Research Article

A Cross-Layer Security Scheme of Web-Services-Based Communications for IEEE 1451 Sensor and Actuator Networks

Jun Wu,¹,² Ming Zhan,³ Bin Duan,⁴ and Jiang Liu⁵

¹ Research Institute for Secure Systems (RISEC), National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba 305-8568, Japan
² Global Information and Telecommunication Institute (GITI), Waseda University, Tokyo 169-0051, Japan
³ School of Electronic and Information Engineering, Southwest University, Chongqing 400714, China
⁴ College of Information Engineering, Xiangtan University, Xiangtan 411105, China
⁵ School of Electrical and Electronics Engineering, North China Electric Power University, Beijing 102206, China

Correspondence should be addressed to Ming Zhan; zmdjs@swu.edu.cn

Received 17 December 2012; Accepted 1 February 2013

Academic Editor: Mianxiong Dong

Copyright © 2013 Jun Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IEEE 1451 standard has been proposed to provide a common communication interface and transducer electric data sheet format for wired and wireless distributed applications in smart transducers (sensors and actuators). Currently, a unified Web service for IEEE 1451 smart transducers is a must. However, ensuring the security of web-services-based communications for IEEE 1451 smart transducers is an unsolved problem. In this paper, we proposed a cross-layer security mechanism that deals with the requirements of authentication, integrity, confidentiality, and availability across the communication process in IEEE 1451 smart transducers. The scheme contains three cross-layer components logically, including XML Encryption and Signature, SOAP Security Extension, and Web Services Description Language (WSDL) Security Checking. The former two components satisfy the requirements of confidentiality, availability, integrity, authentication, nonrepudiation, and freshness. The third component satisfies the requirement of availability, which can protect the system against denial-of-service (DoS) attack. The three cross-layer security components are integrated seamlessly in our scheme. To evaluate the overhead, we perform tests to evaluate the effect of message size on the performance of the access inquiry web service. The result supports the usefulness and feasibility of our scheme.

1. Introduction

In recent years, sensor and actuator have attracted a lot of attention recently due to their broad applications, ranging from industrial automation to environmental condition monitoring and control-to-intelligent transportation system to homeland defense [1–5]. A smart transducer is a compact unit containing a sensor or actuator element, a microcontroller, a communication controller, and the associated software for signal conditioning, calibration, diagnostics, and communication [6–8]. A smart transducer can enable novel application in and beyond measurement, monitoring, control, and actuating [9].

The behaviors to smart transducers generally call for distributed and remote architecture [10–12]. And these systems usually require a variety of networked interconnections and telecommunication technologies for measurement and control, and the devices are usually made by different manufacturers. Therefore, common and reliable communication interface and data format are important for smart transducers. As a consequence, the Instrumentation and Measurement Society’s Sensor Technology Technical Committee TC-9 at the Institute of Electrical and Electronics Engineers (IEEE) has been working to establish a group of smart sensor interface standards called IEEE 1451 [13–18]. IEEE 1451 standard is proposed to provide a common communication interface and transducer electric data sheet format for wired and wireless distributed applications. It will eliminate the issue of proprietary communication systems utilizing a wide variety of protocols, labels, semantics, and so forth, thus enabling a transducer application to exchange information with different smart transducers independently of a vendor.
From a utility perspective, unified definitions of common data minimize conversion and recalculation of data values for evaluation and comparison in many application systems.

Recently, the working group of Kang Lee, who is the Chairman of the IEEE Instrumentation and Measurement Society’s Technical Committee on Sensor Technology and responsible for the establishment of the suite of IEEE 1451, proposed a unified Web service for IEEE 1451 smart transducers [19]. This work developed the IEEE 1451 standard to a new emerging unified Web service framework. An IEEE 1451 Network Capable Application Processor (NCAP) can be used as a Smart Transducer Web Services (STWS) provider, which provides asset of Web services for the STWS. STWS consumers, such as sensor alert system, OGC-SWE, or other applications, can find the STWS deployed and then invoke the STWS through Simple Object-access Protocol (SOAP)/Extensible Markup Language (XML) message. As a consequence, the use of Web service technologies provides the benefits of low implementation cost and ease of interoperability because Web services can implement service-oriented architectures (SOAs), which enable loosely coupled integration and interoperation of distributed heterogeneous system by using services as component elements in transducer networks. However, on the other hand, Web-services-based communication introduces the cyber security problem.

