Journal of Circuits, Systems, and Computers
Vol. 28, No. 2 (2019) 1930002 (23 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218126619300022

A Short Review of Security-Aware Techniques in Real-Time Embedded Systems*

Hongxia Chai, Gongxuan Zhang, Junlong Zhou†,
Jin Sun, Longxia Huang and Tian Wang

Department of Computer Science and Engineering,
Nanjing University of Science and Technology,
Nanjing 210094, P. R. China
†*jlzhou@njust.edu.cn*

Received 8 January 2018
Accepted 22 April 2018
Published 28 May 2018

With the rapid development of embedded systems, users and services have been greatly facilitated while also experiencing security threats as a result of cyber-attacks and system vulnerabilities. Currently, the real-time embedded system (RTES) focus is to deal with these security issues. In this paper, we introduce a short review of security-aware techniques for RTES. We mainly discuss two common approaches to improve the security of RTESs. The first approach is achieved by exploring specific attacks. The second approach is realized by deploying security-guaranteed services. However, improving the security of embedded systems may cause excessive energy consumption at the same time. Therefore, we investigate the secure and energy-aware RTESs on a wide range of research. In addition, we study a number of common applications used in secure RETSs. This paper stands for providing awareness and better understanding of the current RTES research status as well as technical theory behind it. Hence, the RTES security issues are resolved.

Keywords: Real-time embedded systems; attacks; security-guaranteed services; energy-efficiency; applications.

1. Introduction

The embedded system is a computer application system embedded in the object system and interacts with the object system. It will propose a limitation on the response time of the application system when implementing certain task processes of the object system. Due to the time spent in running the software in the application system, it often cannot meet the limited response time requirement, resulting in the real-time problem of the embedded system. That is to say a real-time embedded

*This paper was recommended by Regional Editor Tongquan Wei.
†Corresponding author.

system (RTES) is an embedded system that accomplishes a specific task within a specified time. The combination of RTES and Internet brings great convenience to our daily life. However, in the mean time, the embedded systems are also facing a variety of security threats inevitably due to its connection to Internet. Once the systems deployed in critical applications such as flight control, financial administration and automotive electronics are attacked by malicious intruders, the resultant loss would be incalculable. Therefore, designing security-aware techniques for RTES becomes a necessity.

Considerable research efforts have been devoted to the designing of a secure RTES, which can be divided into three categories. The first category to realize a secure system is studying and simulating the common security attacks such as side-channel attack (SCA),[1–5] level-based multi-fault attacks (LMFAs),[6] system-level cyber attacks (SLCAs), code injection attacks (CIAs),[7,8] etc. The second category to realize a secure system is providing security-guaranteed services such as security-aware model and framework,[9–13] security-driven hardware design,[14,15] encryption algorithm.[16–19] The deployment of such security-guaranteed services would inevitably consume more energy. However, the energy supply of most embedded systems is very limited since many embedded systems are powered by battery or renewable energy that is either insufficient or unstable. Based on this, energy efficiency needs to be taken into account when designing security-guaranteed services. Recently, the researchers concentrate on developing security-aware energy-efficient task scheduling mechanisms[20–30] that cannot only promote the system safety performance but also improve the system energy efficiency. The third category to realize a secure system is investigating the safety issues in specific applications such as cyber-physical systems (CPSs),[31–33] automotive systems,[34–37] clusters,[38–40] grids,[41–43] etc.

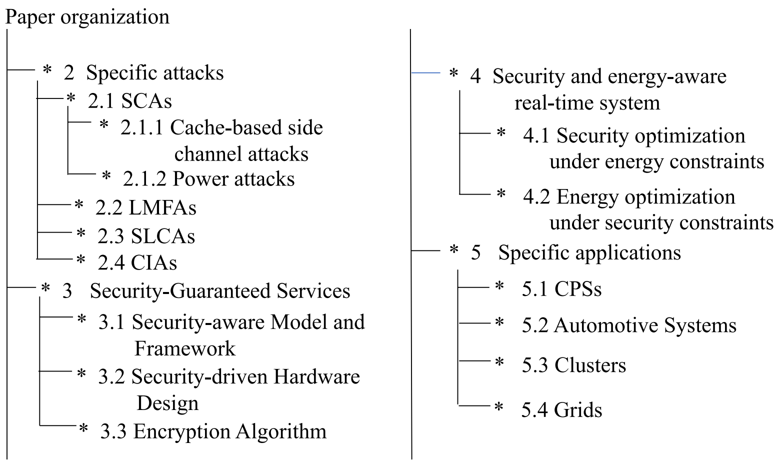
Paper organization
* 2 Specific attacks
* 2.1 SCAs
* 2.1.1 Cache-based side channel attacks
* 2.1.2 Power attacks
* 2.2 LMFAs
* 2.3 SLCAs
* 2.4 CIAs
* 3 Security-Guaranteed Services
* 3.1 Security-aware Model and Framework
* 3.2 Security-driven Hardware Design
* 3.3 Encryption Algorithm
* 4 Security and energy-aware real-time system
* 4.1 Security optimization under energy constraints
* 4.2 Energy optimization under security constraints
* 5 Specific applications
* 5.1 CPSs
* 5.2 Automotive Systems
* 5.3 Clusters
* 5.4 Grids

Fig. 1. Organization of the paper.

Table 1. Main abbreviations used in the paper.

Abbreviation	Definition
RTESs	Real-time embedded systems
SCAs	Side-channel attacks
LMFAs	Level-based multi-fault attacks
SLCAs	System-level cyber attacks
CIAs	Code injection attacks
AES	Advanced encryption standard
CPSs	Cyber-physical systems
RTSs	Real-time systems
DPA	Differential power analysis
WDDL	Wave dynamic differential logic
RTOS	Real-time operating system
MPSoCs	Multiprocessor system-on-chips
MTVOA	Multi-task vulnerability optimization approach
FVUC	Feedback vulnerability and utilization control
CBD	Component-based development
MDD	Model-driven development
DESs	Distributed embedded systems
SCESs	Security-critical embedded systems
RSDP	Random scaling-based dynamic programming
ECU	Electronic control unit
GPGPU	General-purpose graphics processing unit
GED	GPU-based ECU design
RED	Reconfigurable ECU design
DVFS	Dynamic voltage frequency scaling
RNAA	Round to nearest based approximation algorithm
ASD	Attack sequence diagram
S3A	Secure system simplex architecture
TDMA	Time division multiple access
CAN	Controller area network
MILP	Mixed integer linear programming
QoS	Quality-of-service

Contribution and organization: A large amount of state-of-art approaches designed for security-aware RTES are summarized in this paper. The rest of the paper is organized in Fig. 1. First, we discuss the importance of security-aware designs in RETS (Sec. 2). We then introduce two types of approaches to solve the security related issues. One is specific attacks against techniques (Sec. 2) and the other is security-guaranteed services (Sec. 3). Since increasing the security level of services could lead to the increase in energy consumption, we discuss the energy-efficient security-aware techniques in RETS (Sec. 4). In addition, we introduce some key applications of security-critical RETS in various domains (Sec. 5). For the sake of better understanding, we summarize the main abbreviations in Table 1.

