IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 30, 2019, accepted June 25, 2019, date of publication July 1, 2019, date of current version July 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926113

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

RUI ZHAO[1], GUIHE QIN[2], YING LYU[3], AND JIE YAN[2]

[1]Automotive Engineering Department, Jilin University, Changchun 130000, China
[2]Computer Science and Technology Department, Jilin University, Changchun 130000, China
[3]State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130000, China

Corresponding author: Ying Lyu (lvying@faw.com.cn)

This work was supported by the Jilin Science and Technology Key Project under Grant 20150204034GX.

ABSTRACT TTEthernet is a deterministic, congestion-free, and high-bandwidth communication protocol based on the Ethernet standard that provides a powerful network solution for developing safety-critical distributed real-time automotive systems. 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, TTEthernet-based architecture does not have direct support for secure communication. When deploying the security mechanisms on these architectures, a major 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 TTEthernet 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 the TTEthernet-based real-time automotive systems subject to both authentication mechanism constraints and other traditional design constraints. The extensive experiments are conducted to demonstrate the effectiveness and efficiency of the proposed method.

INDEX TERMS TTEthernet, real-time, automotive systems, security, scheduling.

I. INTRODUCTION

TTEthernet [1] is a highly available networking technology that implements time-triggered communication mechanisms over Ethernet standard to satisfy the requirements of fully deterministic, high-speed and low-cost communication. It can guarantee constant latency for multi-hop network communication routes relying on fault-tolerant synchronization services. In addition to time-triggered (TT) traffic, TTEthernet supports rate-constrained traffic (compatible with ARINC 664P7 [2]) and standard Ethernet [3] traffic to provide flexibility. These capabilities make TTEthernet a powerful network solution for developing real-time safety-critical automotive systems [4]–[6].

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. For example, researchers have demonstrated that modern automotive systems are vulnerable to attacks through various interfaces such as OBD-II, Bluetooth, Wi-Fi, DSRC, GPS and 3G/4G [7]–[10]. Once one Electronic Control Units (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 TTEthernet-based architecture, it does not directly provide multicast source authentication to protect data authenticity.

Integrating authentication mechanisms into TTEthernet-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

The associate editor coordinating the review of this manuscript and approving it for publication was Liehuang Zhu.

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.

Given these issues, our major contributions are as follows.

1) First, we apply the TESLA [11] authentication mechanism based on delayed exposure of keys to protect against forgery and replay attacks on TTEthernet. It provides an appropriate trade-off between security level and resource overhead, compared with other multicast authentication approaches. Moreover, we make a modification to the original TESLA in order to improve on the authentication delay.
2) Furthermore, we propose a mixed integer linear programming (MILP) formulation that efficiently solves the optimal scheduling problem of TTEthernet-based real-time automotive systems with authentication mechanism constraints. The scheduling design includes (a) the packing of automotive control and authentication mechanism-related signals to TTEthernet frames, (b) the scheduling of frames on TTEthernet, 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 bandwidth consumption, therefore improving extensibility. To the best of our knowledge, this is the first work to integrate security constraints and other traditional constraints in the scheduling design of TTEthernet-based automotive systems.

The remainder of the paper is organized as follows. Section II overviews some related work. Section III introduces the system model. Section IV presents the security mechanism and security model. Section V formally states the security-aware optimization scheduling problem whose solution is tackled using MILP-based method. Section VI presents experimental results, with conclusions following in Section VII.

II. RELATED WORK

A. MULTICAST AUTHENTICATION

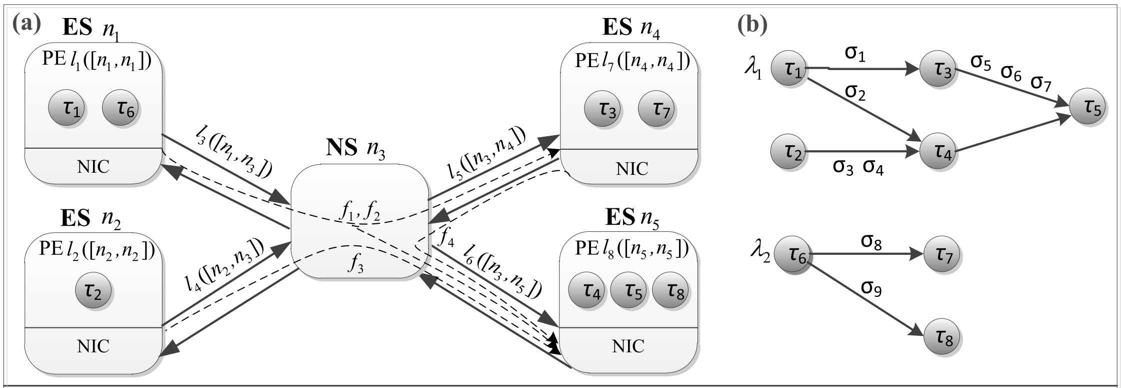
Digital signatures based on public key cryptography provide an elegant method for signing multicast data, but they are not the solution in our context because of the high computational overhead. Although the computational overhead could be alleviated by dedicated circuits, such as FPGAs or ASICs, this will add component costs, an issue that is typically avoided by manufacturers. Schemes using one-time signatures [12]–[15] are much more computationally efficient than traditional public key signatures. However, one-time signatures can incur kilobytes of authentication data per message, that 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 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 [11], [18]–[21], 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 TTEthernet, and make a modification to the original TESLA so that it is more appropriate for our application setting.

B. SCHEDULING

Steiner [22] proposed a scheduling method based satisfiability modulo theory (SMT) for the TTEthernet TT traffic. They defined a set of scheduling constraints and used the SMT solver to find a solution that satisfies all constraints. Steiner [23] proposed to introduce periodic slots into static schedules to help reduce the RC delays. Suethanuwong [24] proposed a scheduling approach to compute the periods and offsets of TT frames. Tămaş-Selicean *et al.* [25] proposed a Tabu-search-based metaheuristic for TT schedule optimization. They [26] also suggested a Tabu-search-based metaheuristic to optimize TTEthernet networks, where in addition to optimal TT schedules, the proposed method provides optimal bandwidth allocation of RC frames. Dvořák [27] developed a three-stage algorithm to create the communication schedules for TT traffic. These works only focused 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.

The system-level scheduling on both signals and tasks has also been studied for TTEthernet-based real-time systems. Zhang *et al.* [28] applied mixed-integer programming (MIP)



(c)

Signals Parameter List

Signals	Period(μs)	Length(byte)	Group	Resource Path
σ_1	5000	8	$S_{1,1}$	$[n_1,n_1]$ $[n_1,n_3]$ $[n_3,n_4]$ $[n_4,n_4]$
σ_2	5000	8	$S_{1,1}$	$[n_1,n_1]$ $[n_1,n_3]$ $[n_3,n_5]$ $[n_5,n_5]$
σ_3	5000	56	$S_{2,1}$	$[n_2,n_2]$ $[n_2,n_3]$ $[n_3,n_5]$ $[n_5,n_5]$
σ_4	5000	56	$S_{2,2}$	$[n_2,n_2]$ $[n_2,n_3]$ $[n_3,n_5]$ $[n_5,n_5]$
σ_5	5000	16	$S_{7,1}$	$[n_4,n_4]$ $[n_4,n_3]$ $[n_3,n_5]$ $[n_5,n_5]$
σ_6	5000	8	$S_{7,2}$	$[n_4,n_4]$ $[n_4,n_3]$ $[n_3,n_5]$ $[n_5,n_5]$
σ_7	5000	16	$S_{7,3}$	$[n_4,n_4]$ $[n_4,n_3]$ $[n_3,n_5]$ $[n_5,n_5]$
σ_8	2500	80	$S_{1,2}$	$[n_1,n_1]$ $[n_1,n_3]$ $[n_3,n_4]$ $[n_4,n_4]$
σ_9	2500	80	$S_{1,2}$	$[n_1,n_1]$ $[n_1,n_3]$ $[n_3,n_5]$ $[n_5,n_5]$

Tasks Parameter List

Tasks	Period(μs)	Execution time(μs)
τ_1	5000	400
τ_2	5000	750
τ_3	5000	200
τ_4	5000	320
τ_5	5000	390
τ_6	2500	400
τ_7	2500	300
τ_8	2500	420

FIGURE 1. System model example, where signals σ_1, σ_2 are packed into frame f_1, σ_8, σ_9 into f_2, σ_3, σ_4 into f_3, and $\sigma_5, \sigma_6, \sigma_7$ into f_4.

to solve the scheduling problem for Ethernet-based TT systems. Craciunas and Oliver [29] formulated the scheduling problem of TTEthernet-based distributed systems using first-order logical constraints and applied SMT and MIP solvers to solve it, respectively. Abuteir and Obermaisser [30] proposed a scheduling algorithm based on neighborhood search for multi-cluster TTEthernet systems. However, none of the above-mentioned works considered the interference of security operations on system applications. In this work, we provide an MILP formulation for solving the scheduling optimization problem of TTEthernet-based real-time automotive systems while meeting the requirements of both information security and functional safety.

