Embedded Security for Internet of Things Forum

Research Activities for Embedded System Security

Yutaka MATSUBARA

Assistant Professor, Nagoya University, Japan

Web: http://www.ertl.jp/~yutaka/profile-e.html

E-mail: yutaka@ertl.jp

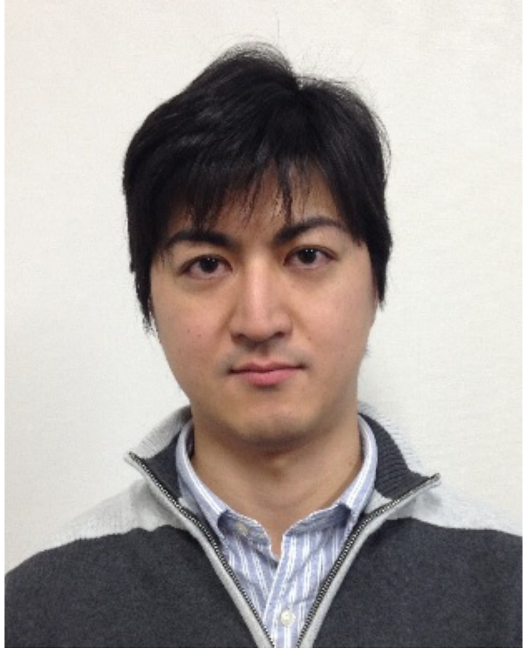
Agenda

- Introduction
- Trends regarding Embedded System Security
- Security by Design in Embedded System Development
- Software Platform for Safety and Security Critical Embedded System

Self Introduction – Yutaka MATSUBARA

Current Positions

- Assistant Professor, Graduate School of Information Science, Nagoya University
- Committee member, TOPPERS Project
- Technical adviser for several companies

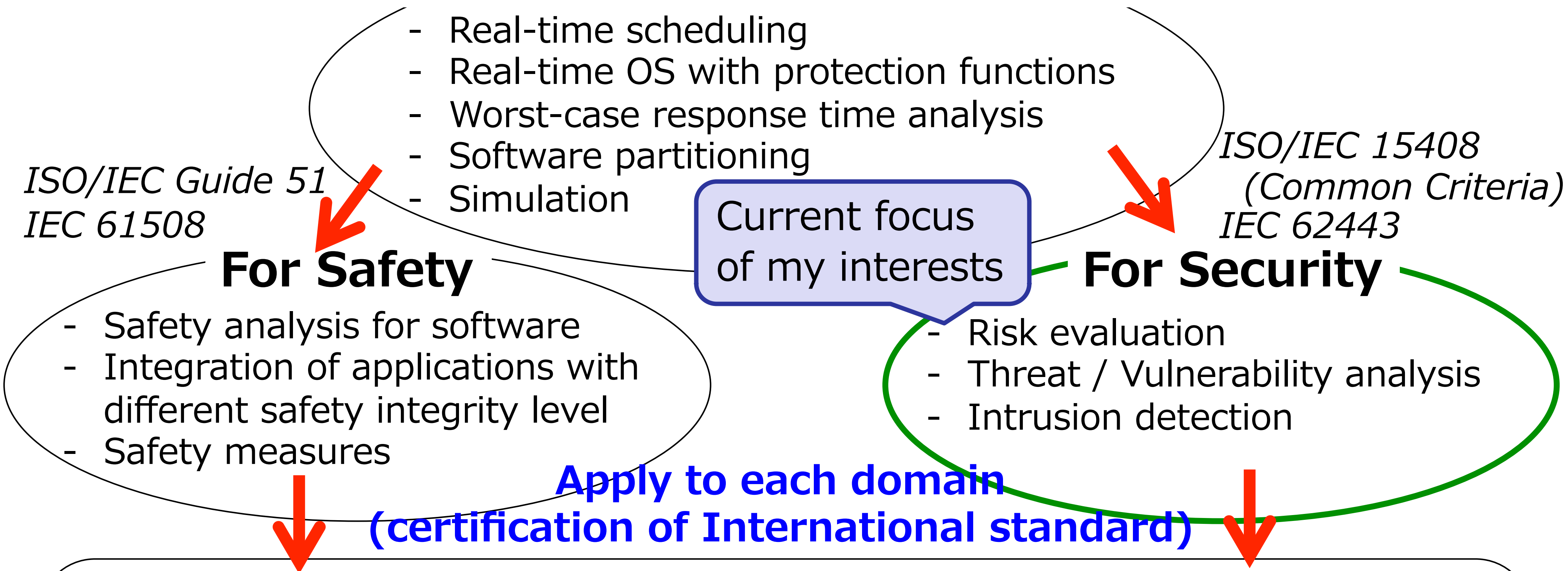


Major Research Topics

- Real-time operating systems and networks for embedded systems
  - e.g. AUTOSAR compatible RTOS, Ethernet AVB
- Real-time scheduling and analysis
- Functional safety and security for embedded systems including IoT devices

My Research Fields

Design, Implementation, Validation of Embedded Systems



- Real-time scheduling
- Real-time OS with protection functions
- Worst-case response time analysis
- Software partitioning
- Simulation

ISO/IEC Guide 51
IEC 61508

For Safety

- Safety analysis for software
- Integration of applications with different safety integrity level
- Safety measures

Current focus of my interests

ISO/IEC 15408 (Common Criteria)
IEC 62443

For Security

- Risk evaluation
- Threat / Vulnerability analysis
- Intrusion detection

Apply to each domain (certification of International standard)

Automotive

ISO 26262

Railway

IEC 62278 (RAMS)

Aircraft

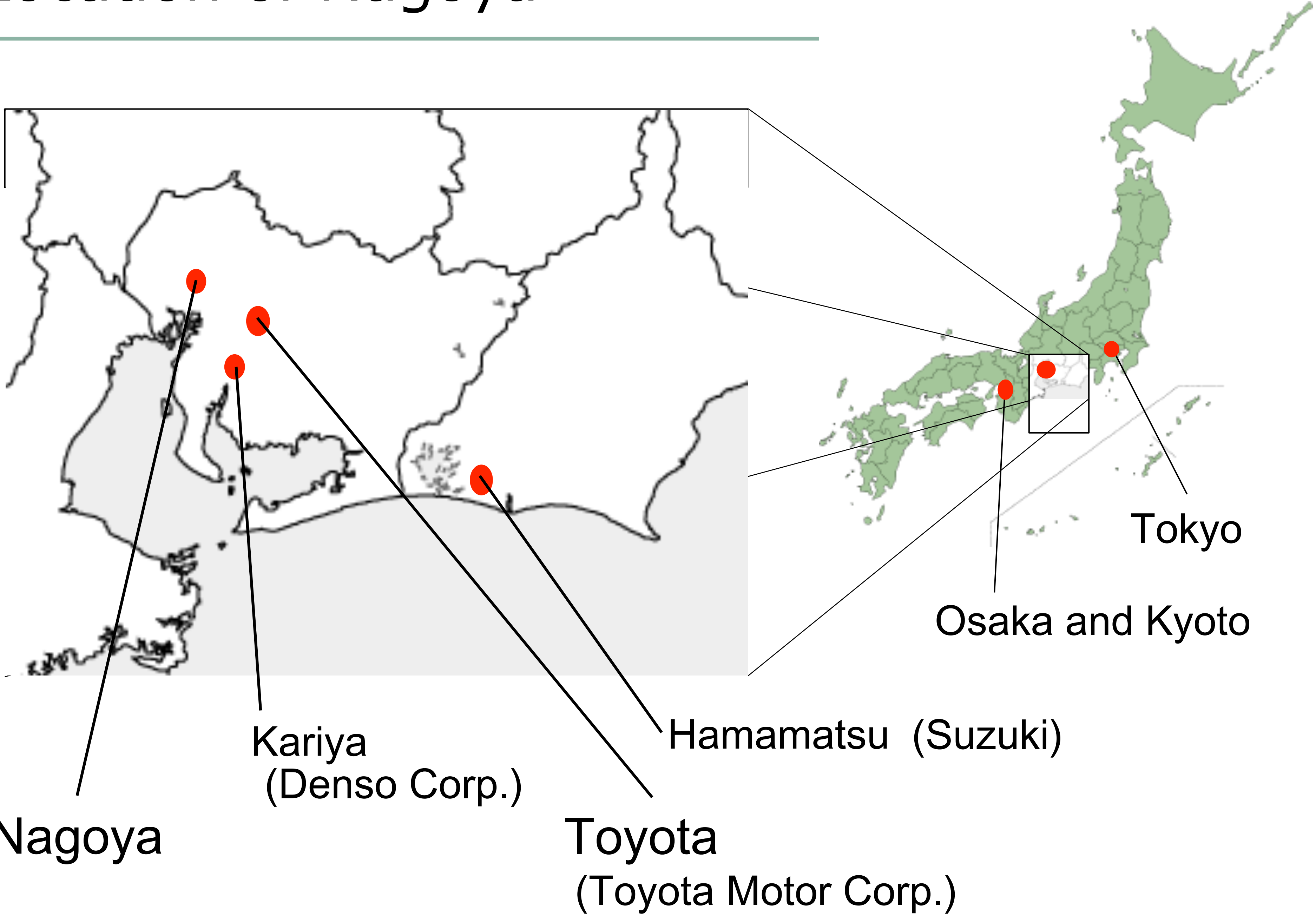
DO-178C
DO-297

Spacecraft

Robot

IEC 61508

Location of Nagoya



Organizational Overview

Embedded and Real-Time Systems Laboratory

- Prof. Takada’s and Prof. Edahiro’s Laboratories
- Many joint projects with industries

→ ***http://www.ertl.jp***

NCES (Center for Embedded Computing Systems)

- Several (relatively) large-scale joint projects with companies including car makers, car component suppliers, and semiconductor makers
- Projects for educating engineers

NCES

→ ***http://www.nces.is.nagoya-u.ac.jp***

TOPPERS Project

- Independent non-profit organization
- Distribution of open-source RTOS and middleware
- Cooperation of academia, industry, public research institutes, and individual engineers

→ ***http://www.toppers.jp***

TOPPERS

Trends regarding Embedded System Security

Why is embedded system security remarkable?

Highly-functional and networked embedded systems

- Increasing embedded products connected to Internet or each others
- Employed existing technologies instead of originally developed software
  - e.g. OS, TCP/IP stack, USB stack, etc.

  →**embedded systems can be attacked by security threats**

Security problems can impact to safety

- Functional safety has been spread in industry.
- But, violation of security policy can lead to violation of safety requirements.

