WAVE 하드웨어 암호 라이브러리에 적합한 효율적인 AES-CCM 구조 설계

이연철¹ · 서화정² · 김호원^{3*}

The Efficient AES-CCM Architecture for a hardware library in the WAVE

Yeon-cheol Lee¹ · Hwa-jeong Seo² · Ho-won Kim^{3*}

¹School of Computer science & Engineering, Pusan National University, Pusan 609-735, Korea

²School of Computer science & Engineering, Pusan National University, Pusan 609-735, Korea

³School of Computer science & Engineering, Pusan National University, Pusan 609-735, Korea

요 약

차량에서의 무선통신의 발달과 함께 WAVE(Wireless access in vehicular environments) 상에서도 보안에 대한 위험성이 증가하였다. 이를 위해 IEEE 1609.2로 스누핑, 도청과 같은 공격으로부터 메시지를 보호하기 위한 보안 서비스를 규정하였다. 이를 보안 라이브러리를 하드웨어 형태로 구현 가능하며 본 논문에서는 이에 적합한 AES-CCM 구조를 설계하였다. 동일 FPGA에서 비교 논문의 구조와 비교하여 27 % 적은 slice를 사용하였으며 기존 라이브러리 모듈에서 레지스터를 공유하였을 경우를 고려하면 약 45 % 적은 slice를 사용한다. 또한 xc5vlx110t-2ff1136 상에서 1355 Gbits/s의 처리량을 보인다.

ABSTRACT

According to developing wireless communications in vehicle, various security threat in the WAVE(Wireless access in vehicular environments) is increased. To protect this, IEEE 1609.2 specify services as for prevent message from attacks such as spoofing, eavesdropping and replay. It is possible to implement a hardware library for defending these attacks. In this paper, we proposed a efficient AES-CCM architecture for the hardware library in the WAVE. We compare our architecture to the previous one in the same FPGA. And our design uses less slices than 27 % of it and less slices than 45 % of it if we share registers that were used by other modules in the library. We also achieves a throughput of 1355 Gbits/s in xc5vlx110t-2ff1136.

키워드 : AES-CCM, 하드웨어, FPGA, WAVE, IEEE1609.2

Key word : AES-CCM, hardware, FPGA, WAVE, IEEE1609.2

접수일자 : 2013. 08. 17 심사완료일자 : 2013. 09. 04 게재확정일자 : 2013. 09. 17

* **Corresponding Author** Ho-won Kim(E-mail:howonkim@pusan.ac.kr, Tel:+82-51-510-1010)

School of Computer science & Engineering, Pusan National University, Pusan 609-735, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2013.17.12.2899>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

이동 통신 기술이 보급된 이래로 무선 통신 활용은 급격히 증가하였으며 다양한 무선 통신 기술들이 활용되고 있다. 특히 이동 중 운송 수단에서의 무선 통신 사용자의 수 또한 계속적으로 증가하였고 이에 대한 기술들도 계속적으로 보급되고 있는 상황이다.

이에 따라 ITS(Intelligent Transportation System) 환경에서 운송 수단들 간의 통신이 가능하도록 WAVE가 제안되었다. WAVE [1]에는 차량과 차량(V2V) 뿐만 아니라 차량과 인프라(V2I)에서의 무선 통신이 가능하도록 이를 IEEE 1609 표준으로 정의하고 있다.

또한 통신량이 급격하게 증가하는 것에 따라 도청, 스누핑, 패킷의 재사용 등과 같은 공격들도 비약적으로 증가하였다. 이를 위해 IEEE 1609.2에서는 통신상에서 발생하는 공격에 대해 방어할 수 있는 보안 서비스들도 포함하고 있으며 IEEE 1609.2에서 사용되는 기법들을 정리하면 다음과 같다. 공개키 기반 전자 서명으로 ECDSA, ECIES 등을 사용하며 타원곡선암호(elliptic curve cryptography)에 기반하고 있기 때문에 기존에 사용하던 암호화 기법이나 인증 기법보다 좀더 보안성이 우수하다. 또한 데이터 암호화 및 복호화를 수행하기 위해 AES-CCM을 사용하며 이를 사용하였을 경우 기밀성(Confidentiality)을 유지할 수 있는 것뿐만 아니라 MIC 값을 추가적으로 생성하기 때문에 무결성(Integrity)도 보장되는 장점을 가진다.

