© Hovsepyan V. Securing data transfer in IOT environment // Ukrainian Scientific Journal of Information Security, 2016, vol. 22, issue 2, p. 131-134.

SECURING DATA TRANSFER IN IOT ENVIRONMENT

Vladimir Hovsepyan

National Polytechnic University of Armenia, Armenia

HOVSEPYAN Vladimir

Date and place of birth: 1992, Yerevan, Armenia.
Education: National Polytechnic University of Armenia 2016.
Current position: PHD student (2nd year).
Research Interests: information security, cloud security, IoT devices.
Publications: more than 10 publications.
E-mail: vladimirhovsepyan@gmail.com

Abstract. *New method has been developed that securing data exchange process. Third party compute module used to secure data coming over different devices to the cloud environment. The module serves as middleware between 2 different endpoints and can handle several connections through wireless or wired connection. The OS adaptation are done in order to easy run custom software solution and adapt device to our goals. The encryption is done through symmetric algorithm, which was adapted to compute module in order to provide a high level speed and security. The two main improvements in algorithm are related to AES instruction usage and processing parallel computing. Corresponding software solution that will manage encryption and decryption process is also described. Software solution consist from 2 parts, where first part is preinstalled in compute module and second part need setup in cloud environment. These 2 parts work together as whole system and ensure data connection securing without requesting lot of changes in exist application environment.*

Key words*: cloud environment, encryption, IoT, AES, wireless connection.*

Introduction

Nowadays security in data communication has become very popular because of growing amount of IoT devices and cloud base applications. The ability to connect, communicate with, and remotely manage an incalculable number of networked, automated devices via the Internet is becoming pervasive, starting from the factory floor to the hospital operating room or to the residential basement. There are thousands application that are asking to provide personal information such as credit card numbers, health status, files and etc. To establish secure communication between client and server in most of cases TLS/SSL protocol used. This approach used by Facebook, Google, Dropbox, Github and many other services. TLS protocol is supported by major Operation Systems and it provides high level security. But there exist devices which can’t support this protocol because of their simplicity and sending information in open packages. Good example is health tracking sensors which have low performance and custom operation system which are not supported by TLS or any other secure protocol. Such problem can have any device which is constructed to do simple stuff and also has low performance to keep the cost lower. So how to protect deeply embedded endpoint devices that usually have a very specific, defined mission with limited resources available to accomplish it? Building such sensors in mind in order to support secure communications will raise their cost.

IoT Devices. The Internet of Things is growing steadily alongside the already well established smartphone, tablet, pc and consumer electronic markets. The Internet of Things can be defined as the connection of everyday objects and machines so that they work seamlessly together across modern networks [1]. The first steps for creation internet of things start in tech companies. Pundits have been discussing the idea for decades, and the first internet-connected toaster was unveiled at a conference in 1989. IoT is simply the network of interconnected things/devices which are embedded with sensors, software, network connectivity and necessary electronics that enables them to collect and exchange data making them responsive. Internet of Things is essentially an architectural framework which allows integration and data exchange between the physical world and computer systems over existing network infrastructure.

For consumers, IoT means you can adjust your home's thermostat from across an ocean. But for business, IoT can create new opportunities to connect with customers and partners – as well as volumes of data to collect, store, and analyze. In few words IoT is the new generation of devices that can be used as for business as for personal. Mobile devices are used frequently to provide easy control over IoT devices and supply the way of data communication.

According to IDC, a global provider of marketing intelligence, IoT is expected to grow at a compound annual growth rate (CAGR) of 19.2% through the forecast period (2014-2020), whereas all the «other» connected devices will grow at a CAGR of 9.5%. The fig. 1 show graphic of IoT versus other connected devices [2].

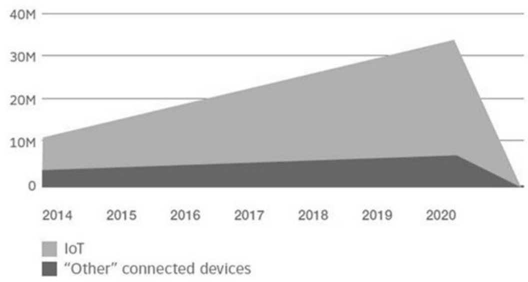

Fig. 1. IoT Versus Other Connected Devices

Most of IoT devices have major problem. Because of performance leak they will not be able to use secured protocols or have custom security system. In order to use https protocol for securing data communication, IoT devices will need to have additional 30mb memory and processor that can handle cryptography. It is obvious that increase of their performance will influence on their price.

