On the Design and Use of Lightweight Cryptography for Cyber-Physical Systems

Hirotaka Yoshida[1]

[1]AIST, Japan

Kolkata, India (16 November 2018)

CPSEC | Cyber Physical Security Research Center

Table of contents

Cyber-physical systems (CPS)

- Cyber-physical systems (CPS) are systems that connect information with physical objects:
  auto-motives, factory automation, energy harvesting, medical devises
- The security in these systems could be safety-critical,
- For deployment of lightweight symmetric cryptography in CPS, problems can be bridging the gap between industry requirements and the publicly-available academic results

A Cyber-Phisical System: Automotives

In-vehicle system

- Short-message performance important:
  - Packets are as short as 8 bytes (CAN) to 64 bytes (CAN-FD).
  - Realtime req. is severe: 1–100ms periodic tasks are processed.
- 50–100 ECUs are employed in a car:
  - Limited cost can be paid for each ECU.
  - Cost comes from circuit size in HW and RAM/ROM size in SW.

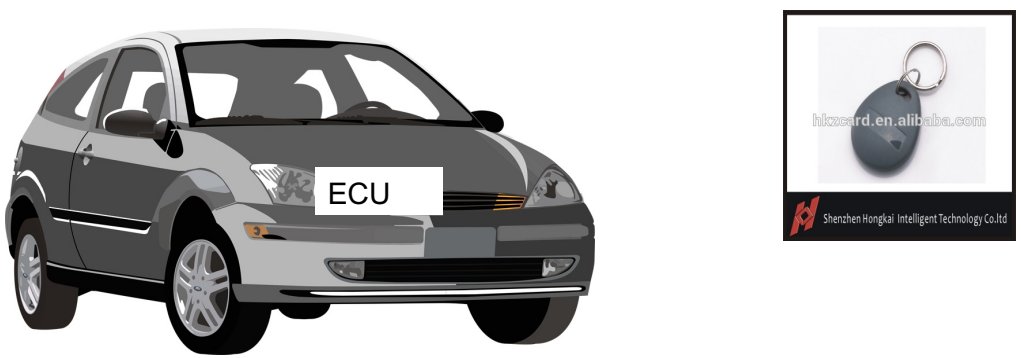


Figure: Cyber Physical

PKES Hacking (2010)

- Tillich, S. and Wójcik, M.: Security Analysis of an Open Car Immobilizer Protocol Stack, Presented at the industry track of the 10th International Conference on Applied Cryptography and Network Security (ACNS'12), (2012)..

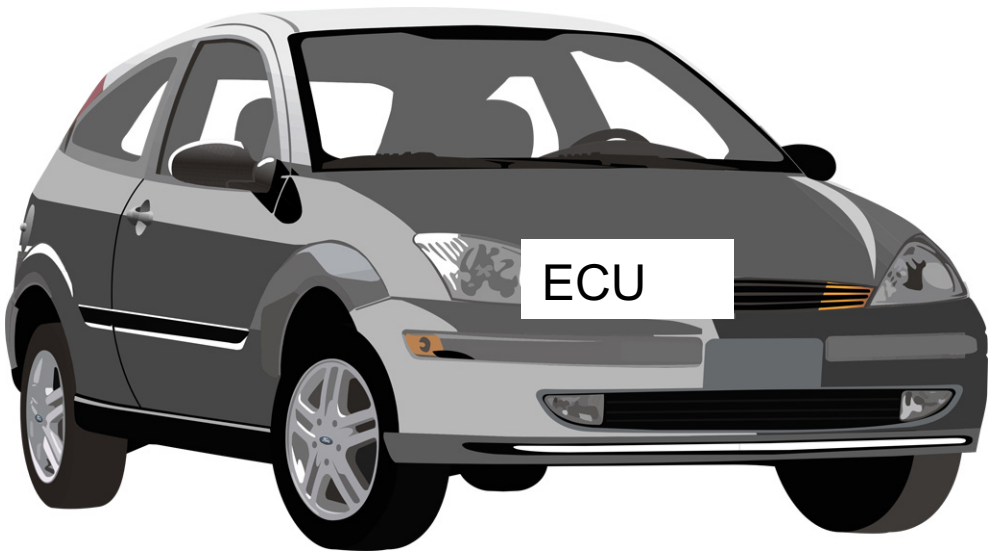
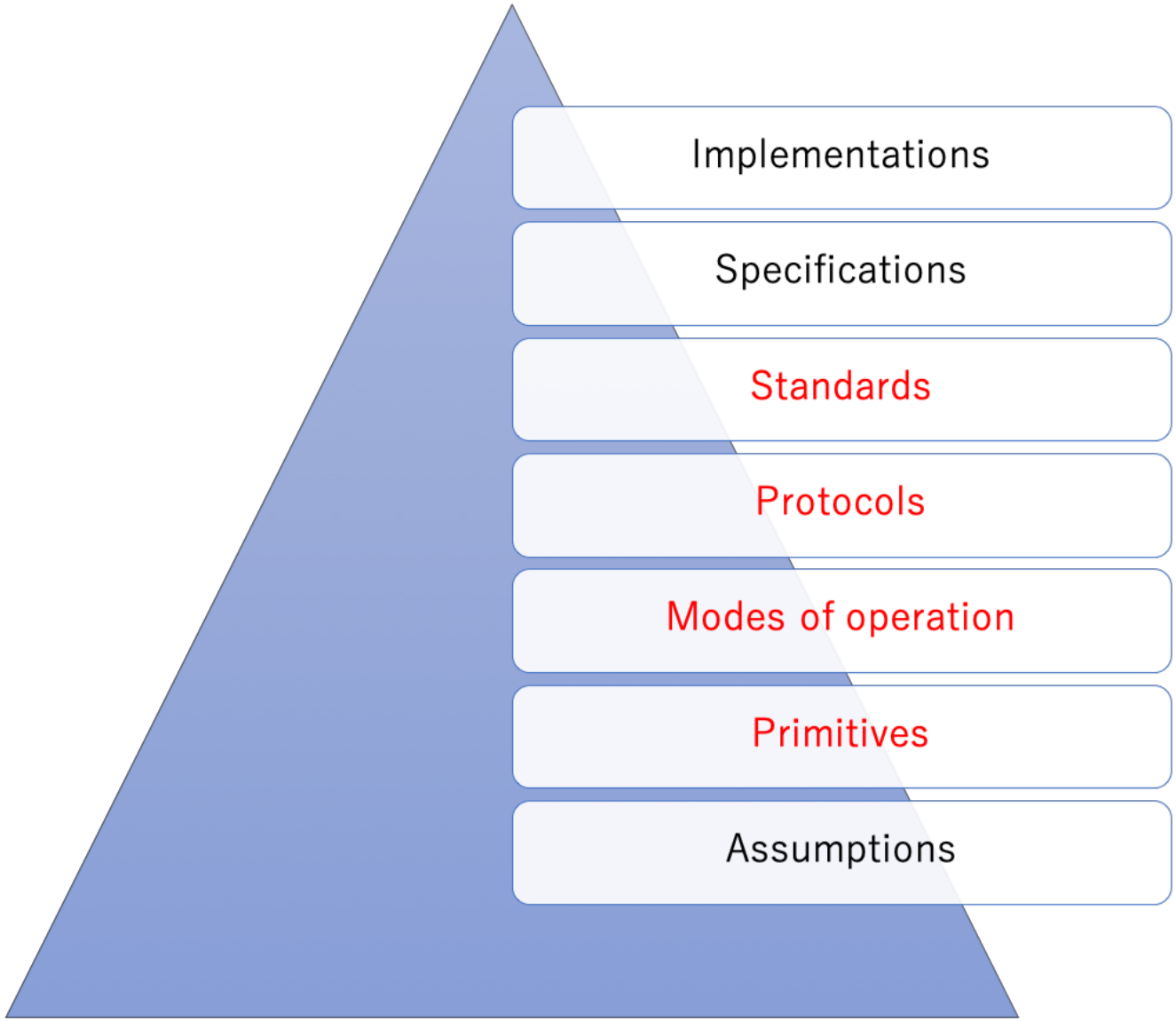


