情報保護學會論文誌
第20卷 第5號, 2010. 10

Sun 등이 제안한 착용 가능한 토큰 시스템의 취약점 분석에 관한 연구

김 정 윤†, 최 형 기‡
성균관대학교

Weaknesses of the new design of wearable token system proposed by Sun *et al.*

Jung-Yoon Kim†, Hyoung-Kee Choi‡
Sungkyunkwan University

요 약

Sun 등은 노트북이나 PDA와 같은 모바일 장치의 보안을 위해, 착용 가능한 토큰 시스템을 제안하였다. 우리는 본 논문을 통해, Sun 등의 시스템이 오프라인 패스워드 추측 공격, 그리고 기지평문 공격에 기반한 중간자 공격에 취약하다는 것을 보여준다. 우리는 성능저하를 최소화 하는 동시에, Sun 등의 시스템의 보안 문제점을 극복하는 해결책을 제시한다. Sun 등의 시스템과 비교하여, 제안하는 프로토콜에서는 연산 능력이 부족한 토큰의 경우 곱셈 연산이 1회 추가되었으며, 연산 능력이 우수한 노트북, PDA와 같은 모바일 장치에서는 지수승 연산이 1회 추가되었다. 제안하는 프로토콜에서는 Sun 등의 시스템에 존재하는 보안 문제점 뿐 아니라, 알려진 어떠한 보안 문제점도 존재하지 않는다. 즉, 제안하는 시스템은 최소한의 추가적인 오버헤드 만으로, Sun 등의 보안 취약점을 모두 극복하였다.

ABSTRACT

Sun *et al.* proposed a new design of wearable token system for security of mobile devices, such as a notebook and PDA. In this paper, we show that Sun *et al.*'s system is vulnerable to off-line password guessing attack and man in the middle attack based on known plain-text attack. We propose an improved scheme which overcomes the weaknesses of Sun *et al.*'s system. The proposed protocol requires to perform one modular multiplication in the wearable token, which has low computation ability, and modular exponentiation in the mobile devices, which have sufficient computing resources. Our protocol has no security problem, which threatens Sun's system, and known vulnerabilities. That is, the proposed protocol overcomes the security problems of Sun's system with minimal overheads.

Keywords: wearable token, authentication, offline password guessing attack, man-in-the-middle attack

I. 서 론

휴대전화, 랩톱 (laptop) 컴퓨터, PDA와 같은 모바일 장치는 사용자에게 편의를 제공하는 반면, 여러 물리적인 위협에 쉽게 노출될 수 있다. 예를 들어, 모바일 장치의 사용자가 자리를 비운 사이, 공격자가 해당 모바일 장치로부터 민감한 정보를 열람할 수 있다. 또한, 모바일 장치의 휴대성으로 인해 공격자는 모바일 장치를 쉽게 탈취할 수 있고, 이는 더욱 심각한 결과를 초래한다.

Corner와 Noble [1], Nicholson 등 [2], 그리고 Sun 등 [3]은 이와 같은 모바일 장치의 물리적 위협의 주된 원인으로서, 사용자가 모바일 장치로부터 떨어져 있는 동안에도 모바일 장치가 계속해서 동작한

접수일(2009년 4월 21일), 수정일(2009년 10월 31일), 게재확정일(2009년 12월 8일)
† 주저자, steal83@ece.skku.ac.kr
‡ 교신저자, hkchoi@ece.skku.ac.kr

〔표 1〕 Sun 등의 시스템의 요약에서 사용되는 표기 및 변수 정리

ID_m	모바일 장치 m의 아이디
PW	사용자의 비밀번호
K_r	토큰에만 저장되어 있는 루트키 (root key)
K_a	$K_a = h(ID_m \| K_r)$으로 연산 가능하며, K_a는 모바일 장치에 저장되어 있음
K_s	토큰과 모바일 장치 사이에 세션이 설립된 기간에만 공유되는 세션키
N_T, N_D	각각 토큰, 모바일 장치가 생성한 난수
$E_k(x)$	대칭키 k를 이용하여 입력값 x에 대해 대칭키 기반 암호화를 수행
$h(x)$	입력값 x에 대해 일방향 해쉬 함수를 수행
g^x	g와 x를 이용하여 디피-헬만 알고리즘 연산을 수행 (모듈로 표기 생략)
$a \| \| b$	a와 b를 연결 (concatenation)

다는 점을 지적하였다. 이러한 문제를 극복하기 위해, Corner와 Noble은 Zero-Interactive Authentication (ZIA) 이라는 새로운 파일시스템을 제안하였다 〔1〕. ZIA를 사용하는 모바일 장치에는 민감한 정보들이 모두 암호화 되어 저장된다. 그리고 암호화 된 정보를 복호화 할 수 있는 키는, 사용자가 착용하고 있는 토큰으로부터 인증을 받은 후에 획득할 수 있다. 따라서 토큰을 착용하고 있는 사용자가 모바일 장치에 인접해 있을 때에만 모바일 장치에 저장되어 있는 민감한 정보의 복호화 및 열람이 가능하다. Nicholson 등은 ZIA를 더욱 상세하게 구현하고 그 성능을 평가하였다 〔2〕.

