Security, Trust and Privacy supported by Context-Aware Middleware

Mario Hoffmann*, Atta Badii**, Stephen Engberg***, Renjith Nair**, Daniel Thiemert**, Manuel Matthess*
*Fraunhofer-SIT, GER, **IMSS, University of Reading, UK, ***Priway, DK

***Abstract*—Driven by new network and middleware technologies such as mobile broadband, near-field communication, and context awareness the so-called ambient lifestyle will foster innovative use cases in building automation, healthcare and agriculture. In the EU project Hydra[1] high-level security, trust and privacy concerns such as loss of control, profiling and surveillance are considered at the outset. At the end of this project the Hydra middleware development platform will have been designed so as to enable developers to realise secure ambient scenarios especially in the user domains of building automation, healthcare, and agriculture.**

This paper gives a short introduction to the Hydra project, its user domains and its approach to ensure security by design. Based on the results of a focus group analysis of the building automation domain typical threats are evaluated and their risks are assessed. Then, specific security requirements with respect to security, privacy, and trust are derived in order to incorporate them into the Hydra Security Meta Model.

How concepts such as context security, semantic security, and virtualisation support the overall Hydra approach will be introduced and illustrated on the basis of a technical building automation scenario.

***Index Terms*—Context-Awareness, Semantic Resolution, Virtualisation**

I. Introduction

DIGITAL revolution over the past few decades has resulted in increasing usage of embedded systems deployed in technologies supporting our life and work styles in the ICT-empowered environment. From the washing machines we use in our homes to the mobile phones and PDAs on which we depend to communicate and work, they all deploy embedded systems. World Semiconductor Trade statistics show that 98 percent of the programmable digital devices are embedded devices [1]. Whilst the plethora of embedded programmable devices available from various manufacturers is re-assuring of a competitive, diverse and hopefully enduring creative base of Research and Development in such critical components, it also makes for a heterogeneous array of devices distributed in the ambient environment which cannot communicate with each other due to lack of a common protocol to provide for the much needed seamless integration.

Imagine being able to control your home TV remotely, using your mobile phone so that you do not have to search for the TV remote anymore; how is such facility going to be supported? The EC co-funded the FP6 IST project Hydra (Networked Embedded System Middleware for Heterogeneous Physical Devices in a Distributed Architecture) to support some of the leading companies and research institutes in Europe in attempting to fulfil the vision of such seamless integration in the ambient environment of heterogeneous devices. Hydra aims to develop the middleware layer for building secure, fault-tolerant Networked Embedded Systems where diverse heterogeneous devices co-operate to achieve a given goal [2]. The emergent world of ambient intelligence and pervasive computing would be closer to realising its full potential if the embedded devices deployed, for example in a home, are able to automatically communicate with each other and thus cooperate to fulfil a task. The Hydra mission is to provide this capability by providing the required secure interoperable middleware.

[1] "HYDRA – Networked Embedded System Middleware for Heterogeneous Physical Devices in a Distributed Architecture", website: http://www.hydra.eu.com/, contract number: IST-2005-034891, duration: 07/2006-06/2010,

II. Hydra Challenge

Security is often a neglected area in application development as developers tend to ignore its importance and display an evasive rather than a responsive mindset with regard to security e.g. as typified by comments such as 'Let us first build the system, we will make it secure afterwards'. Naturally this lack of an approach to security by design also pervades the embedded systems development. Hydra aims to avoid this by having security as one of the primary objectives in the overall architecture integration. It aims to provide a secure middleware enabling developers to implement secure interoperable embedded applications to serve the ambient intelligent environment by providing service-oriented model-driven architecture. The Hydra project will be primarily focusing on 3 demonstrator domains: Home Automation, Healthcare and Agriculture.

The biggest challenge facing Hydra is in providing secure interoperability for embedded applications. How can a Building Automation System and any PDA communicate with each other if they use totally different security mechanisms and standards? How can one make sure that the communication between such devices is not compromised? In the next sections we will attempt to present a bird's eye view of our research within Hydra to derive the requirements and the approaches which we will use in order to fulfil these requirements. In this way we intend to provide some answers to our common concerns to achieve not just secure interoperability but potentially also *cooperativity* amongst heterogeneous embedded systems serving us in the emergent ambient environment.