Figure: A Car and Key Fob

Crypto Stack



Lightweight cryptography

- Growing demand for applications using smart devices: low-end micro-controllers and RFID tags
- Security problems such as confidentiality, data authentication and privacy
- Challenge: design cryptographic primitives or protocols that meet the system requirements
- To meet these requirements, lightweight cryptographic algorithms can be implemented under restricted resources, such as low-cost, low-energy, or low-power environments

Importance of hash functions

- Used in a wide variety of cryptographic applications:
  - Digital Signature Schemes
  - Key Derivation Function
  - Deterministic Random Bit Generators
  - Message Authentication
- Achieve security in these cryptographic applications
- Standardized in ISO/IEC and NIST
- Needed in any cryptographic software library:
  - Randomness extraction
  - Public key encryption

What is a hash function?

- Maps input strings to short output strings of fixed length
- n-bit hash function returns an n-bit hash value
- The description of hash function must be publicly known
- Does not require any secret information for its operation.

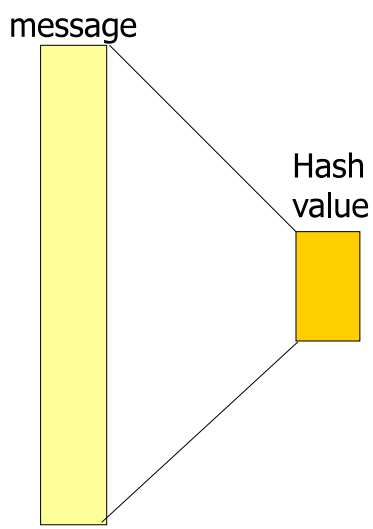


Figure: Hash function

Hash Functions' properties

Hash functions' properties expected in cryptographic applications

- Security property:
  - Preimage resistance
  - Second preimage resistance
  - Collision resistance
  - Indifferentiability from a random oracle
- Performance:
  - Efficiency
  - Hardware/Software implementation flexibility

Hash function crisis (2004-2005)

- Overview of the crisis
  - 2004: MD4 attack by hand
  - 2005: cryptanalysis of hash functions: MD5 and SHA-1.
  - 2006, Federal agencies should stop using SHA-1 for certain applications must use the SHA-2 family for them after 2010.
  - NIST recommends the transition from SHA-1 to SHA-2
  - SHA-2 may be vulnerable to similar techniques
  - Similarities in the design principles between SHA-2 and SHA-1
- The Breakthrough: Wang et al.'s Differential collision search
  - Attack complexity optimization together with differential cryptanalysis
  - Biham and Shamir, Differential Cryptanalysis of the Data Encryption Standard, 1993.

General concepts: Differential cryptanalysis

- i-round characteristic is defined as $(\alpha, \beta_1, \beta_2, ..., \beta_i)$ considered as possible values of $(d(X, X'), d(Y_1, Y_1'), d(Y_2, Y_2'), ..., d(Y_i, Y_i'))$.
- The probability of an i-round characteristic is defined as

$$\Pr[d(Y_1, Y_1') = \beta_1, d(Y_2, Y_2') = \beta_2 ..., d(Y_i, Y_i') = \beta_i) | d(X, X') = \alpha]$$

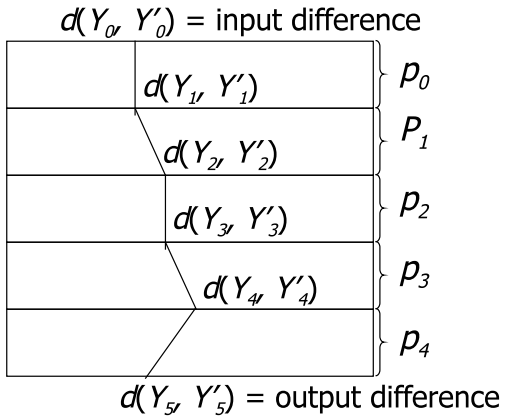


Figure: A differential characteristic (path).

- The aim is to find differential characteristics for the whole cipher, for which probability is significantly higher than 2^{-m} (m: block length).

Introduction to MAME (2005-2007)

- Overview of MAME
  - Hardware-oriented lightweight design requiring 8.2 Kgates.
  - 256-bit hash function

