향상되었고 USB수에 따른 인증서버의 처리량에서는 평균 10.8% 향상된 결과를 얻을 수 있었다.

ABSTRACT

Now a days, as a mass-storage USB becomes comfortable to carry, function of USB is being developed fast. However, there is a problem that the personal information which is stored in USB could be exposed being used with negative purpose without other certification process. This paper suggests OTP(One-Time Password)-based certification protocol of USB to securely protect personal information stored in USB without additional certification information. The proposed OTP based certification protocol of USB not only demands low calculations but also prevents physical approach of USB of other network and does not allow unnecessary service access of user because it conducts simple action and uses one-way hash function. Therefore, communication overhead and service delay is improved. In the experiment, the proposed protocol compares and evaluates throughput of certification server according to the numbers of USB and delay time of packet certification with a device(USB driver) which simply save device and a device(USB Token) which can calculate by oneself. As a result, it is improved as the number of 12.5% in the certification delay time on average and is improved as the number of 10.8% in the throughput of certification server according to the numbers of USB.

키워드

USB, OTP, 디바이스 인증, 프로토콜

Key word

USB, OTP, Device Authentication, Protocol

* 정회원 : 한남대학교 산업기술연구소 전임연구원 (제1저자)
** 정회원 : 한남대학교 멀티미디어학부 교수 (교신저자, ky7762@hnu.ac.kr)
*** 정회원 : 한남대학교 멀티미디어학부 교수

접수일자 : 2011. 03. 09
심사완료일자 : 2011. 06. 12

Ⅰ. 서 론

최근 컴퓨터 저장장치 기술의 발달로 인하여 USB 장치의 대용량화가 가속화 되고 있다. 기업에서는 개인 프라이버시, 기업의 비밀 및 기술정보 유출사고가 빈번히 발생함에 따라 USB 장치에 데이터를 보호하기 위한 보안 기능의 사용량이 증가하고 있다. 특히, 해커의 공격 또는 바이러스로부터 USB 장치를 보호하는 방법은 물리적으로 안전한 영역(TCB: Trusted Computing Base) 에 암호 알고리즘을 수행하는 방법과 사용자 키를 개별적으로 관리하는 방법이 있다[1,2,3].

USB 장치는 크기가 작아 휴대가 간편하며 작은 크기의 문서 파일이나 공인인증서의 저장 등의 역할에만 머무는 것이 아니라 대용량 파일 전송이나 보관, 운영체제의 부팅 디스크 역할, PC 복구 등 다양한 역할을 수행하고 있다[4,5].

USB 장치가 작아지고 대용량의 파일을 빠르게 전송함에 따라 여러 가지 다양한 문제점이 발생되고 있다. 대표적인 문제점으로는 회사의 기밀을 유출하는데 이용하기도 하고, 다수의 기기와 연결되는 특징을 이용하여 악성 코드나 바이러스 등을 유포하는데 이용되기도 한다[6]. 또한 USB 장치내에 저장되어 있는 개인 사용자의 정보가 누출될 수 도 있고 개인적인 문서 및 음악, 동영상 파일 등이 악용될 수도 있다.

USB 장치의 문제점을 해결하기 위해서는 다양한 USB 보안 방법들이 필요하다. 특히, USB 장치를 복제하여 기업이나 주요 공공 기관 내·외부에서 무분별하게 사용할 경우나 사용자에게 입력받은 비밀번호가 무분별하게 사용될 경우 USB 장치내에 저장된 민감한 정보가 유출될 수 있다[5,6]. USB 장치의 이중 사용을 방지하기 위해서는 USB 장치를 관리하는 관리서버와 USB 장치간 올바른 정보가 상호 전달되어야 하고, 관리서버와 USB 장치가 제3자의 불법적인 간섭없이 안전하게 정보가 전달되어 인증 지연 및 인증 서버의 오버헤드가 발생하지 않아야 한다.