이러한 암호 기법들을 하드웨어 라이브러리 형태로 구현하였을 경우 ECDSA에서 사용되는 224비트와 256비트 필드상의 타원곡선암호 모듈은 상당한 하드웨어 면적을 차지한다. 따라서 전자서명 이외의 암호화 모듈을 추가하기 위해서는 면적을 고려한 효율적인 구조가 필요하며 본 논문에서는 AES-CCM에 대한 효율적 구조에 대해 제안한다.

이를 설명하기 위한 논문 구성은 다음과 같다. II장에서는 IEEE 1609.2 및 제안하는 논문에서 사용하는 기술인 AES core 및 AES-CCM에 대해서 설명한다. III장에서는 비교 대상이 되는 논문의 구조를 설명한다. V장에서는 제안하는 AES-CCM 구조에 대해서 설명하며 이를 VI장의 구조와 비교한다. 마지막으로 VII장에서 결론을 맺는다.

II. 관련 기술

2.1. 차량 통신 암호 - IEEE 1609.2

IEEE 1609.2는 메시지 암호화를 통한 기밀성 보장, 알려진 사용자에게 대한 인증, 익명의 사용자에게 대한 브로드캐스트를 위하여 WAVE 상에서 차량 통신 암호를 정의한다.[1] 이를 위한 암호화 연산들을 표 1과 같이 정리하였다.

표 1. WAVE에서 사용되는 암호화 연산

Table. 1 Cryptographic operation in wave

암호화 연산	설명
ECDSA	Signature algorithms
ECIES	Public key encryption algorithms
AES-CCM	Symmetric algorithms
SHA-256	Hash algorithms

ECDSA의 경우 전자 서명 알고리즘으로 NIST SEC2에 명시된 P-224 및 P-256 타원 곡선을 사용한다. 공개키를 사용한 암호화 알고리즘으로 ECIES를 사용하며 P-256를 사용한다. 암호문을 생성하기 위하여 내부에서 SHA-256을 사용하며, 암호화 결과로 공개키 값, 암호문, 태그 값을 전달한다. 따라서 단순히 기밀성을 보장하는 것 뿐 만 아니라 태그 값을 통해서 무결성도 보장한다. 대칭 암호화 알고리즘으로 AES-CCM을 사용하며 NIST에서 정의한 파라미터 중 일부를 제한하고 있으며 VI장에서 이러한 제한 사항에 대해 언급한다. 또한 제안하는 알고리즘이기 때문에 2.3 절에서 자세하게 설명한다. 해쉬 알고리즘으로는 SHA-256을 사용하며 ECIES 및 ECDSA에서 초기 키 쌍을 생성할 때 Random Number Generator로 사용된다. 또한 인증을 위해 사용된다.

2.2. AES core

NIST에서는 DES를 대체하기 위한 알고리즘을 공모하였고 2000년 Rijndael이 AES 알고리즘으로 선정되었다. 128 비트 키를 사용하여 128 비트 크기의 블록을 암호화한다. 128비트의 키를 사용할 경우 10 라운드 과정을 거치게 되며 라운드 과정에는 다음과 같은 부분 연산을 수행한다.[2]

ByteSub : S-box에 의한 바이트 단위의 치환.
 MixColumn : 행 단위의 matrix 연산 수행.
 ShiftRow : 4 offset에 따른 행 단위 shift 연산.
 Key addition : 각각의 라운드 과정에서 생성되는 key 값과의 XOR 연산.

첫 번째 라운드에서는 Key addition 과정만을 수행하며 마지막 라운드의 경우 key addition을 제외하고 나머지 과정들만을 수행한다.

다음으로 구현된 AES core와 비교 논문에 대해서 간략히 요약하면 표 2과 같다. 실제 제안되는 구조의 AES 모듈이 더 많은 Slice 크기를 차지하지만 구현된 전체 AES-CCM 구조에서는 더 적은 면적을 차지하도록 구현하였다.

