Round 함수 병렬화에 따른 SHA-512 하드웨어 구현 분석

여지은, 최소연, 박지운, *유호영
충남대학교 전자공학과
e-mail : jeyeo.cas@gmail.com, sychoi.cas@gmail.com, jwpark.cas@gmail.com, hyyoo@cnu.ac.kr

SHA-512 hardware implementation analysis according to round function parallelism

Jieun Yeo, Soyeon Choi, Jiwoon Park, and *Hoyoung Yoo
Department of Electronics Engineering
Chungnam National University

Abstract

Hash functions are widely used to ensure the integrity of information for network and communication systems. Among the various hash functions, SHA-512 in NIST SHA-2 family provides strong encryption capability and affordable hardware complexity. In this paper, we compare different SHA-512 structures with various parallel factor for round function. According to the experimental results, a small parallel factor leads to small equivalent gate counts and short critical path delay, and a large parallel factor tends to result in high throughput.

I. 서론

일반적으로 암호화 알고리즘은 크게 대칭형 암호화, 비대칭형 암호화, 해시 함수로 나눌 수 있다. 평문을 암호화하고 암호문을 복호화하는 양 방향 대칭형/비대칭형 암호화와는 달리 해시 함수는 단 방향 함수이기 때문에 평문을 암호화하는 것은 가능하지만

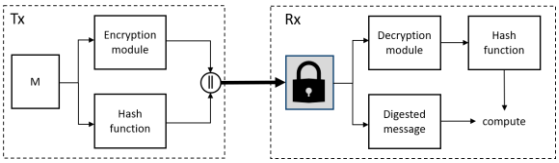

그림 1. 해시 함수의 이용

암호문을 평문으로 복호화 하는 것은 불가능하다. 단방향 함수의 특성을 활용하여 해시 함수는 일반적으로 안정성이 보장되지 않는 채널에서 정보의 무결성을 확보하기 위하여 활용된다 [1]. 그림 1과 같이 송신부에서는 암호화할 메시지를 입력으로 해시 함수의 결과 값인 DM(Digested Message)을 생성하여 메시지와 결합하여 전송한다. 수신부는 수신한 정보로부터 메시지와 DM을 분리하고, 수신된 메시지를 이용하여 송신부와 동일한 해시 함수를 활용하여 새로운 DM을 생성한다. 최종적으로 수신한 DM과 수신부에서 생성한 DM을 비교하여 두 값이 동일하면 전송 중에 메시지에 오류가 발생하지 않은 것으로 판단하고, 동일하지 않다면 전송 중 오류가 발생한 것으로 판단하여 메시지에 대한 결함을 확인할

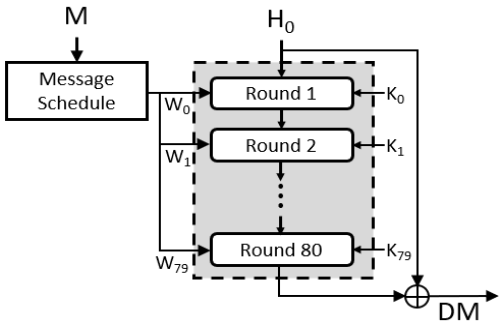

그림 2 SHA-512의 구조

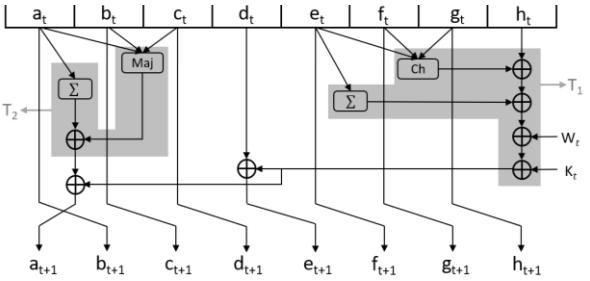

그림 3 SHA-512의 라운드 함수

수 있다 [2].