  →**Not only safety measures but also security measures are important for embedded system safety**

Differences between Safety and Security

	Safety	**Security**
Target Area	• Only safety related parts in target product • Only for failures in target product • Developers and users are reliable as assumption.	• Target product **and connected products** • Developers and users may be unreliable. • Third party's (attacker's) intentionality
State to guarantee property	• Safe state can be defined in almost systems.	• Secure state cannot be defined. • Security Threat will increas in future.
Definition of Level for measures	• SIL(Safety Integrity Level)	• SAL(Security Assurance Level) • TAL(Trust Assurance Level)
International Standard	• Many standards for functional safety have been already published.	• ISO 15408 is used well for information security. • But, standards for embedded system security are under discussion.

Security threats for embedded systems

Home appliances

http://www.insurancejournal.com/news/international/2014/07/18/335214.htm

Medical devices

http://www.theregister.co.uk/2011/10/27/fatal_insulin_pump_attack

Automotives

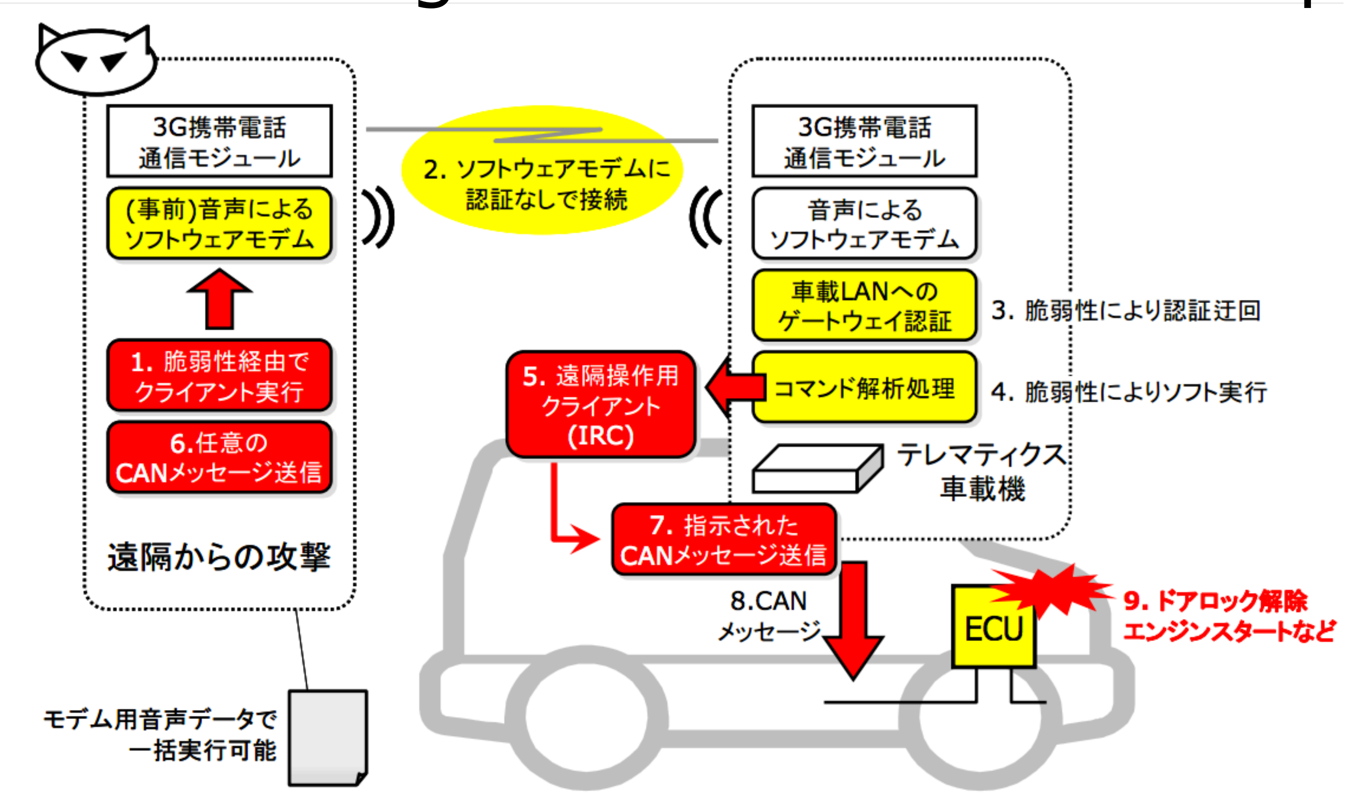
http://www.autoblog.com/2014/07/18/auto-industry-deals-with-hacking-cyber-threats/

Industrial Robots

http://www.iec.ch/etech/2014/etech_0614/ca-1.htm

Threat for Automotive control system

- Prof. Kohno in Univ. of Washington reported that they could attack to automotive control systems in 2010
- In 2011, remote attacks for cars were also succeed through 3G network and CD player



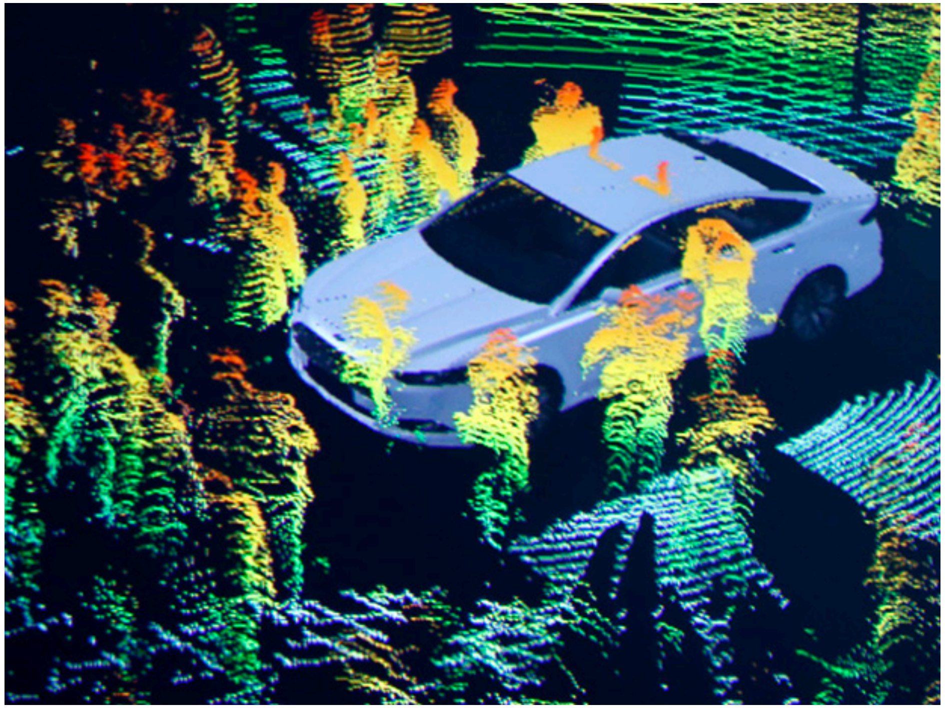
引用：2011 年度自動車の情報セキュリティ動向に関する調査http://www.ipa.go.jp/files/000024413.pdf

Remote Attacks for Automotive

Remote access via OBD-II

http://www.wired.com/2015/08/hackers-cut-corvettes-brakes-via-common-car-gadget

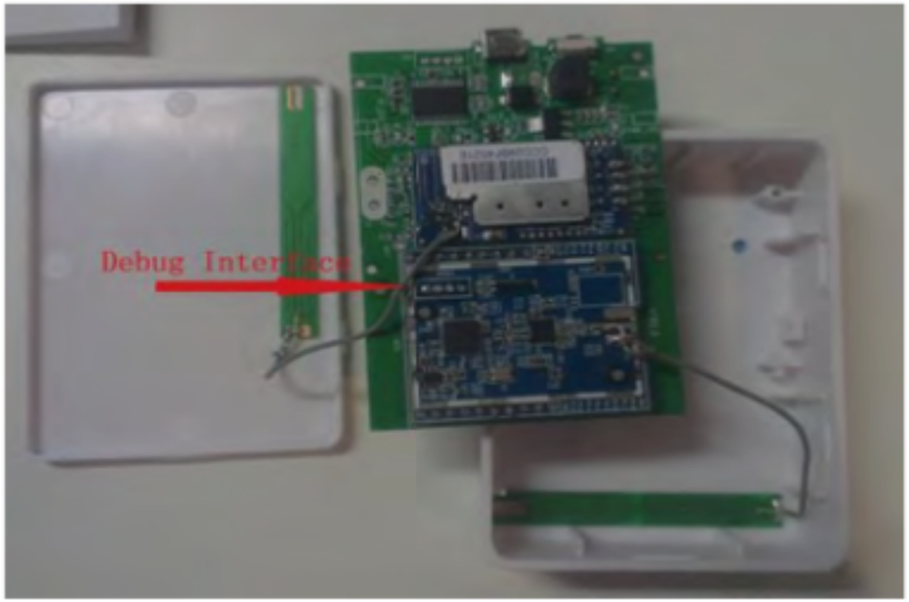
Modification of LIDAR’s signal



http://spectrum.ieee.org/cars-that-think/transportation/self-driving/researcher-hacks-selfdriving-car-sensors?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+IeeeSpectrumCarsThatThink+%28IEEE+Spectrum+Cars+That+Think%29

