Cyber-Security for Controller Area Network and its Security-Aware Mapping

Chung-Wei Lin
cwlin@eecs.berkeley.edu

University of California, Berkeley

DREAMS Seminar, September, 2013

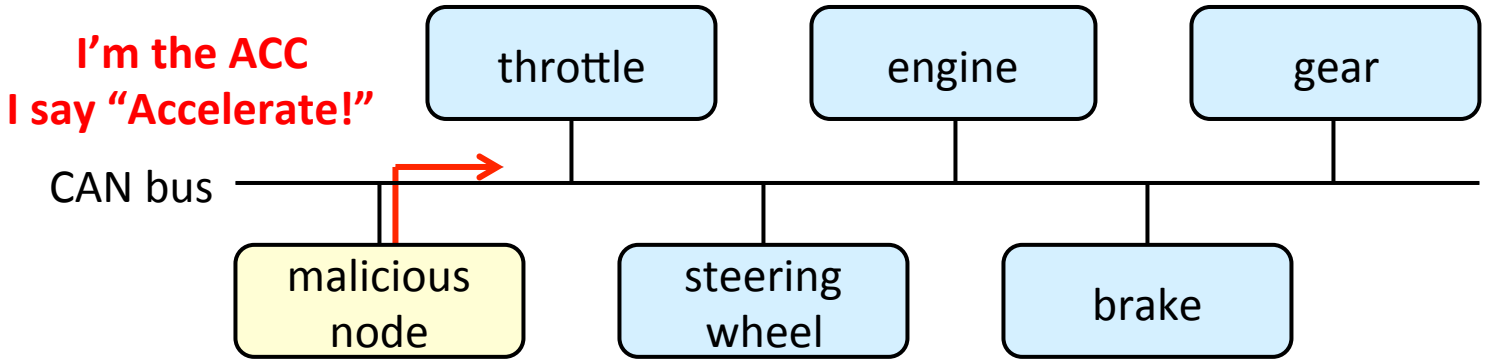
Sponsored by the TerraSwarm Research Center, one of six centers administered by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

Cyber-Security for Automotive Systems

- Cyber-security is a rising issue for automotive systems
  - Modern automotive systems are distributed as networked computers
  - They have more and more interactions with its outside environment, driver, or passengers



- We focus on the Controller Area Network (CAN) protocol
  - It is the most used protocol in current in-vehicle networks
  - It will likely be used for a long time to come in the future

Our Contribution

- We propose a security mechanism for CAN
  - Add Message Authentication Codes (MACs) to messages
- However, adding MACs to an existing design may not lead to optimal or even feasible systems
  - The space in messages may not be enough for MACs
  - The message transmission time increases, which may violate timing constraints and affect system safety
- We further propose an MILP formulation to meet both the security and the safety requirements
  - This is the first work to address security and safety in an integrated formulation in the design automation of automotive systems

Outline

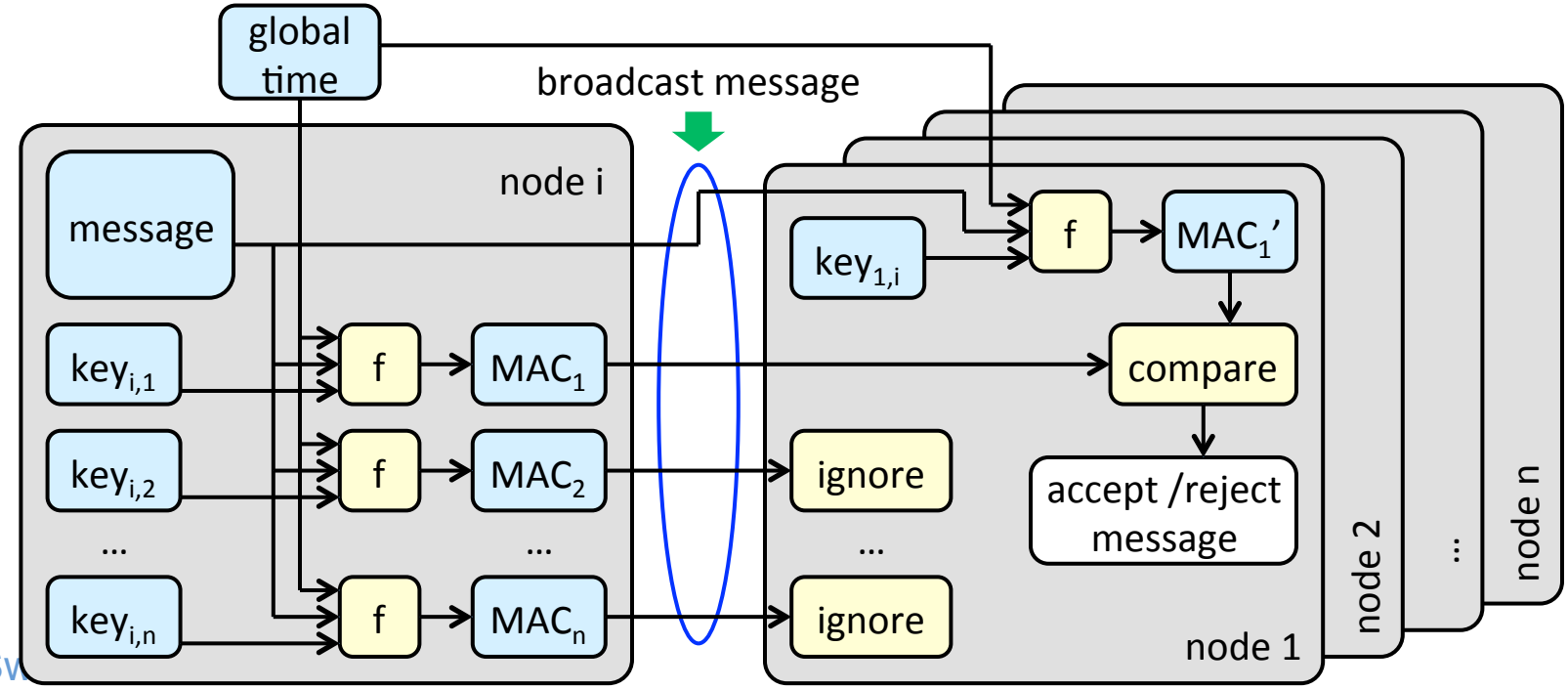
- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

Types of Attacks and Desired Properties

- Types of attacks
  - Interception: unauthorized nodes read data
  - Modification: unauthorized nodes change data
  - Fabrication: unauthorized nodes generate additional data
    - A special case: replay attack
  - Interruption: data becomes unavailable
- Desired properties
  - Confidentiality: data is not read by unauthorized nodes
  - Data integrity: data is not changed by unauthorized nodes
  - Authentication: a receiver or a sender is who it claims to be
- Authentication is one of the most relevant properties for an automotive communication system

