



Computer Security

SHA 2

Hyounghick Kim

Department of Software

College of Software

Sungkyunkwan University

Flavors of SHA

- SHA-0

- SHA-1

- SHA-2

 - SHA-224

 - SHA-256

 - SHA-384

 - SHA-512



Longer hash value = more secure

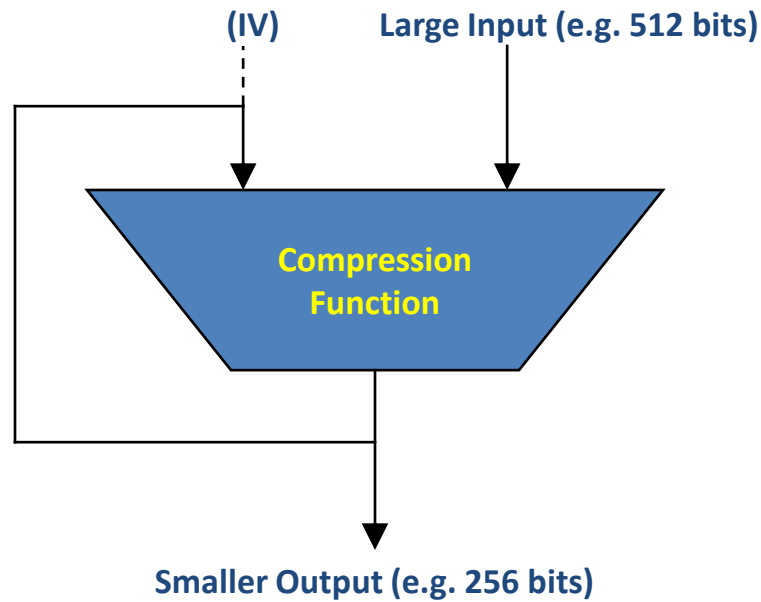
SHA history

- 1993
 - The hash function SHA-0 was issued as a federal standard by NIST
- 1995
 - SHA-1 published as the successor to SHA-0
- 2001
 - NIST revised FIPS 180 and added SHA-2
- 2002
 - SHA-2 variants
 - SHA-256, SHA-384, and SHA-512 published
- 2004
 - SHA-224 published

SHA-256

- Message is processed in 512-bit blocks sequentially, just like SHA-1
- Message digest is 256 bits instead of SHA-1's 160-bits
- 64 rounds instead of 80 rounds of compression
- Algorithm structure same as SHA-1
 - Buffer initiation
 - Padding bits
 - Processing of message
 - Output

Merkle-Damgård Construction



Buffer initiation (initial hash value)