Sun 등은 ZIA의 성능을 더욱 향상시키기 위한 모바일 장치와 토큰 간의 상호인증 프로토콜을 새롭게 제안하였다 〔3〕. 그러나 Sun 등의 시스템은 1) 착용 가능한 토큰의 특성과, 2) 사용자들이 지정하는 비밀번호의 취약성으로 인해, 오프라인 패스워드 추측 공격〔4〕〔5〕〔6〕에 취약하다. 뿐만 아니라, Sun 등의 시스템은 기지평문 공격에 기반한 중간자 공격에도 취약하기 때문에, 공격자는 정상적인 모바일 장치로 위장하여 착용 가능한 토큰으로부터 모바일 장치의 사용권한을 획득할 수 있다. 본 논문을 통해, 우리는 여러 가지 공격 시나리오를 제시함으로써 Sun 등의 시스템의 취약점을 제시하고, 이를 극복하기 위한 향상된 스킴을 제안한다. 우리가 제안하는 스킴은 최소한의 성능저하 만으로 Sun 등의 시스템의 알려진 보안 문제점을 모두 해결한다.

II. Sun 등의 시스템의 요약

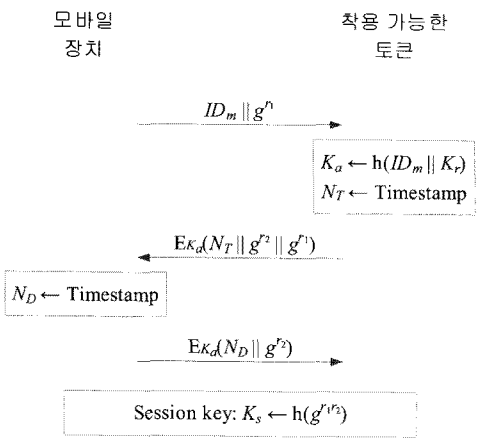
Sun 등의 시스템은 모바일 장치와 착용 가능한 토큰으로 구성되어 있다. 〔표 1〕은 Sun 등의 시스템의 요약을 위해 사용되는 변수 및 함수를 나타낸다.

모바일 장치는 민감한 정보들을 평문으로 저장하지 않고, 파일키 (K_f)를 이용하여 암호화 한 형태로 저장함으로써 민감한 정보들을 보호한다. 그리고 K_f 또한 평문으로 저장되지 않고, 루트키 (K_r)로 암호화 된 형태로 저장된다. 모바일 장치에는 K_r 이 저장되지 않기 때문에, 모바일 장치 만으로는 K_f를 복호화 할 수 없고, 따라서 모바일 장치 만으로는 암호화 된 정보들을 복호화 할 수 없다. 모바일 장치에는 파일키 K_f 대신 비밀키 $K_a = h(ID_m \| \| K_r)$가 저장된다. 단, $h(\cdot)$는 SHA-1과 같은 일방향 해쉬 함수를 의미하고, ID_m은 모바일 장치의 아이디를 의미한다. 모바일 장치에서 민감한 정보를 열람하기 위해서는, 먼저 비밀키 K_a를 이용하여 착용 가능한 토큰과 상호인증을 수행하여 둘 사이의 세션키 K_s를 생성하고, 이후에 토큰으로부터 K_s로 암호화 된 파일키 K_f를 수신한다. 그리고 모바일 장치는 암호화 된 K_f를 K_s로 복호화 하여 K_f를 획득하고, K_f를 이용하여 암호화 된 민감한 정보를 복호화 할 수 있다.

착용 가능한 토큰에는 루트키 K_r을 비밀번호 PW로 XOR한 값인 $K_r \oplus PW$과, 사전에 등록한 모바일 장치들의 각각의 ID가 저장되어 있다. Sun 등은 사용자가 착용 가능한 토큰을 분실할 수 있다고 주장했으며, 분실한 토큰을 습득한 공격자가 토큰을 남용하는 것에 대비하여 최소한 하루에 한 번씩 사용자로부터 비밀번호를 입력 받도록 시스템을 설계하였다. 비밀번호를 입력 받은 후, 토큰은 저장하고 있는 $K_r \oplus PW$에 입력 받은 비밀번호를 XOR 함으로써 K_r을 유도한다. 이후 토큰은 루트키 K_r로부터 비밀키 $K_a = h(ID_m \| \| K_r)$를 계산하고, K_a를 이용하여 모바일 장치와의 상호인증을 수행한다.

사용자가 모바일 장치에 저장되어 있는 암호화 된 정보를 복호화 하고 열람하기 위해서는, 토큰으로부터 파일키 K_f를 수신해야 하며, 이를 위해서는 모바일 장치와 토큰의 상호인증이 먼저 수행되어야 한다. 그리고 토큰은 이 과정에서 설립된 세션키로 파일키 K_f를 암호화 하여 모바일 장치에게 안전하게 전달해야 한다.