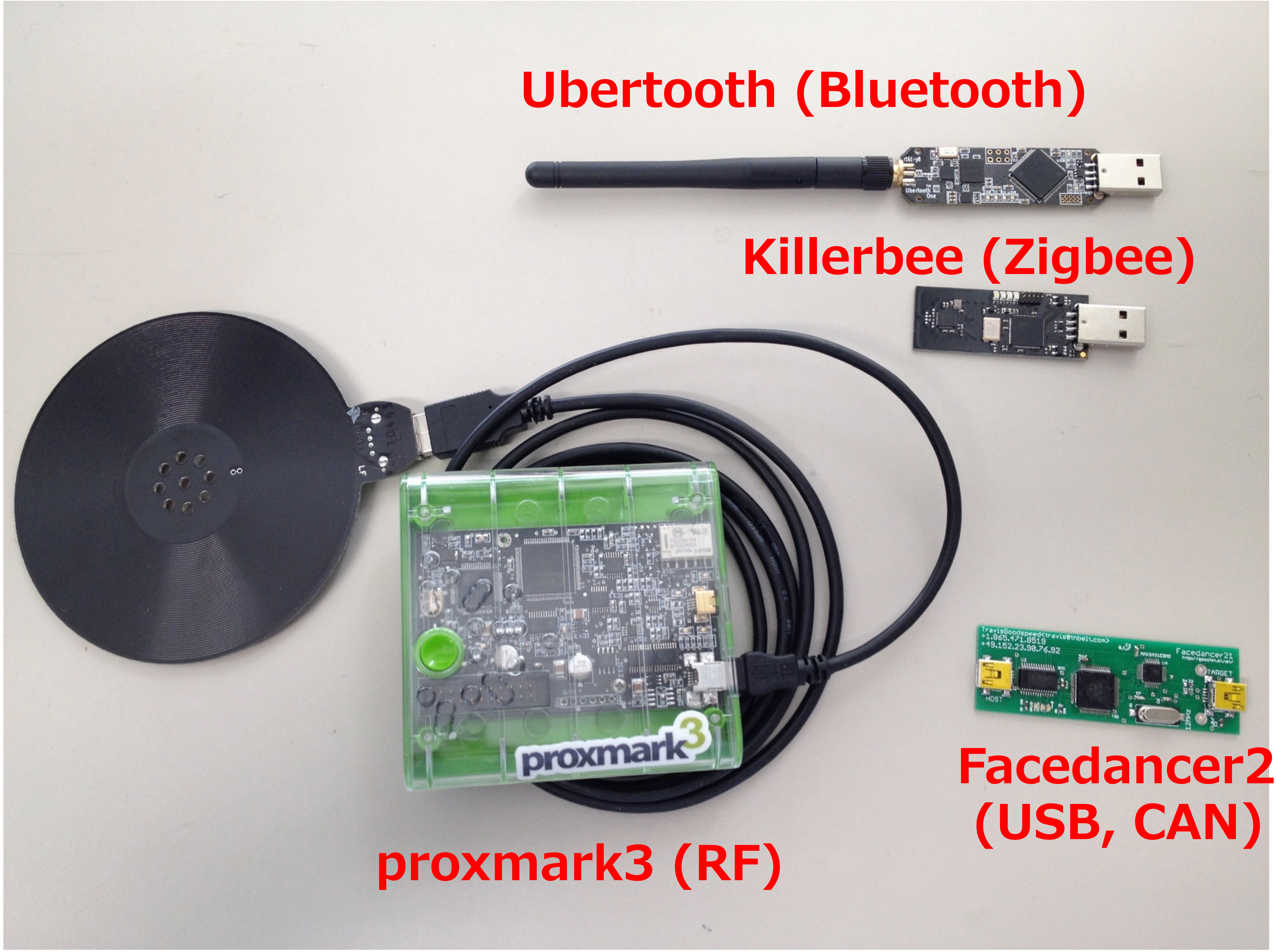
Direct Attack to IoT Gateway

- LI Jun, YANG Qing : I’M A NEWBIE YET I CAN HACK ZIGBEE, DEF CON 23, 2015年
- Injustice operations to home appliances connected to IoT gateway via WIFI and Zigbee
  - Attacked to IoT gateway physically
  - Performed reverse engineering firmware and identified private key for authentication of IoT devices
  - By using the private key, attackers could access to IoT devices



Attacks to gateway will increase in future.

Open source testing tools



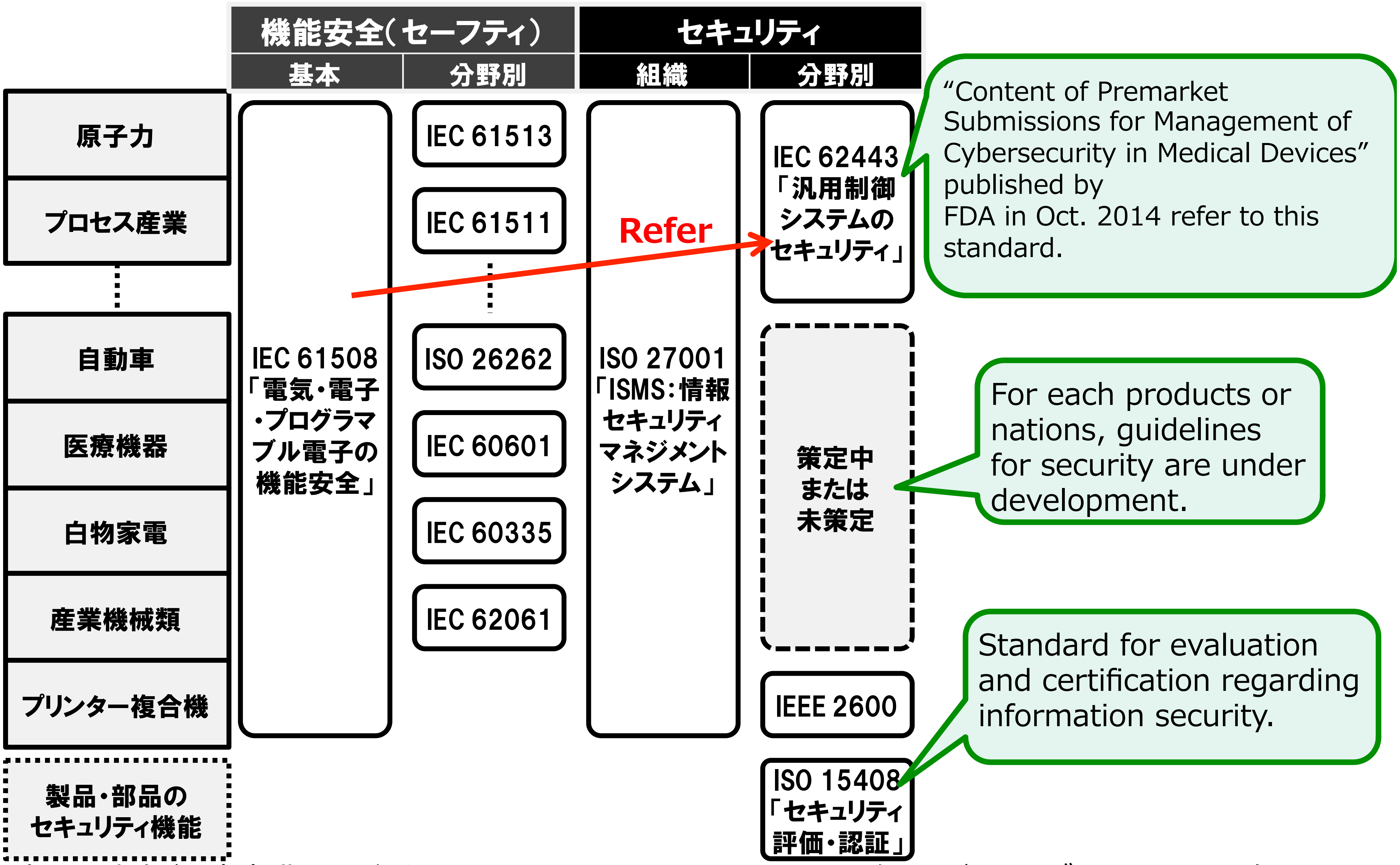
Ubertooth (Bluetooth)

Killerbee (Zigbee)

Facedancer21 (USB, CAN)

proxmark3 (RF)

International Standard for safety and security



機能安全（セーフティ）

基本

分野別

セキュリティ

組織

分野別

原子力

プロセス産業

自動車

医療機器

白物家電

産業機械類

プリンター複合機

製品・部品の
セキュリティ機能

IEC 61508
「電気・電子
・プログラマ
ブル電子の
機能安全」

IEC 61513

IEC 61511

ISO 26262

IEC 60601

IEC 60335

IEC 62061

ISO 27001
「ISMS：情報
セキュリティ
マネジメント
システム」

IEC 62443
「汎用制御
システムの
セキュリティ」

策定中
または
未策定

IEEE 2600

ISO 15408
「セキュリティ
評価・認証」

Refer

"Content of Premarket Submissions for Management of Cybersecurity in Medical Devices" published by FDA in Oct. 2014 refer to this standard.

For each products or nations, guidelines for security are under development.

Standard for evaluation and certification regarding information security.

引用：中部経済産業局：組込みシステムのセキュリティ取組みガイドブック, 2014年

Problems about Embedded System Security

Lack of specialists and engineers

- The number of safety engineers are increasing
- But, the number of security engineers who have enough knowledge about embedded systems are insufficient

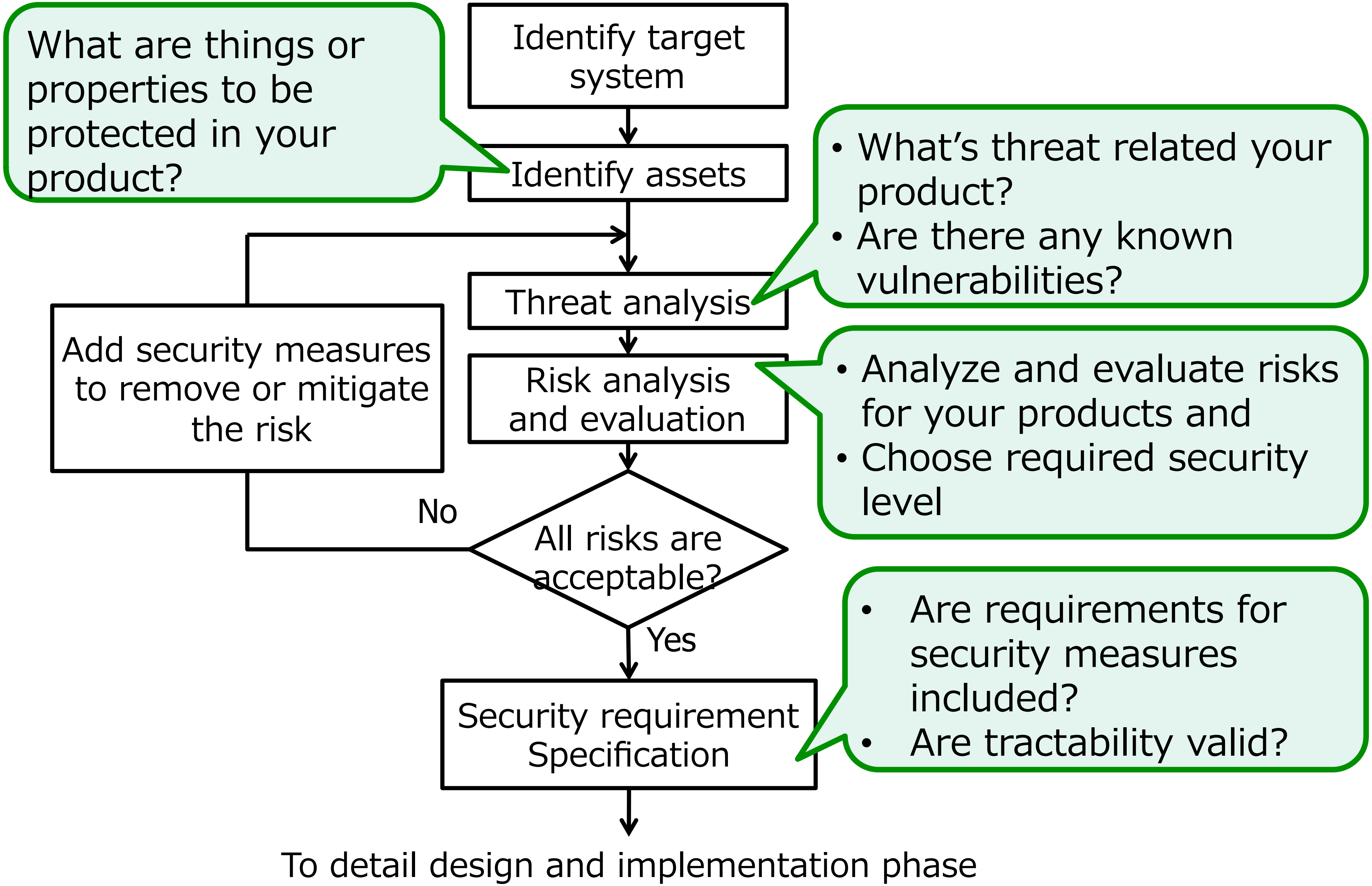
Standards/Guidelines are under development

- International standard for functional safety and Information security have been spread in industry
- But, standard for security of embedded systems, especially for safety-critical systems is not published
- There are several guidelines published by IT companies or governments

Education for security engineers and guidelines are required!

