논문 2009-6-10

대형 자료를 위한 AES 확장에 관한 연구

A Study on AES Extension for Large-Scale Data

오주영*, 고훈준*

Ju-Young Oh, Hoon-Joon Kouh

요 약 모든 정보기술 분야에서 비인가자의 편취나 의도적 변경 등으로부터 정보를 보호하는 것은 핵심적인 문제가 되었다. 이에 안전한 작업진행을 위해 효과적이고 편의한 보안방법들이 요구되는데, 암호화 알고리즘은 많은 연산시간을 요하며, 실제 CPU 시간과 메모리 등의 많은 시스템 자원을 소모한다. 본 논문에서는 대형 자료의 암호화를 위해 평문의 압축, 입력 블록의 가변 크기, 라운드 횟수의 사용자 설정, 소프트웨어 최적화 등의 4가지 특징을 고려한 AES 확장 구조를 제안하였다. 실험은 C++로 수행하였으며 암호화와 복호화에서 개선된 실행시간을 보인다.

Abstract In the whole information technology area, the protection of information from hacking or tapping becomes a very serious issue. Therefore, the more effective, convenient and secure methods are required to make the safe operation. Encryption algorithms are known to be computationally intensive. They consume a significant amount of computing resources such as CPU time and memory. In this paper we propose the scalable encryption scheme with four criteria, the compression of plaintext, variable size of block, selectable round and software optimization. We have tested our scheme by c++. Experimental results show that our scheme achieves the faster execution speed of encryption/decryption.

Key Words : Encryption, Compression, Optimization

Ⅰ. 서 론

유/무선 통신 기반 기술을 활용하는 자료 등은 전송시에 비인가자의 의도적 접근에 의한 편취나 정보변경을 차단해야하며, 한정된 통신선로에 대해 시간제약내에 자료가 송달되어야 하므로 통신선로와 전송 자료의 크기 및 정보의 표현과 응용에 적합한 수준의 보안절차를 요구하게 된다[1]. 특히, 기밀이 요구되는 정보의 크기가 대형 자료인 경우 실시간 자료전송을 위해서는 유한의 전송대역내에서 자료를 전달하기 위해 암호화의 전단에서 자료전송량을 배가할 수 있는 추가 절차가 요구된다. 디지털 정보 전송에 대한 기존의 연구는 DES, 3DES, IDEA, AES 등의 암호화 기법을 벤치마킹하여 암호화의 소재별 성능이나 암호화 자료의 안전성에 기초한 연구가 진행되고 있다[2]. 특히, 암호화에 수반되는 실행 시간과 자원사용량을 무선 환경의 이동 단말기에 적합한 수준으로 최소화하기 위한 다양한 하드웨어 구현 방법과 성능 분석에 대해 연구되고 있다[3]-[4].

본 논문에서는 크기가 크고 구성 내용이 복합적인 자료 전송의 경우에 대해 소프트웨어 구현방식에서 암호화를 빠르게 수행하기 위해 블록 암호화 기법인 AES 표준에 기초하여 입력 평문의 크기에 따라 알고리즘의 주요 변수를 확장성 있게 선택적으로 적용가능하게 하였다. 일반적으로, 블록 암호화의 구현상의 복잡도는 블록의 크기에 비례하여 커지므로 너무 크게 설정하면 효율성이 떨어지게 되고, 너무 작게 설정하면 여러 유형의 암호 공격으로부터 안전할 수 없게 된다[5]-[6]. 뿐만 아니라, 암호문의 견고성은 라운드 횟수를 늘리고 입력키를 크게 설

*정회원, 경인여자대학 정보미디어학부
접수일자 2009.10.11, 수정일자 2009.11.21

정함으로써 향상시킬 수 있다[7]. 본 논문에서는 암호화에 적용되는 주요 변수들을 선택적으로 설정할 수 있도록 하여 구현상의 효율성과 보안 공격에 대한 견고성을 유지할 수 있는 범위 내에서 암호화 시간을 줄일 수 있도록 하였다. 특히, 암호화 실행의 주요 변수가 되는 평문의 입력크기와, 라운드 횟수, 키 크기 등을 선택적으로 적용할 수 있도록 하였으며, 입력 블록의 개수만큼 반복되는 알고리즘의 실행시간을 최소화하고 동시에 전송 효율을 높일 수 있도록 구조가 간단하고 효율이 높은 허프만 압축 절차를 입력 전단 모듈에서 선택할 수 있도록 확장하였다. 이를 통해 입력정보를 압축하여 압축된 정보를 블록으로 설정함으로써 입력 블록의 개수를 줄였다.

