Association for Computing Machinery
Advancing Computing as a Science & Profession

Ubiquity, an ACM publication
January 2016

Ubiquity Symposium

The Internet of Things

Ensuring Trust and Security in the Industrial IoT

By Bernardo A. Huberman

Editors' Introduction

Industrial Internet of Things (IOT) is a distributed network of smart sensors that enables precise control and monitoring of complex processes over arbitrary distances. The great advantage of the industrial IoT is counterbalanced by a security weakness. The insertion of a smart device capable of extracting protected data or malicious actions can infect the whole network with relative ease. Thus it becomes imperative to discover whether or not new devices have the right capabilities and compatibilities with other sensors. This article presents a zero knowledge protocol that achieves precisely that objective while keeping the sensor data private.

Ubiquity Symposium

The Internet of Things

Ensuring Trust and Security in the Industrial IoT

By Bernardo A. Huberman

We are awash with sensors and devices with more processing power than many of the standalone computers that reside on our desks. Smart phones, micro-computers, ambient light control systems, thermostats that can adjust to and report minute changes in our daily life, and the ubiquitous fitness gadgets constitute a whole technological species that is starting to coexist with us through the same Internet environment we populate with our communication devices.

And this is the simple side of what has now become fashionable to call the "Internet of Things" (IoT). The real revolution is taking place in a different setting, an industrial one, where in each industry—from manufacturing to refineries and transportation—myriad smart sensors are connected through shared API's. This new form of networked computing power—the so called "Industrial Internet of Things"—will likely dwarf what we conceive of as the present day Internet.

The industrial IoT has many characteristics that make it different from the consumer smart devices that most people are familiar with. First, the pervasiveness and interconnectivity of smart sensors, coupled with the unpredictability of their inputs, make response times autonomous of human intervention. Whereas a fitness tracker running out of power does not necessitate an urgent response, the failure or delayed emergency signal from a smart sensor controlling several valves in a refinery can trigger an undesirable reaction chain from other sensors and actuators leading to overall system failures.

Second, the Industrial IoT has all the characteristics of an open, distributed system [1, 2] dealing as it does with a large, diverse quantity of information while exhibiting massive concurrency. It is also asynchronous, since the behavior of the environment is not necessarily predictable by the system itself, which leads to the need for autonomous reactions. This points to a necessarily decentralized system, since it would be hard for a central unit to have up-to-date information on the state of the whole system.

Third, the distributed nature of the Industrial IoT makes it open to a host of security threats, since a single break into a component of the distributed fabric can compromise the entire the system [4].

Since the behavior of such open systems has been analyzed in quite detail, in this contribution about the IoT I will focus on the last point—the security aspects of the Industrial IoT. Specifically, I will describe a mechanism that we invented to deal with a pervasive problem with smart sensor networks: Discovering whether or not new devices have the right capabilities and compatibilities with other sensors, while keeping their data private [3]. While there are some existing proposals to address the security problems in IoT, most of them do not offer mechanisms for protecting the privacy of device capabilities or their characteristics. Recently, a group at IBM proposed the use of block chains—the very basis of Bitcoin—for all devices in the world of IoT. While in principle it could offer a robust solution to the security problem, it is unrealistic to imagine the implementation of a blockchain for all devices on a global scale in a way that allows the blockchain to scale as the size of the device network increases.

Since the mechanism that I will describe for solving security and trust issues in the Industrial IoT involves zero knowledge cryptographic techniques, I will first provide a lightning survey of such techniques, to be followed by a simple exposition of the mechanism that solves these problems. The interested reader can consult the reference to our original patent and a paper [5] for a more detailed explanation of what is involved in setting up these mechanisms.

A Secure Protocol

The mechanism exploits the use of two fundamental cryptographic primitives: hash functions and public key systems.

In general, cryptographic functions operate on inputs such as ``messages'' and ``keys,” and produce outputs such as ``cipher texts'' and ``signatures.” It is common to treat all of these inputs and outputs as large integers according to some standardized encoding. Throughout this exposition I assume any value involved in a cryptographic function is a large integer, no matter what it may be called.