2. Specific Attacks

Generally, RTESs are connected to physical devices installed in open public environments, which make the systems vulnerable to security attacks/threats. These

Table 2. Brief introduction of specific attacks.

Attacks	References	Approaches
SCAs	Chen *et al.*[1,2]	ScheduLeak
	Jiang *et al.*[3]	Analytical framework
	Hwang *et al.*[4]	Security coprocessor IC
	Bao and Srivastava[5]	Secure-aware algorithm
LMFAs	Druml *et al.*[6]	Analysis methodology
SLCAs	Song *et al.*	CAML
CIAs	Patel *et al.*[7]	Hardware/software approach
	Patel and Paramewaran[8]	SHIELD

security attacks may cause the crash of the entire system, thus leading to a significant loss. Therefore, it is imperative for RTESs to protect critical information and against the potential attacks. At present, most researchers focus on designing the security-aware techniques for handling attacks such as SCAs,[1–5] LMFAs,[6] SLCAs and CIAs,[7,8] which are briefly introduced in Table 2. The details of these attacks and their corresponding solutions are given in the following sections.

2.1. *SCAs*

SCA is a method of attacking cryptographic devices, which is based on the leakage of side-channel information such as time consumption, power consumption or electromagnetic radiation of the encrypted electronic device. SCA can use that information to break the system, which poses a serious threat to cryptographic devices. SCAs mainly include simple cache-based SCAs and DPAs, which are described in detail below.

2.1.1. *Cache-based SCAs*

Chen *et al.*[1] designed a new reconnaissance cache-based SCA to monitor the target task execution behaviors with accurate task schedule knowledge. This attack can extract the accurate schedule of hard RTSs by using fixed-priority algorithms. In order to extract the task schedule more effectively, they proposed ScheduLeak that utilizes the periodic and predictable properties of hard RTSs to rebuild the schedule. Moreover, this attacker's target is to steal information without being detected rather than to interrupt or disable the RTSs. To achieve this goal, ScheduLeak uses the lowest priority tasks to monitor the system schedule by detecting its busy intervals. Rate Monotonic algorithm, in which a task owns higher priority if its period is shorter, is used in ScheduLeak to determine the priorities of the tasks. The proposed ScheduLeak gives a clear picture on the process of case-based SCAs, which can help designers to develop approaches that prevent these attacks.

2.1.2. *Power attacks*

Jiang *et al.*[3] proposed an analytical framework to quantitatively capture the effect of real-time scheduling methods on the robustness against DPA attacks. The goal of

DPA attacks is to obtain the AES keys used by the system. They defined the robustness of the system against DPA attacks to be the difficulty according to time overhead for an attacker to collect enough information to observe the high correlations between the sample measurement and hypothetical power consumptions. The proposed framework can allow leakage points happen at different times to decrease the underlying correlations.

Hwang *et al.*[4] introduced an SCA-resistant secure coprocessor that integrates two circuit-level technologies to resist DPA attacks. The first technology is WDDL, which is employed to produce logic gates. The second technology is referred to as differential routing (DR), which is utilized to guarantee that the interconnected capacitances of gate output nodes in WDDL are equal. The proposed WDDL and DR are able to improve the orders of size of DPA resistance in the implementation of standard cell ICs to defeat power attacks. Experimental results show that the DPA attack only needs 8,000 encryptions to reveal the whole secret key on the unsecured coprocessor, while cannot even after 1,500,000 encryptions on the secure coprocessor.

Bao and Srivastava[5] proposed a secure-aware algorithm to order aperiodic soft real-time tasks against power attacks based on temperature tracking. The goal of such attackers is to detect the execution order of tasks or to recognize the task running at a specific time. Thus, the authors simulated a powerful attacker behavior that utilizes Kalman Filter based on temperature-tracking, and proposed a metric named side-channel vulnerability factor[44,45] for the information leakage of the side-channel. The proposed algorithm can achieve a balance between the total delay and the information leakage of side-channels. In addition, the authors developed several heuristics in order to reduce the computational complexity.

2.2. *LMFAs*

LMFA is a reliability verification technology that deliberately introduces faults into the system through controlled experiments and observes the behavior (e.g., power supply, power consumption, information leakage) of the system in the event of a fault. Some researchers showed how to emulate LMFAs and how to exploit the resulting system behavior to extract safety-related information.

Druml *et al.*[6] proposed a new analysis approach to assess system behaviors at the design stage. They adopted an emulation-based method to evaluate faults, which could influence system behaviors such as information leakage, power supply and power consumption that provide accurate cyclic analysis in real time. More specifically, the approach emulates multi-fault attacks at system-level (i.e., one type of LMFAs) and utilizes the final system behavior to draw security related information. The power model is produced by a specific characterization process. Both the power model and the characterization process are specific to each tested system and are realized by the test system designer. Experimental results show that, using the proposed analysis approach, power errors can be easily detected and repaired before tape-out.

2.3. *SLCAs*

SLCA utilizes the vulnerabilities and security flaws in the network to attack the hardware and software in the network system, and data in the system. The threats faced by CPSs come from many aspects and will change over time. Security of the system is complicated by the time and safety restrictions of the CPS.

Song *et al.* proposed an anomaly detection method based on machine learning called CAML that evaluates, detects and resumes from SLCAs in RTOSs. To evaluate the efficiency of CAML, they used two unsupervised learning methods that detect the attacks and improve the system resilience to SLCAs without violating deadlines. The first learning method is algorithmic sequence learning based on the method in Ref. 46, which can be used to forecast events or depict frequent modes for sequentially, temporal-ordered datasets. The second learning method is called DBSCAN,[47] which can be used to find clusters. The authors also utilized the fault recovery technique[48] that recovers the system to barrier-free state predictably. Results demonstrated that CAML can greatly enhance the system robustness without a big performance degradation by ensuring RTOS to resist SLCAs.

2.4. *CIAs*

Patel *et al.*[7] proposed a new approach for applications running on multiprocessors to deal with CIAs while minimizing runtime and area overheads. The proposed approach is capable of finding out the accurate basic block in which the attack occurs. This approach adopts a special monitor processor to monitor the application at run-time. In this approach, each single processor communicates with the monitor processor via a FIFO queue and is checked all the time. Static analyses of the timing configuration file and program map are used to get program information, which is encrypted and saved in the monitor processor at compile time. Each basic program block must have security information that identifies itself uniquely at run-time from others. The information obtained by static analysis enables monitor processor at runtime to check the program on each processor.

Patel and Paramewaran[8] presented a hardware/software systematic detection approach SHIELD to recognize the CIAs on MPSoCs. The processors of an MPSoC system can be classified into two categories. One category is the application processors that are used to run software programs and the other category is the monitor processor that is used to monitor the application processors. The monitor processor can quickly analyze the information transmitted from the application processors to itself at runtime by exploiting the custom hardware. SHIELD extracts the execution time of the basic blocks and the control flow of runtime checking, thus can automatically achieve a safe MPSoC architecture and an instrumented binary.