The best solution is to find a way that will not require software or hardware changes in IoT devices and will provide required security during data exchange. Also it is very important to keep data exchange efficiency at high level, because it could be the major requirement for many devices.

Theoretically that could be a device which can handle multiple connections through wireless connection. That device will perform role of central cryptosystem that can receive data from different IoT devices encrypt them and only then send it to the cloud environment. The same process should be applied for receiving data.

In order to create such device, the listed above requirements that should be satisfied, ability to run and maintains custom software solution in order to support all Internet of Things devices. Most of IoT devices have major problem, because of performance leak they are not able to run custom software solution. in order to automatically send, store, revert data.

Performance to process nested software solutions. That device should have enough good performance to process encryption or decryption with almost any delays.

Support wireless connection. Almost all IoT devices are connected to the internet directly or through some middleware. It createshuge amount ways of IoT devices usage. In order to provide good user experince the preferable way of connections for IoT devices are wireless connections. Nowadays they mostly use Bluetooth or Wi-Fi connection. So it is very important to have support for both wireless technologies in our device that will be used to provide secure data communication.

Another important requirement for such device is portability. The communication with IoT devices isn't limited with any specific place. They are used in homes, offices, restaurants and etc... So the device that should provide secured interaction with them must have enough small size and weight that will make it almost unnoticeable for customer who will use it. Also to make device fully portable it should be equipped with external energy sources. It should be accumulator that will supply enough energy to keep device in working state for several hours.

Device Selection

After summarizing requirements mentioned above, more technical requirement is composed:

– device should have multi core processor and half GB of operation memory in order to run decryption and encryption tasks almost without delays;

– Wi-Fi and Bluetooth module;

– OS that give ability to easy run custom programming environment;

– small and portable sizes.

For creation of such device Intel Edison has been used (fig. 2).

The Intel Edison is a tiny computer offered by Intel as a development system for wearable devices. The system was initially announced to be of the same size and shape as an SD card. It is a fully open source hardware and software development environment. It has high-performance, dual-core CPU, integrated Wi-Fi certified in 68 countries, Bluetooth 4, 1 GB DDR and 4 GB FLASH memory.

As an operation system Edison uses Yocto OS that has several disadvantages related to the new modules installation. Because of this instead of Yocto Ubilinux is used, which is OS based on. It has all requirements and after several adaptations, Edison became fully supported by it. For easy device setup packages has been created which setup Ubilinux and adjust all settings automatically. The package also made to installs custom developed software solution that is handling all data securing processes. The Software solution that is installed in device is created in node js environment.

Crypto System

To provide secure communication between cloud and access point a custom software service has been developed. It is composed of 2 different parts, a cloud and client. The above mentioned packaged for configuring Edison also set ups client service automatically, which includes web panel where user can manage encryption and decryption process. In order to secure device communication with the cloud environment the user should register cloud URI in the web panel. Afterwards the registration will give the user ability to download the private key which will be used for data decryption on the cloud side. Client side service also uses Package Capturer that checks if each received package from IoT device should be secured (encrypted) before sending it to the cloud. If package destination URI matches with the user filled URI, then symmetric encryption will be applied using generated private key. Elsewhere it will just resend package to its destination. The same workflow works for received packages but instead of encryption decryption is applied and raw data is sent to IoT device.

Cloud software is Node JS and Java languages. API takes a care for packages decryptions and encryptions on server side and it can be attached to any exist application written in corresponding language. After a small modification of exist application attached API secure the way cloud communicates with IoT device. Whenever the package is received it is analyzed by cloud API to check the constant value in the header that shows if package contains encrypted data. If package contains encrypted data, then it will be decrypted with client provided key otherwise it will not be changed.

