Optimization of Message Encryption for Real-Time Applications in Embedded Systems

Amir Aminifar[1], Petru Eles[2], Zebo Peng[2]

[1]Embedded Systems Laboratory, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland

[2]Embedded Systems Laboratory, Linköping University, Sweden

Abstract—Today, security can no longer be treated as a secondary issue in embedded and cyber-physical systems. Therefore, one of the main challenges in these domains is the design of secure embedded systems under stringent resource constraints and real-time requirements. However, there exists an inherent trade-off between the security protection provided and the amount of resources allocated for this purpose. That is, the more the amount of resources used for security, the higher the security, but the fewer the number of applications which can be run on the platform and meet their timing requirements. This trade-off is of high importance since embedded systems are often highly resource constrained. In this paper, we propose an efficient solution to maximize confidentiality, while also guaranteeing the timing requirements of real-time applications on shared platforms.

Index Terms—Secure Embedded Systems, Secure Real-Time Systems, Real-Time Schedulability, Security and Privacy, Confidentiality Optimization, Message Encryption and Decryption.

1 Introduction and Related Work

TODAY, many embedded and cyber-physical systems are tightly interacting with external networks and components. Such interactions, in turn, make these systems more vulnerable to security attacks and intrusions. This is in particular important since many such systems comprise several critical components, e.g., adaptive cruise control or engine control in automotive systems allow deep intervention in vehicles. Therefore, ignoring the security aspects in cyber-physical systems will have severe consequences. As a result, security is now considered one of the main design challenges in embedded and cyber-physical systems [1]–[5].

The design of secure embedded and cyber-physical systems becomes all the more complex if we consider shared resources, for example as a result of the recent shift in automotive industry towards integrated architectures [6]. This resource sharing, in turn, leads to complex timing behaviors and competition among real-time and control applications for available resources on shared platforms (see Figure 1) [7], [8]. Moreover, there exists an inherent trade-off between security strength and available resources [9]–[11]. That is, the higher the security requirements, the more resources should be alloted for that purpose.

Confidentiality is among the key components of embedded security and has been discussed previously in the context of embedded and real-time systems [12]–[17]. Confidentiality refers to the ability to keep the information secret from unauthorized entities and is often achieved by message encryption/decryption. However, the available resources in many embedded systems, e.g., in the automotive domain [3], are limited, and this resource scarcity highly limits the ability to perform message encryption/decryption for real-time functions. Therefore, the main idea in this paper is to maximize confidentiality, while guaranteeing the timing requirements of all applications on shared platforms.

To optimize confidentiality, Kang and Son [12] propose a lightweight heuristic to optimize the cryptographic key length under real-time constraints. Xie and Qin [13] propose an online algorithm to schedule periodic tasks under security and timing constraints, trying to maximize a weighted sum of the individual security values, a metric defined by the authors. However, their approach does not provide any guarantees with respect to optimality of the total security value. Similarly, in [14], the authors perform an offline optimization for the security of real-time systems. The authors propose two approaches based on integer linear programming and a heuristic. The former suffers from exponential time complexity and the latter does not provide any kind of optimality guarantees.

More recently, Jiang et al. [15], [16] used constraint logic programming to optimize message encryption under real-time constraints. While constraint logic programming provides the optimal solution, due to its exponential time complexity, the authors propose an efficient heuristic. In [17], the authors propose a quasi-static heuristic to optimize quality of service under security and schedulability constraints for dynamic task sets. Their approach is limited to a multi-mode scenario, where all modes are known at design time. Nevertheless, the proposed methodologies in [15]–[17] can only provide suboptimal solutions and the quality of these solutions cannot be quantified.

The existing work in the literature falls under (at least) one of the following categories: (1) those approaches based on integer linear programming [14] and constraint logic programming [15]–[17], which suffer from computation complexity and cannot be used in an online and dynamic setting, and (2) those approaches based on heuristics [12]–[17], which do not provide any optimality or performance guarantees and the solution can be arbitrarily far from the actual optimal.

In this article, we approach the problem of maximizing confidentiality under real-time constraints in dynamic settings, where the applications requirements are subject to change at runtime. This change, for instance, could be due to a multi-mode mechanism where the active task set (and messages) changes at runtime to achieve better performance [17], [18]. Alternatively, this could be due to a change in resource requirements of applications either because of the state-dependency [19]–[21], or in response to an attack [22]. As a concrete example, in the automotive domain, the activation rates of several tasks (e.g., of the crankshaft, gears, or wheels) are proportional to the angular velocity of a specific

component [21]. Hence, we need an efficient online algorithm for confidentiality optimization in such dynamic settings, under timing constraints.

In this article, we first identify the optimal amount of time to allocate to achieving security of each message (encryption and decryption) under real-time constraints analytically. Built on top of this analytical solution, we propose a fully-polynomial time algorithm that provides the optimal solution for the concrete case of iterated block ciphers, RC5 [9] and, its extension, RC6 [23], which is considered to have adequate security for the Advanced Encryption Standard (AES) [24]. In summary, here, we maximize the minimum confidentiality strength in the system, given the available resources, while guaranteeing real-time schedulability.

2 System Model

2.1 Platform Model

We consider a uniprocessor platform (see Figure 1), with several real-time applications running on this platform. Each circle in Figure 1 corresponds to a task which can send/receive messages via the communication infrastructure (handled by the communication controller (CC) unit). To protect confidentiality, each task encrypts the message before sending it over the communication infrastructure and decrypts the received messages.

2.2 Intrusion Detection and Resource Management Task

In this paper, we assume that there exists a host-based intrusion detection and resource management (IDRM) task [25] on each processing node, similar to [22]. This task monitors the workload variations and application requirements and decides on the amount of resources which should be allocated to achieve confidentiality for each message. This variation could be due to a change in the active task set in multi-mode systems [17], [18], a change in resource requirements of applications [19]–[21], or a change in security requirements of applications due to intrusion [22].

The optimization problem discussed in this paper is performed as part of the duties of the host-based intrusion detection and resource management (IDRM) task.

2.3 Real-Time Application Model

Given is a set of m real-time tasks running on a node. Each task τ_j has a worst-case computation time (execution time) of c_j and a minimum inter-arrival time of h_j. We assume implicit deadline for all tasks, i.e., the deadline d_j is equal to the minimum inter-arrival time h_j of task τ_j, i.e., $d_j = h_j$.