〔그림 1〕은 Sun 등의 시스템에서 모바일 장치와 토큰의 상호인증 및 세션키 설립 과정을 나타낸다. 모바일 장치는 랜덤값 r_1을 선택하고, 디피-헬만 알고리



〔그림 1〕 Sun 등의 시스템의 동작 과정

즘[7] 기반의 g^{r_1}를 계산하고, 그 결과를 ID_m과 함께 토큰에게 전달한다. 이를 수신한 토큰은 K_a를 계산한 후, 랜덤값 r_2을 선택하고, 디피-헬만 알고리즘 기반의 g^{r_2}를 계산한다. 그리고 타임스탬프 N_T와 g^{r_2}, g^{r_1}를 K_a로 암호화 하여 모바일 장치에게 전달한다. 모바일 장치는 수신한 메시지를 K_a로 복호화 한 후, 타임스탬프 N_T를 확인하여 재전송 여부를 검사하고, 자신이 저장하고 있는 g^{r_1}과 수신한 g^{r_1}을 비교하여 토큰을 인증한다. 그리고 모바일 장치는 타임스탬프 N_D와 g^{r_2}를 K_a로 암호화 하여 토큰에게 전달한다. 토큰은 수신한 메시지를 K_a로 복호화 한 후, 타임스탬프 N_D를 확인하여 재전송 여부를 검사하고, 자신이 저장하고 있는 g^{r_2}와 수신한 g^{r_2}를 비교하여 모바일 장치를 인증한다. 상호인증이 완료되면 모바일 장치와 토큰은 공통된 세션키 $K_s = \mathrm{h}(g^{r_1 r_2})$를 생성한다.

이후에 토큰은 1초 간격으로 새로운 타임스탬프를 세션키 K_s로 암호화 하여 모바일 장치에게 전달함으로써 연결을 유지한다.

상호인증 이후 모바일 장치는 루트키 K_r로 암호화 된 파일키 $\mathrm{E}K_r(K_f)$를 토큰에게 전달하고, 토큰은 이 메시지를 K_r로 복호화 하여 K_f를 획득한다. 그리고 토큰은 세션키 K_s로 K_f를 암호화 하고, 그 결과값인 $\mathrm{E}K_s(K_f)$를 모바일 장치에게 전달한다. 모바일 장치는 이 메시지를 세션키 K_s로 복호화 하여 파일키 K_f를 획득하고, 이를 이용하여 암호화 된 정보를 열람할 수 있다.

III. Sun 등의 시스템의 취약점 분석

Sun 등은 착용 가능한 토큰은 최소한 하루에 한 번씩 비밀번호를 입력 받도록 되어 있기 때문에, 다른 사용자의 토큰을 획득한 공격자는 토큰을 기껏해야 하루 밖에 사용할 수 없다고 주장하였다. 그러나 공격자는 오프라인 패스워드 추측 공격을 통해 짧은 시간 내에 해당 사용자의 비밀번호를 획득할 수 있으며, 이를 통해 토큰의 지속적인 사용이 가능하다. 또한, 공격자는 기지평문 공격에 기반한 중간자 공격을 통해, 올바른 모바일 장치로 위장하여 토큰으로부터 파일키 K_f를 수신하고, 이를 통해 진짜 모바일 장치에 저장되어 있는 민감한 정보를 열람할 수 있다.

3.1 오프라인 패스워드 추측 공격

오프라인 패스워드 추측 공격[4][5][6]은 사용자의 정상적인 인증 과정에서 발생하는 일부 정보들을 수집한 후, 이를 이용하여 공격자가 오프라인으로 사용자의 비밀번호를 찾는 공격을 의미한다. Sun 등의 시스템에 대한 오프라인 패스워드 추측 공격의 시나리오는 다음과 같다.

먼저, 공격자는 모바일 장치와 착용 가능한 토큰의 정상적인 인증 과정을 도청하여, [그림 1]의 첫 번째 메시지 $ID_m \| g^{r_1}$와, 두 번째 메시지 $\mathrm{E}K_a(N_T \| g^{r_2} \| g^{r_1})$를 수집한다. 이후에 공격자가 해당 토큰을 획득하게 되면, 해당 토큰에 저장되어 있는 $K_r \oplus PW$에 비밀번호 후보 PW'을 XOR하여 $K_r' = K_r \oplus PW \oplus PW'$을 계산한다. 그리고 공격자는 K_r'로부터 $K_a' = \mathrm{h}(ID_m \| K_r')$을 계산하여, 수집한 메시지 $\mathrm{E}K_a(N_T \| g^{r_2} \| g^{r_1})$를 K_a'으로 복호화 한다. 만약 복호화 한 g^{r_1}과 수집한 g^{r_1}이 동일하다면, 공격자는 비밀번호를 발견한 것이고, 그렇지 않다면 다른 비밀번호 후보를 추측하여 위 과정을 반복한다.

일반적으로 사용자들은 강력한 패스워드 보다는 기억하기 쉽고 짧은 패스워드를 사용한다. 뿐만 아니라, 착용 가능한 토큰은 크기가 작기 때문에, 패스워드로 입력할 수 있는 문자의 종류가 제한적이다. 따라서, Sun 등의 시스템의 토큰에서 사용되는 패스워드는 공격자에 의해 노출되기 쉽다.

