# 경량 암호화 통신을 위한 이중암호화 기법

배희경 [1], 심혜연 [2], 이일구 [3]
[1] 성신여자대학교 융합보안공학과 학부생
[2] 성신여자대학교 미래융합기술공학과 박사과정
[3] 성신여자대학교 융합보안공학과 교수

20221098@sungshin.ac.kr, 220237062@sungshin.ac.kr, iglee@sungshin.ac.kr

# Dual encryption technique for lightweight encryption communication

Heegyung Bae[1], Hye Yeon Shim[2], Il-Gu Lee[1, 2]
[1]Dept. of Convergence Security Engineering, Sungshin Women's University
[2]Dept. of Future Convergence Technology Engineering, Sungshin Women's University

**요 약**

IoT(Internet of Things) 기기를 대상으로 하는 보안 위협이 증가하면서 IoT 정보의 기밀성 유지가 중요한 과제로 떠오르고 있다. 따라서 경량, 저가, 저전력 IoT 환경에서 높은 보안 수준을 유지할 수 있는 암호화 방법이 필요하다. 본 연구에서 AES(Advanced Encryption Standard)와 SAES(Simplified AES)를 이용한 이중 암호화 기법을 제안한다. 제안하는 기법은 SAES 로 평문 메시지 전체를 블록 단위로 암호화하고, 각 암호문 블록의 일부 비트를 추출해 AES 로 재암호화한다. 실험 결과에 따르면, 제안한 경량 이중 암호화 기법이 종래 방식보다 암호문의 크기를 32% 줄일 수 있었다.

## 1. 서론

IoT(Internet of Things) 기기의 수가 증가하면서, IoT 에서 송수신되는 데이터가 정보 유출 공격에 노출되고 있다. 이러한 상황에서 데이터의 기밀성을 보장하기 위한 암호화가 필수로 여겨진다[1]. 하지만, 기존의 복잡한 암호화 알고리즘을 경량, 저가, 저전력 요건을 만족해야 하는 IoT 에 적용하기 어렵다. [2] 따라서 적은 자원으로도 기밀성을 보장할 수 있는 새로운 암호 기법의 필요성이 대두되고 있다[3]. 본 논문에서는 SAES(Simplified Advanced Encryption Standard)로 암호화 후 일부를 AES 로 암호화해 적은 자원으로 강력한 보안을 제공하는 이중 암호화 방법을 제안한다.

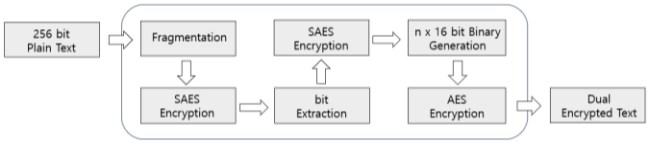
## 2. 관련 연구

Yusuf[4]는 RSA(Rivest Shamir-Adleman)를 이용해 파일을 암호화한 후 RC4(Rivest Cipher 4)를 이용해 재암호화하는 방식을 제안했다. 제안한 방법은 종래 기법보다 암호 키를 추측하기 어려워 높은 기밀성을 보장할 수 있으나, 파일 크기가 커진다는 문제점이 있다.

Jaspin[5]은 클라우드에서 AES 를 통해 암호화를 수행하고 RSA 로 재암호화를 한 후 업로드하는 기법을 제안했다. 제안한 방법은 종래 보다 짧은 암복호화 시간과 높은 처리량을 보였다. 그러나 자원이 풍부한 클라우드 서버 환경에 적합하기 때문에 IoT 기기에서 사용하기 부적절하다.

## 3. 경량 이중 암호화 기법

본 논문에서는 적은 자원을 사용하면서 보안을 강화하기 위해 SAES 와 AES 를 활용한 이중 암호화 방법을 제안한다. SAES 는 AES 의 동작을 간소화한 알고리즘으로 16 비트 키로 16 비트 평문을 암호화한다[6].




(그림 1) 제안하는 암호화 흐름도.

그림 1 은 제안하는 기법의 암호화 흐름도이다. 우선, 256 비트 메시지를 16 비트 크기의 블록으로 분할한다. 분할된 메시지 블록을 SAES 로 암호화한 후 SAES 키로 seed 를 설정해 암호문의 각 블록에서 무

작위로 n 개의 비트를 추출한다. 추출한 비트로 n×16 크기의 바이너리를 구성하고 해당 바이너리를 AES 를 이용해 재암호화한다. 이중 암호화를 수행한 암호문과 SAES 암호문을 병합해 최종 암호문을 생성한다.

복호화는 암호화 과정의 역으로 진행된다. 복호화하여 획득한 n×16 비트의 바이너리를 한 비트 단위로 분리한 후 SAES 키를 이용해 seed 를 설정하여 원래 자리에 삽입한다. 16 비트로 재구성된 각 블록을 SAES 키를 이용해 복호화 하여 평문을 획득한다.