The simulation results showed that SHIELD has the advantages of short runtime and low overhead as compared to existing techniques.

3. Security-Guaranteed Services

In the past, security related issues were not a major concern in RTESs. In such systems, customized components of the system are isolated from each other physically, and security issues are not taken into account when deploying these components. Thus, more and more attacks began to destroy these components. These attacks will compromise the stability of the system and even result in harms to humans and the environment.[49] Considering this, security-guaranteed services become a necessity in the design of RTESs. A large number of security-guaranteed services have been designed in recent years, some of which are listed in Table 3. We describe these services in detail below.

3.1. *Security-aware model and framework*

Pellizzoni *et al.*[9] proposed a simple yet effective model to capture the security constraints between tasks in RTSs. The model is integrated with real-time scheduling algorithms to avoid the information leakage through storage channels on shared resources. The authors especially modeled a constraint, noleak, to capture whether it is necessary to prohibit accidental information sharing between tasks and summarize the concept of real-time scheduling under the security constraint. They also provided schedulable mechanisms for preemptive and nonpriority tasks, which optimally assign real-time tasks to processors.

Abdi *et al.*[10] designed a restart-based framework, ReSecure, to ensure the security of RTSs while meeting the timing constraints of tasks. It asynchronously restarts the platform and loads new images of the application and operating system after each reboot to remove intruders or malicious entities. The authors also provided an

Table 3. Brief introduction of security-guaranteed services.

Services	References	Approaches
Security-aware model and framework	Pellizzoni *et al.*[9]	Generalized model
	Abdi *et al.*[10]	Restart-based framework
	Hasan *et al.*[11]	Contego
	Ma *et al.*[12]	Control and security management
	Paryab[13]	Integrating security mechanisms
Security-driven hardware design	Tan *et al.*[14]	System-level method in MPSoC
	Saadatmand[15]	Method using CBD and MDD
Encryption algorithm	Zhang *et al.*[16]	MTVOM
	Jiang *et al.*[17]	Hardware/software technology
	Jiang *et al.*[18]	FVUC
	Mohan *et al.*[19]	SCRTS

analytical framework for system designers to determine the best trade-offs between the control system performance and the security customizing parameters of the systems while ensuring the security of the physical system. Simulation results showed that ReSecure can guarantee a secure system under the unavailability of complex units.

Hasan *et al.*[11] presented a multi-mode extensible framework, Contego, in order to integrate security tasks into RTSs with timing requirements. Contego consolidates hierarchical scheduling with opportunistic execution to preserve compatibility with traditional systems. It enables the security tasks to operate with the minimal real-time task scheduling order while ensuring its timing constraints. For most cases, Contego opportunistically implements intrusion detection tasks in a passive mode. However, Contego will turn to the ACTIVE operating mode executed potentially at a higher priority while guaranteeing the schedulability of real-time tasks to adapt the changes caused by bad activities. Thus, compared with the previous work,[50] Contego can provide a faster detection. With the Contego, designer can improve the security status and overall safety as well. What's more, a security design metrics for RTSs is provided here and with the Contego.

Taking aperiodic real-time applications into account, Ma *et al.*[12] designed an adaptive risk monitoring and real-time scheduling management mechanism for the uniprocessor system. The system administrator only needs to specify the expected system performance and this mechanism can automatically monitor the system status and adjust its performance to meet these requirements. In order to obtain the adaptive ability to reduce system risk without reducing the soft real-time performance, a two-level feedback scheduling framework is deployed at runtime. The higher-level components automatically monitor the status of waiting queues by admitting or rejecting the newly arrived task and assign reasonable security levels for the prepared tasks. The lower level can dynamically adjust the security level of a prepared task depending on the run-time to keep the security and real-time. Results showed the proposed mechanism has a good amount of adaptability, especially under dynamic system workloads.

Paryab[13] proposed two mechanisms against information leakage, in order to refresh the state of a resource during a context switch among protected tasks and allocate the resources between tasks statically. To integrate these two mechanisms while deriving predictable timing interference limits, the authors introduced a new security model. This model describes the relationship of confidentiality among any pair of tasks, which determines whether the task requires the confidentiality of another task. To determine the least resource state resets and the best distribution of the shared resource parts to tasks, the authors designed an optimization approach into the overhead for protecting confidentiality. They also provided a heuristic optimization method for designers to balance the influences of the two mechanisms while looking for the optimal design.

3.2. *Security-driven hardware design*

Tan *et al.*[14] presented a system-level method that determines the relationships between resources and tasks in MPSoC. In such an MPSoC, each task supervises its own address space regions to isolate tasks without the need of dependence on permissions at run-time. This isolation is needed to reduce the potential influence of untrusted IP blocks or compromise tasks. The proposed mechanisms allow secure sharing of memory and IP blocks between system tasks. As the aim of these mechanisms is to guarantee that resource accesses are allowed only on demand, they avoided central trust by mixing the static allocation of access rights, while assigning access control management to the task itself. To support dynamic changes, they designed a hardware isolation unit based on task resource relationships without the operating system. In this method, some of the original shortcomings and the spare bus-based interconnections between IUs are addressed.

CBD and MDD can deal with the increasing complexity of RTS. Saadatmand[15] proposed a method using CBD and MDD to allow the designers to easily integrate security concerns into the design of systems and determine and run their timing effects and costs. For protecting the sensitive datas in the system component model, this method annotates and identifies them and automatically exports a new component model. The new component model is derived by a series of predefined strategies, each of which defines a different series of possible encryption and decryption methods to be utilized as the security component implementation.

3.3. *Encryption algorithm*

Traditional RM and earliest deadline first (EDF) scheduling algorithms do not take into account security and timing constraints. Zhang *et al.*[16] developed an MTVOA which is based on dynamic scheduling algorithms. To achieve the minimal vulnerability while meeting the timing requirements of distributed RTSs, they assigned MTVOA to the central computer. They chose RC5 for each critical task to presume the relationships between execution time, vulnerability and encryption rounds. A decision map consisting of state nodes and edges is generated to consider all possible cases. Results showed that MTVOA can always realize the vulnerability optimization goals with an improved CPU utilizations.

Jiang *et al.*[17] proposed a method that utilizes hardware/software co-design technology to realize encryption algorithms. The goal of this method is to find out the corresponding process mapping and minimum hardware overhead for system encryption and decryption tasks. To achieve this goal, a heuristic is introduced, which is divided into two parts, list scheduling and resource allocation.[51] The first one controls the outer collaborative design of FPGA resource sharing and allocation. The second one processes the system scheduling. Results showed that the proposed heuristic approach provides relative optimal results and saves a large number of FPGA units.

