Enhancing Security in Cloud-based Cyber-physical Systems

Columbia International Publishing
Journal of Cloud Computing Research
(2016) Vol. 2 No. 1 pp. 18-33
doi:10.7726/jccr.2016.1002
Research Article

Enhancing Security in Cloud-based Cyber-physical Systems

Juha Puttonen[1], Samuel Olaiya Afolaranmi[1], Luis Gonzalez Moctezuma[1], Andrei Lobov[1*], and Jose L. Martinez Lastra[1*]

Received: 27 April, 2016; Accepted: 04 August 2016; Published online: 1 October 2016

Abstract

Cyber-physical systems combine traditional data-processing systems with physical actuation capabilities. The physical aspect introduces new physical risks beyond the typical cyber risks related to information security. Cyber physical systems have an additional physical interface that must be protected from attacks. Furthermore, the physical actuation capabilities increase the potential amount of damage inflicted by a compromised system. The security risks are amplified when cloud computing is applied to provide the necessary data processing capabilities, since such task offloading increases the volume of network communication and the use of external computation resources. Network communication may be intercepted, inhibited, or manipulated. The use of external resources requires that the resource providers are trusted and maintain adequate information security levels. The security levels must be particularly strict, when critical planning and decision-making processes are offloaded on cloud resources. In such cases, poor selection of cloud service providers could compromise the operation of the entire system. This article first reviews recent research on security in cyber-physical systems. Finally, this article investigates methods to improve security in cloud-based cyber-physical systems by analyzing two application examples, a production system based on Service Oriented Architectures (SOA) and a smart-mobility system. Based on the analysis, this article investigates additional security measures to improve security in such systems.

Keywords: Cloud computing; Cyber-physical systems; Security

1. Introduction

Information technology is increasingly applied to automate tasks as well as reduce the amount of energy and materials required for completing them. Consequently, computer systems are merging with physical equipment. The resulting cyber-physical systems (CPS) combine physical actuation capabilities with the data-processing power necessary to carry out complex tasks. Typical examples

*Corresponding e-mail: andrei.lobov@tut.fi
1 FAST Lab., Tampere University of Technology

of CPS range from widely accessible mobile-phone services, such as emergency call systems, to highly restricted devices, such as production equipment in factory automation systems.

Cyber-physical systems generally consist of three layers: the physical layer, network layer, and application layer (Gao et al., 2013). The physical layer includes equipment, such as sensors and actuators, which carry out control commands from the application layer. While the network layer mainly conveys data between the two other layers, it also processes and manages the data (Gao et al., 2013). In addition to controlling the system, the application layer connects the system to external applications.

Similarly, to most information processing systems, the cyber components of cyber-physical systems are increasingly deployed on cloud resources. Among these cyber-physical systems are demand-response (DR) systems, which consist of a cyber communication component and physical control components (Mohan & Mashima, 2014). DR programs automatically schedule system activities to increase reliability and minimize costs (Albadi & El-Sadaany, 2007). Because DR systems frequently manage sporadic events (Mohan & Mashima, 2014), considerable fluctuation inherently occurs in the amount of computing resources required. While cloud computing naturally accommodates such fluctuation, it causes severe information security risks in DR systems (Mohan & Mashima, 2014). Indeed, cyber-physical systems are frequently encountered in critical applications and typically actuate physical processes. Therefore, attacks against them can cause severe damage to both the systems and their environment (Fletcher & Liu, 2011).

The increasing demand for product customization is expediting the adoption of cyber-physical systems in factory automation (Shellshear et al., 2015). Since the inception of programmable logic controllers (PLCs), modern factory automation devices have been cyber-physical: they physically alter product templates and are controlled by miniature, embedded computers. Because the computers are typically expensive and vendor-specific, Garcia et al. (2014) propose the adoption of industrial standards and tools to facilitate the use of inexpensive, open devices. However, cyber-physical systems may be more vulnerable to cyber-attacks if the devices are based on global, open standards and include limited computing power. Such systems are familiar also to malicious users and are likely to lack elaborate security procedures.

The cloud infrastructure providing additional data processing capabilities could become a "gateway" to attack CPS, and vice versa, via CPS one could plot an attack on connected cloud infrastructures. Therefore, well-established security mechanisms are required for merging interactions between CPS and cloud infrastructures. These mechanisms require identification of key elements at CPS and cloud levels and their interaction principles to ensure security for corresponding distributed applications.

This article is structured as follows. Section 2 surveys recent research on security in cyber-physical systems. Then Section 3 analyzes security risks in two different CPS types. Based on the risks identified, Section 4 discusses potential remedies. Finally, Section 5 concludes and identifies topics for further research.

2. Related Research

Since cyber-physical systems include a physical interface, they are subject to additional security

risks absent in purely virtual systems. These risks are particularly apparent in side channel attacks, in which an attacker is able to deduce a cryptographic key by analyzing physically-measurable quantities exposed by the system. Examples of the possible side channels include timing, power consumption, and electromagnetic emissions (Cobb et al., 2013). Agosta et al. (2014) have presented a method of measuring the vulnerability of a system to passive side-channel attacks, in which the attacker knows the input or output data to the encryption algorithm. The physical properties of cyber-physical systems generally allow the deduction of information that should otherwise be restricted (Howser & McMillin, 2013).

When developing countermeasures for security risks, comprehensive risk assessment is necessary in order to optimize the use of limited resources (Xie et al., 2013). Xie et al. argue that the contemporary risk assessment methodologies fail to consider the physical and cyber part of CPS as a single entity (Xie et al., 2013). They therefore present a risk assessment method based on attack trees. An attack tree consists of sub-goal nodes and various attack paths. The risk value of each path is calculated as a function of the damage incurred as well as the threat and vulnerability values associated to each sub-goal node on the tree path (Xie et al., 2013).

Cyber-physical clouds (CPC) allow physical devices, such as sensors and actuators to be virtualized as cloud resources, which can then be acquired on request by several tenants (Hiray & Ingle, 2013). Hiray and Ingle propose a context-aware middleware that restricts access to the resources based on user identity, thereby reducing the risk of, for example, counterfeit sensor values (Hiray & Ingle, 2013). The approach considers contextual parameters, such as location and time, in determining whether a user should be granted access to a resource (Hiray & Ingle, 2013).

