Journal of The Korea Society of Computer and Information
Vol. 24 No. 4, pp. 11-18, April 2019
www.ksci.re.kr
https://doi.org/10.9708/jksci.2019.24.04.011

Ultra-light Mutual Authentication Scheme based on Text Steganography Communication

Wan Yeon Lee*

Abstract

Previous mutual authentication schemes operate on the basis of validated cryptographic functions and hash functions, but these functions require a certain amount of memory capacity. However, since ultra-lightweight IoT devices have a very small amount of memory capacity, these functions can not be applied. In this paper, we first propose a text steganography communication scheme suitable for ultra-lightweight IoT devices with limited resources, and then propose a mutual authentication scheme based on the text steganography communication. The proposed scheme performs mutual authentication and integrity verification using very small amount of memory. For evaluation, we implemented the proposed scheme on Arduino boards and confirmed that the proposed scheme performs well the mutual authentication and the integrity verification functions.

▸Keyword: Ultra-lightweight, IoT, Steganography communication, Mutual authentication, Integrity

I. Introduction

최근들에 사물인터넷 시장의 활성화로 미약한 컴퓨팅 자원(작은 메모리, 한정된 배터리 전원, 낮은 CPU 속도)을 가진 초경량 무선 컴퓨팅 기기들이 많이 출시되어 사용되고 있다[1,2]. 기존에는 초경량 무선 컴퓨터 장치가 인터넷에 연결되지 않고 분리된 네트워크 시스템으로 독립적으로 동작하여 보안 침해로 인한 피해가 크지 않았으나, 모든 컴퓨터 기기들을 연결하는 사물인터넷 환경에서는 보안이 가장 취약한 초경량 컴퓨팅 장치들로 인하여 전체 네트워크의 보안에 심각한 피해가 발생하는 사례들이 발생하고 있다[3,4]. 기존 암호 알고리즘들은 높은 자원 소모(프로그램 실행 크기 증가, CPU 추가 연산량 증가, 배터리 전원 소모 증가)를 요구하므로 미약한 컴퓨팅 자원을 가진 초경량 무선 컴퓨팅 장치들에 적용할 수 없다[5]. 따라서 미약한 컴퓨팅 자원(작은 메모리, 한정된 배터리 전원, 낮은 CPU 속도)을 가진 초경량 컴퓨팅 장치들에게 적용할 수 있는 적은 자원 소모(적은 프로그램 실행 코드, 적은 CPU 연산량, 적은 배터리 전원 소모)를 가진 보안 기능의 개발이 필요하다.

본 논문에서는 초소형 메모리 용량을 가진 임베디드 장치에 적합한 상호 인증 기법을 제안한다. 제안된 상호 인증 방법은 기존의 널리 알려진 고성능 암호 함수와 해쉬 함수 대신에 초경량의 문자 은닉(steganography) 통신 방법을 새롭게 제안하여 활용하였다. 제안된 초경량 상호 인증 방법에서는 클라이언트와 서버가 사전에 은닉 통신 절차를 비밀리에 합의하고, 비밀리에 합의된 은닉 통신을 기반으로 상호 인증과 무결성 검사 기능을 수행한다.

안전한 통신을 위해서 기존에도 많은 암호 기법들이 개발되어 왔다. RSA, ECC와 같은 비대칭키(Asymmetric-key)를 사용하는 암호 알고리즘은 높은 보안성을 가지지만 구현이 복잡하고 많은 하드웨어 자원 소모를 요구하므로 경량 장치에 적용하기에는 부적합하다[6]. DES, Triple-DES, AES와 같은 대칭키(Symmetric-key)를 사용하는 암호 알고리즘은 비대칭키를 사용하는 기법에 비해서 상대적으로 구현이 용이하고 적은 자원 소모를 요구한다[7,8].

최근 들어 비슷한 보안 성능을 가지면서 적은 자원 소모를 요구하는 경량의 대칭키 암호 알고리즘들이 개발되고 있다. 많

• First Author: Wan Yeon Lee, Corresponding Author: Wan Yeon Lee
*Wan Yeon Lee (wanlee@dongduk.ac.kr), Dept. of Computer Science, Dongduk Women's University
• Received: 2019. 02. 08, Revised: 2019. 04. 04, Accepted: 2019. 04. 08.
• This research was supported by the Dongduk Women's University Grant, 2018.

이 사용되고 있는 경량의 대칭키 암호 알고리즘들로는 AES, SEED, ARIA, LEA 등이 있고, 해쉬 알고리즘으로는 SHA128, SHA256 등이 있다[1,9,10]. 이러한 경량 알고리즘은 수십 MBytes 크기의 메모리 용량을 가진 경량 임베디드 장치에서는 무리 없이 동작하지만, 아두이노(Arduino) 보드와 같은 수십 KBytes 크기의 메모리 용량을 가진 초경량 임베디드 장치에서는 종종 정상적으로 동작하지 않는 문제점을 가진다[11]. Seo 등[15]의 연구에서는 경량 대칭키 암호 알고리즘들이 소모하는 프로세서 계산량을 조사하였으나, 메모리 소모량에 대한 조사는 다루어지지 않았다. Lee 등[16]의 연구에서 사물인터넷 환경에 적합한 경량화 블록 암호 알고리즘을 제안하였지만, 제안된 알고리즘의 하드웨어 자원 소모량은 기존의 경량 암호 알고리즘과 비교하여 줄어들지 않았다. 기존의 경량 암호 알고리즘들은 프로세서 소모량을 최소화하도록 설계되었다면, 본 논문에서 제안된 방법은 메모리 소모량을 최소화하도록 설계되었다는 점에서 차별성을 가진다.