We highlight that the intrusion detection and resource management (IDRM) task is also modeled as a sporadic (implicit-deadline) task τ_0, with worst-case execution time overhead c_0 and minimum inter-arrival time h_0.

Each task τ_j sends and receives n_j messages $\gamma_{j,k}$, where $k = 1, \ldots, n_j$. The total number of messages sent and received is $n = \sum_{j=0}^{m} n_j$. We assign to each message $\gamma_{j,k}$ a unique identifier i and denote this message and its corresponding frequency by γ_i and f_i, where $i = \sum_{x=0}^{j-1} n_x + k$ and $f_i = \frac{1}{h_j}$. In other words, the send/receive frequency for each message is the inverse of the minimum inter-arrival time of its corresponding real-time task.

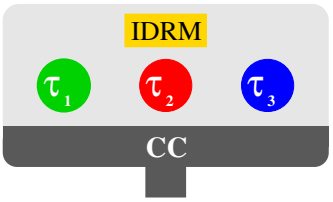

Fig. 1. A processing node hosting three real-time tasks and the intrusion detection and resource management (IDRM) task. These tasks may send/receive messages via the communication network, handled by the communication controller (CC) unit.

3 Preliminaries and Background

In this section, we shall give a brief overview to real-time scheduling on a uniprocessor platform. We, further, discuss the confidentiality notion in the security area.

3.1 Real-Time Scheduling

Today, the platform in the majority of embedded systems, e.g., in the automotive domain [6], is shared among several applications. These applications, therefore, compete for execution on the platform. For safety-critical applications, it is necessary to demonstrate the safety of the system on the shared platform. In the case of real-time applications, this safety implies the notion of schedulability. In short, under a certain scheduling policy, a task set is referred to as schedulable if all tasks of the task set are schedulable. A task is schedulable under a scheduling policy, if the task is guaranteed to meet its deadline.

In this paper, we discuss schedulability analysis under the earliest-deadline-first (EDF) scheduling policy. Under the earliest-deadline-first (EDF) scheduling, in case of competing tasks, the task which has the earliest deadline has the highest priority and executes. Given an independent set of real-time tasks and assuming implicit deadline for each task, it has been proven that a task set is schedulable under the earliest-deadline-first (EDF) scheduling policy if and only if the task set utilization does not exceed one [26], i.e.,

$$\sum_{j=0}^{m} \frac{c_j}{h_j} \leq 1. \tag{1}$$

Note that the above condition is a necessary and sufficient schedulability condition under the earliest-deadline-first (EDF) scheduling policy.[1]

3.2 Confidentiality

Confidentiality is considered to be one of the main components of information security. It refers to the ability to keep the information secret from unauthorized entities and is often achieved by message encryption/decryption. The basic idea is that the more the time and resources are used, the more the efforts to break this encryption, and the higher the confidentiality.

The strength of a cryptography algorithm is typically quantified by the use of cryptanalysis attacks, which measure the strength of the cryptography algorithm by estimating the number of plaintexts needed to successfully mount an attack on the cryptography algorithm. The fundamental principle of cryptography is that *the*

1. Generalization of our proposed approach to the rate-monotonic (RM) scheduling policy is also straightforward, by adapting the right-hand side to 0.69 [26].

TABLE 1
An estimate of the number of plaintexts required to mount an attack on RC5 and RC6 with varying number of rounds can be used to quantify security strength [27], [28].

	$r=8$	$r=12$	$r=16$	$r=20$	$r=24$
Linear attack RC5 [27]	2^{52}	2^{76}	2^{100}	2^{124}	2^{148}
Linear attack RC6 [28]	2^{62}	2^{102}	2^{142}	2^{182}	2^{222}

time and resources required for the attacker to successfully mount a cryptanalysis attack on the system should be considerably (often, exponentially) more than the time and resources required for encryption/decryption.

Let us consider the concrete case of the iterated block ciphers class, and focus on RC5 [9] and, its extension, RC6 [23]. RC6 is considered to have adequate security for the Advanced Encryption Standard (AES) and is able to provide high performance (encryption/decryption speed) on embedded processors [24].

RC6 can provide high security when suitable parameter values are chosen [9]. Hence the importance of parameter optimization. The parameter we are mostly interested in is the number of rounds these algorithms use for encryption. The larger the number of rounds, the more the efforts needed for the adversary, and the higher the confidentiality. More concretely, as shown in Table 1, the number of plaintexts needed to break the encryption is exponential with the number of rounds used for encryption [28]. Therefore, we use the following exponential function for quantifying security strength,

$$e^{\alpha_i r_i + \omega_i}, \tag{2}$$

where r_i denotes the number of rounds. Coefficients α_i and ω_i are constants and depend on the importance of each application (determined by the system designer) as well as the word size [27] and the length of the encryption key [23], [28]. Motivated by the results in Table 1 [27]–[29], and as shown graphically in Figure 2, the exponent $(\alpha_i r_i + \omega_i)$ is modeled as a linear function of r_i.

Introducing message encryption/decryption also affects the schedulability of the system. The safety of the system can be jeopardized, i.e., the system can become unschedulable, if the impact of message encryption/decryption is not considered properly. To address this issue, we can simply consider that the execution time c_j of each task τ_j is increased by the amount of time spent for security purposes. Note that the time complexity of encryption/decryption in RC5 and RC6 algorithms is linear with the number of rounds [9], [23]. Figure 3 shows the worst-case execution times of the encryption and decryption algorithms in RC6 versus the number of rounds on ARM® Cortex®-M3 processor, based on the aiT execution-time analysis tool [30], [31]. Figure 3 clearly demonstrates the linearity of the worst-case execution times of the encryption and decryption algorithms of RC6 with respect to the number of rounds. Hence, we can adapt the schedulability test as follows,

$$\sum_{j=0}^{m} \frac{c_j + \sum_{k=1}^{n_j} o_{j,k} r_{j,k}}{h_j} \leq 1, \tag{3}$$

where $r_{j,k}$ and $o_{j,k}$ denote the number of rounds and the timing overhead of each round for message $\gamma_{j,k}$, respectively. This inequality can be simplified as follows,

$$\sum_{i=1}^{n} o_i r_i f_i \leq 1 - \sum_{j=0}^{m} \frac{c_j}{h_j} = U, \tag{4}$$

