Authentication Issues for Sensors in IoT Solutions

CRISTIAN TOMA, CRISTIAN CIUREA, ION IVAN
Department of Economic Informatics and Cybernetics
Bucharest University of Economic Studies
Piata Romana 6, Bucharest
ROMANIA
E-mails: cristian.toma@ie.ase.ro, cristian.ciurea@ie.ase.ro, ionivan@ase.ro
Website: http://www.dice.ase.ro

Abstract: In first section the IoT – Internet of Things ecosystem is presented with features and particularities. Second section presents the technical terminology and how inter-domain technologies such as: Internet/Semantic and Middleware, RFID/NFC and Smart Objects-embedded devices are linked together. The paper shows an implementation of the authentication procedure with a proximity tag/card implemented in Java SE, in the third section. The conclusions are presented in the fourth section and they present the opportunity to develop a proof of concept project, which may have multiple implementations.

Key-Words: Sensors Authentication, IoT (Internet of Things), Supply Chain Management Security

1 IoT (Internet of Things) Ecosystem Intro

The concept of Internet of Things (IoT) is related to uniquely identifiable objects and their virtual representations in a structure similar with the Internet. This new concept is an innovative solution to realize a quantitative analysis of all the things that surround us. A prerequisite needed for the Internet of Things is the radio-frequency identification (RFID). If all objects and people in real life were equipped with identifiers and smart-tags, they could be managed and inventoried by computers. [1]
People have limited time, attention and accuracy, so that they are not very good at capturing and storing information about all the things from the real world (even if we include 2D barcodes used for Automatic Data Acquisition applications). An alternative view on IoT, from the Semantic Web perspective, focuses instead on making all things addressable by the existing naming protocols, such as URI (this refer to other things than those electronic, smart, or RFID-enabled). The objects themselves can be referred for the moment by other agents, for example by powerful centralized servers acting for their human owners, without conversion.
In [2] is considered that the Internet of Things is the network of physical objects that contain integrated technology to communicate and sense or to interact with their internal states or the external environment.
Figure 1 below present the concept of Internet of Things and the connection between all involved components:

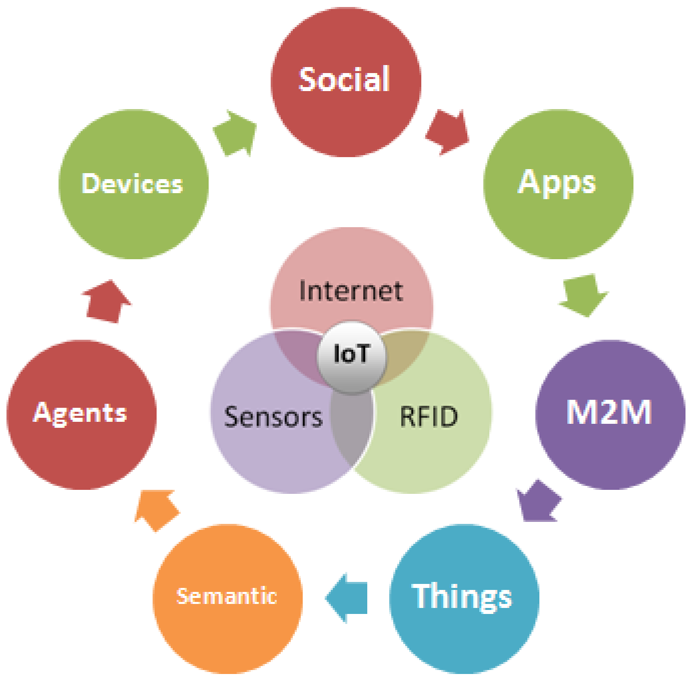


Figure 1. Internet of Things components

For the development of all the applications of the Internet of Things is compulsory to develop a range of technologies and solutions, in which free and open software plays an important role, providing both hardware architecture and open source software, such as development environments, that allows the development of open source applications for the Internet of Things.

