TECHNICAL SCIENCES
Abbrev.: Techn. Sc., No 15(1), Y 2012

SECURING AGENTS USING SECURE DOCKING MODULE

***Emil Gatial*[1], *Zoltán Balogh*[1], *Daniel M. Hein*[2], *Ladislav Hluchý*[1] *Martin Pirker*[2], *Roland Toegl*[2]**

[1] Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
[2] Institute for Applied Information Processing and Communications, Graz University of Technology, Graz, Austria

K e y w o r d s: mobile agents, trusted computing, crisis management support.

A b s t r a c t

Modern communication and computing devices have the potential to increase the efficiency of disaster response. Mobile agents and seamless push-to-talk communication embody decentralised and flexible technologies to leverage this potential. While mobile agent platforms are facing greater variety of security risks compared to a classical client-server approach, trusted computing is capable of alleviating these problems. This document describes design and integration of a Secure Agent Infrastructure (SAI) with a Secure Docking Module (SDM) based on trusted computing principles for crisis management support. SDM provides a single chip security device that replaces the centralized trust decision and point with a suitable distributed solution. The main goal of SDM is protecting information. The protected information is only released to a requesting host device if the host is in a trusted state and adheres to a specific set of policies. SAI relies on the crypto-material protected by SDM thus the mobile agent can be unsealed only if the host machine is in the trusted state. The paper introduces the SDM and SAI technologies, describes motivation of SDM usage, provides summary of the key concepts behind the SDM and SAI. Further we provide analysis of requirements and security considerations as well as the integration points of the proposed architecture with other involved systems and the communication adapters between agents and other legacy systems. The last section concludes the article and presents our current achievements in integration and demonstration of the proposed technologies.

WYKORZYSTANIE SECURE DOCKING MODULE DO ZABEZPIECZANIA SYSTEMU AGENTOWEGO

***Emil Gatial*[1], *Zoltán Balogh*[1], *Daniel M. Hein*[2], *Ladislav Hluchý*[1] *Martin Pirker*[2], *Roland Toegl*[2]**

[1] Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
[2] Institute for Applied Information Processing and Communications, Graz University of Technology, Graz, Austria

S ł o w a k l u c z o w e: mobilny system agentowy, trusted computing, zarządzanie kryzysowe.

Abstrakt

Współczesne rozwiązania teleinformatyczne mogą istotnie zwiększyć efektywność działań w sytuacjach kryzysowych. Systemy mobilnych agentów oraz „bezszwowa" komunikacja push-to-talk stanowią zdecentralizowane oraz elastyczne technologie wnoszące nową jakość do tej domeny. Rozwiązania oparte na mobilnych systemach agentowych są bardziej narażone na różnorodne zagrożenia w porównaniu z klasycznym rozwiązaniem klient–serwer; podatności te jednak mogą być redukowane dzięki zastosowaniu rozwiązań typu Trusted Computing. W artykule przedstawiono budowę oraz integrację Secure Agent Infrastructure (SAI) z Secure Docking Module (SDM) na podstawie zasad Trusted Computing. Rozwiązanie prezentowane w artykule jest przeznaczone do wsparcia zarządzania w sytuacjach kryzysowych. Głównym celem SDM jest ochrona informacji. Chroniona informacja jest udostępniania innym hostom tylko i wyłącznie, gdy znajdują się w stanie zaufanym oraz są w zgodności z określonym zestawem polityk.

W artykule opisano technologie SDM oraz SAI oraz uzasadniono stosowanie SDM. Przedstawiono także najważniejsze zagadnienia związane z SDM oraz SAI. Ponadto przeanalizowano wymagania oraz zagadnienia związane z bezpieczeństwem; wskazano także możliwości integracji zaproponowanej architektury z innymi systemami oraz urządzeniami komunikacyjnymi między agentami a tradycyjnymi systemami. W ostatniej części artykułu podsumowano jego treść oraz przedstawiono obecne osiągnięcia w dziedzinie integracji oraz demonstracji zaproponowanych technologii.

