Research Article

Security-Aware Scheduling for FlexRay-Based Real-Time Automotive Systems

R. Zhao ¹, G. H. Qin ², H. P. Chen ², J. Qin ² and J. Yan²

¹Automotive Engineering, Jilin University, 130000, China

²Computer Science and Technology, Jilin University, 130000, China

Correspondence should be addressed to H. P. Chen; chenhp@jlu.edu.cn

Received 15 December 2018; Revised 23 April 2019; Accepted 2 May 2019; Published 13 June 2019

Academic Editor: Hong-Yu Wu

Copyright © 2019 R. Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

FlexRay is a hybrid communication protocol tailored to the requirements of safety-critical distributed real-time automotive systems, providing support for the transmission of time-critical periodic frames in a static segment and event-triggered frames in a dynamic segment. With the development of intelligence and networking of vehicles, such systems are becoming increasingly connected to external environments; thus, security has become a pressing issue in system design. However, FlexRay-based architecture does not have direct support for secure communication. When deploying the security mechanisms on these architectures, a primary challenge is to guarantee the schedulability of systems, given the tight resource constraints and strict timing constraints. In this paper, we apply an authentication mechanism based on the delayed exposure of one-way key chains to protect the authenticity of messages on FlexRay and make a slight modification to reduce the authentication delay. On that basis, we propose a mixed integer linear programming formulation for solving the scheduling problem of FlexRay-based real-time automotive systems subject to both authentication mechanism constraints and other traditional design constraints. Experimental results demonstrate the effectiveness and efficiency of the proposed method in system design and indicate the necessity of collaborative design between security and other functionalities.

1. Introduction

In recent years the number of electronics-based and software-based functionalities is constantly increasing in modern vehicles. X-by-wire systems are replacing the typical hydraulic and mechanic systems to improve the control efficient and handling stability of vehicles. Active safety systems such as lane departure warning, adaptive cruise control, collision prevention assist, electronic stability program, or blind spot detection make vehicles safer and more comfortable. These new function applications are often implemented on a distributed architecture with high demands on the in-vehicle network in terms of bandwidth and deterministic behavior [1]. To overcome the deficiencies of the predominant event-triggered control area network (CAN) [2] for in-vehicle communication, the time-triggered (TT) architecture based on FlexRay [3] is becoming increasingly more common for implementation of such safety-critical automotive systems [4–6].

FlexRay is a hybrid in-vehicle communication standard with fully deterministic and high bandwidth (10 Mbps). In FlexRay, time is divided into communication cycles, and each cycle contains both a Static Segment (ST) with time-triggered (TDMA) communication and a Dynamic (DYN) Segment with event-triggered communication. The ST segment enables the transmission of time-critical frames in a predefined position of the cycle. The DYN segment is used to transfer less critical frames such as diagnosis and maintenance data. FlexRay includes a dual channel bus specification and strong error-detection mechanisms for increased reliability. These capabilities make it a powerful network solution in automotive domain.

With the development of intelligence and networking of vehicles, automotive systems are becoming increasingly connected to the physical environment, mobile devices, surrounding infrastructures, and other systems. A wide range of communication interfaces increases the risks of systems being compromised by attackers. In [7–11], the researchers

demonstrated that modern automotive systems are vulnerable to attacks through various interfaces such as OBD-II, Bluetooth, Wi-Fi, DSRC, GPS, and 3G/4G. Once one Electronic Control Unit (ECU) of the system is compromised by malicious attackers through any interface, they can gain access to other safety-critical ECUs via internal network and inject malicious messages, thereby inducing system failures. It is therefore important to guarantee the authenticity of the communication data of automotive systems. However, despite the various advantages of FlexRay-based architecture, it does not directly provide multicast source authentication to protect data authenticity.

Integrating authentication mechanisms into FlexRay-based real-time automotive systems is not an easy task. Such systems usually have tight resource constraints, such as limited computing and bandwidth resources, strict timing constraints, and high-performance requirements with respect to latency and extensibility. This makes it virtually impossible to add authentication mechanism after the scheduling design stage without violating the system constraints or degrading the system performances. Therefore it is essential to address security together with other constraints and objectives from the beginning of scheduling design process. *This involves two issues: The first is to deploy an appropriate multicast authentication mechanism considering the resource constraints and timing constraints of the systems; the second is to develop an optimal security-aware design of system scheduling subject to both authentication mechanism constraints and all other traditional design constraints, which are often in conflict and require careful trade-offs.*

A number of studies on multicast authentication mechanisms have been reported in literature [12–22]. Digital signatures based on public key cryptography provide an elegant method for signing multicast data, but they are not the solution in our context because of the high computational overhead. Although the computational overhead could be alleviated by dedicated circuits, such as FPGAs or ASICs, this will add component costs, an issue that is typically avoided by manufacturers. Schemes using one-time signatures [12, 14, 15] are much more computationally efficient than traditional public key signatures. However, one-time signatures can incur kilobytes of authentication data per message, which makes them impractical for automotive systems with the requirement of real-time and efficient data transmission.

In contrast, symmetric cryptography is more suitable for the constrained environments. Simply applying the point-to-point authentication mechanisms, such as appending a message authentication code (MAC) to each message or every other message computed by a secret key shared across all nodes, cannot provide adequate multicast authentication. The problem is that any node which holds the secret key can forge message and impersonate the sender. Several schemes [16, 17] have been based on the concept that a sender shares a unique symmetric key with each receiver to prevent this attack. For each message, the sender generates and sends one MAC for each distinct receiver. However, even for a small number of receivers, the computational and bandwidth overhead makes this approach infeasible for automotive systems with tight resource constraints and strict timing constraints. TESLA

[18] provides multicast authentication based on delayed disclosure of keys by using only symmetric cryptography. The core idea of TESLA is that the sender appends to each message a MAC computed by using a key known only to itself and discloses this key after a short time interval. Each receiver buffers the received frame and then verifies the authenticity after it receives the correct key. TESLA was extended and applied in resource constrained wireless sensor networks by several authors [13, 19–22], because it provides an appropriate trade-off between security level and resource overheads. *In this work, we choose the TESLA mechanism to perform multicast authentication on FlexRay bus and make a modification to the original TESLA so that it is more appropriate for our application setting.*

Moreover, the design synthesis of the FlexRay static communication schedule has also been well studied in the past [1, 23–29]. Schmidt *et al.* [23] addressed the scheduling problem of periodic signals in the FlexRay ST segment. In their study, signals are first packed into message frames while maximizing the utilization; the obtained messages are then scheduled in the FlexRay ST segment while using a minimum number of slots. Kang *et al.* [24] proposed a frame-packing algorithm that allows the packing of signals with different periods into a frame. Schmidt *et al.* [25] proposed an ILP-based method to assign ST slots for the signals. Grenier *et al.* [26] proposed a heuristic to construct the schedule on the ST segment of FlexRay systems. Some studies have further considered using the slot multiplexing mechanism on signal scheduling of FlexRay, which is specifically dedicated for scalable efficient bandwidth exploitation. Lukasiewicz *et al.* [27] applied single slot multiplexing to the scheduling of the ST segment with the objective of increasing the utilization of the FlexRay bus. Zhao *et al.* [28] proposed a fast heuristic as well as an efficient integer linear programming (MILP) approach to optimize the scheduling of the FlexRay ST segment. They [29] also proposed a rectangle bin packing optimization approach to schedule communication signals with timing constraints into the FlexRay static segment at minimum bandwidth cost. Sagstetter *et al.* [1] proposed single-stage and multistage ILP approaches to schedule the FlexRay ST segment. All of the above works only focus on the communication schedule and did not attempt to schedule at the system-level. The isolated signal scheduling may seriously limit the feasibility and performance of automotive applications, which consist of both signals and tasks.

A handful of studies have studied the scheduling on both signals and tasks for FlexRay-based real-time automotive systems. Zeng *et al.* [30] applied MILP optimization framework to solve the scheduling problem for time-triggered systems communicating over the FlexRay ST segment. Hu *et al.* [31] proposed a holistic scheduling algorithm for FlexRay-based time-triggered systems. Lukasiewicz *et al.* [32] presented a modular framework based on ILP to perform schedule synthesis for FlexRay-based time-triggered automotive systems. Roy *et al.* [33] proposed a multiobjective cooptimization approach that synthesizes both the controllers and the task and communication schedules of FlexRay ST segment. However, none of the above-mentioned works considered the

pressing concerns of security for automotive systems, i.e., the interference of security operations on system applications.