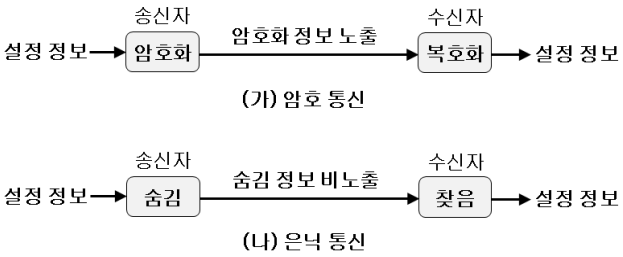


Fig. 1. Comparison of cipher communication and steganography communication

그림 1은 기존의 암호 통신에 기반한 인증 방법과 제안된 은닉 통신에 기반한 인증 방법의 차이를 보여주고 있다. 기존의 암호 통신은 그림 1의 (가)에서 보여주듯이, 인증에 필요한 설정 정보를 암호화하고 암호화된 정보를 통신하는 과정에서 노출을 허용한다. 이러한 접근 방법은 암호화된 정보가 타인에게 노출되더라도 타인은 복호화가 어렵도록 설계되어, 암호화 알고리즘과 복호화 알고리즘이 소모하는 컴퓨팅 자원이 크다. 제안된 은닉 통신은 그림 1의 (나)에서 보여주듯이, 인증에 필요한 설정 정보를 다른 메시지들 내에 숨겨서 통신하는 과정에서 노출되지 않도록 한다. 이러한 접근 방법은 정보를 숨기는 알고리즘과 숨겨진 정보를 찾는 알고리즘에 의해 컴퓨팅 자원의 소모량이 결정된다.

본 논문에서는 수십 KBytes 크기의 메모리 용량을 가진 초경량 임베디드 장치에서도 동작 가능한 문자 은닉 통신 방법을 먼저 제안하고, 이후에 제안된 문자 은닉 통신을 기반으로 동작하는 상호 인증 기법을 추가로 제안한다. 제안된 문자 은닉 통신은 네트워크를 통해서 전송되는 문자 디지털 코드에서 사용되지 않는 영역에 감추고자 하는 정보를 사전에 비밀리에 논의된 은닉 연산을 기반으로 삽입한다. 그리고 제안된 상호 인증 기법은, 기존의 상호 인증 기법[12,13]과 동작 절차는 동일하지만 통신 과정에서 암호 함수 통신 대신에 본 논문에서 제안된 문자 은닉 통신을 기반으로 동작하도록 개선하였다.

본 논문의 나머지 부분은 다음과 같이 구성된다. II 장에서는 기존의 경량 보안 알고리즘들의 성능과 제약 사항에 대해서 설명한다. III 장에서는 제안된 문자 은닉 통신 방법과 상호 인증, 그리고 무결성 확인 기법에 대해서 상세히 설명한다. IV 장에서는 제안된 방법을 실제 시스템상에 구현하여 성능을 평가한다. 마지막으로 V 장에서는 본 논문의 내용을 요약하고 정리한다.

II. Preliminary Knowledge

Lee[11]의 연구에서 현재 널리 사용되고 있는 경량 암호 알고리즘인 AES, SEED, ARIA, LEA, SHA256의 공개 코드(open source)를 아두이노 우노(Arduino Uno) 보드와 아두이노 메가(Arduino Mega) 2560 보드에 탑재하여 동작 성능을 측정하였다. 아두이노 우노 보드는 8비트 마이크로콘트롤러를 사용하고, SRAM 메모리 용량은 2 KBytes로 초소량이고 하드디스크 역할을 수행하는 플래쉬 메모리 용량은 32 Kbytes이다. 플래쉬 메모리에서 8 KBytes는 부트로더가 사용하여 실제로는 24 KBytes만이 사용 가능하다. 아두이노 메가 2560 보드는 동일하게 8비트 마이크로콘트롤러를 사용하고, SRAM 메모리 용량은 8 KBytes로 초소량이고 하드디스크 역할을 수행하는 플래쉬 메모리 용량은 256 Kbytes이다. 마찬가지로 플래쉬 메모리에서 8 KBytes는 부트로더가 사용하여 실제로는 248 KBytes만이 사용 가능하다.

Lee[11]의 연구에서 아두이노 장치의 메모리 용량 제약이 암호 알고리즘의 정상적인 동작에 미치는 영향을 확인하기 위해서, 아두이노 스케치(Sketch) 1.6.5 통합 개발 프로그램을 사용하여 선택된 5개의 경량 암호 알고리즘들을 컴파일하여 실행 코드를 아두이노 우노 보드와 아두이노 메가 2560 보드에 탑재하였다. 아두이노 우노 보드의 동작 실험에서는 SHA256 해쉬 함수만 정상적으로 동작하였고, 나머지 AES, ARIA, SEED, LEA 블록 암호 알고리즘들은 메모리 용량 제약으로 인해서 실행 코드를 보드로 탑재하는 과정에서 에러가 발생하여 동작을 시작조차 할 수 없었다. 그리고 아두이노 메가 2560 보드에서는 대부분의 알고리즘들을 실행 코드들을 보드에 탑재할 수 있었고 정상적으로 동작하였지만, SEED 암호 알고리즘은 탑재 과정에서는 문제가 없었지만 메모리 용량 부족으로 인해서 잘못된 실행 결과가 발생하는 경우가 있었다. 아두이노 메가 2560 보드 실험에서 AES 알고리즘의 최대 실행 코드는 약 25 KBytes, SEED 알고리즘의 최대 실행 코드는 약 30 KBytes, ARIA 알고리즘의 최대 실행 코드는 약 30 KBytes, LEA 알고리즘의 최대 실행 코드는 약 33 KBytes, 그리고 SHA256 알고리즘의 최대 실행 코드는 약 9 KBytes 이었다.

위의 실험은 응용 프로그램을 제외한 순수한 암호 알고리즘만을 대상으로 정상 동작 여부와 실행 코드와 크기를 확인한

것으로, 응용 프로그램 코드를 포함하는 경우에는 더 많은 문제점을 보일 것으로 예상된다. 따라서 초소량의 메모리 용량을 가진 초경량 임베디드 시스템에 적합하도록 기존의 암호 함수와 해쉬 함수를 대체하는 방법을 개발할 필요가 있다.

III. Proposed Scheme

1. Text Steganography Communication

제안된 문자 은닉 통신 방법은 지정된 사용자 이외에는 전송 정보를 감추고, 또한 적은 컴퓨팅 자원 소모를 요구하여 미약한 컴퓨팅 자원을 가진 초경량 컴퓨팅 장치들에게 적용할 수 있는 조건을 만족한다. 이를 위해서 네트워크를 통해서 전송되는 문자 메시지에서 사용되지 않는 영역에 감추고자 하는 정보를 삽입하는 은닉(steganography) 기법[14]을 사용하였다.