3.2 중간자 공격

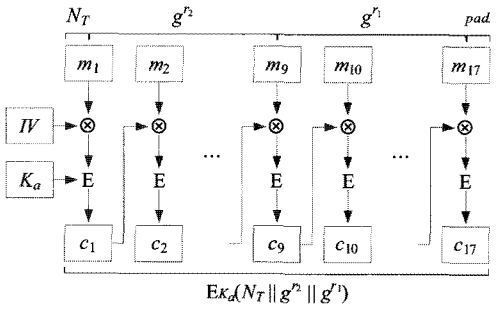
Sun 등의 시스템은 기지평문 공격에 기반한 중간자 공격에 취약하다. [그림 1] 에서 볼 수 있듯이, Sun 등의 시스템에서는 모바일 장치가 착용 가능한 토큰에게 ID_m과 g^{r_1}을 전달하면, 토큰은 타임스탬프

N_T, g^{r_2}, 그리고 g^{r_1}을 모바일 장치와 공유하고 있는 비밀키인 K_a로 암호화 하여 전달한다. 따라서 공격자가 임의의 평문을 선택하여 ID_m과 함께 토큰에게 전달하면, 이를 수신한 토큰은 타임스탬프 N_T, g^{r_2}와 함께 공격자로부터 수신한 평문을 암호화 한 결과를 전달한다. 이러한 사실에 기반하여, Sun 등의 시스템에서는 공격자가 임의의 평문과 그에 대한 암호문을 획득할 수 있는 기지평문 공격이 가능하다. 암호화 키 획득을 목적으로 하는 일반적인 기지평문 공격과 달리, 여기서는 단지 평문 블럭과 그에 대한 암호문 블럭의 획득이 목적이기 때문에, 공격에 소요되는 시간이 더욱 짧다. Sun 등의 시스템에서의 기지평문 공격 시나리오는 다음과 같다.

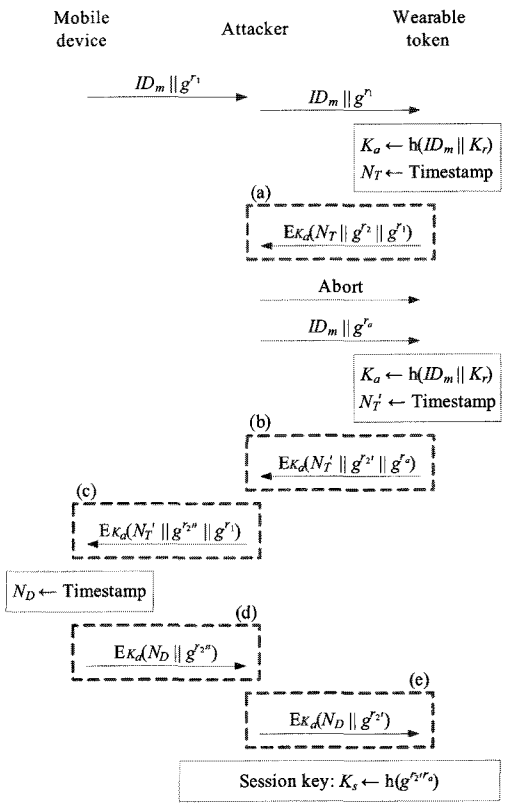
Sun 등의 시스템에서는 암호화 알고리즘으로 AES[8] in CBC mode를 사용하고, g^{r_1} 및 g^{r_2}를 1024비트로 가정하고 있다. AES의 하나의 블럭은 128비트이기 때문에, Sun 등의 시스템에서 N_T || g^{r_2} || g^{r_1}를 암호화 하는 과정은 [그림 2] 와 같다 (단, 타임스탬프 N_T는 64비트로 가정한다). 공격자는 g^{r_1}, *pad*. (단, *pad*.는 패딩을 의미한다), $E_{Ka}(N_T$ || g^{r_2} || $g^{r_1})$을 알기 때문에, [그림 2] 의 m_{10}, m_{11}, ..., m_{17}, c_1, c_2, ..., c_{17}을 알 수 있다. 이제 공격자는 입력값 $m_{10} \oplus c_9$, $m_{11} \oplus c_{10}$, ..., $m_{17} \oplus c_{16}$에 대한 암호화의 결과값이 각각 c_{10}, c_{11}, ..., c_{17} 이라는 사실을 알 수 있다.

공격자는 착용 가능한 토큰에게 임의의 평문을 ID_m과 함께 전송함으로써, K_a를 이용한 암호화의 여러 입력값과 그에 해당하는 결과값을 획득할 수 있다. 이 과정은 모바일 장치와 토큰의 상호인증이 수행되기 이전에 발생하기 때문에, 토큰은 공격 여부를 식별할 수 없다.