The importance of cyber security in sensor and actuator networks is widely recognized. Recently, schemes related to the cyber security for sensor and actuator networks have been widely investigated [20–24]. In particular, cyber security of Web-services-based communication for smart transducers must be implemented [25]. Security issues of communication for smart transducer are described in IEEE 1451.0. However, how the security issues are handled is up to the individual supplier and the responsibilities of communication protocol [13]. As a matter of fact, Web-services-based communication for smart transducers is a new emerging technology, in which few studies have been conducted for security. A common method of implement security is based on a secure transport layer or network layer, which typically includes secure socket layer (SSL), transport layer security (TLS), and network layer security (NLS). For example, TLS and NLS are recommended to secure TCP/IP-based communication for wireless sensor and actuator networks in IEEE 1451.5. However, these security schemes provide security only in a secure channel, and not in files or databases. Furthermore, these techniques do not correspond with the web services architecture in which the intermediaries can manipulate the messages on their way. Once using a secure transport layer, intermediaries are not able to control the message [26, 27].

The Web Services Security (WS-Security) [28, 29] standard was produced by Advancing Open Standards for the Information Society (OASIS) in 2004. The Web Services Security (WS-Security) is an essential component of the Web services protocol stack to provide end-to-end integrity, confidentiality, and authentication capabilities to web services. End-to-end message security assures the participation of nonsecure transport intermediaries in message exchanges, which is a key advantage for web systems and service-oriented architectures. Some security schemes corresponding with WS-Security are proposed for e-mail system, enterprise services system, trust management, and so forth, but cannot be applied directly to smart transducers [28–31].

As a matter of fact, IEEE 1451 standards define a common communication interfaces for networked smart transducers, which include sensors and actuators. The research of sensor and actuator networks is an existing area. In this paper, IEEE 1451 sensor and actuator networks means the networked smart transducers which are based on the common interfaces of IEEE 1451. Because of the communication protocols and data format of IEEE 1451 sensor and actuator networks, the secure communication proposals should have their special features based on IEEE 1451 standards. On the other hand, this paper focuses on the Web services communication security. So the security topics of confidentiality, availability, integrity, authentication, nonrepudiation, and freshness are necessary for the IEEE 1451 sensor and actuator networks, which also must be based on the data format of IEEE 1451.

In this paper, we proposed a cross-layer security scheme for web-services-based communication for IEEE 1451 smart transducers. The rest of this paper is organized as follows. Section 2 analyzes the system architecture and security requirements of IEEE 1451 reference model. Section 3 presents the architecture and security measures of web services. Section 4 presents the proposed security scheme. Section 5 analyzes the security of the proposed scheme. Section 5 evaluates the implementation of the proposed scheme. Finally, Section 6 concludes this paper.

2. System Architecture and Security Requirements

2.1. IEEE 1451 Reference Model. IEEE 1451 standards define a common communication model to connect smart transducers to normalization of integrated, intelligent, and open distributed measurement and control systems (DMCSs). Figure 1 shows the IEEE 1451 reference model. The IEEE 1451 family of standards divides the parts of a smart transducer system into two general categories of devices. One is the Network Capable Application Processor (NCAP) that functions as a gateway between the users’ network and the transducer interface modules (TIMs) [13–18]. In the IEEE 1451 reference model, smart transducers connect with DMCS users through the user communication network. The user communication network is outside of the scope of the IEEE 1451 family of standards. It may be anything that the user desires. The only requirement that is placed on the NCAP is that the NCAP has the appropriate network interface hardware and software [13].

The communications between NCAP and TIM are based on IEEE 1451.X communication modules in both sides, which provide the low levels of the communications protocol [13]. DMCS users interact with smart transducers through public application programming interfaces (APIs) [13]. The applications run in NCAP or remote DMCS system interact with transducers through public application programming interfaces (APIs).
2.2. Unified Web Service for IEEE 1451 Smart Transducers. Web services are typically APIs or web APIs that are accessed via Hypertext Transfer Protocol and executed on a remote system hosting the requested services. Qualities like simplicity, code reuse, and interoperability are also making Web services a de facto standard in the context of DMCSs [32]. The IEEE 1451 working group proposed a unified Web service for IEEE 1451 smart transducers recently [19]. In fact, in the reference model in Figure 1, the NCAP application module is logically an optional complement to provide the functions and service to pass information across the interface between the DMAC users and NCAP [33]. For mapping to Web services, the method introduced in [19] designed smart transducer Web services (STWSs) in the NCAP application services component. Hence, an IEEE 1451 NCAP provides a set of Web services for the STWS, which acts as an STWS provider. As shown in Figure 1, a common basis for the members of the IEEE 1451 family of standards is provided to be interoperable [13]. Hence, STWSs based on the IEEE 1451.0 standard have been defined using Web Services Description Language (WSDL). The DMCS user applications and STWS provider communicate with each other using SOAP/XML messages. The communications between NCAP and TIM are based on IEEE 1451X [19].

