TOSHIBA

Leading Innovation

Session 2. Security-Aware System Design

Design for Endpoint Devices Considering Security

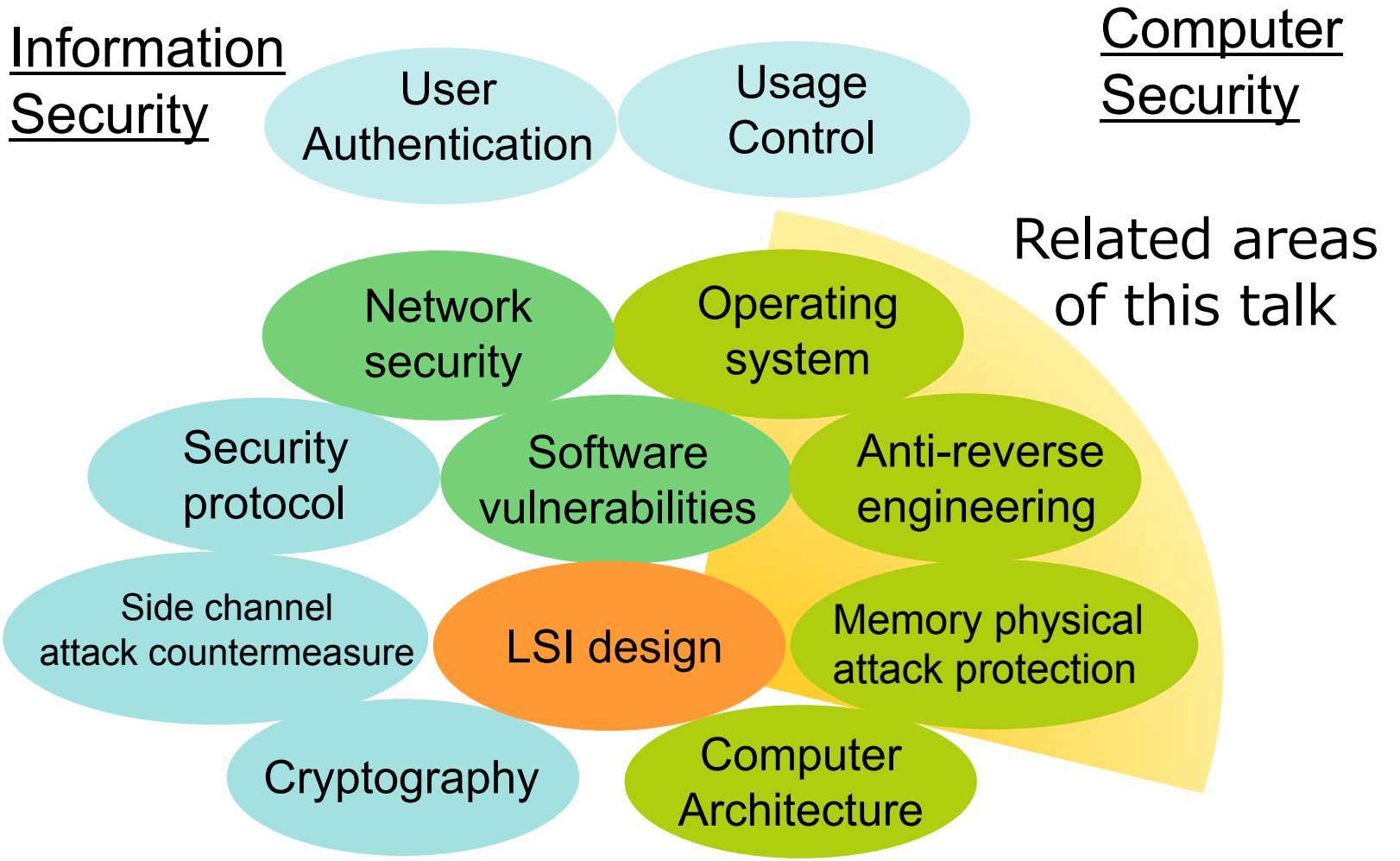
Mikio Hashimoto
(Computer Architecture and Security Systems Laboratory Corporate R & D Center, Toshiba Corp.)