Figure: MAME: bean in Japanese

The underlying block cipher E

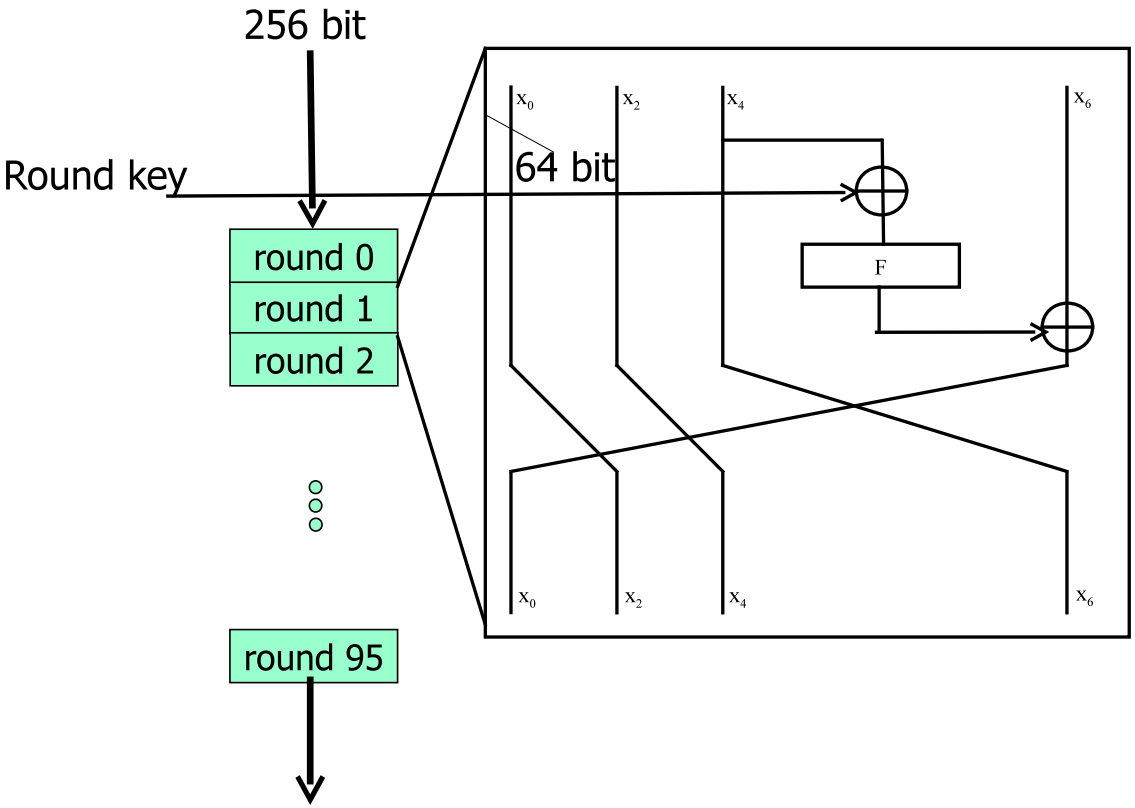
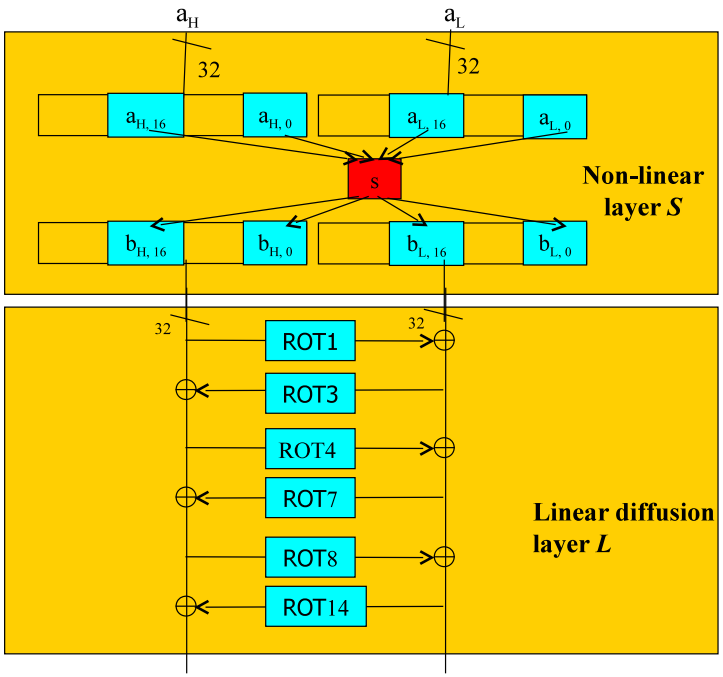


Figure: round function.

F function



- F consists of the non-linear function with 16 4-bit S-boxes and the linear transformation $\mathcal{L}$.

Differential cryptanalysis by the Viterbi algorithm

- The Viterbi algorithm is a recursive optimal solution to the problem of estimating the state sequence of a discrete-time finite-state Markov process observed in memoryless noise
- Application to MAME
  - d_r^i: the distance of a state i at round r
  - t_{ij}: the number of active S-boxes which has been increased through an application of the r-th round.
  - Then $d_{r+1}^j = d_r^i + t_{ij}$

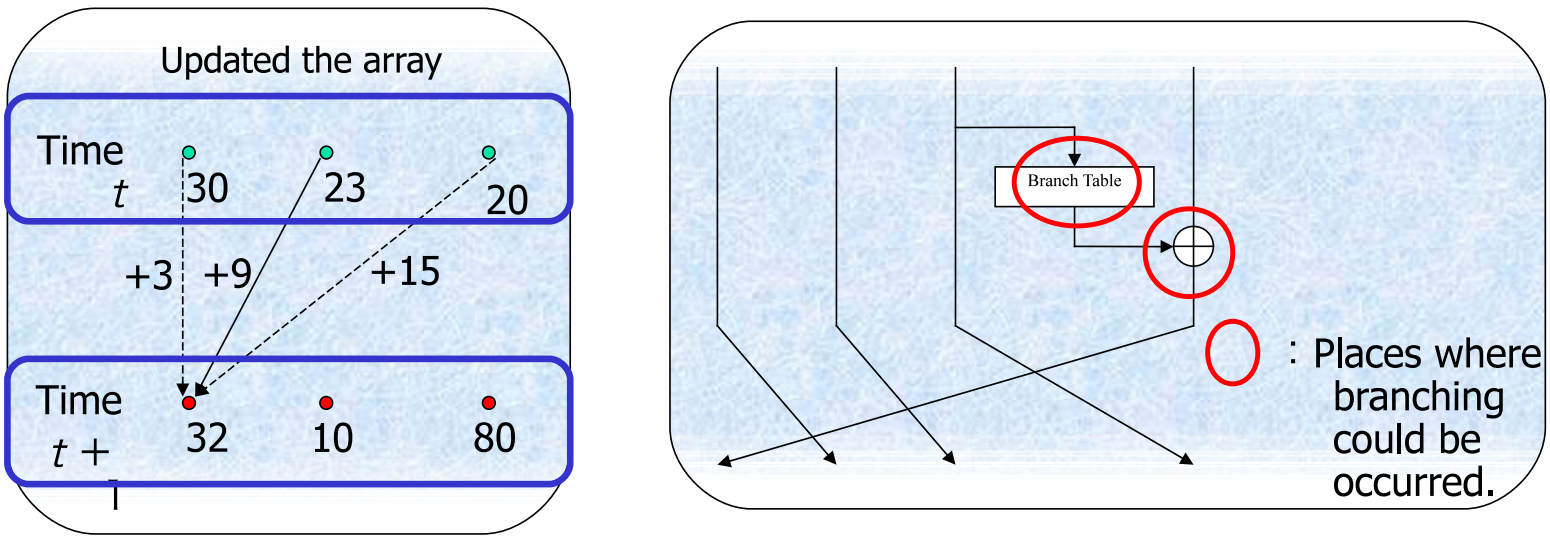


Figure: Computing lower bound of of active S-boxes

Online phase: apply the Viterbi algorithm

- Each state might be defined as a 256-bit difference in the internal state.
- Memory requirement of about 2^{256} bits, which is impractical.
- Truncate a 64-bit word x_i into a 16-bit value $\tilde{x_i}$ by considering 4 input bits of an S-box as a single bit
- Ham($\tilde{x}$) ranges from 0 to 16 and it can be represented as a 5-bit string
- Results in a small memory (2^{20})