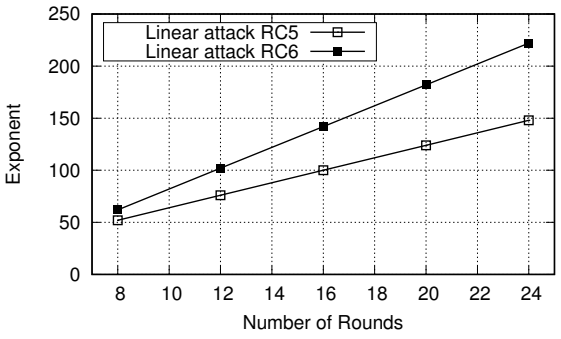

Fig. 2. The exponent of the exponential changes linearly with the number of rounds [27], [28].

by assigning to each message $\gamma_{j,k}$ a unique identifier $i = \sum_{x=0}^{j-1} n_x + k$ and denoting its corresponding frequency by $f_i = \frac{1}{h_j}$, as discussed in Section 2.3. Note that U is the part of the resource utilization that can be afforded for message encryption/decryption, without jeopardizing the schedulability of the system.

Observe that increasing r_i leads to an increase in the security strength based on Equation (2), while it reduces the schedulability margin in Equation (4). Hence the trade-off between the security strength and the amount of resources used for that purpose.

In general, to quantify the security strength, relative to a given message, an exponential function, with respect to time, can be used,

$$e^{\alpha_i t_i + \omega_i}, \tag{5}$$

where the amount of time spent for security is denoted by t_i. Similar to the case of RC6, the schedulability condition can also be adapted as follows,

$$\sum_{i=1}^{n} t_i f_i \leq 1 - \sum_{j=0}^{m} \frac{c_j}{h_j} = U. \tag{6}$$

4 Motivational Example

In this section, we illustrate the problem tackled in this paper based on an SAE benchmark [32], equipped with security parameters from [27]. The Society of Automotive Engineers (SAE) has released a set of requirements and a control benchmark, i.e., the control system of an electric vehicle, to be able to compare the effectiveness of new solutions for safety critical automotive systems [32]. This SAE benchmark, however, does not specify any security parameters. Therefore, here, this SAE benchmark is equipped with security parameters from [27]. According to [27], the longer the word size, the more the number of plaintexts required to mount a successful cryptanalysis attack on the system. Therefore, for highly critical messages, such as torque command to the motor control unit, a longer word size is considered by the application designer, compared to less critical messages, such as high/low contactor commands to battery.

Let us focus on the vehicle controller unit of SAE benchmark, with an intrusion detection and resource management (IDRM) task (τ_0) and three real-time tasks (τ_1, τ_2, τ_3). In this example, each real-time task τ_i sends only one message, γ_1 for high contactor control, γ_2 for low contactor control, and γ_3 for torque command [32]. The task set data are shown in Table 2, where all timing quantities are reported in units of milliseconds.

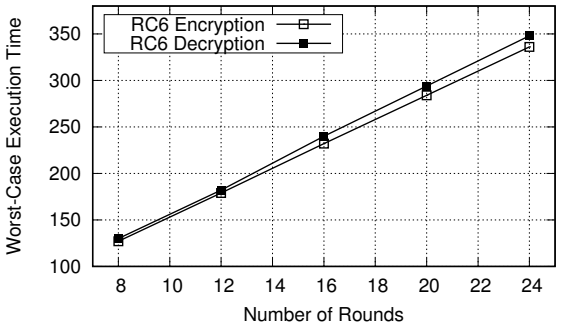

Fig. 3. Worst-case execution times (in microseconds) of RC6 encryption and decryption algorithms versus number of rounds on ARM® Cortex®-M3 processor, based on the aiT worst-case execution-time analysis tool [30]

All tasks are scheduled based on the earliest-deadline-first (EDF) scheduling policy. We assume that message encryption/decryption is done based on the RC5 and its extension RC6 algorithm [23]. To quantify confidentiality strength, we consider the linear cryptanalysis attack [27], [28] for all real-time tasks. The coefficients α_i and ω_i in Table 2 are extracted from [27]. To be consistent with [27], in this example, we consider $e = 2$ in Equation (2). The goal is to maximize the confidentiality strength, as defined in the previous section, by optimizing the number of encryption rounds for each message, under real-time constraints.

The task set utilization margin without the message encryption/decryption is $U = 1 - \sum_{i=0}^{3} \frac{c_i}{h_i} = 0.10$. Since the task set utilization is lower than the utilization limit for the earliest-deadline-first (EDF) scheduling policy, it is possible to use message encryption for the three messages to improve confidentiality. The problem is, therefore, to find the number of rounds each task needs to encrypt the messages, in order to maximize the confidentiality of the entire system, while also satisfying the timing requirements.

Let us first consider a straightforward approach that improves confidentiality by maximizing resource utilization and maximizing the minimum number of rounds among all real-time tasks. In this case, the design solution obtained is given by $r_1 = 16$, $r_2 = 16$, and $r_3 = 17$, which has 100% resource utilization, according to Equation (4). For this design solution, the number of plaintexts required to mount a successful cryptanalysis attack on the system, according to Equation (2), is,

$$\min_{i=1,\ldots,3} \left\{ e^{\alpha_i r_i + \omega_i} \right\} = 2^{100}.$$

However, as we show in the following, maximizing resource utilization does not necessarily lead to maximizing confidentiality.

Let us now use our proposed approach in Section 6.2, to obtain the optimal number of rounds. The optimal solution for the three real-time tasks is given by $r_1^* = 20$, $r_2^* = 20$, and $r_3^* = 15$. In this case, according to Equation (2), the attacker needs approximately 2^{88} plaintexts to mount a successful cryptanalysis attack,

$$\min_{i=1,\ldots,3} \left\{ e^{\alpha_i r_i^* + \omega_i} \right\} = 2^{124}.$$

The solution found by our proposed approach is 2^{24} times more robust to the cryptanalysis attacks, when compared to other solutions. This is because the number of plaintexts required for the cryptanalysis attack to mount a successful attack on the system in our approach is $\frac{2^{124}}{2^{100}} = 2^{24}$ times the straightforward approach.