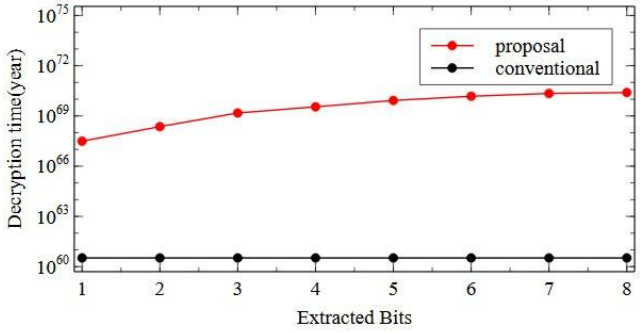
## 4. 실험

본 실험의 SAES 는 [6]의 개념을 Python 으로 구축한 라이브러리를 통해 구현했다[7]. 실험에서는 제안하는 기법과 종래 기법 성능을 비교한다. 평가에 사용되는 성능 지표는 암호문 크기, 메모리 사용량, 추출하는 비트 수에 따른 보안 강도이다.

<표 1> 종래 기법과 제안한 이중 암호화 기법의 성능 비교

| Category | Conventional | Proposal |
|---|---|---|
| Encrypted Text Size(Byte) | 88 | 60 |
| Memory Usage(MB) | 69.59375 | 69.68359 |

표 1 은 256 비트 크기의 평문을 암호화했을 때 제안하는 기법과 종래 기법의 성능을 비교한 표이다. 종래 방식의 암호문 크기가 88byte, 제안 기법의 암호문 크기는 60byte 로 종래의 방법보다 약 32% 작은 암호문을 생성했다. 메모리 사용량은 두 기법 모두 약 70MB 로 큰 차이가 나타나지 않았다.




(그림 2) 추출 비트 수에 따른 복호화 시간.

그림 2 는 추출 비트 수에 따른 복호화 시간이다. SAES 암호문 블록에서 추출하는 비트의 개수에 따라 해독하는 데 필요한 연산량을 시간으로 나타냈다. 추출하는 비트의 수가 n 개일 때, 메시지 전체를 해독하는 데 필요한 연산량은 $C(16,n) \times 2^{256} \times 2^{16}$ 이다. 1 초에 $10^8$번 연산이 가능한 컴퓨터를 사용한다고 가정하고 복호화 시간을 계산한 결과, 추출 비트 수가 늘어날수록 복호화 시간이 증가했다. 추출하는 비트가 8 개 일 때, 메시지 전체를 해독하는 데 소요되는 시간은 $2.5 \times 10^{70}$년으로 높은 기밀성을 보장한다.

## 5. 결론

본 연구에서는 SAES 와 AES 를 결합해 메시지를 이중 암호화하는 기법을 제안하고, 종래의 방법과 비교했다. 실험 결과에 따르면, 제안 기법 사용은 종래 방식 대비 암호문의 크기가 약 32% 감소했다. 더 나아가서, 입력 메시지의 길이의 증가해서 블록 개수가 많아지면 암호문의 크기가 더 큰 폭으로 축소됨을 보였다. 향후 연구에서는 제안한 방법을 실제 환경에서 검증하기 위한 테스트베드를 구현하고 다양한 환경 조건에서 테스트할 계획이다.

**Acknowledgement**

본 논문은 2024 년도 산업통상자원부 및 한국산업기술진흥원의 산업혁신인재성장지원사업 (RS-2024-00415520)과 과학기술정보통신부 및 정보통신기획평가원의 ICT 혁신인재 4.0 사업의 연구결과로 수행되었음 (No. IITP-2022-RS-2022-00156310)

## 참고문헌

[1] Kholoud Y. Najmi, Mohammed A. AlZain, Mehedi Masud, N.Z. Jhanjhi, Jehad Al-Amri and Mohammed Baz, "A survey on security threats and countermeasures in IoT to achieve users confidentiality and reliability," Materials Today: Proceedings, volume 81, part2, 377-382, 2023.

[2] Maitra and Sudip, "Performance evaluation of IoT encryption algorithms: memory, timing, and energy," 2019 IEEE sensors applications symposium (SAS), Sophia Antipolis, France, 2019, pp. 1-6.

[3] A. Tripathy, and B. Singh, "A Study of AES Software Implementation for IoT Systems," 2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2022, pp. 1-4.

[4] D. M. Yusuf, D. R. I. M. Setiadi, E. H. Rachmawanto, C. A. Sari, and R. R. Ali, "Dual Encryption Method for File Security," 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, 2019, pp. 222-227.

[5] K. Jaspin, S. Selvan, S. Sahana, and G. Thanmai, "Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA Algorithm," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2021, pp. 791-796

[6] Musa, Mohammad A., Edward F. Schaefer, and Stephen Wedig. “A SIMPLIFIED AES ALGORITHM AND ITS LINEAR AND DIFFERENTIAL CRYPTANALYSES.” Cryptologia 27, no. 2, 148–77, 2003.

[7] Mayank-02, Simplified-AES, Github, https://github.com/mayank-02/simplified-aes#references, 2020.