홈 네트워크에서 디바이스를 위한 비밀 정보 보호 기법

맹 영 재[†] · 강 전 일[††] · 아 지 즈[†††] · 양 대 헌[††††] · 이 경 희[†††††]

요 약

홈 네트워크에서 홈 디바이스에 저장되는 비밀 정보는 사용자의 안전에 직접적으로 관련이 있어 매우 신중과 안전을 요하지만, 디바이스에서의 비밀 정보의 보호를 위해서는 추가적인 하드웨어의 지원에 기댈 수밖에 없었다. 하지만 많은 홈 디바이스들이 이러한 준비 없이 사용되고 있으며 이러한 디바이스들에 대한 대비 또한 필요하다고 하겠다. 이 논문에서는 이러한 부분에 있어서, 추가적인 하드웨어의 지원 없이 기존의 홈 디바이스가 가지고 있는 네트워킹 기능을 사용하여 비밀 정보를 보호할 수 있는 두 가지 방법과 그 방법들을 동시에 사용하는 방안에 대해서 제안하고, 제안하는 기법에 대해서 보안적 측면과 비용적 측면에서 각각 살펴본다.

키워드 : 홈 네트워크, 비밀 정보 공유, 코드 인증, 안전 우산

Secret Information Protection Scheme for Device in Home Network

YoungJae Maeng[†] · Jeonil Kang[††] · Abedelaziz Mohaisen[†††] · DaeHun Nyang[††††] · KyungHee Lee[†††††]

ABSTRACT

Even though the secret information stored in home device in home network must be handled very safely and carefully, we have no measure for protecting the secret information without additional hardware support. Since already many home devices without consideration of the security have been used, the security protection method for those devices have to be required. In this paper, we suggest two schemes that protect the security information using networking function without additional hardware support, and those hybrid method to supplement the defects of each scheme. We also consider the our proposals in the aspects of security and cost.

Key Words : Home Network, Secret Sharing, Threshold Cryptography, Code Attestation, Safe Umbrella

1. 서 론

홈 네트워크는 일반 가정에서 네트워크 능력을 가진 여러 디바이스들이 네트워크를 구성하여 내부의 다른 디바이스들과 통신하고, 게이트웨이를 통해 외부와 통신하는 네트워크를 의미한다. 이러한 홈 네트워크에서는 사용자들의 편리한 생활환경을 위해 사용자의 입력이 다른 디바이스들의 동작에도 영향을 주게 된다. 한 편으로, 홈 네트워크의 디바이스들은 사용자의 생활과 직접적으로 맞닿아 있기 때문에 안전상의 이유로 말미암아 비밀 정보 보호에 대해 신중해야할 필요가 있다. 이 중에서 한 가지 중요한 과제가 바로 디바이스가 가진 비밀 정보를 공격자로 부터 안전하게 보호 하는 일이다.

TPM(trusted platform module)[9]과 같은 하드웨어 지원 칩을 이용하면 공격자는 물리적으로 디바이스에서 어떠한 정보를 꺼내기 매우 힘들어진다. 하지만 TPM 칩을 이용하고자 할 경우, 디바이스의 하드웨어뿐만 아니라, 운영 체제(operating system)와 응용 프로그램(application)의 지원 또한 필요하기 때문에 전체적인 구조가 복잡해질 뿐만 아니라, 디바이스를 위한 비용이 증가하게 된다. 또한 현재 많은 홈 디바이스들이 이러한 안전상의 고려 없이 사용되고 있으며, 이러한 디바이스들을 위한 하드웨어의 추가는 기대하기 어렵다. 따라서 소프트웨어적인 방법이 존재하여 디바이스의 비밀 정보를 안전하게 보호 또는 은닉할 수 있다면 이러한 부분을 개선할 수 있을 것이다.

하지만 물리적인 보호 장치가 없다면 공격자는 디바이스에 내장되어 있는 비밀 정보를 쉽게 접근할 수 있다. 비밀 정보가 디바이스의 내부에서 암호화 되어 있다고 하더라도 비밀 정보를 암호화한 비밀 키를 다시 꺼낼 수 있고, 이러한 비밀 키가 외부에 있다고 하더라도 자신이 필요한 비밀 정보를 사용하기 위해서는 디바이스 역시 외부에서 인증 과

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원 사업의 연구결과로 수행되었음 (IITA-2006-C1090-0603-0028)
† 준 회 원 : 인하대학교 정보통신대학원 석사과정
†† 준 회 원 : 인하대학교 정보통신대학원 박사과정
††† 준 회 원 : 인하대학교 정보통신대학원 석사
†††† 정 회 원 : 인하대학교 정보통신대학원 조교수(교신저자)
††††† 정 회 원 : 수원대학교 전자공학과 전임강사
논문접수 : 2007년 3월 28일, 심사완료 : 2007년 7월 19일

정을 거친 후 비밀 키를 가져와야 하는데, 이 인증 과정에서 사용하는 정보가 다시 비밀 정보가 되기 때문이다. 이렇게 복잡하고 서로 상충되는 문제를 해결하기 위해서 이 논문에서는 여러 가지 기존에 익히 알려진 기법과 새롭게 제안하는 기법을 조합하여 보인다.

이 논문에서 비밀 공유(secret sharing) 기법과 코드 인증(code attestation) 기법, 그리고 새로운 형태의 안전 우산(safe umbrella) 기법을 사용하여 소프트웨어적으로 어떻게 비밀 정보를 보호하고 은폐시킬 수 있는지 알아본다. 2장에서는 본 논문에서의 가정 사항에 대해 알아보고 3장에서는 본 논문에서 제시하는 안전 우산기법과 비밀 공유기법 그리고 이 두 기법을 혼합한 기법에 대해서 살펴본다. 4장에서는 제안하는 기법에 대한 평가를 보이고 5장은 결론을 담는다.

2. 소프트웨어적인 방법을 이용한 비밀 정보의 보호

2.1 비밀 정보의 정의

비밀 정보란 자신이외의 다른 개체가 그 정보를 알았을 때 직접적으로 다른 개체가 이득을 얻거나 자신이 피해를 보는 정보를 의미한다. 잠재적으로 발생할지 모르는 이득이나 피해에 대한 부분을 제외한 이유는 잠재적인 이득이나 피해는 그것이 어느 정도인지 가늠하기 어려우며, 또한 그 존재가 무엇인지도 식별하기가 어렵기 때문이다. 예로, 어떠한 개체가 통신을 수행하는 것만으로도 잠재적으로는 다른 개체의 이득이나 자신의 피해를 불러올 수 있다.

