블록암호 카운터 모드 최적화 구현 동향

권혁동*, 김현준**, 장경배**, 서화정**
*한성대학교 정보컴퓨터공학과
**한성대학교 IT융합공학부
korlethean@gmail.com, khj930704@gmail.com, starj1023@gmail.com,
hwajeong84@gmail.com

Technology trends of counter mode block cipher optimization

Hyeok-Dong Kwon*, Hyun-Jun Kim**, Kyoung-Bae Jang**, Hwa-Jeong Seo**
*Dept. of Information Computer Engineering, Hansung University
**Dept. of applied IT, Hansung University

요 약

블록암호는 정해진 길이의 평문을 암호화하는 암호 알고리즘으로 정해진 길이보다 더 긴 평문을 암호화하기 위해 다양한 운용모드가 제안되었다. 그 중에서 카운터 모드는 블록암호를 스트림암호 형태로 바꿔주는 모드로, 평문 대신 고정 값인 논스와 블록의 순번인 카운터를 입력 값으로 사용한다. 카운터 모드는 논스 값이 고정이기 때문에 암호 연산 중에 논스가 사용되는 부분의 다른 변수가 모두 고정 값이라면 결과가 항상 동일하다는 특성이 있다. 본 논문에서는 전술한 특성을 사용하여 카운터 모드 최적 구현을 한 블록암호에 대해 정리하며, 각각의 성능을 비교해볼 것이다. 개략적으로 AES 기존 구현물보다 약 16% ~ 32% 정도의 성능 향상을 보이고 CHAM은 약 10% ~ 13% 정도의 성능 향상을 보였다.

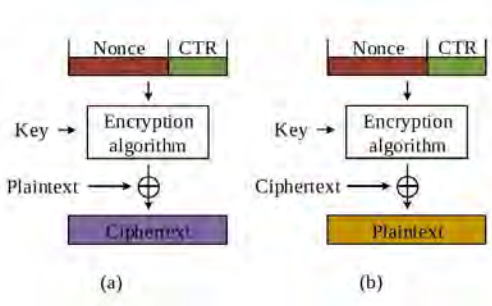
1. 서론

암호 알고리즘은 동작 원리에 따라 블록암호와 스트림암호 두 가지로 나뉜다. 블록암호는 특정 길이의 평문을 일정 블록 단위로 나누어서 암호 연산을 진행한다. 스트림암호는 평문 대신 특정한 난수를 암호화하여 평문과 XOR 연산하는 것으로 암호를 생성한다. 블록암호는 스트림암호에 비해 구조가 단순하며 효율이 뛰어나기 때문에 많이 사용되는 추세이다. 블록암호는 정해진 길이의 평문만 암호화 가능하기 때문에 이를 개선하기 위해 운용모드가 제안되었다. 블록암호 운용모드 중에서 카운터 모드는 평문 대신 고정 값인 논스와 블록의 번호인 카운터를 사용하여 블록암호를 스트림암호 방식으로 동작하게 한다.

본 논문에서는 블록암호의 카운터 모드 최적화 동향을 확인하며, 그 결과를 비교한다. 논문의 구성은 다음과 같다. 2장에서 블록암호 운용모드인 카운터 모드의 특징을 확인한다. 3장에서 각각의 최적 구현에 대해 제시하고 그 성능을 비교한다. 4장에서 결론을 맺는다.

2. 카운터 운용 모드

블록암호 운용모드 중 하나인 카운터 모드는 블록암호를 스트림암호 형태로 동작하게 만드는 운용모드이다. 일반적인 블록암호는 암호화 대상인 평문을 입력 값으로 사용하지만, 카운터 모드는 평문 대신 논스와 카운터 값을 사용한다. 이때 논스는 고정 값이며 카운터 값은 블록의 번호를 의미한다. 이를 도식화하면 [그림 1]과 같다. [그림 1]의 (a)는 암호화 과정을, (b)는 복호화 과정을 의미한다.



(그림 1) 카운터 운용 모드

카운터 운용 모드의 가장 큰 특징은 암호화 알고리즘과 복호화 알고리즘이 동일하다는 것이다. 이는 논스와 카운터 값을 암호화 하여 평문과 XOR 연산을 하는 것으로 암호문을 생성하기 때문이다. 복호화를 하기 위해서는 논스와 카운터 값의 암호화 결과 값을 암호문과 XOR하면 평문이 생성된다. 따라서 암호화 알고리즘 하나만으로 암·복호화가 모두 가능해진다. 따라서 복호화 알고리즘을 따로 구현할 필요가 없기 때문에 구현이 단순해진다.