III. Security Requirements Engineering

In the Hydra project the following security requirements specification process (cf. Fig. 1) is performed in order to ensure security by design: First, we derive a technical scenario from the building automation user domain scenario. Then, we conduct discussion rounds with focus

groups of expert developers who are potential future Hydra middleware users. In the focus group analysis, actors, assets, and roles are identified. Based on the analysis of multilateral communication schemes between those roles we identify high-level threats to Hydra. Following the concept of "security by design" we derive the overall protection goals that have to be taken into account for the design of the Hydra middleware platform. The results of the focus group analysis in combination with the state-of-the-art are the basis for the risk analysis. Here, the identified threats and potential (threat) actors are analysed and described. Probability, impact and effects of successfully performed attacks are assessed and used as input to calculate the risk of a threat. From that point it is then possible to estimate how serious actors should take a threat. Finally, the process results in derived and prioritised security and trust requirements based on the results of the risk analysis.

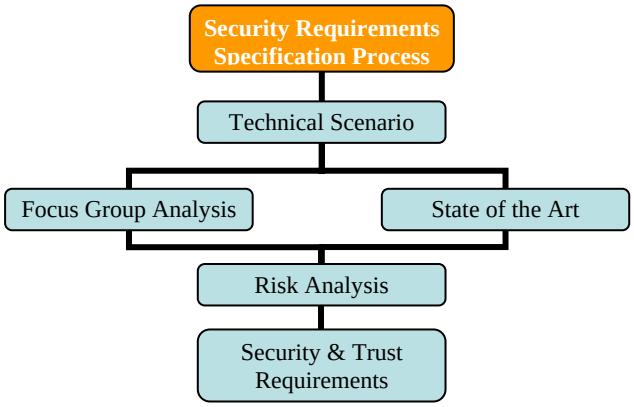

Fig 1. Security Requirements Specification Process

A. *Technical Scenario*

The technical scenario used in our approach is built on the vision scenario for the user domain "Building Automation". Since the vision scenario is not very detailed in terms of technical aspects the technical scenario adds this information.

The aim of the technical scenario is to give the members of the focus group a better and more detailed starting point for their technical interpretations to elicit requirements for the security and trust within the Hydra project.

B. *Results of Focus Group Analysis*

The technical scenario is the starting point of focus group analysis. Here, an initial threat analysis of the technical implications identifies assets to protect such as billing information, user preferences and profiles, as well as communication data, actors such as building operators, service technicians and occupants, and roles such as network operators, content providers and end-user. The analysis of multilateral communication schemes between such roles derives the main protection goals that the developers would expect to be met taking advantage of the future Hydra middleware platform. These comprise: (1) Confidentiality, (2) Integrity, (3) Authenticity, (4) Authorisation, (5) Availability, (6) Non-repudiation, and (7) Privacy.

C. *Results of Risk Analysis*

On the basis of these protection goals the risk analysis defines eight steps as part of a Hydra specific user-centric framework for risks analyses and evaluation. This comprises a pattern-based description of assets, (threat) actors, and threats as well as the assessment of attacks, their impact, their probability, and security implications. The highest risks in our analysis according to the usage scenario are expected to affect user data and identity, where identity comprises both user identities as well as device identities.

D. *Security and Trust Requirements*

The derivation of the security and trust requirements based on the previous results conclude the security analysis. The requirements are prioritised according to their classification into the categories mandatory, desirable and optional requirements. With respect to the risk analysis the most important requirements concern securing confidential information, e.g. private data during transactions, and empowering the user to control both his individual context and the disclosure of personal information to the immediate vicinity as well as to authorised (virtual) parties.

E. *Hydra Synthesis*

Driven by new network and middleware technologies such as mobile broadband, near-field communication, and context awareness the ambient lifestyle will foster innovative use cases in building automation, healthcare and agriculture. However, the more personalised information has to be collected, linked and analysed by ambient systems in order to serve users according to their individual context, the more the specific protection goals have to be balanced between actors in those scenarios underpinned by security and privacy enhancing technologies.

More than 80% of the security and trust requirements have been classified "mandatory" to be fulfilled by the Hydra security model. Most important requirements aim at (1) securing confidential information, (2) authentication mechanisms, (3) context-aware access control, (4) context and semantic reasoning, (5) interoperability of (security) communication protocols, and (6) distributed trust models. In order to fulfil these requirements we propose a security meta-model with the following key characteristics:

- be interoperable with existing security models,
- be extendable,
- allow developers to semantically define security requirements,
- allow developers to virtualise end-users, services, and devices, and
- simplify implementation.

The concepts needed to realise the Hydra Security Meta Model, i.e., (1) context security, (2) semantic security resolution, and (3) virtualisation, will be introduced in detail in the next section.