따라서 이 논문에서 다루고자하는 비밀 정보는 노출되었을 경우 직접적인 영향이 발생하는 정보들, 예를 들어 공개키 암호화 방식의 개인키(private key)나 비밀키 암호화 방식의 비밀키(secret key), 인증을 위한 생체정보(bio-information)나 인증키(authentication key)와 같은 것을 의미한다. 물론 예를 든 프로토콜의 동작에 필요한 정보 이외에도 프로토콜의 동작과 관련 없이 그 자체만으로도 가치가 존재하는 비밀 정보가 존재할 수 있다.

2.2 소프트웨어적인 방법의 한계

소프트웨어적인 방법의 범위를 정하는 문제는 많은 연구에 있어서 중요하게 다루는 문제는 아니다. 일반적으로 소프트웨어적인 방법을 '이미 기존에 있는 하드웨어에서 소프트웨어의 교체를 통한 방법'이라고 본다면, 좁은 의미에서 데이터를 정적으로 '바뀐 소프트웨어'를 통하여 가공하거나 변형·이용하는 방법이라고 볼 수 있다. 이러한 의미에서 비밀 정보는 메모리에 정적으로 저장되기 때문에 소프트웨어적인 방법으로는 이를 지킬 수 없다. 비밀 정보가 하드웨어의 직접적인 영향을 받지 않기 때문이다.

따라서 이 논문에서는 소프트웨어적인 방법에 대한 의미를 보다 확대하여 소프트웨어가 하드웨어를 제어할 수 있으며, 하드웨어 또한 메모리의 정보에 영향을 줄 수 있는 방법을 사용한다. 소프트웨어는 적극적으로 네트워크를 사용하여 자신의 목적을 이룰 것이다. 또한 비밀 정보는 동적으로 메모리에 저장될 것이며 하드웨어의 영향을 받아 상태가 변한다. 이 과정에서 소프트웨어의 개입은 존재하지 않고, 단지 그러한 상황을 소프트웨어가 만들 것이다. 이러한 가정에서도 홈 네트워크의 디바이스가 갖는 물리적 제약사항을 무시하지 않는다.

2.3 홈 네트워크에서의 가정 사항

기본적으로 홈 네트워크의 디바이스는 통신 기능이 있어 다른 디바이스들과 통신을 수행할 수 있으며, 홈 네트워크에는 전체 홈 네트워크를 책임지고 외부와의 통신을 수행하는 게이트웨이(gateway) 서버와 같은 역할을 수행하는 CP(control point)가 존재한다. 디바이스들은 다른 디바이스와 통신을 수행하기 위해서 CP를 거치지 않아도 한 홉(hop)에서 통신이 가능하다. 이는 기본적인 통신 수단을 무선(wireless)으로 가정하기 때문이다. 즉, 어떠한 두 통신 참가자 사이의 메시지를 중간에 가로채서 변조하는 행위는 매우 어렵다.

공격자는 디바이스에 대한 공격이 성공하기 위해서 최소한 τ 이상의 시간이 필요하다. 공격자는 디바이스를 공격하기 위해서는 홈 네트워크에서 벗어나야 한다. 즉, 물리적인 홈 네트워크의 범위 내에서의 디바이스 공격은 매우 어렵다.

CP는 거의 모든 경우에 있어서 충분히 신뢰 가능하며 안전하지만 아직 알려지지 않은 공격에 대해서는 완벽히 안전하진 않다. 따라서 CP는 디바이스들의 어떠한 비밀 정보도 저장하지 않는다. 이 가정은 CP가 공격당하였다고 해도 그 상태까지는 디바이스들의 비밀 정보가 안전하게 유지되도록 한다.

몇몇 지정된 특수한 메시지를 제외한 모든 통신은 DH 키 교환(diffie-hellman key exchange)[1]를 통하여 얻어낸 세션키(session key)를 가지고 통신을 수행한다.

CP와 디바이스들은 인증서를 가지고 있으며 RSA[2] 기반의 공개키 암호화 방식을 사용한다. p와 q는 큰 소수(prime number)이고 $n=pq$이다. CP의 비밀키는 SK_c, 공개키는 PK_c로 $SK_cPK_c \equiv 1 \bmod \phi(n)$ 관계에 있다.

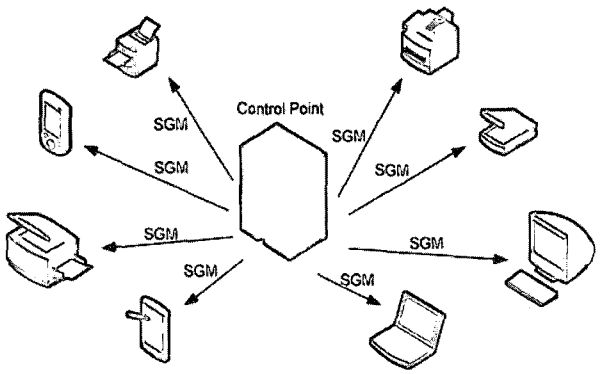
3. 안전 우산(Safe Umbrella) 기법과 비밀 공유 기법

3.1 안전 우산(Safe Umbrella)

3.1.1 프로토콜 개요

이 기법은 비밀 정보를 휘발성 메모리(volatile memory)에 저장하고 디바이스가 처한 상황에 따라 이를 유지하거나 삭제하는 기법이다. 휘발성 메모리의 경우, 데이터의 유지를 위해서 공급되는 전원이 차단되면 그 안의 데이터가 사라져 버리는 특징을 가지고 있다. 디바이스는 자신이 위험한 상황에 처해있다고 판단한다면 자신의 메모리 안의 비밀 정보를 삭제하는 것으로 공격자가 민감하게 반응하는 디바이스로부터 비밀 정보를 얻어내기 힘들게 한다.