Mobile phone networks resemble computing clouds in that both include several computers connected over the network. Indeed, contemporary mobile phones are effectively miniature computers, many of which host an operating system (OS) specialized from a conventional desktop computer OS, such as the Linux-based Android OS. However, mobile phones are additionally connected to conventional telephone networks. Hence, compromised mobile phones can be used in denial-of-service (DoS) attacks against vital public services (Liu et al., 2009).

Smartphones contain extremely powerful processors and sensors that make them ideal mobile cyber-physical systems. However, these advantages also open doors to serious sensor-based privacy theft attacks through sensor abusing. Lei et al. (2013) present a sensor-based voice privacy theft attack called Cyber-Physical Voice privacy Theft Trojan horse (CPVT). The CPVT can be remotely activated by an attacker at arbitrary instants to capture audio through the infected phone. It exploits the vulnerabilities of the Android platform permission mechanism by either secretly elevating its privileges on the mobile phone or distributing its functions into several applications in order to conceal the complete permission pattern. Lei et al. have evaluated the performance of CPVT on five mobile platforms. In addition, they have tested the concealment of CPVT against five antivirus software products and discovered that none of the products was able to distinguish CPVT from normal software. In light of this, Lei et al. propose a behavior based sensor access control framework as a countermeasure to CPVT. The framework dynamically compares Security Behavior Pattern (SBP) of sensor access to application sensor access behavior, which enables the framework to filter unsafe sensor access.

Chonka & Abawajy (2012) present a defense system against HX-DoS (HTTP and XML Denial of Service) attacks targeted at cloud cyber-physical systems. HX-DoS attacks involve transmission of XML and HTTP messages, thereby operating on the application level (Chonka & Abawajy, 2012). HX-DoS attacks can be launched against any cloud-based cyber-physical systems (Chonka & Abawajy, 2012). The defense system proposed by Chonka and Abawajy can be trained to identify HX-DoS messages before they are received by the targeted system.

DoS attacks are particularly damaging for cloud applications offered as software-as-a-service, because the application owners pay for computational resources consumed. By increasing the resource usage, such attacks both reduce the availability of the service and increase the expenses incurred (Ficco & Rak, 2012). While a cloud application may dynamically acquire additional resources to maintain the desired quality of service even under a DoS attack, such a reaction reduces profitability (Ficco & Rak, 2012).

An intrusion-tolerant cloud application can endure DoS service attacks by ignoring potentially malevolent requests. For example, the framework proposed by Ficco & Rak (2012) prevents requests containing an excessive number of nested XML elements from reaching the cloud application.

Ficco & Rak (2015) present a Slowly-increasing Polymorphic DDoS Attack Strategy (SIPDAS), which characterizes DoS attacks that dynamically modify their behavior to evade pattern detection algorithms. Ficco and Rak propose that both the consumption of computational resources and the intensity of incoming requests be considered in order to detect SIPDAS attacks.

Reddy (2014) discusses challenges in the design of security in cloud-based cyber-physical systems (CCPS). Reddy points out that attacks against such a system in the physical domain can have security implications in cyberspace and vice versa.

Rahman et al. (2013) propose a security framework for advanced metering infrastructure (AMI) networks, which allow power consumption to be monitored and controlled in smart grids. A typical AMI network consists of heterogeneous interconnected cyber-physical components as follows (Rahman et al., 2013). The network typically includes millions of smart meters, thousands of collectors, and one or several head-ends. A meter periodically reports energy usage data to a specific collector, which relays it to a head end. A collector additionally forwards control commands from a head-end to the meters. The meters may be connected to a collector either directly or through a network of meters. Security threats to an AMI include typical cyber threats, such as DoS attacks against a head-end through numerous compromised collectors. The security framework proposed by Rahman et al. consists of a parser module, which extracts formal models from configuration templates and constraints, a verification engine, which detects unsatisfied constraints and produces a threat report, as well as a diagnosis module, which analyzes the threat reports and creates remediation plans.

Gao et al. (2013) have identified, classified, and analyzed the typical security threats in each of the three CPS layers (physical, network, and application layers). The threats include physical attacks, sybil attacks, trap doors, and distributed denial of service (DDoS). Furthermore, Gao et al. propose countermeasures such as node authentication, encryption, and access control policy to mitigate the

threats. In addition, Gao et al. classify potential CPS vulnerabilities as originating from either management and policy, network, or platform.

Fletcher & Liu (2011) propose an analytic methodology for identifying security threats to cyber-physical system and deriving security policies against them. The process involves the identification and prioritization of security requirements. An appropriate security policy is then determined from each requirement Fletcher & Liu (2011).

Howser & McMillin (2013) present a model that accommodates complex security domains. They argue that the traditional view of security involving a strict division into safe and unsafe zones is overly simplistic, since security partitions frequently overlap. In addition, they apply the model in examining the control of a passenger car, which is typically in one of four states: controlled by the driver, controlled by a traction control system, controlled remotely, or malfunctioning (Howser & McMillin, 2013).

Wan et al. (2014) propose context-aware cloud services for vehicular cyber-physical systems. They identify three architecture types for context-aware vehicular clouds. In the first type, vehicles access cloud services via roadside infrastructure. In the second type, a set of vehicles allow external users to utilize their computing resources. The third type combines the former two.

Wan et al. (2014) additionally propose including context-awareness in vehicular security mechanisms. The implementation of context-aware vehicular security mechanisms requires a framework including data collection, malicious behavior detection, trust management, and policy management (Wan et al., 2014).

Puttonen, Afolaranmi, et al. (2015) have analyzed security challenges in two examples of cloud-based cyber-physical systems. The next two sections will discuss both security issues prevalent in the systems and potential strategies to mitigating them.