TABLE 2
Example: task set data (the security and real-time parameters are taken from [27] and [32], respectively)

i	c_i	h_i	d_i	o_i	α_i	ω_i
0	5	50	50	–	–	–
1	3	10	10	0.01	6	4
2	3	10	10	0.01	6	4
3	1	5	5	0.02	8	3

The straightforward approach maximizes the resource utilization (100%) and the minimum number of rounds among all messages ($\min_{i=1,\ldots,3}\{r_i\} = 16$). Our approach, with 100% resource utilization and $\min_{i=1,\ldots,3}\{r_i^*\} = 15$ for minimum number of rounds, outperforms the straightforward approach by a factor of 2^{24}, in terms of confidentiality strength. Therefore, only maximizing the resource utilization and/or maximizing the minimum number of rounds does not necessarily lead to the optimal confidentiality strength. This is because the straightforward approach maximizes an intermediate objective (i.e., minimum number of rounds or resource utilization), which entirely ignores the confidentiality parameters α_i and ω_i. Parameters α_i and ω_i depend on both the criticality of the information carried in a message and the security requirements, e.g., the word size [27]. As opposed to the straightforward approach, our proposed method considers confidentiality as its main objective and maximizes confidentiality strength (defined based on parameters α_i and ω_i in Equation (2)), under real-time constraints. *This example demonstrates the need for an approach that takes security requirements of applications into consideration in confidentiality optimization.*

Now, let us consider a scenario where the system configuration changes at runtime. Let us assume that the minimum inter-arrival time (and also deadline) of task τ_3 is changed to $h_3 = 6$, due to the proportionality of task τ_3 activation rate to the angular velocity of the engine, because the torque command can be sent less frequently for lower engine revolutions [21].[2] Our algorithm, in this case, reacts to this change and finds the optimal design solution $r_1^* = 30$, $r_2^* = 30$, and $r_3^* = 22$, for this configuration. This design solution is $\frac{2^{184}}{2^{124}} = 2^{60}$ times more robust against the cryptanalysis attacks, compared to the previous design solution. *This example demonstrates the need for an approach for runtime adaptation of security parameters.*

Finally, let us assume a more computationally efficient IDRM task τ_0 with execution time $c_0 = 2.5$, i.e., the new IDRM task is two times more computationally efficient. In this case, the optimal design solution is given by $r_1^* = 40$, $r_2^* = 40$, and $r_3^* = 30$. This design solution is $\frac{2^{244}}{2^{184}} = 2^{60}$ times more robust against the cryptanalysis attacks, compared to the previous design solution with the more complex and time consuming IDRM task. This is because the IDRM task itself requires resources for execution. And, because the processor is shared among all the tasks (including the IDRM task), the processor bandwidth allocated to running the IDRM task cannot be used to enhance confidentiality of the system. Therefore, a more computationally complex IDRM (i.e., a longer worst-case execution time c_0) leaves

2. The Revolutions Per Minute (RPM) varies from 500 to 9000. This, approximately, translates into a period of 7.5 milliseconds for 9000 RPM, while a much lower RPM (approximately 120 milliseconds) is required for 500 RPM [33].

less processor bandwidth for security purposes. *This example motivates the need for a computationally efficient confidentiality optimization approach at runtime.*

5 Problem Formulation

In this section, we formulate two optimization problems, which are solved optimally in the next section.

5.1 General Optimization

Let us first consider a general optimization problem. Given is a set of m real-time tasks τ_j mapped on one processing node with computation time c_j and minimum inter-arrival time h_j. Considering the earliest-deadline-first scheduling algorithm we have $U = 1 - \sum_{j=0}^{m} \frac{c_j}{h_j}$ (see Equation (6)). There are n messages γ_i sent/received on each node with the real-time parameter f_i and the security parameters α_i and ω_i (see Section 3).

In this paper, we adopt a worst-case perspective. This is because, a system is only as secure as its weakest component. Therefore, our goal is to maximize the minimum security strength among all messages, under real-time constraints, i.e.,

$$\begin{aligned} \max_{t_1,\ldots,t_n} \quad & \min_{i=1,\ldots,n} \{e^{\alpha_i t_i + \omega_i}\} \\ \text{s.t.} \quad & \sum_{i=1}^{n} t_i f_i \leq U, \end{aligned} \tag{7}$$

where $t_i \in \mathbb{R}$ (a real number), the time alloted for encryption/decryption of message γ_i, has to be determined for each message. We find the optimal solution to this problem analytically in Section 6.1.

5.2 Iterated Block Ciphers Optimization

We also address this problem for the specific case of RC5 and RC6 algorithms. The difference compared to the previous optimization is that, now the optimization variables are the number of rounds, which should be natural numbers, hence a non-convex problem. This makes the problem exponentially more complex, for which we find the optimal solution using a fully-polynomial time algorithm (see Section 6.2). This optimization problem is formulated as follows,

$$\begin{aligned} \max_{r_1,\ldots,r_n} \quad & \min_{i=1,\ldots,n} \{e^{\alpha_i r_i + \omega_i}\} \\ \text{s.t.} \quad & \sum_{i=1}^{n} o_i r_i f_i \leq U, \\ & r_i \in \mathbb{N}, \qquad i = 1,\ldots,n, \end{aligned} \tag{8}$$

where the number of rounds $r_i \in \mathbb{N}$ (a natural number) for each message γ_i is the solution to this optimization problem. And, o_i is the overhead for each round of message encryption/decryption, which is given as an input, in addition to the previously discussed parameters.