실험은 일반 문자조합으로 구성된 단순 텍스트 파일과, 특수문자와 도형 및 그래프로 구성된 혼합파일에 대해 일반 파일과 압축시간을 포함하는 압축평문을 암호화/복호화 할 때의 실행시간으로 비교하여 진행하였으며, 파일 크기와 파일 특성에 비례하여 실행시간 효율을 나타낸다.

2장에서는 AES 표준 알고리즘을 3장에서는 제안확장 방법을 설명하고, 4장에서 실험 및 결과를 5장에서 결론을 맺는다.

Ⅱ. AES(Advanced encryption standard)

AES 표준 암호화 알고리즘은 암호화와 복호화에 필요한 키(key)가 동일한 대칭 블록 암호화 알고리즘이다. AES 128/192/256 암호화 알고리즘은 그림 1과 같이 SubBytes, ShiftRow, MixColumns 및 AddRoundKey의 순서로 구성되며 각 10/12/14번 반복 수행한다. 입력 평문을 각 셀이 한 바이트인 4행 4열의 행렬, 128 비트의 데이터 블록으로 분할하여 각 셀에 대한 단계적인 암호화 연산을 수행한다. 암호화 연산은 먼저 입력 데이터 블록인 평문과 원래의 비밀 키 사이에 AddRoundKey가 이루어진 다음에, 네 가지 변환들이 순서대로 수행된다. 이 과정은 처음 (n-1)번의 라운드에 대하여 동일하게 반복되며, AddRoundKey에서는 각 라운드를 위해 생성된 라운드 키를 이용하게 된다. 그리고 마지막 n번째 라운드에서는 MixColumns가 수행되지 않으며 복호화는 암호화 과정의 역순으로 진행된다.