3. Security Challenges in Factory Automation and Smart Mobility Services

3.1 Security in Factory Automation

Modern factory automation systems frequently consist of devices providing web service interfaces. Through the service interfaces, operation requests can be sent to the devices. A typical conveyor service interface would provide product transportation capabilities, while a robotic workstation service interface would provide assembly operation capabilities. Furthermore, an automated guided vehicle might respond to object load and unload requests, as well as requests to navigate to new locations on the factory floor. Such cyber-physical devices have become somewhat common with the advent of the service-oriented architecture (SOA) paradigm (Jammes et al., 2005). However, with unrestricted access to the web service interfaces, an attacker could severely disrupt the operation of the production system and potentially cause significant economic losses.
Fig. 1 demonstrates the cyber-physical nature of an advanced production system. In the figure, the embedded conveyor device controller exposes a web service interface, through which pallet

transfer operations can be remotely started and stopped. Furthermore, the entire system is monitored and orchestrated by two web services, *Service Orchestrator* and *Ontology Service*, which are hosted on a compute cloud. The latter two services can be classified as *orchestration services*. Typically, the *domain services*, such as *Conveyor Service*, would greatly outnumber the orchestration services.

Service Orchestrator is mainly responsible of composing and invoking the domain services, so that they jointly achieve production goals submitted by clients through the *AchieveGoal* operation. The service additionally receives state change notifications from the domain services and accordingly updates the ontology model hosted by *Ontology Service*. In the figure, the conveyor device sends *StateChanged* event notifications to *Service Orchestrator*. The notifications signal events such as conveyor belt activations and changes in pallet locations. Together, the orchestration services may form complex frameworks that compose domain services based on their semantic descriptions. Puttonen (2014) has described such a service composition framework in detail, and Puttonen, Lobov et al. (2015) have elaborated on the planning processes required in the service composition.

The main security risks observable in the system of Fig. 1 involve unauthorized operation requests. Firstly, an attacker could send a single malevolent request to cause damage to the equipment or simply halt the system. Secondly, an attacker could utilize a software program to send a large number of requests with the intent of overburdening the processing capability of the domain services or the orchestration services. Furthermore, the attacker could harness a swarm of compromised computer systems to send such requests, effectively launching a distributed denial of service (DDoS) attack.

The domain services typically need to communicate only with other local production devices and a relatively limited number of orchestration services in their local network. Thus, they are virtually shielded from direct unauthorized service requests. However, at least the *Service Orchestrator* must be able to receive external production requests and therefore be connected to the Internet. Moreover, the orchestration services form a somewhat centralized processing core, which poses a significant vulnerability for the entire system.

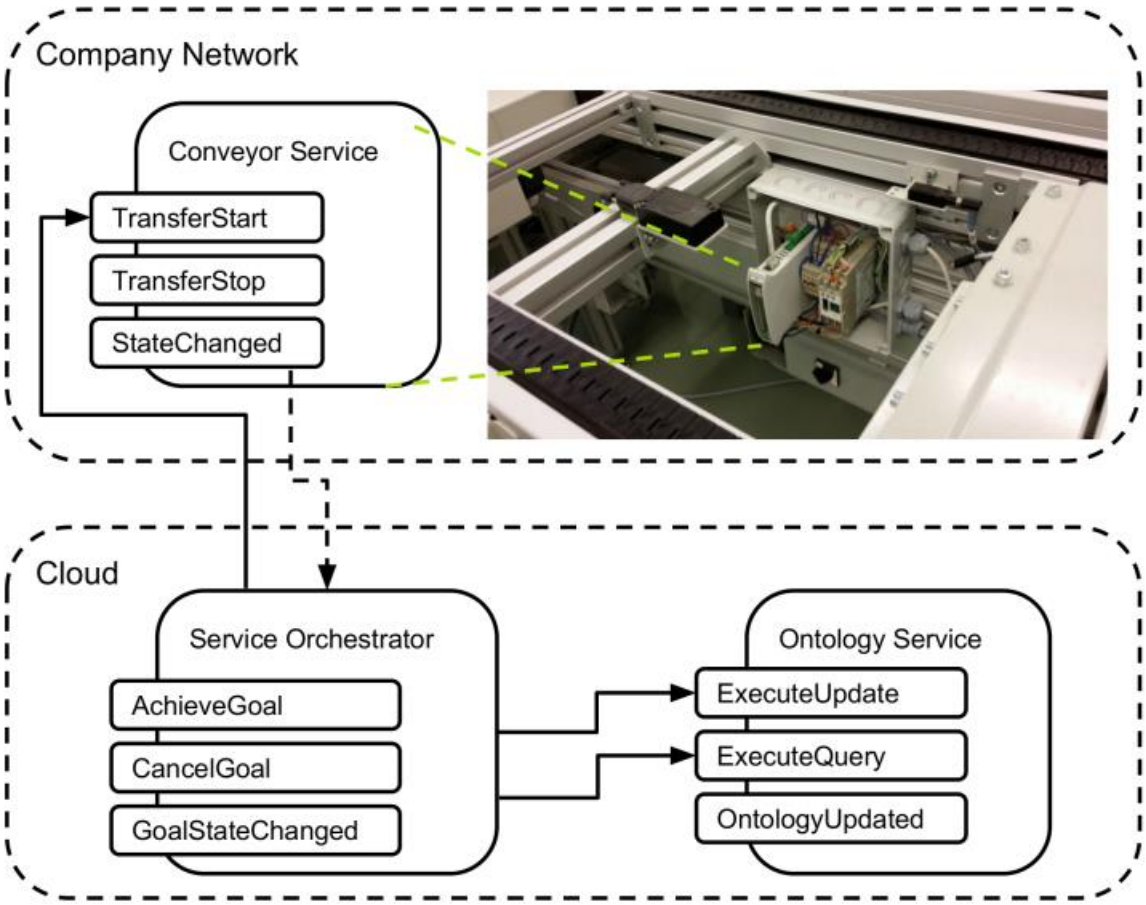

Fig. 1. Cyber-physical production systems consist of devices and web services

The orchestration services require significant computing power, and the computing requirements typically vary depending on incoming production requests. Therefore, the hosting of the services should be offloaded to the cloud to avoid underutilization of expensive computer systems. The use of cloud resources for such critical operations requires that the cloud service providers are carefully selected.