6 Optimization Approach

6.1 General Optimization

In this section, we find the optimal solution to problem stated in Equation (7) analytically. Let us transform the minimum function in the objective to a set of inequalities as follows by introducing a new variable x_0,

$$\begin{aligned} \max_{x_0,t_1,\ldots,t_n} \quad & e^{x_0} \\ \text{s.t.} \quad & \sum_{i=1}^{n} t_i f_i \leq U, \\ & e^{x_0} \leq e^{\alpha_i t_i + \omega_i}, \quad i = 1,\ldots,n. \end{aligned} \tag{9}$$

The above problem, however, is not convex and cannot be solved efficiently. Therefore, we transform the above problem to an equivalent problem, by taking the natural logarithm of both sides of the second set of inequalities. Then, the problem will be transformed to a convex problem which can be solved efficiently,

$$\begin{aligned} \max_{x_0,t_1,\ldots,t_n} \quad & x_0 \\ \text{s.t.} \quad & \sum_{i=1}^{n} t_i f_i \leq U, \\ & x_0 \leq \alpha_i t_i + \omega_i, \quad i = 1,\ldots,n. \end{aligned} \tag{10}$$

Note that we need also to have $t_i \geq 0$, for all messages γ_i. However, from the second set of inequalities we have $t_i \geq \frac{x_0 - \omega_i}{\alpha_i}$. Since we are maximizing x_0, these constraints ($t_i \geq 0, \forall i$) will be satisfied implicitly, if at all possible.

Let us now transform the problem (10) into the standard form,

$$\begin{aligned} \min_{x_0,t_1,\ldots,t_n} \quad & -x_0 \\ \text{s.t.} \quad & \sum_{i=1}^{n} t_i f_i - U \leq 0, \\ & x_0 - \alpha_i t_i - \omega_i \leq 0, \quad i = 1,\ldots,n. \end{aligned} \tag{11}$$

To solve the above problem, we use the KKT (Karush-Kuhn-Tucker) necessary conditions for optimality [34]. According to the KKT conditions, the optimum $\boldsymbol{x}^*$ of the problem

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0, \qquad i = 1,\ldots,n, \end{aligned} \tag{12}$$

must necessarily satisfy the following conditions

$$\begin{aligned} \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) &= \boldsymbol{0}, \\ \lambda_i^* g_i(\boldsymbol{x}^*) &= 0, \qquad i = 1,\ldots,n, \\ \lambda_i^* &\geq 0, \qquad i = 1,\ldots,n. \end{aligned} \tag{13}$$

Considering (11), the first KKT condition provides us with $n+1$ equalities,

$$\begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} + \lambda_0^* \begin{bmatrix} 0 \\ f_1 \\ f_2 \\ 0 \\ \vdots \end{bmatrix} + \lambda_1^* \begin{bmatrix} 1 \\ -\alpha_1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} + \lambda_2^* \begin{bmatrix} 1 \\ 0 \\ -\alpha_2 \\ 0 \\ \vdots \end{bmatrix} + \cdots = 0. \tag{14}$$

The second KKT condition also provides us with $n+1$ equalities as follows,

$$\begin{aligned} &\lambda_0^* (\sum_{i=1}^{n} t_i f_i - U) = 0, \\ &\lambda_i^* (x_0 - \alpha_i t_i - \omega_i) = 0, \quad i = 1,\ldots,n. \end{aligned} \tag{15}$$

From the i-th row ($i \geq 2$) of equality (14) we obtain,

$$\lambda_0^* f_i - \lambda_i^* \alpha_i = 0 \quad \Rightarrow \quad \lambda_i^* = \frac{f_i}{\alpha_i} \lambda_0^*. \tag{16}$$

And from the first row of equality (14) we obtain,

$$-1+\sum_{i=1}^{n}\lambda_i^* = 0 \quad \xrightarrow{\text{Eq.(16)}} \quad -1+\sum_{i=1}^{n}\frac{f_i}{\alpha_i}\lambda_0^* = 0, \tag{17}$$

where the second equality is obtained using (16). The second equality in (17) indicates that $\lambda_0^* > 0$. This, in turn, means that $\lambda_i^* > 0$ (see Equation (16)). Therefore, from Equation (15) we have,

$$\begin{aligned} &\sum_{i=1}^{n} t_i f_i - U = 0, \\ &x_0 - \alpha_i t_i - \omega_i = 0, \quad i = 1,\ldots,n. \end{aligned} \tag{18}$$

From this, we obtain an analytical solution for the amount of time that should be alloted for encryption/decryption of each message,

$$\begin{aligned} x_0 &= \frac{U+\sum_{i=1}^{n}\frac{\omega_i f_i}{\alpha_i}}{\sum_{i=1}^{n}\frac{f_i}{\alpha_i}}, \\ t_i &= \frac{x_0-\omega_i}{\alpha_i}. \end{aligned} \tag{19}$$

Note that, since the problem is convex, Equation (19) provides the optimal solution to the problem (7), discussed in the previous section.

6.2 Iterated Block Ciphers Optimization

In this section, we propose an algorithm built on top of the previous optimization for the specific case of iterated block ciphers, RC5 and RC6 algorithms. This optimization problem, stated in Equation (8), is considerably more complex due to the fact that the number of rounds for each message needs to be a natural number. Therefore, this optimization problem is non-convex, which is generally complex to solve.

We use a similar transformation as in the previous section to obtain the following simpler problem,

$$\begin{aligned} \max_{x,r_1,\ldots,r_n} \quad & x \\ \text{s.t.} \quad & \sum_{i=1}^{n} o_i r_i f_i \le U, \\ & x \le \alpha_i r_i + \omega_i, \quad i = 1,\ldots,n, \\ & r_i \in \mathbb{N}, \qquad\quad i = 1,\ldots,n, \end{aligned} \tag{20}$$

which is a mixed-integer linear programming problem (MILP) since $r_i \in \mathbb{N}$, as opposed to the previous section, and in general computationally complex to solve.

Our proposed approach is shown in Algorithm 1. The algorithm uses the solution in Equation (19) to obtain a decent starting point (Lines 1–3), i.e., within certain limits of the optimal solution according to Theorem 1.[3] In Line 3, we find the closest natural solution (i.e., $r_i \in \mathbb{N}$) to the solution of Equation (19), such that each task has a lower or equal utilization. This solution, however, might be suboptimal. In Line 4, we find a lower limit x on the optimal solution x^* in problem (20).