NAND flash memory has been a critical component in real-time embedded devices, thus improving the security of nonvolatile memories-based embedded system which becomes a great challenge. Based on the actual security protection and write operation on small NAND flash systems, Jiang *et al.*[18] established a security-aware and real-time storage application model. They introduced a mechanism, FVUC, which combines two proportional-integral controllers (vulnerability controller and utilization controller) to construct a large feedback loop. FVUC uses the security level of each storage task to monitor the system state at runtime. It also determines how many flash pages will be encrypted by the cryptographic service to ensure soft real-time requirements and reduce vulnerability. Experimental results showed that FVUC can balance the system vulnerability and utilization, thus achieving a better overall system performance.

Each time a hand-off occurs between tasks that belong to different levels of security, the information may leak through shared resources. To deal with the information leakage problem, Mohan *et al.*[19] proposed a method named SCRTS to integrate security constraints into real-time scheduling algorithms. The main idea of the method is to amend the real-time scheduling method and use shared resources to alleviate the information leakage problem among tasks of different security levels. They used the fixed priority scheduling algorithms for showing how to avoid information leakage among tasks with an acceptable performance overhead. The proposed method allows the designers of the RTSs to properly assess the inherent tradeoffs among real-time guarantees and security requirements.

4. Security and Energy-Aware RTESs

The current task scheduling mechanisms either only consider the security management of the system, or only concern about the effectiveness of energy consumption, and rarely consider these two aspects at the same time. Security service applications, such as cryptographic algorithms, are computationally intensive applications that typically consume large amounts of power, so it is necessary to consider the efficiency of energy usage when introducing security services into embedded systems. In recent years, many researchers took both security and energy efficiency into account, and are currently looking for secure and energy-efficient task scheduling mechanism. This scheduling mechanism is divided into two categories, and the two categories will be introduced below. We listed them in Table 4.

4.1. *Security optimization under energy constraints*

Due to the particularity of its application environment, the energy supply of many embedded systems is very limited. Thus, improving the efficiency of energy usage has become a major topic in RTESs. The introduction of security services will inevitably consume more processing time, which increases the difficulty of meeting real-time

Table 4. Researches on security and energy-aware RTESs.

Classification	References	Approaches
Security optimization under energy constraints	Jiang *et al.*[20]	GA-based heuristic method
	Guo *et al.*[22]	GA-based heuristic method
	Jiang *et al.*[23]	Investigate popular security algorithms
	Jiang *et al.*[24]	Multi-dimensional analysis framework dynamic programming algorithm
Energy optimization under security constraints	Poudel *et al.*[25]	ECU architectures
	Jiang *et al.*[26]	TSHO
	Zhang *et al.*[27]	ESAMA
	Jiang *et al.*[28]	RNAA
	Jiang *et al.*[29]	Genetic algorithm (GA)-based NSGA-II
	Nicolae *et al.*[30]	Self-adaptable security infrastructure

and energy efficient requirements. Therefore, how to adopt a reasonable encryption strategy to improve the system security performance under the timing and energy consumption constraints has become a major challenge for RTESs.

Jiang *et al.*[20] focused on the protection of confidentiality of internal communications in DESs under the energy and time constraints. Reduced communication traffic flow and enhanced message encryption are the two ways to the design process, they used the mapping of the task to processor to reduce communication traffic, and used RC6 for message encryption. To quantify security service strengths, the corresponding energy and time consumption are formulated. Discovering the optimal solution is complexity, to solve the problem, they proposed a heuristic security-aware and energy scheduling method based on GA. Furthermore, acceleration technologies are used to obtain fast convergence. The results have proven the efficiency of the proposed technology.

Previous works did not consider the influence of data size upon energy consumption in cryptographic algorithms. Guo *et al.*[22] calculated the energy consumption for security algorithms including asymmetric algorithms, symmetric algorithms, HMAC algorithms, hash algorithms in the real embedded platform from the perspective of integrity and confidentiality. Besides, they discussed the influences of the safety parameters including the size of key and encryption mode on energy cost. Furthermore, the impact of the size of data on energy cost is analyzed in detail.

With the effective and correct data on timing and energy security methods, the application of cryptographic methods can be extended into RTESs. Jiang *et al.*[23] studied the timing and energy performance of the cryptographic methods. They focused on the impact of the protected data size on time and energy related factors in RTESs, and their objective is to extend the application of cryptographic methods into embedded and security-critical RTSs. To deeply study the timing and energy performance of the cryptographic methods, they presented a multi-dimensional analysis framework from the perspective of energy cost, processing speed and battery power. To simultaneously execute the data collection of voltage and current, they developed a special data collection program based on LabVIEW. The experimental results showed that: (1) the cost of energy is close to the execution time of every encryption method under any size of data with the hardware/software experimental platform limitations, (2) the Gaussian function of data size is referred to power, (3) the unit energy and the speed of processing time are polynomial functions of the size of data.

Unlike the previous energy minimization problem, Jiang *et al.*[24] introduced a novel optimization problem for a series of tasks with multiple security choices and energy constraints. They solved one common resource allocation problem in SCESs to guarantee the security performance while meeting the energy constraints. This problem is a NP problem. Thus, they transformed the problem into a multi-stage decision process and used dynamic programming to address the problem. Based on the two-dimensional states' presentation and random scaling strategy, dynamic programming based on random scaling (RSDP) is developed. The main goal of RSDP is to allocate the most appropriate level of security to each task for minimizing the system security risk while meeting energy budgets. Result showed that RSDP is implemented by decreasing the number of decision states in each phase. RSDP can be easily applied to solve other optimization methods with statistical assurance requirements.

4.2. *Energy optimization under security constraints*

As mentioned above, security protection techniques usually consumes a lot of energy. The rapid depletion of energy or early depletion of the battery can cause mission-critical tasks to fail, resulting in unexpected losses. Therefore, designing energy-saving RTSs has become a necessity.

Poudel *et al.*[25] proposed two new secure MPSoC-based ECU architectures which can effectively satisfy the real-time and security requirements for automotive CPS while meeting the energy constraints. The first architecture is a ECU design (GED) based on GPGPU, which utilizes a coprocessor based on GPGPU and an application processor based on ARM. The second architecture is a RED that integrates the coprocessor based on FPGA with the application processor based on ARM. The outstanding features of GED and RED are: (1) integration of reliability and security primitives while complying with the strict real-time limits of automotive CPS; (2) the capability to implement computation intensive applications in an energy-saving way; (3) the scalability and flexibility achieved by reprogramming and (4) the prevention of the ECU to failure injection and analysis attacks. The experimental results showed that RED and GED consume less energy. In addition, both RED and GED can tolerate more transient failures.

Considering the security and dependability of the general application model with the RTESs supported by DVFS, Jiang *et al.*[26] proposed a vulnerability-based technique to quantify the performance of security of distributed systems communications, and solved problems for security critical systems which simultaneously takes security, dependability, timing and energy requirements into account. In order to incorporate authentication, integrity and confidentiality into their model, they extended this model by considering a model similar to Ref. 52. Their final goal is to minimize the application energy consumption, and the operating mode can be scaled down for tasks to save energy by DVFS. Due to the complexity of this goal, they proposed a hybrid optimization (TSHO) based on tabu search method, which determines the start time of the task, the voltage level, the transmission message time and its security levels to minimize the energy consumption while meeting security, dependability and timing requirements of the application. Experimental results showed that TSHO can ensure all the security and dependability requirements while providing low energy consumption.

