DREAMS: Secure Communication between Resource Management Components in Networked Multi-Core Systems

Thomas Koller*, Gautam Gala†, Daniel Gracia Pérez‡, Christoph Ruland* and Gerhard Fohler†
*Chair for Data Communications Systems, Universität Siegen, Siegen, Germany
{thomas.koller, christoph.ruland}@uni-siegen.de
†Chair of Real Time Systems, Technische Universität Kaiserslautern, Kaiserslautern, Germany
{gala, fohler}@eit.uni-kl.de
‡Thales Research & Technology, Palaiseau, France
daniel.gracia-perez@thalesgroup.com

Abstract—In mixed-criticality systems, resource management services are required to recognize and fulfill system wide high-level constraints, e.g., end-to-end deadlines. This is not possible through individual resources in isolation. Instead, a system-wide view is necessary which requires system-wide decisions. In the European FP7 project DREAMS, services for system-wide adaptability of mixed-criticality applications consuming several resources are provided via a hierarchical resource management. The resource management is a promising target for a passive as well as an active attacker since it deals with critical information of the system. The fact of having the authority to actively take decisions on resource allocation makes it an interesting target. Therefore, security mechanisms are required to ensure an adequate protection of the system’s resource management. This paper introduces the DREAMS secure resource management services, specifically secure communication between the resource managers. Furthermore, the resource management infrastructure is analyzed with respect to possible attacks and suitable countermeasures are discussed. A security library is developed in accordance to the identified countermeasures and implemented as a proof of concept.

Index Terms—mixed-criticality, security, resource management, service based architecture, cyber-physical systems

I. INTRODUCTION

Mixed-criticality systems are widely used in avionics, industrial control or health care where functions at different importance and certification-assurance levels are integrated on a shared computing platform. The objective of the European FP7 project Distributed REal-time Architecture for Mixed Criticality Systems (DREAMS) [1] is to develop a cross-domain architecture and design tools for complex networked systems with support for application subsystems of different criticality that are executed on networked multi-core chips. Resource management is one of the core services provided in DREAMS for a system wide adaptability of mixed criticality applications. System wide high-level constraints cannot be fulfilled by considering individual resources in isolation for complex systems. The approach used in DREAMS provides the system view on an abstract level. It reduces the overhead of distributing the global system state to all system components and only provides the information which is required for a system wide reconfiguration.

From a security point of view, the communication between the resource management components is a critical point of the system as it is a promising target for an attacker. Passive attackers have the possibility to get sensitive information of the system out of the messages being exchanged between the components. Active attackers can manipulate the system by taking false configuration decisions or by sending incorrect information to other components. Hence, secure communication services for the resource management are required to ensure that no confidential data is disclosed, a manipulation of the transferred messages is prevented and the origin of the message is verifiable. These secure communication services are provided by the security library presented in this paper.

The reminder of the paper is organized as follows. Section II presents an overview of related work. The resource management architecture used in DREAMS is described in Section III. Section IV examines attacks on the resource management, defines the required security services and describes the implementation of the security library. Experimental results are presented in Section V. Finally, Section VII concludes the paper and outlines future work.

II. RELATED WORK

An adaptive QoS framework for efficient resource management, called the Matrix approach is presented in [2]. Its goal is the efficient transport of streams with acceptable play out quality in a heterogeneous and dynamic environment by management of both CPUs and networks, based on a global abstraction of system state. The resource management approach in DREAMS is based on the resource management architecture of the Matrix framework. But it is adapted to the DREAMS platform, where several multi-core chips exist and applications can have various criticality levels. The global

The research leading to these results has received funding from the European Union’s Seventh Framework Programme FP7 2007-2013 under grant agreement 610640.
resource manager (GRM) of DREAMS is based on insights from ACTORS [3]. The resource management of the ACTORS project provides adaptability within a single device, based on abstract service levels of CPU availability, application demands and adaptability. Furthermore, the concept of service levels in ACTORS is extended in DREAMS to apply to virtualized hardware resources instead of applications. Reconfiguration issues for Integrated Modular Avionics (IMA) architectures have been explored in the SCARLETT project [4]. While SCARLETT is specifically IMA-based, the DREAMS approach is domain-independent. In addition, the reconfiguration in DREAMS addresses design faults in mixed-criticality systems and assesses real-time behavior. In the project ACROSS [5], the possibility to reschedule communication on the network-on-chip is implemented in the form of a “Trusted Resource Manager”. This component deals with reconfiguration decisions and propagates these to all other components on the local network-on-chip.