또한 카운터 모드는 각각의 입력 값 블록들이 서로 영향을 주지 않는 구조이다. 따라서 손쉽게 병렬화가 가능하며, 이는 높은 성능 향상을 이끌 수 있다.

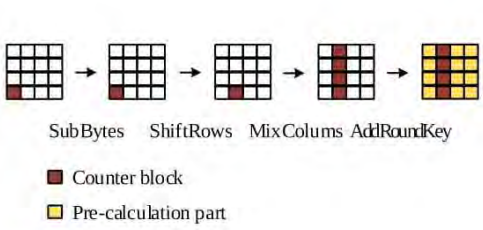
카운터 운용 모드 상에서 논스는 모두 고정 값이며 카운터는 블록의 순번을 의미하기 때문에, 논스와 연산하는 다른 값들이 모두 고정 값이면 연산 결과가 항시 동일함을 알 수 있다. 다음 장에서는 이러한 특징을 사용하여 구현한 구현물들을 확인해본다.

3. 구현 사례

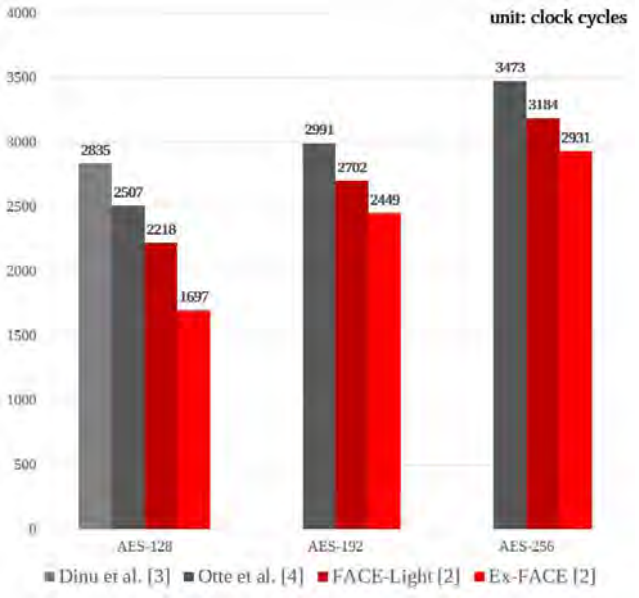
3.1 AES

[1]은 AES의 카운터 모드를 최적한 구현 기법으로, 사전 테이블(Look Up Table, LUT)를 사용하여 구현하였다. AES는 SubBytes, ShiftRows, MixColumns, AddRoundKey의 4단계로 구성되어 있다. 이때 카운터 값은 MixColumns 단계에서 다른 블록으로 확산되게 된다. 즉, MixColumns 단계 직전의 값 중에서 논스 블록과 연산하는 부분은 연산 결과가 항상 동일하다. [1]에서는 1라운드 직후와 2라운드 직후에서 MixColums 직전까지의 연산을 미리 계산해둔다. 그 후 해당 값들을 테이블에 저장해두는 것으로 중간 연산단계를 생략하여 고속으로 동작하게 하였다. [그림 2]는 1라운드 중에서 사전연산이 가능한 부분과 카운터 이동을 묘사하였다.

이후 [1]을 개선한 [2]가 제시되었다. [2]는 [1]의 구현을 개선한 것으로, 카운터 값에 의존하는 반복 부분을 저장하여 한번에 다수의 라운드를 생략할 수 있도록 하였다. 또한 LUT의 사이즈가 줄어들어 8-bit 마이크로컨트롤러를 지원하게 되었고 LUT의 업데이트도 필요 없게 되었다. [2]와 기존 AES 최적 구현물과 성능 비교 결과는 [그림 3]과 같다. AES-128, AES-192, AES-256 모든 경우에서 [1]을 개선한 [2]가 기존 AES 최적 구현물인 [3]과 [4]보다 뛰어난 성능을 보이고 있다.