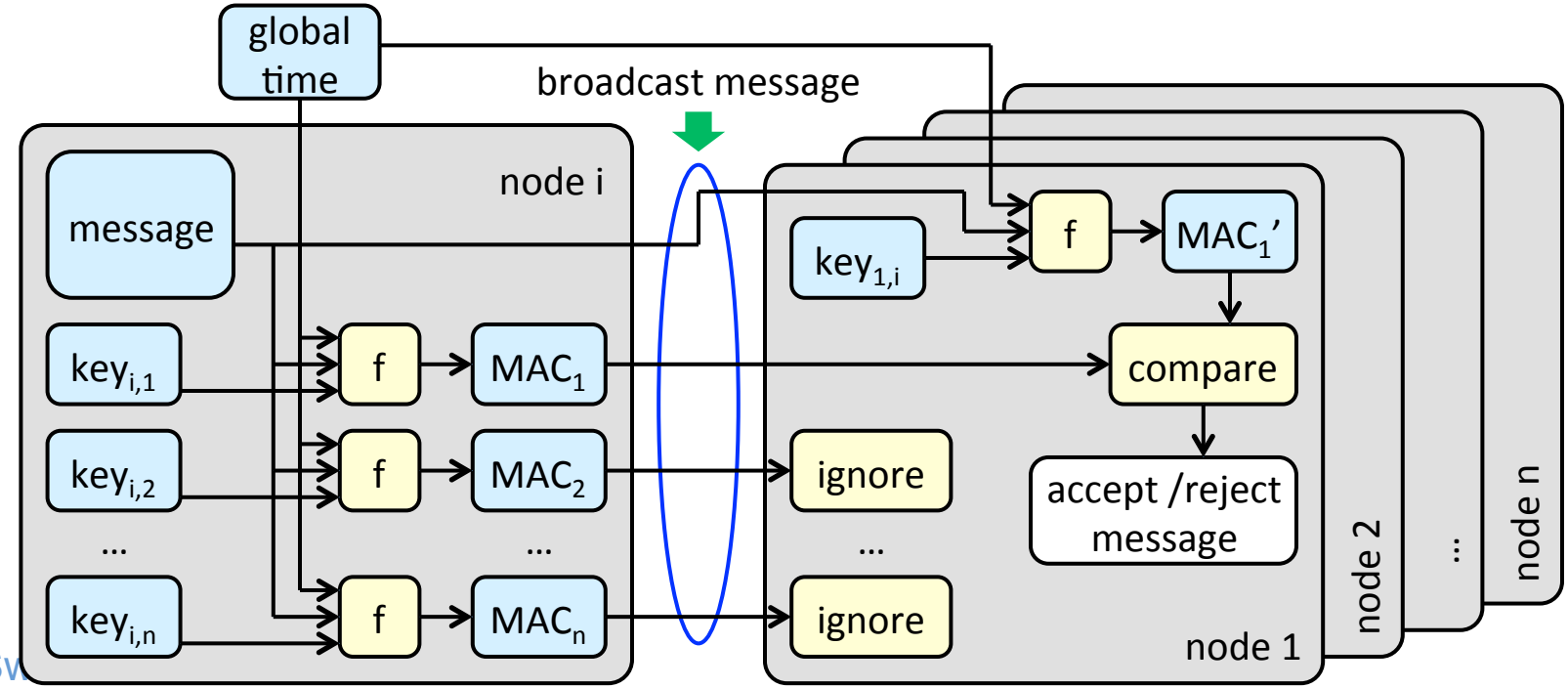
Existing Work [Szilagyi & Koopman]

- Achieve authentication in a broadcast system
  - Each pair of nodes has a shared secret key
  - A sender computes Message Authentication Codes (MACs) and broadcasts the message with the MACs
  - A receiver computes a MAC and compares it with the sent MAC



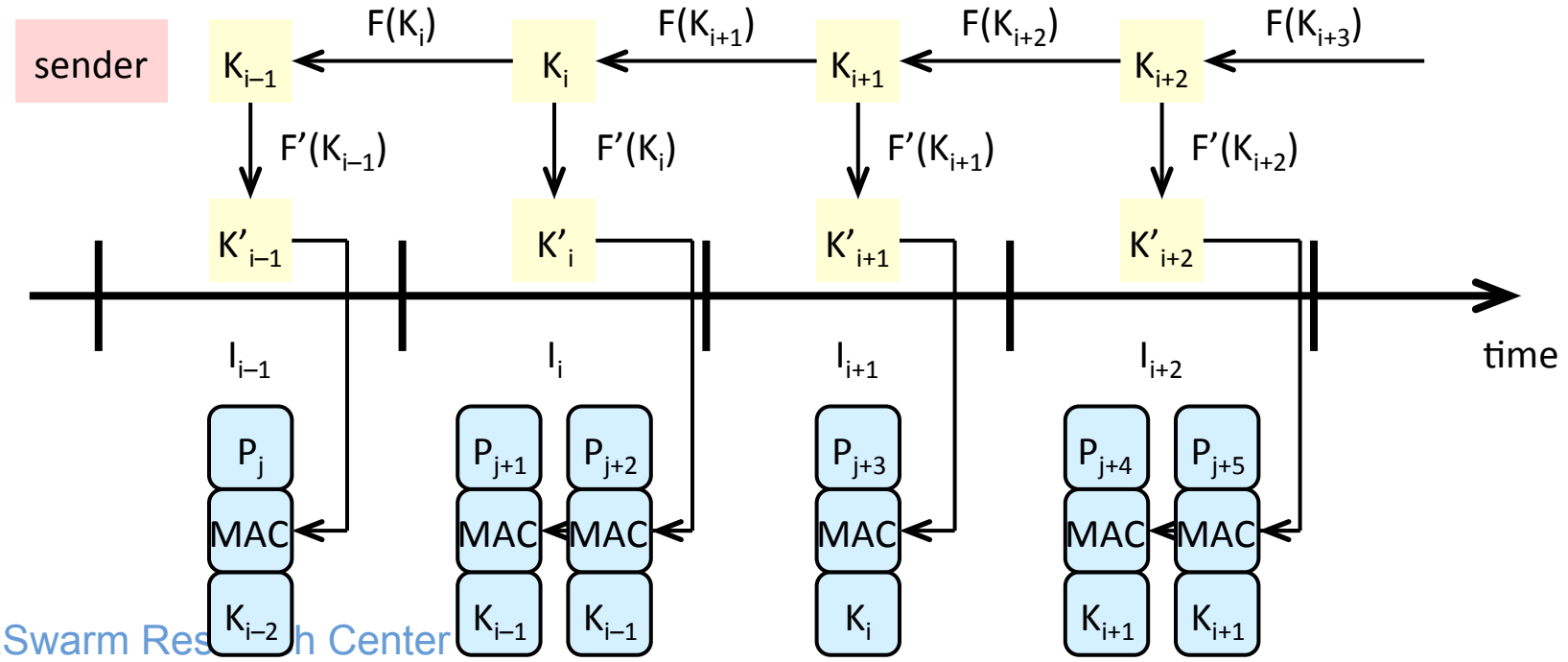
Existing Work [Szilagyi & Koopman]

- Difficulties of applying it on CAN
  - High communication overhead
    - CAN data rate: 500kbps
    - CAN payload size: 64 bits
  - Maintenance of a global time (not supported by CAN)



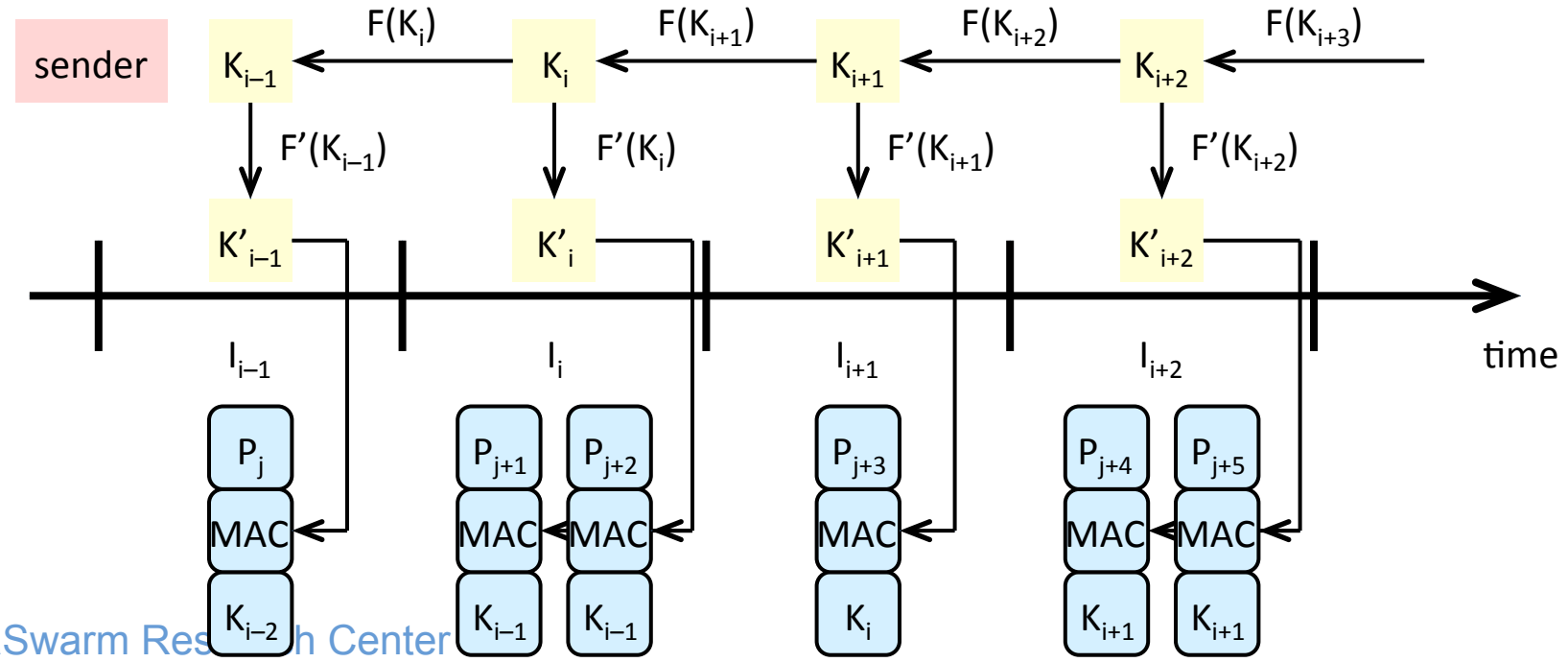
Existing Work – TESLA [Perrig et al.]