이 논문에서는 제3자의 불법적인 간섭없이 사용자의 인증 정보를 안전하게 보호하기 위한 OTP(One-Time Password)기반의 USB 인증 프로토콜을 제안한다. 제안된 OTP 기반의 USB 인증 프로토콜은 일방향 해쉬 함수를 사용하여 단순한 동작을 수행하기 때문에 낮은 계산을 요구할 뿐만 아니라 서로 다른 네트워크에서 USB의 불리적인 접근을 예방하고 사용자의 불필요한 서비스 액세스를 허용하지 않기 때문에 통신 오버헤드 및 서비스 지연이 향상된다.

이 논문의 구성은 다음과 같다. 2장에서는 USB 보안을 하드웨어 보안과 소프트웨어 보안으로 구분하여 USB 보안 취약점에 대해서 분석한다. 3장에서는 USB에 저장되어 있는 사용자의 개인정보를 추가 인증 정보 없이 안전하게 보호하기 위한 OTP 기반의 USB 인증 프로토콜을 제안하고, 4장에서는 제안 기법에 대한 성능평가를 분석한다. 마지막으로 5장에서는 이 논문의 결과를 요약하고 향후 연구에 대한 방향을 제시한다.

Ⅱ. 관련 연구

2.1. USB의 하드웨어 보안

USB의 하드웨어적인 보안 방법은 크게 지문을 이용한 인증방법[7]과 인증 실패시 하드웨어를 파괴하는 방법[8]으로 나뉜다.

지문을 이용하여 인증하는 방법에서는 하드웨어적으로 데이터를 보호하기 위해서 생체 정보 중 지문을 이용하여 사용자의 데이터 접속을 허용하지 않는 방법과 지문과 패스워드를 병행하는 방법이 있다. 또한 관리 서버를 통해 사용자의 ID와 패스워드 등을 등록하여 다수의 사용자가 공동의 USB를 사용하는 방법이 있다.

인증 실패 시 하드웨어를 파괴하는 방법은 일정한 횟수의 인증이 실패했을 경우 USB 내부의 칩셋이나 회로를 파괴하는 방법을 사용한다. 이 방법은 본체에 내장된 암호화 칩을 이용하여 하드웨어적으로 암호화하여 저장하며, 일정 횟수 이상 잘못된 패스워드를 입력했을 경우 암호화 칩셋이 저장된 데이터를 파괴되도록 설계하여 불법적인 사용자가 데이터에 접근할 수 없도록 내부 회로가 파괴되는 방법이다.

2.2. USB의 소프트웨어 보안

USB의 소프트웨어 보안에서는 보안기능을 소프트웨어적으로 지원하기 위해 패스워드 인증방식을 대표

적으로 사용하고 있다. USB의 주요 정보를 소프트웨어적으로 보호하기 위해서 보안과 비 보안 영역으로 나누어 패스워드를 설정한다. 패스워드 인증 전 보안 영역은 USB가 컴퓨터에 연결했을 때 보안 영역이 인식되지 않으며 패스워드 입력 후 인증 과정을 통해 올바른 패스워드가 입력되었을 경우 보안 영역을 마운드 시킨다.

소프트웨어의 다른 보안 방법으로는 패스워드인증 전에는 비보안영역이 인식되고 패스워드 인증 후에는 보안영역이 인식되도록 하는 기법과 지정된 폴더의 파일을 암호화 시키는 방법이 있다.

2.3. USB 보안 취약점

USB 장치는 보통 USB 포트에 접속하여 사용하기 때문에 별도의 사용자 인증 과정이 필요없어 제3자가 악의적인 목적으로 USB 장치를 취득할 경우 큰 피해를 입을 수 있다. 보안 USB의 우회기법으로는 물리적인 방법, 패스워드 전수조사 방법, 지문인증 우회[9], 암호알고리즘 분석 방법, 메모리 덤프를 통한 비밀번호 노출 방법[10] 등이 있다.