Denial of service attacks against the orchestration services could prevent them from maintaining an up-to-date production system model. Moreover, because *Service Orchestrator* conducts AI (Artificial Intelligence) planning, which is computationally intensive, excessive requests render the service unable to promptly process production goals submitted.

An approach to orchestration of web services deployed on cloud resources was described in an earlier research work (Puttonen et al., 2011). While the research proposed a method of balancing workload between cloud resources, it largely omitted security aspects.

3.2 Security in Smart Mobility Services

Smart mobility services provide users with detailed journey options augmented with additional

information, such as cost, duration, CO_2 emissions, and calorie consumption. Since the services relate to physical activities, transportation, and they guide the actions of the users, they can be considered cyber-physical systems.

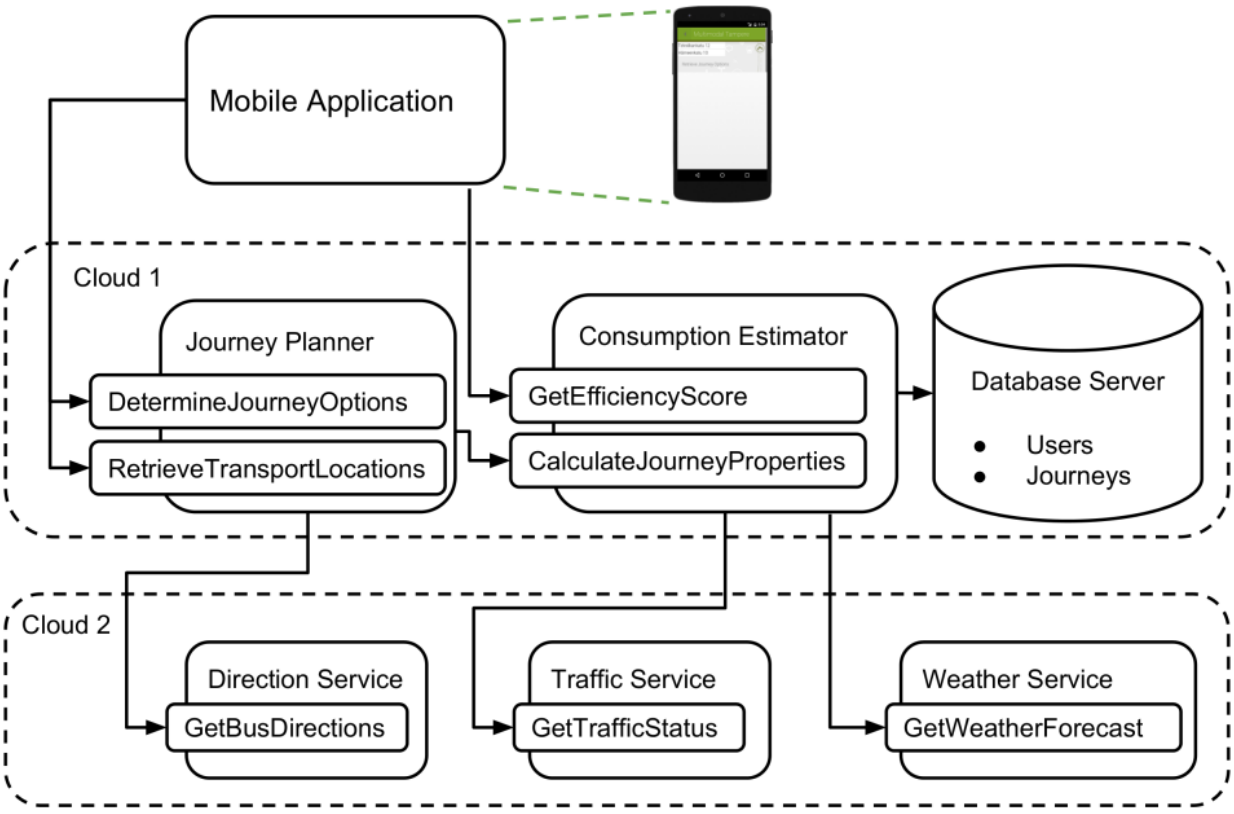

Fig. 2. Smart mobility services typically consist of several components deployed over multiple clouds.

On the one hand, smart mobility services may be entirely based on virtualized computing resources including no physical components. On the other hand, the services inherently require physical sensing capabilities to detect, for example, traffic and weather conditions. In addition, physical actuation capabilities enable the services to provide users with transportation options on request. In the broader context, the systems may even contain actual physical actuators. Hence, smart mobility services may be considered cyber-physical systems.

Fig. 2 illustrates the possible structure of a smart mobility service. The service consists of two web services, *Journey Planner* and *Consumption Estimator*, as well as a database server. The user interacts with the *Journey Planner* service, which invokes external services to acquire directions between the journey start and end points specified by the user. In addition, *Journey Planner* provides an operation through which the user may inquire the locations of transportation facilities, such as parking spaces, bike rental posts, and bus stops.

In finding directions, *Journey Planner* employs various third-party services. Consequently, *Journey Planner* acquires a list of journey options employing different transportation modes, such as walking, driving, cycling, and public transportation. To augment the journey options with further details, *Journey Planner* invokes the *Consumption Estimator* service, which calculates property values for the journey options based on values retrieved from a database server and external services, such as weather data providers. Finally, *Journey Planner* returns the augmented journey options to the user.

User interaction with Smart mobility services is most efficiently achieved through a mobile application, since the web services are typically hosted on cloud resources, and they aim to assist users in transportation. Through the mobile application, the end user may compare different journey options and select the one that has the lowest environmental implications. *Journey Planner* then informs *Consumption Estimator*, so that it may update the database. This allows the mobile client to later retrieve information on the user's performance.