- Also achieve authentication in a broadcast system
  - A sender sends data and MAC first and then sends the corresponding key later
  - A receiver stores data and MAC first and then checks them after receiving the corresponding key



Existing Work – TESLA [Perrig et al.]

- Difficulties of applying it on CAN
  - Increasing message latency
    - CAN data rate: 500kbps
    - CAN payload size: 64 bits
  - Maintenance of a global time (not supported by CAN)



Outline

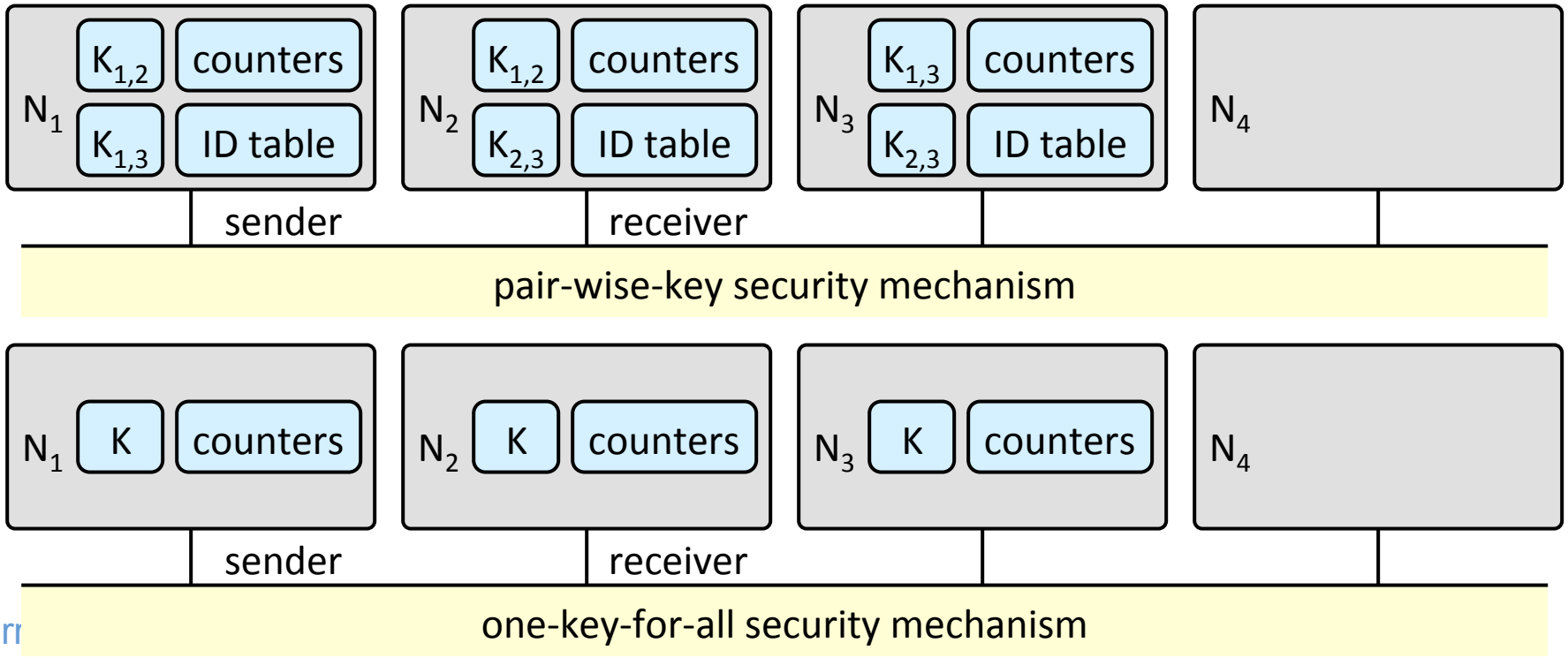
- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

System Model

- There is only one CAN bus, and all nodes (ECUs) are connected to the bus
  - The **sender** of a message is the node sending the message
    - It sends a message by broadcasting it on the CAN bus
  - A **receiver** of a message is a node receiving the message and accepting it by comparing the message ID to its acceptable message ID's
  - A node can use RAM and/or FLASH memory to store data
    - Data in RAM is no longer available after a node reset
    - Data in FLASH is available after a node reset
- Possible scenarios
  - Unexpected reset of a node
  - Expected reset of a node
  - Network fault (message is missing)

Attacker Model

- N_3 (strong attacker) becomes malicious and can access the keys
- N_4 (weak attacker) becomes malicious but cannot access the keys



Attacker Model

- Masquerade attack
  - An attacker sends a message in which it claims to be a node other than itself
- Replay attack
  - An attacker sends a copy of a message it has received from the CAN network
  - The message is not modified or fabricated; it is merely sent to other nodes by a node not entitled to send it
- Not covered in this work
  - Denial-of-Service (DoS) attack which needs hardware solutions
  - A node sends a message which is supposed to be sent by the node itself but the data has been modified

Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

Secret Key and Counter Assignment

- Pair-wise secret keys
  - For each pair of nodes N_i and N_j, they share a secret key $K_{i,j}$
  - All keys are stored in FLASH
- Message-based counters
  - For each message M_i, there is a counter C_i stored in its sender and all of its receivers
    - It is called **sending counter** at the sender side
    - It is called **receiving counter** at a receiver side
    - The values of a sending counter and a receiving counter of M_i may be different due to network faults
  - All counters are stored in RAM but copied to FLASH periodically
    - Crucial for reset mechanisms
    - Compatible with the FLASH burning rate

Basic Operations

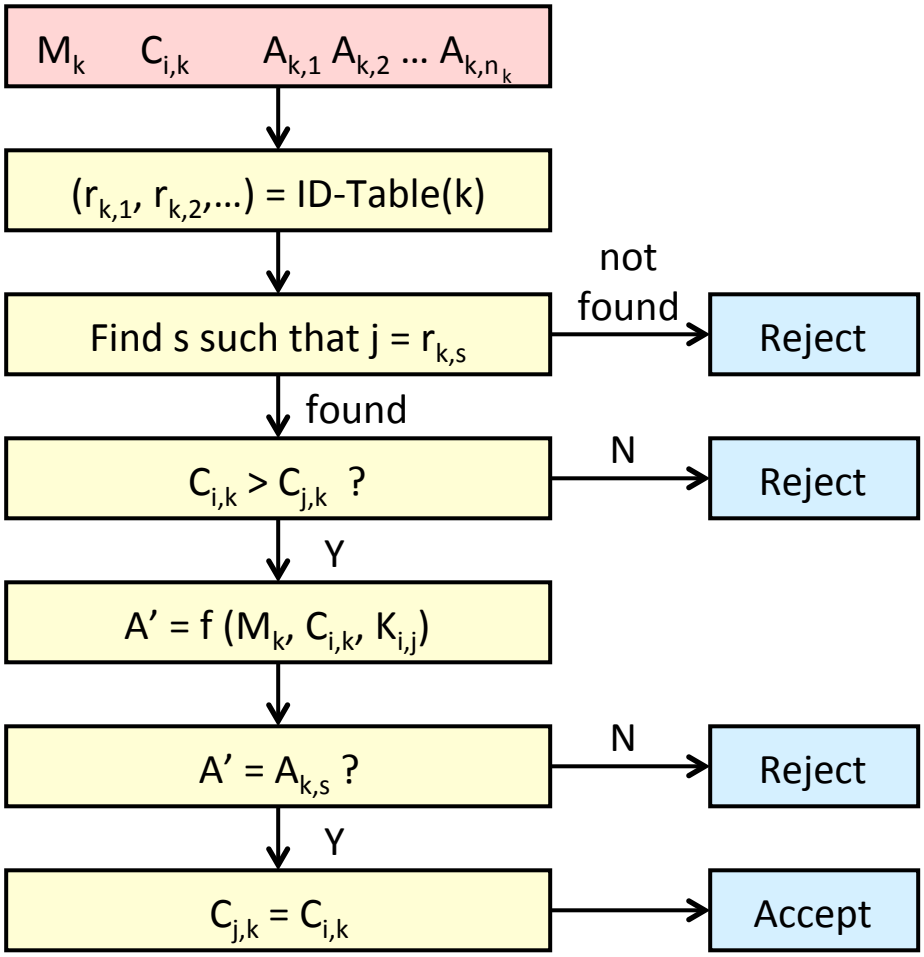
- Sender
  1. Increases the sending counter
  2. Computes the MACs for receivers
     - Uses the message, the sending counter, and the keys
  3. Broadcasts the message, the sending counter, and the MACs