III. SYSTEM

A. HARDWARE ARCHITECTURE

A TTEthernet-based architecture consists of a set of ECUs (usually called end systems, ESes) interconnected by physical links and network switches (NSes). The physical links are fully duplex and the networks can be multi-hop. Each ES is composed of a processing element (PE) containing a CPU, RAM and I/O resources, and a network interface card (NIC). An example of the TTEthernet architecture is presented in Fig. 1(a).

We denote $\mathcal{N}$ the set of communication nodes (ESes and NSes) in a TTEthernet-based architecture, and $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ the set of directional communication links between nodes, i.e., $[\mathrm{n_a n_b}] \in \mathrm{L}[\mathrm{n_a}, \mathrm{n_b}] \in \mathrm{L}$ is an ordered tuple representing a communication link from node $n_a \in \mathcal{N}$ to $n_b \in \mathcal{N}$. Since the scheduling problem addressed in this paper is performed at the system-level, we also consider PEs of the ESes for running tasks in addition to the network link resources. We model the PE of each ES as a directional self-link $[n_a, n_a] \in \mathcal{L}$, which we call PE link resource, connecting an ES $n_a \in \mathcal{N}$ with itself. For simplicity, the network or PE link resource will also be identified and denoted by a single index, as in l_g.

B. SOFTWARE ARCHITECTURE

In our model, ESes are assumed to run 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 TT traffic arbitration model provided by the TTEthernet protocol. In TTEthernet, the transmission of a frame $f_m \in \{$ from its sender T_m^f to multiple receivers typically requires several transfers. The route of a frame is defined via the concept of virtual link, which is a logical data-flow path from one sender ES to one or more receiver ESes. For example, in Fig. 1(a), the virtual link of frame f_1 connects ES n_1 to n_4 and n_5, which can be denoted as $[[n_1, n_3], [n_3, n_4], [n_3, n_5]]$ or $[l_3, l_5, l_6]$. TT communication is done according to communication schedules determined offline and stored in the ESes and NSes. Each ES or NS will protect the network as it will only transmit frames as specified sending times in the schedule table. The start time of a TT frame f_m on each link resource it uses is specified by its period P_m^f and an offset within the period.

C. APPLICATION MODEL

We model an application $\lambda_r \in \Lambda^{app}$ to be processed in the system as a directed, acyclic graph G_r^{app}, 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 $(E_i^\tau, C_i^\tau, P_i^\tau, D_i^\tau)$, where E_i^τ denotes the PE link resource it needs to execute, C_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 $(T_j^\sigma, R_j^\sigma, W_j^\sigma, P_j^\sigma, D_j^\sigma)$, where T_j^σ and R_j^σ denotes the PE link resources required for sending and receiving it, W_j^σ is its bit width, P_j^σ is its period and D_j^σ is its deadline. In addition, its resource path (the ordered sequence of the used resources) is modeled by two sets $\mathcal{U}$ and $\mathcal{Q}$ derived from the base signal set and link resource set, where $(\sigma_j, l_g) \in \mathcal{U}$ denotes that the signal σ_j uses the resource l_g, and $\left(\sigma_j, l_g, l_{g'}\right) \in \mathcal{Q}$ denotes that the signal σ_j uses the resource l_g and $l_{g'}$ in order.

For example, in the application λ_2 of Fig. 1(b), task τ_6 generates a multicast signal, tasks τ_7 and τ_8 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 resource paths. Specifically, we define a set of signals $\mathcal{S}_{g,h}$ containing all of the branches of the h-th multicast signal of each PE link resource l_g, such as signals σ_8 and σ_9 are assumed to be the two branches of the second multicast signal of resource $l_1([n_1, n_1])$, their respective resource paths are listed in Fig. 1(c); thus, we have $\sigma_8, \sigma_9 \in \mathcal{S}_{1,2}$.

A function path $\rho_\varepsilon \in \mathcal{FP}^{app}$ from τ_i to $\tau_{i'}$ is a sequence $\left[\tau_i, \ldots, \tau_{i'}\right]$ of tasks such that there is a link between any two consecutive tasks. For instance, in application λ_1, there is a 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 $\tau_{i'}$ that produces a result that is dependent on the output of τ_i. The deadline D_ε^ρ of function path is set by system designers as an application requirement.

IV. SECURITY MECHANISM AND SECURITY MODEL

A. SECURITY MECHANISM

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 TTEthernet switches, malicious insider code, or tampering with ESes. We consider an active attacker model where an attacker can masquerade as other ESes to inject forged messages and can also replay messages. Attackers accessing the TTEthernet through corrupted nodes will have access to the key material in those ESes. An attacker must not be able to masquerade as any ES they do not already control to perform a successful attack [17].

Additionally, 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 inject a well-formed frame in another node's time slot. And an attack can only take one forgery or replay attempt per valid time slot, since transmitters are only permitted to send a frame per assigned time slot in TTEthernet.

2) OVERVIEW OF THE TESLA MECHANISM

In this work, we apply TESLA authentication mechanism to protect the authenticity of messages on TTEthernet. The main ideas 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.

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 (easy to compute but difficult to invert) using a one-way hash function H as $K_0, K_1, \ldots, K_n$, where $K_\varphi = H(K_{\varphi+1})$, and assigns the keys sequentially to the time intervals, as depicted in Fig. 2. The chain is used in reverse order starting with K_1. To bootstrap TESLA, the sender uses asymmetric cryptography to distribute the initial key K_0 to every receiver in the network authoritatively.

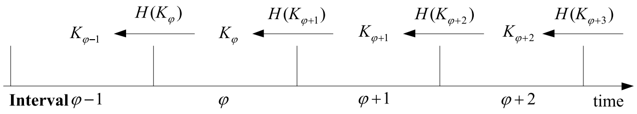


FIGURE 2. TESLA protocol example.

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 $K_{\varphi-d}$ that it can disclose.

When a receiver receives a message in φ-th time interval, it cannot yet verify the authenticity of the message. Instead, it puts the message into a buffer, 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 key $K_{\varphi-1}$ that $K_{\varphi-1} = H(K_\varphi)$.

3) MODIFICATION TO THE TESLA MECHANISM

To ensure the key to be disclosed in an interval can arrive at its receivers on time, TESLA protocol specify that the key must be appended to each message frame in that interval. The repeated transmission and verification of the same key will cause 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.

For this, we make a modification to the original TESLA combining with our application setting. Given the TTEthernet TT traffic provides highly deterministic communication, we specify that each key K_φ is released only once in its next

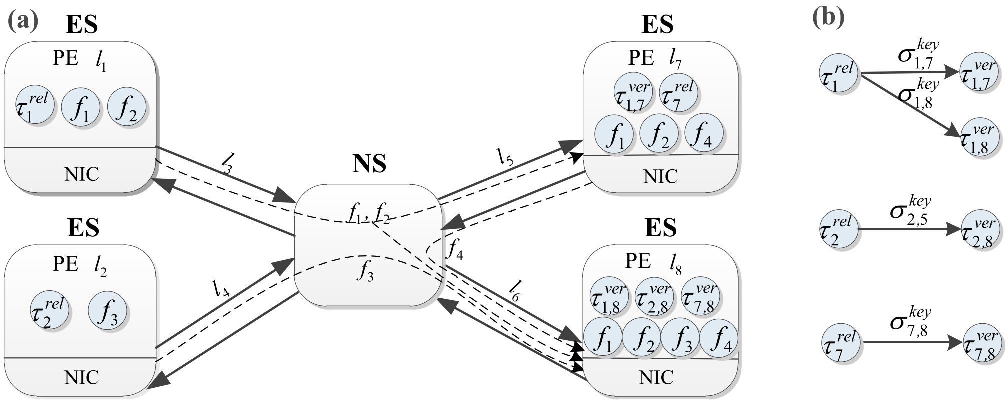


FIGURE 3. Security model example.

interval and is transmitted through the TT traffic to ensure its time determinism.