2015-07-08　International Workshop on Cybersecurity

Outline

- **Technical area and my background**
- **Endpoint “specific” device security**
  - Security processor for endpoint devices
- **Security processor design**
- **Conclusion**

Technical areas



My background

- 1991～1995 OS for packet exchange network
- 1995～1998 Home networking
- 1999～ Digital copyright protection and Security Processor
- 2010～ +Social infrastructure protection

Attack example of endpoints

- **Control systems**
  - Oil pipeline explosion at Turkey (2009)
    - Sophisticated attack with several steps of combination of **endpoint sensors** and central control malfunction

(*) https://ics.sans.org/media/Media-report-of-the-BTC-pipeline-Cyber-Attack.pdf

- **Digital content protection**
  - Piracy by software cracking
    - Illegal disk duplicate software is sold
  - The attacker used open source tools


BloombergBusiness News Markets Insights Video

Mysterious '08 Turkey Pipeline Blast Opened New Cyberwar

by Jordan Robertson and Michael Riley

December 10, 2014 – 7:00 PM JST

Firemen struggle to extinguish the blaze at the Baku-Tbilisi-Ceyhan (BTC) pipeline near the eastern Turkish city of Erzincan on Aug. 7, 2008. The BTC pipeline was burning from an explosion on Aug. 6,

Bloomberg.com 12/10/2014

Endpoint “specific” device security

- **Endpoint characteristics**
  - Field devices
    - Placed on less controllable environment
      - Physical attack
  - Consumer devices
    - Manipulatable using development tool

- **Attack surfaces to software**
  - ①Network attack
  - ②Evil programs

  } Common with servers

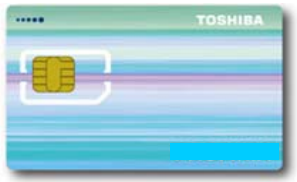
  - ③Physical attack
  - ④Development tools

  } Specific to endpoints

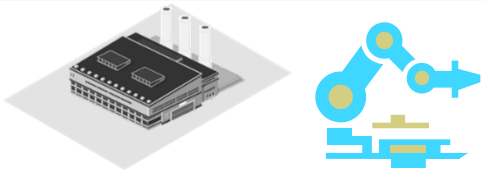
- **Focus on**
  - software reverse engineering countermeasure

Brief history of endpoint security

	Application	Objective
1980 ～	IC cards	Prevent forgery of the **ownership**
1990 ～	Digital Content Protection	**Rules for information handling** at the endpoint
2010 ～	Control Systems	Prevent manipulation on **system behavior**



Preventing content piracy



Target of our security processor at 1999

- Make the most of untrusted open source OS

Design for anti-reverse engineering

- **DVD protection system was cracked (1999)**
  - by software reverse engineering
- **We started LMSP™ research at 1999**
  - LMSP: License-controlling Multiparty Secure Processor
- **Design for value**
  - Value of SoC is supported by software
  - SoC could offer value of software protection
  - Software protection feature will be strong value of SoCs
  - Possible extension to intellectual property protection
- **Security processor works are done parallel**
  - XOM(2000), Aesis (2003), LMSP (2004),…
  - Considering an untrusted OS.