Furthermore, several approaches have been proposed for the codesign of security and other traditional functions of embedded systems. Jiang *et al.* [34] presented a constraint logic programming-based approach to efficiently implement cryptographic algorithms on the distributed real-time embedded systems. The objective of their approach is to determine the minimal hardware overhead and the mapping for encryption and decryption tasks of the system such that the confidentiality requirements for messages were fulfilled and the time constraints were satisfied. For authenticity, Wasicek *et al.* [35] presented an implementation of the TESLA mechanism in a Time-Triggered system but did not consider system-level design issues. Lin *et al.* [36] applied MACs generated by a group session key to protect against masquerade and replay attacks on CANs, and they proposed an integrated MILP formulation to explore the mapping from the functional model to the CAN-based architecture platform. They [37] later proposed optimization algorithms for mapping from a functional model to a TDMA-based architecture platform, while both works focused on network scheduling issues but did not consider the CPU overheads of authentication operations as well as their interference on application tasks scheduling. Gu *et al.* [38] proposed a security-aware optimization approach of FlexRay-based distributed systems, interested in minimizing the total used number of hardware units subject to schedulability. However their approach ignored the key authentication operations of the applied TESLA mechanism and therefore cannot provide adequate security. *In this work, we analyze and model all the additional operations induced by TESLA authentication mechanism and then provide an MILP formulation for solving the scheduling optimization problem of FlexRay-based real-time automotive systems while meeting the requirements of both information security and functional safety.*

To this end, our major contributions are as follows.

- (i) First, we apply the TESLA authentication mechanism to protect against forgery and replay attacks on FlexRay bus. It provides an appropriate trade-off between security level and resource overhead, compared with other multicast authentication approaches. Moreover, we make a slight modification to the original TESLA in order to improve on the authentication delay.
- (ii) Furthermore, we propose an MILP formulation that efficiently solves the optimal scheduling problem of FlexRay-based real-time automotive systems with authentication mechanism constraints. The proposed approach considers end-to-end delays and the precedence constraints induced by information passing between all automotive control and authentication mechanism-related tasks and signals. The scheduling design includes (a) the packing of automotive control and authentication mechanism-related signals to FlexRay frames, (b) the scheduling of frames on FlexRay bus (i.e., slot assignment), and (c) the

scheduling of automotive control and authentication mechanism-related tasks on respective ECU. The optimization objective is to maximize the laxity (difference between deadlines and response times) on time-sensitive function paths, therefore improving timing performance, or to minimize the number of used slots, therefore improving extensibility.

The remainder of the paper is organized as follows. Section 2 introduces the system model and application model. Section 3 presents the security mechanism and security model. Section 4 describes the security-aware optimization scheduling problem whose solution is tackled using MILP-based method. Section 5 shows experimental results, and Section 6 concludes the paper.

2. System Model and Application Model

2.1. System Model. The target platform is a typical distributed time-triggered automotive system: a cluster of ECUs that are connected via the FlexRay bus, as presented in Figure 1(a). The ECU $e_g \in E$ runs the TT real-time operating system in which tasks are executed according to a schedule table that defines their start times. In addition, the network uses the ST segment arbitration model provided by the FlexRay protocol. Tasks are triggered based on a local clock in each ECU. The synchronization of local clocks throughout the system is provided by the FlexRay communication protocol.

In FlexRay, the communication takes place in a recurring cycle, as depicted in Figure 2. Each communication cycle involves the ST segment, DYN segment, symbol window, and network idle time. The static segment applies the TDMA scheme and is composed of several equally sized slots. During any slot in any communication cycle, the sole ECU T_m^f which holds the frame $f_m \in \mathcal{F}$ with the frame identifier equal to the current value of the slot counter, cycle number equal to the current value of the cycle counter is allowed to send counter and the corresponding frame on the bus. Clock synchronization, embedded in the standard, ensures deterministic communication at no additional cost.

To support the efficient use of the ST, FlexRay provides a proprietary slot multiplexing mechanism that allows the alteration of frame contents being sent to this slot from cycle to cycle. The repetition RE_m^f of a frame f_m with the slot multiplexing mechanism should be expressed as a multiple of the communication cycle, with the constraints that $RE_m^f \in \{1, 2, 4, 8, 16, 32, 64\}$ for version 2.1 of FlexRay and $RE_m^f \in \{1, 2, 4, 5, 8, 10, 16, 20, 32, 40, 50, 64\}$ for version 3.0 of FlexRay. The lengths of ST slot L_{slot} , ST segment L_{ST} , and the communication cycle L_{cycle} are usually designed by the automotive Original Equipment Manufacturer (OEM), taking into account the composition of subsystems, reuse of components, and future design extensibility. In this work, because of standardization, we assume these design parameters as given inputs to our schedule synthesis.

2.2. Application Model. We model an application $\lambda_r \in \Lambda_r^{app}$ to be processed in the system as a directed, acyclic graph $\mathcal{G}_r^{app}$,

FIGURE 1: System model example.

FIGURE 2: Basic structure of FlexRay communication cycles with a detailed illustration of the static segment.

where the vertices represent the tasks and the edges represent the signals communicated between tasks.

A task $\tau_i \in \Gamma^{app}$ is characterized by the tuple $(E_i^\tau, L_i^\tau, P_i^\tau, D_i^\tau)$, where E_i^τ denotes the ECU resource it needs to execute, L_i^τ is its execution time, P_i^τ is its period, and D_i^τ is its deadline. An edge linking two tasks τ_i and $\tau_{i'}$ denotes a signal σ_j , produced by τ_i that is available to $\tau_{i'}$. Each task reads its input at its start time and writes its results at the end of execution. A signal $\sigma_j \in \mathcal{S}^{app}$ is characterized by the tuple $(T_j^\sigma, R_j^\sigma, L_j^\sigma, P_j^\sigma, D_j^\sigma)$, where T_j^σ and R_j^σ denote the ECU resources required for sending and receiving it, L_j^σ is its bit width, P_j^σ is its period, and D_j^σ is its deadline.

For example, in the application λ_1 of Figure 1(b), task τ_3 generates a multicast signal, and tasks τ_5 and τ_7 are the receivers of this signal. In our model, each branch of a multicast signal is represented as a separate signal because the branches have different receivers. Specifically, we define a set of signals $\mathcal{S}_{g,h}$ containing all of the branches of the h -th multicast signal of each ECU e_g , such as signals σ_9 and σ_{10} that are assumed to be the two branches of the first multicast signal of ECU e_2 ; thus we have $\sigma_9, \sigma_{10} \in \mathcal{S}_{2,1}$.

A function path $\rho_{\epsilon} \in \mathcal{F}^{app}$ from τ_i to $\tau_{i'}$ is a sequence $[\tau_i, \dots, \tau_{i'}]$ of tasks such that there is a link between any two consecutive tasks. For instance, in application λ_1 , there is a

FIGURE 3: TESLA protocol example.

function path between tasks τ_1 and τ_5 . The latency of function path is defined as the time interval between the start of an instance of τ_i and the completion of the instance of τ_i that produces a result that is dependent on the output of τ_i . The deadline D_ϵ^p of function path is set by system designers as an application requirement.

3. Security Mechanism and Security Model

3.1. Security Mechanism

3.1.1. Attack Model. We assume that an attacker can gain access to such system through a gateway linked with an external network, physical access to FlexRay channel, malicious insider code, or tampering with ECUs. We consider an active attacker model where an attacker can masquerade as other ECUs to inject forged messages and can also replay messages [17]. We assume that the attacker knows about the network schedule (e.g., by applying technical skill to reverse engineer the appropriate systems and protocols or purchasing such information from a third-party) and consequently has the ability to insert well-formed frames into FlexRay bus.

3.1.2. Overview of the TESLA Mechanism. We apply TESLA authentication mechanism to protect the authenticity of messages on FlexRay bus. The main idea behind TESLA is to use time with the one-way key chain for asymmetry to enjoy the benefit of computational efficiency while having the asymmetric security property. TESLA requires loose time synchronization between the sender and the receivers. In this work, TESLA is initiated after the clock synchronization process of FlexRay is completed, so that it uses the synchronized time provided by FlexRay.

In TESLA, time is divided into several intervals with uniform duration P_{int} . Before protocol execution, the sender generates a one-way key chain of self-authenticating values using a hash function H as $K_0, K_1, \dots, K_{n_{int}}$, where $K_\phi = H(K_{\phi+1})$, and assigns the keys sequentially to the time intervals, as depicted in Figure 3. The chain is used in reverse order starting with K_1 . To bootstrap TESLA, the sender uses asymmetric cryptography to distribute the initial key to every receiver in the network authoritatively.

When a sender sends a message in ϕ -th time interval, it appends a MAC to the message computed using a hash function and the key K_ϕ corresponding to the current time interval. The key remains secret for d intervals, so, along with the message, the sender also sends the key $K_{\phi-d}$ that it can disclose.