- Eight 32-bit words instead of five in SHA-1

$H_0 = 0x6a09e667$

$H_1 = 0xbb67ae85$

$H_2 = 0x3c6ef372$

$H_3 = 0xa54ff53a$

$H_4 = 0x510e527f$

$H_5 = 0x9b05688c$

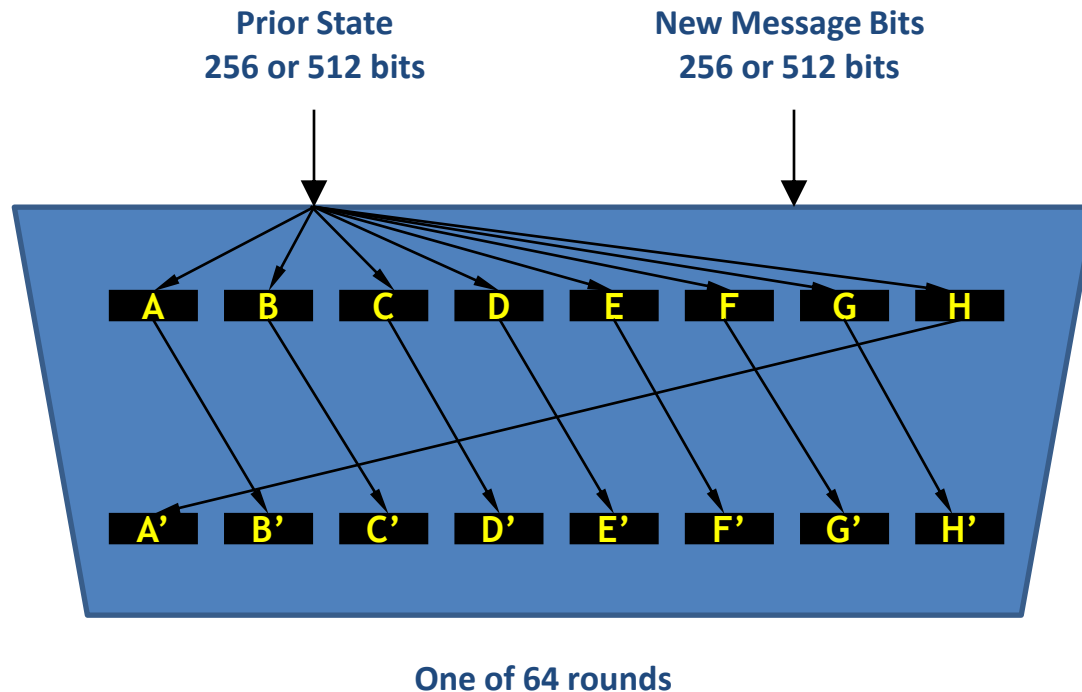
$H_6 = 0x1f83d9ab$

$H_7 = 0x5be0cd19$

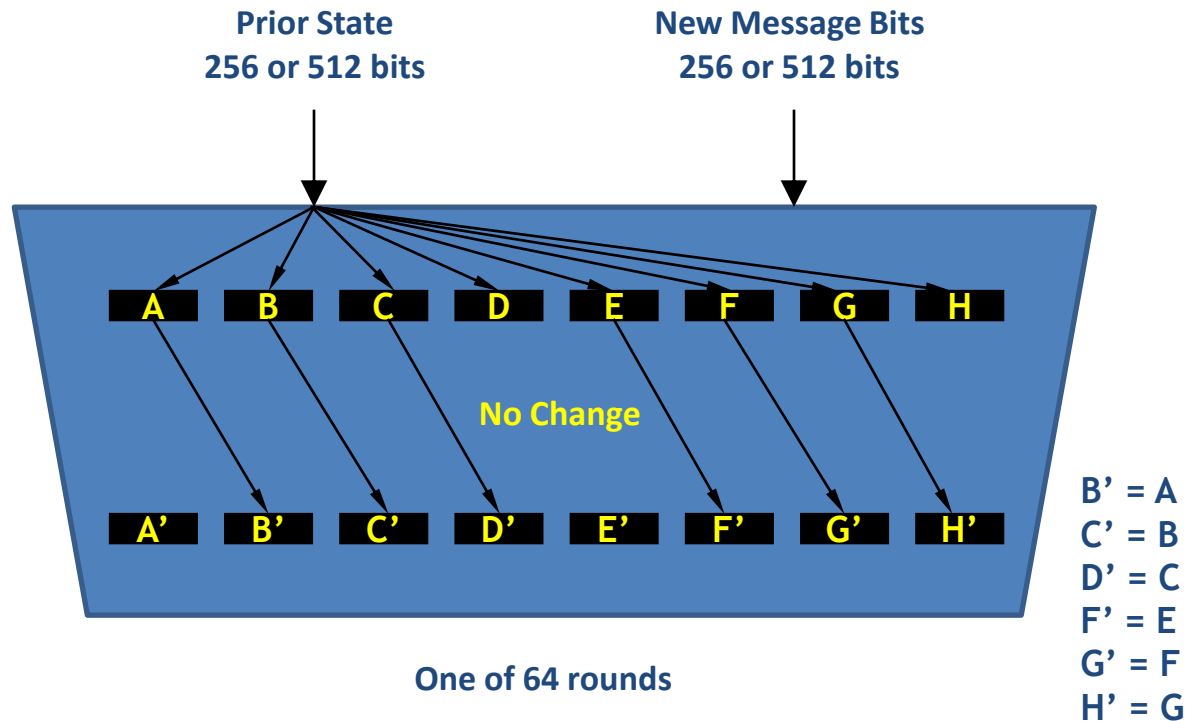
Source code for initiation

```
void sha256_init(SHA256_CTX *ctx)
{
    ...
    ctx->state[0] = 0x6a09e667;
    ctx->state[1] = 0xbb67ae85;
    ctx->state[2] = 0x3c6ef372;
    ctx->state[3] = 0xa54ff53a;
    ctx->state[4] = 0x510e527f;
    ctx->state[5] = 0x9b05688c;
    ctx->state[6] = 0x1f83d9ab;
    ctx->state[7] = 0x5be0cd19;
}
```