제안된 정보 은닉 기법은 고정된 크기의 메시지를 사용하고, 메시지 내에 문자 디지털 코드를 포함하여 정보를 전송하는 통신 시스템 환경을 대상으로 설계되었다. 고정된 크기의 메시지 저장 공간을 사용하는 경우에는 전송하는 문자열의 끝을 표시하여야 하고, 일반적으로 “NULL“ 제어문자를 사용하여 문자열의 끝을 표시한다. 제안된 정보 은닉 기법은 “NULL“ 제어문자 뒤의 남는 공간에 감추고자 하는 정보를 삽입하고, 맨 마지막 바이트에 은닉된 정보가 있는지 없는지 확인할 수 있는 유효성 검사에 필요한 정보를 삽입한다. 수신자는 전송 받은 메시지에서 “NULL“ 제어문자 뒤의 영역에 있는 은닉된 정보와 맨 마지막 바이트의 정보를 사용하여 유효성 검사를 수행한다. 유효성 검사가 성공하면 “NULL“ 제어문자 뒤의 영역에 있는 은닉된 정보를 복원하고, 유효성 검사에 실패하면 “NULL“ 제어문자 뒤에 있는 정보는 쓰레기로 취급한다.

제안된 정보 은닉 통신 기법은 다음의 상세 과정에 따라 동작한다.

1. 송신자는 노출 되어도 상관없는 메시지를 디지털 문자 배열 앞쪽에 삽입하고, 삽입된 메시지 끝의 바로 뒤에 끝을 의미하는 제어문자 “NULL”을 삽입한다.
2. 제어문자 “NULL” 뒤의 남은 영역에 은닉하고자하는 정보를 사전에 약속된 방법을 사용하여 디지털 문자로 변환하여 삽입한다. 그리고 고정된 크기의 메시지 마지막에 위치에 은닉 정보가 있다는 것을 표시하는 유효성 값을 사전에 약속된 방법을 기반으로 계산하여 삽입한다.
3. 송신자는 만들어진 디지털 문자 배열을 고정된 크기의 메시지에 담아서 수신자에게 전송한다.
4. 수신자는 전송된 디지털 문자 배열에서 “NULL” 제어문자 뒤에 있는 디지털 정보들을 분리하여 사전에 약속된 방법을 사용하여 은닉 정보가 있는지를 확인하는 유효성 검사를 수행한다.
5. 유효성 검사를 통하여 은닉 정보가 있음이 확인되면, “NULL” 제어문자 뒤에 있는 디지털 배열에서 사전에 약속된 변환 방법을 사용하여 은닉 정보를 추출한다. 유효성 검사가 실패하여 은닉 정보가 없음이 확인되면, “NULL” 제어문자 뒤에 있는 디지털 문자 배열은 쓰레기로 처리하고 무시한다.

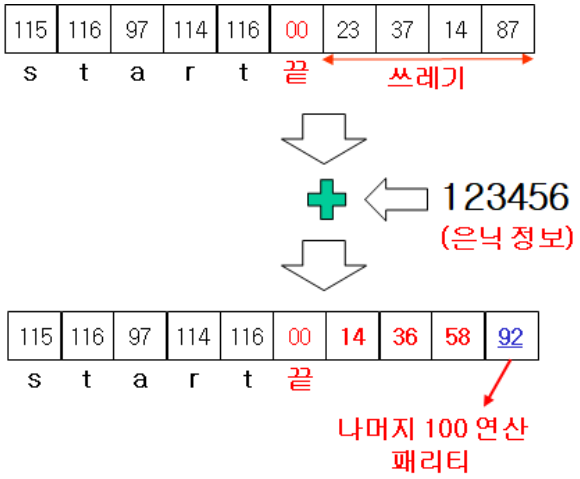


Fig. 2. Working example of the proposed text steganography communication

그림 2는 통신용 메시지 크기가 10 바이트(bytes)로 고정된 시스템 환경에서, 제안된 문자 은닉 통신 기법의 동작 예를 보여주고 있다. 먼저 송신측에서 노출되어도 상관없는 메시지 “start”를 아스키(ASCII) 디지털 코드로 변환하여 메시지 앞쪽에 삽입하고, 그 뒤 쪽에 제어문자 “NULL”의 아스키 디지털 코드 0을 삽입한다. 그리고 은닉하고자 하는 정보 “123456”에 대해서 “12”, “34”, “56”으로 분리하여 각각 더하기 2를 수행하여 변환된 “14”, “36”, “58” 값을 뒤쪽 나머지 영역에 저장한다. 그리고 메시지 마지막 위치에는 유효성 검사를 위해서, 나머지 영역에 저장된 디지털 값들의 합에 대해서 나머지 100 연산을 수행하면 나머지 연산 결과 값이 0이 되도록 하는 패리티 값을 삽입한다. 위의 예에서 92의 값을 마지막 위치에 삽입하면 “(14+ 36+ 58+ 92) 나머지 100 = 0”이 되어 은닉 정보가 있음을 암시한다. 만약 은닉 정보가 없다면 마지막 위치에 92 이외의 값을 삽입하여, 나머지 영역에 있는 값들은 의미가 없는 쓰리게 정보들임을 표시한다.

수신자는 전송된 메시지에서 제어문자 “NULL” 뒤의 나머지 영역에서 디지털 정보들을 분리하여 은닉 정보가 있는지 유효성 검사를 수행한다. 송신자에서 유효성 표시에 사용된 방법과 동일하게, 나머지 영역에 저장된 값들의 합이 나머지 100 연산 결과 값이 0인지 확인한다. “(14+ 36+ 58+ 92) 나머지 100 = 0”이므로 은닉 정보가 있음을 확인할 수 있다. 그리고 송신자에서 사용된 은닉 방법에 상응하는 복원 방법을 사용하여 은닉 정보를 복원한다. 송신자에서 더하기 2를 수행하였으므로, 복원 방법에서는 각각의 디지털 코드에서 빼기 2를 수행한다. “14”는 “12”로 복원되고, “36”은 “34”로 복원되며, “58”은 “56”으로 복원된다. 최종적으로는 “123456”의 정보가 복원된

다. 유효성 검사가 실패하면 디지털 코드 이외의 영역에 있는 정보는 쓰레기로 취급한다.