IV. Towards Security, Privacy and Trust

A. *Security Context – Logical Boundaries of Middleware.*

Security Context defines the logical security boundary of any instantiation between the middleware and device, between middleware and an application and between middleware and users.

The boundary given by the security context is flexible and depends primarily on developer and user choices about how much trust is placed in the Hydra middleware. As a default, developers cannot be assumed to trust Hydra. But by choice, developers and users can then delegate security aspects to Hydra.

In this way, as a facilitator rather than a guarantor of security, Hydra provides for security-aware design and development by enabling developers of embedded systems and to include security and privacy aspects in their applications.

B. Semantic Security – Model- and Rules Driven Security.

Semantic model driven interoperability also requires security to be resolved at the semantic level. This is both to ensure translation between heterogeneous entities, for applications to delegate security decisions to the middleware layer and to ensure dynamic adaptability according to the specific context needs. This is one of the most complex and critical aspects of the middleware as it transposes all levels and all elements. Whilst the Hydra middleware is not intended to enforce a specific security model on devices or applications, it is nonetheless responsible for ensuring interoperability in even sensitive applications such as Healthcare or Asset Protection.

Semantic Security not only has to be flexible and interoperable, it also has to be dynamic so that new rules or requirements can be added or changed in near real-time. A security alert may alter and raise security requirements elsewhere. A new security policy may be implemented affecting functionality of some applications. This may be due to new technical capabilities or new Security models or primitives.

C. Context – Inter-Context Isolation and Intra-Context Resolution.

The main security element is the understanding of logical context as the replacement of (or addition to) a physical security understanding. In any physical space many contexts have to be able to overlap without interfering. The special context defining a geographically extended space is a domain context which a sovereign owner controls and sets rules for. To manage risk, a stakeholder perspective is critical. Different stakeholders and devices do not see the same context the same way, i.e. one or two stakeholders may know more about a context than others for instance through the concept of virtualisation.

In Hydra we consider three layers of context: security-defined, rules-defined and dynamic. What a stakeholder *can*, what a stakeholder is *authorised* to, and, what is *assumed* -such as learned through interaction and logical resolution.

D. Virtualisation – Tools to Design with Context.

As perimeter security fails due to integration and increasing interconnections, virtualisation or separation between the physical and logical representation of an entity becomes the primary security mechanism – the main security paradigm is shifting from identification to virtualisation. Virtualisation can occur at many levels, with many different tools, as logical reduction or combination and often even nested as messages travel across contexts. A very simple example is the use of session handle identifiers with end-to-end encrypted communication to shield stakeholders against any leakage of context information in transport. The same can happen on a proxy-level as VPNs (Virtual Private Networks) when people from work access specific services or devices at home. Server virtualisation concepts are known, but they have to be adapted to the device and end-user level.

E. Towards a Security Meta model

For a model driven middleware, the above concepts have to be integrated by way of a Security Meta-model, i.e. a model of security models making otherwise incompatible security models interoperable. Today only very simple models exist such as FIPS, but we do not have tools to compare and translate between security models. We cannot represent partial resolutions having been made at one point as input for security resolution at a later point. We cannot compare two identity models as all main aspects are implicit to the identity model with jurisdiction and end-to-end traceability as obvious needs.

With virtualisation as the emerging paradigm, the need for a security meta-model becomes obvious and critical for interoperability and security in a world of heterogeneous devices and communication.

V. A USAGE SCENARIO

In order to illustrate the necessity and benefits of the Hydra Security Meta Model, we currently implement a demonstrator scenario (cf. Fig. 2). The demonstrator scenario is based on the technical building automation scenario used as the starting point of the security analysis in III.A. In this scenario, a service technician sent by a service provider needs physical access to a faulty heating system of a resident who is currently not at home.

The steps 1 to 4 in Fig. 2 focus the security challenges and how these will be resolved through the realisation of specific parts the Hydra Security Meta Model:

The scenario starts with a critical malfunction in the heating system that has been detected by a device specific Hydra Proxy in step 1. Note that in current home and office automation systems Hydra Proxies serve as virtual representations of legacy devices in the Hydra network. On the one hand they take into account device specifics and on the other hand they take advantage of the Hydra network's security mechanisms. Future systems are envisioned to be Hydra enabled, i.e. they talk Hydra protocol by default.

Once the heating system's Hydra Proxy has interpreted the malfunction as critical the proxy sends an error message to the Hydra-based Building Automation System (HBAS). This intelligent network node can take further context information into account and finally sends as a result of the context resolution process a service request to the resident who is currently not at home. The HBAS request includes the error protocol and recommends calling a service provider to fix the problem.