Zhang *et al.*[27] focused on the mixed-security-critical distributed RTES (MSCDRES), and they achieved the goal of security assurance and energy balancing in the realization of task mapping. In the processing of solving the goal, they formulated a model of energy consumption, as the related constraints in this model was complicated, minimizing energy consumption is difficult. To handle this challenging issue, they proposed a heuristic algorithm based on GA named ESAMA that begins with a large number of task mappings produced by the greedy strategy. In ESAMA, the task mapping of the solution to the problem is configured as a chromosome. Experimental results showed ESAMA is efficiency that can trade off minimal energy consumption while guaranteeing the related constraints.

Jiang *et al.*[28] solved the uniprocessor scheduling problems in many security-critical RTSs designs with limited energy budget, and scheduled a series of periodical real-time tasks whose purpose is to minimize energy consumption. The energy consumption of each task contains the extra security protections and normal execution. They used the system monitor to determine the security risk bound (RB) which represents the present security requirement of the entire system, if the system fails to meet the expected RB, then it is considered as an unacceptable system. As the task and security options increase, the state space exponentially increases. To solve this problem, they grouped the security risk into a series of discrete integers to decrease the decision states at each stage. Then, they proposed an approximation algorithm based on RN (RNAA) to allocate the most appropriate security level for each task to minimize energy consumption while meeting the security requirements. Results showed that the completely time polynomial complexity and memory overhead of RNAA are both low.

Jiang *et al.*[29] firstly presented a new optimization problem for modern multi-mode RTESs with FPGA coprocessors where the task sets are dynamic. Their goal is to guarantee the minimum energy consumption while guaranteeing the security requirements regardless of the mode or safety requirements of the system. They divided this optimization problem into two sub-problems, the runtime and design time optimization. For design time optimization, they chosen NSGA-II based on GA to generate the approximate optimal Pareto curve for each probed mode, and the solutions obtained must meet the FPGA area constraints and schedulability constraints. For runtime optimization, they proposed a greedy approach to efficiently obtain a good performance.

In order to understand how to implement adaptable security mechanisms and indicate the impact on energy consumption better. Nicolae *et al.*[30] proposed a self-adaptable security infrastructure for the embedded systems to minimize energy consumption and provide a basic operational structure. The system goal is to maximize the operating time while keeping the predefined system security levels. This infrastructure contains analysis, enforcement and sensing blocks, when the sensing block discovers the defined parameters in the data sets have changed, it will inform the analysis block to decide what changes must be made in security requirements which are defined in the system goal. At last, the enforcement block will employ changes to the system. The results showed the energy consumption is lower at runtime in the proposed infrastructure, and important energy savings can be achieved even if the security is adapted to the needs of the system in a simple way.

5. Special Applications

With the rapid development of computer network technology, RTSs applications are increasing getting utilized. However, more and more new applications in RTS require sensitive information management whiling meeting the time requirements. Therefore, security requirements need to be incorporated into these real-time applications. Typical applications include CPSs, automotive systems, clusters, grids and so on. The classifications are listed in Table 5.

Table 5. Specific applications.

Classification	References	Approaches
CPS	Abad *et al.*[31] Mohan *et al.*[32] Easwaran *et al.*[33]	System-level simplex extended technique S3A Security analysis structure
Automotive systems	Lin *et al.*[34,35] Kiran *et al.*[36] Apvrille and Roudier[37]	Security-aware design technique Integrated MILP formulation SysML-Sec
Clusters	Xie *et al.*[38] Xie and Qin[39] Tripathi *et al.*[40]	Security overhead model TAPADS Schedule algorithm
Grids	Xie and Qin[41] Singh *et al.*[42] and Surendra *et al.*[43]	SAREG SDSA u-SDSA

5.1. *CPSs*

As systems control and remote coordination trends of large-scale integration, system security has become a growing interest in many real-time CPSs application scenes. The open operating environment of a closed system improves the performance of the system, it also exposes several new attack surfaces in the system. Thus, it is necessary to understand the role of hardware/software components in a real-time CPS as well as how those components expose the CPS to a malicious attacker.

Software components violate the security requirements when unintended deviations present, but this problem in CPSs can be addressed by the improved Simplex architectures. Based on the work,[53] Abad *et al.*[31] proposed a new technique that extends system-level simplex[54] to guarantee the runtime of the system with hard constraints and used platform-wise resets to recover a fully operational state from the faults in CPS. They extended real-time reachability to check the security properties with timing constraints and proved that these security properties not only rely on the system current state as mentioned in Ref. 53, but also rely on its history. The experimental results showed that restarting at runtime is a feasible approach to recover from the faults while guaranteeing the timing and security constraints.

Traditionally, hardware components have been believed to be highly secure, researchers only consider the security of software components. Since the W32. Stuxnet worm attack[55] showed that malicious code can target hardware components easily, hardware-based controllers are not safe either. Mohan *et al.*[32] proposed the secure system simplex architecture (S3A), a novel framework which combined trusted hardware, kind side-channels, operating system methods and the internal real-time functions of such systems to find out intrusions, and protect physical devices to enhance the security of the controllers based on hardware. Another important feature of this framework is that there is no need to modify the source code. Those new architectures and techniques can improve the difficulty faced by potential attackers significantly, thereby enhancing the security of the systems.

Real-time CPSs attacks may only affect the timing behavior, which results in the missing deadlines of the system hardware/software components without effecting functionality, and then leads to serious consequences. To solve this challenging problem, Easwaran *et al.*[33] proposed a new security analysis structure by utilizing ASD, which is displayed by finding new attack possibilities and mapping the existing attack to link the intermediate components, an attack's ultimate performances and sources, to describe the attack surface in real-time CPS clearly. ASD stands for standard sequence diagram which captures and illustrates the deadline attacks, and it consists of a set of objects which can communicate with each other by message. In ASD for a RTS, these objects are: scheduler and resource-sharing protocol (kernel components), hardware with other elements (resources) and applications and tasks (users).

5.2. *Automotive systems*

Through kinds of interfaces, such as short-range wireless access, long-range wireless channels and direct or indirect physical, it is easy to attack modern vehicles which is a security threat security in automotive systems.[56] The systems must take security mechanisms into consideration to prevent this attack, but the overhead of these mechanisms, such as confidentiality, authenticity, integrity, or availability, may thwart the system performance and violate design constraints. Therefore, it is very important to develop an approach to handle security at early design stages as well as all other design constraints. Moreover, the complexity of the system components also makes it necessary to verify that the requirements are agreed with and met by the initial design before any promise to a specific implementation is made.