Introduction

Modern communication and computing devices have the potential to increase the efficiency of disaster response. Mobile agents and seamless push-to-talk communication embody decentralised and flexible technologies to leverage this potential. While mobile agent platforms are facing greater variety of security risks compared to a classical client-server approach, trusted computing (TC) is capable of alleviating these problems. Unfortunately, remote attestation, a core concept of TC, requires a powerful networked entity to perform trust decisions. The existence and availability of such a service in a disaster response scenario cannot be relied upon. One of the challenges of the communication infrastructures for distributed systems is to add new smart functions to existing services which would make the communication more effective and helpful for users. The aim is to provide smart functions via distributed IT systems which should provide a secure distributed paradigm to achieve confidentiality and access to resources. Such infrastructure should further provide a smart negotiating system for parameterization and independent handling of access requests to achieve rapid reaction. A good application of proposed system provides crisis management support that requires existing information from legacy systems of various organizations and from human operators in order to semi-automatically manage the crisis mitigation process or to enact decisions at various management levels. This information collection must be enacted in a secure manner while ensuring trust between both parties – information consumers and information providers. Many actors participate

in a crisis situation, the competences and responsibilities of all parties are explicitly defined in a crisis mitigation plan. Information gathering is enacted either from legacy systems or from human end-users through mobile devices by guided dialog.

Several crisis response systems have been successfully built using multi-agent paradigm and other systems are being developed. Systems like DrillSim (BALASUBRAMANIAN 2006), DEFACTO (MARECKI 2005) and Mobile-FIRST (HONDA 2009) were developed to simulate disaster situation using software agents enabling human actors to act more effectively. More realistic deployment of agent system was developed in the ALADDIN project (JENNINGS 2010) demonstrating the usefulness of decentralised and autonomous agent behaviour in the disaster management domain. VOYAGER (2011) communication platform delivers highly collaborative, dynamic, cross-platform applications and infrastructure for all business situations without the need of overwhelming modification of underlying corporate information systems. Specific use of mobile agents was presented in VEMPR system (MARTIN-CAMPILLO 2009) dealing with reliable access to medical records of victims and in PA-UWNT research project (KOPENA 2005) managing communication in mobile ad-hoc network project and Web-service based applications.

In this article we focus mainly on the concepts of security and trust used in Secure Agent Infrastructure (SAI) developed in the scope of SECRICOM integrated EU project (SECRICOM 2012). The goal of presented SAI is to enable easy collaboration and information sharing among actors in crisis situation, with an emphasis on security and trust of the information. In the following chapter, we present the architecture of SAI communication platform that deals with secure and trusted data collection during the crisis mitigation. We describe concepts of Secure Docking Module (SDM) and Trusted Computing approaches establishing trusted computing environment for SAI. Final part is devoted to description integration of SAI and SDM and to description of testing infrastructure. We conclude with achievements of SAI and SDM integration.

Architecture Design

We present a distributed architecture designed for the management of crisis situations where multiple actors are involved from various organizations with different competences and communicating over IP-based networks including wireless. We decided to design and implement such an architecture using agent paradigm. The distributed agent-based infrastructure is designed as a collection of software services with agent-like features (such as code