하지만 다른 디바이스가 공격당하고 있다고 하더라도, 디바이스는 이러한 사실을 알아내기 쉽지 않다. 디바이스가



(그림 1) 안전 우산 디바이스 비밀 정보 보호 기법 개념도

자신이 처한 상황을 판단할 수 있는 기준은 오로지 자신에게 도달하는 메시지뿐이다. 이러한 목적을 위해서 다른 네트워크 참가자로부터 도달하는 메시지를 사용하는 방법은 두 가지이다. 하나는 공격자의 존재에 대한 보고서(report)로 사용하는 것이고 다른 하나는 공격자가 존재하지 않는 것에 대한 보증서(guarantee)로써 사용하는 것이다. 이 논문에서는 후자의 방법을 사용한다.

이 논문에서는 CP가 충분히 빠른 속도로 공격자가 존재하지 않는 것에 대한 보증 메시지 SGM(safe guarantee message)를 반복적으로 보내는 방법을 제시한다. 디바이스는 올바른 SGM이 도달한다면 자신이 안전하다고 판단하고 그렇지 않다면 디바이스는 자신이 안전하지 않은 상황에 있다고 판단하여 자신의 비밀 정보를 휘발성 메모리에서 삭제한다. 또한 디바이스는 자신의 실행코드에 빈번히 발생할 수 있는 하드웨어 제어의 실패를 제외한 예외(exception) 현상이 발생하였을 경우 비밀 정보를 삭제한다. 따라서 안전 우산 기법을 위해서는 실행코드에서 발생하는 예외에 대한 배려가 필요하며 제3자에 의한 공격에 의해서 발생하는 예외 현상과 내부에서 발생하는 예외 현상을 구별할 수 있는 능력이 필요하다.

3.1.2 서명된 타임스탬프(signed timestamp)

만약 홈 네트워크 안의 CP를 포함한 모든 홈 디바이스들이 시간적으로 동기화 되어 있고 각자 타임스탬프를 생성할 수 있다면 다음과 같은 방법으로 CP가 디바이스의 안전성을 확인시켜줄 수 있다.

단계 1. CP는 자신이 관리하고 있는 모든 홈 디바이스들에게 예상 공격 시간 τ보다 짧은 주기를 가지고 자신의 비밀키 SK_c로 서명된 타임스탬프 t^{SK_c}를 전송한다. 이는 디바이스들로 하여금 이 메시지의 유효성을 판단하도록 한다.

$$CP \rightarrow *: t^{SK_c} \bmod n \qquad (1)$$

단계 2. 모든 홈 디바이스들은 자신이 알고 있는 CP의 공개키 PK_c로 유효 오차 시간 ϵ_t(ϵ_t동안 SGM은 서로 다른 t에 대해서 여러 차례 전송됨을 가정한다)안에 처음 도착한 서명된 타임스탬프로부터 t를 복원하고 자신의 현재 시간 t_{my}와 비교한다.

$$\left|t_{my} - (t^{SK_c})^{PK_c}\right| \le \epsilon_t \ ? \ \text{SAFE} : \text{UNSAFE} \qquad (2)$$

단계 2-0. 만약 복원된 t가 유효한 타임스탬프 형식이 아니라면 이를 무시한다.

단계 2-1. 만약 비교 값이 유효 오차 ϵ_t 이하일 경우 디바이스는 자신이 아직 안전하다고 판단한다.

단계 2-2. 만약 비교 값이 유효 오차 ϵ_t 초과일 경우 디바이스는 자신이 위험하다고 판단하여 비밀 정보를 삭제한다.

단계 3. 만약 일정한 시간 안에 유효한 SGM이 도착하지 않으면 디바이스는 자신의 비밀 정보를 삭제한다.

시간 안에 유효한 SGM이 도착하지 않으면 디바이스는 자신의 비밀 정보를 삭제한다.

3.1.3 프로토콜 평가

공격자는 안전 우산 기법을 사용하는 디바이스로부터 비밀 정보를 얻기 위해서는 반드시 비밀 정보가 메모리에서 삭제되지 않은 상태에서 공격을 수행해야만 한다. 따라서 휘발성 메모리의 비밀 정보가 지워지지 않도록 반드시 전원이 켜져 있는 상태에서 공격이 이루어져야 한다. 여기에 비밀 정보의 삭제 메커니즘이 동작하지 않도록 위 프로토콜 동작부분을 무력화 시키거나 자신이 공격당한다는 사실 자체를 인지하지 못하도록 해야 한다. 무력화 시킬 수 있는 방법은 유효한 SGM을 생성하여 속이는 방법이나 전원이 켜져 있는 상태에서 그 코드 부분만을 실행하지 못하게 하는 것이다. 전자의 경우 CP의 비밀키 SK_c를 알아야하므로 매우 힘들고, 후자의 경우 디바이스의 코드 부분을 고쳐 써넣지 않으면 불가능하다. 디바이스를 공격하여 고쳐 써넣을 수 있다면 가능하지만 공격자는 이러한 공격을 수행하는 동안 일정한 시간 이상이 소요될 뿐만 아니라, 이렇게 고쳐 쓰는 것은 디바이스에서 탐지 가능한 예외 현상이라고 볼 수 있으므로 이는 매우 어려운 일이다. 또한, 이 논문에서 공격자는 비밀 정보를 얻기 위해서만 디바이스를 공격한다고 생각하지만 SGM을 이용하여 서비스 거부(denial of service) 공격을 수행할 수 있다. 하지만 안전 우산 기법은 이러한 공격에 내구성(tolerance)을 제안함으로써 이를 대비하였다.

하지만 이 기법은 비밀 정보를 잃어버리지 않기 위해서 메모리에 전원 공급을 꾸준히 해주어야 하고 SGM도 계속해서 체크해야만 한다. 사용자가 실수로 전원 공급을 차단한다고 하더라도 비밀 정보는 잃어버리게 된다. 이렇게 비밀 정보를 잃어버린 디바이스는 자신을 증명할 수 있는 비밀 정보가 없기 때문에 홈 네트워크에 다시 참여하는 것이 불가능 해진다.