All of the previous works have not yet considered security of communication between the resource management components, however, security vulnerabilities can affect the safety of the overall system. There is a direct relation between security and safety. The relationship between safety and security is described in [6]. Beginning with the main definition of dependability, it describes the related attributes and the threats to dependability and security including failures, errors and faults. Design challenges for security in embedded systems are discussed in [7]. It describes the security requirements for such systems, explains the basic security concepts and gives examples of the influences of security on these systems, e.g., power consumption. Various authenticated encryption (AE) or authenticated encryption with associated data (AEAD) mechanisms are defined in [8]. It defines different mode of operation for block ciphers providing integrity, authenticity and confidentiality. An alternative construction of an AE mode is described in [9]. It uses the ChaCha20 cipher [10] and the Poly1305 message authentication code [11]. ChaCha20 is a stream cipher and a derivate of the Salsa20/12 algorithm [12] which is part of the eSTREAM portfolio [13].

III. RESOURCE MANAGEMENT IN DREAMS

The resource management services in DREAMS are realized by a GRM in combination with local building blocks for resource management. [14] The GRM gathers information from LRM and provides new configurations for the virtualization of resources (e.g., partition scheduling tables or resource budgets). The GRM configuration includes different precomputed configurations of resources (e.g., time-triggered schedules) or parameter ranges (e.g., resource budgets). Local resource management services consist of three major parts: LRM, local resource schedulers (LRSs) and resource monitors (MONs). The LRS performs the runtime scheduling of resource requests (e.g., execution of tasks on processor, processing of queued memory and I/O requests). It is responsible for controlling the access to a particular resource based on the configuration that has been set by the LRM in charge of the supervision and control of that resource. The LRSs in DREAMS supports different scheduling policies (e.g., dispatching of time-triggered actions, priority-based scheduling). The LRM either adopts the configuration from the GRM to particular resources (e.g., processor core, memory, I/O) or selects a new configuration from the ones available and then maps these decisions to the local scheduling policy of the LRS. A MON monitors the resource availability (e.g., energy). MONs also observe the timing of components (e.g., detection of deadline violations) and check the application behavior (e.g., stability of control). Small changes are handled locally by the corresponding LRM, while significant changes are reported to the GRM, who in turn can provide a different configuration at system-level. The resource management services can be arranged across the DREAMS platform in two main configurations: a flat or a hierarchical architectures as shown in Figure 1. All resource-management building blocks can have hardware or software implementations, or a combination of both, depending on the domain and type of the resource. The DREAMS middleware relies on time and space partitioning principles [15]. In this paper, it is considered that those principles are implemented at the chip level by the XTRATUM hypervisor [16].

A. Resource Management Communication

The XTRATUM hypervisor is responsible for transmitting the message between the resource management components. XTRATUM provides sampling and queuing communication ports. Sampling ports are used when the receiver should only get the newest message. The message remains in the port until the message is read by the receiver or the message is overwritten by a new one. Queuing ports are used when all of the messages should be processed by the receiver. If a new message is sent before a old one is read, the messages are buffered. The delivery of the message is done in FIFO order. Since the XTRATUM hypervisor itself does not ensure security of the messages exchanged between resource management components, all messages are sent via a security library.
implemented in the DREAMS middleware. As shown in Figure 2, three main types of channels are used.

The Updates Channel (64 bit queuing channel) is used by a LRM to send status updates and request global reconfiguration. Each update message consists of the configuration being executed by the LRM and the type of the update that is being transmitted, i.e., the reconfiguration request or status update. The Orders Channel (32 bit sampling channel) is used by the GRM to send reconfiguration orders to a LRM. An order message consists of the new reconfiguration to be applied by the destination LRM, and whether the reconfiguration should be applied immediately or at the end of the major cycle (hyperperiod). The Membership Channel (8 bit sampling channel) is used by LRM periodically to send an live signal to the GRM via this channels for membership purposes. If the GRM does not receive a signal from the LRM within each membership period, it may assume that the LRM is not working and triggers a reconfiguration.