When a receiver receives a message in ϕ -th time interval, it first checks the MAC is still secret by determining that the sender could not have yet reached the time interval for disclosing it. If the MAC key is still secret, then the receiver buffers the packet and verifies the authenticity after it gets the correct key K_ϕ . The legitimacy of key K_ϕ can be determined by verifying previously released keys $K_{\phi-1}$ that $K_{\phi-1} = H(K_\phi)$.

3.1.3. Modification to the TESLA Mechanism. There are two following major barriers when using TESLA in FlexRay-based real-time automotive systems:

(i) To ensure the key to be disclosed in an interval can arrive at its receivers on time, TESLA protocol specifies that the key must be appended to each message frame in that interval. This thus causes the waste of bandwidth and computing resources, as well as the increase of the authentication delay which is the most critical part in real-time automotive systems in general.

(ii) In addition, the full-size MAC in TESLA makes FlexRay network communication inefficient; i.e., TESLA has an overhead of 32 bytes per packet.

In this section, we make some modifications to the original TESLA, so that it provides better trade-off between security level and resource overhead.

Modification 1: We specify that each key K_ϕ is released only once in its next interval and is transmitted through the ST segment to ensure its time determinism.

Modification 2: We truncate the MAC size within the application frame, as long as the desired level of security is ensured, e.g., ISO 26262 standard. The standard constrains the permissible probability ϕ of automotive system failure in a time unit P_{unit} . Following this, we assume that the maximum probability of a system failure due to any malicious data in a time unit, P_{unit} , is constrained by ϕ . We define $1 - \phi$ as the security goal of each signal, which represents the quantified performance level with respect to forgery attacks. Without secret key, the generated MAC can be assumed to be random [17]. The probability of successfully forging a valid MAC with the size of L_{MAC} bits is $2^{-L_{MAC}}$. Assuming the period of the signal with the maximum security goal is P_{max}^σ , the MAC size L_{MAC} is truncated to the minimum value satisfying the following constraint:

$$(1 - 2^{-L_{MAC}})^{P_{unit}/P_{max}^\sigma} > 1 - \phi \quad (1)$$

This constraint ensures that the global security probability of all instances of the signal during the time unit P_{unit} is greater than its security goal $1 - \phi$.

FIGURE 4: Security model example.

3.2. Security Model. After applying the modified TESLA authentication mechanism to the system, extra information, i.e., keys, needs to be sent and extra operations (including MAC generation, MAC verification, key generation, and key verification tasks) need to be executed.

The MAC generation and verification tasks of each message frame f_m are executed on its sender and receivers. Moreover, the authentication-related operations of each key are considered as a time-triggered application. Similar to the automotive control applications, we model a key authentication application $\lambda_r \in \Lambda^{sec}$ as a directed, acyclic graph $\mathcal{G}_r^{sec}$, which includes the key release task τ_g^{rel} generated by the sender e_g , key verification tasks $\tau_{g,g'}^{ver}$ generated by the receivers $e_{g'}$ and key signals $\sigma_{g,g'}^{key}$ produced by τ_g^{rel} that is available to $\tau_{g,g'}^{ver}$. Figure 4 illustrates the additional operations after applying the authentication mechanism to the example system in Figure 1.

For simplicity, the key authentication-related tasks or signals are also identified and denoted by a single index, as in τ_i or σ_j . A key verification task $\tau_i \in \Gamma^{sec}$ is characterized by the tuple $(E_i^\tau, V_i^\tau, L_i^\tau, P_i^\tau)$, where E_i^τ and V_i^τ are the ECUs it needs to execute and verify, L_i^τ is its time cost indicating the execution time of the one-way hash function H on the ECU E_i^τ which can be simply measured, and P_i^τ is its period (i.e., the interval duration P_{int} of key release). The choice of interval duration is dictated by the special timing requirements of the automotive systems, which will be described in the next subsection. A key signal $\sigma_j \in \mathcal{S}^{sec}$ is characterized by the tuple $(T_j^\sigma, R_j^\sigma, L_j^\sigma, P_j^\sigma)$, where T_j^σ and R_j^σ are the ECU resources required for sending and receiving it, L_j^σ is its bit width, and P_j^σ is its period. Specially, the design of system scheduling does not need to consider the key release tasks since the keys are generated during the initialization stage thus taking no time.

3.3. Choice of Interval Duration of Key Release. Following the authentication mechanism, the smaller the interval duration is, the more frequently the key authentication applications execute and thus the more processing and communication resources are consumed. But the larger the interval duration is, the longer the response times of the signals take, and thus the greater the likelihood that the signals and function paths will miss their deadlines. Therefore, to efficiently apply the authentication mechanism to the automotive systems, we choose the largest interval duration under the premise of satisfying the timing constraints of the systems.

For a function path $\rho_\epsilon \in \mathcal{F}^{app}$, we let L_ϵ^p denote the number of tasks that need to receive signals arriving from network, also referred to as the signal level. We consider that the interval duration P_{int} is the maximum value satisfying the following constraints:

$$\forall \rho_\epsilon \in \mathcal{F}^{app}, \quad (2a)$$

$$P_{int} \cdot L_\epsilon^p \leq D_\epsilon^p$$

$$P_{int} \leq \min_{\sigma_j \in \mathcal{S}^{app}} D_j^\sigma \quad (2b)$$

$$P_{int} \bmod \gcd_{\sigma_j \in \mathcal{S}^{app}} P_j^\sigma = 0, \quad (2c)$$

$$\text{or } P_{int} = n \cdot \gcd_{\sigma_j \in \mathcal{S}^{app}} P_j^\sigma, \quad n \in \mathbb{Z}^*$$

Relations (2a) and (2b) provide the time limits; i.e., the product of interval duration P_{int} and signal level L_ϵ^p cannot exceed the deadline D_ϵ^p for a function path $\rho_\epsilon \in \mathcal{F}^{app}$. That is because the task that needs receive signals can complete authentication only in the next interval after the signals are transmitted, and thus normally, the whole process of a path ρ_ϵ must have a duration of L_ϵ^p intervals. In addition, the interval duration P_{int} cannot exceed the minimum value of

the deadlines D_j^σ of signals $\sigma_j \in \mathcal{S}^{app}$, to ensure that each signal can be authenticated by the receiver before its deadline. Relation (2c) is bound to the alignment of the key signals and automotive control signals schedules; i.e., the interval duration P_{int} should be an integer multiple or factor of the greatest common divisor (gcd) of the periods of all the signals.

4. Security-Aware Scheduling

4.1. Problem Statement. The security-aware scheduling problem we are addressing in this paper can be formulated as follows. Given the system model, application model, and the security model generated by the authentication mechanism, we decide on the (a) packing of automotive control and authentication mechanism-related signals to messages, (b) scheduling of messages on FlexRay ST segment, and (c) scheduling of automotive control and authentication mechanism-related tasks on respective ECU, such that

- (i) the deadline constraints and the precedence constraints caused by information passing between all tasks and signals are satisfied
- (ii) the constraints related to FlexRay Protocol are satisfied
- (iii) the objective function with respect to timing performance or extensibility is optimized

4.2. Motivational Examples. Let us illustrate the integrated scheduling problem using the setup from Figure 1, where two automotive control applications are executed on the system consisting of four ECUs. The corresponding security applications are depicted in Figure 4. For simplicity, in this example we assume that the execution times of the hash function on all ECUs are 0.2 ms. The lengths of the communication cycle, ST segment, ST slot, and frame payload are 5 ms, 3.264 ms, 0.032 ms, and 16 bytes, respectively. There are 120 slots in the FlexRay ST segment. We assume that there are two time-sensitive paths, one from τ_1 to τ_5 with a deadline of 7.5 ms and the other from τ_2 to τ_8 with a deadline of 5 ms. According to (2a)-(2c), the interval duration P_{int} of key release is 2.5 ms.

A straightforward solution to the security-aware scheduling problem is to (a) pack the signals generated by the same task into a frame and (b) schedule the key authentication-related tasks and frames first and then other tasks and frames using As-Soon-As-Possible (ASAP) scheduling (that is because an automotive control-related frame can be verified by its receiver only after the verification task for its MAC key is completed). For the example in Figure 1, this solution is depicted by the Gantt chart in Figure 5(a). For simplicity, key signals $\sigma_{1,2}^{key}, \sigma_{1,3}^{key}, \sigma_{1,4}^{key}, \sigma_{2,3}^{key}, \sigma_{2,4}^{key}$ are denoted by $\sigma_{11}-\sigma_{15}$, and key verification tasks $\tau_{1,2}^{ver}, \tau_{1,3}^{ver}, \tau_{1,4}^{ver}, \tau_{2,3}^{ver}, \tau_{2,4}^{ver}$ are denoted by $\tau_9-\tau_{13}$. In this solution, the automotive control signals σ_1, σ_2 are packed into frame $f_1, \sigma_9, \sigma_{10}$ into f_2 and $\sigma_3-\sigma_8$ into f_3 , and the key signals $\sigma_{11}-\sigma_{13}$ are packed into $f_4, \sigma_{14}, \sigma_{15}$ into f_5 . In this case, the value of the laxity of time-sensitive function paths is 0.91 ms (the path from τ_1 to τ_5 misses its deadline) and the numbers of the used slot identifiers and slots are 7 and 12, respectively.