In Lines 5–6, we identify the search interval $[x, x_0] = [\underline{x}, \overline{x}]$. Then, we perform a binary search [35] in the interval of $[\underline{x}, \overline{x}]$ to find the best design solution x^*. In Line 8, we divide the interval to two subintervals of $[\underline{x}, x_{\text{mid}}]$ and $[x_{\text{mid}}, \overline{x}]$. In Line 9, we find the number of rounds for each message that provides us with the lowest task set utilization, which still has x_{mid} as its solution. Then, for a given x_{mid}, if $\sum_{i=1}^{n} o_i r_i f_i \le U$ (Line 10), then $x^* \in [x_{\text{mid}}, \overline{x}]$ and we update r_i^* and $\underline{x}$ with the closest natural solution to x_{mid} in the subinterval of $[x_{\text{mid}}, \overline{x}]$ (Lines 11–12); otherwise (Line 13), $x^* \in [\underline{x}, x_{\text{mid}}]$ and we update $\overline{x}$ with the closest natural solution to x_{mid} in the subinterval of $[\underline{x}, x_{\text{mid}}]$ (Line 14).This procedure continues until the interval of $[\underline{x}, \overline{x}]$ includes only one solution (Line 7).

Algorithm 1 Confidentiality Optimization

1: $x_0 = \dfrac{U+\sum_{i=1}^{n}\frac{\omega_i f_i}{\alpha_i o_i^{-1}}}{\sum_{i=1}^{n}\frac{f_i}{\alpha_i o_i^{-1}}}$;
2: $t_i = \frac{(x_0-\omega_i)}{\alpha_i o_i^{-1}}, \quad i = 1,\ldots,n$;
3: $r_i = \left\lfloor \frac{t_i}{o_i} \right\rfloor, \quad i = 1,\ldots,n$;
4: $x = \min_{i=1,\ldots,n}\{\alpha_i r_i + \omega_i\}$;
5: $\overline{x} = x_0$;
6: $\underline{x} = x$;
7: **while** $\overline{x} > \underline{x}$ **do**
8: $\quad x_{\text{mid}} = \frac{\underline{x}+\overline{x}}{2}$;
9: $\quad r_i = \left\lceil \frac{x_{\text{mid}}-\omega_i}{\alpha_i} \right\rceil, \quad i = 1,\ldots,n$;
10: $\quad$ **if** $\sum_{i=1}^{n} o_i r_i f_i \le U$ **then**
11: $\quad\quad \underline{x} = \min_{i=1,\ldots,n}\{\alpha_i r_i + \omega_i\}$;
12: $\quad\quad r_i^* = r_i, \quad i = 1,\ldots,n$;
13: $\quad$ **else**
14: $\quad\quad \overline{x} = \max_{i=1,\ldots,n}\{\alpha_i (r_i - 1) + \omega_i\}$;
15: $\quad$ **end if**
16: **end while**

First, we demonstrate that the initial (natural) solution obtained from Equation (19) is guaranteed to be close to the optimal solution.

Theorem 1: The distance between the optimal solution x^* and the natural solution x obtained from Equation (19), in Lines 1–4 of Algorithm 1, is bounded from above as follows,

$$x^* - x \le \max_{i=1,\ldots,n}\{\alpha_i\}. \tag{21}$$

Proof: In optimization problem (20), the number of rounds for each message should be a natural number, while this is not the case for optimization problem (9). Since optimization problem (9) is less constrained compared to optimization problem (20) and we are solving a maximization problem, we have,

$$x^* \le x_0.$$

Therefore, the distance between the optimal solution x^* and x, which is obtained from Equation (19) in Lines 1–4 of Algorithm 1, is bounded from above by,

$$x^* - x \le x_0 - x.$$

Assuming $x = \alpha_\iota r_\iota + \omega_\iota$ (i.e., message γ_ι is the limiting message in Line 4) and considering Lines 1–4, we have,

$$\begin{aligned} x_0 - x &= \left(\alpha_i o_i^{-1} t_i + \omega_i\right) - \left(\min_{i=1,\ldots,n}\{\alpha_i r_i + \omega_i\}\right) \\ &= \left(\alpha_i \frac{t_i}{o_i} + \omega_i\right) - \left(\min_{i=1,\ldots,n}\left\{\alpha_i \left\lfloor \frac{t_i}{o_i} \right\rfloor + \omega_i\right\}\right) \\ &= \left(\alpha_\iota \frac{t_\iota}{o_\iota} + \omega_\iota\right) - \left(\alpha_\iota \left\lfloor \frac{t_\iota}{o_\iota} \right\rfloor + \omega_\iota\right) \\ &= \alpha_\iota \left(\frac{t_\iota}{o_\iota} - \left\lfloor \frac{t_\iota}{o_\iota} \right\rfloor\right) \\ &\le \alpha_\iota \le \max_{i=1,\ldots,n}\{\alpha_i\}. \end{aligned}$$

3. The value of α_i in Equation (19) is substituted by $\alpha_i o_i^{-1}$ here. This is because α_i in problem (7) is given per second, while in problem (8) is given per round.

Hence $x^* - x \leq \max_{i=1,\ldots,n}\{\alpha_i\}$. □

Theorem 2: Algorithm 1 has a fully-polynomial time complexity of $O(n^2)$, where n is the number of massages.

Proof: First, we emphasize that the optimal solution x^* is bounded from above by x_0 and from below by x (i.e., $x_0\ x \leq x^* \leq x_0$). Since the number of rounds for each message increases in steps of α_i, the number of rounds r_i could take only $\left\lceil \frac{x_0 - x}{\alpha_i} \right\rceil$ values. From Theorem 1 we know that $x^* - x \leq x_0 - x \leq \max_{i=1,\ldots,n}\{\alpha_i\}$. Therefore, for each message, the number of rounds r_i could, at most, take only $\left\lceil \frac{\max_{i=1,\ldots,n}\{\alpha_i\}}{\alpha_i} \right\rceil$ values. Finally, since we have n messages, the number of natural design solutions in the interval of $[x, x_0]$ is bounded from above by $n \cdot k$, where $k = \left\lceil \frac{\max_{i=1,\ldots,n}\{\alpha_i\}}{\min_{i=1,\ldots,n}\{\alpha_i\}} \right\rceil$.