In step 2 the resident receives the authentic request from his HBAS and decides to follow the recommendation. He digitally signs the error protocol and sends it – including a context restricted authorisation token – to a service provider of his choice. The authorisation token will be used in step 3.
Step 3 describes the situation in front of the resident's house. A service agent or technician presents the authorisation token carried – to simplify matters – on a

Hydra enabled

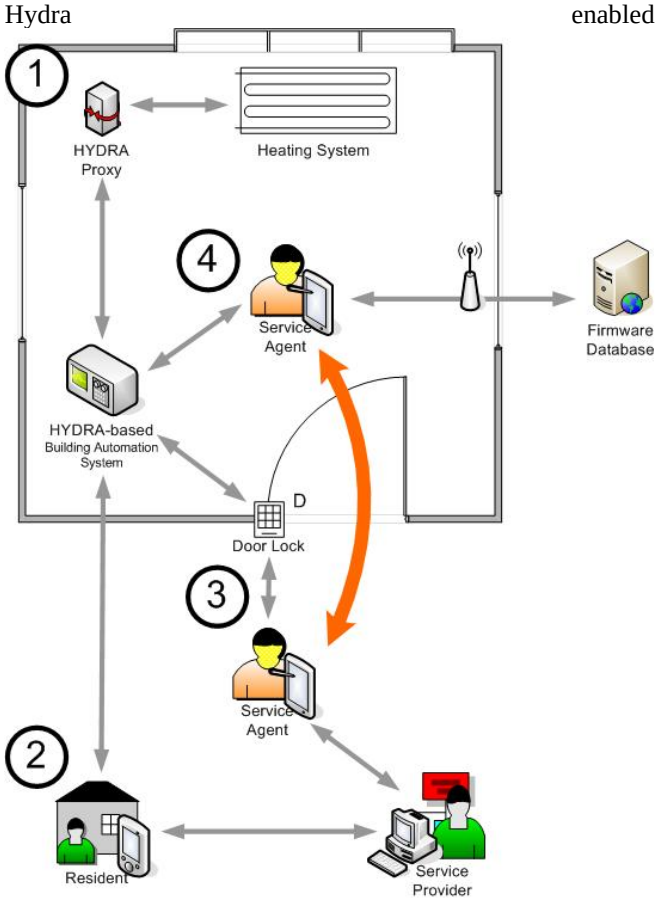

Fig. 2: Demonstrator Scenario

PDA or Smart Phone to the door. The door forwards the token which has only been additionally signed by the service provider to the HBAS that proves it valid and trusted. So the service technician is now allowed to enter the house and gets access to the HBAS in the final step. Note that the HBAS does not ask for the technician's identity. In order to guarantee liability the double-signed authorisation token (by the resident and the service provider) is sufficient.

In the final step – step 4 – the technician gets restricted access to the Internet in order to download the latest version of specific diagnostic software and the heating system's firmware update. After fixing some configuration settings and installing the update of the firmware the heating system works smoothly inside of its specification again.

In addition to the secure authorisation process based on trusted credentials and virtualisation introduced above the demonstrator will be improved by two steps during the next months. Firstly, semantic security resolution will add trusted authentication in the Hydra network even to non Hydra devices. Secondly, the rather simple role-based access control (RBAC) above will be enhanced to context-based access control (CBAC) to support more dynamic and unforeseen scenarios. The final demonstrator will be shown at CeBit fair 2008.

VI. Summary and Outlook

In this paper we have presented the approach to security, privacy and trust supported by a context-aware middleware. We have presented our process of gathering the requirements for a middleware for heterogeneous networked embedded systems in the Hydra project. Furthermore we have introduced our approach to meet those requirements which are based on context, semantics, and, a security meta-model.

Further research in the project will be focussed on how the context can be represented in order to support the proposed security models. Further, we plan to investigate how different security models can be represented semantically based on ontologies in order to realise interoperability. The final outcome will then be the Security Meta-Model, in addition to a software development kit and an integrated development environment, which will enable developers to involve security aspects from the initial stages of embedded application development.

References

[1] Study of worldwide trends and R&D Programs in Embedded Systems in view of maximizing the impact of a technology platform in the area, FAST GmbH for the European Commission, ftp://ftp.cordis.europa.eu/pub/ist/docs/embedded/final-study-181105_en.pdf.

[2] The Hydra Project, http://www.hydra.eu.com.