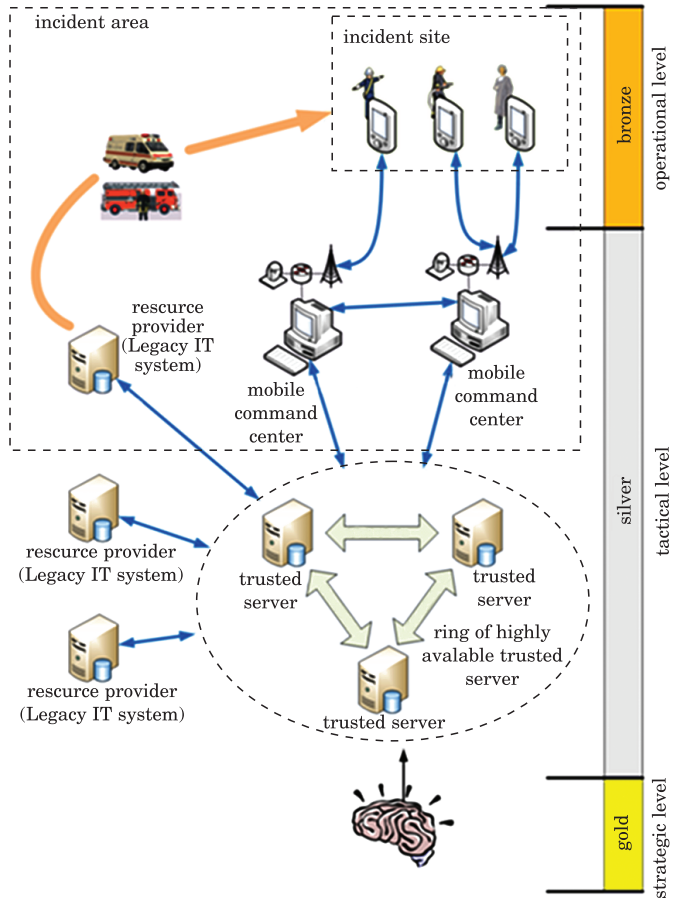

Fig. 1. An overview of Secure Agent Infrastructure applied in crisis management scenario

mobility) which would execute in a secure and trusted manner. Agent technology was selected due to the ability to fulfill such requirements through support of mobile and dynamically deployable executable code. Other advantages of agent-based systems are that they can help overcoming temporal or longer term communication network failures, save network bandwidth by being executed remotely and deliver only the execution results, provide means to execute code on remote host platforms in a trusted and secure manner or deploy code on host platforms on demand. The role of agents in the architecture is primarily coordinated collection of information. Information gathering is enacted either from legacy systems or from human end-users through mobile devices by guided dialog. With respect to requirements the overall agent

infrastructure must be a secure, robust and failure resistant system. Because validity and authenticity of gathered information is a key factor for decision making in crisis management, trust must be set between agents and third party information systems. Also, agents must trust the host platform providers. The required level of trust for agents is based on a special hardware module – SDM providing TC functionality.

The home platform for agents is a network of Trusted Servers (TS) as it is depicted in the above figure (Fig 1). There are many different users involved in crisis management. Each type of user has a different level of responsibility, performs different tasks and requires different information (CRADDOC 2008). Gold Commanders who are in charge of producing strategy require information about the incident and about its effects on the wider area. They rarely need to make instant decisions, so have some time available to absorb information. Silver and Bronze Commanders are usually located closer to an incident site and need more detailed information about the incident and the resources available to them, as they have to turn the Gold-level strategy into a response, but are not as concerned with events outside the incident. They may have to make quick decisions as events unfold. Response Team Commanders and responders who are implementing a response have limited time in which to take in information and, as such, only need information relative to their immediate task. The coordination of responders; actions as well as providing live information to commanders in Silver and Gold level are the most important challenges in crisis management.

Concept of Docking Station Functionality

The SDM should allow agents to dock on a secure communication infrastructure by ensuring the state of the device it is supporting. The SAI is a distributed system and operates on confidential data. Therefore, the system must protect its integrity against data loss/theft and data modification. In a distributed system, data protection concerns are not limited to data transmission. As the data are processed in different physical computing platforms it must be established that all data processing entities adhere to the same security policy for the data. The data security policy adherence is enforced by ensuring the software configuration of a computing platform before it is connected to the SECRICOM infrastructure. To this end the SDM protects communication keys and credential information and only releases this information to the host platform if this platform is in an approved software configuration. The process of establishing the fact that a platform has an approved software configuration is called local attestation verification. Concep-

tually, the SDM protects a small set of key pairs for asymmetric cryptography, but in general is capable of protecting arbitrary data up to a specific size. The SDM's key protection facilities are a standard function, which could already be implemented with today's smart cards or hardware security modules. The SDM extends this standard function by only releasing these keys to a host device if and only if this host device is in a trusted state. This host device is called Trusted Docking Stations (TDS). The relationship between SDM and TDS is depicted in the figure below (Fig. 2).