The component deployment configuration in Fig. 2 is arbitrary in that none of the components actually need to reside on the same cloud. However, the figure emphasizes that the deployment of the external services is fixed and determined by their authors. Thus, the external services will typically reside on different clouds than the three application core components. Regardless of the deployment configuration chosen, the smart mobility service should function in a similar manner to the end user. In this respect, smart mobility services typically fulfill the definition of multi-cloud applications, whose components are deployed on the resources of several cloud service providers (Rios et al., 2015).

While the external services invoked by *Journey Planner* and *Consumption Estimator* may employ sensors and actuators subject to physical threats, the actual mobility application mainly exists in cyberspace and is therefore subject to traditional information security threats. For example, the database server hosts most of the data processed by the application, such as user profiles. Therefore, any communication between the database and the *Consumption Estimator* service should be carefully secured.

By capturing the messages exchanged between the database and web services, an attacker could gain access to personal information of the users, such as details on the journeys taken and transportation modes employed. Moreover, it the database is hosted on an unreliable cloud service provider, the confidentiality of the entire data content is compromised.

Because the *Journey Planner* and *Consumption Estimator* depend on several third-party services, they are vulnerable to denial of service attacks, which aim to increase the response latency of the target service. Therefore, the two services should be tolerant to failures of the sub-services. For example, having detected increased response latency in a weather service, *Consumption Estimator* should automatically switch to another data source. Ideally, *Consumption Estimator* would select an appropriate sub-service based on on-line semantic web service descriptions instead of relying on hard-coded service provider lists.

Since the smart mobility services process private user information, they must employ sophisticated user authentication and authorization methods. Otherwise, an attacker could directly access

arbitrary user information under the guise of fraud identities. However, such methods would be vulnerable to so-called man-in-the-middle-attacks, in which an attacker impersonates the smart-mobility services, thereby obtaining the secret keys used to encrypt the communication.

Rios et al. (2015) propose a framework supporting the development of security-aware multi-cloud applications. In addition, they describe the security challenges that the framework will solve in smart mobility services such as the one considered in this article.

4. Security Risk Mitigation Strategies

4.1 Reducing Security Risks in Factory Automation Systems

Because unauthorized access forms the most obvious security risk in a cyber-physical production system, the security of such systems can be improved by shielding the domain services and orchestration services within a protected network. The services inside the network have access to the Internet but the network firewall blocks all incoming traffic. While the orchestration services require considerable computing power, the mere hosting of the software components requires negligible resources. All computationally intensive tasks can still be offloaded to the cloud. Fig. 3 illustrates an arrangement where all orchestration services are inside a protected network, albeit *Service Orchestrator* relies on an Artificial Intelligence (AI) planning service, *Planner*, which is deployed on the cloud.

Typically, the *Planner* service would send event notifications signaling the completion of planning processes, and *Service Orchestrator* would subscribe to receive them. However, *Service Orchestrator* is unable to receive such notifications, since all incoming messages are blocked by the company network firewall. To acquire the solution plans produced by the *Planner* service, *Service Orchestrator* must periodically invoke the *RetrieveSolution* operation of the former service.

Since *Service Orchestrator* executes the solution plans produced by the *Planner* service, potential attackers should be inhibited from assuming the role of the latter service. Such malevolent action can be prevented by deploying the service in a virtual private network that inhibits access from outside the company network.

The protection provided by the network firewall additionally renders external clients unable to directly issue production orders to *Service Orchestrator*. To enable authorized clients to submit production requests, an extra level of indirection is necessary. An *Orchestrator* service is deployed in a safe location, typically a trusted cloud service provider, and secured from unauthorized access. An *Orchestration Engine* service is deployed within the company network to monitor the workload on the local *Service Orchestrator*, and retrieve new production goals from *Orchestrator* by invoking its *RetrieveGoal* operation whenever the number of solution plans in execution phase is below a certain threshold. To prevent other production systems from initiating the same production task, *Orchestration Engine* additionally invokes the *SetOrderStatus* operation to tag the order with the status code *IN_PROGRESS*. When *Service Orchestrator* finally notifies *Orchestration Engine* of either completing the goal or failing, the latter service invokes the *SetOrderStatus* operation to tag the order with the status code *COMPLETED*. To avoid unnecessary polling, *Orchestration Engine*

subscribes to *GoalStateChanged* event notifications from *Service Orchestrator* instead of repeatedly invoking the *ListCurrentGoals* operation.