공격자는 이러한 기지평문 공격에 기반한 중간자



[그림 2] CBC mode를 이용한 N_T || g^{r_2} || g^{r_1}의 암호화 과정



[그림 3] 기지평문 공격에 기반한 중간자 공격

공격을 통해 모바일 장치로 위장할 수 있다. [그림 3] 은 중간자 공격 과정을 나타낸다. 먼저 모바일 장치가 공격자에게 상호인증 요청을 위해 ID_m과 g^{r_1}을 전송하면, 공격자는 이를 토큰에게 전달한다. 그러면 토큰은 비밀키인 K_a를 계산하고, [그림 3] 의 메시지 (a)와 같이 타임스탬프 N_T, g^{r_2}, g^{r_1}을 생성하여 이를 K_a로 암호화 한 후 모바일 장치로 가장한 공격자에게 전달한다. 이를 수신한 공격자는 토큰과의 상호인증 과정을 처음부터 다시 수행하기 위해 토큰에게 abort 메시지를 송신한다. 그 후 공격자는 랜덤값 r_a를 선택하고 ID_m과 g^{r_a}를 토큰에게 전송한다. 이를 수신한 토큰은 [그림 3] 의 메시지 (b)와 같이 타임스탬프 N_T', $g^{r_2'}$, g^{r_a}을 생성하여 이를 K_a로 암호화 한 후 모바일 장치로 가장한 공격자에게 전달한다.

이제 공격자는 토큰으로 위장하여 모바일 장치에게 자신을 인증하기 위해, [그림 3] 의 메시지 (b)를 변조한다. 즉, 메시지 (b)가 [그림 2] 와 같이 17개의 블럭으로 구성되어 있을 경우, 공격자는 메시지 (b)의 암호문 블럭 c_8, c_9, ..., c_{17}을 메시지 (a)의 암호문

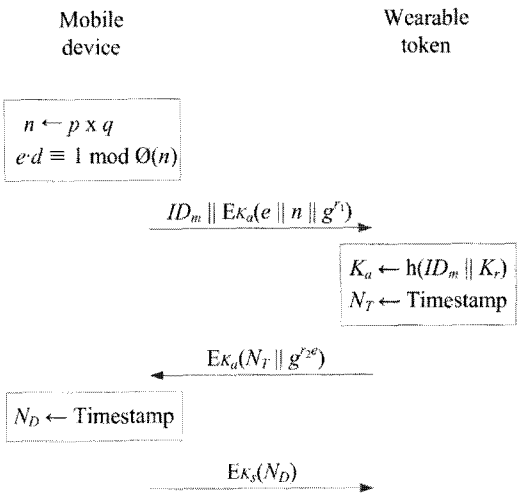
블럭 c_8, c_9, ..., c_{17}으로 대체하면, 메시지 (b)는 메시지 (c)와 같이 $E_{K_a}(N_T' \,||\, g^{r_2}{}'' \,||\, g^{r_1})$로 변조된다. $g^{r_2}{}''$와 $g^{r_2}{}'$은 마지막 블럭 두 개만 다르고 나머지 블럭은 동일하다. 그리고 공격자는 메시지 (c)를 모바일 장치에게 전달한다.

메시지 (c)를 수신한 모바일 장치는 비밀키 K_a로 메시지 (c)를 복호화 하고, 타임스탬프 N_T'을 확인한 후, 저장된 g^{r_1}과 수신한 g^{r_1}을 비교하여 토큰으로 위장한 공격자를 인증한다. 메시지 (c)의 g^{r_1}에 대한 암호문 블럭이 메시지 (b)의 것으로 대체되었기 때문에, 인증은 성공한다.

이후 모바일 장치는 [그림 3] 의 메시지 (d)와 같이 타임스탬프 N_D와 $g^{r_2}{}''$를 K_a로 암호화 하여 토큰으로 위장한 공격자에게 전달한다. 공격자는 토큰에게 모바일 장치로서 인증 받기 위해, 메시지 (d)의 마지막 암호문 블럭을 수정하여 메시지 (d)를 메시지 (e)로 변조해야 한다. 이를 위해 공격자는 기지평문 공격으로 수집해두었던 암호 알고리즘의 입력값과 결과값 쌍을 이용하여, [그림 3] 의 메시지 (b)로부터 $g^{r_2}{}'$을 찾는다. 즉, AES in CBC mode에서 암호문 블럭 c_i에 대한 입력값은 $c_{i-1} \oplus m_i$이고, 이 값에 c_{i-1}을 XOR하면 m_i를 구할 수 있다. 공격자는 기지평문 공격으로 수집해두었던 암호 알고리즘의 입력값과 결과값 쌍을 이용하여, 위와 같은 방식으로 $g^{r_2}{}'$을 찾을 수 있다. 그리고 공격자는 기지평문 공격으로 수집해두었던 값들을 이용하여 [그림 3] 의 메시지 (d)를 메시지 (e)로 변조한다. 메시지 (d)의 $g^{r_2}{}''$과 메시지 (e)의 $g^{r_2}{}'$은 마지막 블럭 두 개만 다르기 때문에, 공격자는 메시지 (d)의 마지막 블럭 두 개만 대체하면 된다.

토큰은 메시지 (e)를 K_a로 복호화 하여 타임스탬프 N_D를 확인하고, 저장된 $g^{r_2}{}'$와 수신한 $g^{r_2}{}'$을 비교하여 모바일 장치로 위장한 공격자를 인증한다. 인증에 성공하면 공격자와 토큰은 세션키 $K_s = h(g^{r_2{}' r_a})$를 공유한다. 공격자는 메시지 (d)를 변조하는 과정에서 $g^{r_2}{}'$을 이미 획득하였기 때문에, $g^{r_2}{}'$에 대해 자신이 생성한 r_a로 모듈로 기반의 지수연산을 수행하고, 그 결과를 해쉬연산 함으로써 K_s를 구할 수 있다.