Offline phase result: table representing the difference propagation through $\mathcal{L}$

$Ham(\mathcal{L}(\tilde{x}))$	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
1	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	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
4	0	0	0	0	0	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
6	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
7	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	0	0	1	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
16=$Ham(\tilde{x})$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- It took us several hours on 4 PCs with a Xeon processor running at 2 GHz to perform the experiments.

Toward improvements of bounds

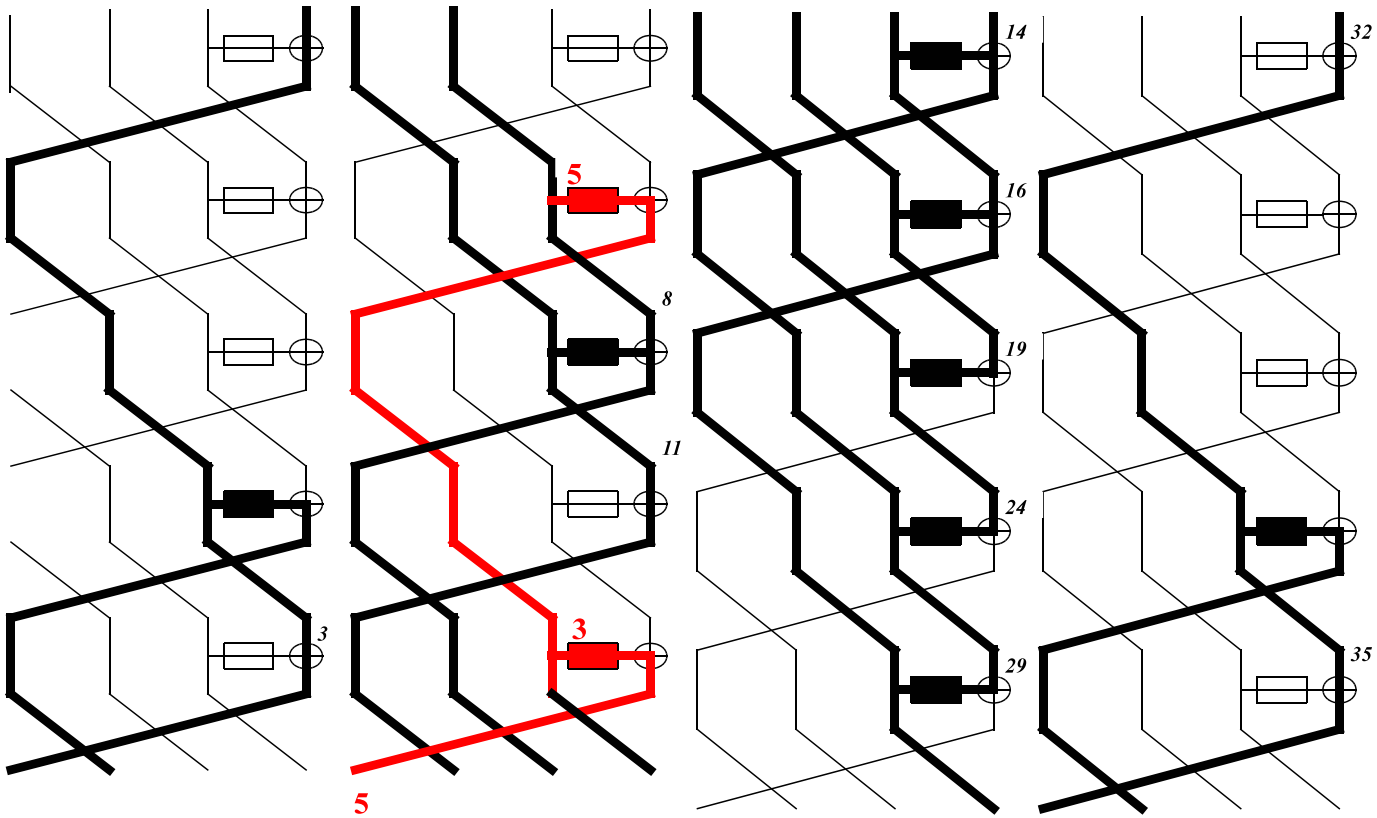


Figure: A best differential path

- $D_{min} > 130$ for MAME reduced to 58 rounds out of 96.

The NIST SHA-3 Competition (2007-2012)

- Overview of the competition
  - 51 candidates to advance to the first round in December 2008
  - 14 to advance to the second round in July 2009
  - 5 finalists - BLAKE, Grøstl, JH, Keccak, and Skein
  - NIST selected Keccak as the winning algorithm on October 3, 2012
- Lessson learned from our submission, Lesamnta
  - Stay at only first round
  - Compression function attack due to too simple round-constant
  - Not broken as full hash
  - One of the smallest RAM

The design goals of Lesamnta-LW

- Compact and fast, optimized for lightweight applications in a wider variety of environments
- Our primary target CPUs are 8-bit
- RAM is important requirement
- For short message hashing, good performance tradeoffs
- 2^{120} security level achieved with a high security margin:
- Provide proofs reducing the security of Lesamnta-LW to that of the underlying block cipher performance

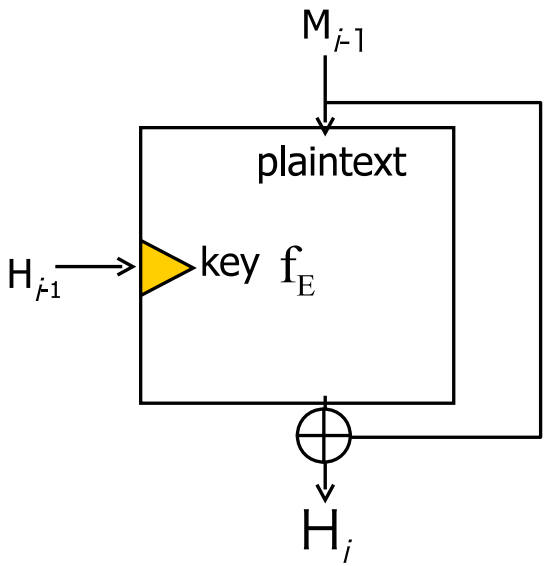
The motitivation for Lesmanta-LW

- Low-cost 8-bit CPUs are popular
  - Over 4 billion 8-bit controllers were sold in 2006
  - RAM is critical for crypto primitives