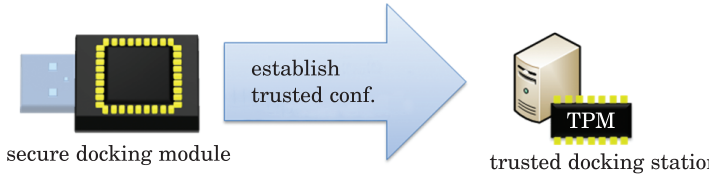

Fig. 2. Relationship between Secure Docking Module and Trusted Docking Station

A trusted platform software configuration is a specific software configuration. This software configuration is measured by a Trusted Platform Module (TPM). The combination of a SDM with a TDS is called a Secure Docking Station (SDS).

Trusted Computing

Generally, TC approaches were summarized in the work (PEARSON 2002). Trusted computing as specified by the Trusted Computing Group (TCG 2007) enables the authentication of a computing platform's software configuration. The software configuration is measured and mapped to a single value. The authenticity of this value is corroborated by signing it with a unique private key. This process is called attestation. Attestation allows a verifying entity to establish the software identity of a platform and correlate it with a configuration that enforces a set of required policies. If a platform's software configuration adheres to this set, we refer to this software configuration as trusted software configuration. For the attestation process to be valid, the software configuration measurements must be protected against tampering, the private signing key must be protected against misuse and compromise. Also, the private signing key must be bound to the measured platform. For these reasons, the core component of TC is a trusted module which fulfills these requirements. The components of the architecture can be broken down into

different blocks, namely Secure Boot, Base System, Trust Management and Virtualization Partitions. The following figure (Fig. 3) illustrates these blocks.

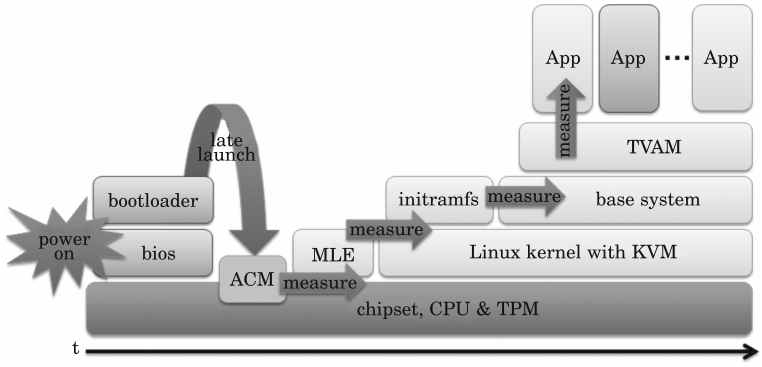

Fig. 3. Overview of the main components of the platform: Secure Boot, Base System, Trust Management and Virtualization Partitions. Trusted components are in green, untrusted are in red. The timeline indicates the different phases of platform boot

The Secure Boot block is responsible for initializing the system to a predefined configuration that requires close cooperation of hardware and software modules. We use Intel TXT as physical platform (Intel TXT 2011). The software side is accomplished by using a standard boot loader (GRUB) along with SINIT and tboot. Upon power-on, the platform performs a conventional boot, but does not start an operating system; instead, the MLE is prepared and a TXT late launch is performed. The precise, desired software configuration is specified by the administrator in the form of policies stored in the TPM. The LCP is evaluated by SINIT and specifies which MLE is allowed to be executed. tboot's policy is called Verified Launch Policy (VLP), and it contains known-good values for measurements of the Linux kernel and its temporary ram disk initramfs. A secure boot is performed into a hardware guaranteed state and the chain of trust is extended over the kernel and initramfs. If the measurements do not match the expected values provided by the VLP, tboot will shut the platform down. The startup code in the initramfs ensures an unbroken chain-of-trust; it measures the file system image of the full Base System into a PCR before it is mounted.

The Base System is a customized Linux operating system. The kernel is augmented with the Kernel-based Virtual Machine (KVM) hypervisor module. KVM requires common Commodity PC platform equipped with virtualization extensions. KVM can run multiple virtual machines, where each virtual machine has private virtualized hardware like a network card, hard disk, graphics adapter, etc. Those virtual devices are forwarded to QEMU (QEMU

2012), fast software which emulates a full hardware platform. To support deterministic PCR measurement, the Base System's file system must remain read-only. A temporary file system provides the needed read-write storage during platform operation. However, changes to the Base System do not survive a reboot of the platform. This ensures robustness of the base system image to malicious modifications. Management of the virtual partitions itself is done by a component called TVAM, the Trusted Virtual Application Manager. Virtualization Partitions may host any system normally running stand-alone. This can be an unmodified out-of-the-box Linux or Windows system, or a heavily customized system.

