Journal of Systems Architecture 61 (2015) 282–292

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Energy Optimization of Security-Critical Real-Time Applications with Guaranteed Security Protection ☆

Wei Jiang [a,*], Ke Jiang [b], Xia Zhang [c], Yue Ma [d]

[a] School of Information and Software Engineering, University of Electronic Science and Technology of China, China
[b] Department of Computer and Information Science, Linköping University, Sweden
[c] Department of Computer Science, University of Texas at Dallas, USA
[d] Department of Computer Science and Engineering, University of Notre Dame, USA

ARTICLE INFO

Article history:
Received 8 December 2014
Received in revised form 13 April 2015
Accepted 27 May 2015
Available online 12 June 2015

Keywords:
Embedded system
Security
Energy
Real-time
Design optimization

ABSTRACT

Designing energy-efficient applications has become of critical importance for embedded systems, especially for battery-powered systems. Additionally, the emerging requirements on both security and real-time make it much more difficult to produce ideal solutions. In this work, we address the emerging scheduling problem existed in the design of secure and energy-efficient real-time embedded systems. The objective is to minimize the system energy consumption subject to security and schedulability constraints. Due to the complexity of the problem, we propose a dynamic programming based approximation approach to find efficient solutions under given constraints. The proposed technique has polynomial time complexity which is half of existing approximation approaches. The efficiency of our algorithm is validated by extensive experiments and a real-life case study. Comparing with other approaches, the proposed approach achieves energy-saving up to 37.6% without violating the real-time and security constraints of the system.

1. Introduction

Real-time embedded systems are facing more and more severe security threats [1], e.g., due to the integration of new communication interfaces. One of the emerging needs is to protect sensitive data in critical embedded systems [2]. Since snooping, spoofing and altering sensitive data can lead to significant information losses or serious system failures [3], resulting in great loss of finance or human lives. We refer to such systems as Security-Critical Real-Time Systems (SCRTSs). Examples of SCRTSs are flight control systems, satellite communication systems and radar tracking systems, which all have high security demands. To protect SCRTS against potential threats, a series of security services, i.e., integrity, confidentiality and authentication protection, need to be considered in the design process of SCRTS. With the best security protections selected with respect to concrete demands, SCRTSs would be effectively protected.

One primary obstacle against the development of SCRTS is energy consumption, as security protections usually demand a significant amount of energy or power expenditure [4]. Additionally, most of SCRTS are battery-powered and even implemented under no-nursing environment. Quick energy consumption or early exhaustion of batteries may lead to failure of critical tasks, resulting in unexpected outcomes, such as the energy incurred failure of Mar's Path Finder and NASA Spatial systems. Hence, the design of SCRTSs, considering security and energy factors together, has become of vital importance, and an imperative work to do.

In SCRTSs, the major challenge of delivering security protections lies in the conflicting interests among minimizing energy consumption, satisfying the real-time requirement and maximizing security protection. There exists many cryptographic algorithms suitable for SCRTSs, in which a general trend of the performance trade-off could be observed among security risk, energy overhead and execution time overhead under different security levels, as shown in Fig. 1. In general, the cryptographic algorithm with higher security level (depending on its robustness against attacks) can achieve higher security protection (lower security risk) at the cost of more computation time and energy. For example, the implementation of RC5 (Rivest Cipher 5) will consume much more execution time and energy than that of RC4 (Rivest Cipher 4). Task scheduling policy plays an important role for achieving high performance in real-time systems. Unfortunately, traditional real-time scheduling approaches were mostly designed to guarantee timing requirements only [5]. Recently, security-aware

☆ This paper is an extended version of a conference paper [1].
* Corresponding author.
E-mail address: weijiang@uestc.edu.cn (W. Jiang).

http://dx.doi.org/10.1016/j.sysarc.2015.05.005
1383-7621/© 2015 Elsevier B.V. All rights reserved.

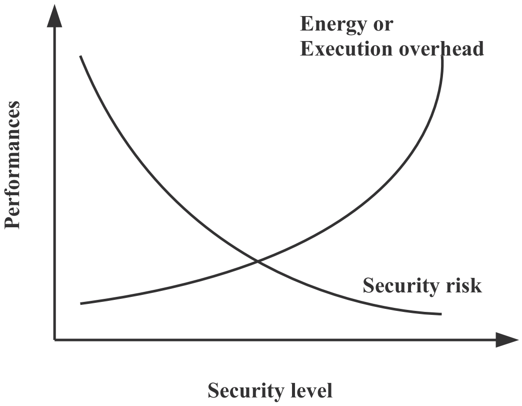

Fig. 1. Performance VS security level.

real-time scheduling has become a hot research topic [3,6–8]. However, all these works did not consider the energy aspect, which may deliver solutions with very high energy consumptions.