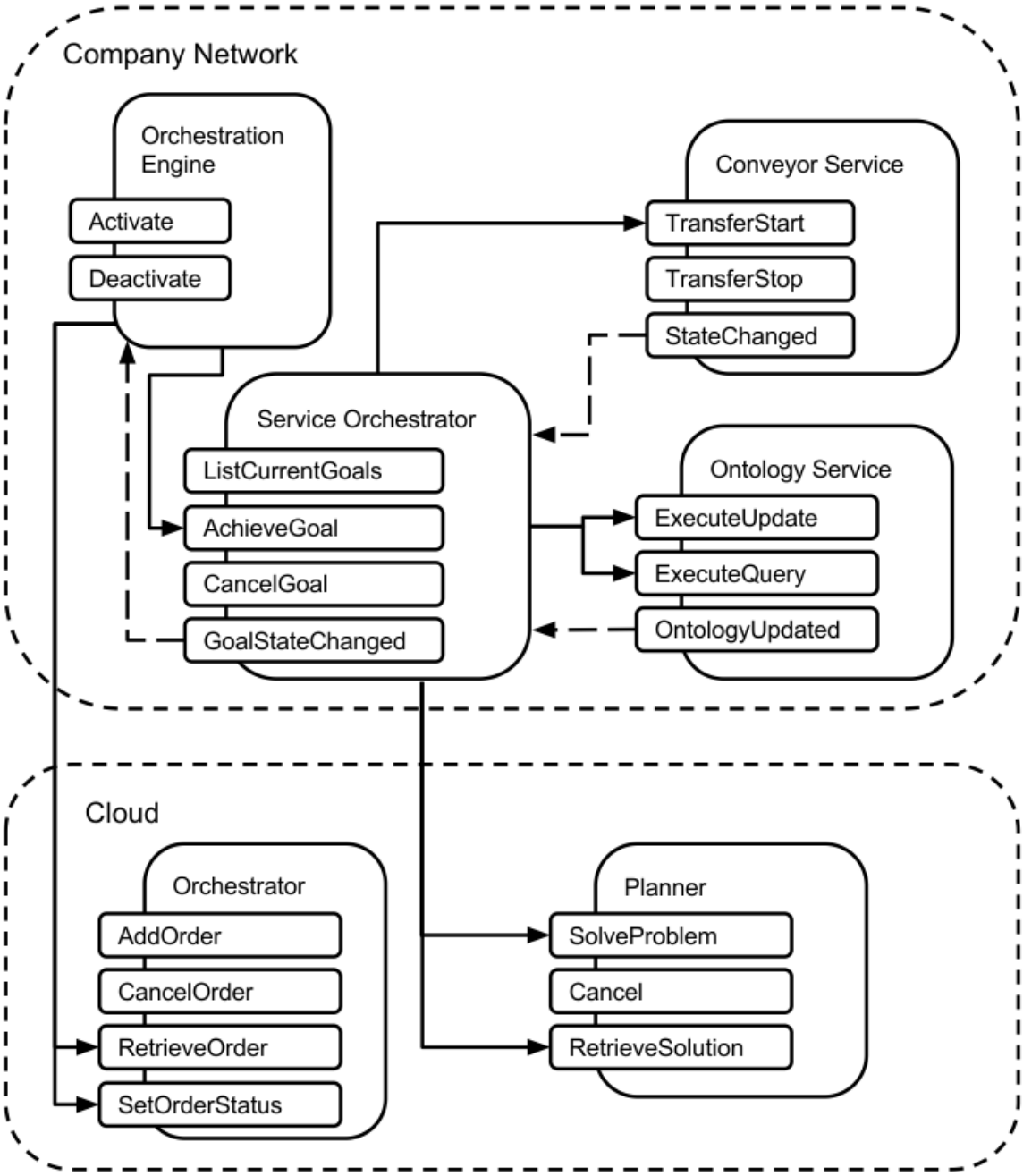

Fig. 3. Cyber-physical production systems can be protected by enclosing them in a protected network.

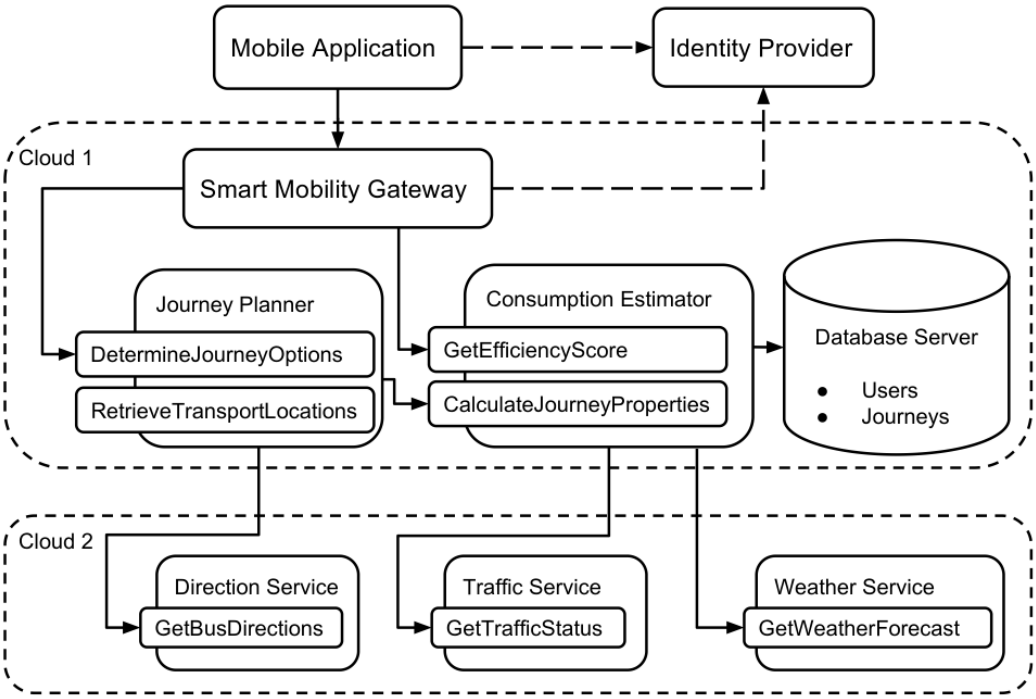

Fig. 4. Smart mobility services may rely on external identity providers.