위의 예에서 사용된 정보 은닉 연산 방법과 유효성 검사 연산 방법은 임의로 변경하여 사용할 수 있으며, 제 3자에게 노출되지 않도록 송신자와 수신자에 의해서 사전에 비밀리에 약속된다.

2. Communication Integrity Verification Scheme

문자 메시지를 전송하는 과정에서 메시지를 중간에 서 변조하여 전달하는 악의적 공격에 대비하기 위해서 무결성 검사 정보를 포함할 필요가 있다. 기존에는 해쉬 함수를 사용하고 있으나[7], 이러한 해쉬 함수를 메모리 용량 초과로 탑재할 수 없는 초경량 장치를 위해서 초경량 무결성 검사 방법을 제안한다.

제안된 무결성 검사 방법에서는 먼저 전달될 문자 메시지의 디지털 코드 값들을 사전에 합의된 임의의 다항식 연산에 적용한다. 그리고 다항식 연산 결과에 대해서 나머지 연산을 적용한 결과 값을 III.1절에서 제안된 문자 은닉 통신 방법을 적용하여 메시지 후반부에 포함하여 전달한다. 무결성 검사에 사용될 다항식 연산과 나머지 연산은 송신자 장치와 수신자 장치간에 사전에 합의하여 비밀리에 공유한다.

만약에 사전에 정의된 다항식 연산이 3•(X1)2 + 2 • (X2)2 + (X3)2 + X4 + ... + Xn 이고, 사전에 정의된 나머지 연산이 “나머지 1000”이라면, 그림 1의 메시지 디지털 코드의 무결성 검사 값은 다음과 같이 계산된다. 다항식 연산을 적용한 값은 3•(115)2 + 2•(116)2 + (97)2 + 114 + 116 = 76,226이고, 나머지 연산을 적용한 값은 “76,226 나머지 1000 = 226”이다. 따라서 “NULL” 제어문자 뒤에 “02”와 “26”의 정보를 III.1절에서 설명한 문자 은닉 통신 기법을 사용하여 전송한다.

매번 무결성 검사 정보를 포함하면 은닉 통신 기법이 노출될 위험이 있다. 따라서 불규칙하게 랜덤 함수를 이용하여 무결정 검사 정보를 포함하는 시기를 결정한다. 예를 들어 송신자와 수신자가 사전에 합의한 비밀키를 seed 값으로 “random(seed) mod 10” 연산을 수행하고, 연산 수행의 결과 값을 송신 메시지에 무결성 검사 정보를 포함시키는 시기로 사용한다. 무결성 검사 정보를 포함하지 않는 메시지들을 계속 누적하고 있다가, 무결성 검사 정보를 포함하는 메시지를 전송할 때에 누적된 메시지들의 디지털 코드 값을 모두 사용하여 무결성 검사 값을 계산하여 전송한다.

수신자 측에서도 동일한 계산식을 사용하여, 언제 무결성 검사 정보 값이 전달되는지 알 수 있다. 수신자 측에서는 무결성 검사 정보가 포함되지 않은 메시지들에 대해서는 판단을 유보하고 누적하여 보관하고 있다가, 무결성 검사 정보가 포함된 메시지가 전달되면 누적된 메시지들을 적용하여 무결성 검사를 수행한다. 만약 무결성이 훼손된 점이 확인되면 누적된 메시지들을 모두 버리고, 송신측에 메시지 재전송 요구를 전달한다.

개발된 정보 은닉 기법과 무결성 검사 기법은 기존의 암호 기술을 적용할 수 없는 초경량 컴퓨팅 기기들의 정보 전송에 대해서 보안 기능을 제공하여, 사물인터넷 시스템의 전체 보안 수준을 향상시킨다.

3. Mutual Authentication Scheme

초경량 장치의 하드웨어 자원 제약으로 기존에 사용된 암호 함수들을 탑재하지 못하는 경우에는 상호인증, 세션키 합의 및 무결성 검사 기능을 제공하지 못한다. 이 경우를 위해서 암호 함수 대신에 은닉 기법을 활용하여 상호인증, 세션키 합의 및 세션키 합의 기능을 개발하였다.

개발된 기법에서는 먼저 클라이언트가 상호인증 시작 요청 메시지를 보낸다. 이때 III.1 절에 설명한 문자 은닉 통신 기법을 사용하여 클라이언트가 본인의 장치 식별자(Identifier)를 은닉하여 전송한다. 장치 식별자를 변환할 때 서버와 사전에 비밀리에 합의된 장치 비밀키를 기반으로 장치 식별자를 변환한다. 변환 방법의 예로 “장치 식별자 + 장치 비밀키”로 변환될 수 있다. 비밀키를 기반으로 장치 식별자를 변환하여 전송하면, 서버는 송신자가 해당 장치 식별자와 비밀키를 모두 알고 있는지 여부를 확인할 수 있다.

클라이언트가 상호인증 시작 요청 메시지에 장치 식별자를 은닉하여 전송할 때, 장치용 난스(nounce) 값을 동적으로 랜덤하게 생성하여 장치 식별자와 같이 은닉하여 전송한다. 이 장치용 난스 값은 나중에 서버로부터 회신 받도록 설계되었고, 이 장치용 난스 값을 이용하여 서버가 진짜 서버인지 가짜 서버인지 확인하는데 사용한다.

상호인증 시작 요청 메시지를 수신한 서버는 은닉된 클라이언트 장치 식별자를 복원하여, 송신자가 장치 식별자와 사전에 합의된 장치용 비밀키를 알고 있는지 여부를 확인한다. 송신자가 정상적인 클라이언트인지 확인이 되면, 시작 확인 메시지를 클라이언트에 회신한다. 이때 클라이언트로부터 송신된 난스 값을 은닉하여 다시 회신하는데, 클라이언트와 사전에 비밀리에 합의된 서버 비밀키를 기반으로 장치용 난스 값을 변환하여 은닉한다. 변환 방법의 예로는 “난스 값 + 서버 비밀키”가 있고, 서버 비밀키와 장치 비밀키는 다른 값을 적용할 수 있다.

