IoT 환경에서 CoAP의 데이터 무결성 검증을 위한 솔팅 및 스테가노그래피 기반 접근법

김건민[1], 이은성[1], 장현지[2], 이서준[2], 김경백[3]
[1]전남대학교 소프트웨어공학과 학부생
[2]전남대학교 인공지능학부 학부생
[3]전남대학교 인공지능융합학과 교수
204869@jnu.ac.kr, 200750@jnu.ac.kr, gka1225@jnu.ac.kr, tjwns1300@jnu.ac.kr, kyungbaekkim@jnu.ac.kr

Enhancing CoAP Security in IoT Environments Using Salting and Steganography for Data Integrity Verification

Geonmin Kim[1], Eunseong Lee[1], Hyeonji Jang[2], Seojun Lee[2], Kyungbaek Kim[3]
[1]Dept. of Software Engineering, Chonnam National University
[2]Dept. of Artificial Intelligence, Chonnam National University
[3]Dept. of Artificial Intelligence Convergence, Chonnam National University

요 약

본 논문은 IoT(Internet of Things) 환경에서 CoAP(Constrained Application Protocol)의 보안을 강화하기 위해 해시 함수와 스테가노그래피를 활용하는 접근법을 제안한다. IoT 장치의 자원 제약을 고려하여, 패킷에 솔팅을 이용한 해시 함수와 스테가노그래피를 사용하여 해시 알고리즘과 솔트 값을 은닉하는 방법을 사용한다. 이 접근법은 자원 소모를 최소화하며 데이터 무결성을 검증한다.

1. 서론

사물인터넷(Internet of Things, IoT)은 다양한 장치와 센서가 네트워크를 통해 서로 연결되어 데이터를 교환하는 시스템을 의미한다. IoT 장치가 다루는 데이터는 사용자와 밀접하여, 개인정보 보호와 데이터 무결성 확보의 중요성이 커지고 있다.

CoAP(Constrained Application Protocol)[1]는 IoT 환경에서 효율적인 통신을 위해 설계된 경량 프로토콜로 낮은 오버헤드를 유지하여 자원 제약이 있는 장치에서도 적합하게 동작할 수 있도록 설계되었지만, 기본적으로 제공하는 보안 메커니즘만으로는 다양한 보안 위협에 충분히 대응하기 어렵다.

본 논문에서는 CoAP 환경에서 데이터의 무결성을 검증하기 위해 솔팅과 스테가노그래피를 활용하는 방법을 제안한다.

2. 솔팅과 스테가노그래피

솔팅[2]은 해시 함수에 임의의 데이터인 솔트 값을 추가하여 해시값을 생성하는 방법이다. 솔트를 사용함으로써 동일한 입력 데이터라도 서로 다른 해시 값을 생성할 수 있으며, 이는 레인보우 테이블 공격이나 무차별 대입 공격에 대한 저항력을 높이고 해시 충돌을 방지하는 데 기여한다.

스테가노그래피[3]는 비밀 메시지를 오디오나 이미지와 같은 특정 커버 데이터에 은닉하여 전송하는 기술이다. 네트워크 스테가노그래피[4]는 프로토콜의 헤더 필드나 페이로드 필드에 데이터를 숨기는 방법을 사용한다. 본 논문에서는 솔트 값을 스테가노그래피를 통해 이미지 파일에 은닉하여 전송하는 방안을 제안한다.

3. 제안하는 무결성 검증 방법

3.1. 패킷 설계

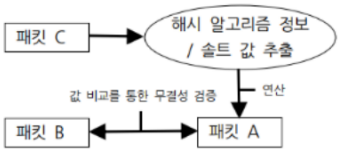
패킷 A는 CoAP 메시지의 기본 구성 요소로, 데이터 전송 시 필요한 최소한의 정보를 포함한다. 패킷 A의 페이로드는 IoT 기기에서 생성된 데이터를 담고 있다.

패킷 B는 패킷 A의 페이로드에 랜덤 솔트를 추가하고 SHA-256 해시 함수 알고리즘을 적용하여 생성된다. 패킷 B는 해시 값과 솔트 값을 포함하여, 원본 데이터의 무결성을 검증의 비교 대상이 된다.