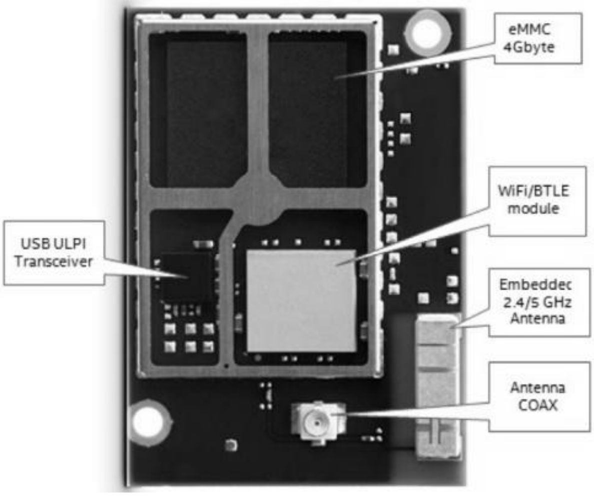

Fig. 2. The Intel Edison

One of the challenges here is to provide high efficiency that will be able to run on Edison device and perform real time data encryption or decryption. To achieve real time encryption AES algorithm was changed and adapted to Edison device.

Advanced Encryption Standard (AES) also referred as Rijndaelis is one of the efficient algorithms in the world. The AES standard has a constant block size of 128 bits with 3 different key sizes of 128 bits, 192 bits and 256 bits, where 10, 12 and 14 encryption rounds will be applied for each key size, respectively. During the encryption and decryption processes, the 16 bytes of data will form a changeable (4*4) array called the state array. During the encryption process, the state array initially consists of the input data. This array will keep changing until reaching the final enciphered data. In the decryption process the state array will start by the enciphered data and will keep changing it until retrieving the original data. The encryption of AES is carried out in blocks with a fixed block size of 128 bits each [3]. The AES cipher calculation is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform the cipher text back into the original plaintext using the same encryption [3].

Using AES for files encryption looked like a good idea until speed testing. The results that give this symmetric algorithm are too slow. The length of key for testing has been selected 128. For 1mb data encryption different libraries use 4-7 seconds which is quite a big amount of time. In case of smartphones this time will increase several times, as the user that records small video (100mb) on his mobile device and wants to encrypt it should wait more than 20 minutes. Also during encryption time most of the device resource will be used and the device will become non-usable. Based on this test we can come to the conclusion that it is not possible to use a pure AES. In order to make AES work faster several improvements have been done.

One of the things that have been done to improve speed of AES is using parallel computing. Parallel computation is a method which allows carrying out several computations simultaneously on two or more microprocessors. Parallel computation can be performed by using multicore and multiprocessor computers having multiple processing elements within a single machine. Parallel computation usage not change the original algorithm itself. Appended written code divides the plaintext into blocks which can be encrypted and decrypted independently. In this way the multiple blocks can be processed simultaneously. The first operation here is separation of the plain text or cipher blocks into independent streams and then application of the AES encryption or decryption procedures. The speed improvement measurement depends upon device (the number of the processing units). It can be faster up to 3 times. In case of mobile devices speed improvement is very important.

Intel AES instructions are a new set of instructions available beginning with the all-new 2010 Intel® Core™ processor family based on the 32nm IntelÂ® micro architecture codename Westmere. The architecture consists of six instructions that offer full hardware support for AES. Four instructions support the AES encryption and decryption, and the other two instructions support the AES key expansion. They offer a significant increase in the performance compared to the current pure-software implementations. Also the AES instruction provides important security protections. In order to support Intel AES instruction for flash player native extension has been written that delegates several operations. AESENC, AESENCLAST, AESDEC, AESDECLAST are defined by the pseudo code. These instructions perform a grouped sequence of transformations of the AES encryption/decryption flows (in fact, they perform the possible longest sequence, without introducing a branch in an instruction). The above described changes can increase performance by more than an order of magnitude for parallel modes of operations. Beyond improving performance, the new instructions help address software side channel vulnerabilities, because they run with data-independent latency and do not use lookup tables.

The 2 mentioned above improvements processed about 40 percent speed improvement compared to default realization.