Assets and attack vector

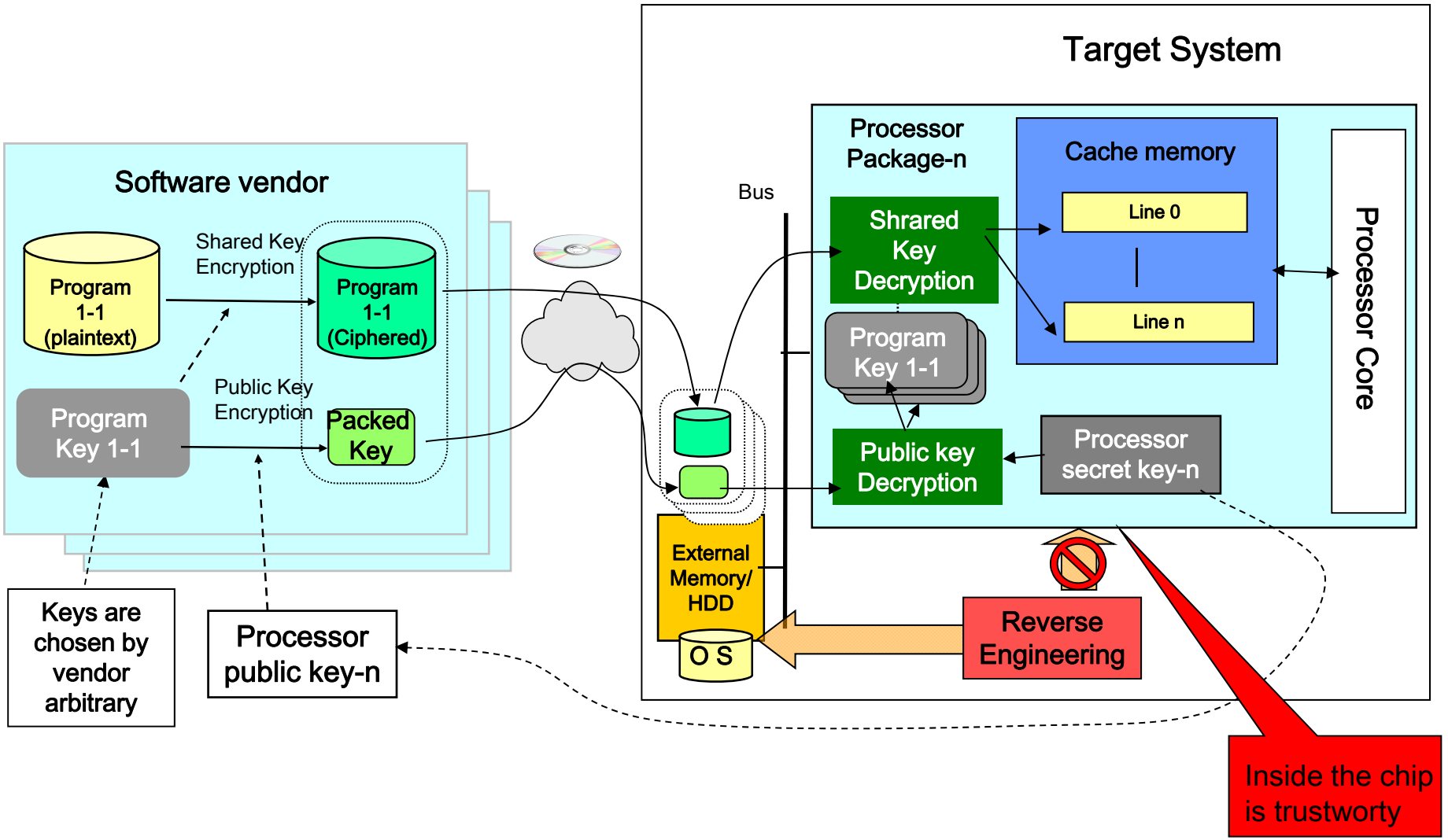
- **Assets**
  - Confidentiality & integrity of target applications
    - Content protection rules and secrets for cipher processing
- **Assumption**
  - Inside chip boundary is secure
- **Attacker capabilities**
  - Physical access and use of development tools
- **Attack vector**
  - Static analysis
  - Dynamic analysis using development tools
    - SW
      - Debuggers and OS modification
    - HW Debugging tools
      - Memory bus analyzer