3.2 비밀 공유와 코드 인증(secret sharing and code attestation)

3.2.1 비밀 공유(secret sharing)의 필요성

공격자가 어떠한 디바이스에 저장되어 있는 비밀 정보를 얻어내기 위해서 취할 수 있는 가장 적극적인 공격 방법은 디바이스를 물리적으로 공격하여 디바이스의 메모리에 저장된 모든 내용을 가져오는 것이다. 이 때 디바이스가 비밀 정보를 비휘발성 메모리(non-volatile memory)에 저장하였다면 소프트웨어적인 방법으로는 공격자가 물리적으로 비밀 정보를 취하는 것을 막을 방법이 없다. 하지만 이러한 부분을 막기 위해서 휘발성 메모리에 비밀 정보를 저장하면 이를 관리하는데 매우 조심스러워져야 한다. 따라서 어떠한 경우에 있어서는 비밀 정보에 대한 사본을 비휘발성 메모리에 저장할 필요가 있다. 이러한 딜레마는 비밀을 분산 저장하는 것으로 회피할 수 있다.

비밀 공유(secret sharing)[3]는 여러 참가자가 하나의 비밀 정보를 나누어 가지는 것을 의미한다. 비밀은 랜덤한 상수를 가진 $t-1$차 다항식 $f(x)=s+a_1x+a_2x^2+\cdots+a_{t-1}x^{t-1}$을 만들고 참여자 i에게 $f(i)$를 나누어 줌으로써 분할 될 수 있다. 그리고 이 중에 t개 이상의 부분 비밀(partial secret)을 모았을 때 다항식의 상수를 복원하여 비밀 값 s를 찾을 수 있다. 이는 Lagrange 보간법을 사용하면 된다. 복원 된 비밀 값의 재사용은 매우 위험하므로 비밀 값은 복원하지 않은 상태에서 비밀 값을 사용할 수 있도록 하는 메커니즘이 필요하며, 임계 암호화(threshold cryptography)[4] 기법은 이러한 요구를 만족한다.

그러나 단순히 비밀 공유나 임계 암호화 기법을 홈 네트워크의 디바이스에 사용한다고 해서 모든 문제가 해결되는 것은 아니다. 비밀 공유나 임계 암호화 기법의 경우 디바이스 안의 자신의 존재를 증명할 수 있는 비밀 정보를 삭제해 버린다는 것과 마찬가지이므로 자신을 증명할 수 있는 방법 또한 사라져 버리게 된다. 이러한 부분은 코드 인증(code attestation) 기법을 통해서 해결될 수 있다.

3.2.2 디바이스 인증을 위한 코드 인증(code attestation) 기법

일반적으로 두 참여자 사이에서의 상호 인증에서는 두 참여자밖에 모르는 정보를 이용하거나 그 상대방이 아니면 알 수 없는 정보를 확인하는 방식을 사용한다. 상대방이 무엇을 알고 있느냐(what you know)를 확인하는 방식이다. 그러나 만약 공격자가 물리적으로 모든 메모리 영역에 존재하는 데이터에 접근할 수 있다면 이러한 인증은 무의미해진다. 공격자 또한 공격을 통하여 인증을 위한 정보를 알아낸 다음에 인증을 수행할 수 있기 때문이다. 이러한 점은 비밀 정보가 올바르다고 하더라도 이것이 실제로 그 디바이스인지는 확신할 수 없게 한다.

이렇게 비밀이 노출되었을지도 모르거나 비밀 자체가 가지고 있지 않은 경우 코드 인증 기법을 통해 디바이스를 인증할 수 있다. 코드 인증 기법은 현재 실행되고 있는 코드를 포함하여 메모리상의 모든 코드와 데이터를 확인한다. 따라서 코드 인증을 위해서는 다른 한쪽에서도 상대방과 동일한 코드를 가지고 있어야 한다. 물론 CP가 디바이스를 코드 인증하기 위해서는 CP가 모르는 데이터는 제외해야만 한다.

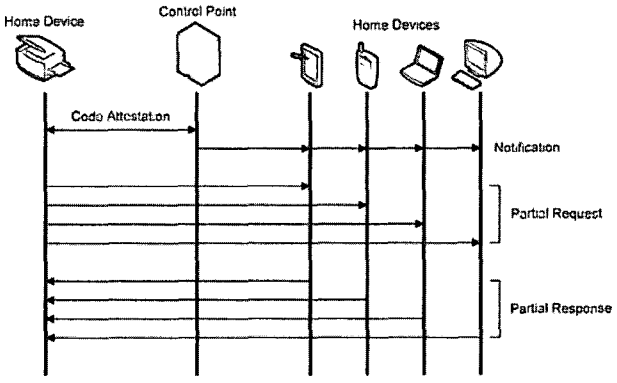
이러한 코드 인증과정을 회피하기 위해서는 디바이스의 메모리와 그 안에 들어 있는 코드를 완전히 알아야 한다. 그러기 위해서 필요한 것은 목표가 되는 디바이스보다 우월한 하드웨어 구조와 능력을 가지고 있어야 하며 거의 완벽하게 에뮬레이션 해야 한다. 디바이스의 코드는 CP와의 협의에 의해서 일부 수정될 수 있다고 보면, 디바이스의 코드를 공격자가 알기 위해서는 디바이스를 공격해서 코드를 직접 획득해야 한다. 공격자가 디바이스를 공격했다는 사실을 감지할 수 있다면 CP는 해당 디바이스와의 코드 인증 자체를 거부할 수도 있다.

한편, 비밀 정보가 들어 있지 않은 디바이스가 공격자에게 공격당해 모든 메모리 영역을 노출 당했다 하더라도 디바이스의 메모리 영역이 공격자에 의해 수정되지 않는 한 이 디바이스는 안전하다고 판단할 수도 있다. 물론, 메모리 영역이 공격자에 의해 수정되었다면 코드 인증은 이를 발견할 것이고 디바이스가 더 이상 유효하지 않다고 판단할 것이다.