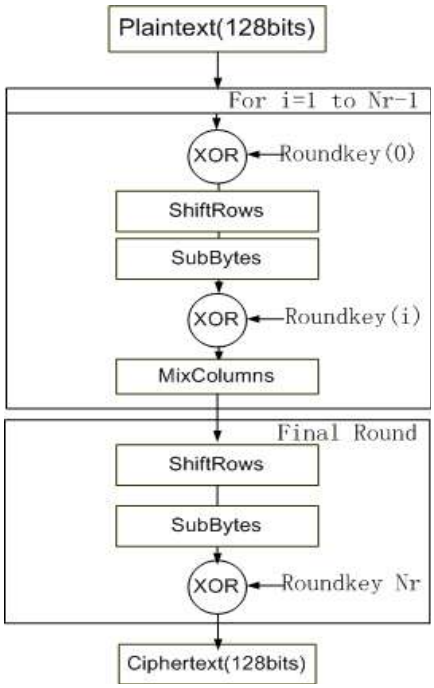


그림 1. AES 알고리즘의 암호화 과정
Fig. 1. Encryption process of the AES algorithm

1. AES 연산

연산 SubBytes는 S-box($GF(2^8)$)를 이용한 블록의 각 바이트에 독립적으로 작용하는 비선형적 바이트 치환이다. S-box는 각 바이트에 대하여 일대일 치환을 수행하기 위한 비선형 치환 테이블로서, 16×16 형태의 256 바이트로 구성된다. 암호화와 복호화에 사용되는 S-box는 서로 대칭이며, 복호화에는 S-box의 역을 사용한다. ShiftRow 연산은 블록의 행 단위로 변환이 이루어진다. 첫째 행을 제외한 나머지 행들에 대해 행 인덱스만큼씩 바이트 단위의 순환 이동을 통해 위치를 교환한다. 행 번호가 0인 첫 행은 쉬프트 되지 않고, 행 번호가 1인 행은 한 번, 2행은 두 번, 3행은 3번 각각 쉬프트 된다. 이를 통해 같은 행에 있는 바이트들이 열의 번호가 낮은 위치로 이동하고 열 번호가 가장 낮은 위치의 바이트는 최상위 열의 위치로 가게 된다. MixColumns 연산은 블록의 각 열들을 4차 다항식 $a(x)$로 취급하여 고정된 다항식 $(03x^3+02x^2+01x+01)$ $c(x)$에 대해, $a(x)\otimes c(x)=\mathrm{mod}\ x^4+1$ 연산을 수행한다. AddRoundKey는 키 생성기에 의해 생성된 라운드 키와 데이터 블록 사이에 XOR 연산을 수행함으로써 이루어진다. 각 라운드 키는 키 스케줄로부터 n_b개의 워드로

구성되며, n_b 워드들은 블록의 열과 각각 더해진다.

AES 알고리즘은 처음 입력되는 암호 키로부터 각 라운드에서 연산할 라운드 키를 생성하기 위해서 키 확장 루틴을 수행한다. 키 확장 루틴은 모든 $n_b(n_r+1)$ 만큼의 4바이트 워드들을 모두 생성한다. 키 확장 루틴의 결과는 4바이트 워드들의 선형 배열로 이루어지고, i가 0 ≤i<$n_b(n_r+1)$의 범위인 [w_i]로 표기된다. 키 확장 루틴에서는 Subword 변환과 RotWord 변환 및 Rcon[i]를 적용하여 라운드 키가 생성된다. Subword 변환은 4바이트 입력 워드로부터 출력 워드를 생성하기 위해 4바이트 각각을 S-box에 적용하여 치환한다. RotWord 변환에서는 입력으로 워드 [a_0, a_1, a_2, a_3]가 입력되었을 때 순환적 치환을 수행하여 워드 [a_1, a_2, a_3, a_0]을 출력한다. 라운드상수 워드 배열인 Rcon[i]는 [x_{i-1}, {00}, {00}, {00}]으로서, x는 {02}의 의미이고, x_{i-1}은 유한체 $GF(2^8)$ 범위 안의 값이 된다. 확장된 키의 첫 번째 n_k 워드들은 처음 입력되는 암호 키로 채워지며 다음에 오는 w_i는 그 전 워드인 w_{i-1}과 n_k 만큼의 위치 전 워드인 w_{i-n_k}와 XOR 연산을 수행한다. n_k 배수만큼 위치한 워드들에 대한 변환에는 앞에서 설명한 변환들이 적용된다. 즉, x_{i-1}과 XOR 연산을 한 후, 순환적 바이트 쉬프트를 수행하는 RotWord 변환을 하고, 4바이트들을 S-box 테이블에서 치환 연산을 하는 SubWord 변환을 한 다음 라운드 상수 Rcon[i]와 XOR 연산을 수행한다[8].

Ⅲ. AES 확장

논문에서 제안하는 대형 자료의 암호화를 위한 AES 확장에 관한 연구의 목적은 암호화 실행시간과 암호문의 견고성 및 구현상의 효율성은 유지하면서 자료 유형과 크기에 적합한 암호화 변수들을 선택할 수 있도록 하는 데 있다. 일반 평문의 유형에 따라 선택할 수 있는 주요 암호화 변수들은 평문 블록의 크기, 키 크기, 알고리즘 내부의 라운드 횟수 등을 사용자 요구와 실행 자원의 수용범위에 맞게 선택적으로 설정하여 암호화 할 수 있도록 하였다. 뿐만 아니라, 표준 AES의 내부 연산에 의해 부과되는 암호화 시간과 대형의 평문을 선택 변수에 따라 견고성은 유지하면서 암호화 실행 시간을 줄이는 것을 목적 함수로 하였다.