B. SECURITY MODEL

After applying the modified TESLA authentication mechanism to the system, extra information, i.e. keys, need 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 the PEs l_g of 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 l_g, key verification tasks $\tau_{g,g'}^{ver}$ generated by the receivers $l_{g'}$ and key signals $\sigma_{g,g'}^{key}$ produced by $\tau_{g,g'}^{rel}$ that is available to $\tau_{g,g'}^{ver}$. Fig. 3 illustrates the additional operations after applying the authentication mechanism to the example system in Fig. 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 $(E_i^\tau, V_i^\tau, C_i^\tau, P_i^\tau)$ where E_i^τ and V_i^τ are the PE link resources it needs to execute and verify, C_i^τ is its time cost indicating the execution time of the one-way hash function H on the PE 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 $(T_j^\sigma, R_j^\sigma, N_j^\sigma, W_j^\sigma, P_j^\sigma)$,where T_j^σ and R_j^σ are the PE link resources required for sending and receiving it, N_j^σ is the last network link resource required for transmitting it, W_j^σ is its bit width and P_j^σ is its period. Similar to the automotive control signals, its resource path is also modeled by two sets $\mathcal{U}$ and $\mathcal{Q}$ derived from the base key signal set and link resource set are defined. 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.

C. CHOICE OF INTERVAL DURATION OF KEY RELEASE

Following the authentication mechanism, the smaller the interval duration, the more frequently key authentication applications execute and thus the more processing and communication resources are consumed. But the larger the interval duration, 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 $\rho_\varepsilon \in \mathcal{FP}^{app}$, we let C_ε^ρ 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 that satisfying the following constraints:

$$\forall \rho_\varepsilon \in \mathcal{FP}^{app}, \quad P_{int} \cdot C_\varepsilon^\rho \leq D_\varepsilon^\rho \tag{1}$$

$$P_{int} \leq min_{\sigma_j \in \mathcal{S}^{app}} D_j^\sigma \tag{2}$$

$$\begin{aligned} P_{int} mod\, gcd_{\sigma_j \in \mathcal{S}^{app}} P_j^\sigma &= 0, \\ P_{int} = n \cdot gcd_{\sigma_j \in \mathcal{S}^{app}} P_j^\sigma, &\quad n \in Z^* \; or \end{aligned} \tag{3}$$

Relations (1) and (2) provide the time limits, i.e. the product of interval duration P_{int} and signal level C_ε^ρ cannot exceed the deadline D_ε^ρ for a function path $\rho_\varepsilon \in \mathcal{FP}^{app}$. 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 C_ε^ρ intervals. In addition, the interval duration P_{int} cannot exceed the minimum value of the deadlines D_j^σ of signal $\sigma_j \in \mathcal{S}^{app}$ to ensure that each signal can be authenticated by the receiver before its deadline. Relation (3) 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.

V. SECURITY-AWARE SCHEDULING

A. PROBLEM STATEMENT

The security-aware scheduling problem we are addressing in this paper can be formulated as follows. Given the system model and the security model generated by the authentication mechanism, we decide on the (1) packing of automotive control and authentication mechanism-related signals to TTEthernet frames, (2) scheduling of frames on TTEthernet, and (3) scheduling of automotive control and authentication mechanism- related tasks on respective PE, such that:

- the deadline constraints and the precedence constraints caused by information passing between all tasks and signals are satisfied,
- the payload size and the usage sequence of the link resources constraints on frames are satisfied,
- the objective function with respect to extensibility or timing performance is optimized.

B. MOTIVATIONAL EXAMPLES

Let us illustrate the integrated scheduling problem using the setup from Fig. 1, where two automotive control applications are executed on the system consisting of four ESes and one NS. The corresponding security applications are depicted in Fig. 3. For simplicity, in this example we assume that the execution times of the hash function on all PEs are 0.1 ms. Although the standard TTEthernet speed is 100 Mbps or higher, in order to describe the data transmission process clearly, we consider that the link speed is only 10 Mbps. We assume that there are two time-sensitive paths, one form τ_1 to τ_5 with a deadline of 5 ms and the other from τ_2 to τ_8 with a deadline of 2.5 ms. According to (1)-(3), the interval duration P_{int} of key release is 1.25 ms.

A Straightforward solution to the security-aware scheduling problem is to (1) pack the signals generated by the same task into a frame, and (2) 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 Fig. 1, this solution is depicted by the Gantt chart in Fig. 4(a), where automotive control signals σ_1, σ_2 are packed into frame $f_{1,}\sigma_8, \sigma_9$ into $f_{2,}\sigma_3, \sigma_4$ into f_3, and $\sigma_5, \sigma_6, \sigma_7$ into f_4, key signal $\sigma_{1,7}^{key}, \sigma_{1,8}^{key}\sigma_{g,g'}^{\mathrm{k}}$,are packed into $f_{5,}\sigma_{g,g'}^{\mathrm{k}}, \sigma_{2,8}^{\mathrm{key}}$ into $f_{6,}\sigma_{7,8}^{key}$ into $f_{7,}$ and key verification tasks $\tau_{1,7}^{ver}, \tau_{1,8}^{ver}, \tau_{2,7}^{ver}, \tau_{2,8}^{ver}$ are simply denoted by $\tau_9 - \tau_{12}$. In this case, the value of the laxity of time-sensitive function paths is -0.2288 ms (the path from τ_2 to τ_8 misses its deadline) and the sum of the bandwidth consumption rates of all communication links is 0.64928.

Fig. 4(b) illustrates an optimal solution with respect to timing performance. This solution increases the laxity of paths to 3.1024 ms and satisfies the deadline constraints of both paths. On the other hand, Fig. 4(c) illustrates an optimal solution with respect to extensibility, which reduces the bandwidth consumption to 0.61664 by packing signals to frames while satisfying the deadline constraints.

C. 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.

1) DEFINITIONS

The notations of the elements and constant parameters were described in the previous definition of system model and security model. Besides, two new binary parameters Z_m^f an Z_j^σ are used to denote the type of signals σ_j and frame f_m (i.e., 1 for automotive control-related signals and frames and 0 for authentication mechanism-related signals and frames). We assume these parameters are given as design inputs. The notations of the decision variables in the MILP formulation are listed in Table 1.

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

Symbol	VARIABLE TPYE	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
$r_{m,g}^f$	binary variable	resource l_g is the receiver of frame f_m
$\mu_{m,g}^f$	binary variable	frame f_m uses resource l_g
$q_{m,g,g'}^f$	binary variable	frame f_m uses the resource l_g and $l_{g'}$ in order
$\theta_{i,i'}^{\alpha,\beta}$	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 resource l_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 resource l_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
a_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 resource l_g
$c_{m,g}^f$	real variable	the execution time of frame f_m on resource l_g
$a_{m,g}^f$	real variable	the finish time of frame f_m on resource l_g
w_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

[a] The values of the binary variables are 1 when the conditions are ture.