Then, in Lines 7–18, we perform a binary search [35] in the interval of $[x, x_0]$ to find the best design solution. Since the number of rounds for each message increases in steps of α_i, we have a uniform distribution for each message. Therefore, the number of points will be reduced to almost half in every iteration of binary search. For each message, if the number of design solutions in $[\underline{x}, \overline{x}]$ is odd, then the number of points in the two subintervals of $[x_{\text{mid}}, \overline{x}]$ and $[\underline{x}, x_{\text{mid}}]$ may differ by one. Considering all messages, in the worst-case scenario, the number of points in the two subintervals may differ by n values. Therefore, in the worst case, the binary search reduces the size of the set logarithmically, $O(\log(n \cdot k))$, until there are only n values left in the set. Then, in the worst case, the binary search reduces the size of the set linearly, $O(n)$. Therefore, in the worst case, the loop runs for $O(\log(n \cdot k) + n) = O(n)$. Considering that each iteration of the loop has a complexity of $O(n)$, because of the minimum/maximum operations, Algorithm 1 has a time complexity of $O(n^2)$. □

Theorem 3: Algorithm 1 provides the optimal solution to optimization problem (20).

Proof: Algorithm 1 uses the binary search [35] to explore all design solutions in the interval of $[x, x_0]$. To prove the optimality of our algorithm, which is based on the binary search, it is sufficient to prove the monotonicity property of the objective function. That is, given two solutions $x^{(1)}$ and $x^{(2)}$, if $x^{(1)} > x^{(2)}$, then $\sum_{i=1}^{n} o_i r_i^{(1)} f_i \geq \sum_{i=1}^{n} o_i r_i^{(2)} f_i$, where $r_i^{(1)}$ and $r_i^{(2)}$ correspond to solutions $x^{(1)}$ and $x^{(2)}$, respectively. Based on the monotonicity property, if $\sum_{i=1}^{n} o_i r_i^{(2)} f_i > U$ for solution $x^{(2)}$, then the algorithms does not check any solution $x^{(1)}$, where $x^{(1)} > x^{(2)}$. The monotonicity property can be proved using $r_i^{(1)} = \left\lceil \frac{x^{(1)} - \omega_i}{\alpha_i} \right\rceil$ and $r_i^{(2)} = \left\lceil \frac{x^{(2)} - \omega_i}{\alpha_i} \right\rceil$, as follows,

$$x^{(1)} \geq x^{(2)} \Leftrightarrow \sum_{i=1}^{n} o_i \left\lceil \frac{x^{(1)} - \omega_i}{\alpha_i} \right\rceil f_i \geq \sum_{i=1}^{n} o_i \left\lceil \frac{x^{(2)} - \omega_i}{\alpha_i} \right\rceil f_i$$
$$\Leftrightarrow \sum_{i=1}^{n} o_i r_i^{(1)} f_i \geq \sum_{i=1}^{n} o_i r_i^{(2)} f_i.$$

□

7 Experimental Results

In the previous section, we demonstrated the optimality of our approach. In this section we evaluate the execution time of our proposed algorithm. We compare the runtime of Algorithm 1 against a mixed-integer linear programming (MILP) implementation in [14], adapted to our system model, since it also provides the optimal solution. We use three state-of-the-art toolboxes (CPLEX, GUROBI, and MOSEK) to solve the problem stated in Equation (20).[4]

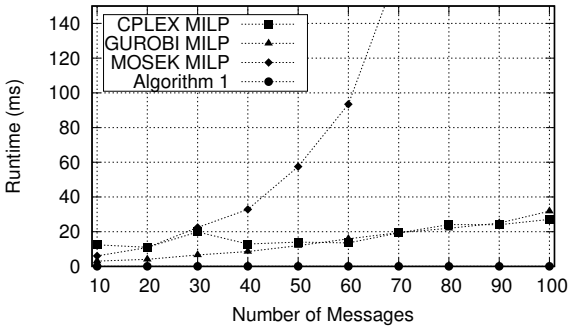

Fig. 4. Runtime of our optimal algorithm against MILP solvers (CPLEX, GUROBI, and MOSEK).

We compare the average runtime of these approaches versus the number of messages sent/received on the platform. We generate 100 random benchmarks for each number of messages, with different task set utilizations between $40\% - 90\%$, assuming the earliest-deadline-first (EDF) scheduling policy. The UUniFast algorithm [36] is used to generate a set of random real-time tasks for a given utilization. The experiments are done on a PC with a quad-core CPU running at 3.6 GHz with 32 GB of RAM and Linux.

The results are shown in Figure 4. For small number of messages, e.g., the 100 random benchmarks with 10 messages, the average runtime of our proposed algorithm is 0.04 ms, while the CPLEX, GUROBI, and MOSEK MILP solvers could find a solution in 12.45, 3.05, and 6.61 ms, respectively, leading to two orders of magnitude improvement by our algorithm in runtime. For large number of messages, e.g., the 100 random benchmarks with 100 messages, the average runtime of our proposed algorithm is 0.06 ms, while the CPLEX, GUROBI, and MOSEK MILP solvers could find a solution only after 27.13, 31.80, 1704.02 ms, respectively. Therefore, for large number of messages, our algorithm outperforms the MILP approach by almost three orders of magnitude.

In summary, our proposed Algorithm 1 is computationally efficient (fully-polynomial time complexity) and is appropriate to be employed at runtime.

8 Conclusions

In this paper, we highlight the importance of an efficient online algorithm for managing the trade-off between the available amount of resources and the security strength. We obtained the optimal amount of time that should be allocated to message encryption/decryption analytically. We also proposed an algorithm for the specific case of iterated block ciphers. This algorithm has polynomial time complexity with respect to the number of messages. We compared our optimal algorithm against a mixed-integer linear programming approach and demonstrated the efficiency of our proposed approach.

4. The MATLAB code for this comparison is available at: https://documents.epfl.ch/users/a/am/aminifar/www/confidentiality_optimization.zip

ACKNOWLEDGMENTS

This work has been partially supported by the Hasler Foundation through the Project No. 15048, and by the ONR-G through the Award Grant No. N62909-17-1-2006.

REFERENCES

[1] M. Wolf, A. Weimerskirch, and C. Paar, *Embedded Security in Cars: Securing Current and Future Automotive IT Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, ch. Secure In-Vehicle Communication, pp. 95–109.