Security by Design

Design Process for Security


Identify target system
Identify assets
Threat analysis
Risk analysis and evaluation
All risks are acceptable?
No
Yes
Add security measures to remove or mitigate the risk
Security requirement Specification
To detail design and implementation phase

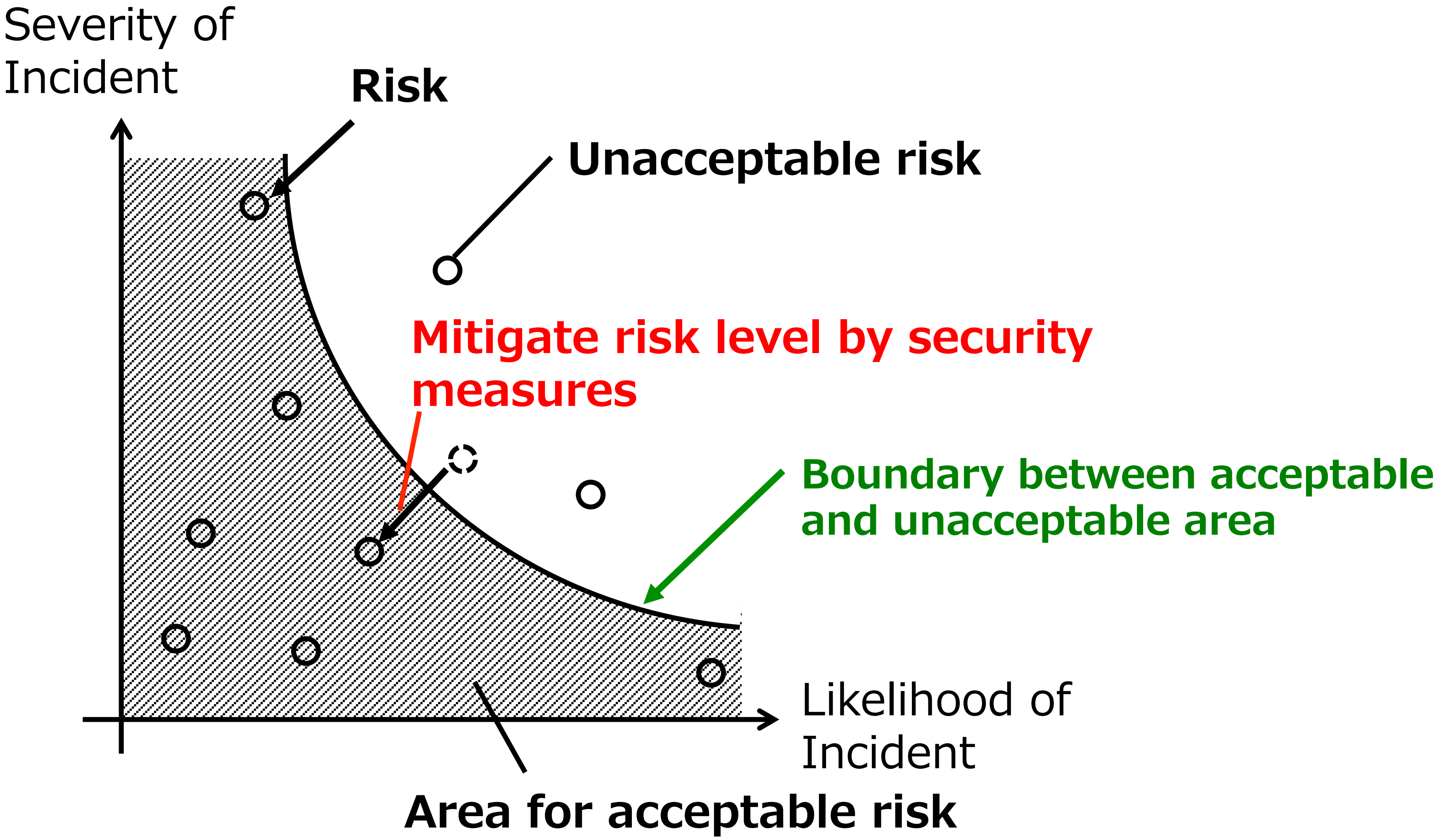
What are things or properties to be protected in your product?

- What’s threat related your product?
- Are there any known vulnerabilities?

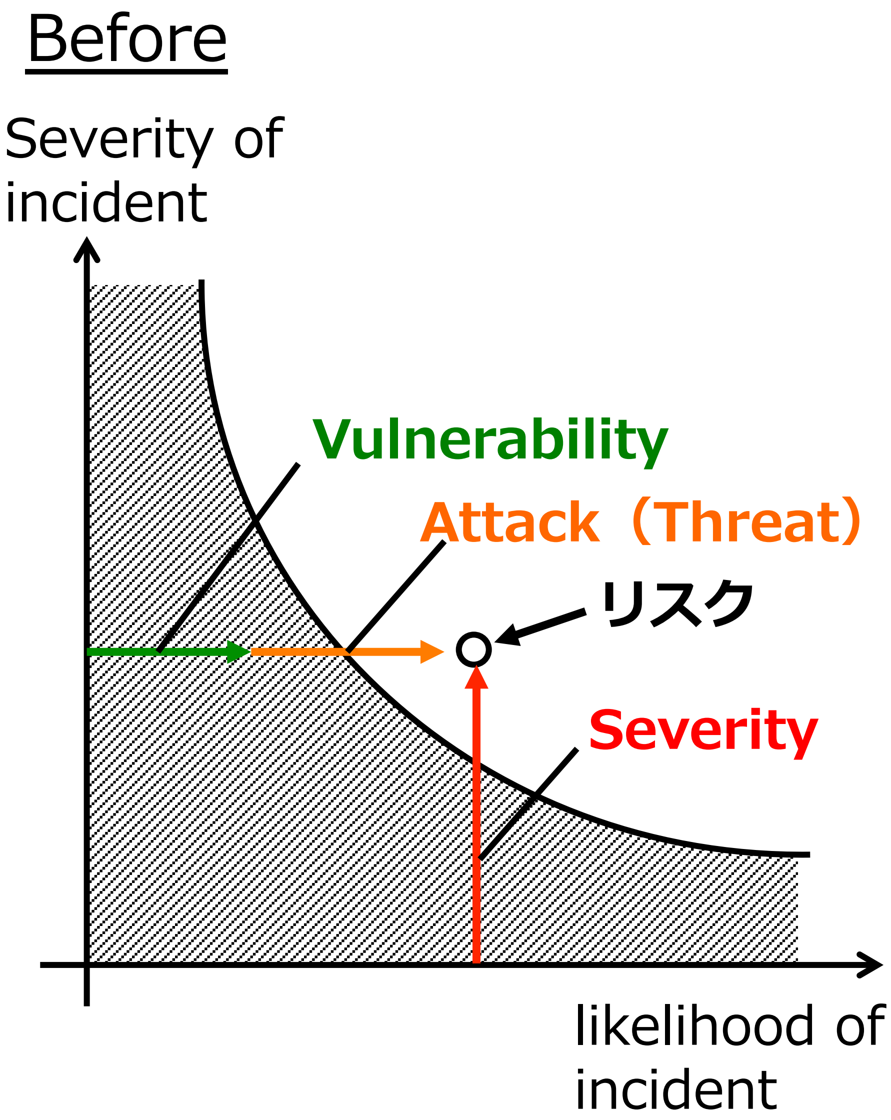
- Analyze and evaluate risks for your products and
- Choose required security level

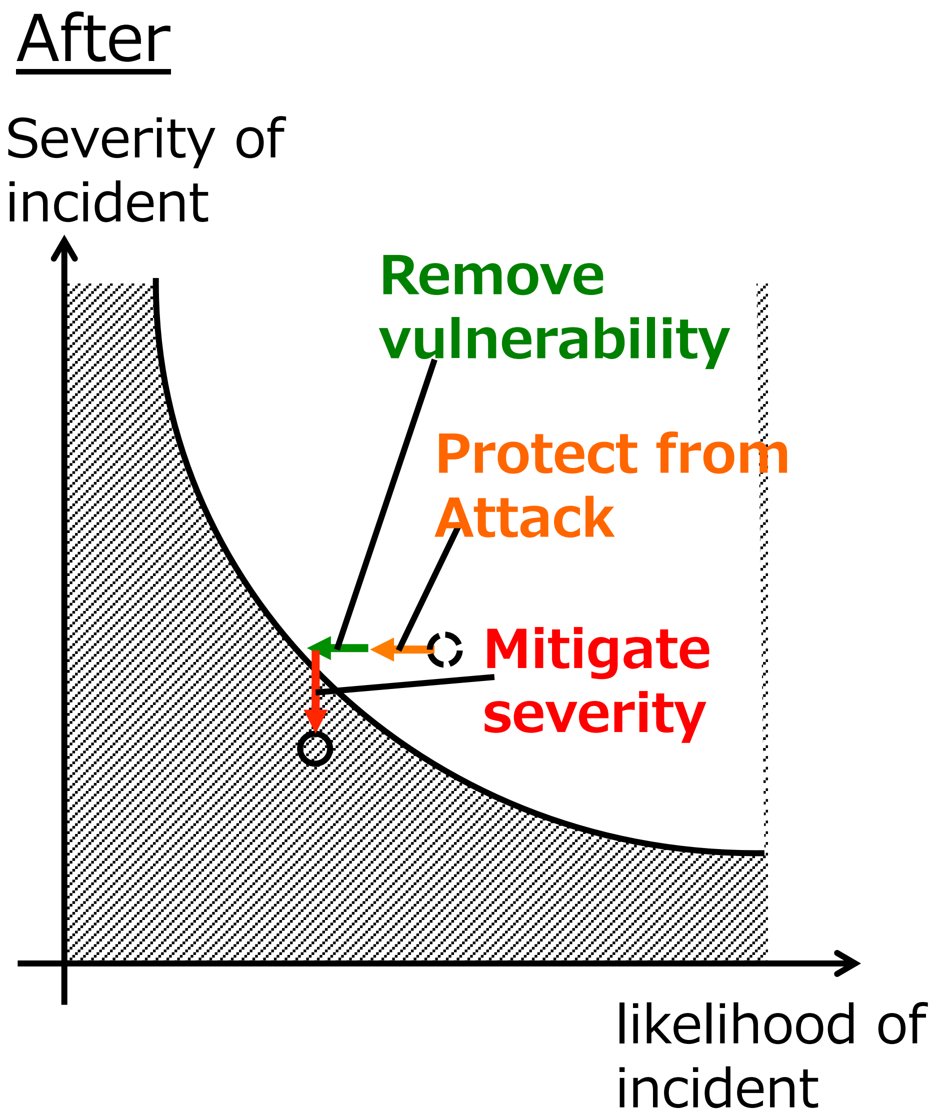
- Are requirements for security measures included?
- Are tractability valid?