2.3. Security Requirements. In this paper, we consider the security of the communication between the DMCS users and IEEE 1451 smart transducers purposely.

The security issues of the communication in the smart transducers are the responsibilities of IEEE 1451.X but not IEEE 1451.0 [13], and security in IEEE 1451.X is based on the specified communication protocol, such as bluetooth in IEEE 1451.5. However, the security of the communication between STWS consumers and STWS providers is an unsolved problem, which should be designed based on IEEE 1451.0 combined with Web services.

Recently, the security requirements of data exchange in sensor and actuator networks have been widely discussed, which include [34–37]:

(i) confidentiality: confidentiality or secrecy has to do with making information inaccessible to unauthorized users. A confidential message is resistant to revealing its meaning to an eavesdropper.

(ii) Availability: availability ensures the survivability of network services to authorized parties when needed despite denial-of-service (DoS) attacks. A denial-of-service attack could be launched at any OSI (open system interconnect) layer of a sensor network.

(iii) Integrity: integrity measures ensure that the received data is not altered in transit by an adversary.

(iv) Authentication: authentication enables a node to ensure the identity of the peer node with which it is communicating.

(v) Nonrepudiation: nonrepudiation denotes that a node cannot deny sending a message it has previously sent.

(vi) Access control: access control implement the process of identifying nodes as well as authorizing and granting nodes the access right to information or resources.

(vii) Freshness: this could mean data freshness and key freshness. Since all sensor networks provide some form of time-varying measurements, we must ensure that each message is fresh. Data freshness implies that each data is recent, and it ensures that no adversary replayed old messages.

The above requirements are in conformance with the security requirements of data exchange described in ISO/IEC 29180 working draft [38], which is standard under development for security framework for ubiquitous sensor network.

In the above security requirements, access control can be performed based on access control scheme. The security access control scheme introduced in [23] is useful for IEEE 1451 smart sensors.

Other security requirements, which are confidentiality, availability, integrity, authentication, nonrepudiation, and freshness, should be implemented based on IEEE 1451.0 integrated with Web services.

IEEE 1451 standards defines six transducer services [13, 19, 33], which are TimDiscovery, TransducerAccess, TransducerManager, TedsWithManager, CommManager, and AppCallback. Table 1 shows the security requirements of communication of the responding services.

![Figure 1: IEEE 1451 reference model.](image-url)
Table 1: Security requirements of communication process.

| Service                  | Security requirements for the communications                                      |
|--------------------------|-------------------------------------------------------------------------------------|
| TimDiscovery             | Availability, integrity, nonrepudiation, and freshness                             |
| TransducerAccess         | Confidentiality, availability, integrity, authentication, nonrepudiation, and freshness |
| TransducerManager, TedsManager, and CommManager | Confidentiality, availability, integrity, authentication, nonrepudiation, and freshness |
| AppCallback              | Integrity, nonrepudiation, and freshness                                           |

3. Web Services Architecture and Web Services Security

3.1. Web Services Architecture. Today, the ability to seamlessly exchange information between internal business units, customers, and partners is vital for success; yet most organizations employ a variety of disparate applications that store and exchange data in dissimilar ways and therefore cannot “talk” to one another productively. Web services have evolved as a practical, cost-effective solution for uniting information distributed between critical applications over operating system, platform, and language barriers that were previously impassable.

Web services [39] are in simple terms object methods exposed via HTTP using pure SOAP messages. The major components or layers of a Web Service Protocol Stack include:

1. Extensible Markup Language (XML) layer: providing a means for communicating over the Web using an XML document that both requests and responds to information between two disparate systems.

2. Simple Object Access Protocol (SOAP) layer: a XML Messaging specification, which allows the sender and the receiver of XML documents to support a common data transfer protocol for effective networked communication.