message	counter	MACs

payload format

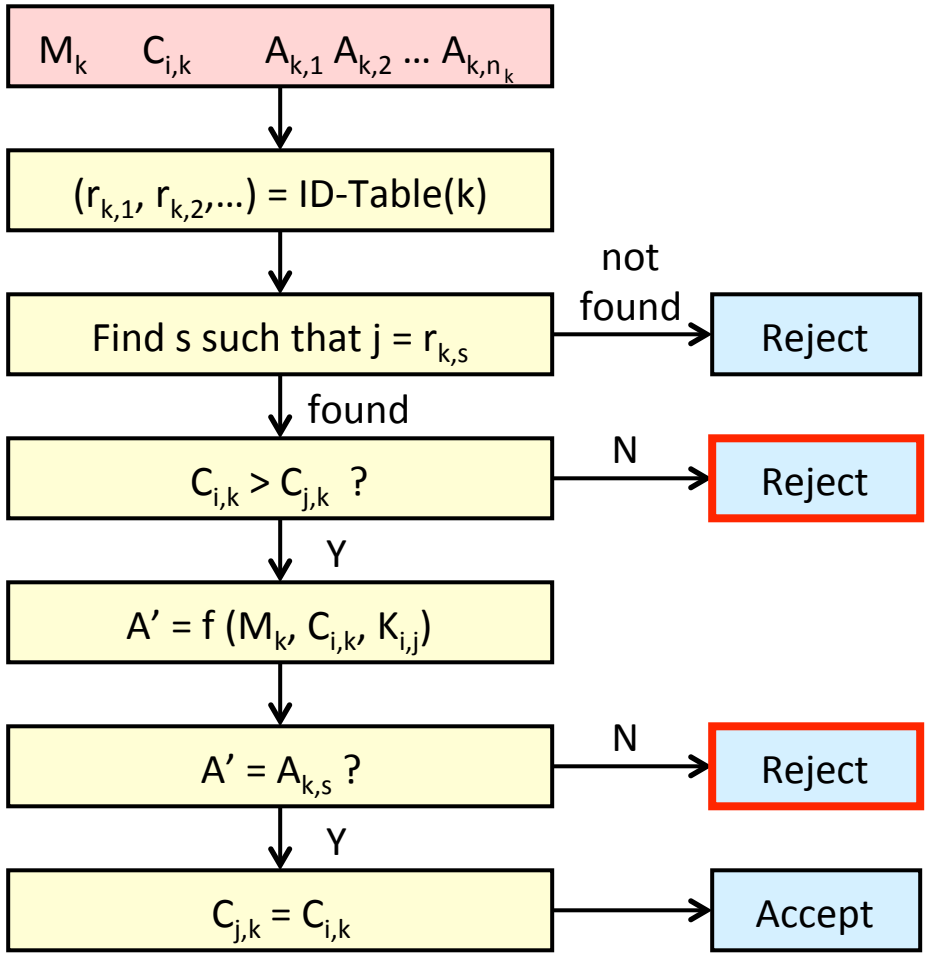
- Receiver
  1. Checks its ID table to decide which key and counter to be used
  2. Checks if the sending counter > the receiving (stored) counter
     - Is the message fresh?
  3. Computes MAC'
     - Uses the (received) message, the sending (received) counter, and the (stored) key
  4. Checks if MAC' is equal to the received MAC
  5. Updates the receiving counter

Basic Operations – Receiving Flow



N_i	node i (sender)
N_j	node j (receiver)
M_k	message k
$C_{i,k}$	sending counter for M_k (stored in N_i)
$C_{j,k}$	receiving counter for M_k (stored in N_j)
n_k	#receivers of M_k
$r_{k,s}$	index of the s-th receiver of M_k
$A_{k,s}$	the s-th MAC of M_k

Basic Operations – Security Guarantee



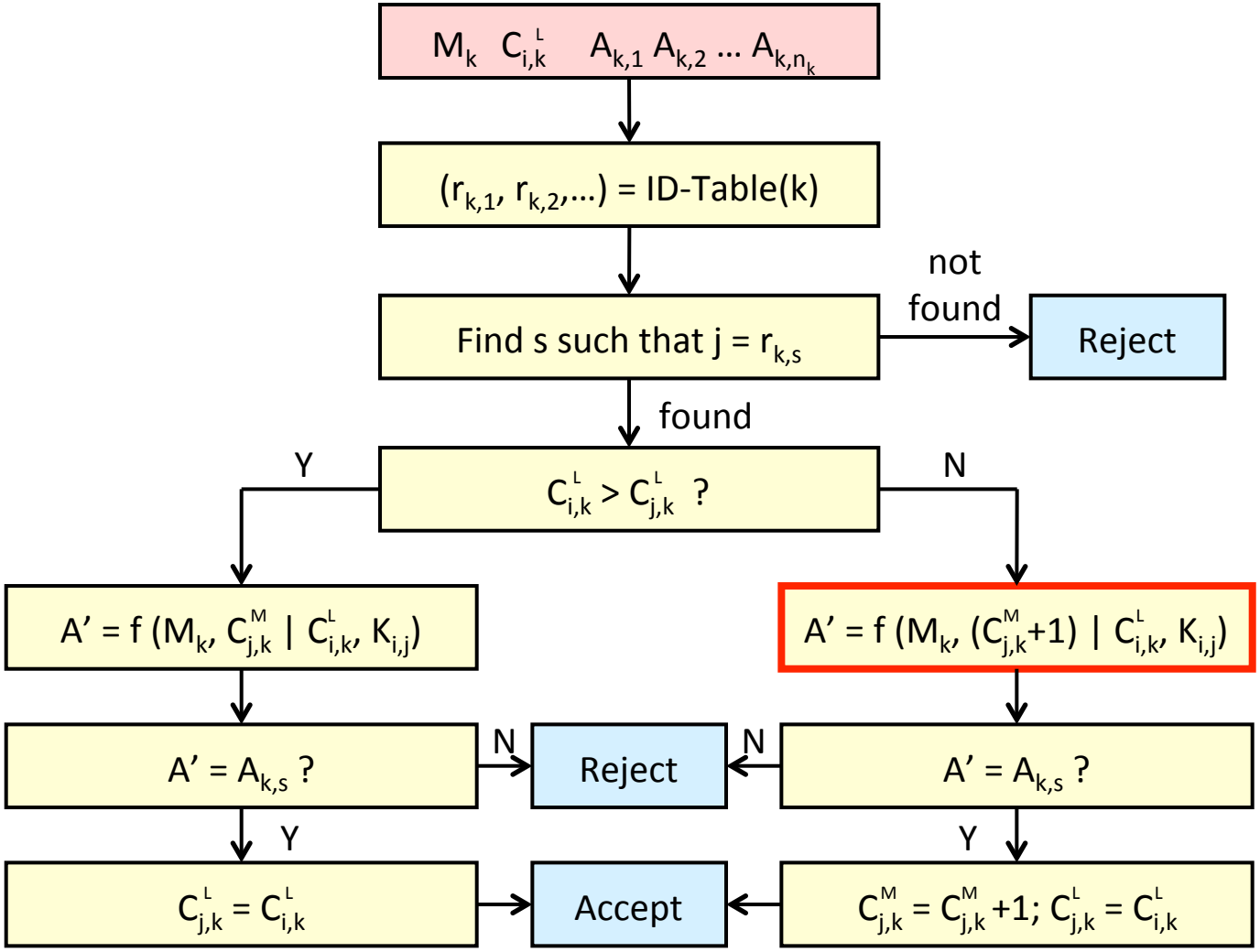
protect against a replay attack

protect against a masquerade attack

Sending Partial Counter

- We cannot afford to use many bits for the counter
  - There are only 64 bits for payload in CAN
- A counter C is divided into C^M and C^L
  - C^M: the most significant bits of C
  - C^L: the least significant bits of C
- Only C^L is sent!

