ARIRANG : SHA-3 PROPOSAL

CIST, Korea Univ.

Presented by Jongsung Kim

joshep@cist.korea.ac.kr

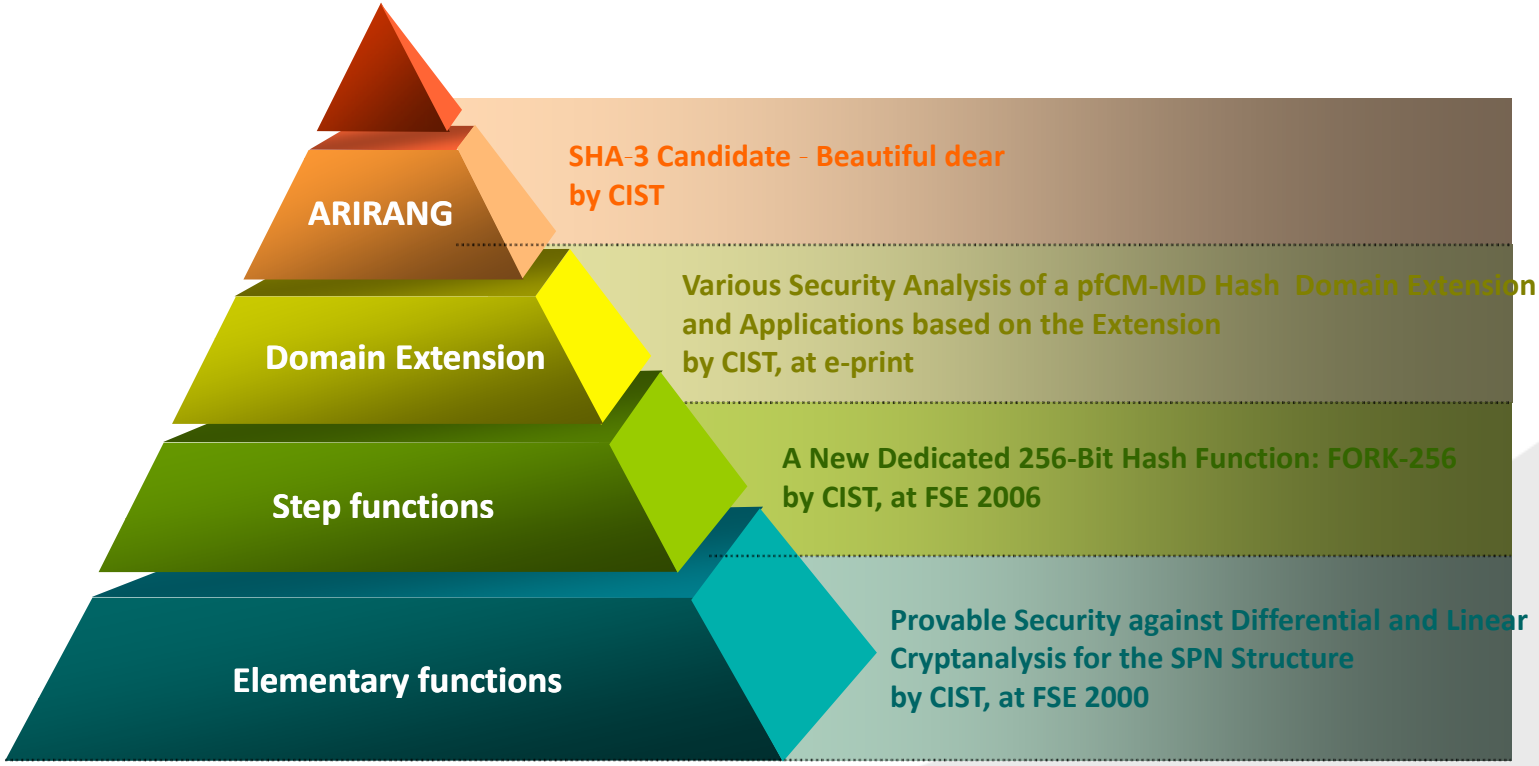
Designers:

Donghoon Chang, Seokhie Hong, Changheon Kang, Jinkeon Kang, Jongsung Kim, Changhoon Lee, Jesang Lee, Jongtae Lee, Sangjin Lee, Yuseop Lee, Jongin Lim, Jaechul Sung

Contents

1. Introduction

is the most popular and best-known Korean folk song.
'Ari' means beautiful, and `Rang' means dear.



2. Design Rationale

DESIGN RATIONALE is on six criteria.

1. Resistance against all known attacks.
2. Formal security proofs of the domain extension of the ARIRANG family.
3. Implementation efficiency.
4. Design logics of a block cipher.
5. Compatibility with application of original SHA-family.
6. Design simplicity.

3. Specification of ARIRANG (1)

ARIRANG is an iterative hash algorithm family.

◆ ARIRANG-family includes four hash algorithms.

- ARIRANG-224, ARIRANG-256, ARIRANG-384, ARIRANG-512.

◆ Each algorithm depends on message digest size.

	Message Size	Block Size	Word Size	Message Digest Size
ARIRANG-224	$< 2^{64}$	512	32	224
ARIRANG-256	$< 2^{64}$	512	32	256
ARIRANG-384	$< 2^{128}$	1024	64	384
ARIRANG-512	$< 2^{128}$	1024	64	512

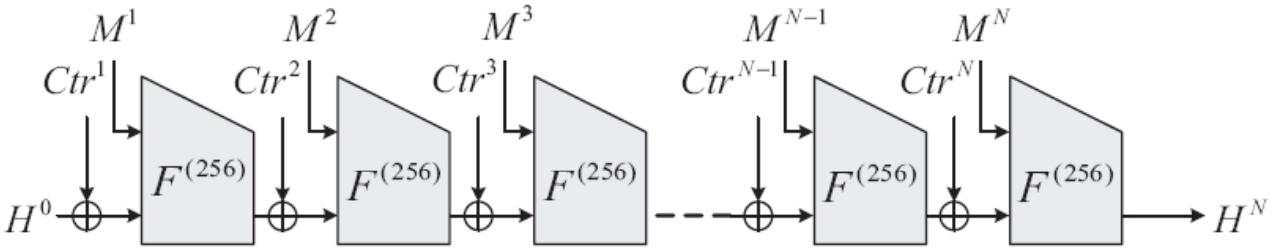
3. Specification of ARIRANG (2)

STRUCTURE

ARIRANG consists of three functions :
Preprocessing, CounterAddition, CompressionFunction

```
ARIRANG256(M)
{
    ARIRANG256_Preprocessing()

    For i = 1 to N
        ARIRANG256_CounterAddition(H^{i-1}, Ctr^i)
        ARIRANG256_CompressionFunction(H^{i-1}, M^i, H^i)
}
```



3. Specification of ARIRANG (3)

PREPROCESSING shall take place before hash computation begins.

1. Padding the Message.
2. Parsing the Padded Message
3. Setting the Counter Values (Ctr^i)
4. Setting the Initial Hash Value (H^0)

3. Specification of ARIRANG (4)

PREPROCESSING_- 1 padding the Message.

- The *l*-bit message is padded by the following padding rule
- so that the length of the padded message is ensured to be a multiple of 512 bits.

1. The bit "1" is appended to the end of the message, followed by *k* zero bits, where *k* is the smallest non-negative integer such that

$$l + 1 + k \equiv 448 \bmod 512.$$

2. The 64-bit binary representation of the integer *l* is again appended.

3. Specification of ARIRANG (5)

PREPROCESSING_- 2 parsing the Padded Message.

◆ After a message has been padded,

it is parsed into N 512-bit blocks $M^1, M^2, \cdots, M^N$.

◆ Message block i is denoted by the concatenation of sixteen words,

$M_0^i \,||\, \cdots \,||\, M_{15}^i$. Where $|M_j^i| = 32$ bits

3. Specification of ARIRANG (6)

PREPROCESSING_- 3 setting the Counter Values (Ctr^i)

◆ The purpose of setting the counter value Ctr^i is to ensure

- Each compression function of ARIRANG used to produce a message digest is mutually different.

◆ $Ctr^1 = 0$ / $Ctr^2 = 1$ / $Ctr^3 = 2$ /⋯ / $Ctr^{N-1} = N - 2$ / $Ctr^N = P$

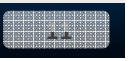
◆ P = 0xB7E151628AED2A6A obtained by taking the first 64 bits of the fractional parts of a normal number e

3. Specification of ARIRANG (7)

PREPROCESSING_- 4 setting the Initial Hash Value (H^0)

◆ These words are obtained by taking the first 32 bits of the fractional parts of the square roots for the first eight prime numbers each.

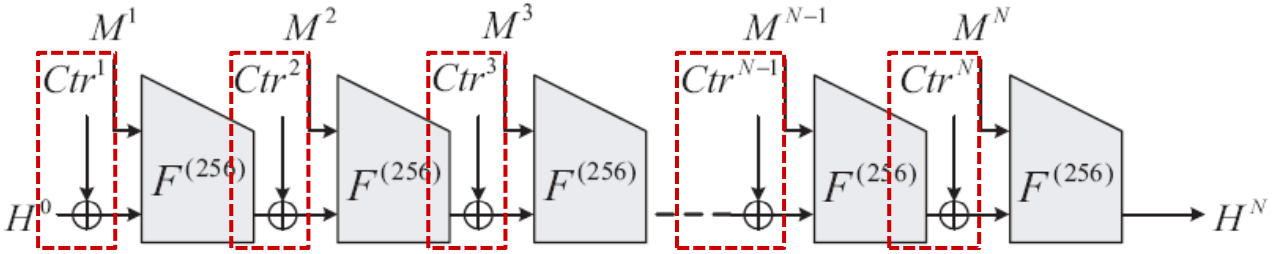
$$
\begin{aligned}
H_0^0 &= \texttt{0x6A09E667}, \\
H_1^0 &= \texttt{0xBB67AE85}, \\
H_2^0 &= \texttt{0x3C6EF372}, \\
H_3^0 &= \texttt{0xA54FF53A}, \\
H_4^0 &= \texttt{0x510E527F}, \\
H_5^0 &= \texttt{0x9B05688C}, \\
H_6^0 &= \texttt{0x1F83D9AB}, \\
H_7^0 &= \texttt{0x5BE0CD19}.
\end{aligned}
$$



3. Specification of ARIRANG (8)

COUNTER ADDITION updates the $(i-1)$th intermediate hash value, H_{i-1}, with the Counter Value, Ctr^i.

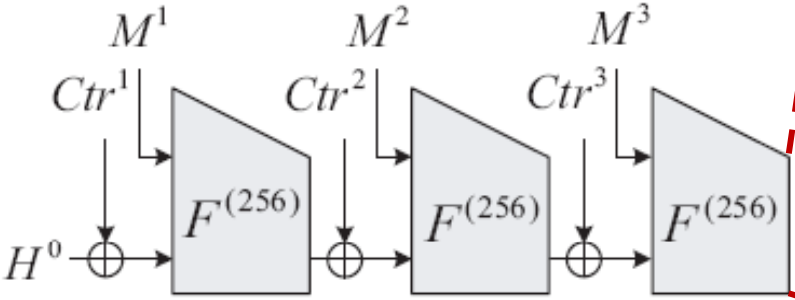
$$\begin{aligned}&\text{ARIRANG256\_CounterAddition}(H^{i-1}, Ctr^i)\\&\{\\&\quad H_0^{i-1} = H_0^{i-1} \oplus Ctr_0^i\\&\quad H_4^{i-1} = H_4^{i-1} \oplus Ctr_1^i\\&\}\end{aligned}$$

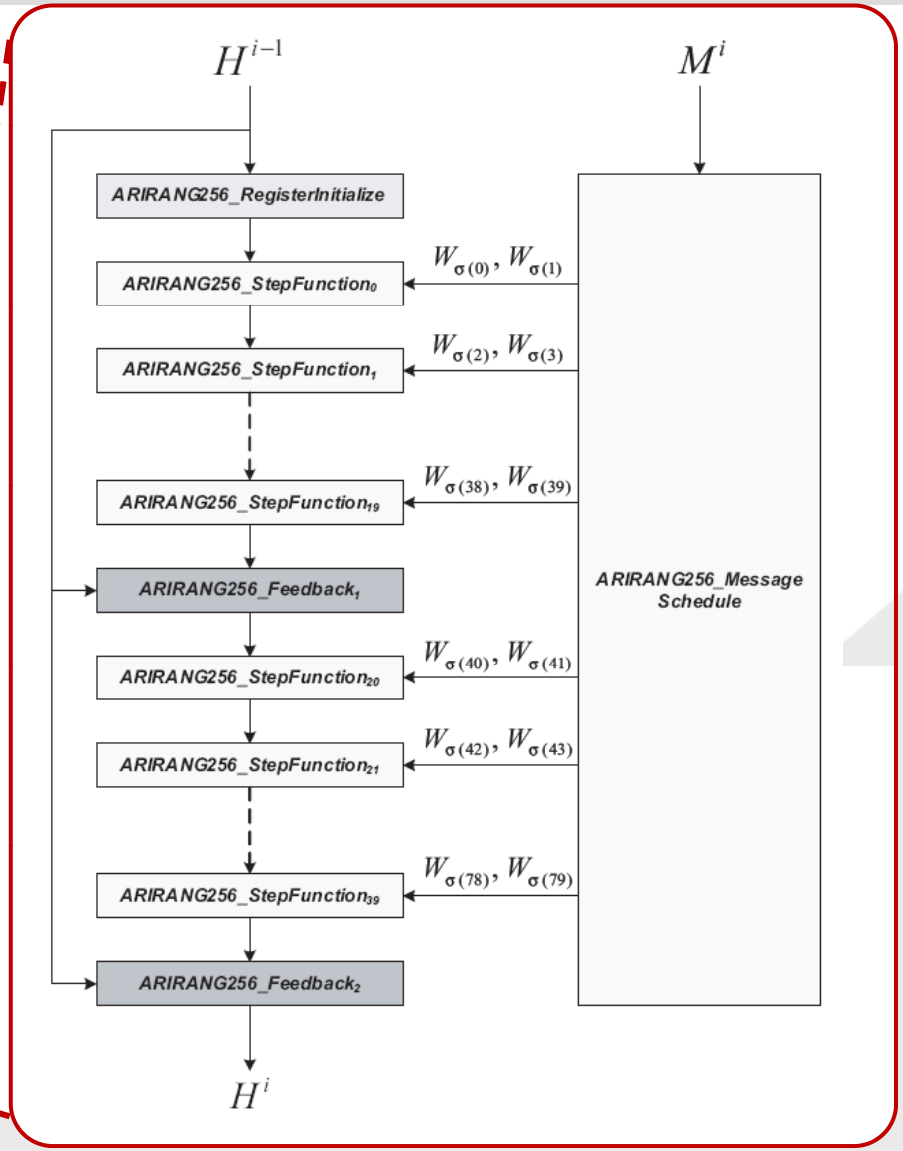


3. Specification of ARIRANG (9)

COMPRESSION FUNCTION consists of four functions.

1. MessageSchedule
2. RegisterInitialize
3. StepFunction
4. Feedback





3. Specification of ARIRANG (10)

MESSAGE SCHEDULE consists of the message word expansion and the message word ordering.

Message Word Expansion

message words $M^i_0, M^i_1, \ldots, M^i_{15}$ → Expanded message words $W_0, W_1, W_2, \ldots, W_{31}$

EXAMPLE

$$\text{For } t = 0 \text{ to } 15$$
$$W_t = M^i_t$$

$$W_{16} = (W_9 \oplus W_{11} \oplus W_{13} \oplus W_{15} \oplus K^{(256)}_0) \lll 5$$
$$W_{17} = (W_8 \oplus W_{10} \oplus W_{12} \oplus W_{14} \oplus K^{(256)}_1) \lll 11$$
$$W_{18} = (W_1 \oplus W_3 \oplus W_5 \oplus W_7 \oplus K^{(256)}_2) \lll 19$$
$$W_{19} = (W_0 \oplus W_2 \oplus W_4 \oplus W_6 \oplus K^{(256)}_3) \lll 31$$

3. Specification of ARIRANG (11)

MESSAGE SCHEDULING consists of the message word expansion and the message word ordering.

MESSAGE WORD EXPANSION

+

MESSAGE WORD ORDERING

s		1 round		2 round		3 round		4 round	
		σ(s)		σ(s+20)		σ(s+40)		σ(s+60)	
0	1	16	17	20	21	24	25	28	29
2	3	0	1	3	6	12	5	7	2
4	5	2	3	9	12	14	7	13	8
6	7	4	5	15	2	0	9	3	14
8	9	6	7	5	8	2	11	9	4
10	11	18	19	22	23	26	27	30	31
12	13	8	9	11	14	4	13	15	10
14	15	10	11	1	4	6	15	5	0
16	17	12	13	7	10	8	1	11	6
18	19	14	15	13	0	10	3	1	2

3. Specification of ARIRANG (12)

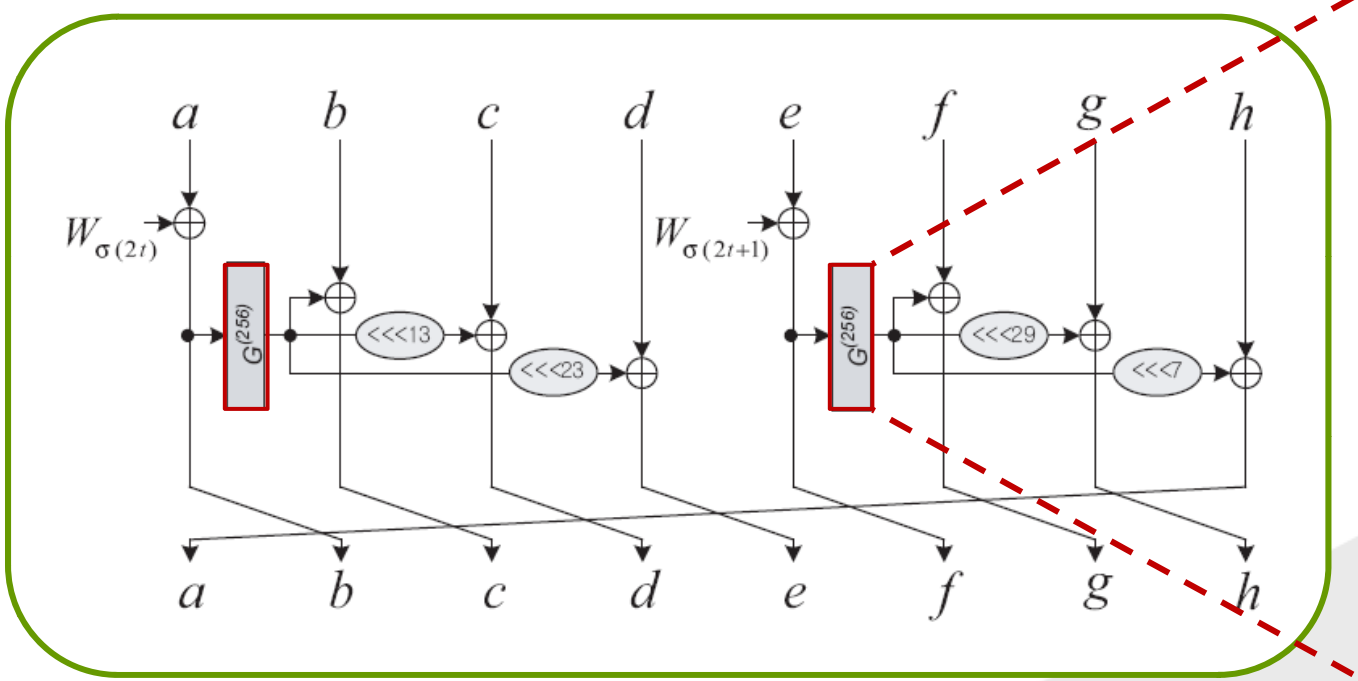
Register Initialize initializes the eight working variables, *a, b, c, d, e, f, g,* and *h*, with the (*i* – 1)st intermediate hash value, H^{i-1}

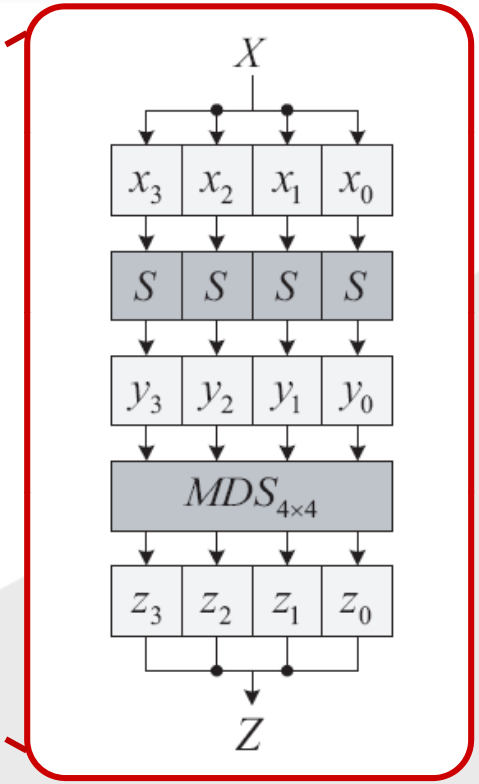
$$
\begin{aligned}
&\text{ARIRANG256\_RegisterInitialize}(H^{i-1},\ a,\ b,\ c,\ d,\ e,\ f,\ g,\ h)\\
&\{\\
&\qquad a = H_0^{i-1},\ b = H_1^{i-1},\ c = H_2^{i-1},\ d = H_3^{i-1}\\
&\qquad e = H_4^{i-1},\ f = H_5^{i-1},\ g = H_6^{i-1},\ h = H_7^{i-1}\\
&\}
\end{aligned}
$$

3. Specification of ARIRANG (13)

StepFunction

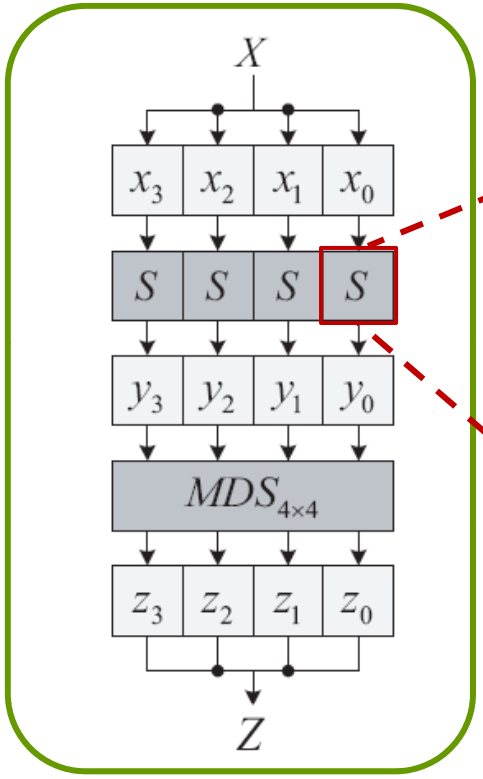
updates the eight working variables, *a, b, c, d, e, f, g*, and *h*, with the function $G^{(256)}$ and the two input message words, $W_{\sigma(2t)}$, $W_{\sigma(2t+1)}$.





3. Specification of ARIRANG (14)

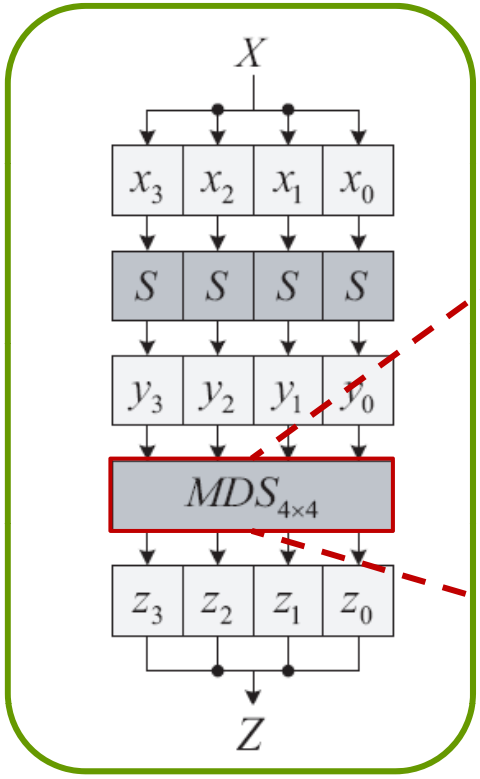
S-BOX is the same as the SubBytes transformation of AES.



S(x||y)	y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

3. Specification of ARIRANG (15)

MDS is the same as the MDS transformation of AES.

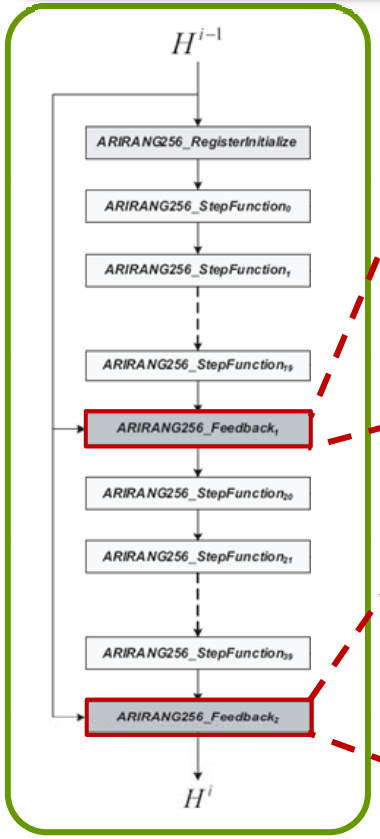


$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

3. Specification of ARIRANG (16)

Feedback

updates the eight working variables, *a, b, c, d, e, f, g*, and *h*, with H^{i-1}.



$$\begin{aligned}&\text{ARIRANG256\_Feedback}_1(H^{i-1}, a, b, c, d, e, f, g, h)\\&\{\\&\quad a = a \oplus H_0^{i-1},\ b = b \oplus H_1^{i-1},\ c = c \oplus H_2^{i-1},\ d = d \oplus H_3^{i-1}\\&\quad e = e \oplus H_4^{i-1},\ f = f \oplus H_5^{i-1},\ g = g \oplus H_6^{i-1},\ h = h \oplus H_7^{i-1}\\&\}\end{aligned}$$

$$\begin{aligned}&\text{ARIRANG256\_Feedback}_2(H^i, H^{i-1}, a, b, c, d, e, f, g, h)\\&\{\\&\quad H_0^i = a \oplus H_0^{i-1},\ H_1^i = b \oplus H_1^{i-1},\ H_2^i = c \oplus H_2^{i-1},\ H_3^i = d \oplus H_3^{i-1}\\&\quad H_4^i = e \oplus H_4^{i-1},\ H_5^i = f \oplus H_5^{i-1},\ H_6^i = g \oplus H_6^{i-1},\ H_7^i = h \oplus H_7^{i-1}\\&\}\end{aligned}$$

4. Security Analysis (1)

SECURITY ARIRANG has strong resistance to existing attacks.

Attack	Security	Attack	Security
Collision attack	$2^{n/2}$	First preimage attack	2^{n}
Near collision attack	$2^{n/2}$	Second preimage attack	2^{n}
Pseudo collision attack	$2^{n/2}$	Indifferentiability	$2^{n}/N(3N-1)$

- **Resistance to**

Local collision attack	Fixed-Point-Finding Attack	Length-Extension Attack
Multicollision Attack	Slide Attack	Trapdoor-based Attack

4. Security Analysis (2)

SECURITY ARIRANG supports HMAC, PRF, and Randomized Hashing.

◆ Formal security proofs of the domain extension of the ARIRANG family

◆ Formal security proofs of

- HMAC-ARIRANG
- two PRF constructions based on ARIRANG
- randomized Hashing with ARIRANG

5. Implementation (1)

ENVIRONMENT of software implementations

- Intel Core 2 Duo Processor 2.53GHz
- 2GB RAM
- ANSI C compiler in the Microsoft Visual Studio 2005

32 bit processor	Windows Vista Ultimate 32 bit
64 bit processor	Windows Vista Ultimate 64 bit

5. Implementation (2)

EFFICIENCY

Software implementation results of ARIRANG-256/512 and SHA-256/512 (Cycle/Byte)

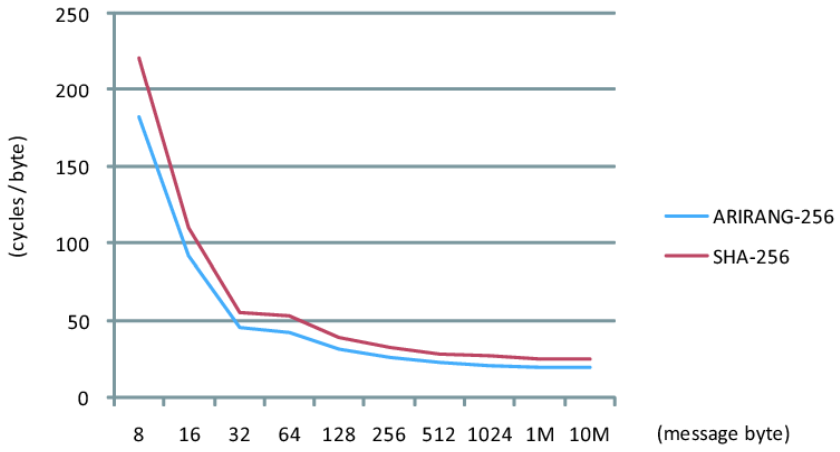
(Cycle/Byte)		Message Length (Byte)						
		8	32	64	256	1024	1M	10M
32-bit Oriented	ARIRANG-256	182.8	46.3	42.6	26.4	21.6	20	20.1
	SHA-256	220.9	55.8	53.4	32.6	27.1	25.4	25.4
64-bit Oriented	ARIRANG-512	210.5	52.4	25.9	17.7	12.9	11.2	11.3
	SHA-512	241.1	58.8	29.6	21.7	16.1	14.2	14.3

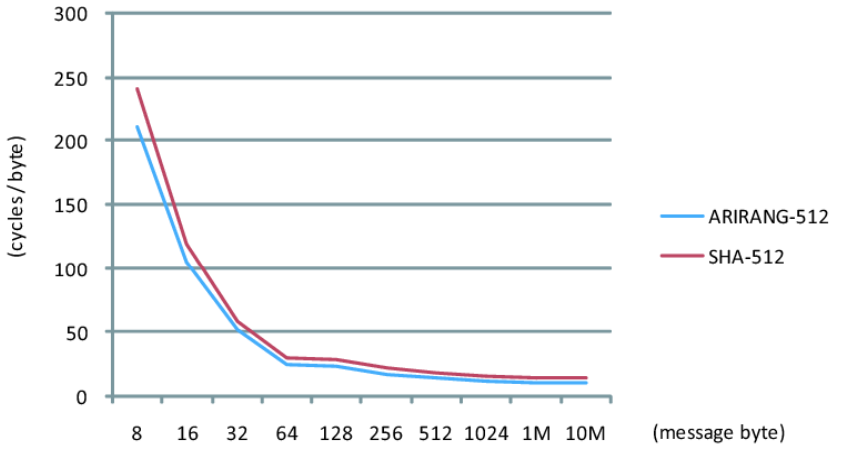
5. Implementation (3)

EFFICIENCY

Software implementation results of ARIRANG-256/512 and SHA-256/512 (Cycle/Byte)

The speed of ARIRANG-256/512 is about 25% faster than SHA-256/512.





Q&A