Blockchain IoT (BIoT): A New Direction for Solving Internet of Things Security and Trust Issues.

Invited Talk

Pascal Urien
Telecom ParisTech, Saclay University, LTCI
23 avenue d'Italie, 75013, Paris, France
Pascal.Urien@Telecom-Paristech.fr

***Abstract*— The Blockchain is an emerging paradigm that could solve security and trust issues for Internet of Things (IoT) platforms. We recently introduced in an IETF draft ("Blockchain Transaction Protocol for Constraint Nodes") the BIoT paradigm, whose main idea is to insert sensor data in blockchain transactions. Because objects are not logically connected to blockchain platforms, controller entities forward all information needed for transaction forgery. Never less in order to generate cryptographic signatures, object needs some trusted computing resources. In previous papers we proposed the Four-Quater Architecture integrating general purpose unit (GPU), radio SoC, sensors/actuators and secure elements including TLS/DTLS stacks. These secure microcontrollers also manage crypto libraries required for blockchain operation. The BIoT concept has four main benefits: publication/duplication of sensors data in public and distributed ledgers, time stamping by the blockchain infrastructure, data authentication, and non repudiation.**

Keywords— Blockchain; Internet of Things; security;;

I. INTRODUCTION

According to [1], the Internet of Things is a network of connected things. A thing is made with a CPU, memories (RAM, ROM, EEPROM, FLASH…), and IO buses. It comprises at least one network interface such as Wi-Fi, Bluetooth or ZigBee, and is equipped with sensors and actuators. According to [2] "*a short list of requirements includes tamper resistance and secure communications and storage*". In order to achieve a secure and trusted IoT framework, we suggest that the objects should support three basic features:

- Secure Communication, i.e. strong mutual authentication between end entities, privacy, and integrity for exchanged information;
- Secure Storage, required for secret values used by communication and enforced by tamper resistant devices such as secure elements;
- Node Integrity, which requires secure updates and secure boot. Physical technologies dealing with multi processors or logical technologies such as sandbox may effectively contribute to increase resistance to intrusion.

II. IOT FRAMEWORKS & SECURITY

Multiple IoT frameworks [3] are specified by multiple organizations, with different security features. Here is a non exhaustive list:

- Thread, supported by NEXT boards is based on 6LOWPAN, deals with DTLS and password, and realizes *Commissioner-Joiner* architecture.
- The Open Connectivity Foundation (OCF) defines a framework that uses 6LOWPAN, DTLS with authentication features, access control list (ACL), REST APIs. It is for example supported by the Iotivity framework.
- The MBED stack from the ARM company is supported by an IBM kit, and comprises multiple protocols such as IPv4, 6LOWPAN, TLS/DTLS, HTTP, CoAP, MQTT, LWM2M.
- The HAP ("*HomeKit Accessory Protocol*") from Apple implements security features at application level. It deals with Bluetooth, Wi-Fi, HTTP, JSON, and the *Secure Remote Password* procedure (SRP, RFC 5054).
- Brillo and Weave are procvided by Google and run over the Intel® Edison board. Brillo is a 35MB footprint OS. Weave is a communication platform, which supports IEEE 802.15.4 (Zigbee, Thread), BLE, Wi-Fi, Ethernet and HTTPS. Application framework uses JSON schemas and XMPP**.** Authentication is performed thanks to *Google Authentication Server* (AS) according to the OAuth 2.0 standard.
- Philips Hue Bulbs are following the *Zigbee Light Link* (ZLL) specification. A same link key is shared by all nodes. A bridge manages bulbs, whose commands are transported over IP/UDP interface in clear form.

In summary security features may be enforced by multiple layers such as:

- Operating System in charge of software integrity and secure updates;
- MAC security features, for example IEEE 802.11i for Wi-Fi, pairing for Bluetooth, unique master key & shared link key for Zibgee, AES Key for Lora clients, HMAC client key for SigFox;
- TLS/DTLS stacks, used as identity modules and dealing with symmetric or asymmetric credentials;
- Application features, performing authentication procedures and enforcing messages privacy and integrity.

III. Trust For IoT

The papers [4][5] demonstrate an attack against smart bulbs able to download embedded software over the air. This attack performs software update first by exploiting a Zigbee implementation bug, and second by the recovery via a side channel attack ("*Correlation Power Analysis*"), of the AES key used for update procedures, which is shared by a community of bulbs.