Comparison of improved algorithm with original Table 1

Data Size	AES	AES Improved
1MB	0,8-0,9 second	0,6-0,7 second
10MB	7-8 seconds	5-5,5 second

Conclusion

In this paper described solution for problem related with secure data communication between IoT device and cloud environment. The created device serves as middleware point for IoT device and cloud environment. The improvement for default AES implementation described, in order to improve algorithm speed. Custom software solution is created for handling encryption and decryption tasks for received or sent data. Also some Operation System adaptation also help to adapt Intel

Edison device to our goals. The described solution can work not for only IoT devices but also for other devices that have corresponding wireless technologies.

References

[1] The Developer's IoT Playbook: What to Expect and Where to Turn to Fast-Track Your IoT Initiatives [Електронний ресурс]. – Режим доступу: https://www.business.att.com/content/whitepaper/idc-developers-iot-playbook.pdf.

[2] Internet of Things in Logistics, DHL Trend Research, Cisco Consulting Services [Електронний ресурс]. – Режим доступу: http://www.dhl.com/.content/dam/Local_Images/g0/New_aboutus/innovation/DHLTrendReport_Internet_of_things.pdf.

[3] MacGillivray C. Worldwide Internet of Things Forecast 2015-2020, IDC [Електронний ресурс]. – Режим доступу: http://www.idc.com/getdoc.jsp?containerId=prUS25658015.

[5] Saraireh S. Secure Data Communication System Using Cryptography [Електронний ресурс]. – Режим доступу: http://airccse.org/journal/cnc/5313cnc10.pdf.

УДК 004.056.3 (045)

Овсепян В. Безпечний обмін даних для Інтернету Речей
Анотація. *Розроблено метод, який забезпечує безпечний обміну даними за допомогою портативного обчислювального модуля. Обчислювальний модуль служить третьою стороною для захисту даних, що знаходяться з різних ІО пристроях або хмарних системах. Він є проміжним шаром між 2 різними кінцевими точками і може обробляти кілька з'єднань через бездротові або провідні мережі. Адаптація ОС виконано для того, щоб легко запускати призначені для користувача програмні рішення і адаптувати пристрій до цілей. Шифрування здійснюється за допомогою симетричного алгоритму, який був адаптований до обчислювального модуля для того, щоб забезпечити високу швидкість і рівень безпеки. Два основних удосконалення алгоритму пов'язані з використанням інструкції AES і використання паралельних обчислень. Відповідне програмне рішення, яке буде керувати шифруванням і процес розшифровки також описані. Програмне рішення складається з 2-х частин, де перша частина спочатку встановлена в обчислювальному модулі, а друга частина потребує установки в хмарному середовищі. Ці 2 частини працюють разом, як одна системи і забезпечують безпечний обмін даних, не потребуючи багато змін в середовищі програми.*
Ключові слова: *безпека, Інтернет речей, хмарні технології.*

Овсепян В. Безопасный обмен данных для Интернета Вещей
Аннотация. *Разработан метод, который обеспечивает безопасный обмена данными при помощи портативного вычислительного модуля. Вычислительный модуль служит третей стороной для защиты данных, поступающих из различных ИВ устройств или облачных систем. Оно является промежуточным слоем между 2 различными конечными точками и может обрабатывать несколько соединений через беспроводные или проводные сети. Адаптация ОС выполнено для того, чтобы легко запускать пользовательские программные решение и адаптировать устройство к целям. Шифрование осуществляется с помощью симметричного алгоритма, который был адаптирован к вычислению модуля для того, чтобы обеспечить высокую скорость и уровень безопасности. Два основных усовершенствования алгоритма связаны с использованием инструкции AES и использованием параллельных вычислений. Соответствующее программное решение, которое будет управлять шифрованием и процессом расшифровки также описан. Программное решение состоит из 2-х частей, где первая часть изначально установлена в вычислительном модуле, а вторая часть нуждается в установке в облачной среде. Эти 2 части работают вместе, как одна система и обеспечивают безопасный обмен данных, не запрашивая много изменений в среде приложения.*
Ключевые слова: *безопасность, интернет вещей, облачные технологи.*

Отримано *4 травня 2016 року*, затверджено редколегією *18 травня 2016 року*