[2] T. Hoppe, S. Kiltz, and J. Dittmann, *Computer Safety, Reliability, and Security: 27th International Conference, SAFECOMP 2008 Newcastle upon Tyne, UK, September 22-25, 2008 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures, pp. 235–248.

[3] F. Sagstetter, M. Lukasiewycz, S. Steinhorst, M. Wolf, A. Bouard, W. R. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty, "Security challenges in automotive hardware/software architecture design," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, 2013, pp. 458–463.

[4] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, *Cryptographic Hardware and Embedded Systems - CHES 2013: 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*. Springer Berlin Heidelberg, 2013, ch. Non-invasive Spoofing Attacks for Anti-lock Braking Systems, pp. 55–72.

[5] S. Mohan, M. K. Yoon, R. Pellizzoni, and R. Bobba, "Real-time systems security through scheduler constraints," in *Proceedings of the 26th Euromicro Conference on Real-Time Systems*, 2014, pp. 129–140.

[6] M. D. Natale and A. L. Sangiovanni-Vincentelli, "Moving from federated to integrated architectures in automotive: The role of standards, methods and tools," *Proceedings of the IEEE*, vol. 98, no. 4, pp. 603–620, 2010.

[7] G. C. Buttazzo, *Hard Real-Time Computing Systems*. Kluwer Academic, 1997.

[8] A. Aminifar, "Analysis, design, and optimization of embedded control systems," Ph.D. dissertation, Linköping Studies in Science and Technology, 2016.

[9] R. L. Rivest, *The RC5 encryption algorithm*. Springer, 1995, pp. 86–96.

[10] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Transactions Embedded Computing Syststems*, vol. 3, no. 3, pp. 461–491, 2004.

[11] P. Kocher, R. Lee, G. McGraw, and A. Raghunathan, "Security as a new dimension in embedded system design," in *Proceedings of the 41st Annual Design Automation Conference (DAC)*. ACM, 2004, pp. 753–760, moderator-Ravi, Srivaths.

[12] K.-D. Kang and S. H. Son, "Systematic security and timeliness tradeoffs in real-time embedded systems," in *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2006, pp. 183–189.

[13] T. Xie and X. Qin, "Improving security for periodic tasks in embedded systems through scheduling," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 3, 2007.

[14] M. Lin, L. T. Yang, X. Qin, N. Zheng, Z. Wu, and M. Qiu, "Static security optimization for real-time systems," *IEEE Transactions on Industrial Informatics*, 2009.

[15] K. Jiang, P. Eles, and Z. Peng, "Optimization of message encryption for distributed embedded systems with real-time constraints," in *IEEE 14th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, 2011, pp. 243–248.

[16] ——, "Co-design techniques for distributed real-time embedded systems with communication security constraints," in *Proceedings of the 15th Conference for Design, Automation and Test in Europe (DATE)*, 2012, pp. 947–952.

[17] ——, "Optimization of secure embedded systems with dynamic task sets," in *Proceedings of the 16th Conference for Design, Automation and Test in Europe (DATE)*, 2013, pp. 1765–1770.

[18] S. Samii, P. Eles, Z. Peng, and A. Cervin, "Quality-driven synthesis of embedded multi-mode control systems," in *Proceedings of the 46th Design Automation Conference*, 2009, pp. 864–869.

[19] A. Cervin, J. Eker, B. Bernhardsson, and K. E. Årzén, "Feedback–feedforward scheduling of control tasks," *Real-Time Systems*, vol. 23, no. 1–2, pp. 25–53, 2002.

[20] D. Henriksson and A. Cervin, "Optimal on-line sampling period assignment for real-time control tasks based on plant state information," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 4469–4474.

[21] G. C. Buttazzo, E. Bini, and D. Buttle, "Rate-adaptive tasks: Model, analysis, and design issues," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2014, pp. 1–6.

[22] R. Mahfouzi, A. Aminifar, P. Eles, Z. Peng, and M. Villani, "Intrusion-damage assessment and mitigation in cyber-physical systems for control applications," in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. ACM, 2016, pp. 141–150.

[23] R. L. Rivest, M. J. B. Robshaw, R. Sidney, , and Y. L. Yin, "The rc6 block cipher," in *the 1st Advanced Encryption Standard (AES) Conference*, 1998.

[24] J. Nechvatal, E. Barker, L. Bassham, W. Burr, and M. Dworkin, "Report on the development of the advanced encryption standard (aes)," DTIC Document, Tech. Rep., 2000.

[25] D.-Y. Yeung and Y. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern Recognition*, vol. 36, no. 1, pp. 229–243, 2003.

[26] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 47–61, 1973.

[27] J. Borst, B. Preneel, and J. Vandewalle, "Linear cryptanalysis of rc5 and rc6," in *FSE*, vol. 99. Springer, 1999, pp. 16–30.

[28] S. Contini, R. L. Rivest, M. J. B. Robshaw, R. Sidney, , and Y. L. Yin, "The security of rc6 block cipher," Tech. Rep., 1998. [Online]. Available: https://people.csail.mit.edu/rivest/ContiniRivestRobshawYin-TheSecurityOfTheRC6BlockCipher.pdf

[29] B. S. Kaliski and Y. L. Yin, "On the security of the rc5 encryption algorithm," RSA Laboratories Technical Report TR-602. To appear, Tech. Rep., 1998.

[30] C. Ferdinand and R. Heckmann, *aiT: Worst-Case Execution Time Prediction by Static Program Analysis*. Springer US, 2004, pp. 377–383.

[31] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra *et al.*, "The worst-case execution-time problem–overview of methods and survey of tools," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, no. 3, p. 36, 2008.

[32] H. Kopetz, "A solution to an automotive control system benchmark," in *1994 Proceedings Real-Time Systems Symposium*, Dec 1994, pp. 154–158.

[33] J. Kim, K. Lakshmanan, and R. Rajkumar, "Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems," in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, April 2012, pp. 55–64.

[34] M. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms*. Wiley, 2006.

[35] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[36] E. Bini and G. C. Buttazzo, "Measuring the performance of schedulability tests," *Real-Time Systems*, vol. 30, no. 1-2, pp. 129–154, 2005.