Because TLS/DTLS stacks are often used in IoT frameworks as identity layer, and because we believe that tamper resistance is needed in order to avoid object hijacking, we suggest using secure elements as identity modules running TLS/DTLS stack and acting as communication firewall [3][6].

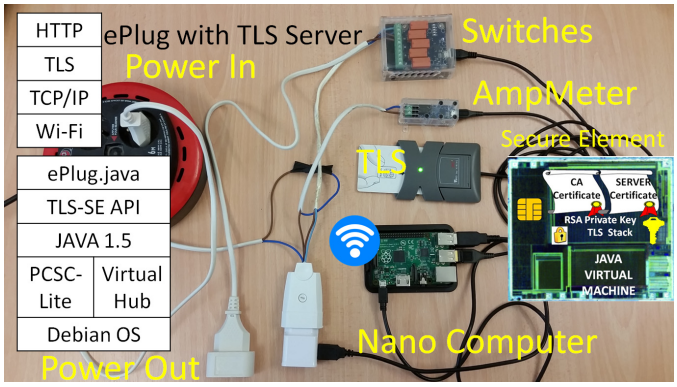

Fig. 1. An ePlug according to the drawbridge concept

In [7] we demonstrated an ePlug (see figure 1) comprising a Raspberry Pi, with a Wi-Fi interface, a switch actuator, a current sensor, and a secure element embedding a TLS stack. According to the *so called* drawbridge concept, a unique 443 TCP port is opened on the device. All application messages are protected by the TLS record layer, TLS sessions are established after a mutual authentication involving certificates and private keys both for server and client. Raspberry Pi uses a single SD memory for OS storage (about 8GB), what can give some insurance about integrity. Nerveless the operating system complexity is a possible security issue, since it may comprise malicious backdoors.

In [8] we developed the idea of the four quarter architecture (see figure 2), and we demonstrated the concept for a smart thermometer. According to this model an object is built over four physical and logical layers.

- A General Purpose Unit (GPU) that manages the object. It is the main processor that runs either a tiny operating system or just the needed software. As an illustration the ATMega2560 chip manages 256KB of FLASH and 8KB of SRAM. Update operations may (and should) be secured by cryptographic means.
- A radio SoC that includes processor with RAM and external non volatile memory, for example 36KB of SRAM associated to an external 4 MB SPI FLASH. It is also in charge of the MAC security, usually dealing with asymmetric cryptography. It supports procedures for software updates.
- An identity module, i.e. a secure element (javacard) running a TLS stack. These tamper resistant components embed a few KB of FLASH memory and 10 KB of SRAM at the most. The TLS stack is written according to the javacard langage, and enforces an application firewall between the radio SoC and the GPU.
- Sensors and actuators are driven by the main processor according to REST messages transported by radio packets. As we previously mentioned, authorization mechanisms are optionally supported.

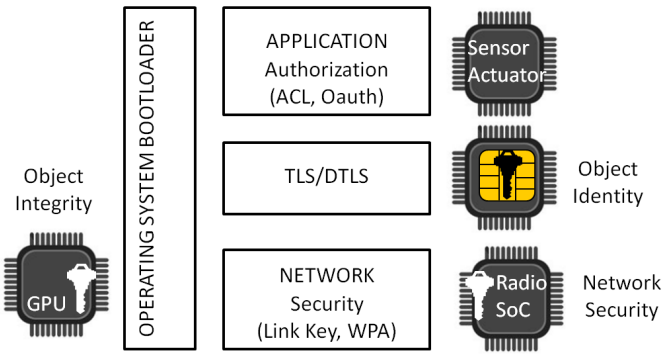

Fig. 2. The Four Quatter Archictecture

The four quarter architecture can secure over the air software updates both for GPU and radio SoC. Never less it is not able to avoid malicious updates performed by hackers who have physical access to devices. This feature implies the availability of secure download operations, typically associated to symmetric secret key stored and used in electronics chips.

IV. Blockchain IoT (BIoT)

The Bitcoin crypto currency was introduced in 2008 [9], in a famous paper written by the anonymous author *Satoshi Nakamoto*, who also released the win32 software *Bitcoin.exe* [10], about 16,000 lines of C++ code, and 6 MB binary size. The first blockchain platform was a set of computers running this software, and exchanging information thanks to a peer to peer scheme over the internet network.