물리적인 방법에서는 USB를 구성하고 있는 메모리의 데이터에 제3자가 불법적으로 접근하기 위해서는 메모리 스틱의 플래시 메모리를 분리하여 다른 USB에 연결하는 방법을 사용할 수 있다. 패스워드 전수조사 방법은 인증이 성공할 때까지 모든 가능한 패스워드를 계속 입력하는 방법을 사용한다. 그러나, 이 방법은 현재 USB 장치에서 연속적으로 일정한 횟수의 잘못된 패스워드를 입력할 경우 반 영구적으로 잠기어 포맷을 해야하거나 USB가 자동으로 포맷하는 방법을 사용해야 한다.

지문인증 우회방법은 USB 구성정보가 EEPROM에 저장되기 때문에 읽거나 변경하는 것이 손쉽고 구성요소 변경을 통해 일반 사용자는 다른 사용자의 개인영역 정보를 취득할 수 있다.

암호 알고리즘 분석 방법은 데이터 보호를 위해 사용된 암호 알고리즘을 분석하여 데이터를 획득하는 방법이다. 그러나 AES와 같은 높은 보안성을 가진 알고리즘을 사용하는 보안 USB에는 해당되지 않는 방법이다. 메모리 덤프를 통한 비밀번호 노출방법은 프로세스가 실행중일 때 멤모리 덤프를 실행하여 평문 그대로의 패스워드를 얻는 방법이다.

그러나 이 방법은 컴퓨터가 켜진 상태에서 USB가 연결되어 있을 때에만 패스워드를 획득할 수 있어 제한적으로 사용한다는 단점이 있다.

Ⅲ. OTP 기반의 USB 인증 프로토콜

이 장에서는 USB에 저장되어 있는 사용자의 개인정보를 추가 인증 정보 없이 안전하게 보호하기 위한 OTP 기반의 USB 인증 프로토콜을 제안한다.

3.1. 개요

제안된 OTP기반의 USB 인증 프로토콜에서는 USB 장치내에 저장되어 있는 사용자의 정보를 안전한 채널을 통해 로그인 과정과 검증 과정에서 OTP 운영 정보를 인증서버에게 전달하도록 한다. 제안 프로토콜에서는 서버에 사용자 인증 테이블을 가지고 있지 않기 때문에 USB장치나 서버의 오버헤드가 낮다.

USB 장치는 USB 마스터와 USB 슬레이브가 직접적으로 USB 포트를 사용하여 연결되고 있음을 가정하며 USB 장치는 USB 리더의 통신 범위를 벗어날 경우 일정 시간이 경과한 후 인터럽트가 발생하여 통신을 중지한다.

사용자의 개인키와 USB 장치의 랜덤키는 해쉬 과정에 의해 사용되며, 랜덤키는 연속적인 시간 동기화 문제와 replay 공격을 피하기 위해서 USB 장치가 전에 사용하지 않았던 값을 사용한다. 인증서버는 공개키를 저장하고 있어 MCU(Micro Control Unit)로부터 USB장치의 PIN을 체크하고 해쉬 결과로부터 랜덤 코드를 검색한 후 검색된 코드를 인증 서버가 보유하고 있는 정보와 비교하여 인증이 성공적으로 이루어지지 않으면 인증을 종료한다.

3.2. 용어정의

제안 프로토콜에서 사용하는 주요 용어를 정의하면 표 1과 같다.

표 1. 용어 정의
Table 1. Notaion

용어	정의
N	인증서버에 등록된 USB의 총 수
AS	인증 서버
UI_i	인증서버에 등록된 i번째 USB 정보
ID_X	X의 인식자
PU_X	X의 공개키
PR_X	X의 개인키
K_{AS}	인증 서버의 마스터 키
$Pass$	패스워드
$h(), H()$	충돌 저항 해쉬 함수
$E(X,P)$	정보데이터 P를 X키로 암호화
$D(X,P)$	정보데이터 P를 X키로 복호화
$H_N^i(\bullet)$	인증서버에 i번째 등록된 USB 수로 해싱한 함수
$M_1 \oplus M_2$	M_1과 M_2의 xor 연산

3.3. 등록과정

USB 장치의 등록 과정은 USB 장치가 인증 서버에 등록할 때 동작된다. USB 장치의 사용자 정보는 사전에 안정한 경로를 통해 인증 서버에 저장한다고 가정한다. 인증서버는 (그림 1)과 같은 필드를 USB 장치로부터 전달받아 인증서버의 데이터베이스에 저장한다.