Figure 5(b) illustrates an optimal solution with respect to timing performance. This solution increases the laxity of paths to 3.936 ms and satisfies the deadline constraints of both paths. On the other hand, Figure 5(c) illustrates an optimal solution with respect to extensibility, which reduces the number of the used slot identifiers and slots to 4 and 8, respectively, by packing signals to frames while satisfying the deadline constraints.

4.3. MILP-Based Optimization Scheduling Approach. We use an MILP formulation to find an optimal solution to the security-aware scheduling problem with respect to timing performance- or extensibility-related cost functions. In an MILP framework, the system is represented with constant parameters, decision variables, and constraints based on the parameters and variables. The objective function, defined over the same sets of parameters and variables, characterizes the optimal solution. MILPs can be solved very efficiently by various solvers. In this work, we employ the LINGO solver.

4.3.1. Definitions. The notations of the elements and constant parameters were described in the previous definition of system model and security model. We assume these parameters are given as design inputs. The notations of the decision variables in the MILP formulation are listed in Table 1.

4.3.2. Constraints. In this section, we present the various constraints on frame packing, frame scheduling, task scheduling, data dependency, and end-to-end latency.

(a) Frame Packing

$$\forall g, \sigma_j \in \mathcal{S}_g, \quad (3a)$$

$$\sum_{f_m \in \mathcal{F}_g} x_{j,m} = 1$$

$$\forall g, \sigma_j \in \mathcal{S}_g, \quad (3b)$$

$$\sum_{f_m \notin \mathcal{F}_g} x_{j,m} = 0$$

$$\forall j, m, \quad (3c)$$

$$x_{j,m} \cdot RE_m^f \leq P_j^\sigma$$

$$\forall m, \forall \sigma_j, \sigma_{j'} \in \mathcal{S}_{g,h}, \quad (3d)$$

$$x_{j,m} = x_{j',m}$$

$$\forall m, \forall \sigma_j \in \mathcal{S}_{g,h}, \quad (3e)$$

$$x_{j,m} = \sum_{\sigma_{j'} \in \mathcal{S}_{g,h}} y_{j',m}$$

Equations (3a) and (3b) guarantee that the signal σ_j is packed into exactly one frame with the same type from the same ECU, where $\mathcal{S}_g$ and $\mathcal{F}_g$ denote the sets of the signals and frames from ECU e_g , respectively. Constraint (3c) guarantees that the period of a signal is greater than or equal

FIGURE 5: Motivational examples.

to the repetition of the frame into which the signal is packed. Equation (3d) ensures that each branch of a multicast signal is packed into the same frame. Equation (3e) ensures that exactly one branch of a multicast signal adds its length to the frame.

$$\forall m, j, \quad x_{j,m} \leq \mu_{m,S_j^\sigma}^f \quad (4a)$$

$$\forall m, j, \quad x_{j,m} \leq \mu_{m,R_j^\sigma}^f \quad (4b)$$

$$\forall m, g, \quad \mu_{m,g}^f \leq \sum_{\sigma_j \in \{\sigma_j \in \mathcal{S} \mid R_j^\sigma = e_g \text{ or } S_j^\sigma = e_g\}} x_{j,m} \quad (4c)$$

$$\forall m, \forall \sigma_j \in \mathcal{S}^{app}, \quad x_{j,m} \leq \pi_m^f \quad (4d)$$

$$\forall m, \quad \pi_m^f \leq \sum_{\sigma_j \in \mathcal{S}^{app}} x_{j,m} \quad (4e)$$

TABLE 1: The notations of binary variables and real variables.

Symbol	Variable Type	Implication
$x_{j,m}$	binary variable ^a	signal σ_j is packed into frame f_m
$y_{j,m}$	binary variable	signal σ_j adds its length to frame f_m
$z_{m,k,w}$	binary variable	frame f_m is packed into the k -th slot of the w -th cycle
$v_{g,k,w}$	binary variable	the k -th slot of the w -th cycle is owned by ECU e_g
$v_{g,k}$	binary variable	the k -th slot is owned by ECU e_g
$r_{m,g}^f$	binary variable	ECU e_g is the receiver of frame f_m
$\mu_{m,g}^f$	binary variable	ECU e_g executes authentication-related operations of frame f_m
θ_m^f	binary variable	frame f_m is not empty
π_m^f	binary variable	frame f_m includes automotive control-related signals
$\mathfrak{S}_{i,i'}^{\alpha,\beta}$	binary variable	the α -th instance of task τ_i starts earlier than the β -th instance of task $\tau_{i'}$
$\delta_{g,m,m'}^{\alpha,\beta}$	binary variable	the α -th instance of frame f_m starts earlier than the β -th instance of frame $f_{m'}$ on ECU e_g
$\eta_{g,i,m}^{\alpha,\beta}$	binary variable	the α -th instance of task τ_i starts earlier than the β -th instance of frame f_m on ECU e_g
o_i^τ	real variable	the start time of task τ_i
o_j^σ	real variable	the start time of signal σ_j
c_j^σ	real variable	the finish time of signal σ_j
$o_{m,g}^f$	real variable	the start time of frame f_m on ECU e_g
$o_{m,n}^f$	real variable	the start time of frame f_m on FlexRay bus
l_m^f	real variable	the length of frame f_m
ϕ_m^f	real variable	the time interval in which the MAC key of frame f_m is released

$$\forall m,$$

$$\theta_m^f \leq \sum_{\sigma_j \in \mathcal{S}} x_{j,m} \quad (4f)$$

ensures that the frame length does not exceed the limit L_{frame} allowed by FlexRay.

(b) Frame Scheduling

$$\forall m,$$

$$\theta_m^f \geq \frac{\sum_{\sigma_j \in \mathcal{S}} x_{j,m}}{M} \quad (4g)$$

(i) Slot Assignment

$$\forall m,$$

$$\sum_{w \in \{0, \dots, RE_m^f - 1\}} \sum_{k \in \{1, \dots, n_{slot}\}} z_{m,w,k} \leq 1 + M \cdot (1 - \theta_m^f) \quad (6a)$$

$$\forall m,$$

$$1 - M \cdot (1 - \theta_m^f) \leq \sum_{w \in \{0, \dots, RE_m^f - 1\}} \sum_{k \in \{1, \dots, n_{slot}\}} z_{m,w,k} \quad (6b)$$

$$\forall m,$$

$$\sum_{w \in \{0, \dots, RE_m^f - 1\}} \sum_{k \in \{1, \dots, n_{slot}\}} z_{m,w,k} \leq M \cdot \theta_m^f \quad (6c)$$

$$\forall m, w, \forall k \in \{0, \dots, RE_m^f - 1\},$$

$$n_{rep} \in \left\{ 0, \dots, \frac{H}{(RE_m^f \cdot L_{cycle})} - 1 \right\} \quad (6d)$$

$$z_{m,w,k} = z_{m,w+n_{rep} \cdot RE_m^f, k}$$

Constraints (4a)-(4c) guarantee that an ECU e_g executes authentication-related operations on the frame f_m only if there exists a signal σ_j packed into f_m and e_g is the sender or receiver of the signal. Constraints (4d) and (4e) ensure that a frame belongs to the application type only if there exists an automotive application-related signal packed into it. Constraints (4f)-(4g) ensure that a frame is not empty only if there exists a signal packed into it, where M is a large constant for linearization in this paper.

$$\forall m,$$

$$l_m^f = \sum_{\sigma_j \in \mathcal{S}} y_{j,m} \cdot L_j^\sigma + \pi_m^f \cdot L_{MAC} \quad (5a)$$

$$\forall m,$$

$$l_m^f \leq L_{frame} \quad (5b)$$

Equation (5a) calculates the total length of the frame, including the data payload and the MAC length (that is only contained in the automotive control frame). Constraint (5b)

Constraints (6a)-(6c) ensure that each nonempty frame is scheduled in exactly one slot at a specific base cycle.

Constraint (6d) ensures that each frame f_m multiplexes the assigned slot with repetition of RE_m^f cycles.