A blockchain infrastructure collects transactions that are signed by private keys, more precisely by the ECDSA procedure dealing with elliptic curves. Most of blockchain

work with the secp256k1 curve, which uses 32 bytes random number as private keys. The blockchain address is computed from the public key.

The main blockchain features are the following:

- A consensus algorithm is applied in order to build blocks; involved entities (typically the miners) check the transactions coherence and their authenticity.
- There is a remunerated competition for the block certification and the computing of a new branch, which is based on algorithms such as Proof of Work (PoW) or Proof of Stake (PoS).
- There is no third trusted party; each participant generates his keys.
- All the ledger components, i.e. transactions and blocks are stored in widely distributed databases.

We distinguish four main services offered by blockchain platforms:

- The transfer of crypto money from a blockchain address to another:
- The execution of programs;
- The storage and timestamp of data, charged with crypto money;
- The use of multi-signatures, i.e. in blockchain such as Bitcoin a transaction may be signed by several entities.

Let's take a look at the OP_RETURN instruction available for Bitcoin transaction scripts. It does nothing BUT it stores up to 80 bytes. According to the *opreturn.org* website (in June 2018), about 50% of Bitcoin transactions include an OP_RETURN instruction used for multiple purposes and services. OP_RETURN protocols are usually identified by the three first bytes of the associated payload.

As an illustration the transaction:

https://blockchain.info/fr/tx/01ee70227b26d14c3f33e1283cd66a6b7d6e8d8cb97b9cc77dd913d90f779cf1

(whose right part is the transaction identifier) includes an OP_RETURN instruction whose content is the URL:

http://www.enst.fr/~urien/Base58(SHA256(file)).jpg

in which the right part is the encoding in the base 58 of the hash value of a file. Because a certificate is basically a signed hash, it is a way to compute a document certificate without a third trusted party. The transaction fee is computed from the transaction size, about 200 SATOSHIs per byte.

In today IoT infrastructures, sensor data are stored in the cloud. Usually they are not authenticated, and non repudiation services are not available.

In the IETF draft [12] we suggest to encapsulate sensor data in blockchain transactions. The expected benefits are the following:

- Publication & Duplication;
- Authentication;
- Timestamp.

The functional architecture (see figure 4) comprises objects storing blockchain private keys and identified by blockchain addresses, and controllers with full internet connectivity. Controllers have access to ledger databases and therefore may collect contexts needed by blockchain transactions Objects may include identity modules built over secure elements, are therefore can securely store private keys. Furthermore X509 certificates may be associated to these private keys and used for identity purposes.

Let's consider an Ethereum transaction (see figure 3). It comprises the following fields:

- The recipient's address of the message;
- A nonce, an integer incremented for every transaction;
- A value, the number of WEIs to be transferred to the message recipient;
- A gasLimit representing the maximum number of executed EVM (*Ethereum Virtual Machine*) instructions;
- A gasPrice representing the price to be paid per instruction;
- An optional data field, either general information or EVM code or smart contract call;
- The ECDSA signature, used to authenticate the sender.

```
F8 74 // RLP List, length= 116 bytes
   0C // nonce 1 byte =12 decimal
   85 06FC23AC00 // gasPrice = 30 GWei
   83 013880     // gasLimit = 80000 gas
   // recipient address 20 bytes
   94 6BAC1B75185D9051AF740AB909F81C71BBB221A6
   80 // Null Ether Value
   // Data 15 bytes "Temperature=25C"
   8F 54656D706572617475726 53D323543
   1B // recovery parameter, 1 byte(27=+, 28=-)
   A0 // r, 32 bytes, ECDSA r parameter
      A9B58980F76EE6284800B82A2B5DF13E
      456887EC0CF426A5E5D6A738EB1784ED
   A0 // s, 32 bytes, ECDSA s parameter
      629633C6A3ED5FEE0FB40E2D1CF25134
      5B885D372857B1A6C4762C9BE914281F
```

Fig. 3. Example of an Etherreum transaction used for IoT purpose.