3. Web Services Description Language (WSDL) layer: playing an important role in the overall Web services architecture since it describes the complete contract for application communication.

4. Universal Description, Discovery and Integration (UDDI) layer: a platform-independent, Extensible Markup Language- (XML-) based registry, which represents a way to publish and find web services over the Web.

Figure 2 shows the protocol stack architecture of Web Services.

3.2. Web Services Security. Web Services Security is based on open W3C-approved XML standards [40, 41], which provide the security foundation for applications of Web services. The standards are platform neutral, thus promoting interoperability. Also, OASIS published the standards for defining the security expanding method for SOAP message exchange [42].

4. The Proposed Security Scheme

4.1. Basic Idea and Model. The basic idea and model of the proposed security scheme are shown in Figure 3. The goal of the security scheme is to satisfy the security requirements of the data exchange. The proposed approach can be viewed as “cross-layer design” at the messaging layer and description layer in Web Services protocol stack. The scheme contains three components logically, including XML Encryption and Signature, SOAP Security Extension, and WSDL Security Checking. The former two components satisfy the requirements of confidentiality, availability, integrity, authentication, nonrepudiation, and freshness. The third component satisfies the requirement of availability, which can protect the system against DoS attack. For mapping to Web Services, the security scheme is designed based on the layer architecture of Web services protocol stack. Also, the scheme is designed in conformance with the Web services security standard. Most important, the three components of the security scheme are based on IEEE 1415 transducer services, services API, and XML schema of API, respectively, which are defined in IEEE 1451 standards. As described in IEEE 1451.0, all text strings in the Transducer Electronic Data Sheet (TEDS) shall conform to W3C Recommendation Extensible Markup Language (XML) 1.0 (Second Edition).

4.2. Secure XML Messaging Layer. A security token represents a collection of claims, which is used to prove one’s identity and provide the foundation for ensuring the confidentiality, integrity, nonrepudiation, and freshness of the data. Web Services Security standard defined several security tokens, including X. 509 certificate token, username/password token, Kerberos token, and SAML token. The security token most commonly used in DMCSs and sensor networks is username/password token [43–48].

Table 2 lists the notations used throughout the description of the security scheme for ease of reference.

4.2.1. Secure Messages of TimDiscovery. Figure 8 shows the protocol of secure message of TimDiscovery. For generating the signature, the client node first generates a fresh nonce \( R_C \). Then, she computes the digital digest of her own password
together with the fresh nonce based on hash based message authentication code (HMAC). The password is stored in both the memory of client node and the server node. Clearly, the signature provides a nonrepudiation property. This is true because only the client node herself can generate it, and the fresh nonce guarantees its freshness. Next, nonce and created time are the additional elements to resist against the replay attack. Then, the client node generates a signature RequestMAC, which is for RequestParameters. RequestParameters is the original message of the access request. Next, the client node sends out Username, ReqParameters, T, R_{U}, and RequestMAC. After receiving the message from the client node, the server node retrieves the parameters from the message. Then, S computes the RequestMAC' based on the parameters from the message. After that, S verifies the signature through comparing RequestMAC and RequestMAC'. Then, a symmetric key is derived based on the password and a 16-bit random value G. Next, S computes the signature of response and the symmetric key. These values then are sent back to U. After U gets the message, U can derive the symmetric key.

### 4.2.2 Secure Messages of TransducerAccess, TransducerManager, TedsManager, and CommManager

Figure 9 shows the protocol of secure message of TransducerAccess, TransducerManager, TedsManager, and CommManager. The client node U firstly generates a signature, which includes the generation of a fresh nonce R_{U} and the computation of the digital digest of her own password together with the fresh nonce based on HMAC. As a matter of fact, the generation process of the signature in this section is similar to the process of TimDiscovery.

### Table 2: Notation used by the secure authentication protocol.

| Notation | Meaning |
|----------|---------|
| U        | A services consumer |
| S        | A services provider |
| Key      | Shared secret key between U and S for symmetric encryption and decryption |
| R_{A}    | A nonce generated by entity A, usually it is a randomized value to defend replay attack |
| T        | The created time of message |
| Salt     | A random number to for derive the symmetric key |
| (M_{1}, M_{2}) | Concatenation of two messages |
| HMAC (M) | Calculate MAC for message M based hash function |
| H (M)    | Apply one-way function to message M |
| {M}_{Key} | Encrypt message M by symmetric key |
| ReqParameters | Parameters involved in the request |
| ResParameters | Parameters involved in the response |