(ii) Slot Ownership (FlexRay 2.1)

$$\forall m, w, k, \quad (7a)$$

$$z_{m,w,k} \leq v_{T_m^f,k}$$

$$\forall k, \quad (7b)$$

$$\sum_{e_g \in E} v_{g,k} \leq 1$$

$$\forall g, w, k, \quad (7c)$$

$$v_{g,k} \leq \sum_{f_m \in \mathcal{F}} z_{m,w,k}$$

FlexRay 2.1 has its requirement for the slot ownership. If a slot is owned by one ECU, then the ownership applies to every cycle. Constraint (7a) states that if a frame is mapped to a specific cycle and slot, then its source ECU must own the slot. Constraint (7b) ensures that every slot can be assigned to a most one ECU, and (7c) sets the slot ownership to null.

(iii) Slot Ownership (FlexRay 3.0)

$$\forall m, w, k, \quad (8a)$$

$$z_{m,w,k} \leq v_{T_m^f,w,k}$$

$$\forall w, k, \quad (8b)$$

$$\sum_{e_g \in E} v_{g,w,k} \leq 1$$

$$\forall g, w, k, \quad (8c)$$

$$v_{g,w,k} \leq \sum_{f_m \in \mathcal{F}} z_{m,w,k}$$

In comparison to version 2.1 of the FlexRay protocol, FlexRay 3.0 standard introduces several changes on slot ownership, and it allows a slot to be shared among different ECUs in the form of cycle multiplexing for improving the utilization of the bus. Constraint (8a) ensures that if a frame is mapped to a specific cycle and slot, then the slot is assigned to its source ECU at this cycle. Constraint (8b) ensures that a slot at a specific cycle can be assigned to a most one ECU, and (8c) states that if there are no frames to be assigned to one slot of one cycle, then the ownership of the slot at this cycle is null.

(iv) Timing-Related

$$\forall m, k, \forall w \in \{0, \dots, RE_m^f\}, \quad (9a)$$

$$o_{m,n}^f \leq w \cdot L_{cycle} + (k-1) \cdot L_{slot} + M \cdot (1 - z_{m,w,k})$$

$$\forall m, k, \forall w \in \{0, \dots, RE_m^f\}, \quad (9b)$$

$$w \cdot L_{cycle} + (k-1) \cdot L_{slot} - M \cdot (1 - z_{m,w,k}) \leq o_{m,n}^f$$

$$\forall m, \quad (9c)$$

$$o_{m,n}^f \leq M \cdot \theta_m^f$$

$$\forall m, \quad (9d)$$

$$o_{m,T_m^f}^f + L_{T_m^f}^e \leq o_{m,n}^f + M \cdot (1 - \pi_m^f)$$

$$\forall m, g, \quad (9e)$$

$$o_{m,g}^f \leq o_{m,n}^f + L_{slot} + M \cdot (1 - r_{m,g}^f) + M \cdot (1 - \pi_m^f)$$

$$\forall m, g, \quad (9f)$$

$$o_{m,g}^f \leq M \cdot \mu_{m,g}^f$$

Constraints (9a) and (9b) ensure that if a nonfull frame f_m is packed into the k -th slot of the w -th cycle, its start time on FlexRay bus is equal to the start time of the slot. Constraints (9c) ensure that the start time and finish time of an empty frame on FlexRay bus are equal to zero. Constraint (9d) states that an automotive control frame f_m must be signed by its sender before being sent out, where $L_{T_m^f}^e$ denotes the execution time of the selected MAC computation function on its sender T_m^f . And (9e) ensures that the frame f_m must be verified after the reception of itself. Constraint (9f) ensures that the start time of a frame on its unused ECU (i.e., neither its sender nor its receiver) is equal to zero.

$$\forall j, m, \quad (10a)$$

$$o_j^\sigma - M \cdot (1 - x_{j,m}) \leq o_{m,T_j^\sigma}^f$$

$$\forall j, m, \quad (10b)$$

$$o_{m,T_j^\sigma}^f \leq o_j^\sigma + M \cdot (1 - x_{j,m})$$

$$\forall \sigma_j \in \mathcal{S}^{app}, \forall m, \quad (10c)$$

$$c_j^\sigma - M \cdot (1 - x_{j,m}) \leq o_{m,R_j^\sigma}^f + L_{R_j^\sigma}^e$$

$$\forall \sigma_j \in \mathcal{S}^{app}, \forall m, \quad (10d)$$

$$o_{m,R_j^\sigma}^f + L_{R_j^\sigma}^e \leq c_j^\sigma + M \cdot (1 - x_{j,m})$$

$$\forall \sigma_j \in \mathcal{S}^{sec}, \forall m, \quad (10e)$$

$$c_j^\sigma - M \cdot (1 - x_{j,m}) \leq o_{m,n}^f + L_{slot}$$

$$\forall \sigma_j \in \mathcal{S}^{sec}, \forall m, \quad (10f)$$

$$o_{m,n}^f + L_{slot} \leq c_j^\sigma + M \cdot (1 - x_{j,m})$$

$$\begin{aligned} \forall \sigma_j \in \mathcal{S}^{app}, \\ c_j^\sigma \leq D_j^\sigma \end{aligned} \quad (10g)$$

Constraints (10a) and (10b) guarantee that the start time o_j^σ of a signal σ_j equals the start time $o_{m,T_j^\sigma}^f$ of the frame f_m into which the signal is packed on its sender T_j^σ . Similarly, (10c) and (10d) guarantee that the finish time c_j^σ of an automotive control signal $\sigma_j \in \mathcal{S}^{app}$ equals the finish time $o_{m,R_j^\sigma}^f + L_{R_j^\sigma}^e$ of the frame f_m into which the signal is packed on its receiver. Given that the key signal $\sigma_j \in \mathcal{S}^{sec}$ is not required to have a MAC verification processing, (10e) and (10f) guarantee that its finish time c_j^σ equals the finish time $o_{m,n}^f + L_{slot}$ of the frame f_m into which it is packed on the FlexRay bus. And (10g) specifies that the finish time of each automotive control signal is within its deadline.

$$\forall g, \forall f_m, f_{m'}, m \neq m',$$

$$\alpha \in \left\{ 0, \dots, \frac{\text{lcm}(RE_m^f, RE_{m'}^f)}{RE_m^f} - 1 \right\}, \quad (11)$$

$$\beta \in \left\{ 0, \dots, \frac{\text{lcm}(RE_m^f, RE_{m'}^f)}{RE_{m'}^f} - 1 \right\}$$

$$\begin{aligned} \alpha \cdot RE_m^f + o_{m,g}^f + L_g^e \\ \leq \beta \cdot RE_{m'}^f + o_{m',g}^f + M \cdot (1 - \delta_{g,m,m'}^{\alpha,\beta}) + M \end{aligned} \quad (12a)$$

$$\cdot (1 - \mu_{m,g}^f) + M \cdot (1 - \mu_{m',g}^f)$$

$$\begin{aligned} \beta \cdot RE_{m'}^f + o_{m',g}^f + L_g^e \\ \leq \alpha \cdot RE_m^f + o_{m,g}^f + M \cdot \delta_{g,m,m'}^{\alpha,\beta} + M \cdot (1 - \mu_{m,g}^f) \\ + M \cdot (1 - \mu_{m',g}^f) \end{aligned} \quad (12b)$$

In the final of this part, constraints (12a) and (12b) ensure that the authentication-related operations of two frames do not overlap on any ECU. The variable $\delta_{g,m,m'}^{\alpha,\beta}$ is used for switching; i.e., one of (12a) or (12b) is trivially satisfied depending on $\delta_{g,m,m'}^{\alpha,\beta}$.

(c) Task Scheduling

$$\forall \tau_i \in \Gamma^{app}, \quad (13a)$$

$$o_i^\tau + L_i^\tau \leq D_i^\tau$$

$$\forall g, \forall \tau_i, \tau_{i'} \in \Gamma_g, i \neq i',$$

$$\alpha \in \left\{ 0, \dots, \frac{\text{lcm}(P_i^\tau, P_{i'}^\tau)}{P_i^\tau} - 1 \right\}, \quad (14)$$

$$\beta \in \left\{ 0, \dots, \frac{\text{lcm}(P_i^\tau, P_{i'}^\tau)}{P_{i'}^\tau} - 1 \right\}$$

$$\alpha \cdot P_i^\tau + o_i^\tau + L_i^\tau \leq \beta \cdot T_{i'}^\tau + o_{i'}^\tau + M \cdot (1 - \vartheta_{i,i'}^{\alpha,\beta}) \quad (14b)$$

$$\beta \cdot T_{i'}^\tau + o_{i'}^\tau + L_{i'}^\tau \leq \alpha \cdot P_i^\tau + o_i^\tau + M \cdot \vartheta_{i,i'}^{\alpha,\beta} \quad (14c)$$

Constraint (13a) determines that the finish time of a task τ_i , i.e., the sum of its start time o_i^τ and execution time L_i^τ , is within its deadline. Constraints (14b) and (14c) ensure that two tasks do not overlap on any ECU, where the variable $\vartheta_{i,i'}^{\alpha,\beta}$ is used for switching.