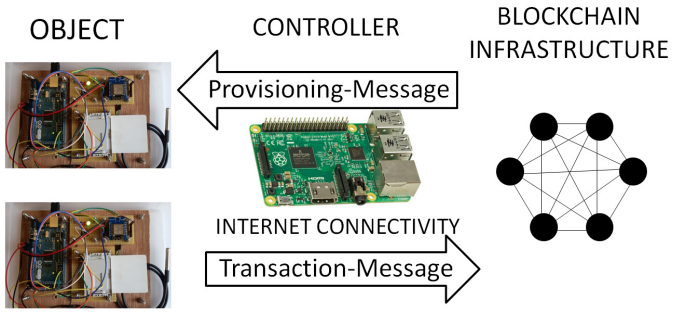

Fig. 4. Illustration of BIoT Messages.

The transaction is encoded according to the RLP ("Recursive Length Prefix") format. Figure 3 shows an Ethereum transaction, whose data field includes a temperature data generated by a sensor. The transaction can be retrieved at the following URL:

https://etherscan.io/tx/0xd6904d832462ae17718c69e9caa0c3f3bed458382ac1f4e43b1aadd8e94744ad (whose right part is the transaction identifier)

According to the Ethereum Yellow paper [11] the transaction fee is computed according to the relation:

gasPrice = 21000 + data_size_in_bytes x price_per_byte

with price_per_byte equal to 68 for a non null byte and 4 for a null byte. The data size of the transaction being equal to 15, the fee is therefore 22020 WEIs.

Because an object is not aware of the blockchain context the controller forwards the parameters needed for a transaction in the Provisioning-Message (see figure 4). This message is encoded according to the JSON syntax and comprises the following attributes:

{"type": 1, "nonce": 12, "gasPrice": 30, "gasLimit": 80000, "address": "6B…A6", "value": 0 }

The object returns a signed transaction (see figure 3) inserted in the Transaction-Message encoded according to the JSON syntax that comprises the following fields:

{"type": 1, "transaction": "F874…1F"}

V. PERSPECTIVES

We believe that the Blockchain IoT (BIoT) paradigm could be an efficient solution for sensor data publication. An experimental platform (see figure 5) is currently under definition.

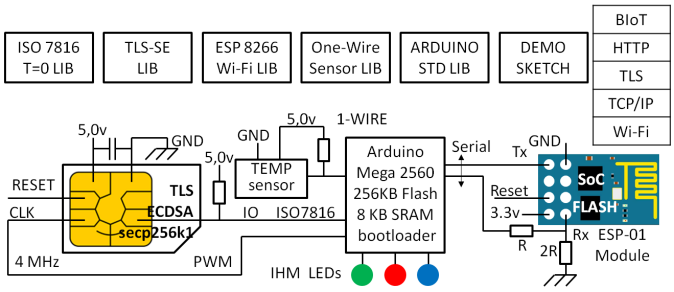

Fig. 5. Experimental BIoT platform under definition.

VI. REFERENCES

[1] .K. Pretz,"The Next Evolution of the Internet", theinstitute.ieee.org 2013

[2] "Rebooting the IT Revolution: A Call to Action" (SIA/SRC), 2015

[3] P. Urien, "Introducing TLS/DTLS Secure Access Modules for IoT frameworks: Concepts and experiments", IEEE ISCC 2017

[4] C. O'Flynn, "A Lightbulb Worm? Details of the Philips Hue Smart Lighting Design", Black Hat USA 2016

[5] E. Ronen, C. O'Flynn, A. Shamir, A. Weingarten, "IoT Goes Nuclear: Creating a ZigBee Chain Reaction", 2017 IEEE Symposium on Security and Privacy

[6] IETF draft, "TLS and DTLS Security Modules", draft-urien-uta-tls-dtls-security-module-04.txt, June 2017.

[7] P. Urien, "Securing The IoT With TLS/DTLS Server Stacks Embedded In Secure Elements: An ePlug Usecase", IEEE CCNC 2017.

[8] P.Urien, "An Innovative Security Architecture for Low Cost Low Power IoT Devices Based on Secure Elements", demonstration at the IEEE CCNC 2018 conference and at the CES2018 ComSoc Kiosk.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", www.bitcoin.org, 2008, [retrieved: June, 2017]

[10] P. Huang, "A Dissection of Bitcoin", ISBN 9781329754812, January 2016

[11] G. Wood, Ethereum Yellow Paper, "Ethereum : a Secure Decentralized Generalized Transaction Ledger", EIP-150, 2015, http://yellowpaper.io/, [retrieved: July, 2017]

[12] IETF draft, "Blockchain Transaction Protocol for Constraint Nodes", draft-urien-core-blockchain-transaction-protocol-00.txt, March 2018