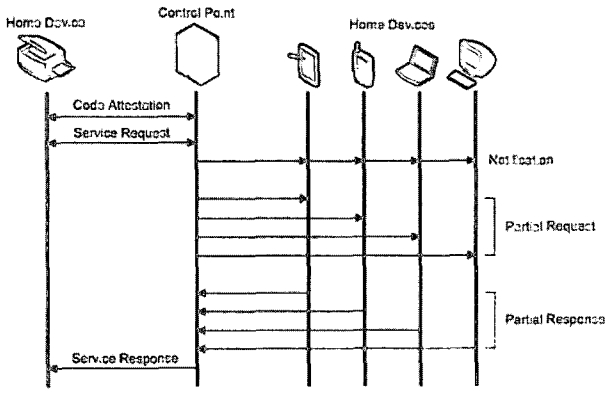
코드 인증은 보통 원격으로 실행 코드를 디바이스에게 전송해주고 이를 실행하게 하거나, 아니면 자체적으로 코드 인증을 위한 코드를 담고 있어 스스로의 메모리를 증명해보일 수도 있다. 이 때, 중요한 것은 메모리에 코드 인증을 회피하고자하는 다른 공격코드를 넣을 가능성을 완전히 배제하거나 다른 공격코드가 넣어져 있다면 이를 찾아내는 방법이다. 유명한 코드 인증 방법으로는 SWATT[5]이나 PIONEER[6], 원격 코드 인증[7]이 있다.

3.2.3 홈 디바이스를 위한 비밀 공유

홈 네트워크에 처음 참여한 디바이스는 CP와 인증을 수행한 후 자신의 인증서를 CP로부터 얻는다. 이 때 사용되는 공개키와 비밀키는 디바이스가 생성할 수도, CP가 생성하여 디바이스에게 줄 수도 있다. 전자는 비밀키를 CP에게 안전하게 전송하고 자신의 메모리에서 삭제해야 하며, 후자는 CP가 비밀키 자체를 디바이스에게 주지 않는다. 만약 디바



(그림 2) 비밀 정보의 완전 복원



(그림 3) 비밀 정보를 이용한 결과의 획득

이스가 추가적인 비밀 정보를 가지고 있다면 이를 CP에게 전송한 후 메모리에서 삭제한다.

디바이스의 비밀 정보를 가진 CP는 랜덤한 상수를 가진 $t-1$차 다항식 $f(x)$를 생성하고 $f(i)$를 각각의 다른 디바이스들에게 나누어준다. 여기서의 차수 t와 $k(n$개의 디바이스 중 비밀을 복구하기 위한 k개의 디바이스)는 홈 네트워크의 환경과 요구되는 보안 수준에 따라 정해진다. 모든 정보를 나누어준 CP는 모든 부분 비밀 정보나 비밀 정보를 자신의 메모리에서 완전히 삭제해야한다.

비밀 정보를 완전히 복원하기 위해서는 디바이스가 자신의 부분 비밀 정보를 가지고 있는 디바이스들에게 직접 부분 비밀 정보를 요청해야한다. 이때, CP는 디바이스에 대한 코드 인증 결과를 다른 디바이스들에게 알려줘야 한다. 이러한 알림 메시지(notification message)는 CP의 비밀키로 암호화 되어 있어 다른 디바이스들은 CP의 공개키로 이를 확인할 수 있으며, 이에 대한 유효 시간이 정해져 있다. 부분 비밀 정보 요청에 대한 응답은 각각의 세션키로 암호화 되어 있거나, 요청하는 홈 디바이스의 공개키로 암호화 되어 있어야 하지만 디바이스가 자신의 비밀키를 바로 아는 것은 불가능하다고 했으므로 후자의 방법은 불가능하다.

임계 암호화 기법을 사용하는 것과 같이 부분 비밀 정보를 이용하여 어떠한 일을 수행한 후 수행된 결과들을 모아 원래 비밀 정보를 이용한 것과 동일한 결과를 얻을 수 있다면 홈 디바이스는 CP에게 코드 인증 후에 서비스 요청(service request)을 이용하여 CP에게 자신이 서비스 받고 싶다는 의사를 밝히고 이에 대한 서비스를 받을 수 있다. CP는 이러한 서비스 요청에 대해서 홈 디바이스를 대신하여 각각의 다른 디바이스들에게 부분 요청을 보내고, 그에 대한 응답들을 모아 합친 후 서비스 응답(service response)로 서비스를 요청했던 디바이스에게 필요한 서비스를 제공한다.

3.2.4 프로토콜 평가

홈 디바이스의 비밀 정보를 얻어내기 위해서 어떤 공격자는 마치 그 홈 디바이스인 것처럼 행동하여 비밀 정보를 복원하려고 시도할 것이다. 그러기 위해서는 코드 인증을 수행해야 하지만 코드 인증의 경우 동일한 메모리 구조를 가진 디바이스의 경우 이를 성공하는 것은 불가능하다. 동일한 메모리를 가지고 있다는 것은 공격코드를 자신이 가지고 있을 수 없다는 것과 같고, 이는 다시 그 디바이스가 공격을 하지 않는다는 것을 의미하기 때문이다. 즉, 공격코드를 가지고 정상적인 디바이스인 척 할 수는 없다.

안전 우산 기법과 다르게 코드 인증을 통해서 인증을 수행하기 때문에 네트워크를 떠났던 디바이스라고 할지라도 홈 네트워크에 다시 참여할 수 있다. 하지만 비밀키 또한 다른 디바이스들에게 분산되어 저장되는 점은 여러모로 통신비용을 증가시키는 결과를 가져온다. 비밀키의 경우 누군가 자신에게 보내는 메시지를 확인하는 등 자신이 보낸 메시지라는 것을 증명하기 위해서 빈번히 사용될 것이기 때문에 이를 분산 시켜놓으면 비밀키가 필요할 때마다 CP에게 서비스를 요청해야만 하는 것과 같다.

3.3 혼합 기법

3.3.1 안전 우산 기법과 비밀 공유 기법의 장·단점

안전 우산 기법과 비밀 공유 기법은 각각의 좋은 점이 존재하나 동시에 여러 단점이 존재하게 된다. 안전 우산 기법은 비밀키를 자유롭게 사용할 수 있으나, 디바이스의 안정성에 영향을 받고 네트워크로의 재참가나 지워버렸던 비밀 정보의 복원이 불가능하다. 반면 비밀 공유 기법은 코드 인증을 통하여 네트워크에 재참가할 수 있고 비밀 정보를 복원할 수 있으나 비밀키를 사용하기 위해서는 CP의 도움이 필요하다.