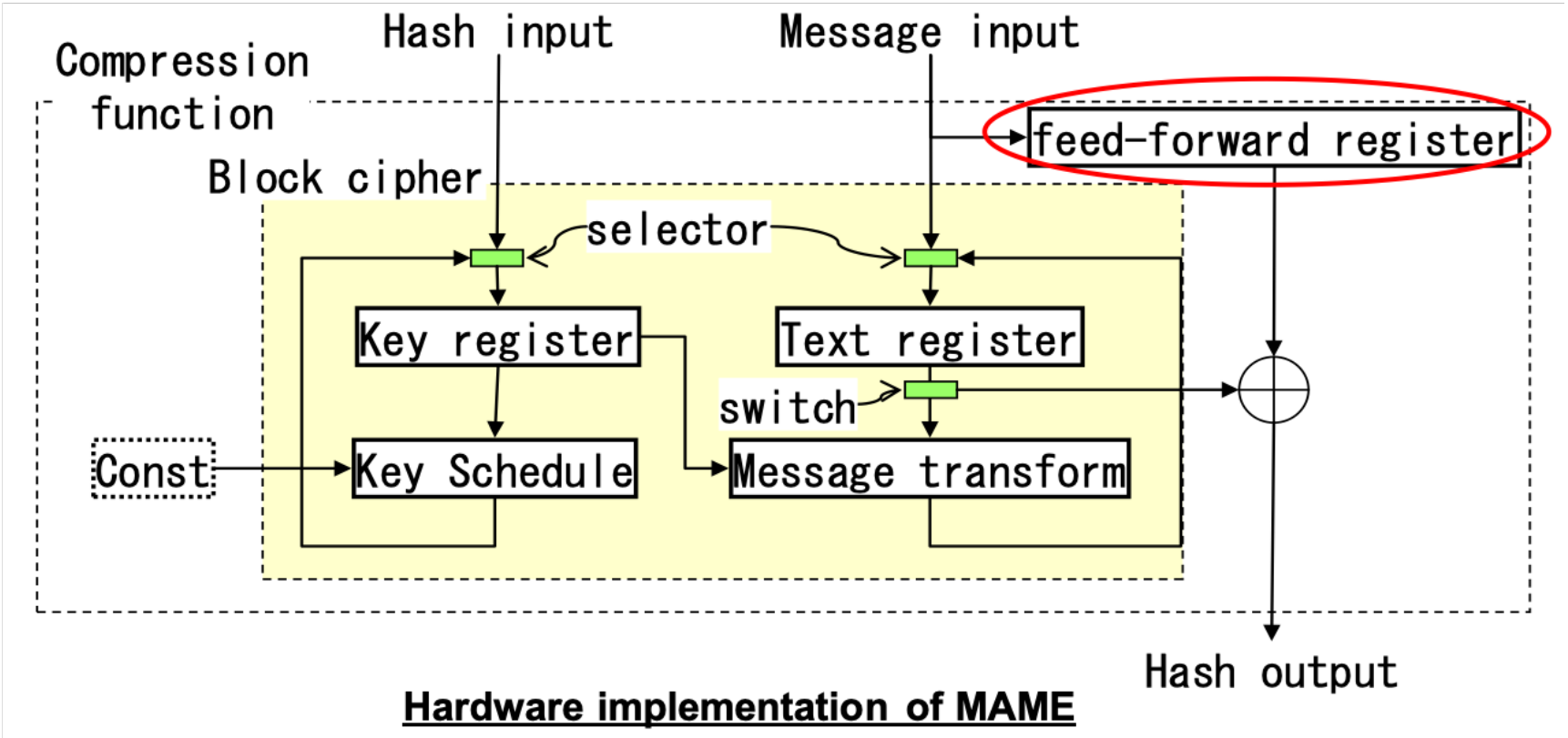
RAM (byte)	Microchip Technology	Freescale	National Semiconductors	NXP	Atmel	Renesas
- 255	PIC10 PIC12	RS08 HC08	COP8	80C51		
256 - 511	PIC16 PIC18	HC08 HCS08	COP8	80C51		
512 -	PIC16 PIC18	HC08 HCS08	COP8	80C51	megaAVR	H8

MMO mode used in MAME Compression function

- MAME uses Matyas-Meyer-Oseas mode with 256-bit block cipher
- Good: block cipher analysis is relevant to hash function analysis



The Problem with MMO



Hardware implementation of MAME

The cost

Circuit		GE
transform	Non-linear Layer	640
	Linear Layer	576
Total		**8200**

The structure of Lesamnta-LW

- LW1 mode can be proved to be collision resistant if the underlying block cipher behaves as a pseudo-random function
- LW1 mode does not have the feedforward of inputs, which contributes to a small memory