ID	$\exp_i$	$time$	$Check\ info$	$Group$

그림 1. 보안 토큰내 저장된 인증 정보
Fig. 1. Authentication Information saved within Security Token

그림 3에서 각 필드의 세부적인 정보는 다음과 같다.

- ID : 보안 토큰에 저장되어 있는 USB 장치의 신원정보
- $\exp_i$: 인증서버로부터 전달받은 USB 장치 정보의 현재 유효상태
- $time$: USB 장치의 유효시간
- $Check\ info$: USB 장치의 이중 사용 유무정보 (0 or 1)
- $Group$: USB 장치의 그룹 정보

등록과정이 완료되면 USB 장치의 인식자 ID_{USB}, 패스워드 $Pass$, 인증 서버 간 공유될 마스터 키 K_{AS}가 저장된다. 여기서 인식자 ID_{USB}는 USB 장치가 Random() 함수에서 임의로 생성한 인식자이며 마스터 키 K_{AS}는 인증서버에 등록하는 과정에서 발급받은 유효상태 정보 $\exp_i$와 USB 장치의 인식자 ID_{USB}를 XOR하여 생성한다. 인증 서버는 다음 과정을 수행한다.

- 단계 1 : 인증서버는 USB 장치로부터 전달된 인식자 ID_{USB}와 마스터 키 K_{AS}를 $h(ID_{USB}, K_{AS})$로 계산한 후 사용자의 패스워드 $Pass$를 $H_N(\bullet)$에 적용한 $H_N^i(Pass)$를 $h(ID_{USB}, K_{AS})$와 XOR 연산을 수행한다.

$$UI_i = H_N^i(Pass) \oplus h(ID_{USB}, K_{AS}) \quad \text{(식 1)}$$

- 단계 2 : 인증서버는 USB 장치의 메모리에 ID_{USB}와 UI_i을 저장하기 위해 USB 장치에 ID_{USB}, UI_i, i를 전달한다.

- 단계 3 : USB 장치에 ID_{USB}와 UI_i을 전달하고 난 후 인증 서버는 ID_{USB}와 K_{AS}를 $h()$ 함수에 의해 계산된 정보를 쌍으로 하여 인증서버의 데이터베이스에 저장한다.

3.4. 인증과정

인증과정에서는 USB 장치가 하드웨어에 접속할 경우 하드웨어는 USB 장치의 정보를 인증하기 위해서 인증서버에 인증여부를 확인하는 것이 아니라 보안 토큰을 사용해 인증 유·무 정보를 인증서버를 통해 확인한다. 인증서버는 USB의 정보가 남용되는 것을 예방하기 위해서 (그림 2)과 같은 필드를 USB 장치로부터 전달받아 인증서버의 데이터베이스에 저장되어 있는 정보와 비교 후 정보가 일치하지 않을 경우 통신을 중단한다. USB 장치는 다음과 같은 인증과정을 수행한다.