Fundamental concept for Security Measures

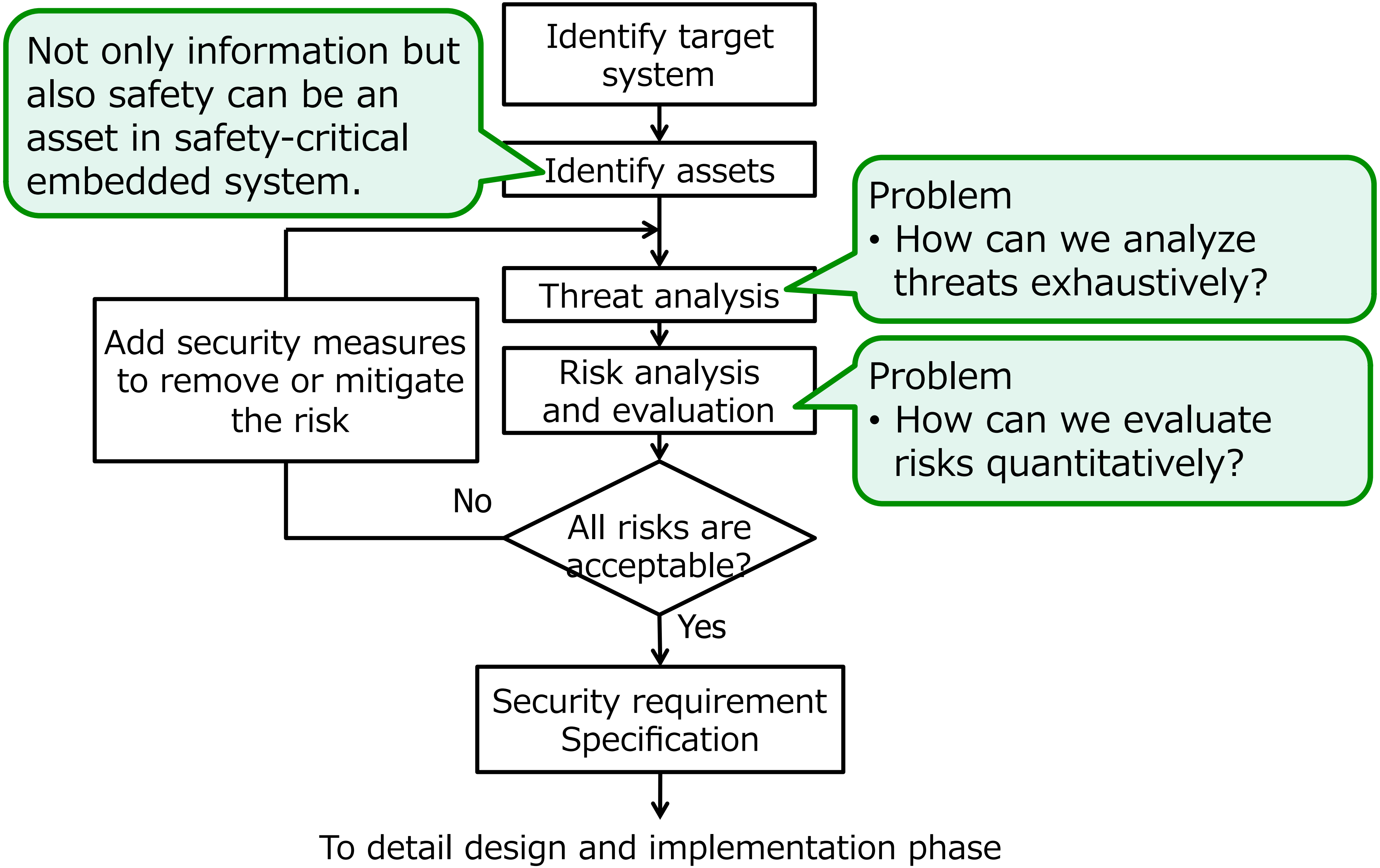

Severity of Incident
Risk
Unacceptable risk
Mitigate risk level by security measures
Boundary between acceptable and unacceptable area
Likelihood of Incident
Area for acceptable risk

Mitigation of risk level





Considerable Point in Practical Design Phase



Not only information but also safety can be an asset in safety-critical embedded system.

Problem
- How can we analyze threats exhaustively?

Problem
- How can we evaluate risks quantitatively?

Identify target system
Identify assets
Threat analysis
Risk analysis and evaluation
All risks are acceptable?
No
Add security measures to remove or mitigate the risk
Yes
Security requirement Specification
To detail design and implementation phase

Security (Threat) Analysis

Objectives

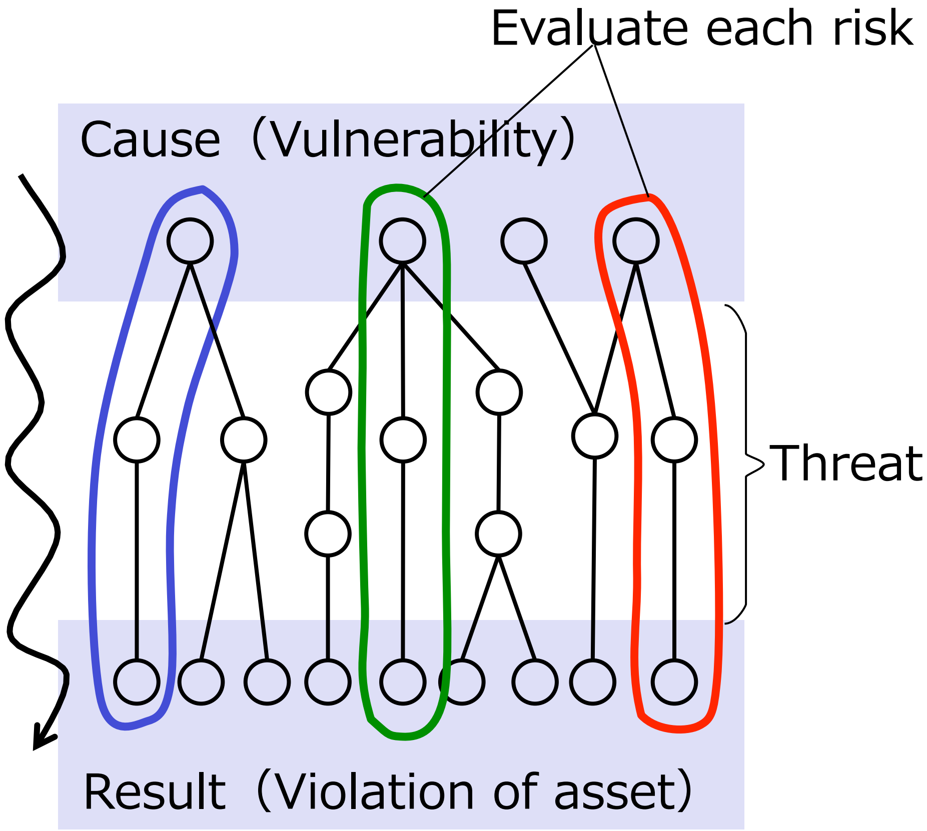
- Identify security threads regarding a target system, and document results
- Find unintended vulnerabilities
- Threat analysis should be performed each design phase repeatedly

Problems

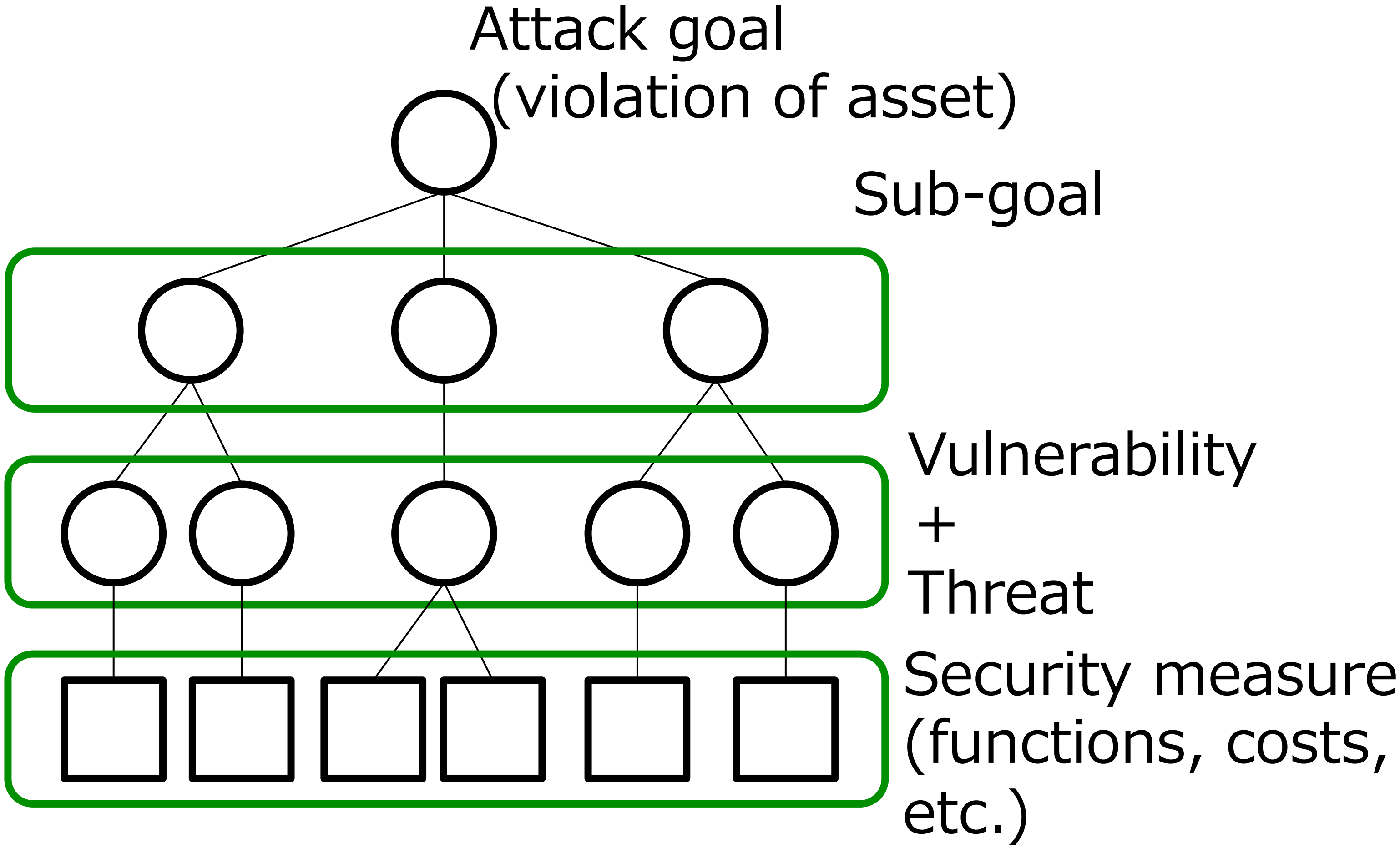
- How can we analyze security threats exhaustively?