IV. SECURE RESOURCE MANAGEMENT COMMUNICATION

An unsecured resource management has several weak spots that can be exploited by a malicious attacker. These attacks are categorized in attacks on the resource management components itself and in attacks on the communication process of the resource management services. Attacking specific resource management components, the attacker can masquerade as a GRM or LRM. This allows the attacker to select wrong scheduling tables or he can use invalid scheduling parameters. The same applies to configuration tables or parameters. Sending wrong information to the resource management component at the higher level, the resource management could take wrong decisions. In case of a GRM, for example, the attacker can send global reconfiguration messages and in case of an LRM, the attacker could locally reconfigure a subsystem and/or send incorrect status updates to the supervisor resource manager. Attacking the communication process of the resource management services, the attacker can perform a sniffing attack which provides him sensitive information about the behavior of the system. He only needs to intercept the transferred messages. Man-in-the-middle, spoofing and packet injection attacks lead to the same risks described as the attacks on specific components, e.g., the usage of wrong configurations and wrong scheduling tables. Here, the attacker has to masquerades himself as a GRM or an LRM. The same risks apply for replay attacks, but at least the configuration or scheduling was valid before. Nevertheless, the system or a part of the system will not operate as intended, e.g., on the basis of the old information, the GRM would select an inappropriate global configuration.

A. Security Services and Mechanisms

The attacks described above can be prevented using the following security services for resource management communication. Confidentiality, which ensures the privacy of information. Integrity, which ensures that data is not modified. Authenticity, which ensures that data is genuine and that the actual origin of the data is the same as the claimed origin. Access Control, which allows access based on permissions. These four security services are provided by the cryptographic mechanisms encryption and message authentication code: Encryption (and the reverse operation decryption) is used to ensure confidentiality. It protects against the sniffing or eavesdropping attacks described above. A message authentication code is used to ensure integrity, authenticity and, with some additional effort, to provide access control. In the proposed implementation, this mechanism ensures that data cannot be modified unnoticeably (integrity), that the data is genuine and that the actual origin of the data is the same as the claimed origin (the sender is part of the trusted group; authenticity) and that the sender is allowed to use the service (access control). The authenticity service in combination with access control protects against the attacks on specific resource management components described above. The attacker cannot masquerade as a valid resource management component. The same applies to man-in-the-middle, spoofing and packet injection attacks targeting the communication process of the resource management services. Without knowledge of the correct cryptographic key, the attacker is not able to act as a valid component, because the receiver of the message cannot decrypt or authenticate the message correctly. Replay attacks are prevented by using a time-varying parameter in the message authentication code protected message. The attacker cannot modify the time-varying parameter and the receiver can check the time-varying parameter of the received message to detect a replayed message. A mechanism using digital signatures could also be used to provide authenticity and access control. But because of performance reasons this asymmetric mechanism is not considered, as this is much slower than the usage of symmetric algorithms.

B. Security Service Classification

The common usage of the security properties are classified into four categories as presented in Table I. No security service is used on security level 0. To recognize manipulation of the data, integrity is provided on security level 1. Security level 2 provides authenticity in addition to integrity. This allows to detect a manipulation of data and the data origin as well as
the communication partner can be verified. Confidentiality in combination with integrity and authenticity is provided on security level 3. Confidentiality combined with integrity and authenticity because without integrity, it would not protect against data manipulation and authenticity is included in the mode of operation AE or AEAD. The usage of these modes saves the necessity to provide a fifth security level providing only integrity and confidentiality. Access control is not listed in the table because it is independent of the described security levels. If it is required, it can be used in parallel to the selected security level.

C. Security Requirements for the Channel Types

1) Updates Channel: The updates channel is used to send status updates and to request global reconfigurations. An attacker aiming on this channel can get information about the status and can perform manipulated or faked status updates or reconfiguration requests. Hence, integrity is required to protect against manipulations and authenticity is required to ensure the origin of the message. As the messages may contain sensitive information, confidentiality is also required. This causes the usage of security level 3 for the updates channel.

2) Orders Channel: The orders channel is used to send reconfiguration orders to the lower-level resource management component, e.g., from the GRM to an LRM. As correct reconfiguration orders are necessary to ensure the reliability of the system, this messages must not be manipulated or faked. The messages sent on the orders channel also may contain sensitive information. Hence, all security properties are required for the orders channel. This causes the usage of security level 3 for the orders channel just as for the updates channel.