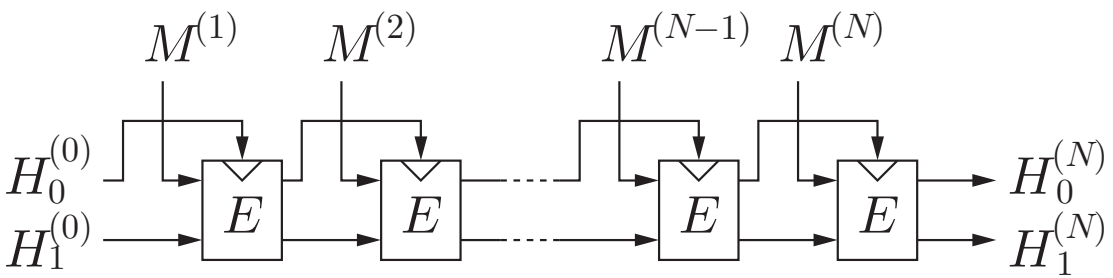
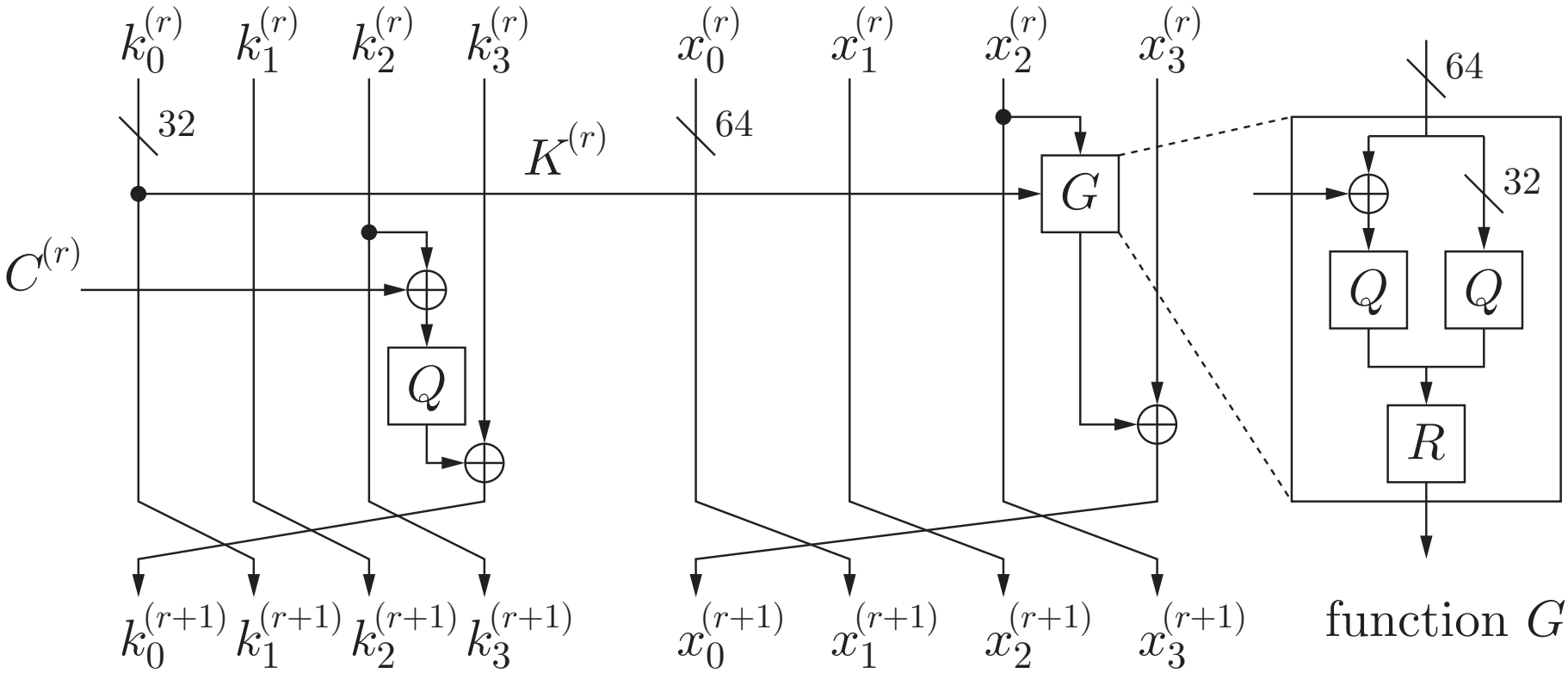


Figure: The structure of Lesamnta-LW

The underlying block cipher for Lesmanta-LW

- Designed to be compact in software/hardware, and to offer a reasonable speed on high-end/low-end CPUs



Security analysis

- The best way to verify this pseudo-randomness, is to apply block cipher analysis techniques to the block cipher E, and to check whether this reveals any weakness or non-random behavior
- We evaluate the security of Lesamnta-LW and the underlying block cipher against all relevant attacks
  - Differential Attacks
  - Linear Attacks
  - Higher Order Differential
  - Interpolation Attack
  - Impossible Differential Attack
  - Related-key Attacks
  - Collision Attacks Using Message Modification
  - Attacks on the Lesamnta Compression Function Using Self-Duality

Our software implementation estimates on an 8-bit CPU Renesas H8

- We have estimated speed and ROM/RAM size of Lesamnta-LW and SHA-256 on an 8-bit CPU Renesas H8

Algorithm	Bulk Speed (cycles/ byte)	Short Message (cycles/ message)	ROM (CONST. +CODE) (byte)	RAM (byte)
SHA-256	1033.3	66434	32 + 37034	330
	1046.9	67308	288 + 5046	330
	1281.1	82296	288 + 948	330
Lesamnta-LW	1650.9	52828	512 + 20006	50
	1736.5	55568	768 + 1346	50
	2055.0	65760	768 + 370	54

- Requires only 50 byte of RAM while achieving 3478 cycles/byte for short (128-bit) messages on an 8-bit CPU:
  - 84% smaller than SHA-256 while running 21% faster

Tpms Hacking

- Rouf, I., Miller, R. D., Mustafa, H. A., Taylor, T., Oh, S.,Xu, W., Gruteser, M., Trappe, W. and Seskar, I.: Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study,19th USENIX Security Symposium, 2010, Proceedings, USENIX Association, pp. 323–338 (2010).

Figure: A Car and TPMS sensor

ISO/IEC 29192-5 Lightweight Hash (2012-2016)

- Lesamnta-LW
  - 256-bit hash function using a block cipher employing AES components
  - low RAM-used (50 Byte) implementation on 8-bit microcontrollers is possible
  - presented in ICISC 2010 conference and IEICE journal
- Spongent
  - Sponge function-based hash function
  - hash length supports 80, 128, 160, 224, 256
  - Presented in CHES 2011 conference
  - Low-gate count (738GE) hardware implementation is possible
- Photon
  - Sponge function-based hash function
  - hash length supports 80, 128, 160, 224, 256
  - Presented in conference(CRYPTO 2011)
  - Low-gate count (865GE) hardware implementation is possible

Apply Lesamnta-LW to TPMS (2016-2018)

Lesamnta-LW can produce multiple independent PRFs (NIST LWC WS).