Sending Partial Counter – Receiving Flow



Sending Partial Counter – Discussion

- Advantages
  - We can assign the length of a counter up to 32 (or even 64) bits so that it is never overflowed
  - The communication overhead can be much reduced
- Potential disadvantage
  - Problem
    - If $| C^L |$ = 8 bits, how does the receiver knows to update (C^M, C^L) from (0, 255) to (1, 0) or (2, 0)?
  - Solution: update (C^M, C^L) from (0, 255) to (1, 0)
    - The latter case (2, 0) happens only if the receiver misses 255 **consecutive** messages
    - Even if the worst case happens, the receiver will reject more messages than expected and try to reset counters

Reset Mechanism – Self-Healing

- A node resets by itself without using new messages
- Steps
  - FLAG = 0; a node writes counters into FLASH every P seconds
  - If a node resets
    - If it is expected, it tries to write counters into the FLASH
      - If the writing is sure to be successful, then FLAG = 1 (committing to FLASH)
      - Otherwise, the scenario is the same as that of the unexpected reset
    - If it is unexpected, … (it cannot guarantee to do anything, so of course it cannot guarantee it can write on FLASH) and FLAG stays at zero
  - When a node wakes up
    - If FLAG = 1, restore all counters from FLASH and set FLAG = 0
    - If FLAG = 0, restore all counters from FLASH (last counters saved) and **increase them by Q**, and stores them into FLASH

Q is the upper bound of the number of messages sent within the period P
Different counters can be associated with different values of Q for different messages

Reset Mechanism – Self-Healing

- Advantages
  - A node resets by itself without the need of additional messages to reset the other nodes
  - There is no security loss if Q is large enough
- Disadvantages
  - Possible (but not always) false rejections (a receiving counter may jump from C to C + Q)
  - Trade-off if Q ≠ the upper bound of #messages in P seconds
    - Q is larger, more false rejections; Q is smaller, possible replay attacks
    - Q = the upper bound of #messages sent in P, no replay attacks
- Note
  - A false rejection is just the same as a message missing due to network fault

Alternative Reset Mechanism – RESET Message

- Key concepts
  - A RESET message to set all counters of all nodes to 0
  - A REQUEST message to achieve fault tolerance
  - New session keys to prevent attacks
    - A random generated number is included in a RESET message
- Two approaches
  - Any node can generate a random number and send a RESET message to all other nodes
  - Only one "special master" node can generate a random number and send a RESET message to all other nodes

Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

Test Case and Setting

- A real industrial test case
  - 17 security-critical messages among 138 messages
- Constraints
  - The total length of MACs and LSB of the counter should be smaller than or equal to 32 bits
  - P(successful attack) $\leq$ P
    - Depends on the length of a MAC
  - P(counter out of synchronization) $\leq$ Q
    - Depends on the length of LSB of a counter

Analysis Results – #receivers = 1

- If we want to guarantee that
  - P(successful attack) $\leq 10^{-4}$
  - P(counter out of synchronization) $\leq 10^{-4}$

  then there are
  - 3% & 6.25% increase on the bus load & the average message latency

P	Q							
	10^{-1}		10^{-4}		10^{-7}		10^{-10}	
	Load	Avg L.	Load	Avg L.	Load	Avg L.	Load	Avg L.
10^{-1}	1.0094	1.0241	1.0113	1.0267	1.0131	1.0288	1.0150	1.0322
10^{-4}	1.0282	1.0591	**1.0300**	**1.0625**	1.0310	1.0646	1.0338	1.0668
10^{-7}	1.0469	1.0987	1.0488	1.1007	1.0507	1.1040	1.0526	1.1061
10^{-10}	---	---	---	---	---	---	---	---

"Avg L.": average message latency; "---": no feasible solution; original bus load 376.44kbps & average message latency 11.535ms are both scaled to 1

Analysis Results – #receivers = 3

- The feasible region is reduced
  - Because there may be no enough bits available for 3 MACs
- Implication: need to consider the trade-off between security and performance in the design stage
  - Decrease sizes of messages, or decrease #receivers of messages

P	Q							
	10^{-1}		10^{-4}		10^{-7}		10^{-10}	
	Load	Avg L.	Load	Avg L.	Load	Avg L.	Load	Avg L.
10^{-1}	1.0244	1.0506	1.0263	1.0571	1.0282	1.0591	1.0300	1.0625
10^{-2}	1.0413	1.0832	1.0432	1.0883	1.0451	1.0968	1.0469	1.0987
10^{-3}	1.0582	1.1213	1.0601	1.1232	---	---	---	---
10^{-4}	---	---	---	---	---	---	---	---

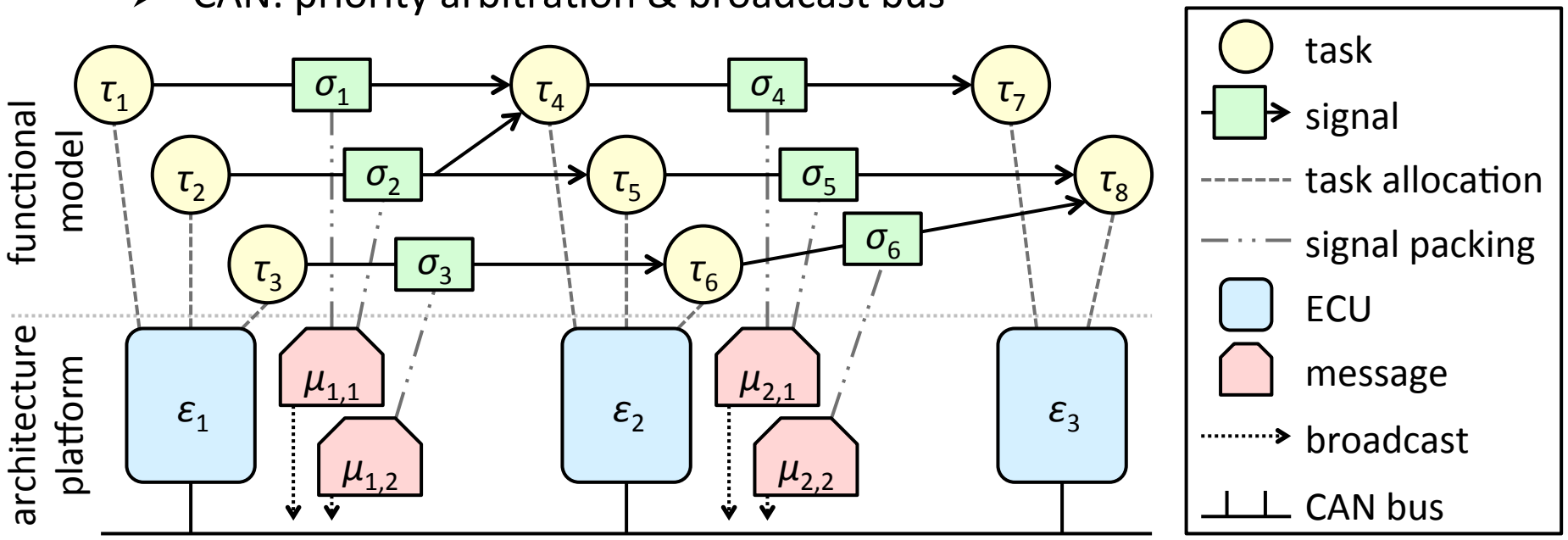
"Avg L.": average message latency; "---": no feasible solution; original bus load 376.44kbps & average message latency 11.535ms are both scaled to 1

Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

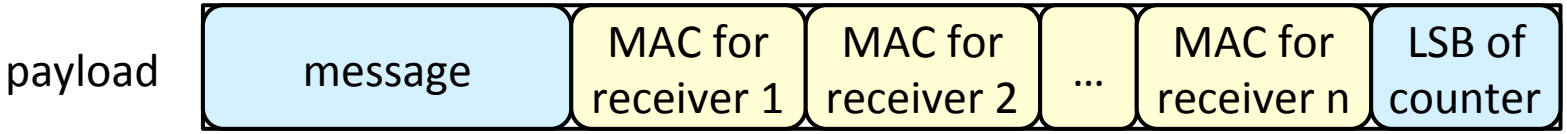
System Model

- Functional model: tasks and signals
  - Priority assignment of tasks
- Architecture model: ECUs, messages and CAN bus
  - Priority assignment of messages
  - CAN: priority arbitration & broadcast bus