다양한 해시 함수 중 NIST에서 설계한 SHA Secure Hash Algorithm)함수가 가장 널리 활용된다 [3]. SHA는 임의의 길이를 가지는 메시지를 입력받아 고정된 길이의 DM를 출력하며, 최종 DM을 출력하기 위해서 라운드 함수의 연산을 반복하여 수행한다. SHA의 종류는 SHA-1, SHA-2, SHA-3가 존재하며 우수한 보안성과 적절한 수준의 암호/복호화 복잡도를 제공하는 SHA-2가 가장 널리 활용되고 있다 [4]. SHA-2는 해시 함수의 최종 출력인 DM의 길이에 따라 SHA-224, SHA-256, SHA-384, SHA-512로 구분이 된다. 본 논문에서는 복잡한 SHA-512에 대해 다양한 병렬화 계수를 가지는 하드웨어 구조를 구현하고 다양한 성능 지표에 대하여 비교, 분석한다.

II. 본론

SHA-512는 그림 2와 같이 크게 메시지 스케줄 모듈, 라운드 함수로 구성된다. 입력 블록 M은 1024비트로 구성되어 있으며 메시지 스케줄 모듈을 거쳐 64비트 워드 W_t ($0 \le t < 80$)를 식 (1)에 따라 생성한다 [5].

$$W_t = \begin{cases} M_t, & 0 \le t \le 15 \\ \partial_1^{512}(W_{t-2}) \oplus W_{t-7} \oplus \partial_0^{512} W_{t-15} \oplus W_{t-16}, & 16 \le t \le 79 \end{cases} \quad (1)$$

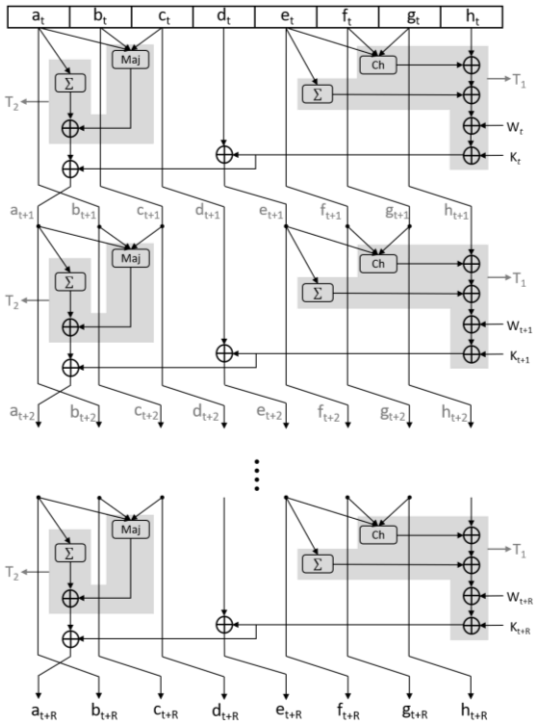

그림 4. 한 클럭 사이클에서 수행하는 라운드 함수가 R인 경우의 라운드 함수 구조

이때 $\partial_0^{512}(x)$ 와 $\partial_1^{512}(x)$ 는 각각 $\partial_0^{512}(x) = ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x)$ 와 $\partial_1^{512}(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x)$에 의하여 계산되고, $ROTR^n(x)$는 n-비트 순환 우 천이(n-bit circular right shift), $SHR^n(x)$는 n-비트 좌 천이(n-bit left shift)를 의미한다 [5]. 그림 2에 나타난 라운드 함수는 워드 W_t와 키 K_t를 입력으로 사용하는데, 키 K_t ($0 \le t < 80$)는 라운드 별로 지정된 64비트의 상수 값으로 [5]에서 제공된다. 식 (1)에 의해 생성된 워드 W_t ($0 \le t < 80$) 와 키 K_t ($0 \le t < 80$)는 각 라운드 함수에 입력되어 최종적으로 512비트의 DM을 생성한다.