Design for mechanism

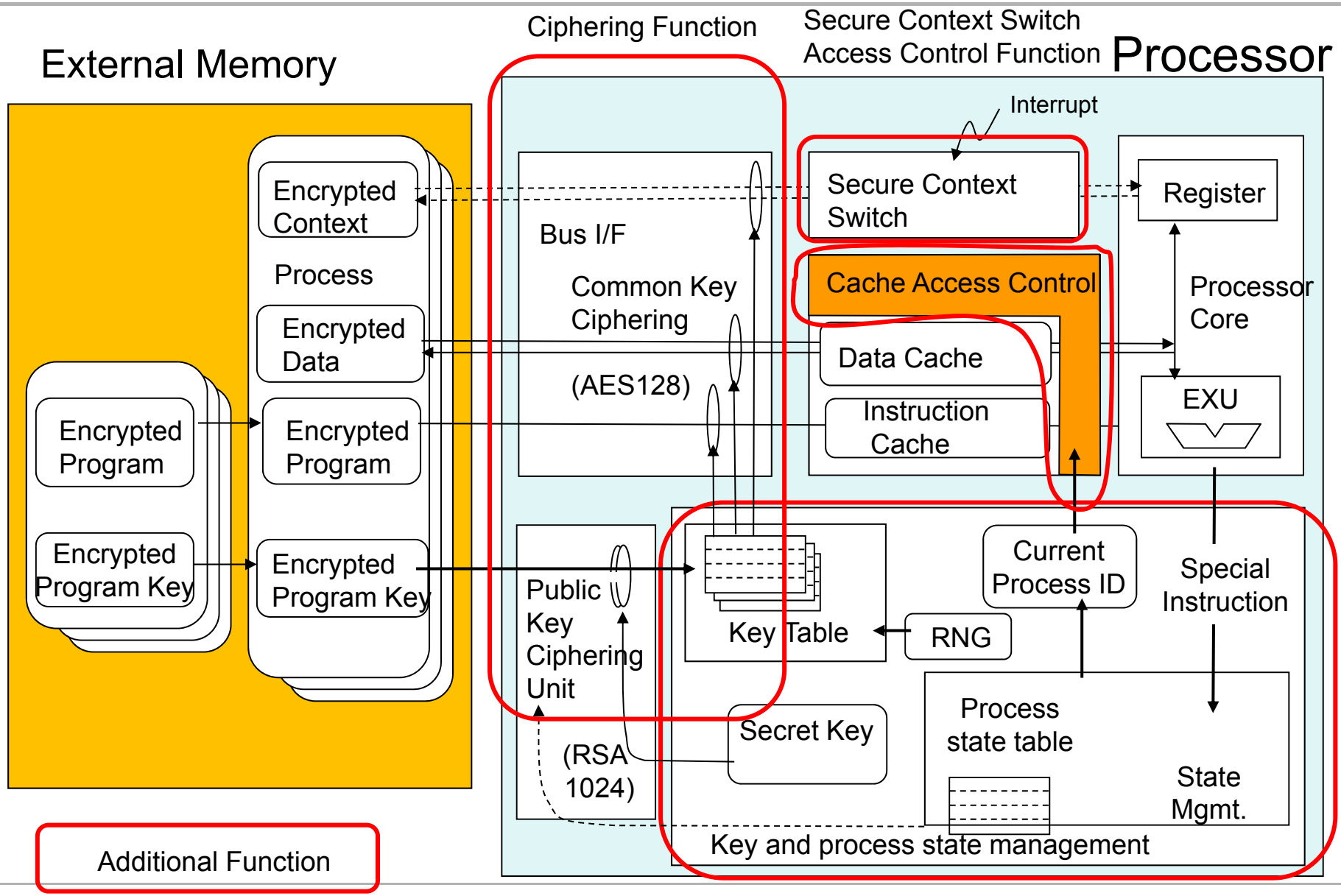
- **Problem**
  - Application protection against a hostile OS
    - Developers and users are potential attackers
  - Memory encryption
    - for physical attack

- **Approach**
  - Application process management and cipher by hardware
    - Protect applications, not OS
  - Isolation of security attributes
    - Processor HW management
      - Confidentiality and integrity protection of processes
    - OS SW management
      - Availability of processes (CPU time allocation)