Security Mechanism

- A message is sent with MACs (one for each receiver) to protect against masquerade attacks
  - Each receiver can authenticate it by checking if the corresponding MAC is equal to the MAC computed by itself
- A message is also sent with a counter to protect against replay attacks
  - Each receiver can check if the message is fresh or not

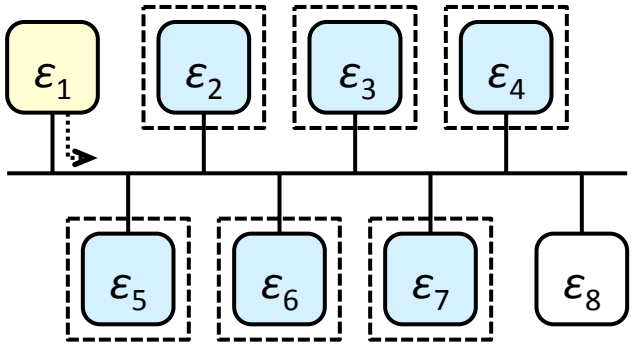


- Due to the limited size of the payload, only the least significant bits of the counter is sent with the message
  - Reset mechanisms are provided to avoid out-of-sync counters

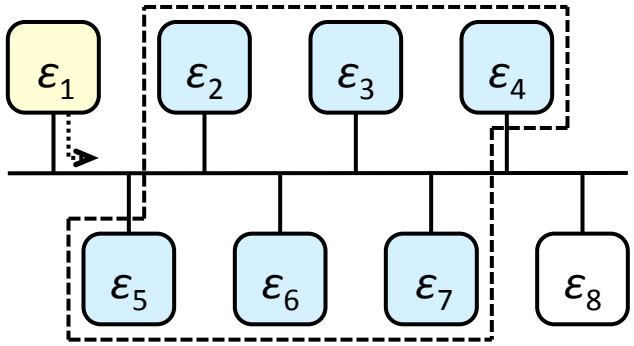
Indirect Attack and Direct Attack

- Indirect attack
  - Definition: an attacker does not have the shared secret key between a sender and a receiver
  - Result: it can only guess a MAC and attempt to make a message accepted by the receiver
- Receiving group
  - Definition: a set of receivers sharing one secret key with the sender of the message
- Direct attack
  - Definition: an attacker gets the shared secret key between a sender and a receiver
  - Result: it can pretend as the sender and send a message to the receiver

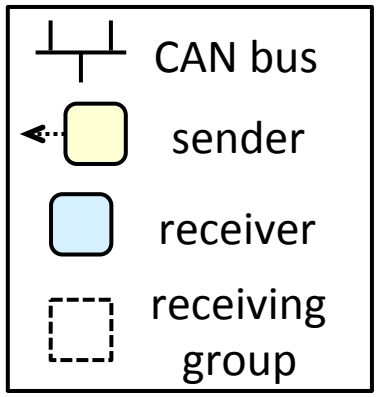
Key Distribution



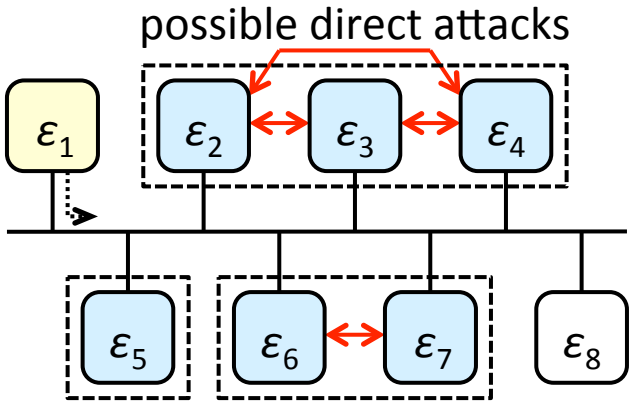
pair-wise key distribution



one-key-for-all key distribution



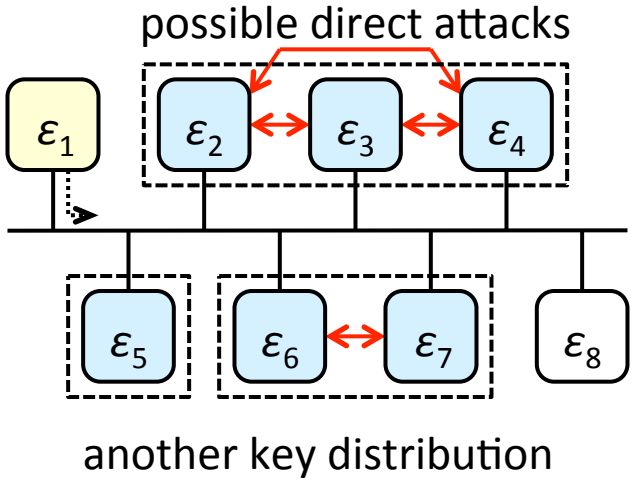
- Pair-wise key distribution
  - 6 MACs required and no direct attack
- One-key-for-all key distribution
  - Only 1 MAC required but direct attacks between any pair of receivers
- Tradeoff between security and bandwidth utilization



another key distribution

Security Constraints

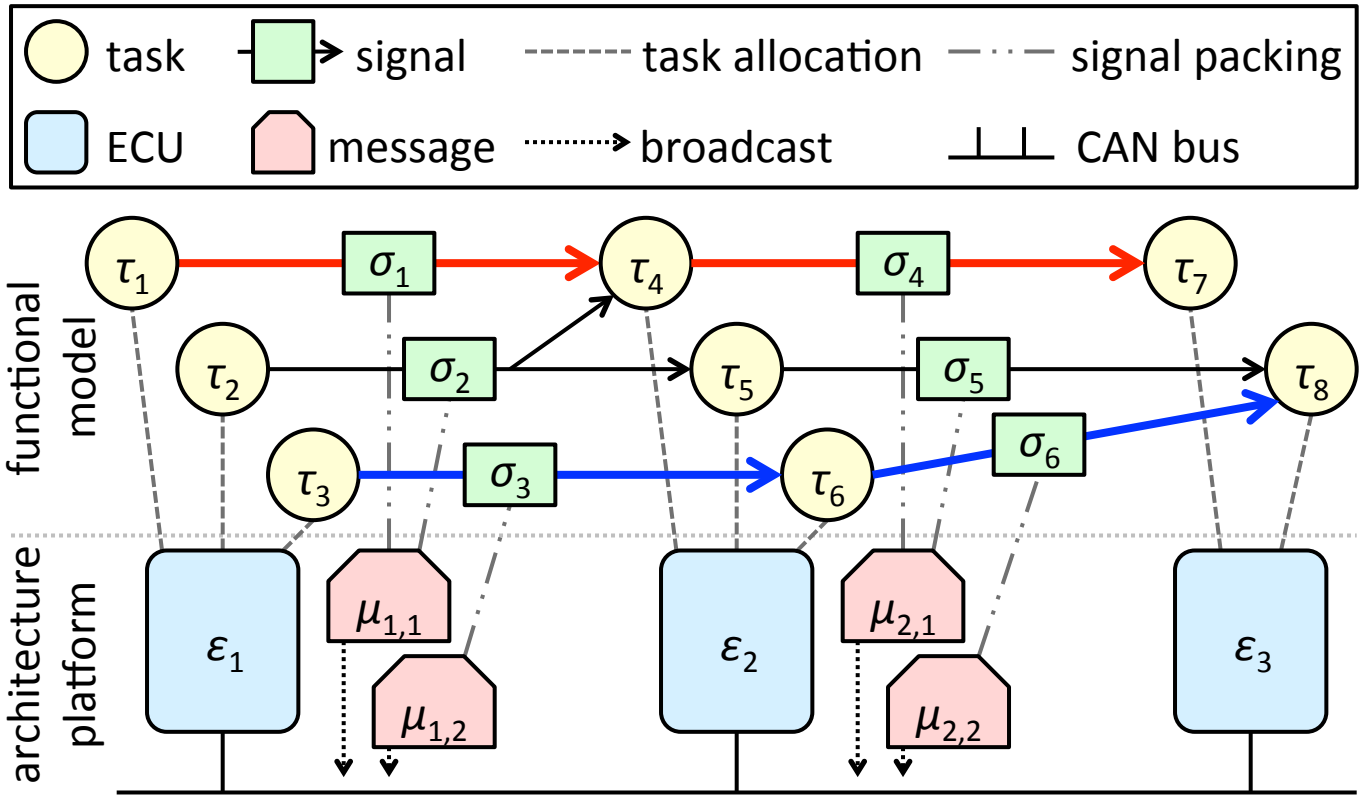
- Example
  - ε_5 is extremely critical, so no other receiver is assigned in its receiving group
    - No direct attack toward it
  - ε_2, ε_3, and ε_4 are not critical, so they are assigned in the same receiving group
    - Possible direct attacks between them