2) CONSTRAINTS

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

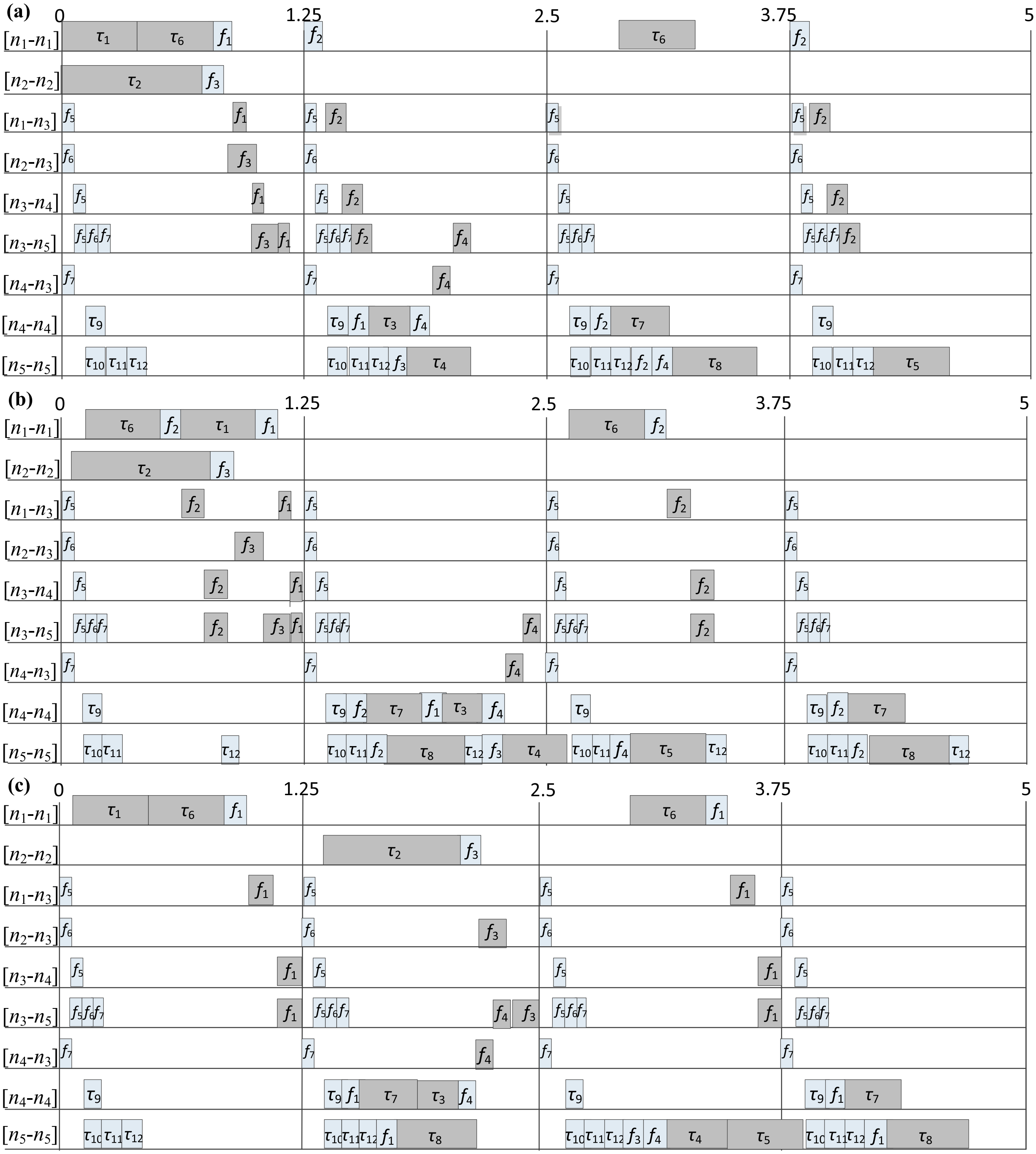


FIGURE 4. Motivational examples.

(a) *Frame Packing*

$$\forall g, \sigma_j \in \mathcal{S}_g, \quad \sum\nolimits_{f_m \in \left\{\mathcal{F}_g \,|\, Z_g^f = Z_j^\sigma\right\}} x_{j,m} = 1 \tag{4}$$

$$\forall g, \sigma_j \in \mathcal{S}_g, \quad \sum\nolimits_{f_m \in \left\{\mathcal{F}_g \,|\, Z_g^f \neq Z_j^\sigma\right\}} x_{j,m} = 0 \tag{5}$$

$$\forall j, m, \quad x_{j,m} \cdot P_m^f \leq P_j^\sigma \tag{6}$$

Equations (4) and (5) guarantee that the signal σ_j is packed into exactly one frame with the same type from the same PE link resource, where $\mathcal{S}_g$ and $\mathcal{F}_g$ denote the sets of the signals and frames from PE resource l_g, respectively. Constraint (6) guarantees that the period of a signal is greater than or equal to the period of the frame in which the signal is packed into.

$$\forall m, \forall \sigma_j, \sigma_{j'} \in \mathcal{S}_{g,h}, \quad x_{j,m} = x_{j',m} \tag{7}$$

$$\forall m, \forall \sigma_j \in \mathcal{S}_{g,h}, \quad x_{j,m} = \sum\nolimits_{\sigma_{j'} \in \mathcal{S}_{g,h}} y_{j',m} \tag{8}$$

Equation (7) ensures that each branch of a multicast signal is packed into the same frame. Equation (8) ensures that exactly one branch of a multicast signal adds its length to the frame.

$$\forall \left(\sigma_j, l_g\right) \in \mathcal{U}, \forall m, \quad x_{j,m} \le \mu^f_{m,g} \tag{9}$$

$$\forall m, g, \quad \mu^f_{m,g} \le \sum\nolimits_{(\sigma_j, l_g)\in U} x_{j,m} \tag{10}$$

$$\forall \left(\sigma_j, l_g, l_{g'}\right) \in Q, \forall m, \quad x_{j,m} \le q^f_{m,g,g'} \tag{11}$$

$$\forall m, g, g', \quad q^f_{m,g,g'} \le \sum\nolimits_{\left(\sigma_j, l_g, l_{g'}\right)\in Q} x_{j,m} \tag{12}$$

$$\forall m, j, \quad x_{j,m} \le r^f_{m,R^\sigma_j} \tag{13}$$

$$\forall m, \forall l_g \in \mathcal{L}^{PE}, \quad r^f_{m,g} \le \sum\nolimits_{\sigma_j \in \left\{\mathcal{S} \mid R^\sigma_j = l_g\right\}} x_{j,m} \tag{14}$$

Constraints (9) and (10) guarantee that a frame f_m uses the resource l_g only if there exists a signal σ_j packed into f_m and the signal uses resource l_g. Similarly, (11) and (12) guarantee that a frame f_m uses the resources l_g and $l_{g'}$ in order only if there exists a signal σ_j such that σ_j uses the two resources in order and is packed into f_m. And (13) and (14) guarantee that the PE link resource l_g is the receiver of the frame f_m only if there exists a signal σ_j packed into f_m and the receiver of the signal is l_g, where $L^{PE} \subset L$ denotes the set of PE link resources and $\mathcal{S} = \mathcal{S}^{app} \bigcup \mathcal{S}^{sec}$ denotes the set of all signals.

$$\forall m, w^f_m = OH + \sum\nolimits_{\sigma_j\in\mathcal{S}} y_{j,m} \cdot W^\sigma_j + Z^f_m \cdot W^{MAC} \tag{15}$$

$$\forall m, w^f_m \le W^f_{max} \tag{16}$$

Equation (15) calculates the total length of the frame, including the frame overhead OH, the data payload, and the MAC length (that is only contained in the automotive control frame). Constraint (16) ensures that the frame length does not exceed the limit W^f_{max} allowed by TTEthernet.

(b) *Frame scheduling*

$$\forall m, g, o^f_{m,g} + c^f_{m,g} + a^f_{m,g} \le M \cdot \mu^f_{m,g} \tag{17}$$

Constraint (17) ensures that the start time $\mathrm{o}^{\mathrm{f}}_{\mathrm{m,g}}$, execution time $c^f_{m,g}$, and finish time $a^f_{m,g}$ of a frame f_m on its unused link resource l_g are equal to zero, where M is a large constant for linearization in this paper.

$$\forall m, \forall l_g \in \mathcal{L}^{net}, \quad c^f_{m,g} \le w^f_m / V^l_g + M \cdot (1 - \mu^f_{m,g}) \tag{18}$$

$$\forall m, \forall l_g \in \mathcal{L}^{net}, \quad w^f_m / V^l_g - M \cdot \left(1 - \mu^f_{m,g}\right) \le c^f_{m,g} \tag{19}$$

$$\forall m, \forall l_g \in \mathcal{L}^{PE}, \quad c^f_{m,g} = \mu^f_{m,g} \cdot Z^f_m \cdot C^l_g \tag{20}$$

Constraints (18) and (19) state that the execution time $c^f_{m,g}$ of a frame f_m on its used network link resource l_g equals to the quotient of its length w^f_m and the configured link speed V^l_g, where $\mathcal{L}^{net} \subset \mathcal{L}$ denotes the set of network link resources.