2. IoT Technical Details

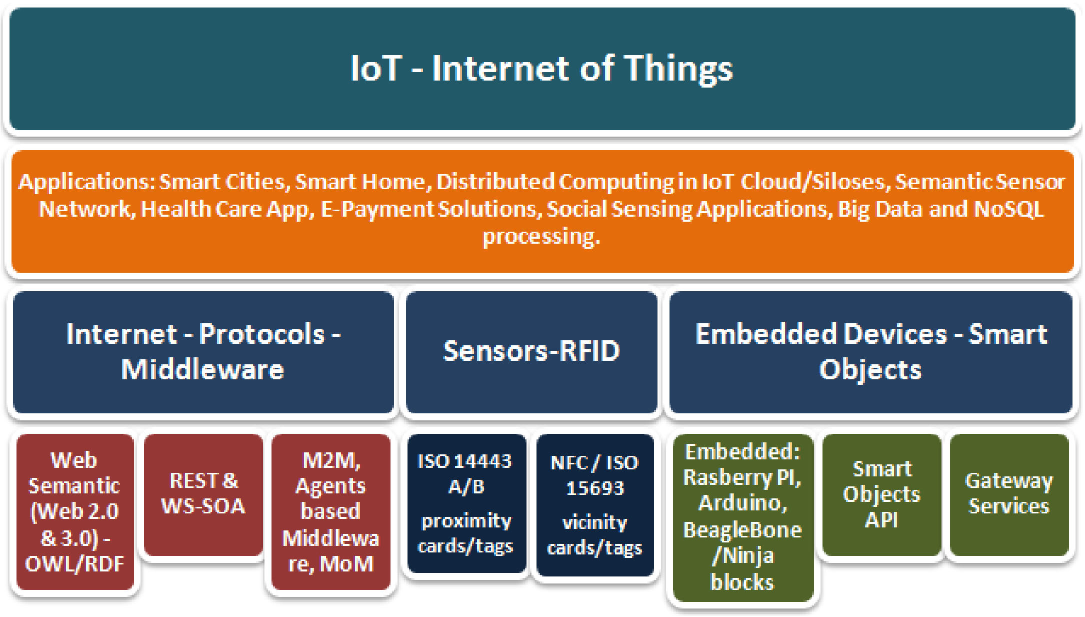
The IoT is composed of many ***sensors***, application platforms, user platforms, and so on, creating a community ecosystem. A sensor is a component that collects and delivers information about things in a specified area. By building a shared common platform such as IoT will create a richer ecosystem for all the people, by enabling the development of innovative systems to focus on the value-added of the human physical touch points of connected objects and the services that manage these objects [3].

The Internet of Things means to connect in real-time people and objects from the real physical world together in a network of sensing, reasoning, and action. The IoT connects people and things together with software products and applications.

A ***Smart Object*** contains IoT data and information, also metadata and software agent code resources, such as application software event handler. Samples of Smart Objects are embedded devices such as Rasberry Pi, Arduino, BeagleBone/Ninja Blocks. The ***data model*** for broad interoperability is represented by the Smart Object API.

The Smart Object API is represented by a Semantic Web application for the Internet of Things that provides linked-data interactions

between application software agents and IoT endpoints, sensors and user devices, which are pluggable in real-time [3].
The IoT vendor ***silos*** provide high level cooked APIs from cloud services of Smart Object API, enabling integration of IoT resources from top to bottom of the stack. The IoT of today means no interoperability and existence of many vertical and horizontal silos.
The Smart Object API supports the concept of a Smart Gateway, which is working as a Smart Object intermediate for devices on the network, adding semantic descriptors and offering a service interface for the device representation on the Internet. Sensors and gateways must be programmed for each service they need to interact with.
IoT applications consist of sensors and actuators end points, user device end points, and application software that connect the endpoints in the representation of a directed graph. We can also build a graph of resources consisting of Smart Objects connected to services and other Smart Objects, based on the related IoT ontologies, and the Smart Object API. [3]
The interoperability and interdependence between multiple devices is becoming a common characteristic because people are trying to build their own Internet of Things by getting all their smart devices to be connected in the cloud. The IoT needs a standard to interact with other devices and this standard must enable software for easier interaction.
The actual Internet of Things consists of many different sensor networks and protocols, connected to special cloud services and offering access through smart mobile devices and browser applications. It is unusual for these separate ***silos*** to cooperate or interact one with each other. [4]