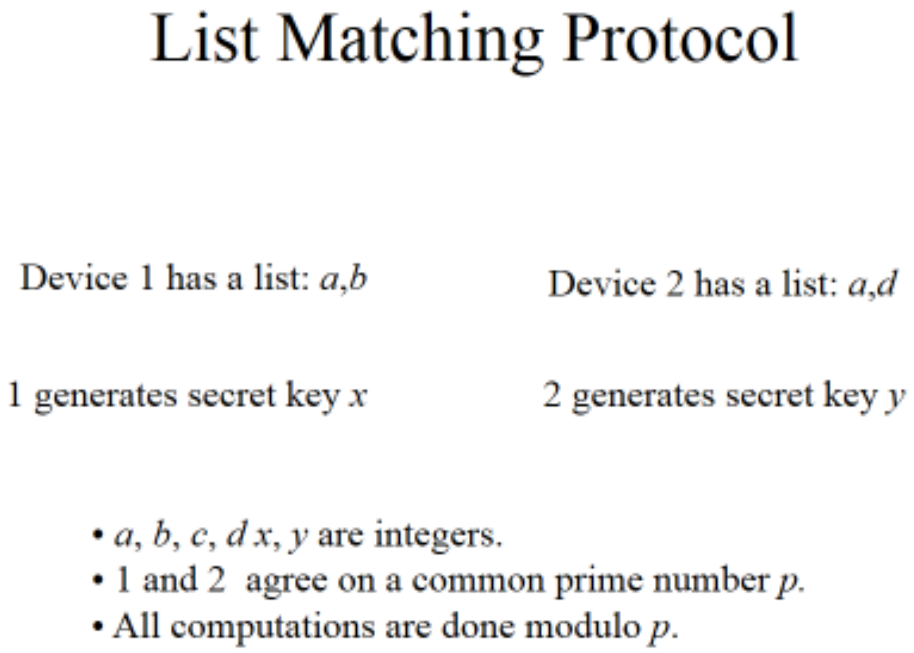
A cryptographic hash function, ***H***, is a mathematical transformation that takes a message *m* of any length, and computes from it a short fixed-length message, which we will call ***H(m)***. This fixed length output has the important property that there is no way to find what message produced it short of trying all possible messages by trial and error. Equally important, even though there may exist many messages that hash to the same value, it is computationally infeasible to find even two values that ``collide.” This practically guarantees the hash of a message can ``represent” the message in a way that makes it very difficult to cheat. An even stronger property we will require is the output of a cryptographic hash function cannot be easily influenced or predicted ahead of time. Thus someone who wanted to find a hash with a particular pattern (beginning with a particular prefix, say) could do no better than trial and error. In practice, hash functions such as MD-5 and SHA (secure hash algorithm) are often assumed to have these properties.

Public key encryption relies on a pair of related keys, one secret and one public, associated with each individual participating in a communication. The secret key is needed to decrypt (or sign), while only the public key is needed to encrypt a message (or verify a signature). A public key is generated by those wishing to receive encrypted messages, and broadcasted so the message’s sender can use the key to encode the message. The recipient of this message then uses their own private key in combination with their public key to decrypt the message. While slower than secret key cryptography, public key systems are preferable when dealing with networks of devices that need to be reconfigured fairly often. Popular public key systems are based on the properties of modular arithmetic.

Now onto the mechanism that removes the disincentive inherent in having to disclose the private content of the data and capabilities of a new sensor, which is inserted into an industrial network. Conversely, we don’t want the device to learn any of the data and capabilities of the installed base of sensors in the enterprise. And yet we want it to be able to interact with those devices that share some similar capabilities and contain signatures that certify them as belonging to the network.

For the sake of clarity I'll describe such a mechanism in pictorial fashion. Consider two devices, device 1 and device 2, each of which has a list of attributes. We want to find out if any of those attributes (it could be ID numbers, capabilities, memory, etc.) are common to both of them. I’ll assume the lists contain two items each, device 1 with items *a* and *b* and device 2 with items *a* and *d.*

The procedure for discovering if any of the elements in the lists are common to both devices without revealing works as follows:



List Matching Protocol

Device 1 has a list: *a*,*b*

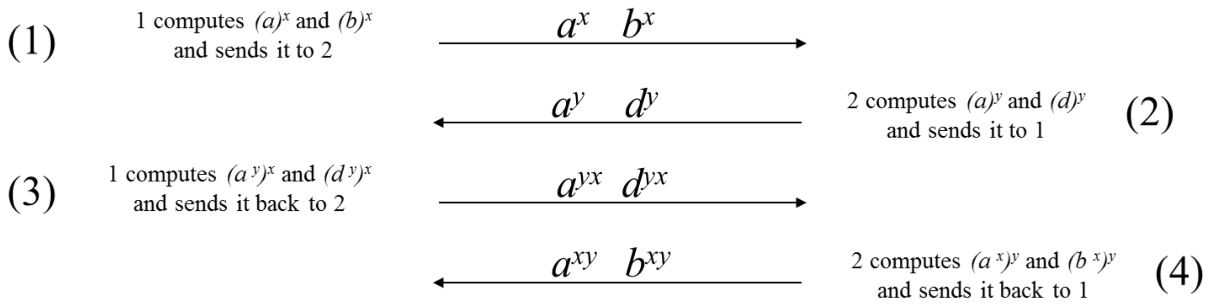
Device 2 has a list: *a*,*d*

1 generates secret key x

2 generates secret key y

- a, b, c, d x, y are integers.
- 1 and 2 agree on a common prime number p.
- All computations are done modulo p.

The procedure works as follows:


(1)
1 computes $(a)^x$ and $(b)^x$ and sends it to 2
$a^x \quad b^x$
$a^y \quad d^y$
2 computes $(a)^y$ and $(d)^y$ and sends it to 1
(2)
(3)
1 computes $(a^y)^x$ and $(d^y)^x$ and sends it back to 2
$a^{yx} \quad d^{yx}$
$a^{xy} \quad b^{xy}$
2 computes $(a^x)^y$ and $(b^x)^y$ and sends it back to 1
(4)

Since $a^{xy} = a^{yx}$ both devices know they both have a but can not decode the other elements.

Notice the entire security of the operation would be compromised if device 1 or device 2 were able to compute either *x* or *y* from the data they initially sent to each other. But that is almost impossible because of the intractability of the discrete logarithm problem: Given integers *a* and *b* and prime *p*, it is computationally hard to find and integer ***n*** such that

$$b^n = a \ (mod \ p)$$

This method, which I illustrated using only two inputs from both devices, generalizes to any large set of data and thus allows for either device to find whether or not they have a set in common without revealing what the data.

Conclusion

The ease of communication among devices and the ensuing exchanges of data mediated by the Internet have raised interesting problems concerning both the security of the data exchanged, and the need to keep it private in the context of the Industrial IoT. In particular given the ease with which smart sensors can be inserted into an industrial setting raises the issue of its certification and ability to interact with other sensors in the network in a trusted fashion without revealing the content of its data or capabilities. While the standard answer would resort to a trusted third party or the implementation of a blockchain, or to desist in having the data exposed to manipulations that could actually reveal its nature or the identity of the target.

As I have shown, it is possible to use zero knowledge techniques to solve this problems in ways that ensure privacy without having to resort to trusted third parties. Moreover, this mechanism can be implemented and deployed on most smart sensors.

The adoption of these techniques will accelerate the adoption of a distributed network of smart sensors and machines in the service of enterprise, thus leading to an Industrial IoT that will coexist with the one we are all familiar with as of today.

References

[1] Huberman, B. A. *The Ecology of Computation*. MIT Press, Cambridge, 1999.

[2] Hewitt, C. Offices are open systems. *ACM Transactions on Information Systems (TOIS)* 4, 3 1986, 271-287.

[3] Huberman, B., Sorkin, S., and Tyler, J. US Patent 8,316,234, Encode Attribute Matching on Communication Devices. 2012.

[4] ADEPT: An IoT Practitioner Perspective. IBM. 2015

[5] Huberman, B. A., t Franklin, M., and Hogg, T. Enhancing privacy and trust in electronic communities. In *Proceedings of the First ACM Conference on Electronic Commerce*. ACM Press, New York, 1999, 78-80.

About the Author

Bernardo A. Huberman is a Senior HP Fellow and Director of the Mechanisms and Design Lab of HP Labs.

DOI: 10.1145/2822883