- Two major factors that affect direct and indirect attack risks are quantitatively measured and given as parameters
  - For each signal, the total risk of direct attacks should be bounded
  - For each receiver, the corresponding MAC length (the MAC length of its receiving group) should be long enough

Safety Constraints

- The worst-case end-to-end latency of a path should be bounded

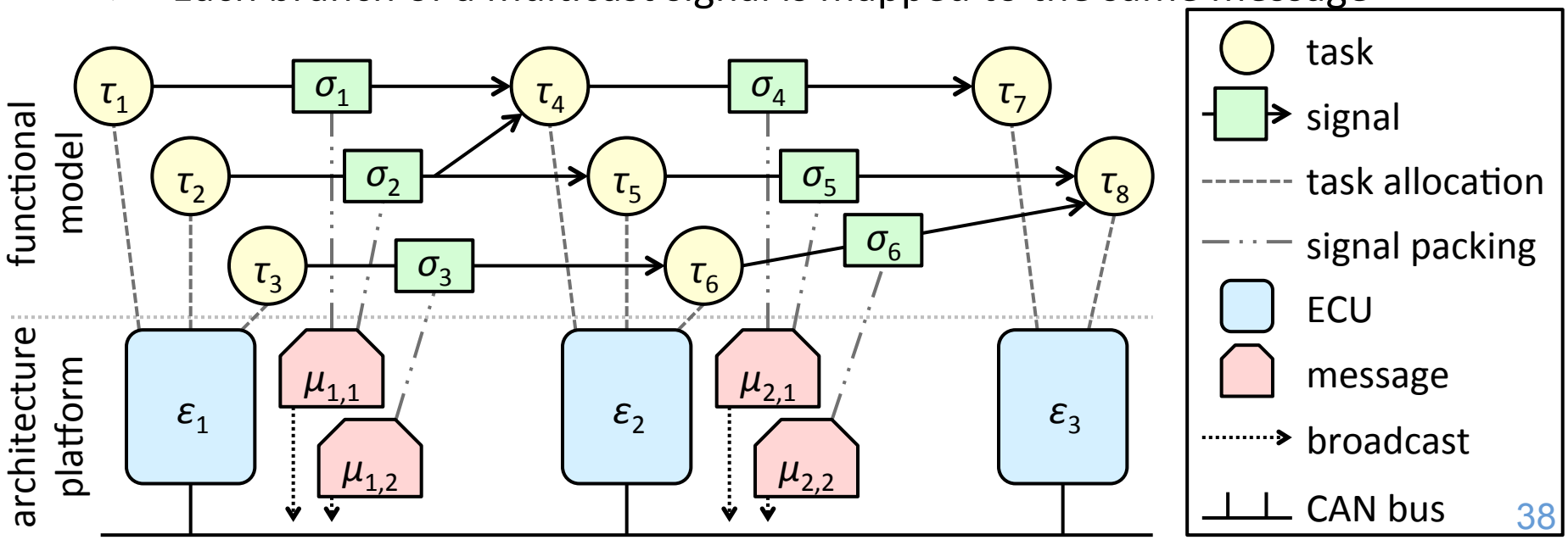


Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

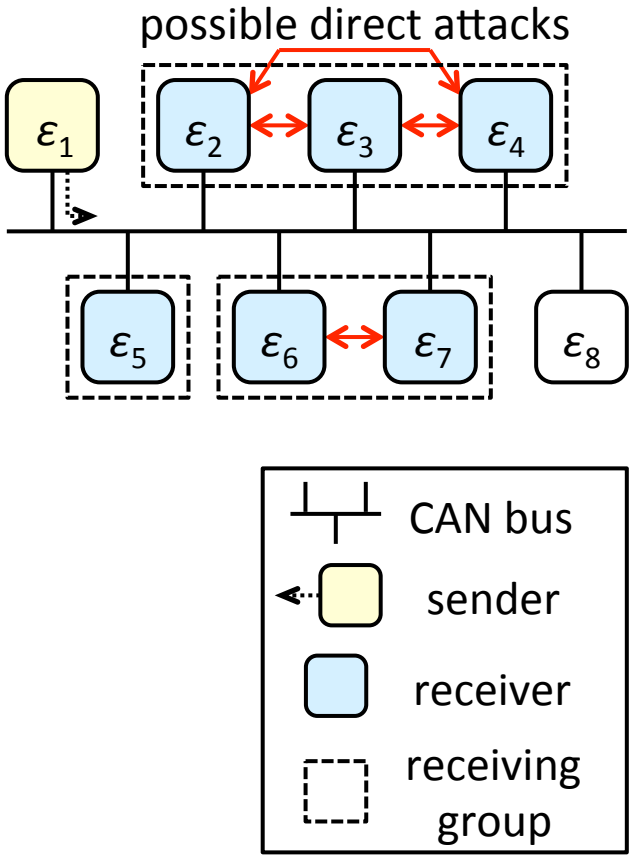
Constraints: Allocation & Packing

- Each task is allocated to exactly one ECU
- Each signal is packed into exactly one message
  - The source task of a signal is allocated to the source ECU of its packed message
  - The period of a signal is equal to the period of its packed message
  - Each branch of a multicast signal is mapped to the same message



Constraints: Security

- For each signal σ, the total risk of direct attacks should be bounded
  - $R_{\sigma,2,3} + R_{\sigma,2,4} + R_{\sigma,3,4} + R_{\sigma,6,7} \leq R_{\sigma}$
- For each receiver, the corresponding MAC length should be long enough
  - $L_2 \leq L_{MAC1}$; $L_3 \leq L_{MAC1}$; $L_4 \leq L_{MAC1}$
  - $L_5 \leq L_{MAC2}$
  - $L_6 \leq L_{MAC3}$; $L_7 \leq L_{MAC3}$
- The values of all R's and L's depend on
  - How critical a message is falsely accepted
  - How likely an existing ECU is compromised



Constraints: End-to-End Latency

- Task response time: $r_i = C_i + \Sigma_{j \in HP(i)} \lceil (r_i / T_j) \rceil C_j$
  - C_i: the computation time of task i
  - T_i: the period of task i
  - HP(i): the set of tasks with higher priority than task i
- Message response time: $r_i = B_i + C_i + \Sigma_{j \in HP(i)} \lceil (r_i - C_i) / T_j \rceil C_j$
  - B_i: the blocking time of message i
  - C_i: the computation time of message i
  - T_i: the period of message i
  - HP(i): the set of tasks with higher priority than message i
- Signal response time = that of its packed message
- Path end-to-end latency: $r_{\tau_1} + (T_{\sigma_1} + r_{\sigma_1}) + (T_{\tau_2} + r_{\tau_2})$