- 단계 1 : USB 장치는 인증을 위해서 i번째 해쉬 결과를 $H_N^i(Pass)$처럼 생성한다.

$$Generate\ \ H_N^i(Pass), i \in integer \quad \text{(식 2)}$$

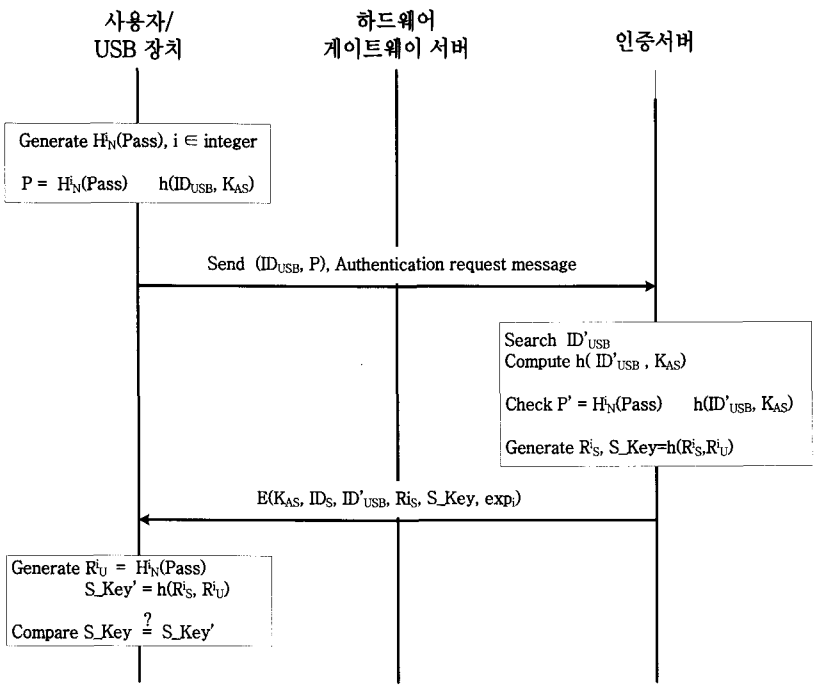

그림 2. USB 인증 과정
Fig 2. USB Authentication Process

• 단계 2 : i번째 인증 결과 값 $H^i_N(Pass)$과 인식자 ID_{USB}와 마스터 키 K_{AS}를 해쉬한 $h(ID_{USB}, K_{AS})$을 XOR한다.

$$P = H^i_N(Pass) \oplus h(ID_{USB}, K_{AS}) \quad \text{(식 3)}$$

• 단계 3 : USB 장치는 (ID_{USB}, P)와 함께 인증 요청 메시지를 인증 서버에게 전달한다.

$$Send\ (ID_{USB}, P),\ Authentication\ request\ message \quad \text{(식 4)}$$

• 단계 4 : 인증서버는 데이터베이스에 저장되어 있는 ID'_{USB}를 확인하고 $h(ID'_{USB}, K_{AS})$을 계산하여 $H^i_N(Pass) \oplus h(ID'_{USB}, K_{AS})$을 체크한다.

$$P' = H^i_N(Pass) \oplus h(ID'_{USB}, K_{AS}) \quad \text{(식 5)}$$

• 단계 5 : 인증서버는 난수값 R^i_S을 생성한 후 $R^i_U = H^i_N(Pass)$와 함께 (식 6)처럼 해쉬 함수에 적용하여 USB 장치와 인증서버 사이의 공유키를 생성한다.

$$Generate\ R^i_S,\ S\_Key = h(R^i_S, R^i_U) \quad \text{(식 6)}$$

• 단계 6 : 인증서버는 USB 장치에게 인증과정에서 생성한 정보와 보안 토큰을 USB 장치에게 전달한다.

$$E(K_{AS}, ID_S, ID'_{USB}, R^i_S, S\_Key, \exp_i) \quad \text{(식 7)}$$

유효상태 정보 $\exp_i$를 사용하는 경우 보안 토큰의 보안키를 갱신하지 않고 서버에 보안 토큰을 요청할 경우에 보안 토큰내에 저장되어 있는 유효상태 정보 $\exp_i$와 사용자의 개인정보 $Check\,info$ 를 체크하여 보안키의 갱신 유・무를 통해 사용자를 인증할 수 있다.