As mentioned above, the MAC generation and verification operations of each frame are considered the executions on the PEs of its sender and receivers. Thus (20) states that the execution time $c^f_{m,g}$ of an automotive control frame f_m on its used PE link resource l_g (i.e., its sender or one of its receivers) equals to the execution time C^l_g of the selected MAC computation function on l_g. In addition, since the use of key signals is not required to have the MAC generation and verification processes, the execution time $c^f_{m,g}$ of a frame f_m that contains only key signals on its used PE l_g is equal to zero.

$$\forall m, g, \quad a^f_{m,g} = o^f_{m,g} + c^f_{m,g} \tag{21}$$

$$\forall m, g, g', \quad a^f_{m,g} - M \cdot (1 - q^f_{m,g,g'}) \le o^f_{m,g'} \tag{22}$$

Constraint (21) determines that the finish time $a^f_{m,g}$ of a frame f_m on the resource l_g equals to the sum of the start time $o^f_{m,g}$ and execution time $c^f_{m,g}$ on this resource. Constraint (22) ensures that if a frame f_m uses the resources l_g and $l_{g'}$ in order, its finish time $a^f_{m,g}$ on the resource l_g is before than its start time $o^f_{m,g'}$ on $l_{g'}$.

$$\begin{aligned}
&\forall g, \forall f_m, f_{m'}, m \neq m', \alpha \in \left\{0, \ldots, lcm\left(P^f_m, P^f_{m'}\right) / P^f_m - 1\right\}, \\
&\beta \in \{0, \ldots, lcm(P^f_m, P^f_{m'}) \Big/ P^f_{m'} - 1 \\
&\alpha \cdot P^f_m + a^f_{m,g} \le \beta \cdot P^f_{m'} + o^f_{m',g} + M \cdot \left(1 - \delta^{\alpha,\beta}_{g,m,m'}\right) \\
&+ M \cdot \left(1 - \mu^f_{m,g}\right) + M \cdot \left(1 - \mu^f_{m',g}\right)
\end{aligned} \tag{23}$$

$$\begin{aligned}
&\beta \cdot P^f_{m'} + a^f_{m',g} \le \alpha \cdot P^f_m + o^f_{m,g} + M \cdot \delta^{\alpha,\beta}_{g,m,m'} \\
&+ M \cdot \left(1 - \mu^f_{m,g}\right) + M \cdot \left(1 - \mu^f_{m',g}\right)
\end{aligned} \tag{24}$$

Constraints (23) and (24) ensure that two frames never preempt each other on any resource. The variable $\delta^{\alpha,\beta}_{g,m,m'}$ is used for switching, i.e., one of (23) or (24) is trivially satisfied depending on $\delta^{\alpha,\beta}_{g,m,m'}$.

$$\forall j, m, \quad o^\sigma_j - M \cdot (1 - x_{j,m}) \le o^f_{m,T^\sigma_j} \tag{25}$$

$$\forall j, m, \quad o^f_{m,T^\sigma_j} \le o^\sigma_j + M \cdot (1 - x_{j,m}) \tag{26}$$

$$\forall m, \forall \sigma_j \in \mathcal{S}^{app}, \quad a^\sigma_j - M \cdot \left(1 - x_{j,m}\right) \le a^f_{m,R^\sigma_j} \tag{27}$$

$$\forall m, \forall \sigma_j \in \mathcal{S}^{app}, \quad a^f_{m,R^\sigma_j} \le a^\sigma_j + M \cdot \left(1 - x_{j,m}\right) \tag{28}$$

$$\forall m, \forall \sigma_j \in \mathcal{S}^{sec}, \quad a^\sigma_j - M \cdot \left(1 - x_{j,m}\right) \le a^f_{m,N^\sigma_j} \tag{29}$$

$$\forall m, \forall \sigma_j \in \mathcal{S}^{sec}, \quad a^f_{m,N^\sigma_j} \le a^\sigma_j + M \cdot (1 - x_{j,m}) \tag{30}$$

$$\forall \sigma_j \in \mathcal{S}^{app}, \quad a^\sigma_j \le D^\sigma_j \tag{31}$$

Constraints (25) and (26) guarantee that the start time o^σ_j of a signal σ_j equals to the start time o^f_{m,T^σ_j} of the frame f_m in which the signal is packed into on its sender T^σ_j. Similarly, (27) and (28) guarantee that the finish time a^σ_j of an automotive control signal $\sigma_j \in \mathcal{S}^{app}$ equals to the finish time a^f_{m,R^σ_j} of the frame f_m in which the signal is packed into on its receiver. Given that the key signal $\sigma_j \in \mathcal{S}^{sec}$ is not required to have a MAC verification processing, (29) and (30) guarantee that its finish time a^σ_j equals to the finish time a^f_{m,N^σ_j} of the frame f_m in which it is packed into on the last network link resource N^σ_j that transmits it. And in the final of this part,

(31) specify that the finish time of each automotive control signal is within its deadline.

(c) *Tasks Scheduling*

$$\forall \tau_i \in \Gamma^{app}, \quad o_i^\tau + C_i^\tau \leq D_i^\tau \tag{32}$$

$$\forall g, \forall \tau_i, \tau_{i'} \in \Gamma_g, i \neq i', \alpha \in \left\{0, \ldots, lcm\left(P_i^\tau, P_{i'}^\tau\right)/P_i^\tau - 1\right\},$$
$$\beta \in \{0, \ldots, lcm(P_i^\tau, P_{i'}^\tau)/P_{i'}^\tau - 1\}$$
$$\alpha \cdot P_i^\tau + o_i^\tau + C_i^\tau \leq \beta \cdot P_{i'}^\tau + o_{i'}^\tau + M \cdot \left(1 - \theta_{i,i'}^{\alpha,\beta}\right) \tag{33}$$
$$\beta \cdot P_{i'}^\tau + o_{i'}^\tau + C_{i'}^\tau \leq \alpha \cdot P_i^\tau + o_i^\tau + M \cdot \theta_{i,i'}^{\alpha,\beta} \tag{34}$$

Constraint (32) determines that the finish time of a task τ_i, i.e., the sum of its start time o_i^τ and execution time C_i^τ is within its deadline. Constraints (33) and (34) ensure that two tasks never preempt each other on any PE resource, where the variable $\theta_{i,i'}^{\alpha,\beta}$ is used for switching.

(d) *Data dependency*

$$\forall \lambda_r \in \Lambda^{sec} U \Lambda^{app}, \quad (\sigma_j, \tau_i) \in G_r^{app}, \quad a_j^\sigma \leq o_i^\tau \tag{35}$$
$$\forall \lambda_r \in \Lambda^{app}, \quad (\tau_i, \sigma_j) \in G_r^{app}, \quad a_i^\tau \leq o_j^\sigma \tag{36}$$

Constraints (35) and (36) guarantee that the predecessor must complete its execution before all its successors start in an automotive control or key authentication application.

$$\forall m, \quad \varphi_m^f = \left\lceil \frac{o^f_{m,T_m^f}}{P_{int}} \right\rceil \tag{37}$$

$$\forall m, \forall l_g \in L^{net}, \quad \left\lceil \frac{a^f_{m,g}}{P_{int}} \right\rceil \leq \varphi_m^f \tag{38}$$

According to the authentication mechanism, an automotive control frame is accepted and stored awaiting to be authenticcated 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, (37) first computes the number of the time interval φ_m^f in which its MAC key is released. And then, (38) guarantees that the start time of a frame on its sender and the finish time of the frame on any one of the used network link resource belong to the same time interval.

$$\forall g, f_m \in \left\{\mathcal{F}_g \,|\, Z_g^f = 1\right\}, \forall \tau_i \in \left\{\Gamma_g^{sec} \,|\, V_i^\tau = T_m^f\right\},$$
$$\underset{om,g}{f} \geq a_i^\tau + \varphi_m^f \cdot P_{int} - M \cdot \left(1 - r^f_{m,g}\right) \tag{39}$$

Besides, an automotive control frame will be available to its receiver after the verification task for its MAC key is completed. Therefore, (39) guarantee that the start time of the verification operation of a frame $f_m \in \left\{F_g \,|\, Z_m^f = 1\right\}$ on its receiver must 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 PE l_g.

$$\forall l_g \in \mathcal{L}^{PE}, f_m \in \left\{F_g \,|\, Z_m^f = 1\right\}, \quad \forall \tau_i \in \Gamma_g,$$
$$\alpha \in \{0, \ldots, \frac{lcm\left(P_i^\tau, P_m^f\right)}{P_i^\tau} - 1, \},$$
$$\beta \{\in 0, \ldots, \frac{lcm\left(P_i^\tau, P_m^f\right)}{P_m^f} - 1\}$$
$$\alpha \cdot P_i^\tau + o_i^\tau + C_i^\tau \leq \beta \cdot P_m^f + o^f_{m,g} + M \cdot \left(1 - \eta^{\alpha,\beta}_{g,i,m}\right)$$
$$+ M \cdot \left(1 - \mu^f_{m,g}\right) \tag{40}$$
$$\beta \cdot P_m^f + o^f_{m,g} + c^f_{m,g} \leq \alpha \cdot P_i^\tau + o_i^\tau + M \cdot \eta^{\alpha,\beta}_{g,i,m}$$
$$+ M \cdot \left(1 - \mu^f_{m,g}\right) \tag{41}$$

Constraints (40) and (41) ensure that the frame verification tasks and other tasks never preempt each other on any PE resource during execution, where the variable $\eta^{\alpha,\beta}_{g,i,m}$ is used for switching.