암호화 실행시간과 연관된 블록 암호화의 처리율은 (1)과 같이 정의된다[9].

$$Tp=\frac{BlockSize}{(rn+1)\times Clockcycle} \quad (1)$$

변수 rn은 암호화 라운드 횟수를, $Blocksize$는 각 평문 블록의 입력 크기를 의미한다. 따라서, 암호화 처리율은 입력 데이터블록의 크기와 암호화 진행을 위한 라운드 횟수에 의해 좌우된다. 반면, 외부공격에 대한 암호문의 견고성은 라운드 횟수에 비례해서 증가하게 되므로, 먼저 실행 시간을 최소화할 수 있는 가용범위의 블록 크기를 우선 결정한다. 블록 크기를 결정하기 위해 1MB 평문 입력에 대해 2×2~500×500블록 크기의 선험적 구간 실험 결과 암호화 실행시간은 20×20~30×30의 블록 크기에서 최소 실행시간 구간을 형성하였다.

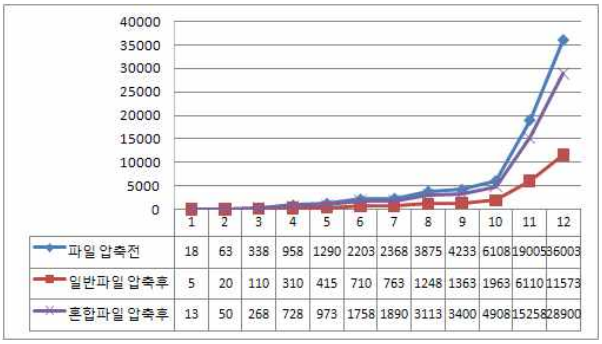
표 1. 입력 데이터
Table 1. Input data

Data		단순파일		혼합파일	
		압축전	압축후	압축전	압축후
size (KB)	1	7	2	7	5
	2	25	8	25	20
	3	135	44	135	107
	4	383	124	383	291
	5	516	166	516	389
	6	881	284	881	703
	7	947	305	947	756
	8	1550	499	1550	1245
	9	1693	545	1693	1360
	10	2443	785	2443	1963
	11	7602	2444	7602	6103
	12	14401	4629	14401	11560

AES 암호와 시간은 반복되는 라운드 내부의 연산에 의해 부과되므로 자료파일이 큰 경우에는 입력 평문의 크기를 줄임으로써 입력 블록들에 의해 반복되는 라운드 수를 줄여야한다.

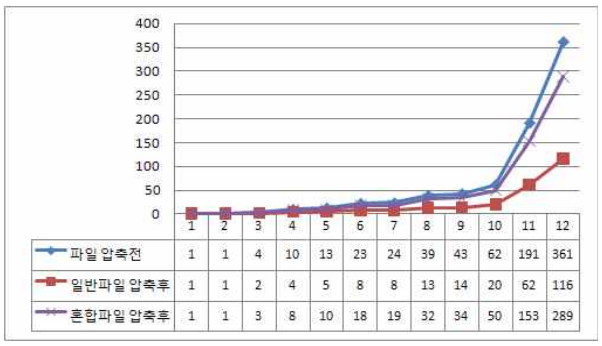
표 1과 같이 단순파일과 혼합파일의 12개의 입력 데이터에 대한 라운드 횟수변화 실험에서 파일 유형과 무관하게 압축 전후의 파일 크기에 대해 반복되는 AES 라운드 수는 그림 2, 그림 3과 같이 압축율과 입력 블록의 크

기에 비례해서 현격하게 줄어드는 것을 확인하였다.



	1	2	3	4	5	6	7	8	9	10	11	12
파일 압축전	18	63	338	958	1290	2203	2368	3875	4233	6108	19005	36003
일반파일 압축후	5	20	110	310	415	710	763	1248	1363	1963	6110	11573
혼합파일 압축후	13	50	268	728	973	1758	1890	3113	3400	4908	15258	28900

그림 2. 압축 전후 라운드 수 : 블록크기 2×2
Fig. 2. Round count before and after compression: block size 2×2



	1	2	3	4	5	6	7	8	9	10	11	12
파일 압축전	1	1	4	10	13	23	24	39	43	62	191	361
일반파일 압축후	1	1	2	4	5	8	8	13	14	20	62	116
혼합파일 압축후	1	1	3	8	10	18	19	32	34	50	153	289

그림 3. 압축 전후 라운드 수 : 블록크기 20×20
Fig. 3. Round count before and after compression: block size 20×20