According to TDMA and the CAN-based protocol[57] results, Lin *et al.*[34,35] proposed a common security-aware design technique named TDMACAN for dealing with the security problems under the timing constraints and limited resources. TDMACAN relies on the design in Refs. 58 and 59, where the architectural platform and the functional model were initially caught separately and merged by the mapping process. Unlike traditional method, TDMACAN not only maps the functional model to the architectural platform, but also discusses the selection of security mechanism and the selection of the architecture. To solve the mapping problem, they proposed a three-step approach, where each step expresses the security mapping problem as a MILP problem.[60] Experiments showed that the TDMACAN in system design were effective without breaking design constraints and also demonstrated that it was essential to consider the security during the design stages.

To solve the remaining problem in Refs. 34 and 35, Kiran *et al.*[36] proposed an integrated MILP formulation. They used three AT89C51 micro-controllers, each one uses receiver pins of its respective micro-controllers and transmitter to connect. The proposed formulation provides an optimal solution but has a high complexity. To solve complex problems, they also proposed a three-step approach, unlike the approach above,[34,35] here each step addresses some mapping problem in a brief MILP formulation. Experimental results showed that the approach achieves a comparable result quality compared with the method[34,35] and is more efficient.

The mapping of the security logic components or functional to hardware architecture often causes design faults due to the lack of understanding. Apvrille and Roudier[37] presented SysML-Sec, which enables security experts to cooperate with system designers to develop and design the embedded system. SysML-Sec was based on OMG's SysML and stood by open-source toolkits which is based on an accepted safety verification toolkit, and it associated software/hardware co-design with the process of security problems. Based on an iterative process in the center of the hardware/software architecture, the extension of SysML solves the security concern, and exported the captured requirements into safety-and-cryptographic mechanisms and safety attributes which can be formally validated. In the range of the EVITA project,[62] this method has been used in the safety design of automotive embedded systems.

5.3. *Cluster*

In the past decade, clusters are becoming increasingly popular as cost-effective and powerful platforms for running parallel applications. As sensitive data require special protection and safe-guard against unauthorized access, security has become a key issue for real-time applications on clusters. Recognizing that, many researchers have investigated real-time applications on clusters in terms of security requirements.

Xie *et al.*[38] presented a security overhead model after investigating the scheduling problem of many real-time tasks with different security requirements to measure the security overheads caused by security critical tasks. To integrate security requirements into scheduling of the real-time applications on clusters by using the proposed model, they proposed a security-aware scheduling strategy named SAREC, which realizes high security quality for real-time applications on clusters and meets its timing constraints. SAREC-EDF incorporates EDF algorithm into SAREC to evaluate the performance of SAREC, and the experimental results showed that SAREC-EDF achieved a significant improvement during the overall system performance.

Xie and Qin[39] not only addressed the precedence relations, but also solved the tasks' allocation problem in clusters for parallel applications with timing and security constraints. They proposed a task parallel method called TAPADS to combine time and security into the procedure of making allocation decisions. TAPADS uses the security level refinement and critical path analysis to maximize schedulability, by measuring the probability of satisfying the task's deadline and security quality by measuring risk-free probabilities. Experimental results also showed that the performance of the cluster is significantly improved in terms of schedulability and security quality by using TAPADS. Therefore, TAPADS is able to realize high security quality for real-time applications while satisfying the timing constraints.

Besides security and timeliness, efficiency is also an important property in applications running on clusters. Specifically, an efficient scheduling method achieves better performance according to the accepted security value and the number of tasks received. Tripathi *et al.*[40] proposed a schedule algorithm to maximize the success ratio, while meeting the task's minimum security on clusters, by introducing the deferred method and load balancing, and to minimize the preemption overhead by balancing the load of nodes on clusters. The proposed scheduling method first considers the timing requirements, and the rest of the requirements are subject to quality constraints. Results showed that the success ratio is improved while meeting the task's minimum security.

5.4. *Grids*

Today, grids are becoming the most cost-effective scheme for complex scientific computing and real-time applications. In addition to providing high QoS, such systems also provide a high throughput. However, common real-time scheduling algorithms such as the work[62] cannot meet the security requirements of real-time applications on grids.

Xie and Qin[41] presented a dynamic scheduling algorithm named SAREG, which is able to maximize the level of security of each accepted job while keeping very high assurance ratios for real-time applications on grids. They introduced a novel performance indictor-security value by measuring the security quality experienced by the whole real-time tasks whose deadlines are able to be met. To improve the practicability of SAREG, they proposed a security mathematical model that describes the scheduling framework formally.

Singh *et al.*[42] proposed a secure dynamic scheduling algorithm named SDSA for real-time applications on grids. The security lever in SDSA is dynamic according to the calculation of the time and the deadline of the packet while keeping the satisfaction of the QoS. In addition, SDSA forms a complete graph in an exchanged network environment that is connected via an agent node. Then, SDSA uses computing elements or grids to make sure delivery with the best security by using Per-packet encryption[63] and delivers the packets un-serviced to the next neighbor node that has the minimum queue length. The results showed that SDSA made sure fairness is relative to the completion time as well as security level of the packets.

However, SDSA does not consider the following two matters: (1) a node may have a high load as well as a small queue length and (2) how to select packets when more than two packets reach the same deadline. Hence, Surendra *et al.*[43] proposed a secure and dynamic u-SDSA based on utilization control, which uses utilization in place of queue length as the determinant of system load. It sends the package with minimum security level to one of the neighboring nodes with the lowest utilization value. Different from SDSA, u-SDSA can not only change the new packets security lever dynamically for real-time applications on grids to guarantee security and scheduling, but also maximize the guarantee ratio while maximizing the security level and best overall performance on average. The reason is that u-SDSA distributes incoming packets based on the utilization of nodes. Results showed that the u-SDSA algorithm is better than SDSA[64,65] and improves the overall performance.

6. Conclusions and Future Directions

In this paper, we investigate the security issues that exist in RETSs. We introduce several security-related technologies and specifically describe the current research status of these technologies. Some researchers simulated the behavior of a particular attack and analyzed it, which allows the designers of the RETSs to provide security advices at the design stage. Some researchers captured a specific attack, then established an analysis framework to analyze it and used a different security processor in RETSs to resist it. Besides studying the action of the attacks, another approach to realize a secure system is by introducing security services in the RETSs such as different encryption algorithms for encrypting the data.

Enhancing the security will inevitably be the sacrifice part of the energy, as the embedded system energy is limited, so the researchers must take the energy constraints into account. Some systems require strict energy requirements, so these systems need to design energy as a goal, security and real-time feature as the constraints. Some embedded systems require strict security requirements, so security is required as a primary goal, energy and real-time feature as the constraints. As RTESs are becoming more popular in various fields, we list several common applications, for example CPSs and automotive systems. Knowing these details is a great help in studying the security of embedded systems.

We believe that the future of embedded systems will face more and more serious security problems. To resist those attacks, strengthen security services, research energy efficiency and security efficiency will help the development of RETSs.

Acknowledgments

The authors are very grateful to the editor and reviewers for their suggestions to improve the quality of paper. This work was supported by the National Natural Science Foundation of China (61272420 and 61502234), and the Natural Science Foundation of Jiangsu Province of China (BK20150785).