Securing Agents in Trusted Environment

The SAI actually provides the software components (HECTOR 2005) needed to run agents. Moreover, TDS uses SDM to setup a TC environment and thus enforces the policies required by the legacy systems. SDM releases the protected cryptographic material if and only if the TDS was booted into the trusted state; that means the platform is in the well known state. The DSAP service employs the SDM for storing the TDS private key, which is used to decrypt incoming agent's symmetric key to be run in a trusted environment.

The root of trust is established between the agents' home platform and host platform (HP) by audited agent code before its usage will take place. The audit process must ensure that the agent does only what its creator states it should do, and that it does not contain any malicious code, which may jeopardize the integrity of the HP. Establishing the trust between an agent and a HP is depicted in the next figure (Fig. 4).

Agent repository (AR) holds the set of certified agent Java classes or jar files. The code of agents may vary from executing simple DB query to complex management of HP resources. It is up to the agent designer to implement an agent's functionality, but with respect to the fact that the code must be audited and certified whether by the HP provider or by a trusted third-party authority. Based on the code certification the HP provider can trust the code running his or her HP. When Process Management Subsystem (PMS), which is specialized system coordinating data collection, decides to issue an agent it queries AR to obtain the classes implementing the agent. Here, PMS is able to verify the certificate of agent classes. Next, an instance of agent object is created by PMS where the agent attributes are set. The agent object and its classes are encrypted using an AES key secured by $TDS_1PubK_{E/D}$ public key (referred to as key encapsulation) (PSEC-KEM 2008) of HP. After the encrypted agent is moved on the HP, the DSAP service decrypts the AES key using $TDS_1PrK_{E/D}$

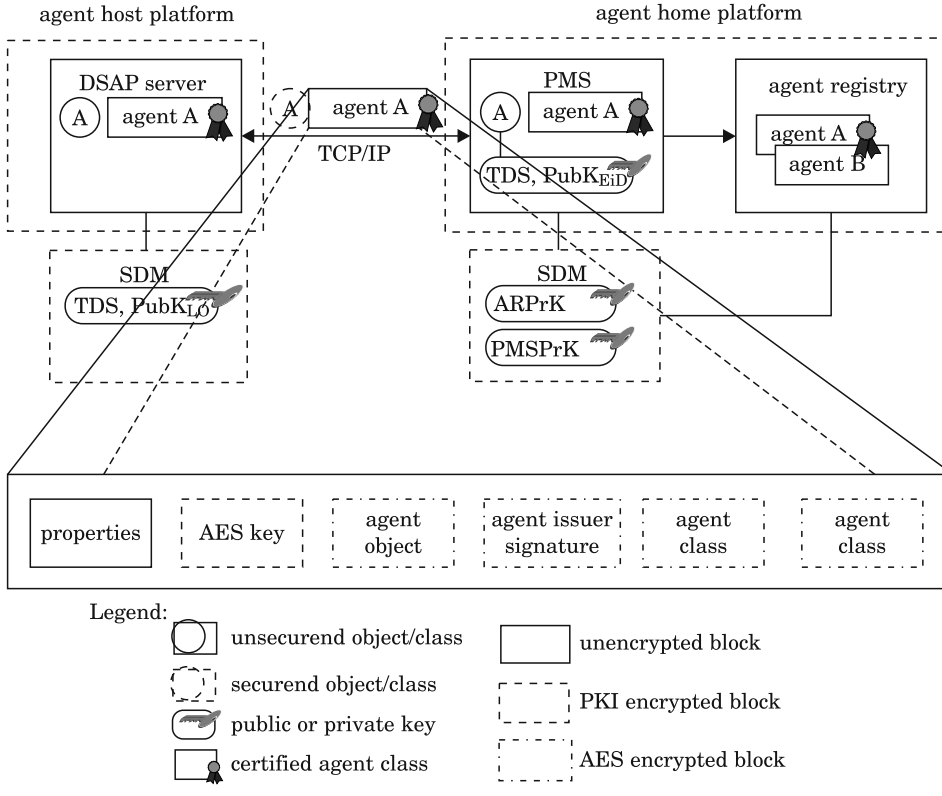

Fig. 4. The scheme of DSAP concept to establish secure and trusted communication of agents