서버가 시작 확인 메시지를 회신할 때, 서버용 난스 값을 동적으로 랜덤하게 생성하여 장치용 난스 값과 같이 은닉하여 전송한다. 이 서버용 난스 값은 상호 인증이 완료된 이후에 세션용 비밀키 값으로 활용된다. 세션용 비밀키 값은 상호 인증이 수행될 때마다 바뀌는 값으로, 장치용 비밀키와 서버용 비밀키 값은 상호 인증 과정에서만 사용되어, 상호인증 완료된 이후에는 세션키를 사용하여 문자 은닉을 수행한다. 은닉 방법의 예로, “은닉 정보 + 세션키”가 될 수 있다.

시작 확인 메시지를 수신한 클라이언트 장치는 은닉된 장치용 난스 값을 복원하여 자신이 송신했던 난스 값과 일치하는지 여부를 확인한다. 두 값이 일치되면 회신한 서버가 사전에 합의된 서버용 비밀키를 알고 있음을 확인할 수 있다. 클라이언트 장치는 일치된 장치용 난스 값이 은닉된 시작 확인 메시지를 수신하면, 상호 인증이 성공했음을 확인하고 서버와 일반 메시

지 송수신을 시작한다. 클라이언트가 메시지 송신할 때는 장치용 난스 값을 기반으로 정보를 은닉하고, 서버가 메시지를 회신할 때에는 서버용 난스 값을 기반으로 정보를 은닉한다. 클라이언트가 은닉된 정보를 복원할 때에는 서버용 난스 값을 기반으로 계산하고, 서버가 은닉된 정보를 복원할 때에는 장치용 난스 값을 기반으로 계산한다.

상호 인증이 완료된 이후의 일반 메시지 송수신시에는 무결성 훼손 확인을 위해서, Ⅲ.2 절에서 설명한 무결성 검사 값을 메시지에 은닉하여 같이 전송한다. 무결성 검사 값을 은닉할 때는, 위에서 설명한 상호 인증 과정에서 생성된 클라이언트 장치용 난스 값과 서버용 난스 값을 사용하여 무결성 검사 값을 은닉한다. 무결성 검사 값을 변환하는 방법의 예로, 사전에 비밀리에 합의된 다항식에 메시지 값을 적용하고 이후에 "(다항식 출력 값) 나머지 (장치용 난스 값)"이 있다.

제안된 기법의 상세 동작을 도식화하기 위해서 아래와 같은 수식들을 정의하여 사용하였다.

- 난스(nounce) : 사전 예측이 안되고 반복되지 않는 랜덤 값 (현재 시간 값의 나머지 연산 결과 값을 주로 활용)
- A | B : 메시지 A 값과 메시지 B 값을 연결
- $S_{키}$(입력) : Ⅲ.1 절에서 설명하였듯이, 입력 디지털 값에 대해서 "키" 값을 기반으로 변환하여 사용되지 않는 메시지 나머지 영역에 삽입하는 기능
- $H_{키}$(입력) : Ⅲ.2 절에서 설명하였듯이, 입력 디지털 값을 먼저 사전의 합의된 다항식에 적용하고 이후에 다항식 출력 값을 "키" 값을 기반으로 나머지 연산을 수행한 결과 값

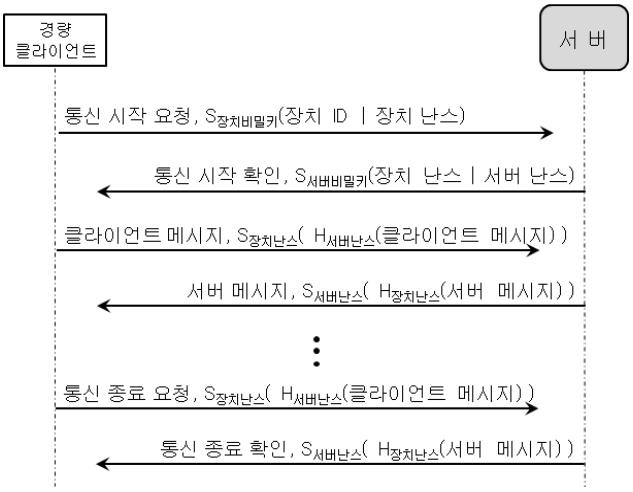


Fig. 3. Communication protocol of the proposed mutual authentication

그림 3은 위에서 정의된 수식들을 기반으로 클라이언트와 서버 간의 제안된 상호 인증 절차와 무결성 검사 기능이 포함된 일반 통신 흐름을 보여주고 있다. 그림 3의 첫 번째 메시지는 클라이언트 장치가 통신 시작 요청 메시지를 서버에게 보내는 과정이다. 클라이언트 장치는 이 메시지를 송신할 때에 장치용 난스 값을 동적으로 생성하고, 생성된 장치용 난스 값과 클라이언트 장치 식별자(Identifier) 값을 연결한 값을 입력으로 은닉 함수 S( )를 적용하여 통신 시작 요청 메시지 후반부에 숨긴다. 은닉 함수 S( )에 사용되는 키 값으로, 사전에 클라이언트와 서버가 동시에 저장하고 있는 클라이언트 장치용 비밀키를 사용한다.

그림 3의 두 번째 메시지는 서버가 통신 시작 확인 메시지를 회신하는 과정이다. 서버용 장치는 이 메시지를 송신할 때에 서버용 난스 값을 동적으로 생성하고, 수신된 클라이언트용 난스 값과 연결하여 은닉 함수 S( )에 입력 값으로 적용하여 통신 시작 확인 메시지 후반부에 숨긴다. 은닉 함수 S( )에 사용되는 키 값으로, 사전에 클라이언트와 서버가 동시에 저장하고 있는 서버용 비밀키를 사용한다.