Linearization

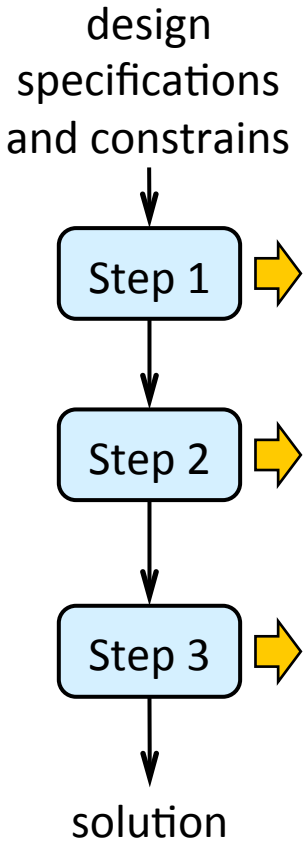
- Inequality of three binary variables: $\alpha + \beta + \gamma \neq 2$
  - $\alpha + \beta + \gamma \neq 2 \leftrightarrow \alpha + \beta - \gamma \leq 1; \alpha - \beta + \gamma \leq 1; -\alpha + \beta + \gamma \leq 1$
- Ceiling function: $\lceil f \rceil$
  - Replace $\lceil f \rceil$ by an integer β
  - $\lceil f \rceil = \beta \leftrightarrow 0 \leq \beta - f \leq 1$
- Multiplication of two binary variables: $\alpha \cdot \beta$
  - Replace $\alpha \cdot \beta$ by a binary variable γ
  - $\alpha \cdot \beta = \gamma \leftrightarrow \alpha + \beta - 1 \leq \gamma; \gamma \leq \alpha; \gamma \leq \beta$
- Multiplication of a binary variable α and a real variable x: $\alpha \cdot x$
  - Replace $\alpha \cdot x$ by a real variable y
  - $\alpha \cdot x = y \leftrightarrow 0 \leq y \leq x; x - M(1 - \alpha) \leq y \leq M\alpha$
    - M: a large constant

Objective Function

- Minimize the summation of the end-to-end latencies of selected paths
- Alternative: minimize the total security risk

MILP-Based Algorithm

- A three-step algorithm



Given by Previous Steps	Defined by Assumptions	Decision Variables
	signal packing message priority group assignment	task allocation task priority
task allocation task priority	group assignment	signal packing message priority
task allocation task priority signal packing message priority		group assignment
solution		

Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

Heuristic Algorithm

- Initialization
  - Calculate weight $w_{i,j}$: an estimation of how much benefit we can gain by mapping the two tasks τ_i and τ_j onto the same ECU
- Task allocation
  - Follow the descending order of $w_{i,j}$
  - Greedily assign two tasks onto the same ECU without violating utilization constraints
- Signal packing
  - Greedily merge two signals without violating payload size constraints
  - Greedily merge MACs without violating security constraints
- Priority assignment
  - Assign priorities of tasks and messages based on the Rate Monotonic policy

Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

Test Case

- Part of Comprehensive Safety Vehicle
  - Support distributed functions with end-to-end computations
  - Collect data from 360-degree sensors to the actuators
  - Consist of throttle, brake and steering subsystems and of advanced HMI (Human-Machine Interface) devices
- Some information
  - 41 tasks
  - 83 signals
  - 9 ECUs
  - One single CAN bus with the speed 500kb/s

Experimental Setting

- Security requirements
  - 50 signals are selected with required MAC lengths ranging
    - From 30 bits to 10 bits for CAN
    - From 128 bits to 64 bits for CAN-FD (Flexible Data-Rate)
  - The maximum allowed security risk of each signal is simplified so that no more than 2 ECUs can be assigned to the same receiving group
- Safety requirements
  - 171 paths are selected with deadlines 300ms or 100ms
- Other information
  - The program was implemented in C/C++ and CPLEX 12.5
  - The experiments were run on a 2.5-GHz processor with 4GB RAM

Comparison with the Greedy Heuristic Algorithm

Protocol	Step X	Results after Step X				
		Objective (ms)	MAX L_{300} (ms)	MAX L_{100} (ms)	Bus Load (kb/s)	Runtime (s)
CAN	1	11070.61	127.92	90.72	76.92	~ 3,600
	2	11069.88	127.82	90.62	45.57	< 600
	3	11069.62	127.79	90.59	31.52	< 10
Heuristic		23114.50	---	---	---	1.4

- The MILP-based algorithm can find a feasible solution and outperform the heuristic algorithm
- Observations at Steps 2 and 3
  - There is little improvement because the message response times are much smaller than the task and message periods
  - However, the bus load is significantly reduced

Experiment on CAN-FD

Protocol	Step X	Results after Step X				
		Objective (ms)	MAX L_{300} (ms)	MAX L_{100} (ms)	Bus Load (kb/s)	Runtime (s)
CAN	1	11070.61	127.92	90.72	76.92	~ 3,600
	2	11069.88	127.82	90.62	45.57	< 600
	3	11069.62	127.79	90.59	31.52	< 10
CAN-FD	1	11075.08	128.56	91.22	211.74	~ 3,600
	2	11073.67	128.39	91.05	176.47	< 600
	3	11071.69	128.14	90.80	98.33	< 10

- Steps 2 and 3 reduce the bus load significantly, showing the effectiveness of signal packing and our flexible key distribution scheme
- The greedy heuristic cannot find a feasible solution in this case (with bus speed at 500kb/s)

Comparison with Non-Integrated Approaches

- Setting 1: at Steps 1 and 2, all messages have at most 32 bits used for data, leaving 32 bits for MAC bits
  - Pair-wise key distribution: no feasible solution
    - Reason: some messages require more than 32 MAC bits
  - One-key-for-all key distribution: no feasible solution
    - Reason: some messages have too high security risks
- Setting 2: at Steps 1 and 2, all messages have at most 64 bits used for data, probably leaving no bit for MAC bits
  - Pair-wise key distribution, one-key-for-all key distribution, and our flexible key distribution scheme: no feasible solution
    - Reason: some messages use almost all 64 bits
- It is necessary to consider security together with other metrics during mapping
  - It may be difficult or even impossible to add security later

Outline

- Introduction
- Security Mechanism for Controller Area Network
  - Background
  - System and Attacker Models
  - Our Security Mechanism
  - Performance Analysis
- Security-Aware Mapping for Controller Area Network
  - System Model and Constraints
  - MILP-Based Mapping Algorithm
  - Heuristic Algorithm
  - Experimental Results
- Conclusion and Future Work

Conclusion

- Describe a security mechanism that can be used to retro-fit the CAN protocol
  - Protect against masquerade and replay attacks
  - Have a low communication overhead
  - Do not need to maintain a global time
- Address both the security and the safety in the design space exploration of automotive systems
  - An MILP formulation that explores
    - Task allocation
    - Signal packing
    - MAC sharing,
    - Priority assignment

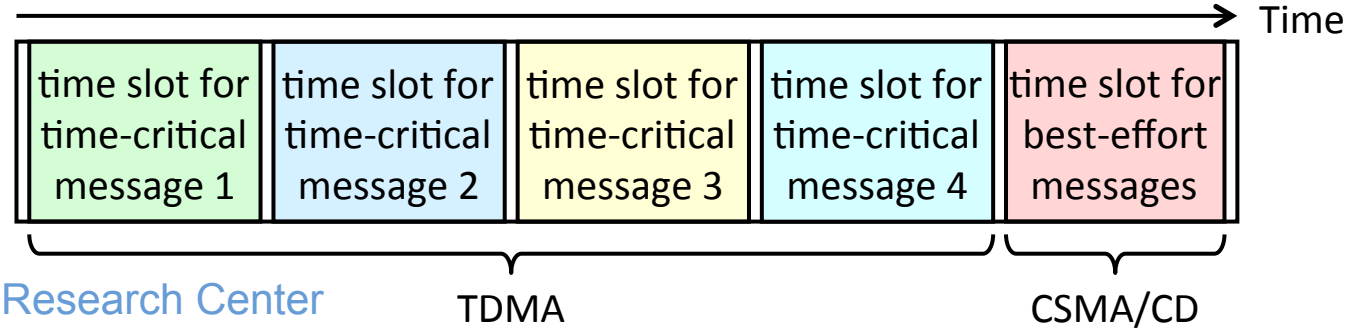
    and meets both security and safety constraints

Future Work

- More general and heterogeneous distributed systems
  - There are many existing security mechanisms (RSA, digital signature, TESLA, etc.) we can use
  - We will focus more on the mapping (synthesis) part of this problem
- What properties we should capture from a functional model or an architecture platform?
  - Functional model
    - Requirement(s) of data integrity and/or confidentiality
    - Constraint(s) on performance and/or security
  - Architecture platform
    - Computational resource (speed, power, etc.)
    - Communication resource (bandwidth, global time, etc.)

Time-Trigger Ethernet

- Ethernet (and its extensions) will be used in the next generation of vehicles (also in many distributed systems)
- Why consider Time-Trigger Ethernet (TTEthernet) first?
  - A good design space exploration example
    - TDMA vs. CSMA/CD
    - Time-critical vs. best-effort
    - Global time vs. no global time
  - A general solution
    - No security constraint: basic mapping for TTEthernet
    - No time-triggered portion: security-aware mapping for basic Ethernet



Q&A

Thank You!