References

1. C. Y. Chen, A. E. Ghassami and S. Mohan, A reconnaissance attack mechanism for fixed-priority real-time systems (2017) 02561, available at https://arxiv.org/pdf/1705.
2. C. Y. Chen, G. Amiremad and N. Stefan, Schedule-based side-channel attack in fixed-priority real-time systems (2015) 1–20, available at http://hdl.handle.net/2142/88344.
3. K. Jiang, L. Batina and P. Eles, Robustness analysis of real-time scheduling against differential power analysis attacks, *IEEE Computer Society Annual Symp. VLSI* (Tampa, FL, USA, 2014), pp. 450–455.
4. D. D. Hwang, K. Tiri and A. Hodjat, AES-based security coprocessor IC in 0.18-*muhboxm*CMOS with resistance to differential power analysis side-channel attacks, *IEEE J. Solid-State Circuits* **41** (2006) 781–792.
5. C. Bao and A. Srivastava, A secure algorithm for task scheduling against side-channel attacks, *Proc. 4th Int. Workshop Trustworthy Embedded Devices* (Arizona, USA, 2014), pp. 3–12.
6. N. Druml, M. Menghin and D. Kroisleitner, Emulation-based fault effect analysis for resource constrained, secure, and dependable systems, *Euromicro Conf. Digital System Design IEEE Computer Society* (Los Alamitos, CA, USA, 2013), pp. 337–344.

7. K. Patel, S. Parameswaran and S. L. Shen, Ensuring secure program execution in multiprocessor embedded systems: A case study, *IEEE/ACM/IFIP Int. Conf. Hardware/Software Codesign and System Synthesis* (Salzburg, Austria, 2007), pp. 57–62.
8. K. Patel and S. Parameswaran, SHIELD: A software hardware design methodology for security and reliability of MPSoCs, *ACM/IEEE Design Automation Conf.* (Anaheim, CA, USA, 2008), pp. 858–861.
9. R. Pellizzoni, N. Paryab and M. Yoon, A generalized model for preventing information leakage in hard real-time systems, *IEEE Real-Time and Embedded Technology and Applications Symp* (Seattle, WA, USA, 2015), pp. 271–282.
10. F. Abdi, M. Hasan and S. Mohan, ReSecure: A restart-based security protocol for tightly actuated hard real-time systems, *Workshop on Security and Dependability of Critical Embedded Real-Time Systems*, 2016, pp. 47–54.
11. M. Hasan, S. Monowar and R. Pellizzoni, Contego: An adaptive framework for integrating security tasks in real-time systems (2017), available at https://arxiv.org/pdf/1705.00138.
12. Y. Ma, W. Jiang and N. Sang, An adaptive risk control and security management for embedded real-time system, *Seventh Int. Conf. Availability, Reliability and Security* (Prague, Czech Republic, 2012), pp. 11–17.
13. N. Paryab, Integrating security mechanisms in hard real-time systems (2017), available at http://hdl.handle.net/10012/9566.
14. B. Tan, M. Biglari-Abhari and Z. Salcic, Towards decentralized system-level security for MPSoC-based embedded applications, *J. Syst. Archit.* **80** (2017) 41–55.
15. M. Saadatmand, T. Leveque and A. Cicchetti, Managing timing implications of security aspects in model-driven development of real-time embedded systems, *Int. J. Adv. Secur.* **5** (2012) 68–80.
16. X. Zhang, J. Zhan and W. Jiang, A vulnerability optimization method for security-critical real-time systems, *IEEE Eighth Int. Conf. Networking, Architecture and Storage* (Xi'an, China, 2013), pp. 215–221.
17. K. Jiang, P. Eles and Z. Peng, Co-design techniques for distributed real-time embedded systems with communication security constraints, *Design, Automation and Test in Europe Conf. Exhibition* (Dresden, Germany, 2012), pp. 947–952.
18. W. Jiang, Y. Ma and X. Zhang, Adaptive security management of real-time storage applications over NAND based storage systems, *J. Netw. Comput. Appl.* **52** (2015) 139–153.
19. S. Mohan, M. Yoon and R. Pellizzoni, Integrating security constraints into fixed priority real-time schedulers, *Real-Time Syst.* **52** (2016) 644–674.
20. W. Jiang, X. Zhang and J. Zhan, Design optimization of secure message communication for energy-constrained distributed real-time systems, *J. Parallel Distrib. Comput.* **100** (2016) 1–15.
21. J. Zhou *et al.*, Thermal-aware task scheduling for energy minimization in heterogeneous real-time MPSoC systems, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **35** (2016) 1269–1282.
22. Z. Guo, W. Jiang and N. Sang, Energy measurement and analysis of security algorithms for embedded systems, *IEEE/ACM Int. Conf. Green Computing and Communications* (Sichuan, China, 2011), pp. 194–199.
23. W. Jiang, Z. Guo and Y. Ma, Measurement-based research on cryptographic algorithms for embedded real-time systems, *J. Syst. Archit.* **59** (2013) 1394–1404.
24. W. Jiang, K. Jiang and Y. Ma, Resource allocation of security-critical tasks with statistically guaranteed energy constraint, *IEEE Int. Conf. Embedded and Real-Time Computing Systems and Applications*, 2012, pp. 330–339.