IoT - Internet of Things
Applications: Smart Cities, Smart Home, Distributed Computing in IoT Cloud/Siloses, Semantic Sensor Network, Health Care App, E-Payment Solutions, Social Sensing Applications, Big Data and NoSQL processing.
Internet - Protocols - Middleware
Sensors-RFID
Embedded Devices - Smart Objects
Web Semantic (Web 2.0 & 3.0) - OWL/RDF
REST & WS-SOA
M2M, Agents based Middleware, MoM
ISO 14443 A/B proximity cards/tags
NFC / ISO 15693 vicinity cards/tags
Embedded: Rasberry PI, Arduino, BeagleBone /Ninja blocks
Smart Objects API
Gateway Services
Figure 2. Internet of Things tech items

The Internet of Things has many applicability areas and fields, such as the healthcare sector, the retail sector, transportation services, educational services, industry and so on.

3 Smart Tags/Cards for Things Authentication and Data Integrity

In a retail or supply chain management integrated solutions, RFID – vicinity/proximity tags or cards will be attached to the products and ***things,*** in order to store data and meta-data about the products. The ***sensors*** will read or modify the data stored in RFID labels attached to the things. In order to provide authentication and data integrity, the solution will have an authentication procedure that requires minimum of processing and power consumption, but in the same time it will be strong enough to avoid security pitfalls.
The minimum authentication model will require a mechanism similar with CRYPTO1 from Mifare 1K/4K proximity cards and a higher secure model would be inspired from Sony FeliCa cards and VISA/Mastercard DDA (dynamic authentication) procedure from banking cards (MULTOS or Java Card).
Mifare DESFire tags/cards have only contactless interface for communications and they are fully compliant to the ISO/IEC14443A (1-4). They have 7 bytes UID ("Double Size UID") and from point of view of CPU & OS, there are the following features:

- Asynchronous CPU core
- (3) DES coprocessor
- Fixed Command Set
- No Customer ROM codes

The Mifare DESFire file-system is able to handle:

- up to 28 application / card
- up to 16 files / application
- up to 14 keys / application
- 1 master-key for card maintenance
- Plain, (3)DES encrypted, or MAC-ed data transmission
- On-Chip Backup management

This section presents an authentication procedure specific to Mifare DESFire tags/cards that have certain particularities and requirements. An improved procedure should be sufficient for RFID tags/cards authentication and it is presented as Java source code statements for non-sensitive operations (the complete source code may be obtained from the authors):

```
public static boolean
authDesFireEV1() throws
Exception
{
      boolean authResult =
false;
      byte[] byteArrayPICCCh =
new byte[authByteArrayLen];

      try
      {
            byte[] ret =
null;//store the binary
responses from the cards

            // Get PICC
Challenge: /send 900A0000010000
            byte[] APDUCommand =
new byte[]
            {(byte) 0x90, (byte)
0x0A, 0x00, 0x00, 0x01, 0x00,
0x00};
            ret = term.send(0,
APDUCommand, 0,
APDUCommand.length);

      //1. PCD-Host < PICC-Card
Challenge Seed
      System.arraycopy(ret, 0,
byteArrayPICCCh, 0, ret.length-
2);
      System.out.println("1.
PCD-Host < PICC-Card Challenge =
"+JCInfo.toHex(byteArrayPICCCh))
;

//2.1. Calculate d(PICC) = DES-
CBC(IV=0x0, Key=0x0, PICC
Challenge Seed)
                   byte[]
desKeyMaterial = new
byte[desKeyLen];
                   //byte[] iv
={0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00};
                   byte[] iv =
new byte[desKeyLen];
     // …
     //Create Cipher -
DES/CBC/PKCS5Padding for SunJCE
     // …

     Cipher desCipher =
Cipher.getInstance("DES/CBC/NoPa
dding");


desCipher.init(Cipher.DECRYPT_MO
DE, desKey, ips);
     byte[]
decByteArrayPICCChalange =
desCipher.doFinal(byteArrayPICCC
h);
     System.out.println("2.1
PCD-Host Calculus - d(PICC) =
");

     byte[] pcdRandomSeed = new
byte[authByteArrayLen];
     Random r = new Random();
     r.nextBytes(pcdRandomSeed);
     System.out.println("2.1
PCD-Host Calculus - PCDSeed =
");

     //2.2. Calculate PCDResp =
e(PCDSeed + ROL8(d(PICC seed)))
     byte[]
pcdRandomSeedChallenge4PICC =
new byte[2 * authByteArrayLen];
     // …

desCipher.init(Cipher.ENCRYPT_MO
DE, desKey, ips);
     byte[]
encPcdRandomSeedChallenge4PICC =
desCipher.doFinal(pcdRandomSeedC
hallenge4PICC);
     System.out.println("2.2
PCD-Host Calculus - PCDSeed +
rol8(d(PICC seed)) = ");
     //…
        }
        catch (Exception ex)
        {

      ex.printStackTrace();
            throw new
Exception(ex.getMessage());
        }
        return authResult;
    }
```