  → Analysis should be performed from several viewpoints by using multiple analysis methods

  - *As is the case with safety analysis*



Top-down approach: Attack Tree

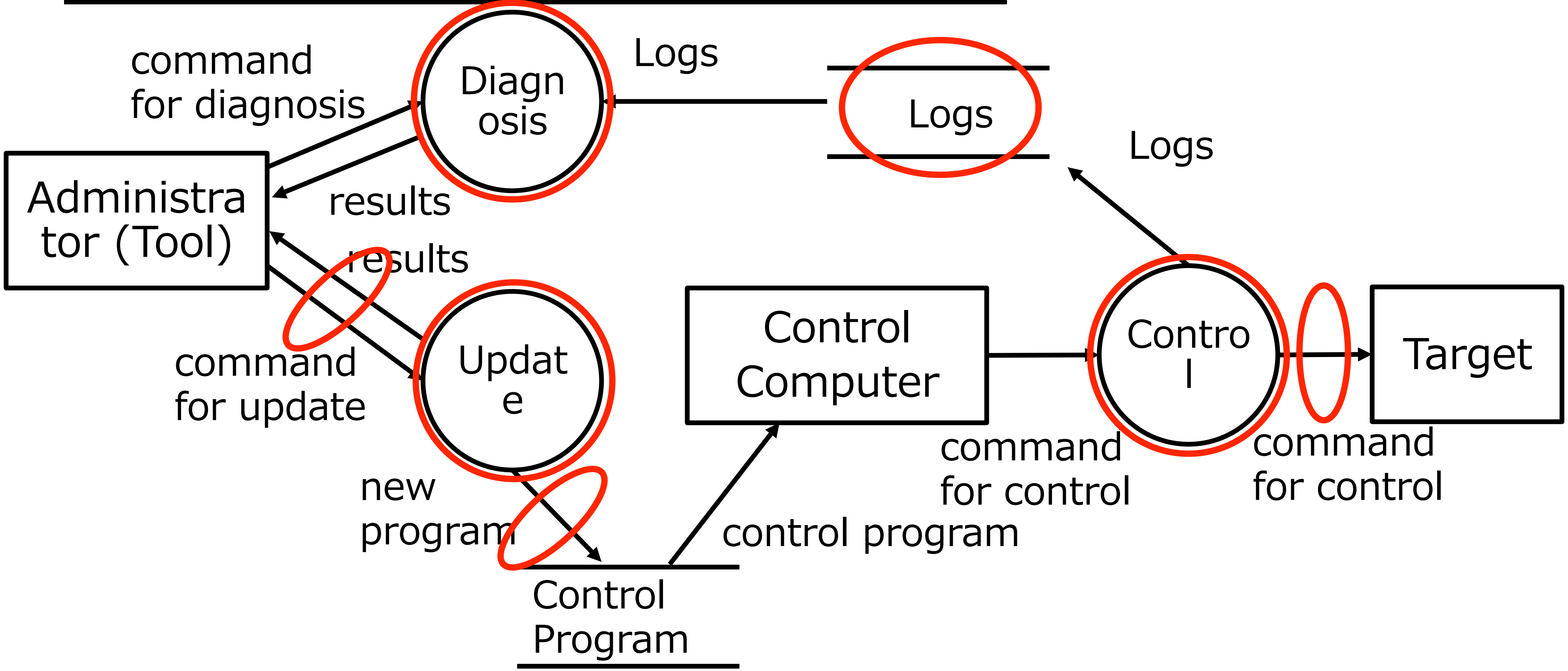


From attacker’s view, vulnerability and attack methods are analyzed.

Problem : It’s very hard for designers who do not have sufficient knowledge about security attacks to analyze exhaustively

HAZOP-based methods

Analysis for DFD (Data Flow Diagram)



- Analyze effects for violations in confidentiality, integrity and availability of each data and process
- Analyze vulnerability which can lead unacceptable violations

→***Easy to use compared to top-down approach***

Guidewords to support threat analysis

STRIDE

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

http://msdn.microsoft.com/ja-jp/magazine/cc163519.aspx

→***This method is for IT-security***

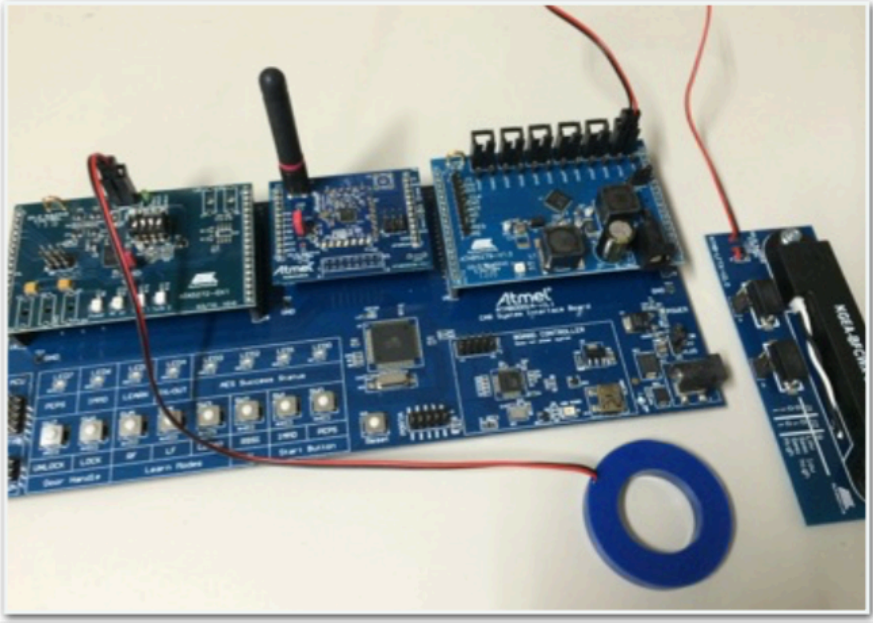
Our proposal

Target	**Guideword**
Service	Omission
	Commission
	Early
	Late
Data, Device, Commnunication	Probe
	Scan
	Flood
	Authenticate
	Spoof
	Bypass
	Modify
	Read