Software distribution on the LMSP

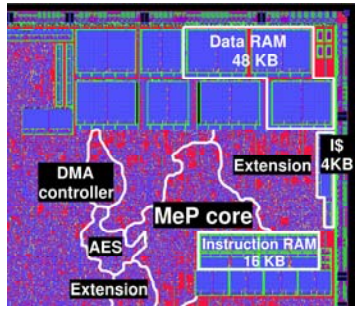


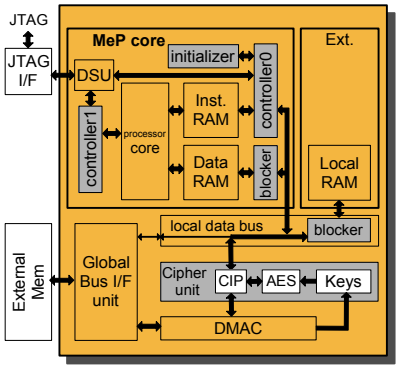
Hardware design for process protection



Functional Testbed and after

- **LMSP functional testbed is completed successfully (2003)**
- **Simple secure processor MeP-c4A was developed**
  - Applied to few products
  - No multivendor feature
    - Limited application only
- **LMSP was not deployed, why?**
  - Overheads (chip area, speed)
  - Incompatible with existing OS
    - Shared libraries, …
  - For content protection
    - “Software obfuscation” is sufficient
    - Secure boot has good compatibility
- **Design for deployment was insufficient**





Leaning from the lesson

- **New security value (software protection) was exploited**
- **The core technology development was successful**
- **Obstacles**
  - Threats were not so serious at those days
  - Insufficient design for deployment
  - Embedded processor architecture was converged to ARM
    - Toshiba original RISC core development was stopped
- **Time and place were not adequate**

Reprise of secure processor

- **2013: Intel® SGX (Software Guard Extension)**
  - Security processor with strong integrity protection
  - This is still a Intel vision proposal, no official product announcement
- **Supposed background**
  - Untrusted cloud provider
  - Vulnerabilities of OS and critical middleware
    - Heartbleed (SSL), Shellshock (Shell), POODLE (SSL), …
  - Without strong module isolation, broad range of system cracking might be raised
  - Social infrastructure requires higher level of protection
  - Integrity verification is strengthen in SGX
- **Social infrastructure is candidate for security processor application**

Conclusion

- **Endpoint specific device security**
- **Looking back an early security processor design**
  - New security value was exploited
  - Hardware design
  - Design for deployment
- **Reprise of security processor?**
  - To avoid cyber attack on social infrastructure

References

- **LMSP**

  Mikio Hashimoto, Hiroyoshi Haruki, and Takeshi Kawabata, "Secure Processor Consistent with Both Foreign Software Protection and User Privacy Protection," in Proceedings of Security Protocols 12th International Workshop,: Springer, 2006, LNCS vol. 3957, pp. 276-286.

- **MeP-c4A**

  Takeshi Kawabata, Takanori Tamai, Mikio Hashimoto, Takashi Miyamori: Security Enhanced Embedded Processor using Local Memory Protection Mechanism, in Proceedings of Cool Chips IX, 2006, pp. 143-157.

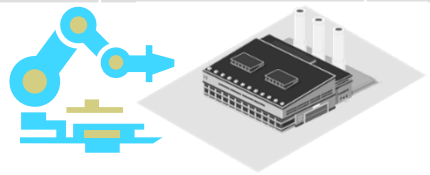
  – https://www.toshiba.co.jp/tech/review/2007/high2007/high2007pdf/0703.pdf

TOSHIBA

Backup slides

Security targets

	Application	Objective	technology	Attack Incentive	Damage	Complexity
1980 ~	IC cards	Prevent forgery of the **ownership**	Dedicated chip	+++	++	+
1990 ~	Digital Content Protection	**Rules for information handling** at the endpoint	ASIC & Software technique (on PC)	++	+	++
2010 ~	Control Systems	Prevent manipulation on **system behavior**	COTS CPU with security?	+	++++	+++



Required Protection Level

Protection Cost

Software protection and user privacy