세 번째 메시지부터는 상호 인증이 성공한 이후에, 클라이언트 장치와 서버 간에 일반 메시지 송수신을 진행하는 과정이다. 클라이언트 장치가 메시지를 서버에게 송신할 때에는 무결성 검사 값을 생성하여 메시지 후반부에 숨겨서 전송한다. 무결성 검사 함수 H( )를 적용할 때는 두 번째 메시지에서 수신된 서버용 난스 값을 키 값으로 사용한다. 그리고 은닉 함수 S( )를 적용할 때에는 클라이언트 장치용 난스 값을 키 값으로 사용하여 S( ) 함수를 수행한다. 반면 서버가 클라이언트 장치에게 메시지를 전송할 때에는, 무결성 검사 함수 H( )의 키 값으로 클라이언트 장치 난스 값을 사용하고, 은닉 함수 S( )의 키 값으로 서버용 난스 값을 사용한다.

개발된 방법은 다음과 같이 상호 인증 기능을 수행한다. 클라이언트 인증은, 서버가 수신한 시작 요청 메시지에 은닉된 클라이언트 장치 식별자를 사전에 비밀리에 합의된 장치 비밀키를 기반으로 복구하면, 서버 측에서 송신자가 해당 장치 식별자와 비밀키를 알고 있는지 여부를 확인할 수 있다. 서버 인증은, 클라이언트 장치가 수신한 시작 확인 메시지에 은닉된 장치 난스 값을 사전에 비밀리에 합의된 서버 비밀키를 기반으로 복구하면, 서버가 서버 비밀키를 알고 있는지 여부와 본인이 보낸 인증 요청 메시지 대한 응답 메시지인지 여부를 확인할 수 있다.

개발된 방법은 다음과 같이 세션키 합의 기능을 수행한다. 사전에 합의된 장치 비밀키와 서버 비밀키는 상호 인증 과정에서만 사용하여 노출을 최소화한다. 그리고 상호 인증 과정에서 동적으로 생성하여 상호간에 은닉 전송한 장치용 난스 값과 서버용 난스 값을 세션용 비밀키로 사용한다. 이 세션용 비밀키는 상호 인증이 수행될 때마다 바뀌는 특성을 가지고, 상호 인증이 완료된 이후의 일반 메시지 송수신시에 정보 은닉을 위한 비밀키로 사용된다.

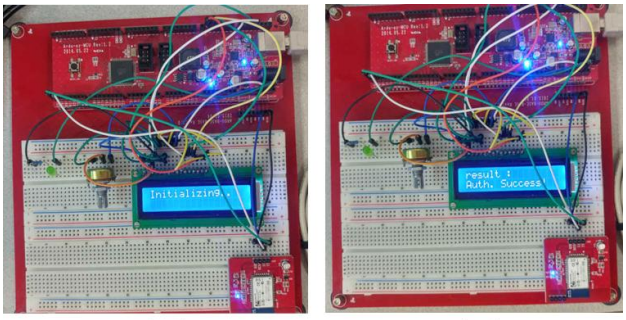
개발된 방법은 다음과 같이 통신 메시지의 무결성 검사 기능을 수행한다. 송신된 메시지에 은닉되어 전달된 무결성 검사 값과 수신측에서 자체 계산한 무결성 검사 값이 서로 일치하면 수신된 메시지가 전송 중간에 변조되지 않은 정상적 메시지로 확인한다. 그리고 전달된 무결성 검사 값과 수신측에서 자체 계산한 무결성 검사 값이 다르면, 수신된 메시지를 변조된 메시지로 판별하고 폐기한다.

개발된 기법은 은닉 함수를 주기적으로 바꾸어서 보완성을 강화시킬 수 있다. 예를 들어 비밀키 값을 사용한 정보 은닉 과정에서, (정보 + 비밀키) 연산과 (정보 + 2*비밀키) 연산을 번갈아 가면서 사용하여 보완성을 강화시킬 수 있다.

개발된 정보 은닉 기법은 최근에 사물인터넷의 활성화로 미약한 하드웨어 자원(작은 메모리, 한정된 배터리 전원, 낮은 CPU 속도)을 가진 초경량 무선 컴퓨터를 위한 보안 기술의 수요를 만족시킨다. 기존 암호화 알고리즘들은 높은 자원 소모(프로그램 실행 코드 증가, CPU 추가 연산량 증가, 배터리 전원 소모 증가)를 요구하여 초경량 무선 컴퓨터에 적용할 수 없는 반면, 개발된 정보 은닉 기법은 미약한 하드웨어 자원(작은 메모리, 한정된 배터리 전원, 낮은 CPU 속도)을 가진 초경량 무선 컴퓨터에 적용할 수 있는 적은 자원(적은 프로그램 실행 코드, 낮은 CPU 연산량, 적은 전원 소모량)만을 사용한다.

IV. Performance Evaluation

성능 평가를 위해서 제안된 방법을 실제 시스템으로 구현하였다. 경량 클라이언트 장치로는 ATmega MCU가 탑재된 RNU사의 Ardu-Base 보드를 사용하였고, 서버 장치로는 안드로이드 7.1.1 기반의 Nexus 5X 스마트폰을 사용하였다. 클라이언트 장치와 서버 간에는 블루투스 무선 통신을 32 바이트 고정 크기로 메시지를 주고받는다. 클라이언트 장치는 2 KBytes의 메모리 용량과 32 KBytes의 하드디스크 용량만을 지원하고, 블루투스 통신을 위해서 RN42-BF10 하드웨어 모듈을 사용하였다.