본 연구의 목적함수인 대형 자료의 암호화에 있어서 단계별 AES 연산은 유지하면서 암호화 정도와 자료 유형에 적합한 암호화 변수들을 선택할 수 있게 하고 암호화 실행시간의 효율을 높이기 위한 확장 구조는 그림 4와 같다. 실험에 의해 확인된바와 같이, 파일의 유형에 따라 입력단에서 평문 압축을 선택할 수 있도록 하고, 압축된 평문에 대한 블록의 크기와 라운드 횟수를 사용자 정의에 의해 선별적으로 적용할 수 있도록 하였다. AES 내부의 암호화 연산은 표준 AES와 동일하며 ApplyKey와 SubBytes에 대한 루프 펼침과 ShiftRows에 대한 루프 펼침 및 병합의 소프트웨어 최적화 기법을 적용하였다.

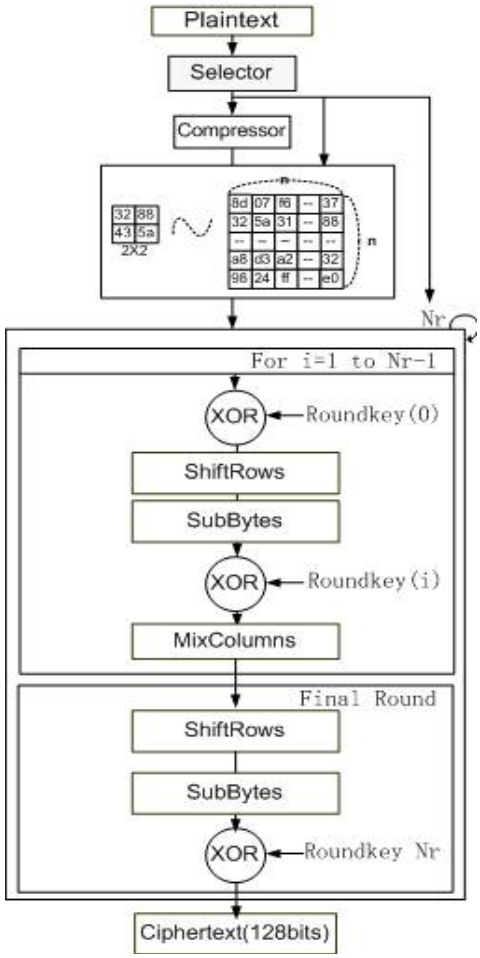


그림 4. 확장된 AES 구조
Fig. 4. Extended AES scheme

Ⅳ. 실험 및 결과

제안 방법에 의한 암호화는 표 1과 같이 12종의 단순 파일과 혼합파일 각각에 대해, 먼저 영문과 한글이 혼용된 단순 문자로만 구성된 단순 텍스트 파일과, 문자와 도형 및 그래프의 조합으로 구성된 혼합 파일로 구분하여 각각, 압축이전과 압축 이후의 파일을 입력으로 하여 실험하였으며 블록 크기는 20×20, 키 크기는 160비트, 라운드 횟수는 20으로 선택하였다. 프로그램은 Code::Blocks v8.02에서 C++로 구현하여 MinGW GCC로 컴파일 하였으며, 플랫폼은 Intel 2.4Ghz CPU와 2GRAM으로 실험하였다.

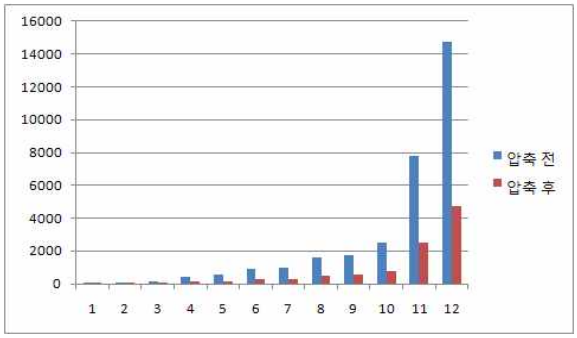


그림 5. 단순 파일 암호화 실행시간(ms)
Fig. 5. Encryption time of simple file(ms)