(d) Data Dependency

$$\begin{aligned} \forall \lambda_r \in \Lambda^{sec} \cup \Lambda^{app}, (\sigma_j, \tau_i) \in \mathcal{G}_r^{app}, \\ c_j^\sigma \leq o_i^\tau \end{aligned} \quad (15a)$$

$$\begin{aligned} \forall \lambda_r \in \Lambda^{app}, (\tau_i, \sigma_j) \in \mathcal{G}_r^{app}, \\ o_i^\tau + L_i^\tau \leq o_j^\sigma \end{aligned} \quad (15b)$$

Constraints (15a) and (15b) guarantee that the predecessor must complete its execution before all its successors start in an automotive control or key authentication application.

$$\forall m,$$

$$\varphi_m^f = \left\lceil \frac{o_{m,T_m}^f}{P_{int}} \right\rceil \quad (16a)$$

$$\forall m,$$

$$\left\lceil \frac{o_{m,n}^f + L_{slot}}{P_{int}} \right\rceil \leq \varphi_m^f \quad (16b)$$

$$\forall g, \forall \tau_i \in \{\Gamma_g^{sec} \mid V_i^\tau = T_m^f\},$$

$$\begin{aligned} o_{m,g}^f \geq o_i^\tau + L_i^\tau + \varphi_m^f \cdot P_{int} + M \cdot (1 - r_{m,g}^f) + M \\ \cdot (1 - \pi_m^f) \end{aligned} \quad (16c)$$

According to the authentication mechanism, an automotive control frame is accepted and stored awaiting to be authenticated by its receiver only when the key used to generate its MAC remains secret; i.e., the sender has not reached the time interval for releasing this key. Since the key is defined to be released in its corresponding next time interval, the transmission of each frame must be completed before the start of the next interval. Therefore, for a frame f_m , (16a) first computes the number of the time interval φ_m^f in which its MAC key is released. And then, (16b) guarantees that the start time of a frame on its sender and its finish time on the FlexRay bus belong to the same time interval.

Besides, an automotive control frame will be available to its receiver after the verification task for its MAC key is completed. Therefore, (16c) guarantees that the start time of an automotive control frame on its receiver must be later than the finish time of the verification task for its MAC key, where

Γ_g^{sec} denotes the set of the key verification tasks executed on ECU e_g .

$$\forall g, m, \forall \tau_i \in \Gamma_g, \quad \alpha \in \left\{ 0, \dots, \frac{\text{lcm}(P_i^\tau, RE_m^f)}{P_i^\tau} - 1 \right\}, \quad (17)$$

$$\beta \in \left\{ 0, \dots, \frac{\text{lcm}(P_i^\tau, RE_m^f)}{RE_m^f} - 1 \right\}$$

$$\alpha \cdot P_i^\tau + o_i^\tau + L_i^\tau \leq \beta \cdot RE_m^f + o_{m,g}^f + M \quad (18a)$$

$$\cdot (1 - \eta_{g,i,m}^{\alpha,\beta}) + M \cdot (1 - \mu_{m,g}^f) + M \cdot (1 - \theta_m^f)$$

$$\beta \cdot RE_m^f + o_{m,g}^f + L_g^e \leq \alpha \cdot P_i^\tau + o_i^\tau + M \cdot \eta_{g,i,m}^{\alpha,\beta} + M \quad (18b)$$

$$\cdot (1 - \mu_{m,g}^f) + M \cdot (1 - \theta_m^f)$$

Constraints (18a) and (18b) ensure that the authentication-related operations of the frames and the other tasks do not overlap on any ECU resource, where the variable $\eta_{g,i,m}^{\alpha,\beta}$ is used for switching.

(e) End-to-End Latency

$$\forall \rho_\epsilon \in \mathcal{FPP}^{app}, \quad c_{Des_\epsilon^\rho}^\tau - o_{Src_\epsilon^\rho}^\tau \leq D_\epsilon^\rho \quad (19)$$

Constraint (19) ensures that the end-to-end delay must be less than the deadline for each function path, where Des_ϵ^ρ and Src_ϵ^ρ are the source and sink object of the function path $\rho_\epsilon \in \mathcal{FPP}^{app}$, respectively.

4.3.3. Objective Functions. Subject to the above constraints, we can seek optimality with respect to different cost functions. A quite important objective, related to timing performance, is to maximize the laxity (difference between deadlines and response times) among all latency-sensitive function paths:

$$\text{Maximize } \sum_{\rho_\epsilon \in \mathcal{FPP}^{app}} D_\epsilon^\rho + o_{Src_\epsilon^\rho}^\tau - c_{Des_\epsilon^\rho}^\tau \quad (20)$$

We can alternatively minimize the consumption of network bandwidth, therefore improving extensibility. For FlexRay 2.1, an important objective related to extensibility is to minimize the number of the used slot identifiers:

$$\text{Minimize } \sum_{e_g \in E} \sum_k v_{g,k} \quad (21a)$$

For FlexRay 3.0, the objective is to minimize the number of the used slots:

$$\text{Minimize } \sum_{e_g \in E} \sum_w \sum_k v_{g,w,k} \quad (21b)$$

5. Experimental Results and Discussion

In order to assess the effectiveness and efficiency of the proposed MILP-based security-aware scheduling approach (hereafter referred to as MILP-S), we conducted extensive experiments by scheduling a number of real-time automotive applications on the FlexRay-based system architecture. The MILP is solved using LINGO 11.0 on a machine with a 2.8 GHz processor and 8 GB memory. The MACs are computed using hash function HMAC-MD5. We measure the execution times of HMAC-MD5 on Infineon TriCore, a representative automotive 32-bit microcontroller. A MAC generation/verification operation takes 11 μ s.

In all experiments, the cost functions with respect to latency or extensibility are used as the criterions of performance evaluation. To assess the impact of the additional authentication mechanism on the system performances after using the proposed MILP-S, we compare the results of MILP-S and two non-security-aware scheduling optimization approaches, MILP-NS and ASAP-NS. The MILP-NS is based on the same MILP formulation, but it does not consider the authentication mechanism-related operations. The ASAP-NS is to (a) pack the signals generated by the same task into a frame and (b) schedule the tasks and frames using As-Soon-As-Possible (ASAP) scheduling. Such a solution would be chosen by a good designer without the help of the dedicated optimization tool. It should be noted that, since ASAP-based security-aware scheduling approach cannot obtain the feasible scheduling solutions (i.e., satisfying all the design constraints) in all experiments, this section does not present the results of this approach.

5.1. Case Study: An Automotive X-by-Wire System. The system consists of 8 ECUs interconnected by the FlexRay bus. There are 25 tasks and 55 signals in the X-by-Wire applications. Tables 2 and 3 show the periods, the worst-case execution time of tasks (in microseconds), and the sizes of signals (in bits). The lengths of the communication cycle, ST slot, and frame payload are 1 ms, 0.035 ms, and 16 bytes, respectively. There are 22 slots in the FlexRay ST segment. The maximum security goal of all signals is $1 \cdot 10^{-8}$ per hour, which corresponds to the most stringent safety level of ISO 26262, i.e., Automotive Safety Integrity Level D. The MAC size is truncated to 48 bits according to constraint (1).

Table 4 depicts the comparison results of MILP-S, MILP-NS, and ASAP-NS with respect to latency and extensibility laxity metric functions. It is shown that MILP-S can guarantee the schedulability of system with authentication mechanism overheads and constraints whether the slot multiplexing mechanism of FlexRay 2.1 or FlexRay 3.0 is applied. Specifically, when the extensibility is taken as optimization objective and FlexRay 3.0 is applied, the number of slots used by MILP-S is greater than that of the non-security-aware scheduling optimization approaches MILP-NS and ASAP-NS. This is because each sender needs to transmit the released key in each time interval after using the TESLA authentication mechanism and consumes more slots. Even so, for safety-critical automotive systems, a small portion of their bandwidth resources is still worth achieving security.

TABLE 2: Tasks of the x-by-wire system.