결과적으로 공격자는 모바일 장치로 위장하여 토큰과 상호인증에 성공할 수 있고, 토큰으로부터 세션키 K_s로 암호화 된 파일키 K_f를 수신하여 모바일 장치에 저장되어 있는 암호화 된 정보들을 복호화 할 수 있다.



[그림 4] 제안하는 스킴의 동작 과정

IV. 제안하는 스킴

우리는 Sun 등의 시스템에서 발생할 수 있는 오프라인 패스워드 추측 공격을 방지하고, 기지평문 공격의 발생 가능성을 차단하기 위한 향상된 스킴을 제안한다. 제안하는 스킴은 기존 Sun 등의 시스템과 유사한 성능을 유지하면서, Sun 등의 시스템의 보안 문제점들을 해결한다.

[그림 4] 는 제안하는 스킴의 동작 과정을 나타낸다. 먼저 모바일 장치는 사전에 소수 p, q를 선택하여 $n = p \times q$를 계산하고, RSA[9] 알고리즘에 기반하여 $e \cdot d \equiv 1 \bmod \emptyset(n)$을 만족하는 키 e와 d를 생성한다. 여기서 $\emptyset$는 Euler's totient function을 나타낸다. 또한, 모바일 장치는 랜덤값 r_1을 선택하고, $g^{r_1} \bmod n$을 계산한 후, ID_m과 함께 $E_{K_a}(e \,||\, n \,||\, g^{r_1} \bmod n)$을 토큰에게 전송한다. 이를 수신한 토큰은 랜덤값 r_2을 선택하고, $r_2 \times e$를 계산한 후, $g^{r_2 e} \bmod n$을 계산한다. 그리고 토큰은 K_a를 계산한 후, 타임스탬프 N_T와 $g^{r_2 e} \bmod n$를 키 K_a로 암호화 한 결과를 모바일 장치에게 전달한다. 모바일 장치는 수신한 메시지를 K_a로 복호화 한 후, 타임스탬프 N_T를 확인하여 재전송 여부를 검사하는 동시에 토큰을 인증한다. 즉, 모바일 장치는 복호화 한 타임스탬프 N_T가 유효한지 확인함으로써 토큰을 인증할 수 있다. 토큰의 인증에 성공하면 모바일 장치는 $g^{r_2 e} \bmod n$에 대해 키 d로 모듈로 기반의 지수연산을 수행하여 $g^{r_2} \bmod n$를 계산한다. 그리고 모바일 장치는 토큰과의 세션키 $K_s = g^{r_2 r_1} \bmod n$를 계산하고, 타임스탬프

N_D를 K_s로 암호화 하여 토큰에게 전달한다. 토큰은 공통된 세션키 $K_s = g^{r_1 r_2} \bmod n$를 생성한 후, 수신한 메시지를 K_s로 복호화 하고, 타임스탬프 N_D를 확인하여 재전송 여부를 검사하는 동시에 모바일 장치를 인증한다. 즉, 토큰은 복호화 한 타임스탬프 N_D가 유효한지 확인함으로써 모바일 장치를 인증할 수 있다. 모바일 장치의 인증에 성공하면 토큰은 모바일 장치와의 통신을 수행한다.

V. 제안하는 스킴의 안전성 분석

본 장에서는 제안하는 스킴의 안전성을 분석한다. 안전성 분석의 대상으로는 오프라인 패스워드 추측 공격, 기지평문 공격, 중간자 공격을 선정하였다. 이 공격들은 Sun 등의 시스템에서 발생가능한 공격들로서, 제안하는 스킴은 이 공격들을 차단하여 Sun 등의 시스템을 개선하기 위해 제안되었다.

5.1 오프라인 패스워드 추측 공격

Sun 등의 시스템에서는 인증에 사용되는 값들 중, 패스워드를 제외한 모든 값을 공격자가 획득할 수 있다. 따라서, 공격자는 패스워드를 미지수로 하는 1차 방정식을 설립할 수 있고, 1차 방정식을 만족하는 미지수를 추측하여 방정식의 성립 여부를 확인함으로써 패스워드를 찾아낼 수 있다. 일반적으로 사용자들은 기억하기 쉬운 취약한 패스워드를 사용하는 경향이 있기 때문에, 공격자가 설립한 1차 방정식의 미지수인 패스워드는 짧은 시간 내에 공격자가 찾아낼 수 있다.

그러나 제안하는 스킴에서는, 공격자가 획득할 수 없는 미지수가 2개 (r_1 혹은 r_2, 패스워드) 이기 때문에, 공격자는 2차 방정식을 설립할 수 밖에 없다. 물론, 2차 방정식이라도 미지수들이 모두 취약한 패스워드만으로 구성되어 있다면, 공격자는 짧은 시간 내에 방정식을 만족하는 미지수 2개의 조합을 쉽게 찾아낼 수 있을 것이다. 그러나 제안하는 스킴의 미지수 중 1개는 난수 (r_1 혹은 r_2) 이기 때문에, 공격자는 이 미지수를 찾기 위해 상당한 시간 동안 전수조사를 수행해야 한다. 이 전수조사는 현실적으로 성공하기 어려울 정도로 많은 시간을 필요로 한다. 따라서, 제안하는 스킴은 오프라인 패스워드 추측 공격으로부터 안전하다.