표 2. xc3s4000-4fg900에서 AES core
 Table. 2 AES core in xc3s4000-4fg900

Author	Slice	Frequency (MHz)
target Design [2]	633	100.08
Our Design	907	99.162

2.3. AES-CCM

AES-CCM은 NIST에서 제시하는 암호화 모드 중 하나이며 암호화 및 복호화 뿐 만 아니라 MIC 값을 생성하는 특징을 가진다[4][5]. 그림 1은 AES-CCM의 기본적인 구조를 잘 나타내고 있는 그림이다 [6][7].

그림 1. AES-CCM 기본 구조
 Fig. 1 the basic AES-CCM architecture

그림에서와 같이 CBC 모드와 CTR 모드를 조합하여 구성된다. 암호화 및 MIC 생성의 경우 실선으로 표현하였으며, 복호화 및 인증 과정의 경우는 점선으로 표시한다. CBC 모드에서는 채인 형태로 AES 결과 값과 B_i 값이 재사용되며 최종 생성된 CBC 모드의 결과 E 가 MIC 값을 나타낸다.

구조를 설명하기 위하여 이를 알고리즘으로 정리하였으며 또한 이는 V장 제안하는 구조에서의 알고리즘과 비교를 위하여 사용된다.

Algorithm1

steps:

1. set $B_o, \dots, B_r$ using (N, A, P) .
2. $T = AES_k(B_o)$.
3. for $i = 1$ to r do.
- 3.1. $T = AES_k(B_i \oplus T)$.
- end.
4. set $Ctrl_0, \dots, Ctrl_r$.
5. for $i = 1$ to r do.
- 5.1. $S_i = AES_k(Ctrl_i)$.
- end.
6. $S = S_1 \| S_2 \| \dots \| S_r$.
7. return $C = (P \oplus S) \| (T \oplus S_o)$.

Algorithm1의 경우 암호화 및 MIC 생성이며, Algorithm2의 경우 복호화 및 인증 과정이다. 초기 $B_o, \dots, B_r$ 블록을 생성하기 위하여 Nonce 값을 나타내는 N , Associated data A , 평문 P 이 사용된다. 이 때, WAVE 상에서는 A 값을 사용하지 않는 것이 특징이며 이외의 제한 사항의 경우는 V장에서 설명한다. 또한 $Ctrl_i$ 를 구성하기 위해서 N 이 사용되며 N 값 뒤에 카운터 값을 추가된 형태로 $Ctrl_i$ 블록이 구성된다. Algorithm1과 Algorithm2에서 사용하는 $AES_k(\cdot)$ 의 경우 II장에서 설명한 AES의 암호화 모듈을 수행하게 된다.

Algorithm2

steps:

1. If $\mathcal{C}len \leq \mathcal{T}len$ Then return INVALID.
2. set $\mathcal{C}tr_0, \dots, \mathcal{C}tr_r$.
3. for $i = 1$ to r do.
- 3.1. $S_i = AES_k(\mathcal{C}tr_i)$.
- end.
4. $P = C \oplus (S_1 \| S_2 \| \dots \| S_r)$.
5. $T' = C \oplus S_0$.
6. set $B_o, \dots, B_r$ using (N, A, P) .
7. $T = AES_k(B_0)$.
8. for $i = 1$ to r do.
- 8.1. $T = AES_k(B_i \oplus T)$.
- end.
9. If $T \neq T'$
Then return INVALID , else return P .

III. Target 알고리즘

[3]에서는 CBC 및 Counter 모드 각각에 AES 모듈을 적용한 형태로 AES-CCM을 구현하였으며 그림 2와 같은 구조를 가진다. Authentication 모듈에서는 CBC-MAC을 생성하기 위해 AES 모듈을 포함하며 Encryption 모듈에서는 Counter 모드 수행을 위한 AES 모듈이 포함된다. 중앙에 위치한 Control Unit을 통해서 값을 각각의 모듈을 컨트롤하며 실제 Authentication 및 Encryption 모듈 내부에 각각의 모듈을 컨트롤하는 C.U.를 다시 포함하는 구조이다. CBC 및 Counter가 동시에 수행되며 생성된 값은 아래 MUX를 통해서 출력 되도록 하였다. 그림2에서는 실제 Block을 구성하는 $B_0, \dots, B_r$ 을 저장하기 위한 RAM 구조는 포함하고 있지 않다.