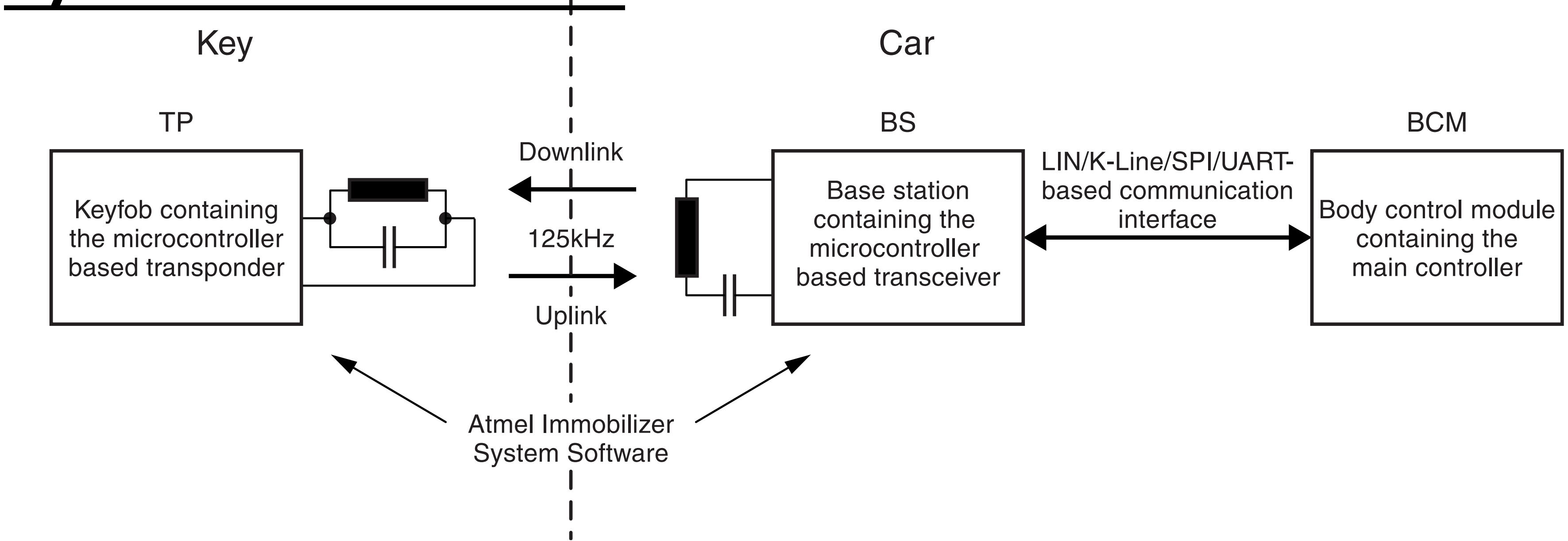
Case Study: Open Source Immobilizer

Target System

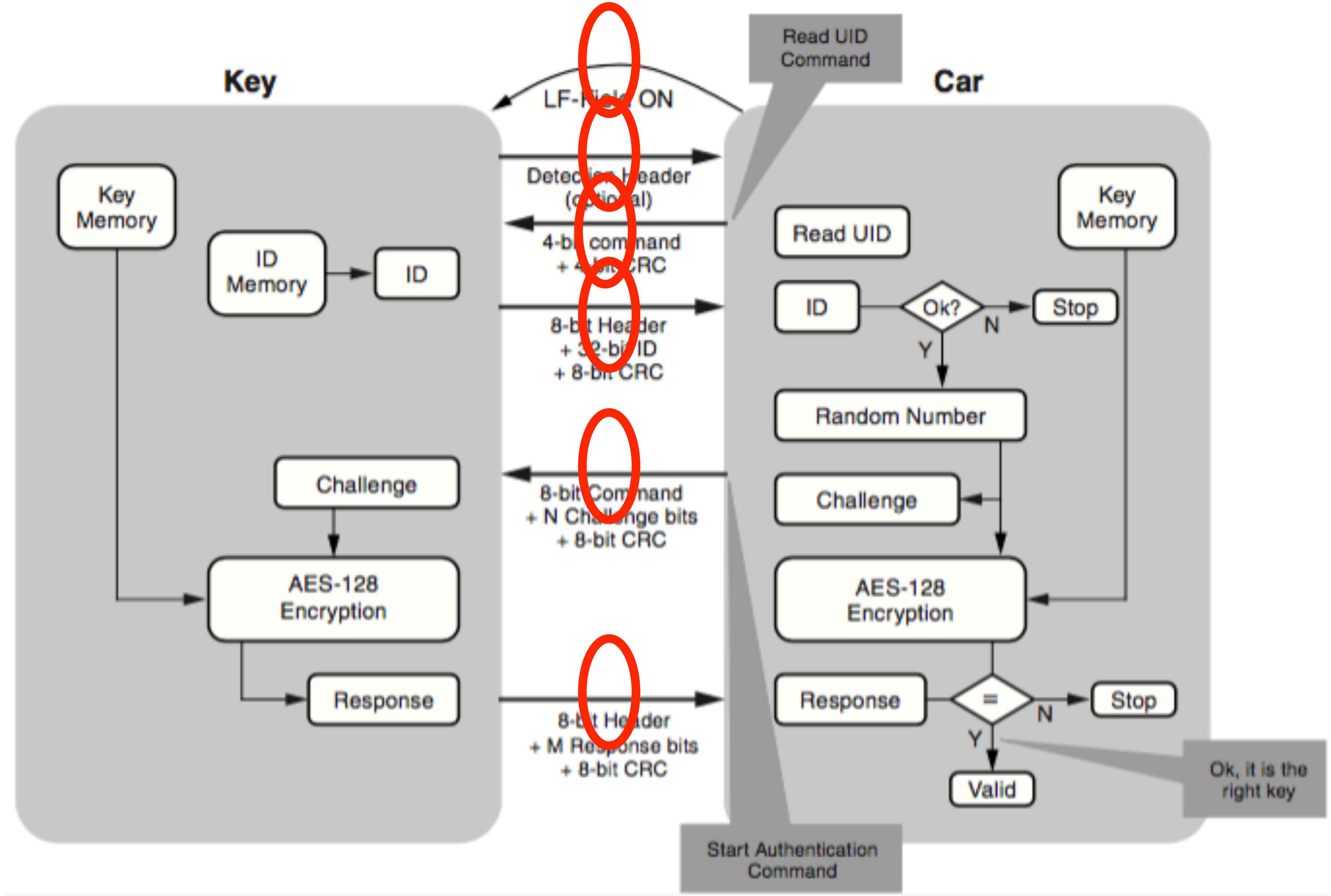
- Open source immobilizer prototype system by Atmel
- All document and software are opened



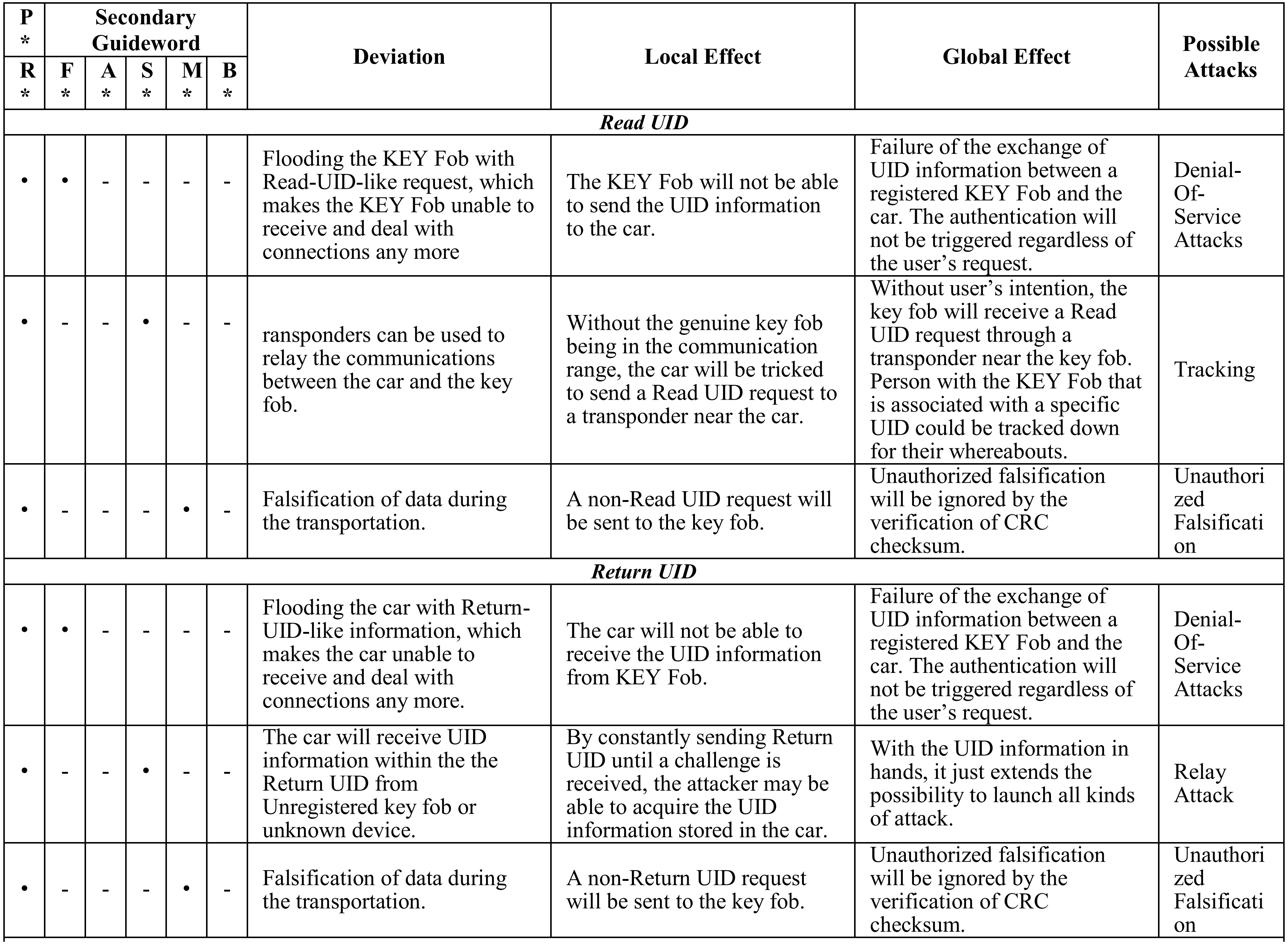
System Construction


Key
Car
TP
BS
BCM
Keyfob containing the microcontroller based transponder
Downlink
125kHz
Uplink
Base station containing the microcontroller based transceiver
LIN/K-Line/SPI/UART-based communication interface
Body control module containing the main controller
Atmel Immobilizer System Software

Analysis using sequence diagram


Key
Car
Read UID Command
LF-Field ON
Detection Header (optional)
4-bit command + 4-bit CRC
8-bit Header + 32-bit ID + 8-bit CRC
8-bit Command + N Challenge bits + 8-bit CRC
8-bit Header + M Response bits + 8-bit CRC
Key Memory
ID Memory
ID
Challenge
AES-128 Encryption
Response
Read UID
Key Memory
ID
Ok?
N
Stop
Y
Random Number
Challenge
AES-128 Encryption
Response
=
N
Stop
Y
Valid
Ok, it is the right key
Start Authentication Command

Part of Results



P *	Secondary Guideword					Deviation	Local Effect	Global Effect	Possible Attacks
R *	F *	A *	S *	M *	B *				
Read UID									
•	•	-	-	-	-	Flooding the KEY Fob with Read-UID-like request, which makes the KEY Fob unable to receive and deal with connections any more	The KEY Fob will not be able to send the UID information to the car.	Failure of the exchange of UID information between a registered KEY Fob and the car. The authentication will not be triggered regardless of the user's request.	Denial-Of-Service Attacks
•	-	-	•	-	-	ransponders can be used to relay the communications between the car and the key fob.	Without the genuine key fob being in the communication range, the car will be tricked to send a Read UID request to a transponder near the car.	Without user's intention, the key fob will receive a Read UID request through a transponder near the key fob. Person with the KEY Fob that is associated with a specific UID could be tracked down for their whereabouts.	Tracking
•	-	-	-	•	-	Falsification of data during the transportation.	A non-Read UID request will be sent to the key fob.	Unauthorized falsification will be ignored by the verification of CRC checksum.	Unauthorized Falsification
Return UID									
•	•	-	-	-	-	Flooding the car with Return-UID-like information, which makes the car unable to receive and deal with connections any more.	The car will not be able to receive the UID information from KEY Fob.	Failure of the exchange of UID information between a registered KEY Fob and the car. The authentication will not be triggered regardless of the user's request.	Denial-Of-Service Attacks
•	-	-	•	-	-	The car will receive UID information within the the Return UID from Unregistered key fob or unknown device.	By constantly sending Return UID until a challenge is received, the attacker may be able to acquire the UID information stored in the car.	With the UID information in hands, it just extends the possibility to launch all kinds of attack.	Relay Attack
•	-	-	-	•	-	Falsification of data during the transportation.	A non-Return UID request will be sent to the key fob.	Unauthorized falsification will be ignored by the verification of CRC checksum.	Unauthorized Falsification

J. Wei, Y. Matsubara, H. Takada, "HAZOP-based Security Analysis for Embedded Systems: Case Study of Open Source Immobilizer Protocol Stack", IWSSS2015, Jun 2015.