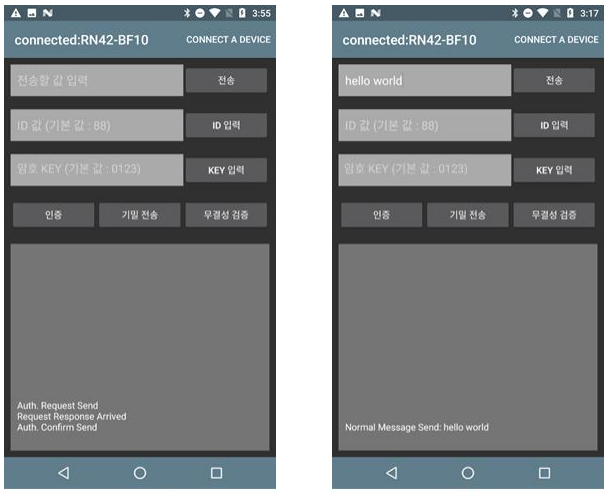


(가) (나)

Fig. 4. Implemented client device

제안된 상호 인증 기능과 무결성 검사 기능을 아두이노 스케치 개발환경에서 구현하여 클라이언트 보드에 탑재하였다. 아두이노 스케지 1.6.5 버전에서 컴파일한 이후의 실행 코드는 약 1.2 KBytes로 극소량의 메모리 사용량을 가진다. 프로그램 상에서 클라이언트 장치 식별자 값으로 “88”을 사용하고, 비밀키로는 “1234”를 사용하였다. 그림 4는 경량 클라이언트 장치가 서버와 통신을 통해서 상호 인증이 정상적으로 수행된 결과를 보여주고 있다. 그림 4의 (가)는 클라이언트 장치가 통신 시작 요청 메시지에 장치 식별자와 장치용 난스 값을 은닉하여 서버에게 보내고 확인 메시지를 기다리고 있는 상태를 LCD 창에 출력하고 있다. 그림 4의 (나)는 서버로부터 통신 시작 확인 메시지를 받고, 은닉된 장치용 난스 값을 복원하여 일치 여부를 확인 후에 “Auth. Success” 실행 결과를 LCD 창에 출력하고 있다.

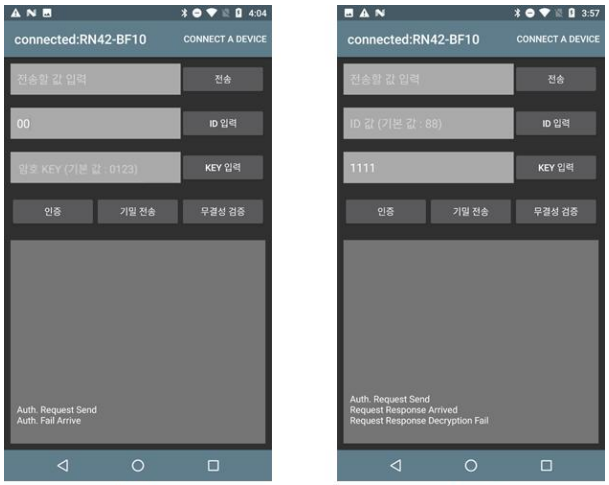
서버 측에서 수행하는 상호 인증 기능과 무결성 검사 기능은 기법은 Android Studio 개발환경에서 안드로이드 앱(App)으로 구현하여 Nexus 5X 스마트폰에 탑재하였다. 그림 5의 (가)는 서버 장치가 클라이언트와 상호 인증 요청을 받아서 상호 인증을 정상적으로 수행 완료한 결과를 보여주고 있다, 그림 5의 (나)는 상호 인증 이후에 서버에서 생성된 난스 값을 비밀키로 사용하여 메시지를 클라이언트에 정상적으로 전송하는 결과를 보여주고 있다.



(가) (나)

Fig. 5. Android App of implemented server device

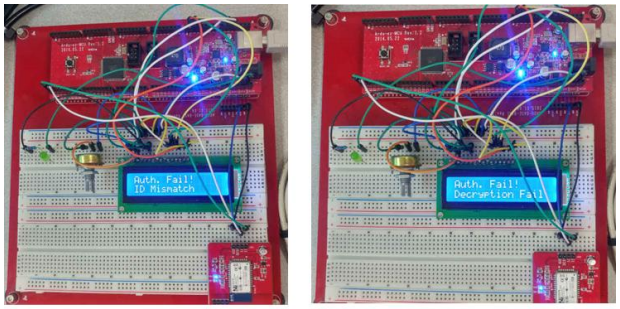
그림 6은 잘못된 클라이언트 장치 식별자를 사용하거나 사전에 합의된 비밀키와 다른 값을 적용하는 경우에, 서버측의 인증 실패를 보여주고 있다. 그림 6의 (가)는 서버가 보관하고 있는 클라이언트 장치 식별자 값을 “88” 대신 “00”으로 변경하는 경우의 동작 과정을 보여주고 있다. 클라이언트 장치가 “88”의 장치 식별자 값을 은닉하여 전송하면, 서버는 은닉된 정보를 장치 식별자 값 “88”을 복원하고 자신이 보관하고 있는 장치 식별자 값 “00”과 비교하여, 불일치 이유로 인증을 거부하는 결과이다. 그림 6의 (나)는 서버가 사용하는 비밀키 값을 “1234” 대신 “1111”로 변경하는 경우의 동작 과정을 보여주고 있다. 서버가 수신된 메시지에서 비밀키 “1111”를 사용하여 은닉된 클라이언트 장치 식별자를 복원하는데, 변경된 비밀키 값으로 인해서 복원된 장치 식별자 값이 “88”과 상이하여 인증을 거부하는 결과이다.



(가) (나)

Fig. 6. Authentication failure of sever device

그림 7은 클라이언트측의 인증 실패 결과를 보여준다. 그림 7의 (가)는 잘못된 클라이언트 장치 식별자를 사용하는 경우의 클라이언트 동작 결과로, 그림 6의 (가)에서 보여주는 서버측의 인증 실패 회신 메시지를 받고 LCD 창에 “Auth. Fail! ID Mismatch”라는 실행 결과를 출력하고 있다. 그림 7의 (나)는 사전에 합의된 비밀키와 다른 값을 적용하는 경우의 클라이언트 동작 결과로, 그림 6의 (나)에서 보여주는 서버측의 인증 실패 회신 메시지를 받고 LCD 창에 “Auth. Fail! Decryption Fail”라는 실행 결과를 출력하고 있다.