3) Membership Channel: The membership channel is used to send a periodic life signal to the upper-level resource management component. For this channel type, it has to be ensured that the messages are not modified and that the actual origin of the message is the same as the claimed origin. No confidential data is sent through this channel. Hence, only integrity and authenticity are required on this channel. This causes the usage of security level 2 for the membership channel.

Replay attacks are prevented for all of the three channel types by using security level 2 or 3. From security level 2 onwards, a time-varying parameter is included which in turn is protected by a message authentication code.

D. Implementation

The security library can be implemented as a support library or as an additional sublayer. The Figure shows the data flow between the different (software) components. The main difference is the handling of the communication process. Implementing the security library as a support library, the resource management component has to take care of sending and receiving the messages. Secured messages have to be relayed to the security library. The resource management components itself are the staring and end points of the secure communication. Implementing the security library as an additional layer between the resource management components and the corresponding XTRATUM instances, the security library takes care of sending and receiving the messages. The security libraries are the staring and end points of the secure communication. Both approaches ensures a secure end-to-end communication. The advantage of the second option using the security library as an additional sublayer is the direct communication between the security libraries of the associated resource management components.

Figure 3 shows the data flow of the hierarchical communication between a GRM (left), a first level LRM (middle) and a second level LRM (right). Each resource management component has its own security library. Messages that are sent by the GRM to the first level LRM are secured by the security library of the GRM. The security library uses the ports provided by XTRATUM to send the message to the security library of the first level LRM. This security library decrypts the message and verifies the integrity and authenticity of it and forwards the message to the first level LRM. The same applies to the communication between the first level LRM and the second level LRM. The security library of the first level LRM has to handle both, the ports to the GRM and the ports to the second level LRM.

As described in Section IV-B an AE mode of operation is used. The AEAD construction ChaCha20-Poly1305 is used in the security library. Compared to a software implementation of AES-128-GCM, the ChaCha20-Poly1305 construction is about three times faster. [17] Using dedicated hardware for AES-128-GCM, AES is still faster. But in the DREAMS project, no AES hardware extension considered, because of the used heterogeneous hardware platforms (see Section VI).

The security level 3 frame format is shown in Figure 4. The frame is segmented into three parts. The first part is the header part which is not encrypted but authenticated by the MAC. It includes the length of the header (aad.length), the length of

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Security Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Security Service</td>
</tr>
<tr>
<td>1</td>
<td>Integrity</td>
</tr>
<tr>
<td>2</td>
<td>Integrity &amp; Authenticity</td>
</tr>
<tr>
<td>3</td>
<td>Integrity, Authenticity &amp; Confidentiality</td>
</tr>
</tbody>
</table>

Table 1: Classification of Security Services

![Figure 3. Secure hierarchical communication](image-url)
the ciphertext, a nonce which in turn includes a time-varying parameter, the source and the destination ID, the port descriptor, the port type and a flag field with variable length. The second part of the frame is the ciphertext containing the encrypted plaintext. The third part is the MAC, securing the first two parts. Security level 3 increases the message at least by 416 bit: 288 bit for the header and 128 bit for the MAC. The security level 2 frame is similar to the frame of security level 3. But instead of the ciphertext in the second part of the frame there is the plaintext. The MAC in the third part authenticates again both, the first part (header) and the second part (plaintext). Security level 2 increases the message at least by 416 bit: 288 bit for the header and 128 bit for the MAC. Security level 1 is not implemented in the security library. Both the required message size and the required calculation time to provide integrity are increased by the same amount as providing authenticity. So, instead of using security level 1, security level 2 can be used directly. In security level 0, no security mechanism is used. The header only contains the plaintext length. The second part of the frame is the plaintext itself and the third part, the MAC, is skipped. Security level 0 increases the message by 32 bit.

For all of these security levels, the maximum payload size is $2^{32}$ byte as the size of the ciphertext.length field is 32 bit.

V. Evaluation

The implementation of the security library is evaluated on the Xilinx ZC706 Evaluation Kit with the XC7Z045-2FFG900C All Programmable SoC [18]. The application processor unit of the ZC706 offers a dual-core Cortex-A9 MPCore CPU. On this CPU, the hypervisor XTRATUM is used. XTRATUM provides spatial and temporal isolation provided by partitions to execute a guest operating system or applications. For the evaluation, the resource management components are executed on different partitions. Two resource management components are used: a GRM and a LRM. The communication between these components is realized by the sampling and queuing ports provided by XTRATUM as described in Section III-A.