25. B. Poudel, N. K. Giri and A. Munir, Design and comparative evaluation of GPGPU- and FPGA-based MPSoC ECU architectures for secure, dependable, and real-time automotive CPS, *IEEE 28th Int. Conf. Application-Specific Systems, Architectures and Processors*, 2017, pp. 29–36.
26. W. Jiang, P. Pop and K. Jiang, Design optimization for security- and safety-critical distributed real-time applications, *Microprocess. Microsyst.* **52** (2017) 401–415.
27. X. Zhang, J. Zhan and W. Jiang, Design optimization of security-sensitive mixed-criticality real-time embedded systems, *Proc. ReTiMiCS* (Taipei, Taiwan, 2013).
28. W. Jiang, K. Jiang and X. Zhang, Energy optimization of security-critical real-time applications with guaranteed security protection, *J. Syst. Archit.* **61** (2015) 282–292.
29. K. Jiang, A. Lifa and P. Eles, Energy-aware design of secure multi-mode real-time embedded systems with FPGA co-processors, *Int. Conf. Real-Time Networks and Systems ACM* (New York, NY, USA, 2013), pp. 109–118.
30. B. Nicolae, A. Botezatu and V. I. Manta, Self-adaptable security architecture for power-aware embedded systems, *Proc. 14th Int. Conf. System Theory and Control*, Sinaia, Romania, 2010, pp. 98–103.
31. F. A. T. Abad, R. Mancuso and S. Bak, Reset-based recovery for real-time cyber-physical systems with temporal safety constraints, *IEEE Int. Conf. Emerging Technologies and Factory Automation* (Berlin, Germany, 2016), pp. 1–8.
32. S. Mohan, S. Bak and E. Betti, S3A: Secure system simplex architecture for enhanced security and robustness of cyber-physical systems, *Proc. ACM Int. Conf. High Confidence Networked Systems*, 2012, pp. 65–74.
33. A. Easwaran, A. Chattopadhyay and S. Bhasin, A systematic security analysis of real-time cyber-physical systems, *Asia and South Pacific Design Automation Conf.* (Chiba, Japan, 2017), pp. 206–213.
34. C. W. Lin, B. Zhang and Q. Zhu, Security-aware design methodology and optimization for automotive systems, *ACM Trans. Des. Autom. Electron. Syst.*, Vol. 21 (ACM, New York, NY, USA, 2015), pp. 1–26.
35. C. W. Lin, Q. Zhu and C. Phung, Security-aware mapping for CAN-based real-time distributed automotive systems, *IEEE/ACM Int. Conf. Computer-Aided Design*, 2013, pp. 115–121.
36. D. K. Kiran and L. K. Akalpita, Implementation of real-time distributed automotive systems for security on the basis of controller area network, *Int. J. Eng. Sci. Comput.* **6** (2016) 6499–6503.
37. L. Apvrille and Y. Roudier, SysML-See: A SysML environment for the design and development of secure embedded systems, *Proc. IN-COSE/APCOSEC Conf. System Engineering* (Yokohama, Japan, 2013).
38. T. Xie, X. Qin and A. Sung, SAREC: A security-aware scheduling strategy for real-time applications on clusters, *Int. Conf. Parallel Processing IEEE Computer Society*, 2005, pp. 5–12.
39. T. Xie and X. Qin, A new allocation scheme for parallel applications with deadline and security constraints on clusters, *IEEE Int. Conf. Cluster Computing* (Burlington, MA, USA, 2005), pp. 1–10.
40. S. Tripathi, R. S. Yadav and R. P. Ojha, A utilization based approach for secured real time applications on clusters, *Int. Conf. Advances in Computing, Control, and Telecommunication Technolgies (ACT)*, Trivandrum, Kerala, 2009, pp. 433–438.
41. T. Xie and X. Qin, Enhancing security of real-time applications on grids through dynamic scheduling, *Int. Conf. Job Scheduling Strategies for Parallel Processing* Vol. 3834, 2005, pp. 219–237.

42. S. Singh, S. Tripathi and S. Batabyal, *Secured Dynamic Scheduling Algorithm for Real-time Applications on Grid* (Springer International Publishing, Jaipur, India, 2016).
43. S. Surendra, S. Tripathi and S. Batabyal, Utilization based secured dynamic scheduling algorithm for real-time applications on grid (U-SDSA), *IEEE Int. Conf. Advanced Information NETWORKING and Applications* (Taipei, Taiwan, 2017), pp. 606–613.
44. J. Demme, R. Matrin and A. Waksman, Side-channel vulnerability factor: A metric for measuring information leakage, *Annual Int. Symp. Computer Architecture* (Portland, OR, USA, 2012), pp. 106–117.
45. J. Zhou *et al.*, Energy-adaptive scheduling of imprecise computation tasks for QoS optimization in real-time MPSoC systems, *IEEE/ACM Design, Automation and Test in Europe* (Lausanne, Switzerland, 2017), pp. 1402–1407.
46. H. Mannila, H. Toivonen and A. I. Verkamo, Discovery of frequent episodes in event sequences, *Data Min. Knowl. Discov.*, Vol. 1 (Kluwer Academic Publishers Hingham, MA, USA, 1997), pp. 259–289.
47. C. Braune, S. Besecke and R. Kruse, Density based clustering: Alternatives to DBSCAN, *Partitional Clustering Algorithms* (Springer International Publishing, Switzerland, 2015), pp. 193–213.
48. J. Song, J. Wittrock and G. Parmer, Predictable, efficient system-level fault tolerance in C^3, *Real-Time Systems Symp.*, Vol. 7975 (Vancouver, BC, Canada, 2013), pp. 21–32.
49. J. Zhou *et al.*, Fault-tolerant task scheduling for mixed-criticality real-time systems, *J. Circuits Syst. Comput.* **26** (2017) 1–17.
50. M. Hasan, S. Mohan and R. B. Bobba, Exploring opportunistic execution for integrating security into legacy hard real-time systems, *IEEE Real-Time Systems Symp.*, 2017, pp. 123–134.
51. P. Eles, A. Doboli and P. Pop, Scheduling with bus access optimization for distributed embedded systems, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **8** (2000) 472–491.
52. Z. Guo, W. Jiang and N. Sang, Energy measurement and analysis of security algorithms for embedded systems, *IEEE/ACM Int. Conf. Green Computing and Communications* (Sichuan, China, 2011), pp. 194–199.
53. S. Bak *et al.*, Real-time reachability for verified simplex design, *IEEE Real-Time Systems Symp.* (Rome, Italy, 2014), pp. 138–148.
54. S. Bak, D. K. Chivukula and O. Adekunle, The system-level simplex architecture for improved real-time embedded system safety, *IEEE Real-Time and Embedded Technology and Applications Symp.* (San Francisco, CA, USA, 2009), pp. 99–107.
55. N. Falliere, L. O. Murchu and E. Chien, W32.stuxnet dossier, White Paper (2011).
56. B. Carnevale *et al.*, MACsec-based security for automotive ethernet backbones, *J. Circuits Syst. Comput.* **27** (2017) 1–17.
57. A. Diaz, J. Gonzlezbayon and P. Snchez, Security estimation in wireless sensor network simulator, *J. Circuits Syst. Comput.* **25** (2016) 1–18.
58. J. Zhou *et al.*, Thermal-aware correlated two-level scheduling of real-time tasks with reduced processor energy on heterogeneous MPSoCs, *J. Syst. Archit.* **82** (2017) 1–11.
59. A. Sangiovanni-Vincentelli, Quo vadis, SLD? Reasoning about the trends and challenges of system level design, *Proc. IEEE* 95 (2007) 467–506.
60. C. W. Lin, Q. Zhu and C. Phung, Security-aware mapping for CAN-based real-time distributed automotive systems, *IEEE/ACM Int. Conf. Computer-Aided Design*, 2013, pp. 115–121.
61. A. Ruddle *et al.*, Security requirements for automotive on-board networks based on dark-side scenarios, Technical Report Deliverable D2.3, EVITA Project, (2009).

62. S. H. Son, R. Mukkamala and R. David, Integrating security and real-time requirements using covert channel capacity, *IEEE Trans. Knowl. Data Eng.* **12** (2000) 865–879.
63. Y. Jung and E. Festijo, Securing RTP packets using per-packet selective encryption scheme for real-time multimedia applications, *ETRI J.* **35** (2013) 726–729.
64. S. Singh, S. Tripathi and S. Batabyal, Secured dynamic scheduling algorithm for real-time applications on grid, *Int. Conf. Information Systems Security* (Jaipur, India, 2016), pp. 283–300.
65. T. Wu *et al.*, Soft error-aware energy-efficient task scheduling for workflow applications in DVFS-enabled cloud, *J. Syst. Archit.* **84** (2018) 12–27.