Security Measures in production, operation, abolish phase

Production phase

- Identify confidential information (e.g. firmware, key, password, etc.) for protection of assets
- Manage who can access confidential information
- Prevent leak of confidential information and unintended update

Operation phase

- Manage information about vulnerability, threat found after release of products
- Manage software update process

Abolish phase

- Manage how to remove or abolish confidential information included in products
    - Someone may resell your products

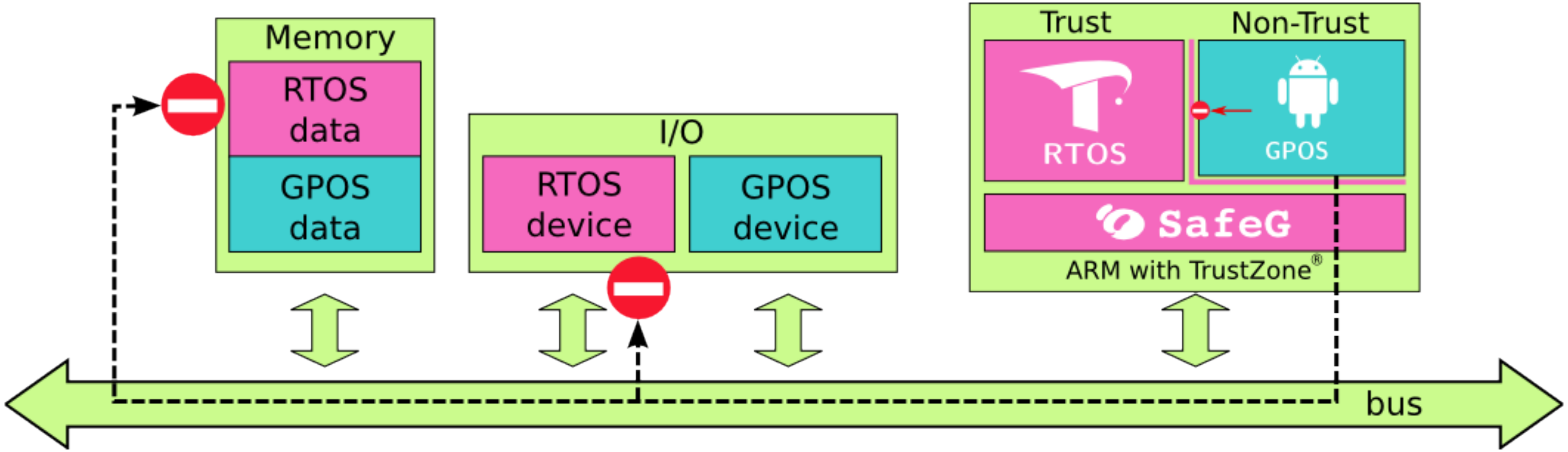
Software Platform for Safety and Security Critical embedded systems

Functional Requirements for software platform for IoT devices

- Security Libraries
  - Crypto library, TLS/DTLS, IPSec, …
- Diverse communication protocols
  - CoAP, MQTT, LWM2M, REST/HTTP,…
- **Protection of software platform**
  - Memory protection, Time protection, …
- Updating of software
  - Updating of OS, libraries, middleware, …
- Energy management
  - Power management of CPU, memory, peripherals, …

TOPPERS/SafeG

- SafeG (Safety Gate) is a dual-OS monitor designed to concurrently execute an RTOS and a GPOS on the same hardware platform.
- SafeG's architecture takes advantage of the ARM TrustZone security extensions which introduce the concept of Trust and Non-Trust states.



https://www.toppers.jp/en/safeg.html

Motivations of New Temporal Partitioning Scheme

Increasing Necessity of Partitioning Function

- For efficient support for functional safety, partitioning function is important for saving software development and verification cost.
- A key technology for application integration (ECU integration)

Lack of *Good* Partitioning Standard

- Timing protection of AUTOSAR has some problems.
  - *Both Vector and EB do not rely on it.*
- ARINC 653 (a standard for avionics systems) approach is too strict for automotive systems.

Necessity of a Standard

- We need a standard partitioning scheme applicable to different RTOS.
  - *We would like to apply it to both ITRON and AUTOSAR.*

Problems of AUTOSAR Timing Protection

Unit for timing protection is too small

- Unit of protection should be partition, rather than tasks and ISRs.
- This causes following two problems.

Schedulability analysis becomes pessimistic

- The max. execution time of the protection hook should be added to the max. exec. time of each task/ISR.

Mode change is not supported

- This problem is serious when a partition is terminated or restarted (how to schedule the restart task?).

Timing protection violation within a trusted function

Complicated specification and implementation

- eg. *DisableAllInterrupts* does not disable all interrupts.

Proposed Temporal Partitioning Scheme

Timing protection by the unit of partition

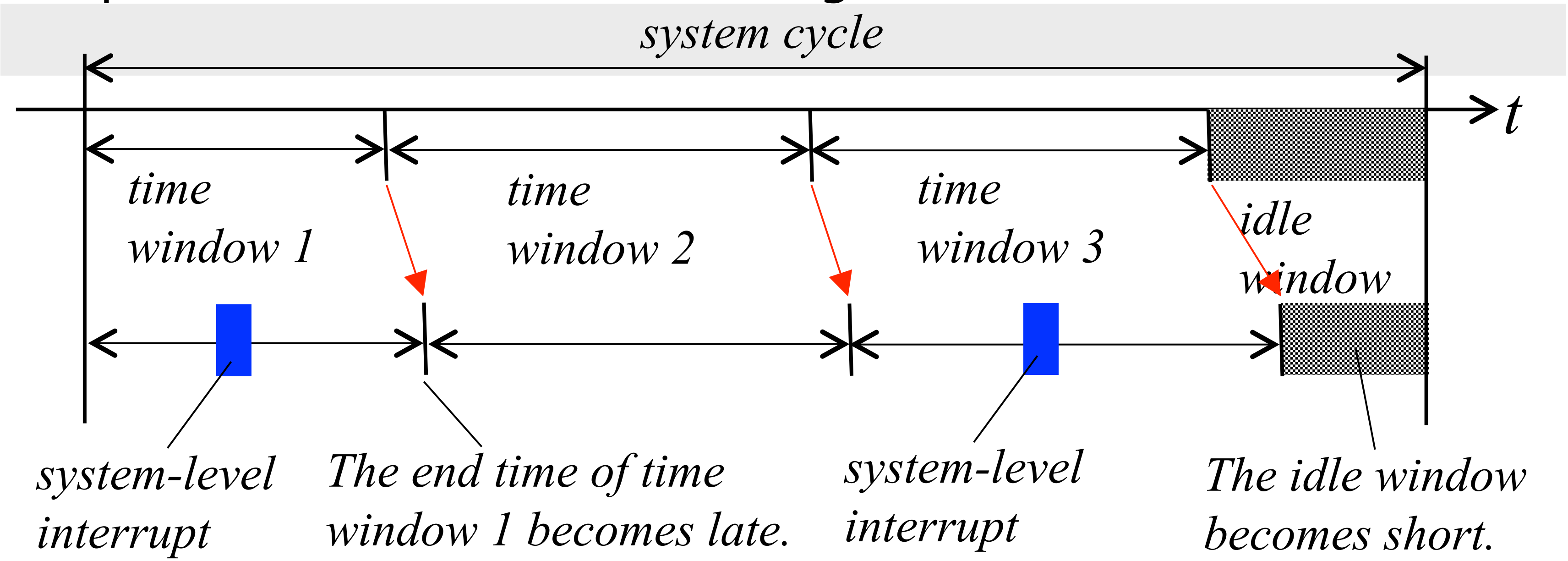
- Extend the ARINC 653 scheme to accommodate system-level interrupts.
  - Interference due to system-level interrupts need to be permitted (not "*strict partitioning*").
  - Monitoring functions for system-level interrupts may be added.

Restricted use of privileged services

- Privileged services for accessing shared resource/device are permissible, but their usage should be restricted.
- *There is an opinion to totally remove the function of privileged services.*

Overview of the Proposed Scheme

- The system cycle is divided into several time windows.
- Each time window is assigned to a partition. The partition is executed with the assigned time windows.
- The idle window is placed at the last of the system cycle and is not assigned to a partition.
- A system-level interrupt does not belong to any partition and is executed regardless of the time window.

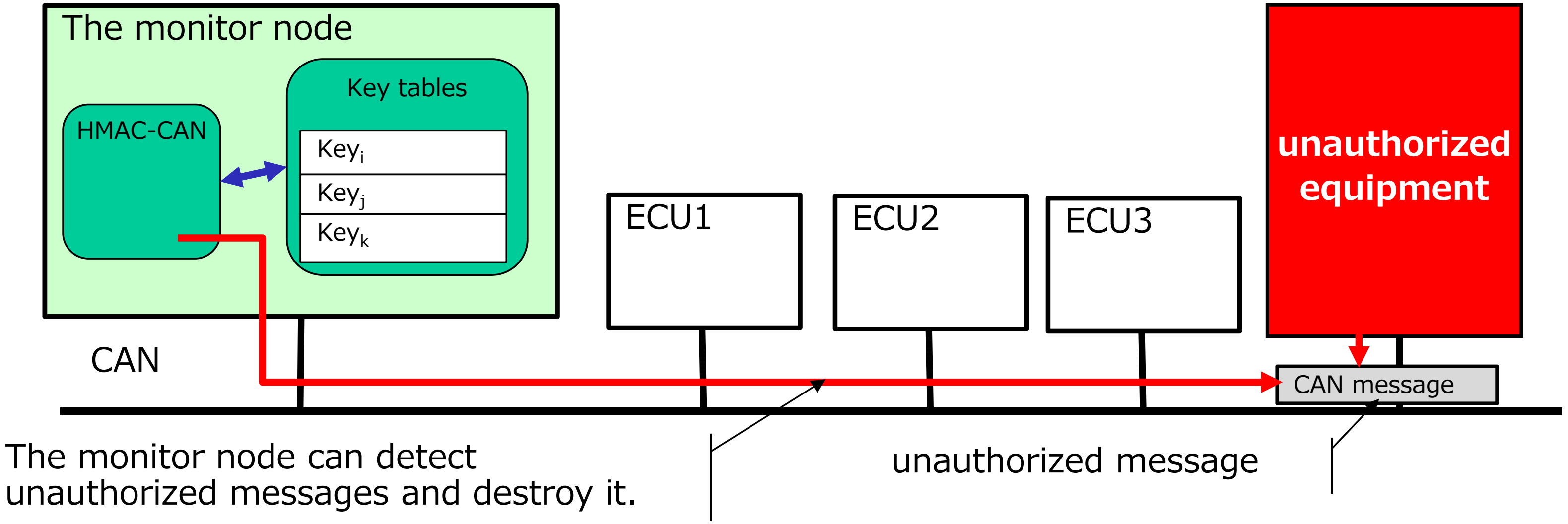


The new scheme will be employed in both of TOPPERS/HRP3 and ATK2 Kernel.

CaCAN (Centralized authentication for CAN network)

Our protocol is designed to authenticate between a monitor node and other ECUs.

- Number of authentication messages = 3 × Number of sending nodes with MACs
- The monitor node has the specialized CAN controller (named HMAC-CAN), but other existing nodes does not change.
- The HMAC-CAN controller can destroy unauthorized message by overwriting it by error frame.



Ryo Kurachi, Yutaka Matsubara, Hiroaki Takada, Naoki Adachi, Yukihiro Miyashita and Satoshi Horihata, "CaCAN - Centralized Authentication System in CAN", ESCAR 2014 Nov 2014.

Summary

- Trends regarding embedded system security are remarkable
- “Security by Design” is important for embedded systems especially for IoT devices
- We have developed and distributed software platform for safety and security critical embedded system including automotive control system

Collaborative project or discussion would be welcome!