5.2 기지평문 공격

Sun 등의 시스템에서는 임의의 노출된 평문에 대해 그 암호문이 드러나기 때문에, 공격자는 정상적인 인증 과정 혹은 공격자가 임의로 유발시킨 인증 과정을 통해 다수의 평문 및 암호문 쌍을 수집할 수 있다.

그러나 제안하는 스킴에서는 임의의 노출된 평문에 대해 그 암호문이 드러나지 않기 때문에, 공격자는 평문과 암호문의 쌍을 수집할 수 없다. 따라서, 제안하는 스킴은 Sun 등이 제안한 시스템에 비해 기지평문 공격으로부터 안전하다.

5.3 중간자 공격

제안하는 스킴에서 공격자가 중간자 공격을 수행하여 세션을 가로채기 위해서는 e, n, $g^{r_1} \bmod n$, $g^{r_2 e} \bmod n$ 등을 공격자의 의도대로 수정해야 한다. 그러나, 위에서 언급한 모든 값은 키 K_a로 암호화 되어 송신되며, 키 K_a는 패스워드를 알고 있는 사용자만이 계산할 수 있는 값이기 때문에, 패스워드를 모르는 공격자는 e, n, $g^{r_1} \bmod n$, $g^{r_2 e} \bmod n$를 자신이 의도한 값으로 수정할 수 없다. 따라서, 제안하는 스킴은 중간자 공격으로부터 안전하다.

VI. 제안하는 스킴의 성능 평가 및 분석

제안하는 스킴은 Sun 등의 시스템과 달리, 소수를 선택하는 연산, 키 e, d를 생성하는 연산을 필요로 한다. 그러나, 이 연산들은 모두 인증이 시작되기 이전에 미리 계산이 가능하기 때문에, 성능 분석 및 평가 시 고려하지 않는다. 한편, 제안하는 스킴의 성능 분석 및 평가를 위해, 모바일 장치의 ID 및 각 타임스탬프를 64비트라고 가정하였고, Sun 등의 시스템에서 디피-헬만 알고리즘에 의해 생성된 값과 제안하는 스킴의 n을 각각 1024비트라고 가정하였다. 그리고 제안하는 스킴의 e와 d 또한 각각 1024비트라고 가정하였다.

[표 2] 는 Sun 등의 시스템과 제안하는 프로토콜의 성능 비교 결과를 나타낸다. 연산 오버헤드 측면에서 볼 때, [표 2] 는 토큰과 모바일 장치에서 수행되는 각 연산의 수행 횟수 및 복잡도를 나타내고 있다. H는 해쉬함수의 수행에 소요되는 시간이며, A는 AES 암/복호화에 소요되는 시간, 그리고 M은 모듈로 곱셈 연산을 의미한다. E는 모듈로 지수 연산을 나

〔표 2〕 Sun 등의 시스템과 제안하는 프로토콜의 성능 비교 결과

		연산 오버헤드					통신 오버헤드
		AES 암/복호화 블록 수 (A)	일방향 해쉬 함수 (H≈A)	모듈로 곱셈 (M≈A)	모듈로 지수승 (E)	전체	송신 bits
Sun 등의 시스템	토큰	26	2	0	2	26A + 2H + 2E ≈ 28A + 2E	2112
	모바일 장치 (노트북)	26	1	0	2	26A + 1H + 2E ≈ 27A + 2E	2176
	전체	52	3	0	4	52A + 3H + 4E ≈ 55A + 4E	4288
제안하는 프로토콜	토큰	34	1	1	2	34A + 1H + 1M + 2E ≈ 36A + 2E	1088
	모바일 장치 (노트북)	34	0	0	3	34A + 3E	3200
	전체	68	1	1	5	68A + 1H + 1M + 5E ≈ 70A + 5E	4288

타낸다. H, A, M은 수행시간이 거의 동일하기 때문에, 모두 A로 나타낼 수 있다. 즉, 가벼운 연산인 A와 무거운 연산인 E를 이용하여 Sun 등의 시스템과 제안하는 스킴의 성능을 비교할 수 있다.

Sun 등의 시스템에서 토큰과 모바일 장치는 각각 26개의 평문 블록에 대해 AES 암/복호화를 수행해야 하며, 일방향 해쉬 알고리즘은 토큰의 경우 2회, 모바일 장치의 경우 1회 수행해야 한다. 또한 토큰과 모바일 장치는 모듈로 지수 연산을 각각 2회씩 수행해야 한다. 제안하는 스킴에서 토큰과 모바일 장치는 각각 34개의 평문 블록에 대해 AES 암/복호화를 수행해야 하며, 일방향 해쉬 알고리즘은 토큰의 경우 1회 수행해야 하고, 모바일 장치의 경우 수행하지 않아도 된다. 또한 모듈로 지수 연산은 토큰의 경우 2회, 모바일 장치의 경우 3회 수행해야 한다. 추가적으로, 제안하는 스킴에서는 토큰이 곱셈 연산을 1회 수행해야 한다.