private key of the HP (received from SDM) and uses this key to decrypt an agent. The HP usually provides access to some resources that a specific agent is able to process. Here, PMS is responsible for choosing the right type of agent and for setting him up to provide the required results. The results are encrypted using the same AES key and sent back to PMS.

Testing Infrastructure

The coordination of agents in SAI platform was tested in the scenario of free hospital beds reservation, while rescuing injured people. The infrastructure, (Fig. 5) comprises four fictive hospital information systems, where each system is attached to DSAP platform secured by SDM module (Linux OS). Next, the dedicated host platform running PTT client (Windows OS) is included in order to support end users to communicate with SAI via PTT

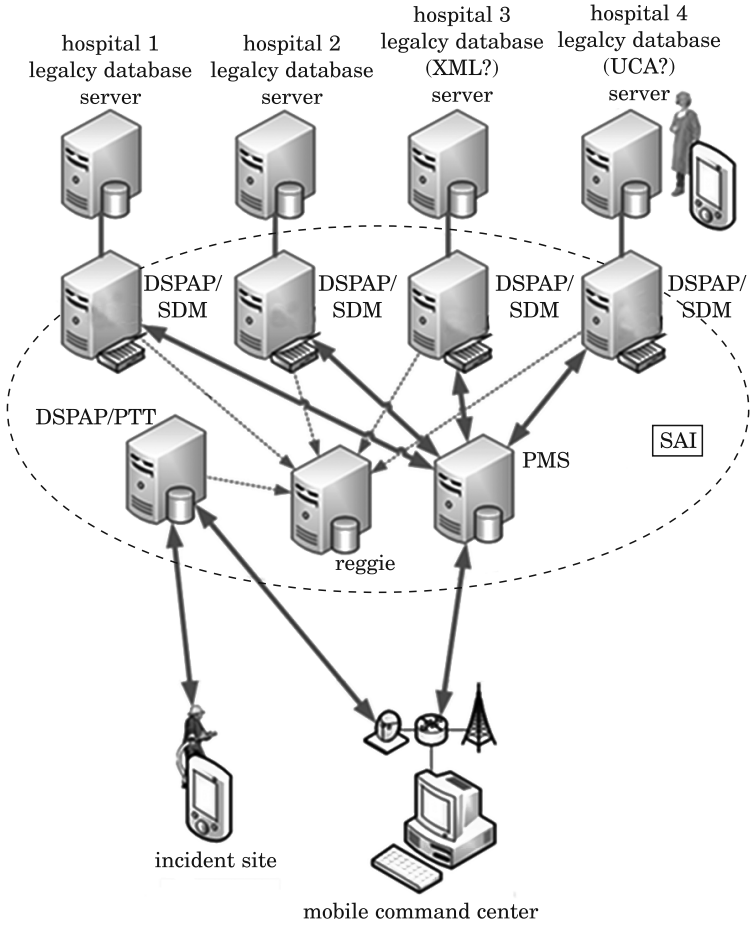

Fig. 5. SAI testbed infrastructure

enabled devices. Reggie component contains registry of DSAP services available in the network. Finally the specialized component called PMS coordinates agent communication and deployment.

The process of SAI enabled crisis management support starts when a first responder needs to find the closest hospital with a particular type of injury treatment. He can directly specify the type of process in the PTT enabled device or call the command centre. User communication agent (delegated by PMS) then collects information using guided dialog requesting the injury type, number of injured people and the position of incident (by location name or by GPS coordinates). By submitting the request the PMS sends information delivery agents to every nearby hospital system to query specific data and send

them back to PMS. PMS then automatically reserves specific number of free hospital beds which are closest to incident location.