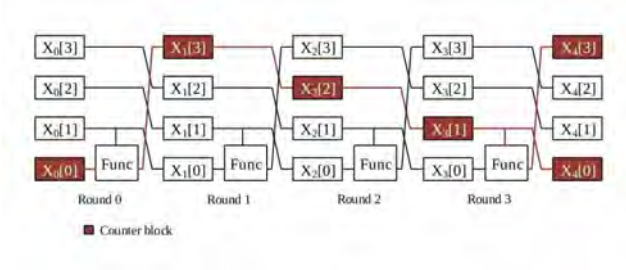
(그림 2) AES 카운터 운용 모드



(그림 3) AES 구현물 성능 비교

3.2 CHAM

CHAM[5]은 국산 경량암호로 개선된 버전이 [6]에서 제안되었다. [6]을 대상으로 하는 CHAM 카운터 모드 최적 구현이 [7]에서 제안되었다. [7]은 CHAM이 평문 블록을 4개로 나누어서 연산하는 점에 착안하였다. CHAM은 각 라운드마다 두 개의 블록이 연산에 참여하며, 라운드 종료 시 모든 블록이 왼쪽으로 워드 단위 시프트가 이루어진다. 따라서 카운터 블록이 라운드에 참가하여 다른 블록에 전파되기 전 까지는, 각 라운드마다 블록의 연산 결과는 모두 동일하다. [그림 4]는 CHAM의 라운드 함수 동작 중에 카운터의 흐름을 묘사하였다. [그림 4]에서 일정 라운드를 지날 때마다 카운터에 영향받는 블록이 늘어나는 것을 확인할 수 있다.

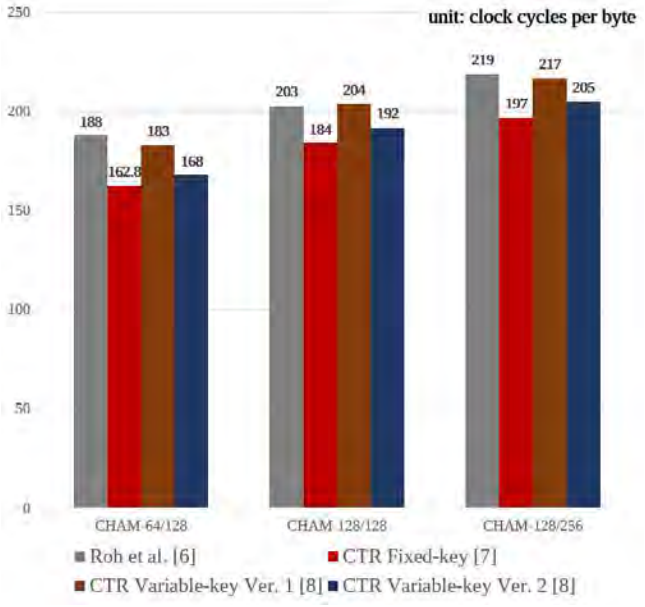


(그림 4) CHAM 카운터 운용 모드의 구조

[7]은 카운터 블록에 영향 받지 않는 부분을 사전연산 할 수 있다는 점에 착안하였다. 때문에 카운터 블록의 이동을 확인하며 영향 받지 않는 부분의 값은 라운드 함수 동작을 생략하고 연산 결과 값을 직접 입력하는 형식으로 구현하였다. 라운드 함수의 연산만큼 연산을 생략할 수 있으므로, 상당한 성능 개선을 보여준다.

하지만 [7]은 키가 고정 상태인 고정키 시나리오를 기반으로 구현하였다. 암호 알고리즘을 사용하는 상황에서는 키가 변경되는 상황도 있을 수 있으므로, 이를 개선한 [8]이 제안되었다. [8]은 [7]과는 다르게 가변키 시나리오를 상정하여 구현하였다. 키가 바뀌면 중간의 연산 결과가 달라진다. 따라서 [7]과 다르게 값을 직접 입력하는 방식이 아닌, LUT를 사용하는 방법을 사용한다. 이때 LUT는 사전연산 함수가 따로 계산하며 두 가지 방식이 있다. 첫 번째는 사전연산만을 진행하는 방식이다. 이 방식은 사전연산에 필요한 값들과 필요한 라운드만 진행하기 때문에 통상적인 사전연산보다 빠르게 진행할 수 있다. 두 번째는 사전연산을 진행하며 1개의 블록을 암호화 하는 방식이다. 블록암호 운용모드를 적용하는 것은 블록이 1개보다 더 많을 때 사용하기 때문에 사전연산을 진행하며 암호화를 1회 진행하는 것은 상당히 효율적인 방식이다. [8]에서는 이 두 가지 방식 중에 두 번째 방식이 더 뛰어난 성능을 보인 것을 제시하였다.