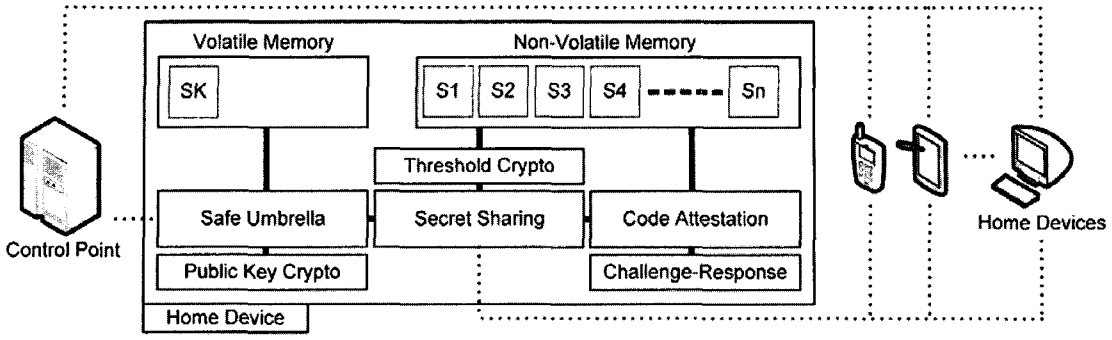
다행스럽게도 안전 우산과 비밀 공유, 이 두 가지 기법을 조합하여 서로의 장점만을 취할 수 있도록 하는 것이 가능하다. 비밀 정보를 저장하는 위치가 서로 모순되지 않아서 비밀 정보가 서로의 기법에 대해서 영향을 받지 않기 때문이다.

〈표 1〉 안전 우산 기법과 비밀 공유 기법의 비교

	안전 우산	비밀 공유
디바이스 인증	디바이스의 비밀키	디바이스의 코드
비밀 정보 저장 위치	자신의 휘발성 메모리	다른 디바이스의 비휘발성 메모리
네트워크의 재참가	불가능	가능
비밀 정보의 복원	불가능	가능
비밀키의 사용	직접적	간접적

3.3.2 안전 우산 기법과 비밀 공유 기법의 혼합

하나의 홈 디바이스 안에서 비밀 정보를 관리하는 부분에 있어서 휘발성 메모리와 비휘발성 메모리에 중복된 데이터가 존재하지 않고 각각의 기법이 다루고자하는 데이터가 서로 다르다는 것은 안전 우산 기법과 비밀 공유 기법을 하나의 디바이스에서 동시에 사용할 수 있다는 것을 의미한다. 안전 우산 기법은 비휘발성 메모리의 데이터에 영향을 주지도 받지도 않고 대신 휘발성 메모리에 자신의 비밀 정보만을 저장·관리 하고 있다. 비밀 공유 기법은 휘발성 메모리의



(그림 4) 비밀 정보 은폐를 위한 홈 디바이스의 내부 구조

데이터에 영향을 주지 않으며 비휘발성 메모리에 다른 디바이스의 부분 비밀을 저장하고 있다.

디바이스가 네트워크에서 벗어나거나 전원을 차단하여 휘발성 메모리에 저장되어 있는 자신의 비밀 정보를 모조리 삭제하였더라도 디바이스의 인증을 위해서 디바이스의 코드를 사용하면 네트워크에 재 참여할 수 있다. 이렇게 네트워크에 다시 참여한 디바이스는 자신의 비밀 정보를 다른 디바이스들로부터 복원할 수 있으며 다시 네트워크를 벗어나거나 전원이 꺼질 때까지 CP에게 별도의 서비스를 요청하지 않아도 자유로이 비밀키를 사용할 수 있다. 아래에서는 이러한 혼합 기법을 설명한다.

단계 1. 홈 네트워크에 참여를 원하는 홈 디바이스는 CP에 코드 인증을 요구한다.

단계 2. 코드 인증을 통해 홈 디바이스가 인증되었다면 그림2와 같은 과정을 통해 비밀 정보를 복구하고 복구된 비밀 정보를 휘발성 메모리에 저장한다.

단계 3. 홈 디바이스가 인증된 후에는 3.1절의 (2)에서와 같이 CP가 모든 디바이스에 전송(broadcast)하는 SGM을 수신한다.

단계 3-0. 수신된 SGM과 디바이스의 타임스탬프의 비교 값이 유효 오차 ϵ_t 이하일 경우 디바이스는 자신이 안전하다고 판단하고 비밀 정보를 유지한다.

단계 3-1. 만약 비교 값이 유효 오차 ϵ_t 초과일 경우 또는 디바이스는 자신이 위험하다고 판단하여 휘발성 메모리에 저장되어 있는 비밀 정보를 삭제한다.

단계 3-2. 만약 유효 오차 ϵ_t안에 유효한 SGM이 도착하지 않으면 디바이스는 자신의 비밀 정보를 삭제한다.

단계 4. 자신이 안전하지 않다고 판단하여 비밀 정보를 삭제한 홈 디바이스는 미리 정의된 네트워크 재참여 시간만큼 기다린 후 단계 1로 돌아가 네트워크 재참여를 시도한다.

4. 홈 디바이스를 위한 비밀 정보 은폐 기법의 평가

4.1 기능적 측면

어떠한 디바이스에 저장되어 있는 정보를 물리적인 공격으로부터 완벽히 지키는 것은 하드웨어의 도움 없이는 매우 힘든 일이다. 어떠한 방법을 사용하여 아무리 복잡하게 비밀 정보를 감춰둔다고 하더라도 그 정보를 접할 방법이 있다면 그 방법을 실행하지 못하도록 하는 것은 소프트웨어적으로 막을 수 없다. 비밀 정보를 꺼내는 방법이 존재하지 않는다면, 그 비밀 정보는 아무런 의미가 없어지므로 결국 어떠한 방법을 사용한다고 하더라도 하나의 디바이스 안에서 자신은 사용할 수 있게 하고 공격자는 모르게 한다는 것은 불가능하다.