(e) *End-to-end latency*

$$\forall \rho_\varepsilon \in \mathcal{FP}^{app}, a^\tau_{Des^\rho_\varepsilon} - o^\tau_{Src^\rho_\varepsilon} \leq D^\rho_\varepsilon \tag{42}$$

Constraint (42) ensure that the end-to-end delay must less than the deadline for each function path, where Des^ρ_ε and Src^ρ_ε are the source and sink object of the function path $\rho_\varepsilon \in \mathcal{FP}^{app}$, respectively.

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:

$$\text{Maximize} \sum\nolimits_{\forall \rho_\varepsilon \in \mathcal{FP}^{app}} D^\rho_\varepsilon + o^\tau_{Src^\rho_\varepsilon} - a^\tau_{Des^\rho_\varepsilon} \tag{43}$$

We can alternatively minimize the consumption of network bandwidth, therefore improving extensibility:

$$\text{Minimize} \sum\nolimits_{\forall\, \mathrm{f_{m\in}\, f}} \sum\nolimits_{\mathrm{l_g \in L^{net}}} \mathrm{c^f_{m,g}/P^f_m} \tag{44}$$

VI. 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 TTEthernet-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 consider the Infineon TriCore, a widely used automotive 32-bit microcontroller, as a representative platform. A MAC generation/verification operation takes 11 μ s on Tricore at 180 MHz [31].

TABLE 2. Tasks of the advanced automotive control system.

Task	ES	Period	Execution time	Task	ES	Period	Execution time
τ_1	n_1	8000	150	τ_{13}	n_1	4000	150
τ_2	n_1	8000	175	τ_{14}	n_2	4000	200
τ_3	n_2	8000	300	τ_{15}	n_5	4000	200
τ_4	n_1	8000	250	τ_{16}	n_3	4000	200
τ_5	n_2	8000	150	τ_{17}	n_5	4000	200
τ_6	n_1	8000	100	τ_{18}	n_3	4000	200
τ_7	n_4	4000	300	τ_{19}	n_1	4000	150
τ_8	n_3	4000	150	τ_{20}	n_3	4000	300
τ_9	n_1	4000	175	τ_{21}	n_6	4000	175
τ_{10}	n_3	4000	300	τ_{22}	n_2	4000	400
τ_{11}	n_1	4000	250	τ_{23}	n_2	4000	150
τ_{12}	n_2	4000	200	τ_{24}	n_1	4000	200

TABLE 3. Signals of the advanced automotive control system.

Signal	Send	Receive	Size	Signal	Send	Receive	Size
σ_1/σ_2	τ_1	τ_3/τ_4	12	σ_{12}	τ_{12}	τ_{14}	10
σ_3	τ_2	τ_4	12	σ_{13}	τ_{15}	τ_{20}	12
σ_4	τ_3	τ_5	8	σ_{14}	τ_{16}	τ_{20}	12
σ_5	τ_4	τ_6	12	σ_{15}	τ_{17}	τ_{20}	12
σ_6	τ_7	τ_{10}	12	σ_{16}	τ_{18}	τ_{20}	12
σ_7	τ_8	τ_{10}	12	σ_{17}	τ_{19}	τ_{22}	10
σ_8	τ_9	τ_{11}	10	σ_{18}	τ_{20}	τ_{22}	10
σ_9/σ_{10}	τ_{10}	τ_{11}/τ_{12}	12	σ_{19}	τ_{21}	τ_{22}	8
σ_{11}	τ_{11}	τ_{13}	10	σ_{20}/σ_{21}	τ_{22}	τ_{23}/τ_{24}	12

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 (1) pack the signals generated by the same task into a frame, and (2) 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.

A. TYPES OF GRAPHICS CASE STUDY: AN ADVANCED CONTROL SYSTEM

We consider a case study from the literature [32], a set of advanced automotive control applications including adaptive cruise control (ACC), electric power steering (EPS), and traction control (TC). There are 24 tasks and 18 signals, 3 of which are multicast signals.

Tables 2 and 3 show the periods and the worst-case execution time of tasks (in microseconds) and the sizes of Signals (in bits). The hardware platform consists of 6 ESes connected via a switched Ethernet network. The speeds of the communication links are 100 Mbps.

TABLE 4. Results of the advanced automotive control system.

Approach	Timing performance	Extensibility
MILP-S	21.72652	0.10416
ASAP-NS	23.1234	0.03024
MILP-NS	26.0906	0.02184

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 authentica-tion mechanism overheads and constraints.

Specifically, 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. These demonstrate that the introduction of authentication function hardly affects the timing performance of systems after using the proposed MILP-S. On the other hand, when the extensibility is taken as optimization objective, the bandwidth used by MILP-S is greater than that of the non-security-aware scheduling optimization approaches MILP-NS and ASAP-NS. This is because that each sender needs to transmit the released key in each time interval after using the TESLA authentication mechanism, and consumes more bandwidth. Even so, for safety-critical automotive systems, a small portion of their bandwidth resources is still worth achieving security.

B. SCALABILITY ANALYSIS

1) SYSTEM CONFIGURATIONS

To assess the scalability of the proposed approach, we generated a set of synthetic applications and network topologies based on realistic automotive system cases. Specifically, the period of tasks and signals are varied among the range [5, 10, and 20 ms]. The average ratio of deadline to period of each time-sensitive function path is 0.7. The speeds of the communication links are set to 100 Mbps. Two broad classes of experiments are conducted as follows.

(a) First, we evaluate the performance of the proposed scheduling approach on different system scales. In the figures of results, the horizontal axis marks the number of PEs, which denotes the scale of the experiments, as the number of PEs and the number of tasks and signals simultaneously grow. The number of PEs on the horizontal axis is varied among the range [4], [8], [16], considering that a typical distributed automotive system such as infotainment or chassis is composed of less than 15 PEs. When the number of PEs is 16, the number of automotive applications, time-sensitive function paths, tasks and signals are 18, 36, 65 and 190, respectively. The average cost of a task is 2 ms and the average size of a signal is 32 bytes.

(b) Second, we evaluate the performance of the proposed scheduling approach on systems with different numbers of time-sensitive function paths. We increase the number of time-sensitive function paths while keeping the same hardware architecture. Specifically, the number of time-sensitive function paths is varied among the range [5], [10], [11], [24]. The number of PEs is 8. The average cost of a task is 0.42 ms and the average size of a signal is 39 bytes.

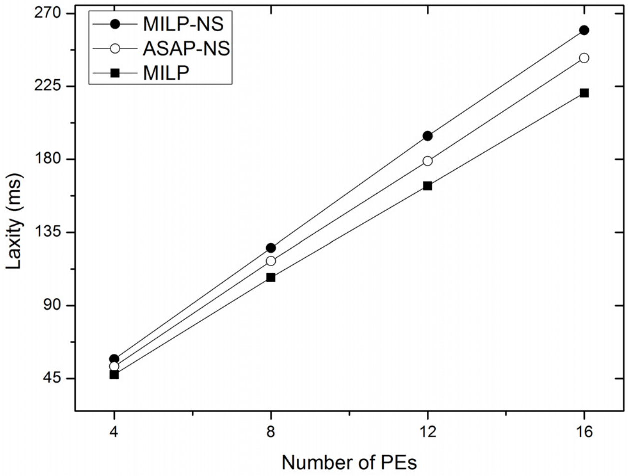


FIGURE 5. Laxities of the scheduling approaches versus number of PEs.

2) RESULTS AND DISSCUSSION

a: INCREASED SCALES OF SYSTEM

Fig. 5 depicts the comparison results of MILP-S, MILP-NS and ASAP-NS with respect to laxity metric function on different system scales. It is shown that MILP-S can guarantee the schedulability of systems with authentication mechanism overheads and constraints in all experiments. In addition, the laxity of all time-sensitive function paths obtained by MILP-S is average 15% lower than that of MILP-NS, and only 9% lower than that of ASAP-NS. These demonstrate that the introduction of authentication function hardly affects the timing performance of systems after using the proposed MILP-S.

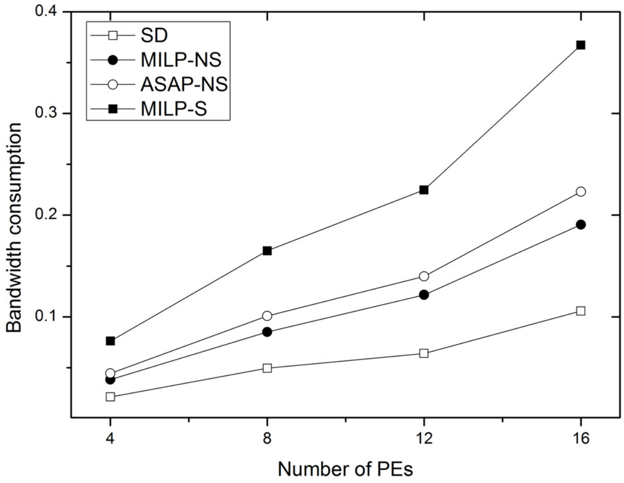


FIGURE 6. Bandwidth consumption ratios of the scheduling approaches versus number of PEs.