[그림 5]는 고정키 시나리오 구현물인 [7]과 가변키 시나리오 구현물 [8], 그리고 기본 형태인 [6]의 알고리즘 성능 비교를 한다. [그림 5]에서 [8]에서 제시한 구현물은 [7]의 구현물보다 조금 더 낮은 성능을 보인다. 이는 [7]은 고정된 연산 결과를 직접 입력하며 [8]은 LUT에서 값을 가져오는 방식이기 때문에 조금 더 낮은 성능을 지니게 된 것이다.



(그림 5) CHAM 카운터 운용 모드 성능 비교

4. 결론

본 논문에서는 블록암호 카운터 모드의 최적 구현에 대한 동향을 확인하였다. 카운터 모드는 낮은 구현 난이도와 병렬화 적용이 용이하기 때문에 활용도가 높다. 또한 각종 인증에 사용되는 GCM(Galois/Counter mode)의 적용에도 유리하기 때문에 그 활용도가 무궁무진하다. 또한 카운터 모드를 적용한 AES, CHAM의 최적 구현 사례에서 고속 구현의 가능성을 확인할 수 있었다. 따라서 실용적으로 사용할 수 있는 기반이 완성되었다 할 수 있다.

본 논문에서는 AES, CHAM의 사례에 대해서만 확인하였다. 하지만 제안 기법들은 선형 연산 위주로 동작하는 다른 블록암호에도 제안 기법들의 적용이 가능하다. 따라서 SPECK, SIMON과 같은 다양한 블록암호에 기존 제시된 기법들을 적용하여 최적 구현을 하는 방안을 후속 연구로 제시할 수 있다.

참고문헌

[1] J. H. Park, and D. H. Lee, “FACE: Fast AES CTR mode Encryption Techniques based on the Reuse of Repetitive Data,” IACR Transactions on Cryptographic Hardware and Embedded Systems, Vol. 2018, No. 3, pp. 469-499, 2018.

[2] K. H. Kim, S. J. Choi, H. D. Kwon, Z. Liu, H. J. Seo, “FACE - LIGHT: Fast AES - CTR Mode

Encryption for Low-End Microcontrollers," International Conference on Information Security and Cryptology(ICISC), Seoul, 2019, pp 102-114.
[3] D. Dinu, A. Biryukov, J. Großsch ¨adl, D. Khovratovich, Y. Le Coree, and L. Perrin, "FELICS - fair evaluation of lightweight cryptographic systems," NIST Workshop on Lightweight Cryptography, Vol. 128, 2015.
[4] D. Otte., "AVR Cryptography Library," Online: http://www.das-labor.org/wiki/AVR-Crypto-Lib/en , 2009.
[5] B. W. Koo, D. Y. Roh, H. J. Kim, Y. H. Jung, D. Lee, and D. Kwon, "CHAM: A Family of Lightweight Block Ciphers for Resource-Constrained Devices," International Conference on Information Security and Cryptology(ICISC), Seoul, 2017, pp 3-25.
[6] D. Y. Roh, B. W. Koo, Y. H. Jung, I. W. Jeong, D. G. Lee, D. S. Kwon, and W. H. Kim, "Revised Version of Block Cipher CHAM," International Conference on Information Security and Cryptology(ICISC), Seoul, 2019, pp 8-25.
[7] H. D. Kwon, H. J. Kim, S. J. Choi, K. B. Jang, J. H. Park, H. J. Kim, and H. J. Seo, "Compact implementation of CHAM Block Cipher on Low-End Microcontrollers," World Conference on Information Security Applications(WISA), 2020,
[8] H. D. Kwon, S. W. An, Y. B. Kim, H. J. Kim, S. J. Choi, K. B. Jang, J. H. Park, H. J. Kim, S. C. Seo, and H. J. Seo, "Designing a CHAM Block Cipher on Low-EndMicrocontrollers for Internet of Things," Multidisciplinary Digital Publishing Institute Electronics, Vol. 9, 2020, 16 pages.