• 단계 7 : USB 장치는 인증서버로부터 전달받은 정보를 검증하기 위해서 등록과정에서 등록한 USB 장치의 패스워드 $Pass$를 $H_N^{'}(\bullet)$함수에 해싱 처리한 R_U^{i}를 생성한 후 $S_Key^{'}(=h(R_S^i, R_U^i))$를 계산한다. 계산된 $S_Key^{'}$와 인증서버로부터 전달받은 S_Key를 비교 한 후 일치하지 않으면 인증을 종료한다.

$$Generate\ R_U^{i'}=H_N^{i}(Pass) \quad \text{(식 8)}$$

$$S\_Key^{'} = h(R_S^i, R_U^{i'})$$

$$Compare\ \ S\_Key \stackrel{?}{=} S\_Key^{'} \quad \text{(식 9)}$$

Ⅳ. 평가

4.1. 보안 평가

4.1.1. Timing 공격

USB 장치와 인증 서버 사이에서 발생 가능한 공격 방법 중 Timing 공격을 예방하기 위해서 제안 프로토콜에서는 인증서버에 등록하는 과정에서 발급받은 유효상태 정보 $\exp_i$와 USB 장치의 인식자 ID_{USB}를 XOR하여 마스터 키 K_{AS}를 생성하기 때문에 시차를 이용한 Timing 공격을 예방할 수 있다.

4.1.2. Replay 공격

USB 장치의 ID를 Random() 함수에 적용하여 ID_{USB}를 생성하고 생성한 ID_{USB}를 이용하여 인증을 수행한다. USB 장치는 인증서버에게 인증정보를 검증받은 후에 유효상태 정보 $\exp_i$ 에 따라 인증과정이 다르게 수행되기 때문에 repay 공격을 예방할 수 있다.

4.1.3. DoS 공격

제안된 USB 프로토콜에서는 유효상태 정보 $\exp_i$와 서명 기법을 함께 사용하여 USB 환경에서 발생할 수 있는 제3자의 악의적인 redirect 및 DoS공격을 방지할 수 있다.

4.1.4. 기밀성

USB 장치와 인증서버 사이에서 사용하고 있는 파라미터들의 기밀성을 보장하기 위해 제안된 USB 프로토콜에서는 S_Key키를 생성하여 마스터 키 K_{AS}로 USB 장치의 정보(ex. ID)를 암호하여 기밀성을 제공한다.

4.2. 성능 평가

4.2.1. 실험환경

이 절에서는 USB 장치와 인증서버 간 간 패킷 인증 지연시간과 처리량을 평가하기 위한 도구로 OPNET을 사용하였다. 실험을 위하여 (표 1)의 실험 시나리오를 사용한다. 실험에서 설정된 USB의 수는 50개이며 하드웨어 게이트웨이 서버가 USB를 동시에 인식할 수 있는 최대 수는 10로 설정한다. 실험 시간은 USB 장치에서 인증서버로 데이터 패킷을 전송한 후 3600초 동안 실험을 수행한다. 버퍼 크기는 100패킷의 크기를 가지는 것으로 가정하며, 각 패킷은 패킷 전송동안 패킷 드롭 확률을 0.01로 한다.

표 2. 실험 환경
Table 2. Experiment Environment

환경 변수	값
USB 수	50
동시 USB 장치 최대 인식수	10
실험시간	3600 s
버퍼 크기	100 packet/s
패킷 드롭 확률	0.01
데이터 패킷 크기	100 bytes
쿼리 패킷 크기	25 bytes
헤더 패킷 크기	25 bytes

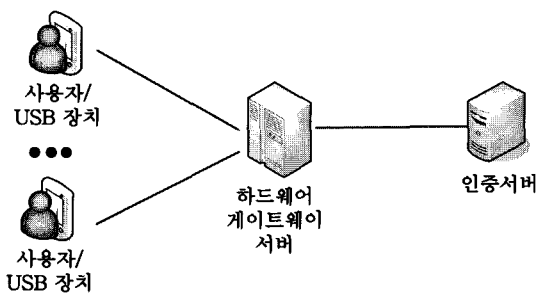

그림 3. 실험 환경 구성
Fig 3. Simulated Scenario

제안된 USB 인증 프로토콜은 (그림 3)과 같다. (그림 3)에서 하드웨어 게이트웨이 서버는 USB 장치를 동시에 인식할 수 있는 수를 50으로 설정한다.