In this paper, we identify the uniprocessor scheduling problem lying in many SCRTS designs considering energy, security and real-time dimensions. More concretely, we aim to schedule a set of periodical real-time tasks with the objective of minimizing energy consumption, while satisfying security and timing constraints. Our approach has polynomial time complexity, and requires bounded memory space. The proposed approach is evaluated on extensive experiments and a real-life case study (a UAV application), and compared with other approaches from existing literatures.

The rest of this paper is organized as follows. Section 2 describes a motivational application and system model. Section 3 formulates the system problem. Sections 4 and 5 present our proposed scheduling mechanism and the simulation results, respectively. Section 6 evaluates the approach on a real-life application. Section 7 reviews the related work. Section 8 concludes this paper.

2. Application and system model

2.1. Motivational application

In this paper we focus on the SCRTSs with limited energy budget, for example, an Unmanned Aerial Vehicle (UAV) depicted in Fig. 2. The UAV is battery driven, and is controlled by an embedded processor. It runs critical tasks that are periodically released, and exchanges information with other peers or service centers. Each task generates or receives some private data that needs to be transmitted over insecure environments. Different data has different requirements of security and deadline guarantee ratios. In order to make the communication secure (i.e., to protect the confidentiality or integrity of the messages), we need to perform cryptographic algorithms like RC5, DES and SHA-2, on the data before or after the normal executions of corresponding tasks. Thus, the energy consumption of each task consists of two parts that are from the normal execution and extra security protections (see Section 2.2). Although there are many available cryptosystems, it is hard to obtain the best choices among different solutions having different execution and energy overhead. Meanwhile, UAV only has limited energy and processing capability. Therefore, how to efficiently allocate resources to protect different data becomes an important design trade-off. In other words, we are aiming to schedule a set of periodic real-time tasks with the objective of minimizing energy consumption while satisfying security and schedulability constraints.

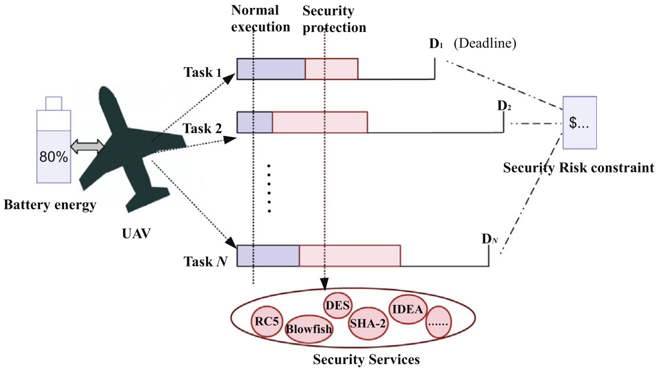

Fig. 2. A motivational application.

2.2. Task model for security-critical real-time systems

We consider a set of periodic security- and energy-aware tasks running on a uniprocessor architecture. Each task T_i is captured by seven design parameters, $T_i = \{BE_i, L_i, S_i, S_i^{DM}, V_i, SR_i, P_i\}$. BE_i denotes the worst case execution time (WCET) of its non-security part. L_i is the size of data that is generated or received by T_i, and needs to be protected using selected security service. S_i and S_i^{DM} are the chosen and designated security levels of T_i, respectively. If S_i^{DM} is achieved, this task is assumed to be absolutely secure. V_i is the security impact value of T_i representing the relative importance of the messages processed by T_i. SR_i is the security risk of T_i indicating the potential loss of the security protection, which will be elaborated in Section 2.4. P_i is the period and also the relative deadline of T_i.

2.3. Time and energy overhead of security critical task

It is known that security protections can be achieved by additional security services, which also compete resources with normal executions. For example, doing AES encryption on one message may reduce the available CPU resource for protecting other messages. So it is indispensable to always allocate the right amount of resource to the security protections among tasks in order to reach the best global security protection while delivering good performances to the tasks.

It is still an open problem of quantifying the security strength of different cryptographic algorithms. Different metric will result in different security level assignments and newly developed algorithms may have higher level but lower overhead like AES-128 in Table 1. So we enumerate the levels based on our reasoning of their security strengths [9,10] in this paper, but the designer can use his own assignments in our techniques. Based on the measurement on a S3C2440 ARM board with 500 MHz CPU and 64 MB SDRAM [11], we obtain the time and energy overheads of seven widely used confidentiality services for protecting 1 KB data (as shown in Table 1). For example, we assign security level 1 to the relatively weakest algorithm RC4 that has the shortest encryption time. In this paper, we only consider periodic tasks, so we assume that the key setup procedures for the security algorithms are prepared

Table 1
Time and energy overhead of confidentiality algorithms.

Ciphers	Time (ms/KB)	Energy (mJ/KB)	Sec. level
RC4	0.0063	2.0237	1
RC5	0.0125	4.0340	2
BLOWFISH	0.0170	5.4696	3
IDEA	0.0196	6.2822	4
SKIPJACK	0.0217	6.9658	5
3DES	0.0654	21.0914	6
AES-128	0.0194	6.2595	7