The message size overhead for the different levels of the security library was already mentioned in Section IV-D. On security level 0, the overhead is induced by the message length field in the header, so the message size increases slightly. Regarding the higher security levels providing integrity, authenticity and, on security level 3, confidentiality, there is no difference concerning the message size between using the security level 2 and the security level 3. The only difference between these two levels is the exchange of the plaintext on level 2 to the ciphertext on level 3. As the used cipher ChaCha20 is a stream cipher, no padding is required and the size of the ciphertext is the same as the size of the plaintext. Using a block cipher, the size would increase by filling the last block. Whereas there is no difference for the message size, the plaintext has to be encrypted on security level 3. This requires additional computation time. On all levels, the used CPU cycles are measured as shown in Table II. The CPU cycles are measured for the complete function call including XTRATUM inter-partition communication functions. The required computation time increases using a higher security level. The highest influence on the computation time has security level 2. Compared to sending a message without security library or with security level 0, there is a huge increase of the message size and the message authentication code has to be computed over the complete message. On security level 3, there is no additional increase of the message size and additionally, only the payload has to be encrypted before calculating the message authentication code over the complete message. For the updates channel, the security level 3 is required. Hence, the message size increases from 64 bit to 480 bit. Compared to sending an unsecured update message at an average of 11577 clock cycles, 83968 clock cycles are required to generate and send the secured message. For the orders channel, also the security level 3 is required. The message size is only 32 bit compared to the 64 bit of the updates channel. So, the message size increases to 448 bit and it requires 83356 clock cycles to send the secured message. This is close to the same as for the updates channel. The security level 2 is required on the membership channel. The 8 bit payload results in a 424 bit secured message. As the payload has not to be encrypted, only 60575 clock cycles are required to send the secured message. On the receiver side, nearly the same amount of clock cycles as on the sender side have to be used to calculate the message authentication code and compare it with the received one and, on security level 3, decrypt the message.

VI. USE CASE: AVIONIC DEMONSTRATOR

The DREAMS avionic demonstrator will highlight the security capabilities of the middleware. The demonstrator combines critical applications with non-critical applications using heterogeneous multi-core platforms, connected using a wired network. Five different applications will be deployed
in the demonstrator, three critical and two non-critical. The critical applications are a Flight Management System (FMS, described in [19]), a Display Management System (DMS), and a Sensors Data Provider (SDP). The non-critical applications are an In-Flight Entertainment (IFE) and a panel, the later with two instances, one for the IFE and another for the DMS. Additional non-critical applications (stressing benchmarks, SB) will be introduced to stress the system. All the applications, but the panels, sit on top of the DREAMS middleware and the XtratuM hypervisor, and they are deployed in two different hardware platforms: the Freescale T4240QDS and the DREAMS Harmonized Platform (based on the Xilinx ZC706 as used in Section V). Communications between the different hardware platforms are ensured by a TTETHERNET network [20]. Figure 5 shows one of the target deployments of the applications over the hardware platforms. Secure communications will be exploited by the LRM and GRM instances in the system as described in the previous sections. In addition, the secure communication library will be used for the communications between the critical applications, displayed as green arrows in Figure 5. For the LRM and GRM instances, the used security levels are defined in this paper. The security levels for the communication channels between the critical applications have to be defined during the development of the avionic demonstrator. The payload to the transferred messages between this application range between 32 byte and 3 kilobyte. Hence, the demonstrator will show the impact of the security library on different message sizes being larger than the message sizes of the resource management.

VII. CONCLUSION

This paper defines the communication architecture of the resource management in DREAMS and analyses it with regard to the security requirements. The resource management is a promising target for an attacker. The secure communication mechanisms presented in this paper protect the resource management against such malicious attacks. We described the resource management architecture used in DREAMS and the used communication channels as well as the content. After examining possible attacks on the resource management, the defined channels were classified into security levels to identify the required security mechanisms. Based on this requirements the security library was developed. The security library is not restricted to the usage for the resource management. In the avionic demonstrator of DREAMS, the security library will be used for varying applications having a wide range of message sizes and different security requirements.

REFERENCES