(가) (나)

Fig. 7. Authentication failure of client device

V. Conclusion

본 논문에서는 초소형 메모리 용량을 가진 임베디드 장치에 적합한 상호 인증 기법을 제안한다. 제안된 상호 인증 방법은 초경량의 문자 은닉 통신 방법을 새롭게 제안하여 활용하였다. 제안된 초경량 상호 인증 방법에서는 클라이언트와 서버가 사전에 은닉 통신 절차를 비밀리에 합의하고, 비밀리에 합의된 은닉 통신을 기반으로 상호 인증과 무결성 검사 기능을 수행한다. 제안된 방법의 성능 평가를 위해서 초소량의 메모리 용량을 가진 아두이노 보드에서 제안된 방법을 실제로 구현하였고, 상호 인증 기능과 무결성 검사 기능을 정상적으로 수행함을 확인하였다.

제안된 정보 은닉 기법은 기존의 암호 기술을 적용할 수 없는 초경량 컴퓨팅 기기들의 정보 전송에 대해서 보안 기능을 제공하여, 사물인터넷 시스템의 전체 보안 수준을 향상시킬 수 있을 것을 기대된다. 예를 들어, 제안된 기법은 다양한 분야에서 최근 들어 널리 사용되고 있는 초경량 기기인 아두이노 보드에 활용될 수 있다. 사용자들이 아두이노 보드를 이용하여 스마트폰의 블루투스 통신을 기반으로 무선 제어를 받는 가정용 보일러 제어기와 같은 스마트 가전 기기들을 직접 제작할 때, 기존에는 블루투스 무선 통신 경로에 보안 기능을 탑재할 수 없었지만 제안된 정보 은닉 기법을 적용하면 보안 기능 탑재가 가능하게 된다. 따라서 아두이노 보드 기반의 신규 개발 장치들의 보안 기능을 향상시킬 것으로 기대된다.

제안된 정보 은닉 기법은 기존의 암호 알고리즘들과 비교하여 상대적으로 보안 견고성이 취약하여, 많은 양의 통신 메시지를 수집하여 분석하면 은닉된 정보가 노출될 가능성이 있다는 한계점을 가진다. 그러나 본 논문에서 다루어진 연구는 초경량 기기에 적합한 보안 기술을 처음으로 소개하여, 관련된 후속 연구들을 촉진시킬 것으로 기대된다.

REFERENCES

[1] Wan Yeon Lee and Yun-Seok Choi, “Optimization of ARIA Block-Cipher Algorithm for Embedded Systems with 16-bits Processors,” International Journal of Internet, Broadcasting and Communication, vol. 8, no. 1, pp. 88-98, Feb. 2016.

[2] M. Ebrahim, S. Khan and U. B. Khalid, “Symmetric Algorithm Survey: A Comparative Analysis,” International Journal of Computer Applications, vol. 61, no. 20, pp. 12-19, Jan. 2013.

[3] S. B. Sasi and N. Sivan, “A Survey on Cryptography Using Optimization Algorithms in WSNs,” Indian Journal of Science and Technology, vol. 8, no. 3, pp. 216-221, Feb. 2015.

[4] Y. W. Law, J. Doumen and P. Hartel, “Survey and Benchmark of Block Cipher for Wireless Sensor Networks,” ACM Transactions on Sensor Networks, vol. 2, no. 1, pp. 65-93, Feb. 2006.

[5] Meikang Qiu, Hai Su, Min Chen, Zhong Ming, and Laurence T. Yang, “Balance of Security Strength and Energy for PMU Monitoring Systems in Smart Grid,” IEEE Communications Magazine, vol. 50, no. 5, pp. 142-149, May 2012.

[6] Rene Hummen, Jan H. Ziegeldorf, Hossein Shafagh, Shahid Raza and Klaus Wehrle, "Towards Viable Certificate-based Authentication for the Internet of Things," ACM HotWiSec, pp. 54-63, April 2013.
[7] William Stalling, Cryptography and Network Security: Principles and Practices (6th Edition), Prentice Hall, 2013.
[8] National Institute of Standards and Technology (NIST), "Advanced Encryption Standards (AES)," Federal Information Processing Standards Publication 197, pp. 1-26, Nov. 2001.
[9] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann and L. Uhsadel, "A Survey of Lightweight-Cryptography Implementations," IEEE Design & Test of Computers, vol. 24, no. 6, pp. 522-533, Nov.-Dec. 2007.
[10] Gi-tae Park, Hyo-joon Han and Jae-hwoon Lee, "Design and Implementation of Lightweight Encryption Algorithm on OpenSSL," The Journal of Korean Institute of Communications and Information Sciences, vol. 39B, no. 12, pp. 822-830, 2014.
[11] Wan Yeon Lee, "Performance Evaluation of Lightweight Security Algorithms for Embedded Systems with Low Capacity," Dongduk Journal of Natural Science and Computer Science, vol. 3, pp. 71-80, Dec. 2018.
[12] Geum-Boon Lee, "A Secure Authentication Method for Smart Phone based on User's Behaviour and Habits," Journal of the Korea Society of Computer and Information, vol. 22, no. 9, pp. 65-71, Sep. 2017.
[13] Seokhyang Cho, "An Efficient Group Key Agreement Using Hierarchical Key Tree in Mobile Environment," Journal of the Korea Society of Computer and Information, vol. 23, no. 2, pp. 53-61, Feb. 2018.
[14] Prem Singh, Rajat Chaudhary and Ambika Agarwal, "A Novel Approach of Text Steganography based on Null Space," IOSR Journal of Computer Engineering, vol. 3, no. 4, pp. 11-17, July-Aug. 2012.
[15] Hwa-Jeong Seo, Tae-Hwan Park and Ga-Ram Lee, "Implementation Technique of Symmetric-Key Cryptography on Lightweight IoT Platforms," Korea Institute of Information Security and Cryptology, vol. 27, no. 6, pp. 15-20, Dec. 2017.
[16] Seon-Keun Lee, "A Study on Lightweight Block Cryptographic Algorithm Applicable to IoT Environment," Journal of the Korea Academia-Industrial Cooperation Society, vol. 19, no. 3, pp. 1-7, 2018.

Authors

Wan Yeon Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from POSTECH, Korea, in 1994, 1996 and 2000, respectively Dr. Hong joined the faculty of the Department of Computer Science at Dondguk Women's University, Seoul, Korea, in 2011. He is currently a Professor in the Department of Computer Science, Dongdu Women's University. He is interested in embedded system, system software, and system security.