The data and meta-data for products/things might be stored in files inside the file system memory layout and they will be encrypted with the session key negotiated during the authentication procedure.

4 Conclusions

In 2020 there is estimation that will be 50 billion IoT devices in the market. At least all the consumers of “Java enabled” and Embedded Linux smart objects will be in the main target group for open source solutions. The smart objects are processing the data collected from sensors and for instance, from RFID tags/cards. The authentication process is an important approach taking into account NFC and RFID domains expansion.
Besides the authentication process of RFID tags/cards by the IoT sensors, the paper represents the basic know-how for developing a proof of concept that will demonstrate how the RFID/NFC tags/cards are written with meta-data information, and then periodically tags/cards area read by RFID/NFC reader/writer devices (sensors), in order to do the things tracking. The obtained information is securely and collaborative aggregated into a unified data model and processed using distributed computing methodologies over “big data”.

The proof of concept project results may include:

1. The formal models, architecture, REST/Web-services/communications protocols and M2M data-structures;
2. The beta version of the software libraries that implement products/services tracking and clustering:

- sensor control using devices SDKs; the reading/writing of the data formats from/in RFID vicinity/proximity tags/cards and optionally, reading 2D barcodes of the products/services;
- “big data” processing and semantic parsing, via distributed computing model and implementation, using embedded devices/boards (Internet of Things ‘smart objects’, e.g. Raspberry-PI board) for cloud micro-instance deployment and standard PCs/laptops;
- secure communications from sensors to ‘smart-objects’ via IoT Service Gateways.

During the development of the proof of concept project, we estimate that we will create pre-requisites to offer Java implementation for Smart Object API, to enhance the existing security and communications protocols for REST Interface/Web-services, and improvement of M2M/IoT data models plus value added services for existing and new IoT deployed “silos”-es.

References:

[1] Wikipedia, *Internet of Things*, Available at: http://en.wikipedia.org/wiki/Internet_of_Things

[2] *Open Smart Cities I: Open Source Internet of Things*, Available at: http://observatorio.cenatic.es/index.php?option=com_content&view=article&id=807:open-smart-cities-i-open-internet-of-things&catid=94:tecnologia&Itemid=137

[3] Michael Koster, *Data models for the Internet of Things*, Available at: http://iot-datamodels.blogspot.ro/

[4] Tom Vu, *The Internet of Things: Inspiration and Requirements*, Available at: http://blog.makezine.com/2013/04/18/the-internet-of-things-inspiration-and-requirements/

[5] C. Aggarwal, N. Ashish, and A. Sheth, *The Internet of Things: A Survey from The Data-Centric Perspective*, "*Managing and Mining Sensor Data*", Springer, 2013, ISBN 978-1-4614-6309-2.

[6] Lu Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning, *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems (Wireless Networks and Mobile Communications)*, Auerbach Publications, 2008, ISBN 978-1420052817, 336 pg.

[7] Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds.), *Architecting the Internet of Things*, Springer, 2011, ISBN 978-3-642-19157-2, 351 pg.

[8] Charalampos Doukas, *Building Internet of Things with the Arduino (Volume 1)*, CreateSpace Independent Publishing Platform, 2012, ISBN 978-1470023430, 352 pg.

[9] Adrian McEwen, Hakim Cassimally, *Designing the Internet of Things*, John Wiley & Sons, 2013, ISBN 978-1118430620, 260 pg.