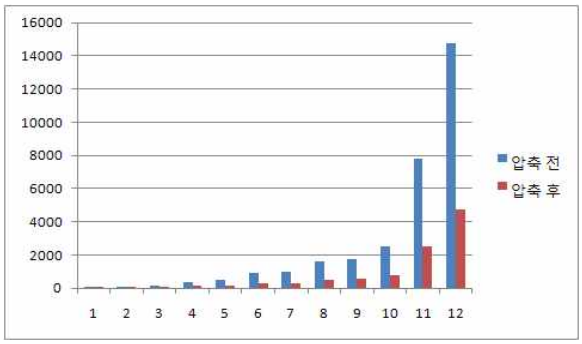


그림 6. 단순 파일 복호화 실행시간(ms)
Fig. 6. Decryption time of simple file(ms)

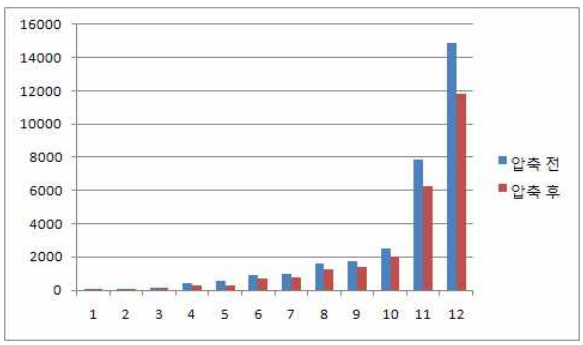


그림 7. 혼합 파일 암호화 실행시간(ms)
Fig. 7. Encryption time of complex file(ms)

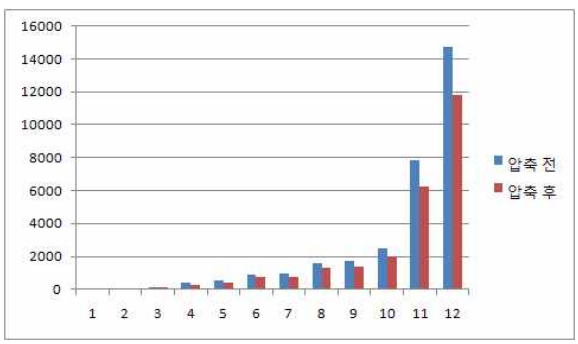


그림 8. 혼합 파일 복호화 실행시간(ms)
Fig. 8. Decryption time of complex file(ms)

실험결과 단순파일의 암호화 및 복호화에 소요되는 실행시간은 최소 55%에서 최대 70.8%, 평균 66.7%를, 단순파일 복호화에는 최소 55%에서 최대 86.9%, 평균 66.7%를, 혼합파일 암호화에는 최소 20%에서 최대 44%, 평균 25.4%를, 혼합파일 복호화에는 최소 19%에서 최대 64%, 평균 24.6%의 실행시간 개선 효과를 보였으며 그림 9와 같다. 파일크기에 대한 실행시간 영향은 균등한 반면 압축에 경과되는 실행시간 부하에 의해 단순파일과 혼합파일간의 효율성에는 큰 차이를 나타낸다.

그림 9. 실행시간 개선(%) 비교
Fig. 9. Processing time improvement(%) comparisons

V. 결 론

본 논문에서는 대형 자료의 암호화에 수반되는 실행시간과 전송자료의 크기를 최소화하기 위해 AES 블록 암호 표준 알고리듬의 확장에 있어서 암호화 성능의 주요변수인 입력 블록의 크기와 개수 및 알고리즘의 라운드 횟수를 암호화 요구수준에 맞게 선택적으로 지정할 수 있도록 하였으며, 대형 자료의 보안 전송이 가능하도록 평문을 압축하여 압축 블록을 암호화하도록 하였다. 이를 위해 암호화 전단에 허프만 압축으로 입력블록 수를 줄여 블록의 개수에 비례해서 반복되는 라운드 수를 줄였으며, 루프 펼침과 루프 병합을 적용하여 라운드 내부의 반복 실행구문에 의한 실행시간을 최소화하였다. 유연한 확장과 다양한 플랫폼 수용성을 고려하여 데이터 블록의 수는 사용자 선택에 의해 최소 블록으로부터 사용자 지정블록까지 선택 적용할 수 있도록 하였다. 실험결과는 자료 유형에 종속되지 않고 대부분의 입력데이터에 대해 시간 개선을 얻을 수 있었으며, 일부 예제에서는 허프만 압축에 수반된 소요시간으로 인해 시간 개선이 미약한 경우도 있었다.