패킷 C는 패킷 B에서 사용한 해시 알고리즘과 솔트를 스테가노그래피를 사용하여 은닉한다. CoAP

메시지의 페이로드 또는 옵션 필드에 정보를 숨긴다. 이 방식은 CoAP 메시지의 비트 패턴에 정보를 은닉하는 기술을 활용한다.

3.2. 무결성 검증

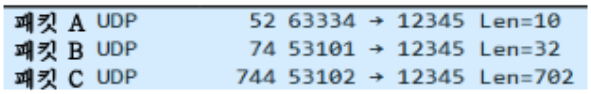


(그림 1) 제안하는 패킷의 무결성 검증 과정

수신 측에서는 CoAP 메시지(패킷 A, B, C)를 수신한 후, 패킷 C에서 스테가노그래피로 숨겨진 해시 알고리즘과 솔트를 추출한다. 이 정보를 바탕으로 패킷 A의 데이터에 해시 함수와 솔트를 적용하여 생성된 해시 값이 패킷 B의 해시 값과 일치하는지 확인하여 패킷 A와 패킷 B의 무결성을 검증한다.

4. 성능 평가

위의 패킷 A, B, C를 Python 3.9.6의 환경에서 생성 후 생성 후 UDP로 전송하여 실험을 진행하였다. 해시 알고리즘과 솔트 값은 1KB 크기의 이미지 파일에 은닉 후 패킷 C를 생성하였다.



(그림 2) Wireshark를 이용한 패킷 확인

패킷 A는 평문, 패킷 B는 패킷 A의 데이터를 기반으로 한 해시값, 패킷 C는 해시 알고리즘과 솔트 값이 은닉된 이미지 파일을 포함하고 있으므로 (그림 2)와 같은 패킷의 길이 증가가 나타났다.

패킷의 암호화 과정에 추가된 데이터로 인한 길이 증가로 발생하는 패킷 전송 시간의 지연을 확인하기 위해 패킷 A에 해당하는 평문과 패킷 C에 해당하는 암호화된 패킷을 ICMP로 전송하여 Request와 Reply 간의 시간 지연을 측정하여 비교하였다.

회차	1회	2회	3회	4회	5회	6회	평균
평문 패킷	1.184	0.94	0.947	1.04	1.061	0.916	1.015
암호화 패킷	1.317	1.301	1.144	0.986	1.191	0.909	1.141

(표 1) 빈 패킷과 암호화된 패킷의 응답시간(ms) 비교

암호화된 패킷은 평문 패킷과 비교하여 평균적으로 0.126ms의 응답시간 지연이 발생하였다. 이는 네트워크에서 용인 가능한 시간 차이이며, 일부 시도에서는 암호화 패킷의 응답시간이 더 짧은 경우도 나타나 해당 방법이 패킷 전송 시간 지연에 미치는 영향이 크지 않음을 알 수 있다.

5. 결론

본 논문에서는 IoT 환경에서 CoAP의 보안을 강화하기 위해 솔팅을 이용한 해시 함수와 스테가노그래피를 활용하는 접근법을 제안하였다. 이 방법은 CoAP 메시지의 무결성을 검증하고, 스테가노그래피를 통해 보안 정보를 은닉하여 추가적인 보안성을 제공한다. 향후 연구에서는 성능과 실용성을 면밀히 평가하여 기존의 검증 방식과 비교하고, 스테가노그래피가 탐지되었을 때 발생하는 문제점에 대한 연구를 수행할 계획이다.

ACKNOWLEDGEMENT

본 연구는 한국인터넷진흥원(KISA)-정보보안 특성화대학 지원사업의 지원을 받아 수행된 연구임. 본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 첨단분야 혁신융합대학사업의 연구결과입니다.

참고문헌

[1] L. Canuto, L. Santos, L. Vieira, R. Gonçalves and C. Rabadão, "CoAP Flow Signatures for the Internet of Things," 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1-6,

[2] P. Gauravaram, "Security Analysis of salt||password Hashes," 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, Malaysia, 2012, pp. 25-30,

[3] M. Ivasenko, O. Suprun and O. Suprun, "Information Transmission Protection Using Linguistic Steganography With Arithmetic Encoding And Decoding Approach," 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2021, pp. 174-178,

[4]Punam Bedi, Arti Dua, Network Steganography using the Overflow Field of Timestamp Option in an IPv4 Packet, Procedia Computer Science, Volume 171, 2020, Pages 18 10-1818,