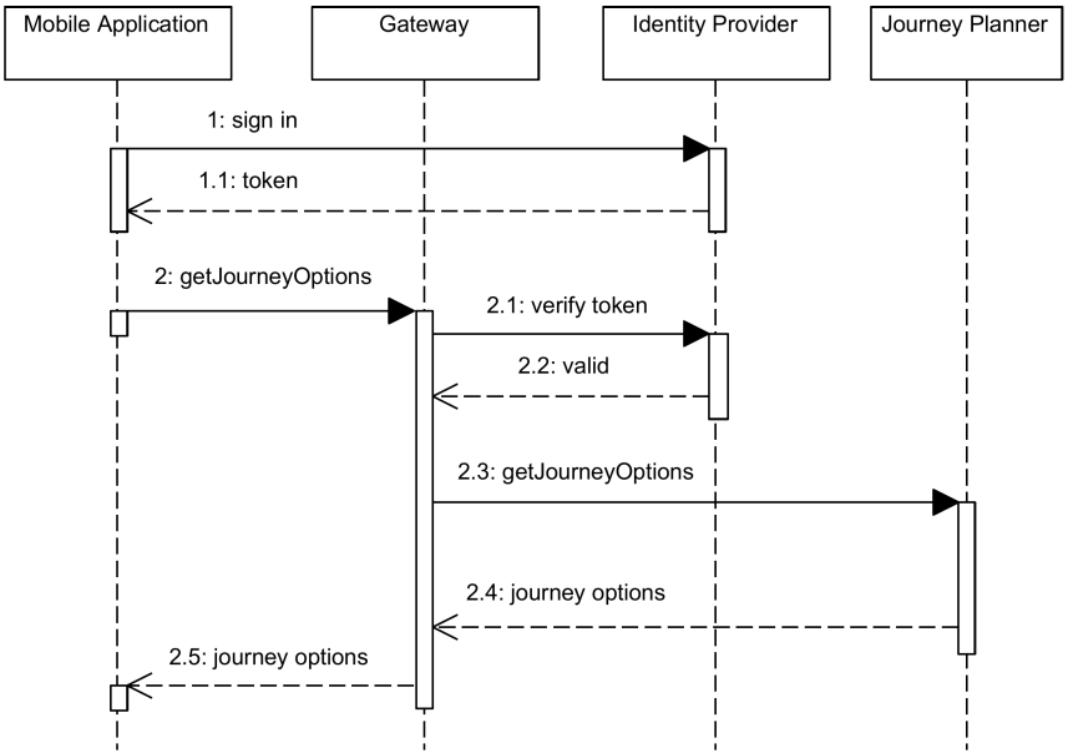

Fig. 5. The mobile application logs the user in through external identity providers.

4.2 Reducing Security Risks in Smart Mobility Services

The communication between the mobile client application and the smart mobility services contains information on user transportation actions. As the services are hosted on the cloud, they may traverse various servers before reaching the intended client. Therefore, the messages should be encrypted to render them indecipherable to attackers capturing the data packages. Such encryption is made possible by the HTTPS (HTTP over SSL) protocol, which encrypts HTTP (Hypertext Transfer Protocol) messages through the SSL (Secure Sockets Layer) protocol.

Unfortunately, smart mobility services typically rely on several external services using plain HTTP messages. Nonetheless, the service requests are triggered by several users, so it should be difficult for an attacker to determine, for example, which end user has triggered the request for certain route directions.

To reduce the risk of fraudulent user access to the smart mobility services, they should be collected behind a central gateway that employs a trusted identity provider. Such identity management services are provided by several popular social media and on-line productivity tools. The approach would reduce the risk of man-in-the-middle attacks, since the client credentials would not be transmitted to the smart mobility services. As illustrated in Fig. 4, the mobile application initially needs to login to the identity provider to receive an authentication token. Then it can invoke any operations provided by the smart mobility services as long as it includes the token in each request. The smart mobility services will use the token to retrieve the user identifier from the identity provider and map it to a user in the local database.

To eliminate the dependency to external identity providers, the *Smart Mobility Gateway* may allow users to register directly with a user name and password. However, also in this case the gateway grants the user an access token to be included in all subsequent communication during the session. The *Gateway* generates the token value from the user account data and a secret key. With the secret key, the gateway is also able to translate an access token into the correct user account.

The sequence diagram of Fig. 5 illustrates the communication flow between the different components when an identity provider is employed in user authentication. The smart mobility services gateway validates the authentication token on each request, and the subservices, such as *Journey Planner* only provide business functionality.

Similarly, to the Factory Automation scenario, a large number of the smart mobility services can be enclosed in a private network, shielding them from unauthorized access. In fact, only the central gateway component must be externally accessible. The individual subservices, such as *Journey Planner* and *Consumption Estimator*, may largely ignore security aspects, which significantly simplifies their design and implementation.

5. Conclusions

Advanced security measures should be implemented for open, standards-based, inexpensive devices, such as modern production device controllers. Such security measures must be developed through systematic methods, and they should consider the limited device resources. Nevertheless,

as the computational capacity of embedded controllers rapidly increases, some devices may even support implementation of the open web services security standards.

Since cyber-physical systems frequently involve web services that provide control over physical devices, access to the services should be restricted. However, it is challenging to ensure the confidentiality of communication between the services, particularly when several of them are developed by external authors and deployed on unknown infrastructure. When possible, access to individual services should be restricted to central gateway components. Enclosing the sub-services in private networks significantly reduces the number of system components vulnerable to attacks and effectively focuses the security mechanisms onto more manageable subsystems.

This article has briefly introduced two types of cyber-physical systems: production systems and smart mobility services. The physical aspect is considerably more prominent in factory automation systems, thereby complicating security considerations. However, smart mobility services inherently process and store private end user information. In public infrastructures, confidentiality and privacy of the end users becomes the primary goal besides functional application requirements. While smart mobility services typically have limited physical interfaces, they frequently rely on external service providers, which makes it difficult to ensure that adequate security policies are enforced.

The cloud infrastructure should include dedicated components for ensuring application security. The components can focus on the three basic security aspects: Confidentiality, Integrity, and Availability. The first is achieved mainly through encryption, integrity requires authentication mechanisms implemented at the device level, and availability can be achieved through redundancy of communication channels. Nonetheless, careful selection of cloud service providers is a prerequisite to achieving a secure cloud infrastructure.

Acknowledgement

The work described in this article was carried out in the context of the research projects *MUlti-cloud Secure Applications* (MUSA) and *ICT Cloud-based Platform and Mobility Services: Available, Universal and Safe for all Users* (MoveUs). MUSA is under the EU Research and Innovation programme Horizon 2020 (H2020), grant agreement number 644429, and MoveUs is under the European Commission's 7th framework programme, grant agreement number 608885.

References

Agosta, G., Barenghi, A., & Pelosi, G. (2014, October), Securing software cryptographic primitives for embedded systems against side channel attacks. In 2014 International Carnahan Conference on Security Technology (ICCST) (pp. 1-6). http://dx.doi.org/10.1109/CCST.2014.6987032

Albadi, M., & El-Saadany, E. (2007, June). Demand Response in Electricity Markets: An Overview. In IEEE Power Engineering Society General Meeting, 2007 (pp. 1-5).

Choka, A., & Abawajy, J. (2012, September). Detecting and Mitigating HX-DoS Attacks against Cloud Web Services. In 2012 15th International Conference on Network-Based Information Systems (NBiS) (pp.

429-434).