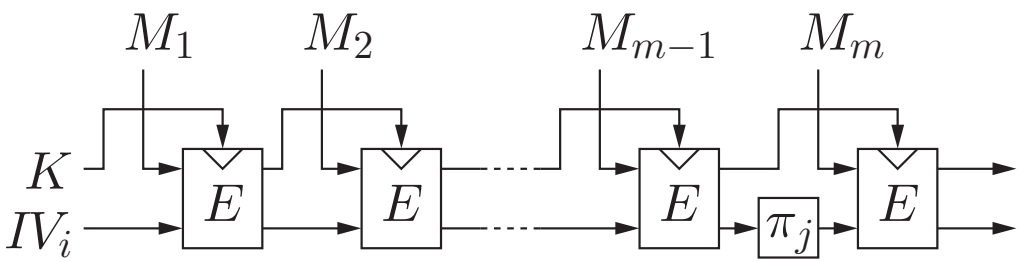
Theorem

Lesamnta-LW with MDP produces multiple independent PRFs with

- *a single key* K
- *multiple IVs* $\mathcal{V} = \{IV_1, IV_2, \ldots, IV_a\}$
- *multiple permutations* $\Pi = \{\pi_1, \pi_2, \ldots, \pi_d\}$

$\Longleftarrow$ E *is PRP and*

1. $\pi(x) \neq \pi'(x)$ *for every* $\pi, \pi' \in \Pi \cup \{id\}$ *and every* $x \in \Sigma^{n-w}$
2. $\pi(IV) \neq \pi'(IV')$ *for every* $(\pi, IV), (\pi', IV') \in (\Pi \cup \{id\}) \times \mathcal{V}$



These PRFs can be applied to TPMS (escar 2018).

Cyber physical systems

- Real time requirement

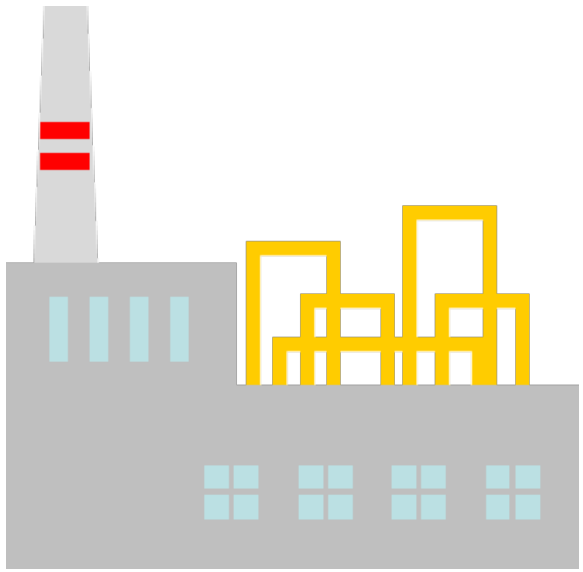


Figure: A Factory

The problem for PLC

- AES-CTR can be problematic for programmable logic controller (PLC)

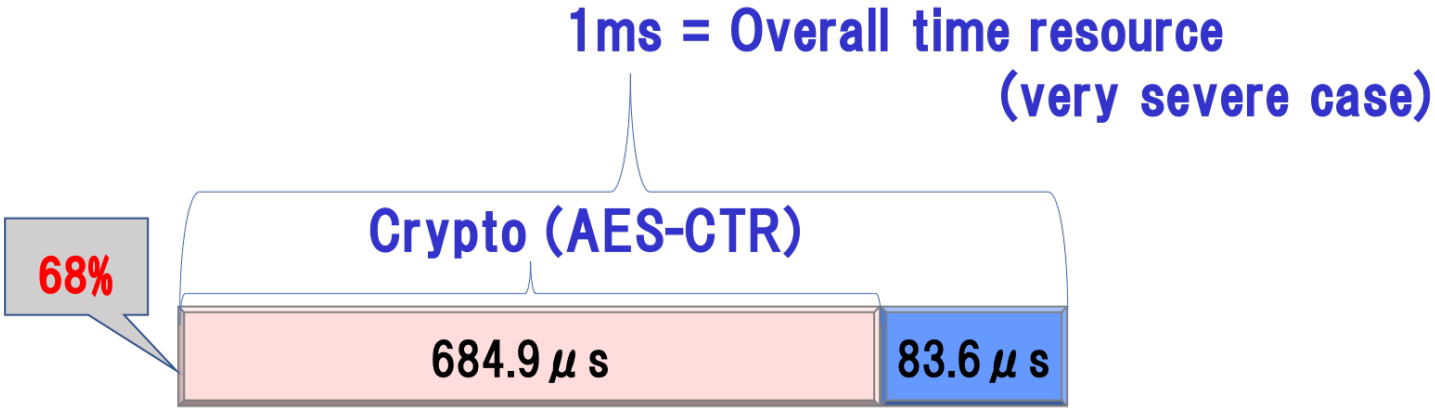
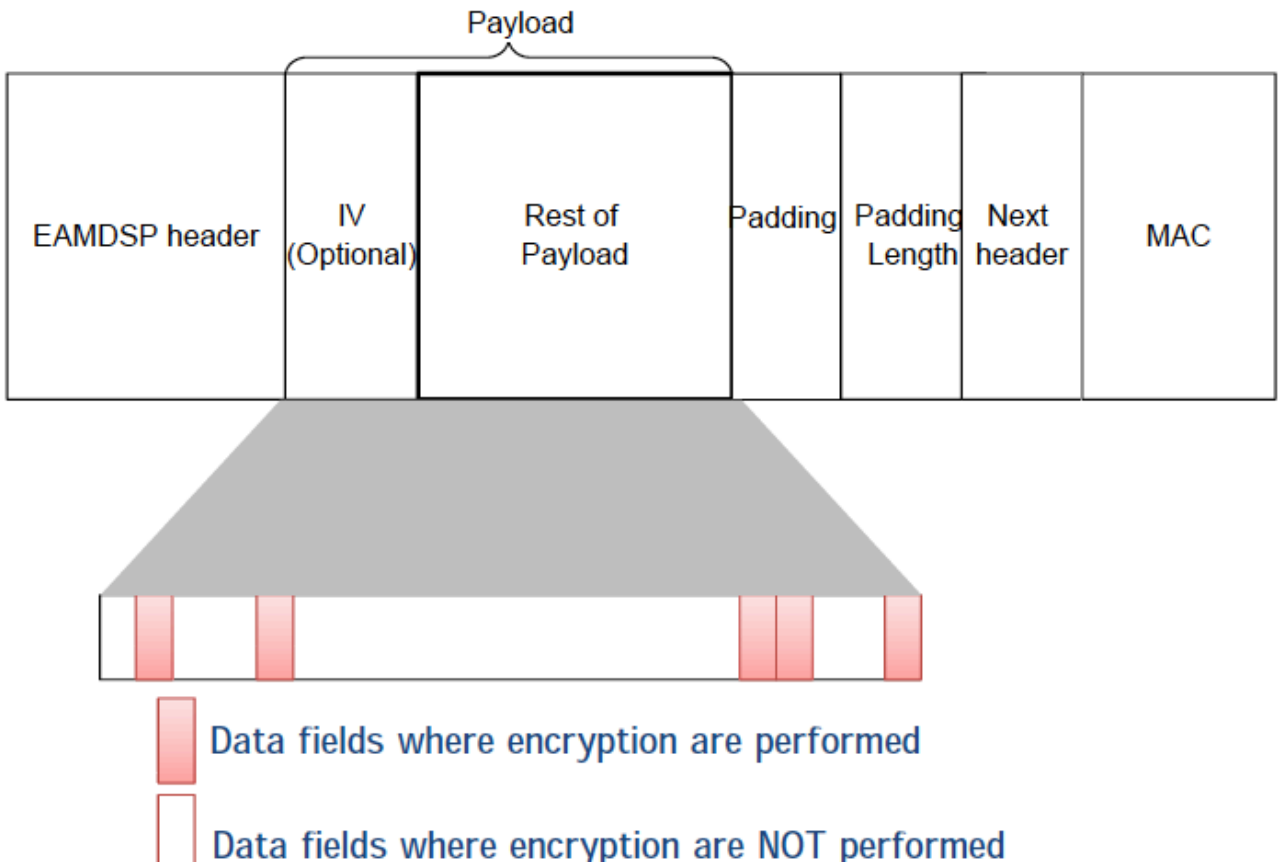