The password is stored in both the memory of client node and the server node. Next, she generates a security token ET based on username/password method. Nonce and created time are the additional elements to resist against the replay attack. Then, she encrypts the ReqParameters based on the symmetric key. ReqParameters is the original message of the access request. Next, the client node sends out \{ReqParameters\}_{Key}, ET, and ResponseMAC. After receiving the message and the security token from client node, server
node retrieves the password by corresponding C from the local database, then calculates the PasswordDigest, compares it with PasswordDigest, and authenticates the identity of client node as being equal or not. After verification, the sever node S sends an access response message, including a signature of response parameters, ResParameters, to ensure the integrity and nonrepudiation. After getting the response message, U will verify the signature of ResParameters and then derive ResParameters from the message.

4.2.3. Secure Messages of AppCallback. The security requirements of AppCallback are as same as those of TimDiscovery except that AppCallback lacks availability. In the proposed scheme, ensuring availability is the responsibility of the security design of the WSDL layer. In addition, as defined in IEEE 1451.0 standard, AppCallback is implemented when applications that need advanced features exist [13]. AppCallback is implemented after TimDiscovery, which means that the key for symmetric encryption and decryption has already been generated when AppCallback is implemented. Hence, at XML messaging layer, the security mechanisms for securing message of AppCallback is as same as those of TimDiscovery but key generation is not needed.

4.3. Secure WSDL Layer. The security design of message layer cannot deal with the requirements of availability because the XML encryption and decryption can only ensure the confidentiality, availability, integrity, authentication, nonrepudiation, and freshness. Current Web services architecture does not consider validation of Web services messages against WSDLs during message processing. This could pose a potential security risk to enterprise servers hosting Web services. We secure the availability at the WSDL layer.

In fact, the most important aspect of a Web service is the service description using the Web Services Description Language (WSDL) that describes the messages, types, and operations of Web service and the contract to which the Web service guarantees it will conform [49]. WSDL plays an important role in the overall Web services architecture since it describes the complete contract for application communication. Smart transducer Web services (STWSs) in [19] are defined using Web services WSDL. WSDL is extensible to allow the description of endpoints and their messages regardless of what message formats or networks protocols are used to communicate. We secure the WSDL layer security based on the method in [50].

The considerations above regarding SOAP message validation lead to the Web service firewall, called CheckWay. Figure 4 shows the integration of a Web service firewall between Web service client and server. The security WSDL compiler gets the Web service server’s Web service description, generates the corresponding XML message schema, “hardens” the description, and advertises the modified description to a Web service client. The CheckWay Gateway validates all SOAP messages against the schema, forwards the message if it is valid, and rejects the message if it is not. The next step is now to consider how to obtain an XML schema for the message validation and which problems regarding the firewalls performance emerge from the validation process. In order to answer the first question, a closer look at Web service client/server interaction and the Web service interface description is required. The compiling process is shown in Figure 5.

The SOAP message’s structure belonging to a Web service description is defined by information spread all over the description document. The description must be traversed and the information necessary for a specific service or operation must be merged into a message definition.

5. Security Analysis

The basic secure authentication protocol for TimDiscovery, TransducerAccess, TransducerManager, TedsManager, CommManager, and Callback can provide a nonrepudiation property because only the client node herself can generate it, and the fresh nonce guarantees its freshness. Integrity property
A port references a binding by the binding attribute.

A binding describes the web service operations.

Contains a same-structure operation referencing a message element.

Contains message parts referring to XML schema type/element.

Define XML data types and elements for SOAP.

End

Figure 5: Compiling a Web service description.

Figure 6: SOAP message length impact on process time.

Figure 7: Power consumption on sensor.

can also be proved based on HMAC. Moreover, after the nonce and created time are added into the data packet, the receiver can check whether the nonce has been received before or whether the message is created in a very recent time. Thereby, nonce and created time combined into data packets can resist replay attack. Also, we consider the DoS attack model that consists in injecting bogus messages into the system. And before verifying PasswordDigest, only a hash computation needs to be implemented. At the same time, before verifying PasswordDigest, few values need be stored. Therefore, our protocol can resist DoS attack to some extent.

Beside the basic above security, the authentication protocols of TransducerAccess, TransducerManager, TedsManager, CommManager use both symmetric encryption and MAC for the message. Therefore, these protocols can provide confidentiality and authentication for the communications.