Cobb, W.E., Balwin, R.O., & Laspe, E.D. (2013, June). Leakage Mapping: A Sytematic Methodology for Assessing the Side-Channel Information Leakage of Cryptographic Implementations. ACM Trans. Inf. Syst. Secur., 16(1), 2:1-2:29

Ficco, M., & Rak. M. (2012, January). Stealthy Denial of Service Strategy in Cloud Computing. IEEE Transactions on Cloud Computing, 3(1), 80-94. http://dx.doi.org/10.1109/TCC.2014.2325045

Ficco, M., & Rak. M. (2012, July). Intrusion Tolerance in Cloud Applications: The mOSAIC Approach. In 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS) (pp. 170-176). http://dx.doi.org/10.1109/CISIS.2012.202

Fletcher, K., & Liu, X. (2011, June). Security Requirements Analysis, Specification, Prioritization and Policy Development in Cyber-Physical Systems. In 2011 5th International Conference on Secure Software Integration Reliability Improvement Companion (SSIRI-C) (pp. 106-113).

Gao, Y., Peng, Y., Xie, F., Zhao, W., Wang, D., Han., … Li, Z. (2013, October). Analysis of Security Threats and Vulnerability for Cyber-Physical Systems. In 2013 3rd International Conference on Computing Science and Network Technology (ICCSNT) (pp. 50-55)

Garcia, M.V. and Perez, F. and Calvo, I. and Moran, G. (2014, September). Building industrial CPS with the IEC 61499 standard on low-cost hardware platforms. In 2014 IEEE Emerging Technology and Factory Automation (ETFA) (pp. 1-4).

Hiray, S. & Ingle, R. (2013, November). Context-Aware Middleware in Cyber Physical Cloud (CAMCPC). In 2013 International Conference on Cloud Ubiquitous Computing Emerging Technologies (CUBE) (pp. 42-47). http://dx.doi.org/10.1109/CUBE.2013.18

Howser, G. & McMillin, B. (2013, July). A Multiple Security Domain Model of a Drive-by-Wire System. In Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual (pp. 369-374). http://dx.doi.org/10.1109/compsac.2013.62

Jammes F., Smit H., Martinez Lastra, J. L. & Delamer, I. M. (2005, Sept). Orchestration of service-oriented manufacturing processes. In 2005 IEEE Conference on Emerging Technologies and Factory Automation (vol. 1, p. 617-624) http://dx.doi.org/10.1109/ETFA.2005.1612580

Lei, L., Wang, Y., Zhou, J., Zha, D., & Zhang, Z. (2013, July). A Threat to Mobile Cyber-Physical Systems: Sensor-Based Privacy Theft Attacks on Android Smartphones. In 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (pp. 126-133)

Liu, L., Zhang, X., Yan, G., & Chen, S. (2009). Exploitation and Threat Analysis of Open Mobile Devices. In Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (pp. 20-29). New York, NY, USA http://dx.doi.org/10.1145/1882486.1882493

Mohan, A., & Mashima, D. (2014, May). Towards Secure Demand-Response Systems on the Cloud. In 2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 361-366). http://dx.doi.org/10.1109/DCOSS.2014.47

Puttonen, J. (2014). A Sematically Enhanced Approach for Orchestration of Web Services in Factory Automation Systems (Dr. Tech. thesis, Tampere University of Technology, Tampere, Finland). Retrieved 2014-09-15, from http://dspace.cc.tut.fi/dpub/handle/123456789/22371

Puttonen, J., Afolaranmi, S. O., Gonzalez Moctezume, L., Lobov, A., & Martinez Lastra, J. L. (2015, November). Security in Cloud-based Cyber-physical Systems. In Workshop on Security and Privacy in Systems and Communication Networks (SecureSysComm 2015). Krakow, Poland. http://dx.doi.org/10.1109/3pgcic.2015.30

Puttonen, J., Lobov, A., & Martinez Lastra, J. L. (2011, January). An Approach to Service Deployment to the Service Cloud. In ICONS 2011: The Sixth International Conference on Systems (pp. 122-127). St. Maarten, The Netherlands Antilles.

Puttonen, J., Lobov, A., Cavia Soto, M. A., & Martinez Lastra, J. L. (2015, October). Planning-based semantic web service composition in factory automation. Advanced Engineering Informatics, 29(F4), 1041-1054. http://dx.doi.org/10.1016/j.aei.2015.08.002

Rahman, M., Al-Shaer, E. & Bera, P. (2013, March). A Noninvasive Threat Analyzer for Advanced Metering Infrastructure in Smart Grid. IEEE Transactions on Smart Grid, 4(1), 273-287. http://dx.doi.org/10.1109/TSG.2012.2228283

Reddy, Y. (2014, December). Cloud-Based Cyber Physical Systems: Design Challenges and Security Needs. In 2014 10th International Conference on Mobile Ad-hoc and Sensor Networks (MSN) (pp. 315-322).

Rios, E., Iturbe, E., Orue-Echevarria, L., Rak, M., & Casola, V. (2015, May). Towards Self-Protective Multi-Cloud Applications - MUSA – a Holistic Framework to Support the Security-Intelligent Lifecycle Management of Multi-Cloud Applications. In Proceedings of the 5th International Conference on Cloud Computing and Services Science (p. 551-558). Lisbon, Portugal. http://dx.doi.org/10.5220/0005492905510558

Shellshear, E., Berlin, R., & Carlson, J. (2015, March). Maximizing Smart Factory Systems by Incrementally Updating Point Clouds. IEEE Computer Graphics and Applications, 35(2), 62-69. http://dx.doi.org/10.1109/MCG.2015.38

Wan, J., Zhang, D., Zhao, S., Yang, L., & Lloret, J. (2014, August). Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. IEEE Communications Magazine, 52(8), 106-113. http://dx.doi.org/10.1109/MCOM.2014.6871677

Xie, F., Lu, T., Guo, X., Liu, J., Peng, Y., Gao, Y. (2013, October). Security Analysis on Cyber-physical System Using Attack Tree. In 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (pp. 429-432). http://dx.doi.org/10.1109/iih-msp.2013.113