Fig. 6 depicts the comparison results of MILP-S, MILP-NS and ASAP-NS with respect to extensibility metric function on different system scales. The SD is the total fraction of the network bandwidth that is required by all signals. It can be calculated by $SD = \sum_{\sigma_j \in \mathcal{S}^{app}} \sum_{l_g \in \{L^{net} | (\sigma_j, l_g) \in \mathcal{U}\}} W_j^{\sigma} / (V_g^{l} \cdot P_j^{\sigma})$. First, it is shown that MILP-S returns the biggest bandwidth consumption that is approximately 3.4 times SD; and the bandwidth consumptions of ASAP-NS and MILP-NS are 1.9 times and 1.6 times SD, respectively. Moreover, as the number of PEs increases, the differences in bandwidth consumption between the security-aware MILP-S and non-security-aware MILP-NS and ASAP-NS grow slightly. This is because as the number of PEs increases, the authentication function requires more bandwidth resource to transmit the keys they release.

Furthermore, Fig. 7 shows the runtime of the MILP solver for each of these experiments. In our experiments, we set a 3600 s time limit. For both optimization metrics, the solver is able to find the optimal solution within the time limit when the number of PEs is less than 16; and return a feasible solution when the number of PEs is 16.

b: INCREASED NUMBER OF FUNCTION PATHS

Fig. 8 depicts the comparison results of MILP-S, MILP-NS and ASAP-NS with respect to latency metric function on systems with different numbers of time-sensitive function paths. First, it is shown that MILP-S can still guarantee the schedulability of systems with authentication mechanism overheads and constraints. Second, the laxity of all function

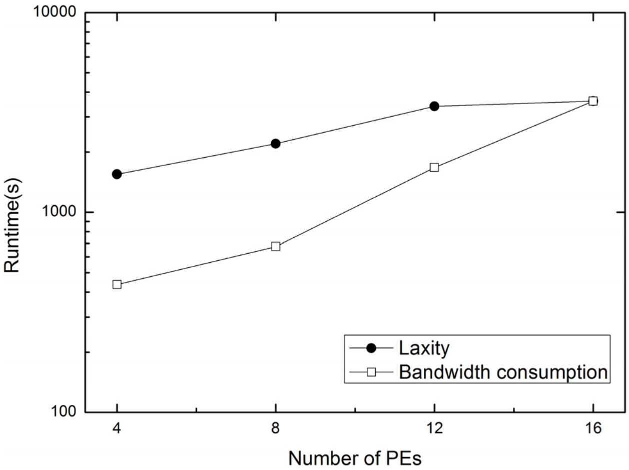


FIGURE 7. Runtime of the MILP solver versus number of PEs.

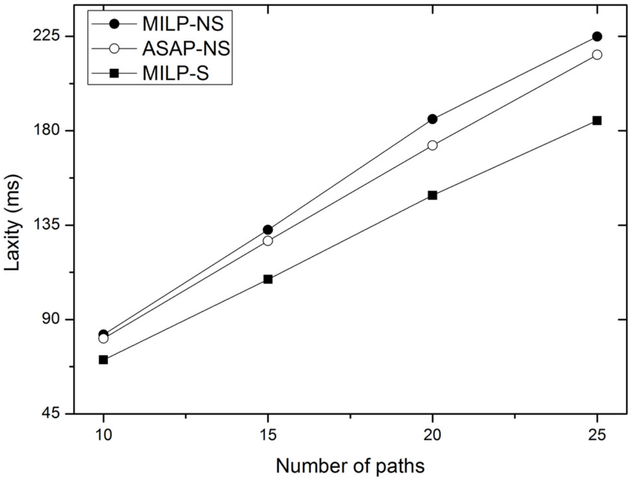


FIGURE 8. Laxities of the scheduling approaches versus number of paths.

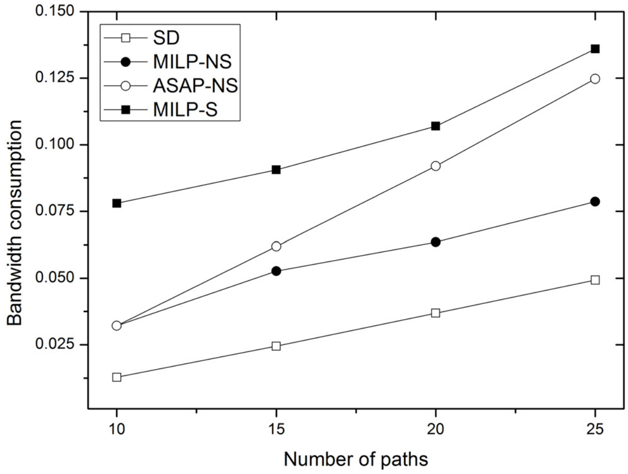


FIGURE 9. Bandwidth consumption ratios of the scheduling approaches versus on number of paths.

paths obtained by MILP-S is average 17% lower than that of MILP-NS, and only 12% lower than that of ASAP-NS.

Similarly, Fig. 9 depicts the comparison results of MILP-S, MILP-NS and ASAP-NS with respect to extensibility metric

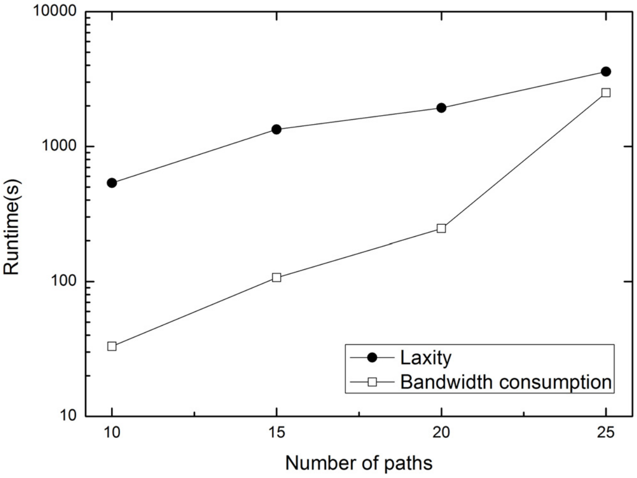


FIGURE 10. Runtime of the MILP solver versus number of paths.

TABLE 5. The notations of elements and sets.

Symbol	Implication
τ_i	the i-th task
σ_j	the j-th signal
f_m	the m-th frame
l_g	the g-th link resource
λ_r	the r-th application
ρ_ε	the ε-th path
Src_ε^ρ	the source task of path ρ_ε
Des_ε^ρ	the sink task of path ρ_ε
Γ	the set of tasks
Γ^{app}	the set of automotive control-related tasks
Γ_g	the set of tasks executed on resource l_g
Γ_g^{sec}	the set of the key verification tasks executed on resource l_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 l_g
$\mathcal{S}_g$	the set of the signals from resource l_g
$\mathcal{F}$	the set of frames
$\mathcal{F}_g$	the set of the signals from resource l_g
$\mathcal{L}$	the set of link resources
$\mathcal{L}^{PE}$	the set of PE link resources
$\mathcal{L}^{net}$	the set of network link resources
$\varLambda^{app}$	the set of automotive control applications
$\varLambda^{sex}$	
	the set of key authentication applications
$\mathcal{FP}^{app}$	the set of time-sensitive function paths

function on systems with different numbers of time-sensitive function paths. It is shown that MILP-S returns the biggest bandwidth consumption that is approximately 3.3 times the SD. And the bandwidth consumptions of ASAP-NS and MILP-NS are 1.8 times and 1.3 times SD, respectively. In addition, as the number of paths increases, the differences in bandwidth consumption between the security-aware MILP-S and non-security- aware MILP-NS and ASAP-NS decrease. This is because when the number of function paths grows and the number of PEs remains constant, each PE

TABLE 6. **The notations of constant parameters.**

Symbol	Implication
C_i^τ	the execution time of task τ_i
P_i^τ	the period of task τ_i
D_i^τ	the deadline of task τ_i needs to verify
V_i^τ	the PE link resource that key verification task τ_i needs to verify
T_j^σ	the PE link resource that sends signal σ_j
R_j^σ	the PE link resource that receives signal σ_j
N_j^σ	the last network link resource that transmit key signal σ_j
W_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 PE link resource that sends frame f_m
P_m^f	the period of frame f_m
Z_m^f	the type of frame f_m
C_g^l	the execution time of the MAC computation function on PE link resource l_g
V_g^l	the speed of network link resource l_g
D_ε^ρ	the deadline of path ρ_ε
W_{max}^f	the upper limit of frame length
W^{MAC}	the length of MAC
OH	the frame overhead
P_{int}	the interval duration of key release
M	a large constant for linearization

produces more signals, thus providing more possibility of optimization of frame packing.