6. Implementation

6.1. Time Overhead of Process of CheckWay Gateway. In this section, we present the important aspect of the performance results. We evaluate the effect of message size on the performance of the access inquiry Web services of CheckWay gateway. A laptop is used to simulate the CheckWay gateway, which includes the Intel Core i5 M520 and 2 GB memory. In the implementation, we varied the message size by increasing the number of XML elements contained in the response message. As shown in Figure 6, the time consumption of CPU required to process an account inquiry request depends on the number of elements returned in the response message. The longest message is 400 times larger than the smallest message, but the increase in CPU consumption is less than 5-fold. Please note that in the implementation a SOAP message of 50 elements contains 1 KB of data, but the message itself has a length of 2.3 KB because of the XML tags.

6.2. Power Consumption of Sensor. It is very important to verify the feasibility of the implementation of the proposed scheme on resource-constrained sensors. In this subsection, we estimate the energy consumption of sensor using PowerTOSSIM [51], which is an energy modeling extension of TOSSIM for the simulation of MICAz mote. Here, we take TimDiscovery message authentication as the example for evaluation. The energy consumption is measured for five components: CPU, RADIO, LED, SENSOR, and EEPROM. We fix the time of execution equal to 1200 simulated seconds, which is because the motes in PowerTOSSIM take boot time of 10 seconds. In our scheme, storing security data performed by EEPROM component and computations performed by CPU component slightly increase the energy consumption, where radio transmission is not always necessary and accordingly the RADIO component energy consumption is greatly reduced. As shown in Figure 7, the energy consumption of our scheme is acceptable for resource-constrained WSNs.
### 7. Conclusion

To secure the web-services-based communications for networked IEEE 1451 smart transducers, we proposed a cross-layer security mechanism, which is based on the layer architecture of Web services protocol stack. The security requirements are derived from IEEE 1451 and Web service communications, and the design is consistent with existing applications of IEEE 1451 web services communication utilities and an information security standard. Moreover, the scheme is designed in conformance with the Web Services Security standard. Most important, the three components of the security scheme are based on IEEE 1451 transducer services, services API, and XML schema of API, respectively, which are defined in IEEE 1451 standards. The effect of message size on the performance of the access inquiry web service is tested, which verifies the feasibility of our scheme. The proposed scheme provides an efficient reference security model of web-services-based communications for networked IEEE 1451 smart transducers.

### Acknowledgment

The authors would like to thank the guest editor and anonymous reviewers for their helpful and constructive comments.
References

[1] E. Y. Song and K. Lee, "Understanding IEEE 1451—networked smart transducer interface standard—what is a smart transducer?" IEEE Instrumentation and Measurement Magazine, vol. 11, no. 2, pp. 11–17, 2008.

[2] K. Ota, M. Dong, and X. Li, "TinyBee: mobile-agent-based data gathering system in wireless sensor networks," in Proceedings of the IEEE International Conference on Networking, Architecture, and Storage (NAS09), pp. 24–31, Hunan, China, July 2009.

[3] J. Liang, X. Zeng, W. Wang, and H. Chen, "L-shaped array-based elevation and azimuth direction finding in the presence of mutual coupling," Signal Processing, vol. 91, no. 5, pp. 1319–1328, 2011.

[4] S. R. Rossi, A. A. de Carvalho, A. C. R. da Silva et al., "Open and standardized resources for smart transducer networking." IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 10, pp. 3754–3761, 2009.

[5] B. Liu, H. Chen, Z. Zhong, and H. V. Poor, "Asymmetrical round trip based synchronization-free localization in large-scale underwater sensor networks," IEEE Transactions on Wireless Communications, vol. 9, no. 11, pp. 3532–3542, 2010.

[6] K. B. Lee and E. Y. Song, "Object-oriented application framework for IEEE 1451.1 standard," IEEE Transactions on Instrumentation and Measurement, vol. 54, no. 4, pp. 1527–1533, 2005.

[7] M. Dong, K. Ota, X. Li, X. Shen, S. Guo, and M. Guo, "HARVEST: a task-objective efficient data collection scheme in wireless sensor and actor networks," in Proceedings of the 3rd International Conference on Communications and Mobile Computing (CMC3), pp. 485–488, Qingdao, China, April 2011.