4.2.2. 실험결과

(그림 4)는 하드웨어 게이트웨이 서버가 USB 장치를 최대 50개까지 인식 할 수 있도록 설정 한 후 USB 장치 수에 따른 평균 인증 지연시간을 평가하고 있다. 실험 결과 제안기법은 USB 토큰의 속성정보에 따라 인증과정을 달리하기 때문에 기존 기법보다 평균 13.5%와 11.8% 향상된 결과를 보이고 있다.

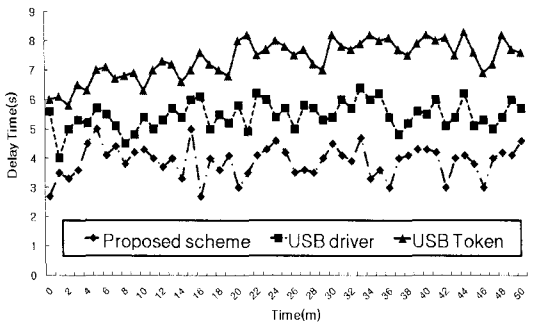

그림 4. 패킷 인증 지연시간
Fig 4. Packet Authentication Delay Time

(그림 5)는 USB 장치 수에 따른 인증서버의 처리량을 보여주고 있다. USB 장치수가 평균 5, 15, 25 일경우 병목현상으로 인해 인증서버의 처리량이 급격하게 늘어나는 현상이 발생하였으며, 제안된 USB 인증 프로토콜에서는 USB 장치가 인증서버에게 인증정보를 검증받은 후에 유효상태 정보 exp_i 에 따라 인증과정이 다르게 수행되기 때문에 USB 장치 수 증가에 따른 처리량이 단순파일 저장매체(USB driver)와 자체 연산 가능한 매체(USB Token)보다 처리량이 일정비율로 증가하고 있다.

(표 3)은 인증 서버에서 생성되는 키의 생성시간을 RSA와 ECC 알고리즘에 따라 비교분석하고 있다. (표 3)의 결과처럼 제안 프로토콜은 USB driver와 USB 토큰보다 RSA와 ECC 알고리즘을 사용할 경우 키 생성 시간이 평균 20% 낮게 나타났다.