이 논문에서 제안하는 기법은 기존의 소프트웨어적인 방법의 범주를 넓혀 사용하고 있다. 네트워킹이라는 하드웨어를 필요로 하는 기술을 추가적으로 사용하는 것이 그것이다. 그러나 홈 네트워크에서의 디바이스는 반드시 네트워킹 기능이 있기 때문에 이러한 추가적인 방법이 홈 디바이스들의 물리적인 능력을 무시한 것은 아니다. 물론, 그렇기 때문에 제안하는 기법을 다른 환경에서 사용할 수는 없을 것이다.

4.2 보안적 측면

4.2.1 코드 인증 회피를 통한 비밀 정보의 복원

공격자에게 노출되는 순간 비밀 정보는 더 이상 비밀이 아니게 된다. 노출된 비밀 정보가 의미가 있는지는 다른 논의겠지만 여기에서는 휘발성 메모리를 사용하여 비밀 정보를 삭제하도록 하였다. 삭제된 비밀 정보는 비밀 공유 기법을 사용하여 복원할 수 있지만 복원하기 위해서는 디바이스의 코드 인증 과정을 거쳐야만 한다. 코드 인증 기법은 우월한 하드웨어를 가진 공격자에 대해서 안전하지 않지만 이러한 하드웨어를 물리적으로 홈 네트워크 안으로 가져와서 CP와 통신을 해야 한다는 점에서 CP를 속이고 특정 디바이스인 척하는 것은 어려울 것이다.

비밀 정보가 들어있지 않은 디바이스의 공격이 성공하여 이 디바이스를 이용하여 CP로부터 코드 인증을 받고 자신이 원하는 비밀 정보를 복원하여 취득하기 위해서는 해당 디바이스에 이러한 일을 위한 공격 코드를 집어넣어야만 한다. 그러나 이러한 공격 코드는 코드 인증 시에 검출될 것이므로 코드 인증은 실패할 것이고 이러한 공격은 성립하기 힘들다. 공격자는 코드 인증을 회피하는 공격 코드를 사용할 수도 있지만 이러한 문제는 코드 인증 기법들에서 주요

하게 다루는 문제이기 때문에 코드 인증 자체를 회피하는 공격은 성공하기 힘들 것이다.

4.2.2 디바이스의 전원이 켜져 있는 상태에서의 공격

만약 공격자가 전원이 켜져 있는 디바이스를 공격하여 전원을 유지한 채로 공격을 성공할 수 있다면 공격자는 디바이스의 비밀 정보를 손에 넣을 수도 있다. 비밀 정보를 손에 넣기 위해서는 네트워킹과 안전 우산 기법이 원활하게 동작하도록 보장할 필요가 있다. 그렇지 않다면 순간적으로 실행 코드의 주도권을 빼앗아 안전 우산 기법을 무력화 시켜야 한다. 또한 이러한 공격을 수행하기 위해서 공격자는 물리적인 홈 네트워크 안에서 공격을 수행해야 한다. 보안성이 공격에 드는 비용과 얻는 이득에서의 차에 기반 한다고 보면, 이러한 위험을 부담해야할 만한 비밀 정보는 홈 디바이스에 저장해서는 안 될 것이다.

4.2.3 콘트롤 포인트(Control Point)

이 논문의 가정에 따라 CP를 공격하여 성공하는 게 매우 힘들다고 해도 공격자가 CP를 자신의 제어 아래에 놓을 수 있다면, 공격자는 자신이 원하는 디바이스의 비밀 정보를 손에 넣을 수 있다. 그러나 비밀 정보는 CP가 자체적으로 저장하고 있지 않고, 디바이스는 다른 디바이스와 직접 통신을 통해 비밀 정보를 모으기 때문에 공격자는 해당 디바이스인 척 다른 디바이스들을 속일 필요가 있다.

4.2.4 다수의 디바이스

공격자는 다수의 디바이스를 공격하여 디바이스들이 보관하고 있는 부분 비밀을 모아 비밀을 복원하려 할 수 있다. 비밀 공유는 하나의 대상에 몰리는 공격을 여러 대상에 분산시키는 것을 목적으로 하므로 이는 합리적인 공격 수단이 되지 않는다. 이는 비용과 관련된 문제로써, 높은 보안성을 위해서는 보다 많은 디바이스를 공격자가 공격하게 하는 것이 바람직하다.

4.3 비용적인 측면

4.3.1 SGM 메시지의 유효성 판단 비용

RSA는 아주 많은 연산을 필요로 한다. 그러나 일반적으로 RSA를 사용하는 많은 응용 분야에서 공개키는 충분히 작은 값(e.g. $e=3$)을 선택하여, 암호화할 때 더 빠른 성능을 갖도록 한다. 서명을 확인하는 작업은 비밀키를 사용하여 복호화한 평문 메시지를 공개키를 사용하여 암호화하는 것과 같으므로 이러한 이점을 취할 수 있다. 실제로 RSA연산에 소요되는 시간과 메모리 등을 측정한 연구 [8]에서는 $e=2^{16}+1$인 1024비트 RSA의 공개키 알고리즘이 8Mhz로 동작하는 ATmega128에서 0.43초가 소요되는 것을 보였으며 현재 또는 미래의 홈 디바이스가 요구되는 홈 네트워크 서비스를 처리하고 주변 기기들과의 통신이 가능하다면 적어도 단순히 센싱된 정보를 전달하는 목적의 센서노드인 ATmega128이상의 연산능력을 가진다고 가정하고 작은 값의 공개키 사용(e.g. $e=3$)을 고려하면 서명된 메시지를 확인하는데 소요되는 시간은 더욱 단축될 것이다. 따라서 많은 연산 능력을 가진 CP와 달리 적은 연산 능력을 가진 디바이스라고 하더라도, 디바이스는 안전 우산 기법을 사용하여 CP의 비밀키로 서명된 타임스탬프나 난수의 유효성을 판단하기 위해서 비교적 적은 비용을 들여 이를 수행할 수 있다.

4.3.2 임계 암호화 기법과 비밀 복원 비용

임계 암호화 기법의 사용은 많은 지수 연산을 필요로 한다. 그러나 다행스럽게도 임계 암호화 기법은 CP에게 서비스를 요청하는 방식으로 이루어지므로 디바이스의 입장에서는 서비스를 요청하고 받는 것 이외에 추가적인 어떠한 비용도 존재하지 않는다.