[8] L. Chen, W. Chen, B. Wang, X. Zhang, H. Chen, and D. Yang, "System-level simulation methodology and platform for mobile cellular systems," IEEE Communications Magazine, vol. 49, no. 7, pp. 148–155, 2011.

[9] H. Chen, G. Wang, Z. Wang, and H. So, "Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks," IEEE Transactions on Wireless Communications, vol. 16, no. 1, pp. 108–116, 2012.

[10] M. Staroswiecki, "Intelligent sensors: a functional view," IEEE Transactions on Industrial Informatics, vol. 1, no. 4, pp. 238–249, 2005.

[11] K. Ota, M. Dong, J. Wang, S. Guo, Z. Cheng, and M. Guo, "Dynamic itinerary planning for mobile agents with a content-specific approach in wireless sensor networks," in Proceedings of the 72nd IEEE Vehicular Technology Conference Fall (VTC2010-Fall), pp. 1–5, Ottawa, Canada, September 2010.

[12] G. Wang and H. Chen, "An importance sampling method for TDOA-based source localization," IEEE Transactions on Wireless Communications, vol. 10, no. 5, pp. 1560–1568, 2011.

[13] "IEEE standard for a smart transducer interface for sensors and actuators—common functions, communication protocols, and transducer electronic data sheet (TEDS) formats," IEEE Standards Board, IEEE Std 1451.0-2007, 2007.

[14] "IEEE standard for a smart transducer interface for sensors and actuators—network capable application processor (NCAP) information model," IEEE Standards Board, IEEE Std 1451.1-1999, 1999.

[15] "IEEE standard for a smart transducer interface for sensors and actuators—transducer to microprocessor communication protocols and transducer electronic data sheet (TEDS) formats," IEEE Standards Board, IEEE Std 1451.2-1997, 1997.

[16] "IEEE standard for a smart transducer interface for sensors and actuators—digital communication and transducer electronic data sheet (TEDS) formats for distributed multidrop system," IEEE Standards Board, IEEE Std 1451.3-2003, 2003.

[17] "IEEE standard for a smart transducer interface for sensors and actuators—wireless communication protocols and transducer electronic data sheet (TEDS) formats," IEEE Standards Board, IEEE Std 1451.5-2007, 2007.

[18] "IEEE standard for a smart transducer interface for sensors and actuators—Transducers to radio frequency identification (RFID) systems communication protocols and transducer electronic data sheet formats," IEEE Standards Board, IEEE Std 1451.7-2010, 2010.

[19] E. Y. Song and K. B. Lee, “STWS: a unified web service for IEEE 1451 smart transducers; IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 8, pp. 1749–1756, 2008.

[20] B. Panja, S. K. Madria, and B. Bhargava, "A role-based access in a hierarchical sensor network architecture to provide multilevel security," Computer Communications, vol. 31, no. 4, pp. 793–806, 2008.

[21] L. Maccari, L. Mainardi, M. A. Marchitti, N. R. Prasad, and R. Fantacci, "Lightweight, distributed access control for wireless sensor networks supporting mobility," in Proceedings of the IEEE International Conference on Communications (ICC’08), pp. 1441–1445, Beijing, China, May 2008.

[22] Y. Zhou, Y. Zhang, and Y. Fang, "Access control in wireless sensor networks," Ad Hoc Networks, vol. 5, no. 1, pp. 3–13, 2007.

[23] J. Wu and S. Shimamoto, "Usage control based security access scheme for wireless sensor networks," in Proceedings of the IEEE International Conference on Communications (ICC’10), May 2010.

[24] K. Ota, M. Dong, Z. Cheng, J. Wang, X. Li, and X. Shen, "ORACLE: mobility control in wireless sensor and actor networks," Computer Communications, vol. 35, no. 9, pp. 1029–1037, 2012.

[25] K. B. Lee and M. E. Reichardt, "Open standards for homeland security sensor networks—sensor interconnection and integration through Web access," IEEE Instrumentation and Measurement Magazine, vol. 8, no. 5, pp. 14–21, 2005.

[26] J. Viegra and J. Epstein, "Why applying standards to web services is not enough," IEEE Security and Privacy, vol. 4, no. 4, pp. 25–31, 2006.

[27] E. Kleiner and A. W. Roscoe, "On the relationship between web services security and traditional protocols," Electronic Notes in Theoretical Computer Science, vol. 155, no. 1, pp. 583–603, 2006.

[28] Oasis Consortium, WS-Security specification, 2004, https://www.oasis-open.org/.