Task	ECU	Period	Execution time	Task	ECU	Period	Execution time
τ_1	e_7	8000	350	τ_{14}	e_7	8000	360
τ_2	e_8	8000	330	τ_{19}	e_6	8000	770
τ_3	e_3	8000	200	τ_{20}	e_6	8000	770
τ_4	e_4	8000	190	τ_{21}	e_7	8000	305
τ_5	e_6	8000	200	τ_{22}	e_8	8000	500
$\tau_6/\tau_7/\tau_8/\tau_9$	$e_3/e_4/e_5/e_6$	8000	860	τ_{23}	e_8	8000	455
$\tau_{10}/\tau_{11}/\tau_{12}/\tau_{13}$	$e_3/e_4/e_5/e_6$	2000	40	τ_{24}	e_1	2000	780
$\tau_{15}/\tau_{16}/\tau_{17}/\tau_{18}$	$e_3/e_4/e_5/e_6$	8000	780	τ_{25}	e_2	8000	890

TABLE 3: Signals of the x-by-wire system.

Signal	Send	Receive	Size	Signal	Send	Receive	Size
$\sigma_1 - \sigma_4$	τ_1	$\tau_6/\tau_7/\tau_8/\tau_9$	32	σ_{25}	τ_8	τ_{20}	8
$\sigma_5 - \sigma_8$	τ_2	$\tau_6/\tau_7/\tau_8/\tau_9$	32	$\sigma_{26} - \sigma_{28}$	τ_9	τ_{19}	8
$\sigma_9 - \sigma_{10}$	τ_3	$\tau_7/\tau_8/\tau_9$	32	$\sigma_{29} - \sigma_{33}$	τ_{15}	τ_{21}	4
$\sigma_{11} - \sigma_{12}$	τ_4	$\tau_6/\tau_8/\tau_9$	32	$\sigma_{34} - \sigma_{27}$	τ_{16}	τ_{21}	4
$\sigma_{13} - \sigma_{16}$	τ_5	$\tau_6/\tau_7/\tau_8$	32	$\sigma_{38} - \sigma_{41}$	τ_{17}	τ_{21}	4
$\sigma_{17} - \sigma_{18}$	τ_{10}	$\tau_{11}/\tau_{12}/\tau_{13}$	4	$\sigma_{42} - \sigma_{45}$	τ_{18}	τ_{21}	4
$\sigma_{19} - \sigma_{20}$	τ_{14}	$\tau_{15}/\tau_{16}/\tau_{17}/\tau_{18}$	8	$\sigma_{46} - \sigma_{49}$	τ_{21}	τ_{22}/τ_{23}	4
$\sigma_{21} - \sigma_{22}$	τ_6	τ_{19}/τ_{20}	4	$\sigma_{50} - \sigma_{52}$	τ_{24}	τ_{21}	32
$\sigma_{23} - \sigma_{24}$	τ_7	τ_{19}/τ_{20}	8	$\sigma_{53} - \sigma_{55}$	τ_{25}	τ_{21}	32

When the extensibility is taken as optimization objective and FlexRay 2.1 is applied, the number of the slot identifiers used by MILP-S is the same as that of MILP-NS and even less than that of ASAP-NS. These demonstrate that, although the introduction of authentication function requires more slots to transmit the keys, these slots share the slot identifiers as far as possible by using the proposed MILP-S, thereby not degrading the extensibility of systems in this scenario.

On the other hand, when the timing performance is taken as optimization objective, the laxity of all time-sensitive function paths obtained by MILP-S is slightly lower than that of the non-security-aware scheduling optimization approaches MILP-NS and especially ASAP-NS, whether the slot multiplexing mechanism of FlexRay 2.1 or FlexRay 3.0 is applied. These demonstrate that the introduction of authentication function hardly affects the timing performance of systems after using the proposed MILP-S.

5.2. Scalability Analysis. To assess the scalability of the proposed approach, we evaluate the performance of the proposed scheduling approach on systems with different numbers of time-sensitive function paths. We generated a set of synthetic applications based on realistic automotive system cases. Specifically, the periods of tasks and signals are varied among the range [5, 10, and 20 ms]. The lengths of the communication cycle, ST slot, and frame payload are 2.5 ms, 0.032 ms, and 16 bytes, respectively. There are 45 slots in the FlexRay ST segment. The number of ECUs is 8. The average ratio of deadline to period of each time-sensitive function path is 0.7. The maximum security goal of all signals is $1 \cdot 10^{-7}$ per hour, which corresponds to the Automotive Safety Level

C of ISO 26262 standard. The MAC size should be no less than 43 bits according to constraint (1). In these experiments, it is set to 6 bytes for alignment.

In the figures of results, the horizontal axis marks the number of paths, which denotes the scale of the experiments, as the numbers of tasks and signals simultaneously grow. The number of time-sensitive function paths is varied among the range [10, 15, 20, 25]. When the number of paths is 25, the numbers of automotive applications, tasks, and signals are 12, 54, and 150, respectively. The average cost of a task is 0.55 ms.

Figure 6(a) depicts the comparison results of MILP-S, MILP-NS, and ASAP-NS with respect to extensibility metric function on systems applying FlexRay 2.1. It is shown that the number of the slot identifiers obtained by MILP-S is average 15% more than that of MILP-NS and 35% less than that of ASAP-NS. These demonstrate that the introduction of authentication function has not led to huge increases in slot identifiers consumption by using the proposed MILP-S.

Similarly, Figure 6(b) depicts the comparison results of MILP-S, MILP-NS, and ASAP-NS with respect to extensibility metric function on systems applying FlexRay 3.0. It is shown that MILP-S used the largest number of slots. In addition, as the number of paths increases, the differences in slot consumption between the security-aware MILP-S and non-security-aware MILP-NS and ASAP-NS decrease slightly. This is because, when the number of function paths grows and the number of ECUs remains constant, each ECU produces more signals, thus providing more possibility of optimization of frame packing and scheduling.

On the other hand, Figure 6(c) depicts the comparison results of MILP-S, MILP-NS, and ASAP-NS with respect to

TABLE 4: Experimental results of the x-by-wire system.

Approach	Extensibility		Time performance	
	FlexRay 2.1	FlexRay 3.0	FlexRay 2.1	FlexRay 3.0
MILP-S	8	64	73.65	74.28
ASAP-NS	18	21	85.63	85.63
MILP-NS	8	15	99.02	99.02

FIGURE 6: Comparison results of MILP-S, MILP-NS, and ASAP-NS for different optimization metrics versus number of paths. (a) Number of the slot identifiers. (b) Number of the slots. (c) Laxity.

latency metric function on systems with different numbers of time-sensitive function paths. Since the MILP-NS and ASAP-NS returned the same laxities when FlexRay 2.1 and FlexRay 3.0 are applied in the systems, and for the MILP-S, the laxity differences between applying FlexRay 2.1 and FlexRay 3.0

were all slight (within 1 ms), and both results are represented as a graph. First, it is shown that MILP-S can still guarantee the schedulability of systems with authentication mechanism overheads and constraints in all cases. Second, the laxity of all function paths obtained by MILP-S is average 22% lower than

FIGURE 7: Runtime of the MILP solver for different optimization metrics versus number of paths. (a) Extensibility. (b) Latency.

that of MILP-NS and only 16% lower than that of ASAP-NS. These demonstrate that authentication function hardly affects the timing performance of systems by using the proposed MILP-S.

At the end of this part, Figure 7 shows the runtime of the MILP solver for each experiment. It is shown that, for both optimization metrics and both versions of FlexRay protocols, the solver is able to find the optimal solution within the time limit when the number of paths is less than 25 and returns a feasible solution when the number of paths is 25. A straightforward application of the proposed MILP approach to systems with hundreds of signals could provide only a feasible solution or not feasible, instead of the optimum. In this case, it is still possible to use the approach incrementally, by dividing the applications into several subsets, as in [34]. The application subsets could be scheduled iteratively starting from the most critical components and moving to less critical ones. This would require solving several consecutive MILP problems. Although this method cannot guarantee optimality, it should provide a good tradeoff between optimality and runtime.

6. Conclusions

In this paper, we have proposed an approach to address both the information security and functional safety in the scheduling design of FlexRay-based automotive systems. An authentication mechanism based on delayed exposure of one-way key chains is applied on TTEthernet to protect against forgery and replay attacks. The authentication mechanism provides an appropriate trade-off between security level and resource overhead. Furthermore, an MILP formulation is proposed for solving the scheduling optimization problem of FlexRay-based real-time automotive systems subject to both authentication mechanism constraints and other traditional

design constraints. The objective of MILP approach is to maximize the laxity on function paths (therefore improving timing performance) or to minimize the number of used slots or slot identifiers (therefore improving extensibility). The experiment results show that the proposed MILP approach can still guarantee the schedulability of systems with authentication mechanism overheads and constraints and achieve good performance with timing and extensibility.