이 논문에서 제시하는 비밀 복원은 오로지 비밀키를 위해서만 사용되지만 이를 위해서는 Lagrange 보간법을 사용해야 하는데, 이러한 비용은 비밀이 분산되어 있는 수에 비례하여 증가하게 된다. 분산이 많이 되었다면 보다 높은 보안성을 확보할 수 있지만, 비밀 복원을 위한 비용이 증가하게 된다. 반대로 분산이 적게 되어 있다면 비밀 복원을 위한 비용이 감소하지만 높은 보안성은 확보할 수 없을 것이다.

4.3.3 코드 인증 비용

일반적으로 코드 인증을 수행하기 위해서는 수많은 해시(hash)나 체크섬(check sum) 연산을 수행해야만 한다. 이 비용이 예상 외로 크다고 할지라도 외부의 공격이 없는 상황에서는 오로지 전원 관리 문제에 따라서 발생하는 것이기 때문에 코드 인증은 빈번히 일어나는 일이 아니므로 네트워크 전체에 부담이 되지 않을 것이다.

5. 결 론

이 논문에서는 홈 네트워크에서 디바이스 내에 저장되는 비밀 정보를 TPM과 같은 하드웨어적 장치에 기반을 두지 않고, 물리적인 공격자로부터 안전하게 보호하는 두 가지 기법에 대해서 제안하였다. 정적으로 데이터를 저장해서 사용하는 소프트웨어적인 방법을 사용해서는 안전하게 비밀 정보를 지킬 수 없기 때문에, 추가적으로 네트워킹 기능과 동적으로 데이터를 관리 저장하는 방법이 사용되었다. 비밀 정보가 존재하지 않는 디바이스의 인증을 위해서 사용하는 코드 인증 위에, 휘발성 메모리의 비밀 정보 관리는 위한 안전 우산 기법, 비휘발성 메모리의 비밀 정보 관리는 비밀 공유 기법을 적용하였고 이를 통해 서로의 단점을 보완하는 기법을 제시하였다.

여러 보안적인 측면과 비용적인 측면에서의 고려사항에 미루어 보아 제안하는 기법은 실제 홈 네트워크에서 사용가능할 것이다. 그러나 높은 보안성을 위해서는 CP의 보안성이 중요하며, 코드 인증 기법 또한 현재의 하드웨어적 한계성을 극복하는 기법이 연구되어야 할 것으로 보인다.

참 고 문 헌

[1] Whitfield Diffie, Martin Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, IT-22(6), pp. 644-654, November 1976.

[2] Ron Rivest, Adi Shamir and Len Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, Vol. 21 (2), pp. 120 - 126. 1978.

[3] Adi Shamir, "How to share the secret," Communications of the ACM, Vol 22, Issue 11 (November 1979), pp. 612-613, 1979.

[4] Y. Desmedt, Y. Frankel, "Threshold cryptosystems," Advances in Cryptology - Crypto '89, LNCS 435, pp. 307-315, 1990.

[5] Arvind Seshadri, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla, "SWATT:SoftWare-based ATTestation for Embedded Devices," In Proceedings of the IEEE Symposium on Security and Privacy, pp. 272-281, May 2004.

[6] Arvind Seshadri, Mark Luk, Elaine Shi, Adrian Perrig, Leendert van Doorn, Pradeep Khosla, "Pioneer: verifying code integrity and enforcing untampered code execution on legacy systems," In Proceedings of the 20th ACM symposium on Operating Systems Principles, pp. 1-16, 2005.

[7] M. Shaneck, K. Mahadevan, V. Kher, and Y. Kim, "Remote Software-Based Attestation for Wireless Sensors," In ESAS 2005, LNCS 3813, pp. 27-41, 2005.

[8] Gura N., Patel A., Wander A., Eberle A., Shantz S. C, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs", CHES 2004, pp. 119-132, 2004.

[9] https://www.trustedcomputinggroup.org/groups/tpm/

맹 영 재

e-mail : brendig@seclab.inha.ac.kr

2006년 8월 인하대학교 컴퓨터 공학과 졸업

2006년 9월~현 재 인하대학교 정보통신대학원 석사

관심분야 : 인터넷 보안, 네트워크 보안

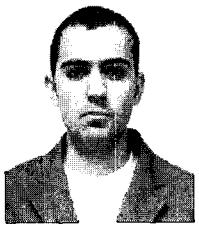
강 전 일

e-mail : dreamx@seclab.inha.ac.kr

2003년 2월 인하대학교 컴퓨터 공학과 졸업

2004년 3월~현 재 인하대학교 정보통신대학원 석사과정

관심분야 : RFID 보안



아 지 즈

e-mail : asm@seclab.inha.ac.kr

2005년 2월 가자대학교 컴퓨터 공학과 졸업

2005년 9월~현 재 인하대학교 정보통신대학원 석사

관심분야 : 네트워크 보안, 암호프로토콜

양 대 헌

e-mail : nyang@inha.ac.kr

1994년 2월 한국과학기술원 과학기술 대학 전기 및 전자 공학과 졸업

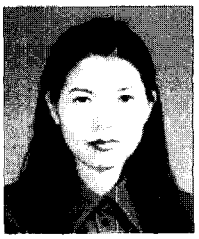
1996년 2월 연세대학교 컴퓨터 과학과 석사

2000년 8월 연세대학교 컴퓨터 과학과 박사

2000년 9월~2003년 2월 한국전자통신연구원 정보보호연구본부 선임연구원

2003년 2월~현 재 인하대학교 정보통신대학원 조교수

관심분야 : 암호이론, 암호프로토콜, 인증프로토콜, 무선 인터넷 보안



이 경 희

e-mail : khlee@suwon.ac.kr

1989년 서울대학교 식품영양학과 학사

1993년 연세대학교 전산과학과 학사

1998년 연세대학교 컴퓨터과학과 석사

2004년 연세대학교 컴퓨터과학과 박사

1993년 1월~1996년 5월 LG소프트(주) 연구원

2000년 12월~2005년 2월 한국전자통신연구원 선임연구원

2005년 3월~현 재 수원대학교 전자공학과 전임강사

관심분야 : 영상처리, 컴퓨터비전, 인공지능, 패턴인식, 생체인식, 얼굴인식, 다중생체인식