Fig. 10 shows the runtime of the MILP solver for each experiment. For both optimization metrics, the solver can find the optimal solution within the time limit in all cases. These demonstrate the effectiveness and efficiency of the proposed approach.

VII. CONCLUSION

In this paper, we have proposed an approach to address both the information security and functional safety in the scheduling design of TTEthernet-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 optimiza-tion problem of TTEthernet-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 bandwidth consumption (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. In future work, we plan to implement encryption mechanism on TTEthernet for protecting the data confidentiality. Meanwhile, we will extend our optimization framework to include all cryptographic operations of the encryption mechanism.

APPENDIX

The list of symbols and constant parameters in the MILP formulation are summarized in Tables 5 and 6 respectively.

REFERENCES

[1] *Time-Triggered Ethernet*, document SAE AS6802, 2011.

[2] *Avionics Full-Duplex Switched Ethernet Network*, document ARINC 664P7, 2009.

[3] *IEEE Standard for Ethernet—Amendment 1:Physical Layer Specifications and Management Parameters for 2.5 Gb/s and 5 Gb/s Operation over Backplane*, IEEE Standard 802.3cb-2018, 2012.

[4] K. Mueller, T. Steinbach, F. Korf, and T. C. Schmidt, "A real-time Ethernet prototype platform for automotive applications," in *Proc. IEEE-ICCE*, Berlin, Germany, Sep. 2011, pp. 221–225.

[5] L. L. Bello, "The case for Ethernet in automotive communications," *ACM SIGBED Rev.*, vol. 8, no. 4, pp. 7–15, Dec. 2011.

[6] T. Steinbach, H. T. Lim, F. Korf, T. C. Schmidt, D. Herrscher, and A. Wolisz, "Tomorrow's In-Car interconnect? A competitive evaluation of IEEE 802.1 AVB and time-triggered Ethernet (AS6802)," in *Proc. IEEE-VTC*, Sep. 2012, pp. 1–5.

[7] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy*, vol. 41, no. 3, May 2010, pp. 447–462.

[8] 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 Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, Sep. 2015.

[9] S. Ivan, V. Nicomette, E. Alalta, Y. Deswarte, M. Kaâniche, and Y. Laarouchi, "Survey on security threats and protection mechanisms in embedded automotive networks," in *Proc. DSN-W*, Jun. 2013, pp. 1–12.

[10] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures," *Rel. Eng. Syst. Saf.*, vol. 96, no. 1, pp. 11–25, Jan. 2011.

[11] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. NDSS*, Feb. 2001, pp. 35–46.

[12] L. Lamport, "Constructing digital signatures from one-way function," Tech. Rep. SRI-CSL-98, 1979. [Online]. Available: https://www.microsoft.com/en-us/research/publication/constructing-digital-signatures-one-way-function/

[13] D. Bleichenbacher and U. M. Maurer, "Directed acyclic graphs, one-way functions and digital signatures," in *Proc. Annu. Int. Cryptol. Conf.*, vol. 94, 1994, pp. 75–82.

[14] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in *Proc.CCS*, Nov. 2001, pp. 28–37.

[15] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, "Time valid one-time signature for time-critical multicast data authentication," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr. 2009, pp. 1233–1241.

[16] C. Szilagyi and P. Koopman, "Flexible multicast authentication for time-triggered embedded control network applications," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jul. 2009, pp. 165–174.

[17] C. Szliagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered embedded control networks," in *Proc. WESS*, Scottsdale, AZ, USA, Oct. 2010, pp. 1–10.

[18] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Proc. 10th Ann. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, May 2003, pp. 263–276.

[19] D. Liu and P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 4, pp. 800–836, Nov. 2004.

[20] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in *Proc. MobiCom*, Rome, Italy, Sep. 2001, pp. 189–199.

[21] A. Perrig, R. Canetti, J. Tygar, and D. X. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proc. IEEE Symp. Secur. Privacy*, Berkeley, CA, USA, May 2000, pp. 56–73.

[22] W. Steiner, "An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks," in *Proc. RTSS*, San Diego, CA, USA, Dec. 2010, pp. 375–384.

[23] W. Steiner, "Synthesis of static communication schedules for mixed-criticality systems," in *Proc. ISORCW*, Newport Beach, CA, USA, May 2011, pp. 11–18.

[24] E. Suethanuwong, "Scheduling time-triggered traffic in TTEthernet systems," in *Proc. ETFA*, Apr. 2012, pp. 1–4.

[25] D. Tămaş-Selicean, P. Pop, and W. Steiner, "Synthesis of communication schedules for TTEthernet-based mixed-criticality systems," in *Proc. ISORCW*, May 2012, pp. 473–482.

[26] D. Tămaş-Selicean, P. Pop, and W. Steiner, "Design optimization of TTEthernet-based distributed real-time systems," *Real-Time Syst*, vol. 51, no. 1, pp. 1–35, 2015.

[27] J. Dvořák, M. Heller, and Z. Hanzálek, "Makespan minimization of time-triggered traffic on a TTEthernet network," in *Proc. IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Jun. 2017, pp. 1–10.

[28] L. C. Zhang, D. Goswami, R. Schneider, and S. Chakraborty, "Task- and network-level schedule co-synthesis of Ethernet-based time-triggered systems," in *Proc. ASP-DAC*, Singapore, Jan. 2014, pp. 119–124.

[29] S. S. Craciunas and R. S. Oliver, "Combined task- and network-level scheduling for distributed time-triggered systems," *Real-Time Syst*, vol. 52, no. 2, pp. 161–200, 2016.

[30] M. Abuteir and R. Obermaisser, "Scheduling of rate-constrained and time-triggered traffic in multi-cluster TTEthernet systems," in *Proc. IEEE Int. Conf. Ind. Inform.*, Jul. 2015, pp. 239–245.

[31] B. Groza and S. Murvay, "Scheduling of rate-constrained and time-triggered traffic in multi-cluster TTEthernet systems," *IEEE Trans. Ind. Inform.*, vol. 9, no. 4, pp. 2034–2042, Apr. 2013.

[32] N. Kandasamya, J. P. Hayesb, and B. T. Murrayc, "Dependable communication synthesis for distributed embedded systems," *Rel. Eng. Syst. Saf.*, vol. 89, no. 1, pp. 81–92, 2015.

RUI ZHAO received the B.D. degree in computer science and technology from Northeast Normal University, Changchun, China, in 2009, and the B.S. and Ph.D. degrees in computer science and technology from Jilin University, Changchun, in 2011 and 2017, respectively.

Since 2017, she has been an Assistant Research Fellow with the College of Computer Science and Technology, Jilin University. Her research interests include the system-level scheduling design of real-time embedded systems and in-vehicle communication networks.

GUIHE QIN was born in Gaomi, Shandong, China, in 1962. He received the B.S. degree in computer science and technology and the Ph.D. degree in electronic and communication system from Jilin University, Changchun, China, in 1988 and 1997, respectively.

From 1988 to 1991, he was a Lecturer with the School of Computer Science and Technology, Jilin University, where he was an Assistant Professor, from 1996 to 2001, and has been a Professor, since 2001. He is the author of three books, more than 100 articles, and more than 30 inventions. His research interests include embedded systems and computer vision.

Dr. Qin's awards and honors include the First Award of Science and Technology Progress of the Trade Unions, the First Prize of Jilin Technology Progress, and the Second Prize of Jilin Technology Progress.

YING LYU was born in Hebei, China. She received the M.S. degree in computer technology from the Hebei University of Technology, Tianjin, China, in 2006. She is applying for Ph.D. degree in vehicle engineering at Jilin University, Changchun, China.

She is currently engaged in system design of automated driving in ICV Department, China First Automotive Works Group. Her research interests include the advanced automotive control systems, parallel and distributed systems, and in-vehicle networks.

JIE YAN was born in Jilin, China. She received the M.S. degree in computer science and technology from Jilin University, Changchun, China, in 2018. She is applying for Ph.D. degree in computer science and technology at Jilin University, Changchun, China.

Her research interests include computer vision and image processing.