Symbols

Elements and Sets

- τ_i : The i -th task
- σ_j : The j -th signal
- f_m : The m -th frame
- e_g : The g -th ECU
- λ_r : The r -th application
- ρ_ε : The ε -th path
- Γ : The set of tasks
- Γ^{app} : The set of automotive control-related tasks
- Γ_g : The set of tasks executed on ECU e_g
- Γ_g^{sec} : The set of the key verification tasks executed on ECU e_g
- $\mathcal{S}$: The set of signals
- $\mathcal{S}^{app}$: The set of automotive control-related signals
- $\mathcal{S}^{sec}$: The set of key signals
- $\mathcal{S}_{g,h}$: The set of all branches of the h -th multicast signal of ECU e_g
- $\mathcal{S}_g$: The set of the signals from ECU e_g
- $\mathcal{F}$: The set of frames
- $\mathcal{F}_g$: The set of the signals from ECU e_g
- E : The set of ECUs
- Λ^{app} : The set of automotive control applications

Λ^{sec} : The set of key authentication applications

$\mathcal{F}^{app}$: The set of time-sensitive function paths

Constants

E_i^τ : The ECU resource that task τ_i needs to execute

L_i^τ : The execution time of task τ_i

P_i^τ : The period of task τ_i

D_i^τ : The deadline that task τ_i needs to verify

V_i^τ : The ECU that key verification task τ_i needs to verify

T_j^σ : The ECU that sends signal σ_j

R_j^σ : The ECU that receives signal σ_j

L_j^σ : The length of signal σ_j

P_j^σ : The period of signal σ_j

D_j^σ : The deadline of signal σ_j

Z_j^σ : The type of signal σ_j

T_m^f : The ECU that sends frame f_m

RE_m^f : The repetition of frame f_m

L_g^e : The execution time of the MAC computation function on ECU e_g

D_ϵ^p : The deadline of path ρ_ϵ

Src_ϵ^p : The source task of path ρ_ϵ

Des_ϵ^p : The sink task of path ρ_ϵ

L_{frame} : The upper limit of frame length

L_{MAC} : The length of MAC

P_{int} : The interval duration of key release

M : A large constant for linearization

L_{cycle} : The length of a FlexRay communication cycle

L_{slot} : The length of a ST slot.

Data Availability

The data from automotive x-by-wire system used to support the findings of this study are included within the article. The scalability analysis-related data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Jilin Science and Technology Key Project Grant [grant number 20150204034GX].

References

- [1] F. Sagstetter, M. Lukasiewicz, and S. Chakraborty, "Generalized asynchronous time-triggered scheduling for FlexRay," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 36, no. 2, pp. 214–226, 2017.
- [2] CAN in Automation, <http://www.can-cia.org/can/>, 2003.
- [3] FlexRay Consortium, *FlexRay Communications System Protocol Specification Version 3.0.1*, 2010.
- [4] J. Broy and K. D. Muller-Glaser, "The impact of time-triggered communication in automotive embedded systems," in *Proceedings of the 2007 Symposium on Industrial Embedded Systems*, pp. 353–356, IEEE, Lisbon, Portugal, 2007.
- [5] K. C. Lee, M. H. Kim, S. Lee, and H. H. Lee, "IEEE-1451-based smart module for in-vehicle networking systems of intelligent vehicles," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 6, pp. 1150–1158, 2004.
- [6] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1222, 2005.
- [7] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaaniche, and Y. Laarouchi, "Survey on security threats and protection mechanisms in embedded automotive networks," in *Proceedings of the 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, pp. 1–12, Budapest, Hungary, 2013.
- [8] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pp. 528–533, IEEE, 2011.
- [9] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks Practical examples and selected short-term countermeasures," *Reliability Engineering & System Safety*, vol. 96, no. 1, pp. 11–25, 2011.
- [10] K. Koscher, A. Czeskis, F. Roesner et al., "Experimental security analysis of a modern automobile," in *Proceedings of the 31st IEEE Symposium on Security and Privacy*, vol. 41, pp. 447–462, 2010.
- [11] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015.
- [12] D. Bleichenbacher and U. M. Maurer, "Directed acyclic graphs, one-way functions and digital signatures," *CRYPTO'94*, pp. 75–82, 1994.
- [13] A. Perrig, R. Canetti, J. D. Tygar, and D. X. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 56–73, Berkeley, Calif, USA, 2000.
- [14] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS)*, pp. 28–37, Philadelphia, Pa, USA, 2001.
- [15] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, "Time valid one-time signature for time-critical multicast data authentication," in *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1233–1241, Brazil, 2009.
- [16] C. Szilagyi and P. Koopman, "Flexible multicast authentication for time-triggered embedded control network applications," in *Proceedings of IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, pp. 165–174, IEEE, Lisbon, Portugal, June 2009.
- [17] C. Szilagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered embedded control networks," in *Proceedings of the 5th Workshop on Embedded Systems Security (WESS)*, pp. 1–10, Scottsdale, Az, USA, 2010.
- [18] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 35–46, 2001.

- [19] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Proceedings of the 10th Network and Distributed System Security Symposium (NDSS)*, pp. 263–276, San Diego, Calif, USA, 2003.
- [20] D. Liu and P. Ning, "Multilevel TESLA: Broadcast authentication for distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 4, pp. 800–836, 2004.
- [21] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 35–46, San Diego, Calif, USA, 2001.
- [22] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 189–199, Rome, Italy, 2001.
- [23] K. Schmidt and E. G. Schmidt, "Message scheduling for the FlexRay protocol: the static segment," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2170–2179, 2009.
- [24] M. Kang, K. Park, and M.-K. Jeong, "Frame packing for minimizing the bandwidth consumption of the flex Ray static segment," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4001–4008, 2013.
- [25] K. Schmidt and E. G. Schmidt, "Optimal message scheduling for the static segment of flexray," in *Proceedings of the IEEE Vehicular Technology Conference (VTC 2010-Fall)*, vol. 44, pp. 1–5, IEEE, Ottawa, ON, Canada, 2010.
- [26] M. Grenier, L. Havet, and N. Navet, "Configuring the communication on FlexRay: the case of the static segment," in *Proceedings of the 4th European Congress on Embedded Real Time Software (ERTS)*, Toulouse, France, 2008.
- [27] M. Lukasiewicz, M. Glaß, J. Teich, and P. Milbredt, "FlexRay schedule optimization of the static segment," in *Proceedings of the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, pp. 363–372, Grenoble, France, 2009.
- [28] R. Zhao, G. H. Qin, and J. Q. Liu, "Optimal scheduling of the flexray static segment based on two-dimensional bin-packing algorithm," *International Journal of Automotive Technology*, vol. 17, no. 4, pp. 703–715, 2016.
- [29] R. Zhao, G.-H. Qin, and J.-Q. Liu, "A rectangle bin packing optimization approach to the signal scheduling problem in the FlexRay static segment," *Frontiers of Information Technology and Electronic Engineering*, vol. 17, no. 4, pp. 375–388, 2016.
- [30] H. Zeng, M. Di Natale, A. Ghosal, and A. Sangiovanni-Vincentelli, "Schedule optimization of time-triggered systems communicating over the flexray static segment," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 1–17, 2011.
- [31] M. Hu, J. Luo, Y. Wang, M. Lukasiewicz, and Z. Zeng, "Holistic scheduling of real-time applications in time-triggered in-vehicle networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1817–1828, 2014.
- [32] M. Lukasiewicz, R. Schneider, D. Goswami, and S. Chakraborty, "Modular scheduling of distributed heterogeneous time-triggered automotive systems," in *Proceedings of the 17th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 665–670, Sydney, NSW, Australia, 2012.
- [33] D. Roy, L. Zhang, W. Chang, D. Goswami, and S. Chakraborty, "Multi-objective co-optimization of flexray-based distributed control systems," in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, Vienna, Austria, 2016.
- [34] K. Jiang, P. Eles, and Z. Peng, "Co-design techniques for distributed real-time embedded systems with communication security constraints," in *Proceedings of the in Design, Automation Test in Europe Conference and Exhibition*, pp. 947–952, IEEE, Dresden, Germany, 2012.
- [35] A. Wasicek, C. El-Salloum, and H. Kopetz, "Authentication in time-triggered systems using time-delayed release of keys," in *Proceedings of the IEEE 14th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*, pp. 31–39, Newport Beach, Calif, USA, 2011.
- [36] C. Lin, Q. Zhu, C. Phung, and A. Sangiovanni-Vincentelli, "Security-aware mapping for CAN-based real-time distributed automotive systems," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 24–31, San Jose, Calif, USA, 2013.
- [37] C. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware mapping for TDMA-based real-time distributed systems," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, IEEE, San Jose, Calif, USA, 2014.
- [38] Z. Gu, G. Han, H. Zeng, and Q. Zhao, "Security-aware mapping and scheduling with hardware co-processors for flexray-based distributed embedded systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 3044–3057, 2016.

Submit your manuscripts at
www.hindawi.com