결과적으로 제안하는 스킴과 Sun 등의 시스템의 연산량의 차이를 비교해보면, 제안하는 스킴이 AES 암/복호화를 토큰과 모바일 장치에서 각각 8개의 블록에 대해 추가적으로 수행해야 하며, 일방향 해쉬 알고리즘은 1회씩 더 적게 수행하고, 모듈로 지수 연산의 경우 모바일 장치만 1회 더 수행해야 한다. 그리고 토큰의 곱셈 연산 1회가 추가되었다. 즉, AES 암/복호화 및 곱셈 연산과 같이 빠른 연산은 토큰과 모바일 장치 양쪽 모두에서 그 횟수가 증가하였으나, 모듈로 지수 연산과 같이 느린 연산은 모바일 장치 측에서만 1회 더 증가하였다. 모바일 장치는 토큰에 비해 우수한 성능을 보유하고 있기 때문에, 모바일 장치에서의 모듈로 지수 연산의 1회 증가는 인증 과정의 전체적인 성능에 큰 영향을 미치지 않는다.

통신 오버헤드 측면에서 볼 때, Sun 등의 시스템에서는 토큰이 총 2112비트를 송신하며, 모바일 장치가 총 2176비트를 송신하여 결과적으로 상호인증 과정에서 총 4288비트가 송신된다. 제안하는 스킴에서는 토큰이 총 1088비트를 송신하며, 모바일 장치가 총 3200비트를 송신하여 결과적으로 상호인증 과정에서 총 4288비트가 송신된다. 즉, 제안하는 스킴은 Sun 등의 시스템에 비해 추가적인 통신 오버헤드가 없다.

VII. 결 론

노트북과 같은 모바일 장치의 물리적 보안을 향상시키기 위해, Sun 등은 토큰과 같은 소형 장치를 이용한 시스템에서의 상호인증 프로토콜을 제안하였다. 우리는 Sun 등이 제안한 시스템이 오프라인 패스워드 추측 공격에 취약하며, 공격자는 기지평문 공격에 기반한 중간자 공격을 통해 모바일 장치로 위장할 수 있다는 사실을 공격 시나리오를 통해 제시하였다. 그리고 우리는 이를 극복하기 위한 개선책을 제시하였다. 제안하는 개선책은 기존 Sun 등이 제안한 시스템에 존재하는 보안 취약점을 가지고 있지 않으며, 토큰과 같은 소형 장치에서 충분히 수행할 수 있는 가벼운 연산이 추가되었다.

참 고 문 헌

[1] M. D. Corner and B. D. Noble, "Zero-interaction authentication," in Proc. 8th Int'l Conf. Mobile Computing and Networking, Georgia, pp. 1-11, Sep. 2002.

[2] A. J. Nicholson, M. D. Corner, and B. D. Noble, "Mobile device security using transient authentication," IEEE Trans. Mob. Comput., vol. 5, no. 11, pp. 1489-1502, Nov. 2006.

[3] D. Z. Sun, J. P. Huai, J. Z. Sun, J. W. Zhang, and Z. Y. Feng, "A New Design of Wearable Token System for Mobile Device Security," IEEE Trans. Consumer Electronics, vol.54, no.4, pp.1784-1789, Nov. 2008.

[4] T. Cao and D. Lin, "Cryptanalysis of Two Password Authenticated Key Exchange Protocols Based on RSA," IEEE Communications Letters, vol. 10, no. 8, pp. 623-625, Aug. 2006.

[5] C. C. Yang and R. C. Wang, "Cryptanalysis of Improvement of Password Authenticated Key Exchange Based on RSA for Imbalanced Wireless Networks," IEICE Transactions on Communications, vol. e88-b, no. 11, pp. 4370-4372, Nov. 2005.

[6] W. C. Ku, "Weaknesses and Drawbacks of a Password Authentication Scheme Using Neural Networks for Multiserver Architecture," IEEE Transactions on Neural Networks, vol. 16, no. 4, pp. 1002-1005, Jul. 2005.

[7] W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, vol. IT-22, pp. 644-654, Nov. 1976.

[8] Joan Daemen and Vincent Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Springer-Verlag, 2002.

[9] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, Feb. 1978.

〈著者紹介〉

김 정 윤 (Jung-Yoon Kim) 정회원
2006년 8월: 성균관대학교 컴퓨터공학전공 학사졸업
2008년 2월: 성균관대학교대학원 전자전기컴퓨터공학과 석사졸업
2008년 3월~현재: 성균관대학교대학원 휴대폰학과 박사과정
〈관심분야〉 차량 간 통신 보안, Pay-TV 보안, 무선통신망 보안

1992년 2월: 성균관대학교 전자공학과 학사졸업
1996년 2월: Polytechnic University in Brooklyn, NY 석사졸업
2001년 2월: Georgia Institute of Technology in Atlanta, GA 박사졸업
2001년~2004년: Lancope 근무
2004년 3월~현재: 성균관대학교 정보통신공학부 부교수
〈관심분야〉 네트워크보안, Traffic characterization and modeling