Conclusion

In this paper we described the integration of secure agents with a secure communication infrastructure for rapid information gathering in a crisis situation. Requirements for using secure agents arose from communication challenges in crisis management problem domain. The concept of SAI shows big potential in the applications using data from different legacy information sources or even different end-users using different communication channels. Moreover, the applications can benefit from the agent mobility and TC by processing data at hosting storage element or in its vicinity. The benefits of SDM as opposed to attestation based on sealing are twofold. First sealing is rather inflexible and does not allow easy migration. The SDM on the other hand can be plugged into any device with the appropriate interface. Furthermore, it is simpler to maintain a set of valid platform software configurations on the SDM, because it represents a single point of management. The second reason is that the SDM is a physical token. Its possession alone contributes to the authentication of the owner and it cannot be plugged into two devices at once. This restricts access to one device at the time.

Acknowledgment

This work is supported by project SeCriCom FP7-218123. This publication is the result of the project implementation: "Industry research in the area of effective work with large data in user oriented applications, ITMS code: 26240220029" and "RPKOM, ITMS code: 26240220064" supported by Operational Programme Research & Development funded by the ERDF.

Translated by AUTHORS

Accepted for print 30.06.2012

References

CRADDOCK R. 2008. *The UK Civilian Command and Control Hierarchy for Crisis Management, Responsibilities and Information Flow.* Thales Research and Technology (UK) Limited.

BALASUBRAMANIAN V., MASSAGUER D., MEHROTRA S., VENKATASUBRAMANIAN N. 2006. *DrillSim: A Simulation Framework for Emergency Response Drills.* Proceeding ISI'06 Proceedings of the 4th IEEE international conference on Intelligence and Security Informatics, pp. 237–248.

HECTOR A., NARASIMHAN V.L. 2005. *A New Classification Scheme for Software Agents.* Proceedings of

the Third International Conference on Information Technology and Applications (ICITA'05), IEEE Computer Society, ISBN:0-7695-2316-1, pp. 191–196.

HONDA J.M. 2009. *Application of Mobile Agent Systems to First Responder Training*. MSc. Thesis, University of California, http://www.cs.ucdavis.edu/research/tech-reports/2009/CSE-2009-13.pdf.

INTEL TXT. 2011. *Intel® Trusted Execution Technology (Intel® TXT)*. Software Development Guide, March, http://download.intel.com/technology/security/downloads/315168.pdf.

JENNINGS N.R. 2010. *ALADDIN End of Year Report*. Southampton, UK: University of Southampton, http://www.aladdinproject.org/wp-content/uploads/2011/02/finalreport.pdf.

KOPENA J., SULTANIK E., NAIK G., HOWLEY I., PEYSAKHOV M., CICIRELLO V.A., KAM M., REGLI W. 2005. *Service-Based Computing on Manets: Enabling Dynamic Interoperability of First Responders*. Journal IEEE Intelligent Systems Archive, 20(5).

MARECKI J., SCHURR N., TAMBE M. 2005. *Agent-based simulations for disaster rescue using the DEFACTO coordination system*. Wiley, pp. 2–19.

MARTIN-CAMPILLO A., MARTI R., ROBERTS S., GARCIA C.M. 2009. *Mobile Agents for Critical Medical Information Retrieving from the Emergency Scene*. In 7th Internacional Conference on Practical Applications of Agents and Multi-Agent Systems.

PEARSON S. 2002. *Trusted Computing Platforms: TCPA Technology in Context*. Published by Prentice Hall, ISBN-10: 0-13-009220-7.

PSEC-KEM. 2008. *PSEC-KEM Specification version 2.2*. NTT Information Sharing Platform Laboratories, NTT Corporation, April 14.

QEMU. 2012. *Quick EMUlator*. http://en.wikibooks.org/wiki/QEMU.

SECRICOM. 2012. *SECRICOM FP7 integrated project*. http://www.secricom.eu/.

TCG. 2007. *TCG Specification Architecture Overview*. Specification Revision 1.4 2nd August.

TPM. 2007. *Trusted Platform Module*. TCG TPM specification; Version 1.2; Revision 103, https://www.trustedcomputinggroup.org/specs/TPM/.

Voyager. 2011. *200Voyager Pervasive Platform*. http://recursionsw.com/Products/voyager.html#.