Figure: Crypto eats too much resourcses on PLC

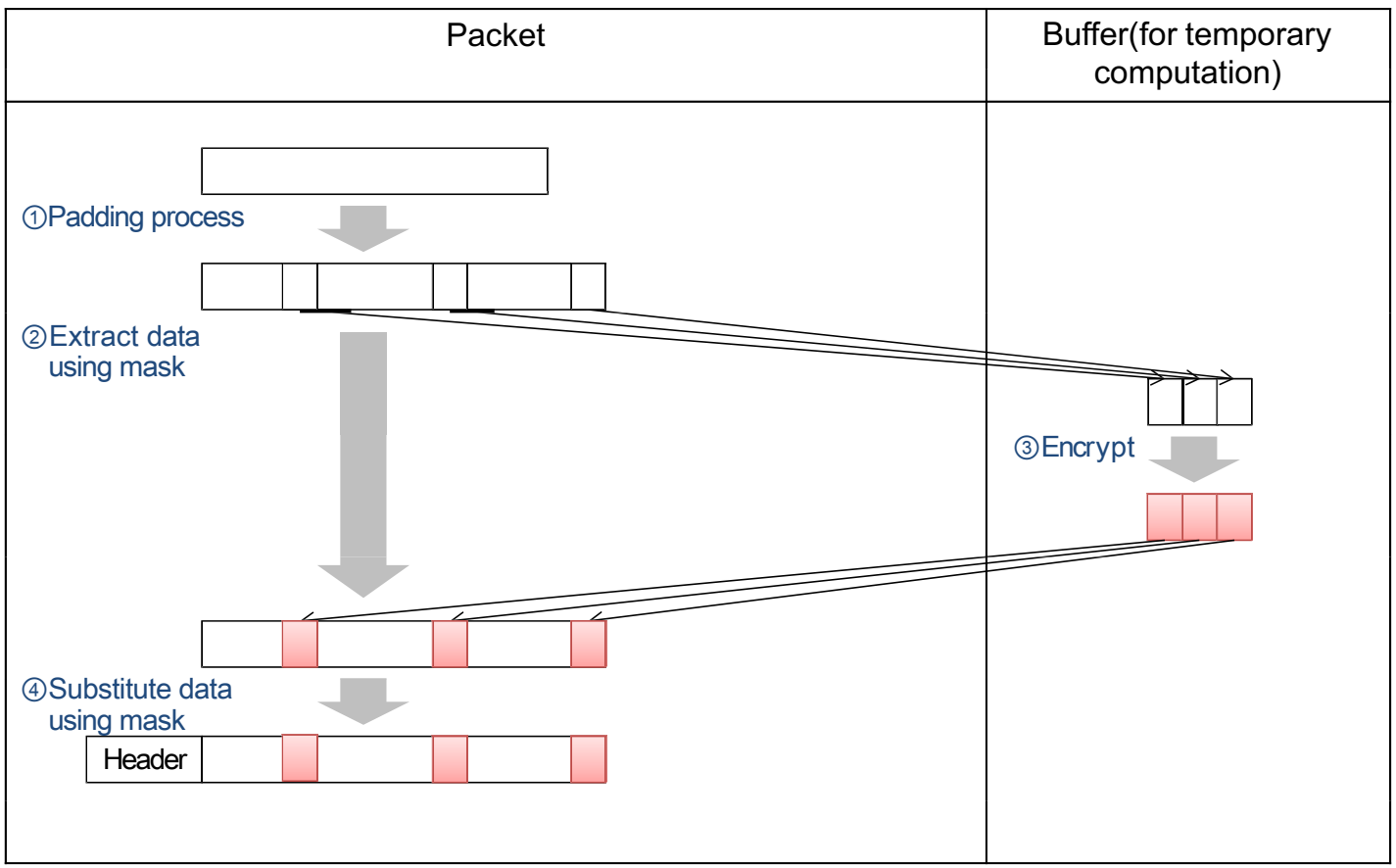
ITU-T X.1362 (2017): EAMD

- EAMD (encryption with associated mask data)
- Reduce the overhead by encrypting the only data that are sensitive



Packet sending flow in EAMD-used communication

- Generate packet using the mask indicating sensitive data location



ISO/IEC 29192-6 lightweight MAC project (2014-)

- Good progress: DIS (Draft of International Standard)
- 3 mechanisms: Chaskey-12, LightMAC mode, Tsudik mode
- Implementation results
  Achieve the speed of 7.0 cycles/byte on ARM Cortex-M4
  - Comparing to AES-128-CMAC, Chaskey achieves 12 time higher speed, program size is 1/20

CPU	Algorithm	Data size (Byte)	Program size (Byte)	Speed (cycles/byte)
Cortex-M4	AES-128-CMAC	128	8,740	89.4
Cortex-M4	Chaskey-8	128	402	7.0

Conclusion

- The requirements-oriented view and the use of the lightweight cryptographic stack (LWCS) could be important for deployment.
- LWCS for size requirement
  - MAME requiring small circuit size
  - Lesamnta-LW requiring small RAM ISO/IEC 29192-5: 2016
  - Its crypto stack presented in NIST LWC 2016 and Escar Asia 2018
- LWCS real-time requirement
  - ITU-T X.1362 (2017) EAMD protocol
  - On-going lightweight MAC project: ISO/IEC 29192-6
  - Chaskey-12, software oriented, fast on ARM
- The *future challenge* for CPS security
  - Resistance against fault analysis could be important from common criteria (ISO/IEC 15408) perspective
  - Design a stream cipher meeting the following requirement: small circuit size of countermeasure against fault analysis

Thank you very much for your attentions.