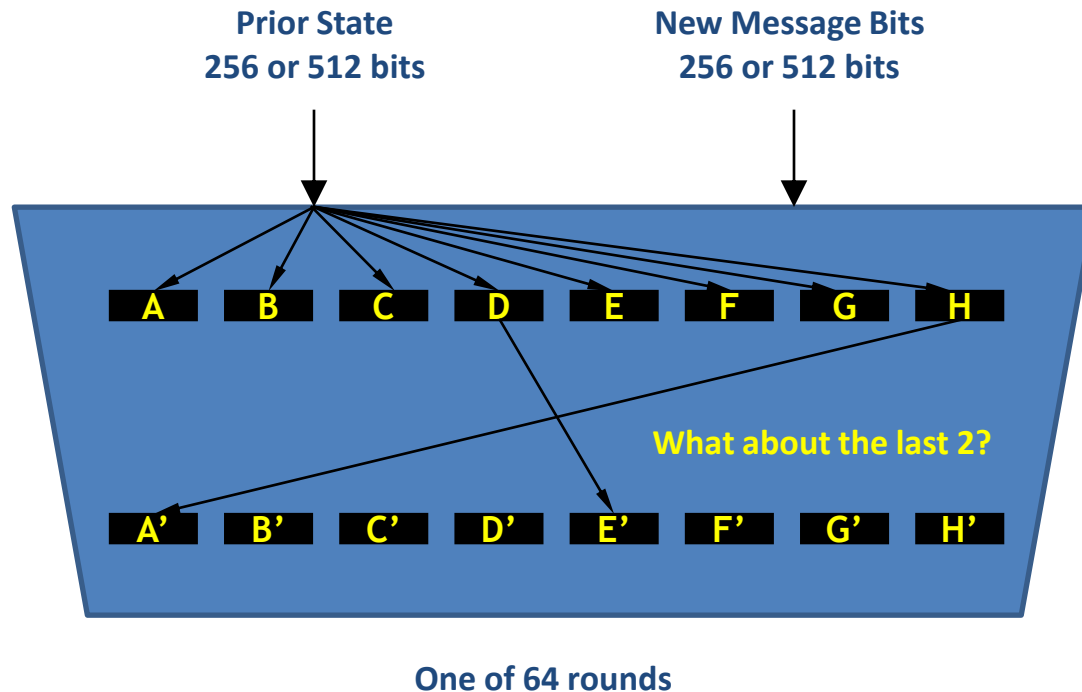
Merkle-Damgård Construction for SHA-2



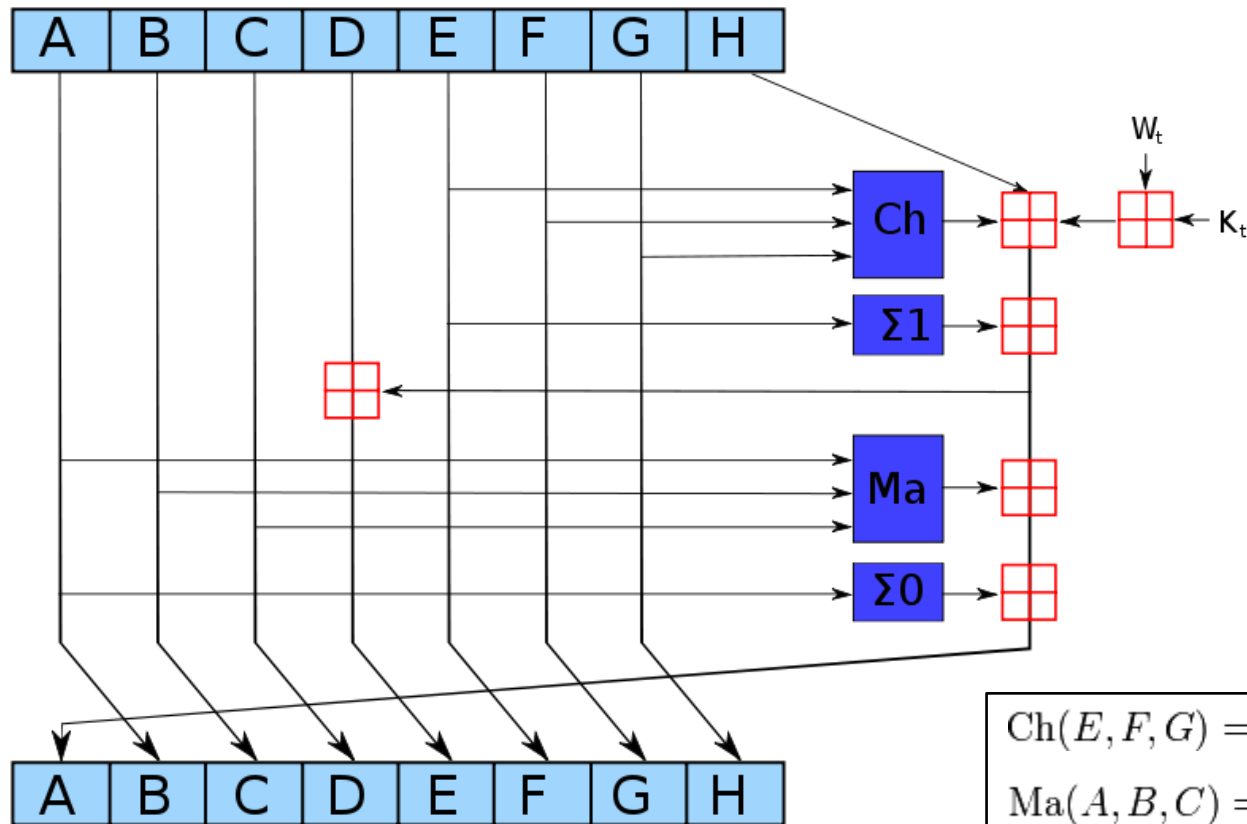
Merkle-Damgård Construction for SHA-2



Merkle-Damgård Construction for SHA-2



Round Function in SHA-2




$$Ch(E, F, G) = (E \wedge F) \oplus (\neg E \wedge G)$$

$$Ma(A, B, C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$$

$$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$$

$$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$$

 Addition modulo 2^{32}

Word expansion for Input W_t

- Each step t ($0 \leq t \leq 63$):
 - If $t < 16$
 - $W_t = t^{\text{th}}$ 32-bit word of M
 - If $16 \leq t \leq 63$
 - $S_0 = (W_{t-15} \text{ rightrotate } 7) \oplus (W_{t-15} \text{ rightrotate } 18) \oplus (W_{t-15} \gg 3)$
 - $S_1 = (W_{t-2} \text{ rightrotate } 17) \oplus (W_{t-2} \text{ rightrotate } 19) \oplus (W_{t-2} \gg 10)$
 - $W_t = S_1 + W_{t-7} + S_0 + W_{t-16}$

Source code for word expansion

```
void sha256_transform(SHA256_CTX *ctx, uchar data[])
{
    ...
    for (i=0,j=0; i < 16; ++i, j += 4)
        m[i] = (data[j]<<24)|(data[j+1]<<16)|(data[j+2]<<8)|(data[j+3]);
    for ( ; i < 64; ++i)
        m[i] = SIG1(m[i-2]) + m[i-7] + SIG0(m[i-15]) + m[i-16];
    ...
}
```

Source code for transformation

```
void sha256_transform(SHA256_CTX *ctx, uchar data[])
{
    ...
    for (i = 0; i < 64; ++i) {
        t1 = h + EP1(e) + CH(e,f,g) + k[i] + m[i];
        t2 = EP0(a) + MAJ(a,b,c);
        h = g;
        g = f;
        f = e;
        e = d + t1;
        d = c;
        c = b;
        b = a;
        a = t1 + t2;
    }

    ctx->state[0] += a;
    ...
}
```

Comparison between SHA's

Algorithm	Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)	Rounds	Operations	Collision
SHA-0	160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rotl	Yes
SHA-1	160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rotl	2^{61} attack
SHA-256/224	256/224	256	512	$2^{64} - 1$	32	64	+,and,or,xor,shr,rotr	None yet
SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80	+,and,or,xor,shr,rotr	None yet

Questions?