[3]에서 제안된 구조의 경우 2개의 AES 모듈을 사용하고 있기 때문에 실제 WAVE 상에서 사용하기 위한 하드웨어 라이브러리를 구현하기에는 상대적으로 큰 면적을 차지한다. 따라서 다음 장에서는 1개의 AES 모듈을 사용하여 AES-CCM 구조를 설명한다.

그림 2. [3]에서 제안하는 AES-CCM 설계 구조
Fig. 2 the AES-CCM hardware architecture by [3]

IV. 구현 내용

제안하는 AES-CCM 구조에서는 하드웨어 비용을 최소화하기 위하여 AES 모듈을 하나만 사용하도록 구현되었다. 또한 RAM에 접근하는 read/write 시간을 줄이고 중간 값을 저장하기 위한 레지스터 및 MUX 비용을 최소화 되도록 설계되었다. 본 절에서는 이에 대해 설명한다.

WAVE 상에서 기본적으로 NIST에서 정의한 파라미터에 따라 AES-CCM 구조를 사용하도록 하고 있으며 일부 파라미터 설정을 아래와 같이 제한한다.

- Associated data를 사용하지 않으며, 따라서 B_0 를 구성하는 Adata = 0이 된다.
- 따라서 $B_1, \dots, B_r$ 까지 평문으로 구성된다.
- 128 비트 MIC 값을 사용.
- 96 비트의 Nonce 값(12 octets).

이러한 특성을 적용하여 AES-CCM 하드웨어 구조를 설계하였고 그림3 와 같다. 초기 설정을 위하여 B0 레지스터 일부분에 96비트 Nonce 값과 평문의 길이를 나타내는 Q값을 저장한다. 또한 key 레지스터에 128비

트 AES를 위한 키 값을 저장하고 RAM에 평문을 저장한다.

그림 3. 제안하는 AES-CCM 설계 구조
Fig. 3 our AES-CCM hardware architecture

32비트 폭의 RAM을 사용하였기 때문에 AES 모듈에 입력으로 사용할 값을 저장하기 위해 4 클럭이 소모되는데 이 클럭 사이클을 줄이고 중간 값을 저장하기 위해 128비트 B, Y, T 레지스터를 구성하였다.

기존 Algorithm1에서는 CBC 모드를 수행하는 Step 3.1.에서 RAM에서 B_i 값을 읽어오며 Step 7에서 암호문을 생성하기 위해서 P 값을 RAM에서 읽어오게 된다. 이때 소모되는 클럭을 줄이기 위하여 Algorithm1의 암호화 및 MIC 생성 과정을 Algorithm3으로 변경한다. Algorithm3에서는 step 2.2에서 B_i 값을 RAM에서 읽어 오며 그 값을 레지스터에 저장하여 이후 step 2.3까지 사용가능하도록 하였다. 또한 AES 모듈이 수행되는 12 클럭 동안 RAM read/write가 발생하도록 하여 실제 AES 암호화 과정이 끝났을 때는 다음 B_i 값이나 Ctr_i 값이 AES 입력으로 미리 저장되어 있도록 하여 추가적인 RAM 접근 시간이 필요하지 않도록 하였다.

이에 따라 제안된 구조에서는 AES 연산 후 중간 값들을 저장하기 위하여 추가적인 레지스터 및 MUX가 추가되어야 하며 이를 최소화하기 위하여 Algorithm3의 steps 2.1.- steps 2.3. 에서와 같이 Counter 모드가 먼저 수행되도록 하였다. 실제로 초기 설계에서 사용되었던 MUX 입력을 상당히 줄일 수 있었고 [3]보다 적은

slice를 사용하여 설계되었다. 또한 Counter 모드가 먼저 수행됨으로써 Counter 모드가 수행되는 동안 RAM에서 CBC 모드에서 사용할 B_0 값을 읽어 올 수 있는 장점을 가진다.

Algorithm3

steps:

1. set B_0 and $T = AES_k(B_0)$.
2. for $i = 1$ To r Do.
 - 2.1. set Ctr_i and $S_i = AES_k(Ctr_i)$.
 - 2.2. $C_{i-1} = B_i \oplus S_i$.
 - 2.3. $T = AES_k(B_i \oplus T)$.
- end.
3. set Ctr_0 and $S_0 = AES_k(Ctr_0)$.
4. return $C' = C_0 \| C_1 \| \dots \| C_r \| T$.

Algorithm4

steps:

1. If $Clen \leq Tlen$ Then return INVALID.
2. set B_0 and $T = AES_k(B_0)$.
3. for $i = 1$ to r do
 - 3.1. set Ctr_i and $S_i = AES_k(Ctr_i)$.
 - 3.2. $P_i = C_i \oplus S_i$.
 - 3.3. $T = AES_k(B_i \oplus T)$.
- end.
4. set Ctr_0 and $S_0 = AES_k(Ctr_0)$.
5. $T = C_0 \oplus S_0$
6. If $T \neq T'$ Then

return INVALID , else

return $P = P_0 \| P_1 \| \dots \| P_r$

Algorithm2 복호화 및 인증 과정도 Algorithm4와 같이 변경하였다. 동일한 기법을 적용하여 RAM의 read/write에 소모되는 클럭을 AES 모듈 수행시간에 숨기도록 하였고 이를 위하여 Counter 모드가 먼저 수행 되도록 하였다.

V. 비교 및 성능 평가

제안된 AES-CCM 구조를 가지고 이를 FPGA 상에서 합성하였다. 표 3은 그 결과를 나타낸다. spartan3, spartan6, vertex5 등 다양한 FPGA 상에서 시뮬레이션을 수행하였다. spartan3의 경우 [3]와 비교를 위하여 합성하였으며 spartan6, vertex5의 경우 WAVE 암호 라이브러리를 적재하기 적합한 FPGA를 찾기 위한 것이다. 보드에 따라 사용한 slice가 약간 차이가 나는 것을 확인할 수 있고 동작주파수의 경우 vertex5에서 가장 높은 결과를 얻을 수 있었다.

[3]와 비교하기 위해 동일한 spartan3 중 c3s4000-4fg900로 합성하였고 결과를 표 4에 정리 하였다. 제안하는 구조의 경우 1,587 slice를 사용하였으며 [3]의 2,154 slice와 비교하여 약 27% slice 감소를 이루었고 거의 동일한 최대 동작 주파수로 동작한다.

표 3. Device에 따른 Slice, BRAMs, Frequency 차이
Table. 3 Differences of Slice, BRAMS, and Frequency in Devices.

Device	Slice	BRAMs	Frequency (MHz)
xc3s4000-4fg900	1,587	11	99.162
sc6slx9-2csg324	1,447	12	194.900
xc5vlx110t-2ff1136	1,434	11	258.081

표 4. spartan3 에서 제안된 design과 design [3] 비교
Table. 4 compare our design to [3] in spartan3

Author	Slice	BRAMs	Frequency (MHz)
target Design [3]	2,154	106	100.08
Our Design	1,587	11	99.162
Ours [-Reg]	1,203	11	N/A

표 4의 Ours [-Reg]의 경우는 ECDSA와 같은 상대적으로 큰 규모의 암호화 모듈에 AES-CCM 구조를 추가하였을 경우를 고려한 것이다. 제안하는 구조에서는 이를 고려하여 설계되었으며 라이브러리 형태로 함께 합성되는 다른 모듈의 레지스터를 공유하는 형태의 구현이 가능하다. 이 때 제안된 AES-CCM 구조에서 사용하는 레지스터를 고려한 비용을 Ours [-Reg]에 계산하였다. 이는 Our Design에서 레지스터로 사용되는 slice 크

기를 제외시킨 수치이며, 이 때 약 45 % slice 감소를 기대할 수 있다.