그림 3는 라운드 함수의 내부 구조를 나타내며, a_t, b_t, c_t, d_t, e_t, f_t, g_t, h_t는 라운드에서 계산된 해시 값을 저장하기 위한 64비트로 구성된 8개의 해시 버퍼이다. 각 해시 버퍼의 초기값은 지정된 상수 값으로 표준문서에서 제공되며, 각 해시 버퍼는 그림 3의 라운드 함수에 의해 업데이트 되고, Σ, Maj, 그리고 Ch 연산은 식 (2)와 같다 [5].

$$\begin{aligned} &\textstyle\sum_0^{512} a = ROTR^{28}(a) \oplus ROTR^{34}(a) \oplus ROTR^{39}(a) \\ &\textstyle\sum_1^{512} e = ROTR^{14}(e) \oplus ROTR^{18}(e) \oplus ROTR^{41}(e) \\ &Maj(a,b,c) = (a\ AND\ b) \oplus (a\ AND\ c) \oplus (b\ AND\ c) \\ &Ch(e,f,g) = (e\ AND\ f) \oplus (NOT\ e\ AND\ g) \end{aligned} \quad (2)$$

Parallelism [R]	Latency [#cycle]	Clock Period [ns]	Critical Path Delay [ns]	Equivalent Gate Count	Throughput [Gbps]	Normalized throughput
80	1	360	358.7	529k	1.42	2684
40	2	180	175.6	346k	1.45	4190
20	4	83	86.7	192k	1.47	7656
10	8	45	43.7	107k	1.46	13644
4	20	20	17.7	44k	1.44	32727
2	40	12	10.2	22k	1.25	56818
1	80	8	6.4	12k	1	83333

표 1. 라운드 반복 횟수에 따른 SHA-512

SHA-512는 해시 결과 값인 DM을 출력하기 위해서 그림 2의 라운드 함수를 80번 반복 수행한다. 하드웨어 설계측면에서 라운드 함수의 수에 대한 병렬화 계수에 따른 다양한 설계가 가능하다. 라운드 함수의 병렬화 계수를 R이라고 가정하면, 그림 4과 같이 한 클럭 사이클에서 수행하는 라운드 함수의 수인 R에 따라 다양한 성능 지표에 대해 트레이드-오프 관계가 존재한다.

본 논문에서는 한 클럭 사이클에서 수행하는 라운드 함수의 수에 따라 달라지는 임계 경로 지연시간과 면적을 비교하여 그 결과를 바탕으로 동작 주파수에 따라 SHA-512가 가장 빠른 구조를 분석한다.

Ⅲ. 구현 및 결론

한 클럭 사이클에서 수행하는 라운드 함수의 수를 다르게 하여 하드웨어로 구현한 후 임계경로 지연시간, 하드웨어 면적, 그리고 단위면적 당 처리량을 비교하여 표 1에 나타냈다. 라운드 함수의 수에 따라 동작 주파수를 달리하였고 CMOS 180 nm 공정을 이용하여 합성을 진행하였다. 표 1에서 알 수 있듯이, 한 클럭 주기에서 수행하는 라운드 함수의 수에 비례하여 임계경로 지연시간이 증가한다. 하드웨어 면적 또한 한 클럭 주기에서 수행하는 라운드 함수의 수에 비례하여 증가하는 것을 알 수 있다. 또한 라운드 함수의 수가 증가해도 처리량 관점에서는 크게 변화하지 않는 것을 알 수 있다.

참고문헌

[1] H.E. Michail, G.S. Athanasiou, G. Theodoridis, C.E. Goutis, "On the development of high-throughput and area-efficient multi-mode cryptographic hash designs in FPGAs", in Integration, the VLSI Journal, vol. 47, no. 4, pp. 387-407, 2014.

[2] M. D. Rote, N. Vijendran, and D. Selvakumar, "High performance SHA-2 core using the round pipelined technique," in Proc. IEEE Int. Conf. Electron. Comput. Commun. Technol., Jul. 2015, pp. 1–6.

[3] R. Chaves, G. Kuzmanov, L. Sousa, and S. Vassiliadis, "Cost-efficient SHA hardware accelerators," IEEE Trans. Very Large Scale Integr. Syst., vol. 16, no. 8, pp. 999–1008, Aug. 2008.

[4] R. Martino and A. Cilardo, "SHA-2 Acceleration Meeting the Needs of Emerging Applications: A Comparative Survey," in IEEE Access, vol. 8, pp. 28415-28436, 2020.

[5] National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, [Online].Available:https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf.