

This work was supported in part by a consignment research from the National Institute on Information and Communications Technology (NiCT), Japan. This work was supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government.

MAME: A compression function with reduced hardware requirements

ECRYPT Hash Workshop 2007

Barcelona, Spain

May 24, 2007

Hiroataka Yoshida, Dai Watanabe, Katsuyuki Okeya, Jun Kitahara

Systems Development Laboratory, Hitachi, Ltd

Hongjun Wu, Ozgul Kucuk, Bart Preneel

Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC



Overview

- The background of our research
 - Hash functions
 - Security properties
- Compression function MAME
 - Design principle
 - Specification
 - Security evaluation
 - SW/HW Performance
- Conclusion



The background for our research

- Ubiquitous systems
- Authentication technology
 - HMAC
- Hash function
 - An algorithm that takes input strings of arbitrary length and maps these to short fixed length output strings.
 - Security properties:
 - Pre-image resistance: it is computationally infeasible to find any input which hashes to any pre-specified output.
 - Second pre-image resistance: it is computationally infeasible to find any second input which has the same output as any specified input.
 - Collision resistance: it is computationally infeasible to find a collision, i.e. two distinct inputs that hash to the same result.



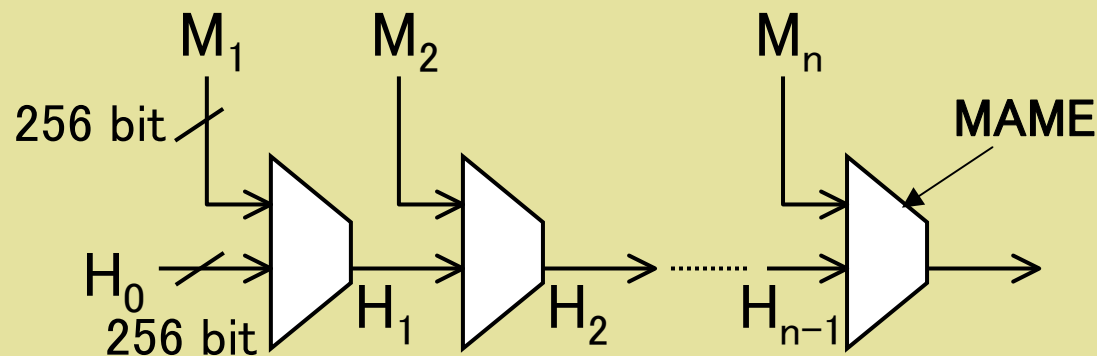
Results on Analysis of Known Hash Functions

- SHA-1
 - Collision attack (2005)
 - Slide attack on the underlying block cipher (2003)
- MD5
 - Collision attack (2005)
 - Attack on the compression function ('94)

- SHA-256
 - 256-bit hash length
 - Initial evaluation(2003)
 - Attack on SHA-256-XOR compression function with 34-rounds out of 64 (2005)
 - Collision path format for the reduced SHA-256 (2006)
- Whirlpool
 - 512-bit hash length
 - Evaluation on the maximum differential characteristic probability
 - Non-randomness property in Whirlpool with 6 rounds
- Tiger
 - 192-bit hash length, collision attack on Tiger with 19 rounds out of 24 (2006)

Our proposal: MAME

- A compression function aimed to realize 256-bit hash functions
- 256-bit chaining variable
- 256-bit message block
- 256-bit output
- High security level regarding collision resistance
 - Secure against differential style attacks (e.g. attacks on SHA-1)
- Compact in hardware
 - One of the smallest hardware implementation of 256-bit hash, today



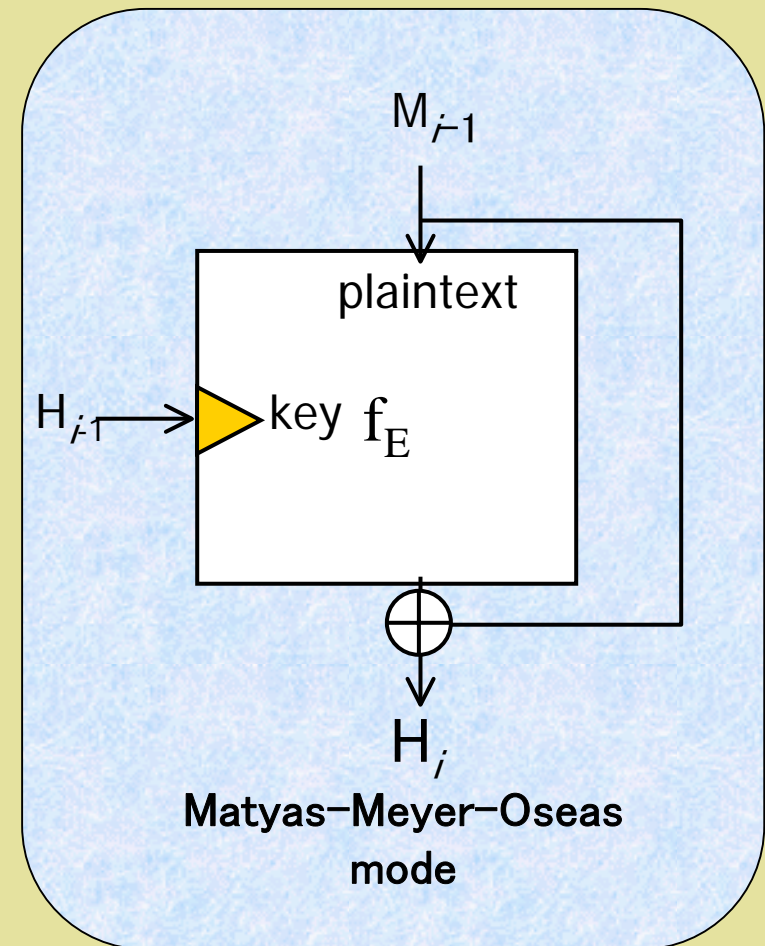


Design principle of MAME

- Requirements as a design goal
 - The gate count of its hardware implementation is small
 - Its software performance on servers is practical
 - Security evaluation on known attacks is possible
- Our approach to achieve the goal
 - Limit ourselves to relatively simple operations such as XORs to reduce hardware complexity
 - Ensuring security by increasing the number of rounds

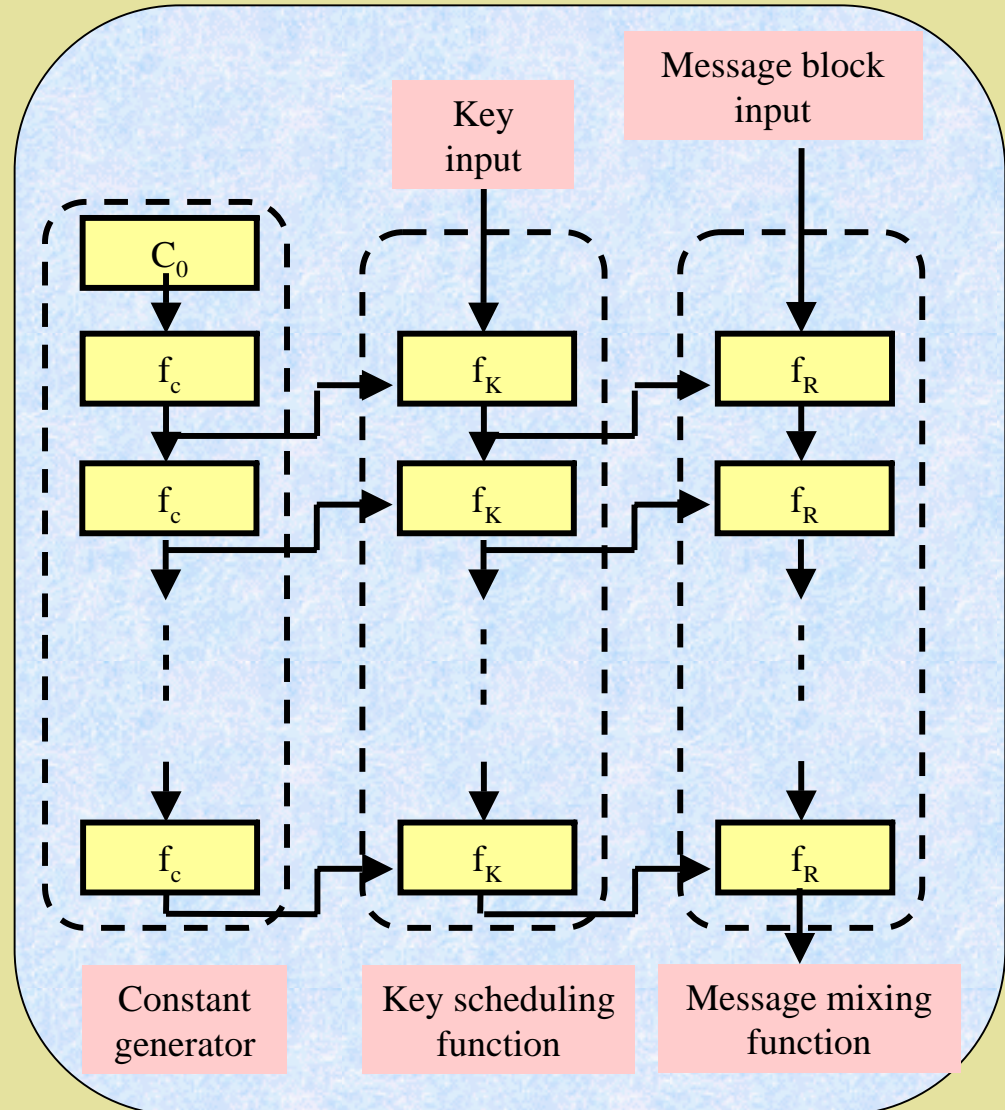
Construction of the compression function

- Adoption of MMO (Matyas-Meyer-Oseas(MMO)) mode
 - Techniques to evaluate the security of Block ciphers are applicable
 - The input over which the attacker can have direct control is plaintext
 - Cf, SHA-1: The input over which the attacker can have direct control is key
 - Implementations of HMAC based on the hash function is secure

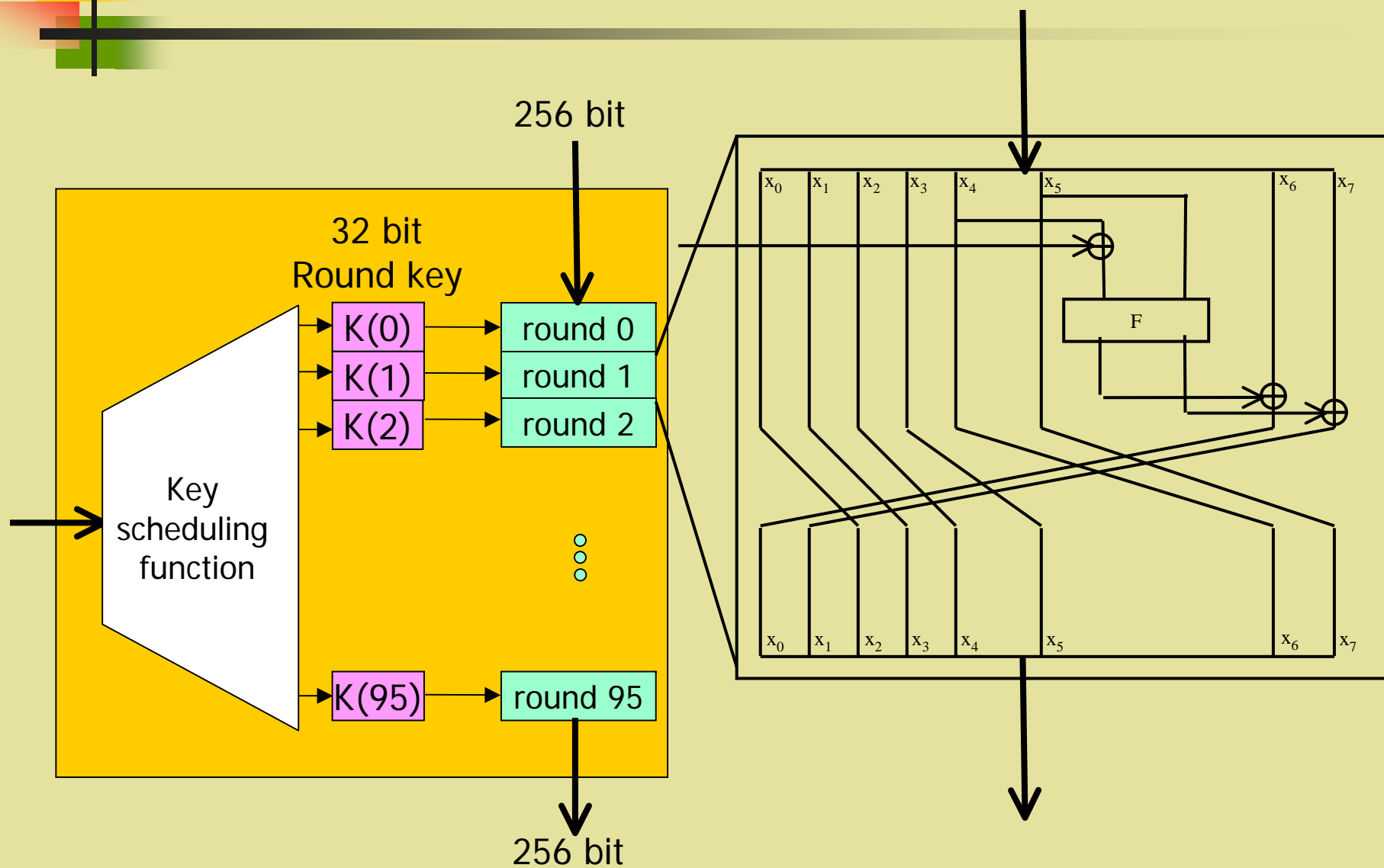


High level view of the block cipher f_E

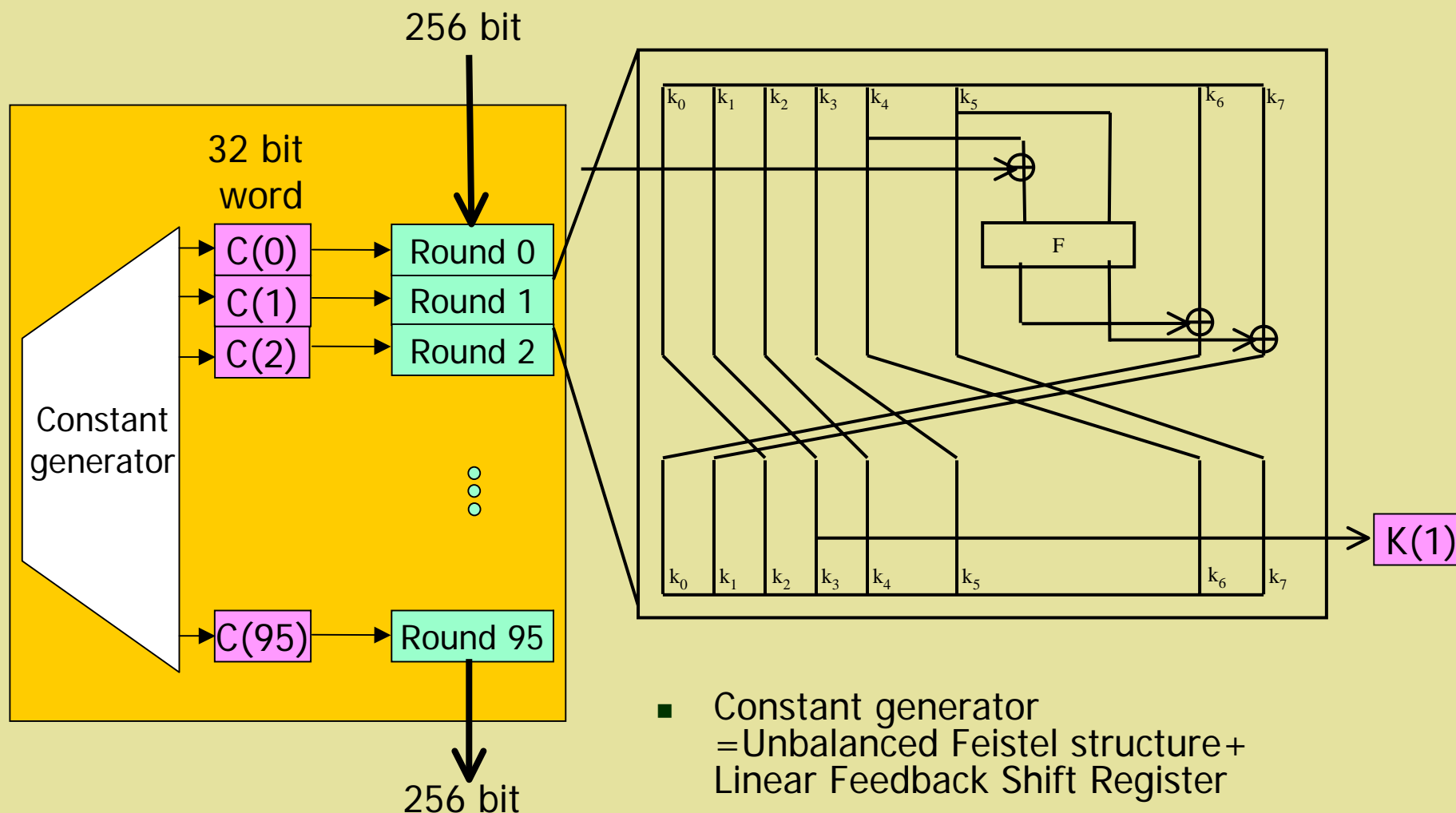
- Key and block lengths are 256 bit
- Message mix function f_R
 - Unbalanced Feistel network
 - The round number is 96
 - F function is a composition of linear diffusion layer L and S-box layer S
- The key scheduling function f_K
 - Almost same as the f_R
 - Constant addition instead of key addition



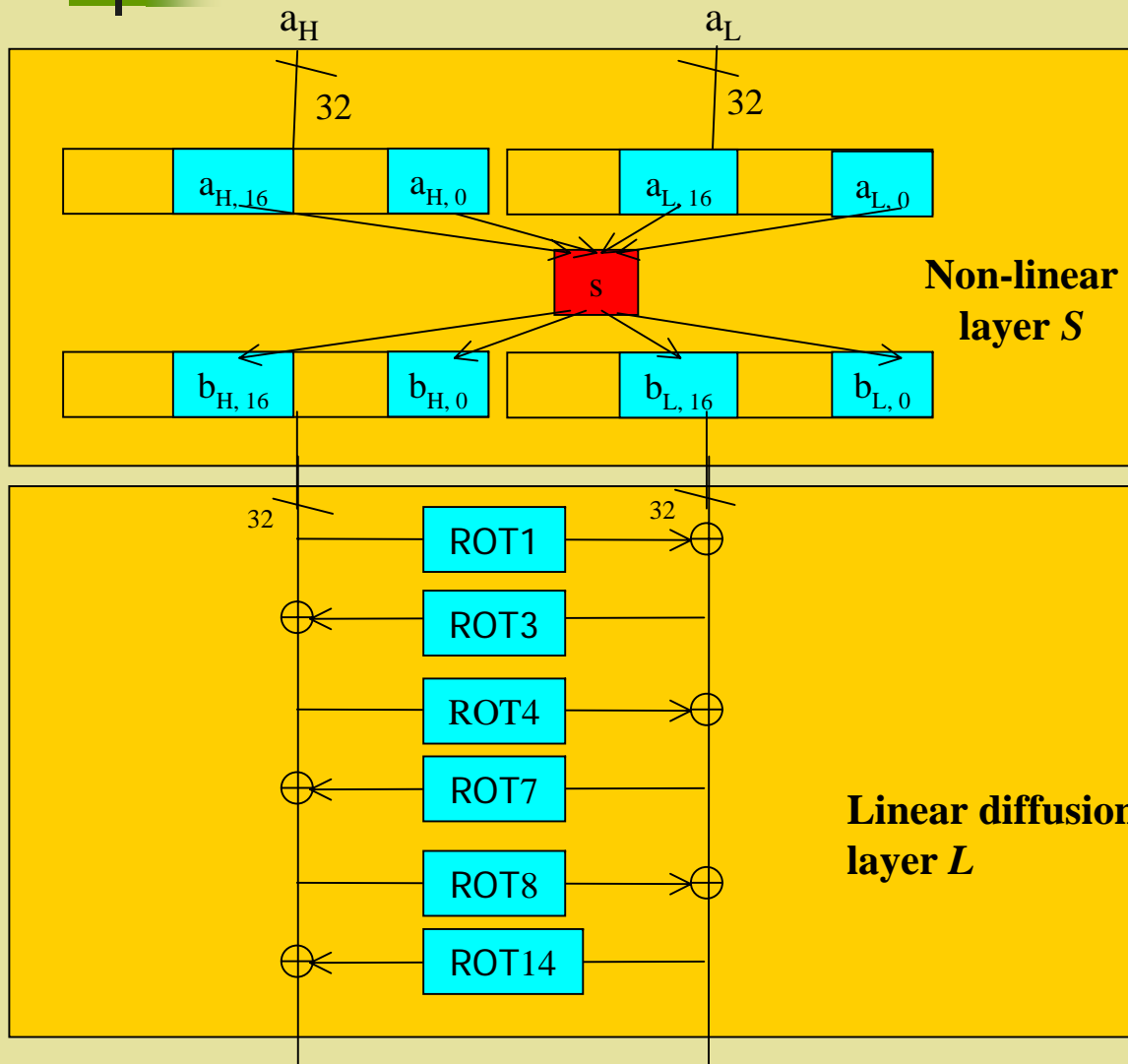
The Round function



The round key generation function



The F function

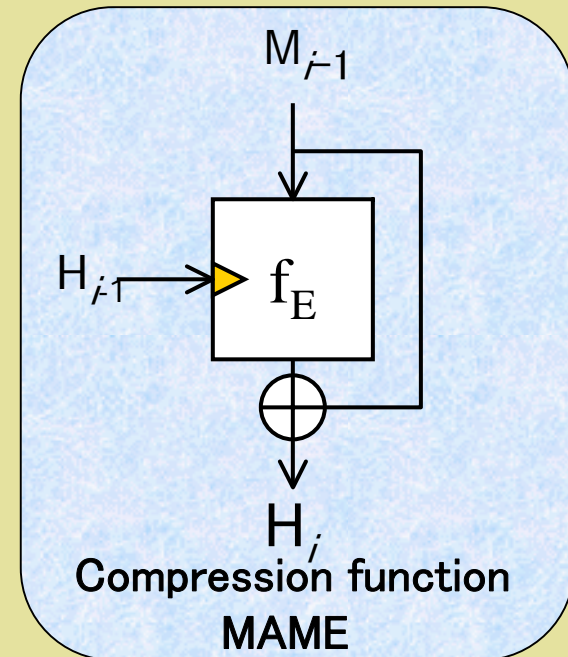


$$S[16] = \{4, 14, 15, 1, 13, 9, 10, 0, 11, 2, 7, 12, 3, 6, 8, 5\}$$

$$Ax^{-1} + C$$

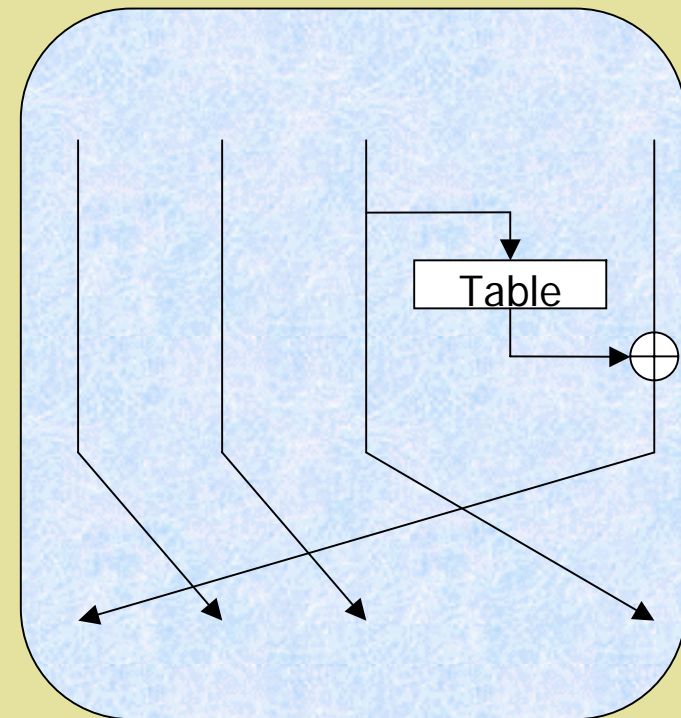
Security evaluation on MAME

- We evaluate the block cipher f_E regarding major attacks on block ciphers
 - **Differential attacks**, Linear attacks
- Our method: Global-local approach
 - = firstly, local (F function) evaluation
 - secondly, global (overall f_E) evaluation
- Our focus is differential attacks
 - Attacks on SHA-1 and MD5 are essentially differential attacks
 - Evaluation from the designers point of view:
 - Consider *all* the differential paths, evaluate the bound for the probabilities
 - Evaluation from the attackers point of view:
 - Construct a specific differential path: prob. 2^{-700} , for 64 rounds



Evaluation on MAME regarding differential attacks

- Evaluation goal:
 - f_E is secure against differential attacks
iff Probabilities for **all** the differential characteristic $< 2^{-256}$
iff maximum differential characteristic probability $< 2^{-256}$
- Evaluation method: Global-local approach
 - Preparation: data truncation
 - The space of differences: 256 bit
 - The space of the Hamming weights of the differences X_{Ham} : 20 bit
 - Local evaluation: F function evaluation
 - Compute maximum differential probability for S-box: 2^{-2}
 - Evaluation on the linear diffusion layer: make the branch table
 - Global evaluation: evaluation on f_E
 - Minimum number of Active S-box $D_{\text{min}} > 128$



Local evaluation (Linear diffusion layer)

- Make a table which represents the transition from the Hamming weight of the input difference and the output difference via linear diffusion layer.

Hamming weights of output differences

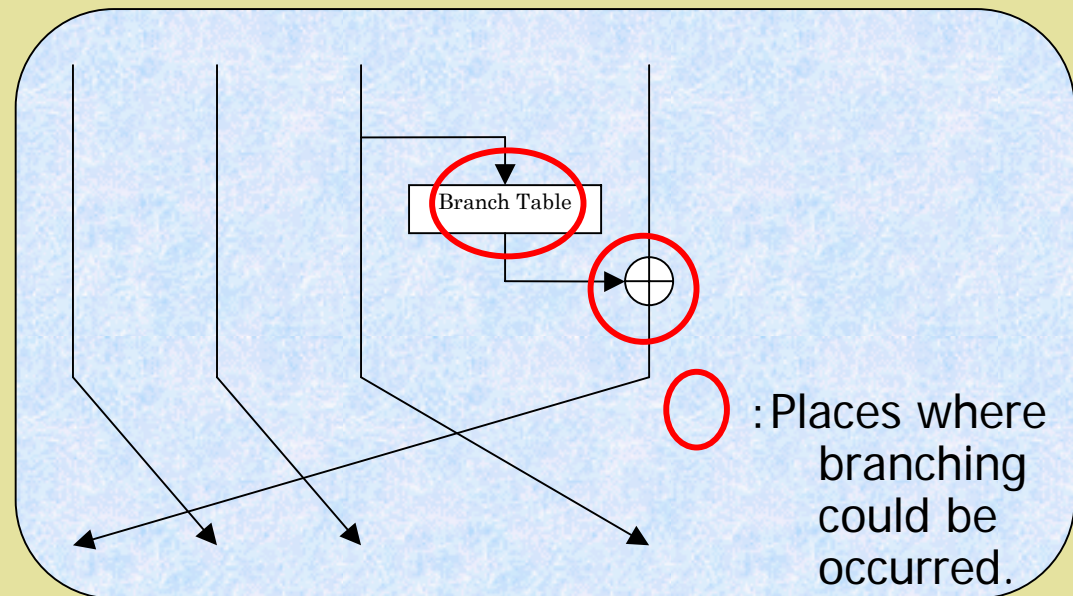
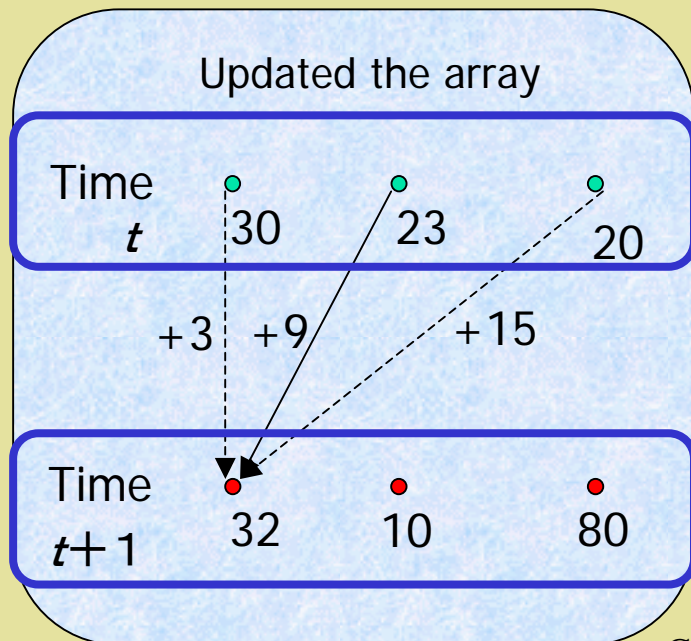
		Branch table																
Hamming weights of input difference		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1
2	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
3	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
...															
15	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- 0: impossible transition
- 1: otherwise

Global evaluation

■ Apply the Viterbi algorithm

- Prepare an array consisting of X_{Ham} elements per round
- Each element in the array stores minimum number of active S-box
- Updated the array round the number times.
- After 96 rounds, the element with minimum number in the array is D_{min}

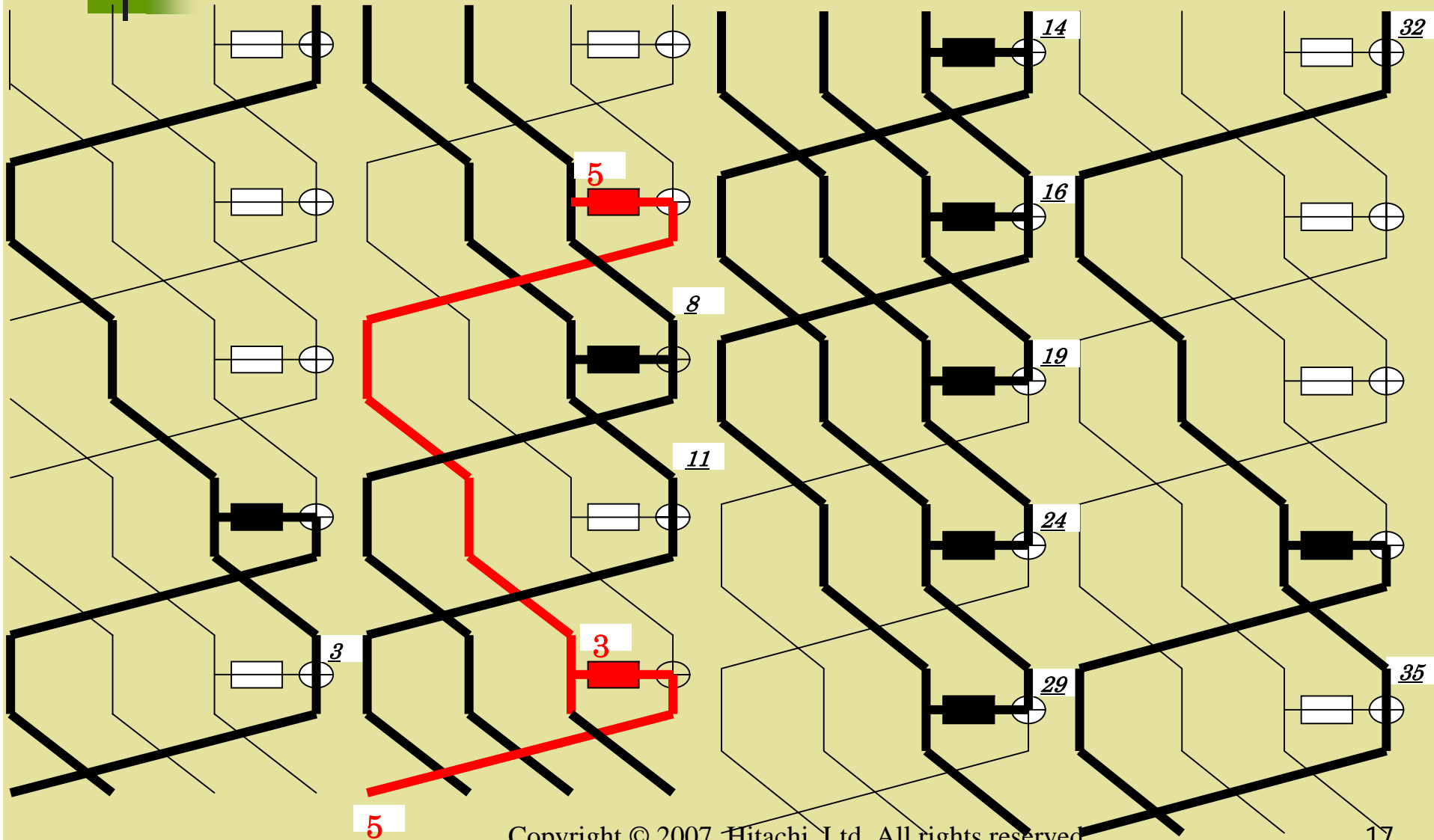




Experimental result

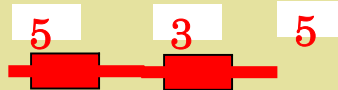
# rounds	D_{min}	# rounds	D_{min}	# rounds	D_{min}
0	0	32	62	64	127
1	0	33	62	65	128
2	0	34	65	66	131
3	1	35	66	67	132
7	11	39	76	71	142
11	15	43	83	75	149
15	27	47	92	79	158
19	33	51	99	83	165
23	43	55	109	87	175
27	50	59	116	91	182
31	59	63	125	95	191

Improve the evaluation by analyzing the best differential path our algorithm found



Result on the improvements

- Our experiments show that the following branching pattern is impossible in practice.



- New search for differential characteristics excluding the above pattern Improved the experimental result.

	<i># of rounds s.t. $D_{min} > 128$</i>	Security margin for differential attacks
Initial evaluation	66	30
Improved evaluation	58	38

Hardware performance comparison

- We evaluate hardware performance of MAME and SHA-256
- Our implementation of MAME:
 - It takes advantage of the use of logical operations in the most part of the design
 - f_K and f_R share the same circuit and processing one round takes one cycle.

	Throughput	Gate count	Clock Frequency
MAME	* 440 Mbps	* 8.2 KGate	300 MHz
SHA-256	* 2600 Mbps	* 18.0 KGate	300 MHz

* Hitachi 0.18 μ m technology



Software performance comparison

- We evaluate software performance of MAME and SHA-256
- Our implementation of MAME:
 - We partially unroll the round functions code to increase the speed.
 - We take a known approach to achieve a bit slice implantation where S-box is transformed into 20 logical operations.

	Time(ms)	RAM (Bytes)
MAME	*49.4	*96
SHA-256	*31.4	*128

*Microcomputer for IC cards



Conclusion

- We presented a new compression function, MAME designed for a hardware oriented hash function.
- We make it clear what the design rational we adopt and evaluate its security applying techniques from block cipher analysis and confirm that there is no weakness in MAME. Our implementation shows some sort of compactness of MAME but this leaves room for further optimizations.