표 5는 합성한 FPGA 보드상에서의 처리량을 계산하였다. 계산 방법은 [3]에서와 동일한 방법을 사용하였으며 식(1)과 같다. 1024 bytes 처리하는 것을 기준으로 하였으며 AES 모듈을 2개 사용하였기 때문에 참고 논문의 x2 클럭을 소모한다고 가정한다.

$$Throughput = \frac{66 * 128 \text{ bits}}{1608 * frequency} \quad (1)$$

spartan 3에서는 520 Gbits/sec가 나왔지만 spartan6 나 vertex5의 경우에는 1000 Gbits/sec 이상의 처리량이 나오는 것을 확인 할 수 있으며 WAVE 암호라이브러리로 사용하기에 적합해 보이며, 이는 [3]보다 비슷하거나 높은 처리량이다.

표 5. 제안된 디자인과 [3] 간의 처리량 비교
Table. 5 compare our throughput to [3]

[3] Design	
Device	Throughput (Mbps)
xc3s400-4fg900	1067
Our Design	
Device	Throughput (Mbps)
xc3s400-4fg900	520
sc6slx16-3csg324	1023
xc5vlx110t-2ff1136	1355

VI. 결 론

본 논문에서는 WAVE 상의 하드웨어 라이브러리를 구성 할 때의 적합한 AES-CCM 구조를 설계하였다. AES-CCM 구조에서 경량화를 고려하여 설계하였으며 기존 라이브러리를 재사용 측면을 고려하여 설계되었다. 또한 메모리 read/write를 AES 모듈 동작 시간 내에서 이루어지도록 설계하였다.

제안하는 구조는 비교되는 논문에 비해 약 27 %의 slice 감소를 이루었으며, 또한 기존 라이브러리의 레지스터를 재사용하는 관점에서는 약 45 %의 slice 감소를 이룬다. vertex5에서 1355 Gbits/sec 처리량을 보인다.

감사의 글

이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.

REFERENCES

- [1] IEEE Std. 1609.2, IEEE Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages, IEEE, New York, N.Y., 2013.
- [2] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." In First Advanced Encryption Standard (AES) Conference. 1998.
- [3] López-Trejo, Emmanuel, F. Rodriguez Henriquez, and Arturo Díaz-Pérez. "An Efficient FPGA Implementation of CCM mode using AES." International Conference on Information Security and Cryptology. Vol. 3935. 2005.
- [4] Whiting, Doug, Niels Ferguson, and Russell Housley. "Counter with cbc-mac (ccm)." (2003).
- [5] Dworkin, Morris J. "SP 800-38C. Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality." (2004).
- [6] Huai, Lian, et al. "An energy-efficient AES-CCM implementation for IEEE802. 15.4 wireless sensor networks." Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC'09. International Conference on. Vol. 2. IEEE, 2009.
- [7] Samiah, Abdul, Arshad Aziz, and Nassar Ikram. "An Efficient Software Implementation of AES-CCM for IEEE 802.11 i Wireless St." Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International. Vol. 2. IEEE, 2007.

이연철(Yeon-Cheol Lee)

2007.3~2013.2 : 부산대학교 정보컴퓨터공학과 학사
2013.3~현재 : 부산대학교 컴퓨터공학부 석사
※관심분야 : IoT, Data Mining, 정보보안, PUF, VLSI 설계

서화정(Hwa-Jeong Seo)

2004.3~2010.2 : 부산대학교 정보컴퓨터공학과 학사
2010.3~2012.2 : 부산대학교 컴퓨터공학부 석사
2012.3~현재 : 부산대학교 컴퓨터공학부 박사
※관심분야 : IoT, 정보보안, RFID/USN, 암호 이론, VLSI 설계

김호원(Ho-Won Kim)

1989. 3~1993. 2 : 경북대학교 전자공학과 학사
1993. 3~1995. 2 : 포항공과대학교 전자전기공학과 공학석사
1995. 2~1999. 2 : 포항공과대학교 전자전기공학과 공학박사
1998. 12~2008. 2 : 한국전자통신연구원(ETRI) 정보보호연구단 선임연구원 / 팀장
2008. 3~현재 : 부산대학교 정보컴퓨터공학부 부교수
※관심분야 : IoT, 스마트그리드 보안, RFID/USN 정보보호 기술, PKC 암호, VLSI 설계, embedded system 보안