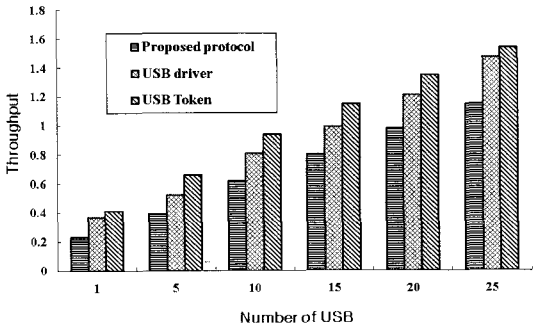

그림 5. USB 수에 따른 처리량
Fig 5. Throughput through USB Number

표 3. 키 생성 시간 비교 분석
Table 3. Perforrmance Compare Analysis of Key Generation Time

단위 : μs

	USB 드라이버	USB 토큰	제안기법
RSA	0.18	0.22	0.15
ECC	0.13	0.19	0.11

V. 결론

본 논문에서는 제3자의 불법적인 간섭없이 USB 장치의 인증 정보를 인증서버에게 안전하게 전달하기 위해서 OTP 기반의 인증 프로토콜을 제안하였다. 제안된 프로토콜은 일방향 해쉬 함수를 사용하여 단순한 동작을 수행하기 때문에 낮은 계산을 요구할 뿐만 아니라 서로 다른 네트워크에서 USB의 불리적인 접근을 예방하고 사용자의 불필요한 서비스 액세스를 허용하지 않기 때문에 통신 오버헤드 및 서비스 지연이 향상되었다. 실험 결과, 제안 프로토콜은 패킷 인증 지연시간과 USB 수에 따른 인증서버의 처리량을 단순파일 저장매체(USB driver)와 자체 연산 가능한 매체(USB Token) 등과 비교 평가하였으며, 실험 결과 인증 지연시간에서 평균 12.5% 향상되었고 USB수에 따른 인증서버의 처리량에서는 평균 10.8% 향상된 결과를 얻을 수 있었다.

향후 연구에서는 제안된 메커니즘을 여러 종류의 USB 환경에 적용할 계획이다.

참고문헌

[1] D. W. Kim, J. W. Han, and K. I. Chung, "Trend of Home Device Authentication/Authorization Technology", Weekly IT BRIEF, No. 1329, pp. 1-11, 2008.

[2] J. H. Kim, J. W. Gi, and C. K. Kim, "A User Authentication Method between Domains Using Privilege Certificates", Korea Institute of Information Security&Cryptology, Journal of KIISC, 18(6A), pp. 75-83, Dec. 2008.

[3] J. S. Moon, D. G. Lee, I. Y. Lee, "Device Authentication/Authorization PRotocol for Home Network in Next Generation Security", Advances in Information Security and Assurance(ISA 2009), LNCS 5576, pp. 760-768, Jun. 2009.

[4] S.Y. Lee, K.B. Yim, K.J. Bae, Taeyoung Jeong, and Jong-Wook Han, "Counterplan of Ubiquitous Home Network Privacy based on Device Authentication and Authorization," Korea Institute of Information Security & Cryptology, Review of KIISC, 18(5), pp.125-131, Oct. 2008.

[5] W.J. Lee, and I.S. Jeon, "Attribute-base Authenticated Key Agreement Protocol over Home Network", Journal of Korea Institute of Information Security & Cryptology (KIISC), 18(5), pp.49-57, Oct. 2008

[6] "Device Certificate Profile for the Home Network", TTAS.KO-12.0052, 2007.

[7] STEALTH MXP FAMILY MXI Security, http://www.mxisecurity.com/

[8] IronKey, https://www.ironkey.com/

[9] P. J. Bakker et al. "Investing Secure USB sticks", Nov. 2007.

[10] S. H. Lee, J. Kwak and I. Y. Lee, "The Study on The Security Solutions of USB Memory", Proceedings of the 4th International Conference Ubiquitous Information Technologies & Applications, 2009(ICUT'09), pp. 1-4, Dec. 2009.

저자소개

정윤수(Yoon-Su Jeong)

1998. 청주대학교 전자계산학과 학사
2000. 충북대학교 대학원 전자계산학과 석사
2008. 충북대학교 대학원 전자계산학과 박사
2009. 9 ~ 현재 한남대 산업기술연구소 전임연구원
※관심분야 : 센서 보안, 암호이론, 정보보호, 네트워크 통신, 이동통신보안

김용태(Yong-Tae Kim)

1984. 한남대학교 계산통계학과 학사.
1988. 숭실대학교 전자계산학과 석사.
2008. 충북대학교 전자계산학과 박사.
2002. 12. ~2006.2 (주)가림정보기술 이사
2010. 8 ~ 현재 한남대학교 멀티미디어 학부 교수
※관심분야 : 모바일 웹서비스, 정보보호, 센서 웹, 모바일 통신보안

박길철(Gil-Cheol Park)

1983. 한남대학교 전자계산학과 학사.
1986. 숭실대학교 전자계산학과 석사.
1998. 성균관대학교 전자계산학과 박사.
2006. UTAS, Australia 교환교수
1998. 8. ~ 현재 한남대학교 멀티미디어학부 교수
2005. 2. 한국정보기술학회 이사 멀티미디어 분과 위원장
※관심분야 : multimedia and mobile communication, network security