본 논문의 확장 구조는 알고리즘의 하드웨어 설계 등

에서 면적 대비 성능의 지표로 활용될 수 있으며, 부하로 작용하는 반복적인 라운드 작업을 해결하기 위한 파이프라인 설계와 하드웨어 소프트웨어 통합설계의 지표로 활용될 수 있다. 향후 연구과제로서, 라운드 연산에서 가장 큰 시간을 소모하는 LUT(look up table) 형태의 S-box 최적 설계 기법과 라운드 연산속도 개선을 위한 파이프라인 삽입 및 다양한 유형의 공격에 대한 결과의 견고성에 대한 정량적인 평가결과를 도출하는 실험이 진행되어야 한다.

참 고 문 헌

[1] Paul A. J, Paul Varghese, Mythili P, "A fast and secure encryption algorithm for message communication," Information and Communication Technology in Electrical Sciences. 2007; 629 - 634

[2] Yang Xiao, Guizani S, Bo Sun, Hsiao-Hwa Chen, Ruhai Wang, "Performance Analysis of Advanced Encryption Standard(AES)," Global Telecommunications Conference, Nov. 27 2006-Dec. 1 2006 Page(s):1 - 5

[3] Hua Li, Jianzhou Li, "A new compact dual-core architecture for AES encryption and decryption," Electrical and Computer Engineering, Canadian Journal of Volume 33. 2008; 209 - 213

[4] Alam, M, Ghosh S, Chowdhury D.R, Sengupta I, "Single Chip Encryptor/Decryptor Core Implementation of AES Algorithm," 21st International Conference on VLSI Design, 4-8 Jan. 2008 Page(s): 693 - 698

[5] Yang Xiao, Guizani S, Bo Sun, Hsiao-Hwa Chen, Ruhai Wang, "Performance Analysis of Advanced Encryption Standard(AES)," Global Telecommunications Conference, 2006: 1-5

[6] Islam M.N, Mia M, Chowdhury M, Matin M.A, " Effect of Security Increment to Symmetric Data Encryption through AES Methodology," Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008; 291 - 294

[7] Lopez-Ongil C, Jimenez-Horas A, Portela-Garcia M, Garcia-Valderas M, Millan E.S, Entrena L, "Smart Hardening for Round-based Encryption Algorithms: Application to Advanced Encryption Standard," On-Line Testing Symposium. 2008; 167 - 168

[8] Kuan Jen Lin, Chin-Mu Hiao, Ching Hung Jhan, "Exploring HW/SW Codesign of AES Algorithm Using Custom Instructions," The 13th International Symposium on Consumer Electronics(ISCE 2009), 2009; 192 - 195

[9] Gogniat G, Wolf T, Burleson W, Diguet J.-P, Bossuet L, Vaslin R, "Reconfigurable Hardware for High-Security/High-Performance Embedded Systems: The SAFES Perspective," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 16(2). 2008; 144 - 155

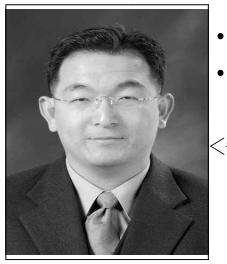
저자 소개

오 주 영(정회원)

• 전자계산학 박사(홍익대학교)
• 2003~현재 경인여대 교수
• 2002 (주)참좋은인터넷 책임연구원
• 2001 (주)이칼로스 선임연구원
• 1998 ETRI 위촉연구원

<주관심분야 : 설계자동화, 통합설계, 암호화/보안>

고 훈 준(정회원)



• 전자계산학 박사(인하대학교)
• 2004~현재 경인여대 교수

<주관심분야 : 디버거, 프로그램보안, 웹서비스, 생물정보학>