[29] Z. Wu and A. C. Weaver, "Using web services to exchange security tokens for federated trust management," in Proceedings of the IEEE International Conference on Web Services (ICWS’07), pp. 1176–1178, Salt Lake City, Utah, USA, July 2007.

[30] M. Anlauf, D. Pavlovic, and A. Suenbuel, "Deriving secure network protocols for enterprise services architectures," in Proceedings of the IEEE International Conference on Communications (ICC’06), pp. 2283–2287, Istanbul, Turkey, July 2006.

[31] L. Liao and J. Schwenk, "Secure emails in XML format using web services," in Proceedings of the 5th IEEE European Conference on Web Services (ECOWS’07), pp. 129–136, Halle, Germany, November 2007.

[32] V. Viegas, J. M. D. Pereira, and P. M. B. S. Girão, "NET framework and web services: a profit combination to implement and enhance the IEEE 1451.1 standard," IEEE Transactions on
[33] E. Song and K. Lee, “Smart transducer web services based on the IEEE 1451.0 standard,” in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (IMTC’07)*, May 2007.

[34] X. Chen, K. Makki, K. Yen, and N. Pissinou, “Sensor network security: a survey,” *IEEE Communications Surveys and Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.

[35] F. Hu and X. Cao, “Security in wireless actor & sensor networks (WASN): towards a hierarchical re-keying design,” in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’05)*, pp. 528–533, April 2005.

[36] M. Shao, S. Zhu, W. Zhang, G. Cao, and Y. Yang, “PDCS: security and privacy support for data-centric sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1023–1038, 2009.

[37] A. S. Tanenbaum, *Computer Networks*, Prentice Hall, Upper Saddle River, NJ, USA, 4th edition, 2003.

[38] “Working Draft of ISO/IEC Draft Standard for Telecommunications and Information Exchange between Systems—Security framework for ubiquitous sensor network,” ISO/IEC Unapproved Draft Std ISO/IEC, 29180, May 2008, http://isotc.iso.org/livelink/livelink?func=ll&objId=8158657&objAction=Open&vernum=1.

[39] M. MacDonald, *Microsoft .NET, Distributed Applications: Integrating XML Web Services and .NET Remoting*, Microsoft, Redmond, Wash, USA, 2003.

[40] W3 Consortium, XML Encryption specification, http://www.w3.org/.

[41] W3 Consortium, XML Signature specification, http://www.w3.org/TR/xmldsig-core/.

[42] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, “Web services security: SOAP message security 1.1,” OASIS Standard Specification, 2006.

[43] S. A. Kiprushkin, N. A. Korolev, and S. Y. Kurskov, “Distributed information measurement and control system for research and education in physics,” in *Proceedings of the 2nd International Conference on Systems (ICONS’07)*, April 2007.

[44] J. Hieb, J. Graham, and S. Patel, “Security enhancements for distributed control systems,” in *Critical Infrastructure Protection*, E. Goetz and S. Shemoi, Eds., pp. 133–146, Springer, Boston, Mass, USA, 2007.

[45] A. Aiello, D. L. Carni, D. Grimaldi, and G. Guglielmelli, “Wireless distributed measurement system by using mobile devices,” in *Proceedings of the 3rd IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS’05)*, pp. 316–319, Sofia, Bulgaria, September 2005.

[46] A. Aiello, D. L. Carni, D. Grimaldi, G. Guglielmelli, and F. Lamonaca, “Wireless distributed measurement system based on PDA and dynamical application repository server,” in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC’07)*, May 2007.

[47] L. Tao, H. Xu, and Z. Zhang, “Distributed inspecting and control system for motor vehicle safety performance,” in *Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC’09)*, pp. 384–387, Zhejiang, China, August 2009.

[48] H. R. Tseng, R. H. Jan, and W. Yang, “An improved dynamic user authentication scheme for wireless sensor networks,” in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM’07)*, pp. 986–990, Washington, DC, USA, November 2007.

[49] D. Panda, “An Introduction to Service-Oriented Architecture from a Java Developer Perspective,” http://www.onjava.com/pub/a/onjava/2005/01/26/soa-intro.html.

[50] N. Gruschka and N. Luttenberger, “Protecting web services from DoS attacks by SOAP message validation,” in *Proceedings of the IFIP TC1121 International Information Security Conference (SEC’06)*, May 2006.

[51] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys’04)*, pp. 188–200, November 2004.