Secure Communication Technology for Devices with Limited Resources

Nikita Vysotskiy[1], Amir Makhmutov[1], Konstantin Mironov[1,2], Marcus Meisel[3], Thilo Sauter[3]

[1] *Chair of Computer Technology and Information Security, Ufa State Aviation University,*
Karl Marx 12
Ufa, Russia
E-mail: chubays123@ya.ru, makhmutovamir15@gmail.com, mironovconst@gmail.com

[2] *Institute of New Materials and Technologies, Ural Federal University*
Mira 19
Ekaterinburg, Russia
E-mail: mironovconst@gmail.com

[3] *Institute of Computer Technology, Technische Universitaet Wien*
Gusshaustrasse 27-29
Vienna, Austria
E-mail: marcus.meisel@tuwien.ac.at, Thilo.sauter@donau-uni.ac.at

Abstract

In this article, we present a proposal for a secure data transmission technology. This technology can be used to build various low-power wireless networks, for example, in smart home systems, industrial wireless sensor networks, communication networks of Smart Grids. The peculiarity of such networks is that the devices from which they consist do not have sufficient power to support execution of asymmetric crypto algorithms. In the proposed architecture, cryptographic protection of data is provided by using exclusively symmetric crypto algorithms.

Keywords: Wireless sensor networks, communication, symmetric cryptography, microcontrollers.

1. Introduction

As the technologies of the Internet of things develop, local networks become more and more popular, the nodes of which are low-power specialized computing devices. Examples of such networks are smart home control systems [1], Smart Grid communication networks and industrial wireless sensor networks.

Like other communication systems, such networks require the protection of data [2]. Particularly acute is the issue of security in networks serving the critical infrastructure - electrical networks, large industrial facilities, etc. At the same time, standard cryptographic protection measures are often ignored when building up such networks. The power of the devices used is not provided to execute asymmetric algorithms.

There are several encryption key management systems, but there is no final solution [3]. Some authors support open key infrastructures (PKI), however, this approach has a number of shortcomings in the framework of application in low-power systems, the most obvious of which is the inability to meet stringent requirements with respect to resources expended and computation time, since the hardware capabilities are strictly limited. Also, the process of revoking compromised keys requires more complex mechanisms than in symmetric schemes, where simple key updating is sufficient [6]. For example, ability to generate keys in devices without the need for communication can be as a solution [4, 5]. In this case, regular updating of keys does not cause additional overheads [6].

In the work of Sauter and Treytl, the infrastructure of symmetric keys for low-power networks was first proposed. The infrastructure is universal with respect to symmetric algorithms of encryption and hardware. Process of work has been tested on a primitive 8051 controller with symmetric encryption algorithms DES and AES [6].

In this article, a technology is proposed with the use of concrete hardware and concrete encryption algorithms. The article is organized as follows: In the second section, we describe the hardware that is proposed to be used to build a secure low-power sensor network. In the third series, cryptographic data protection is discussed, and conclusions and suggestions for further work are given in section 4.

2. Hardware and Communication

Nowadays, there is a huge variety of cryptographic methods of data protecting. There are many algorithms, each of which has its pros and cons [13]. But the main drawback, namely for microcontrollers, is a lot of mathematical calculations, which in turn creates a huge delay when processing data packets. It is necessary to minimize existing encryption algorithms without much damage to cryptographic strength.

To implement complex and simple encryption algorithms we will need microcontrollers with good memory capacity and computational power. As a platform for prototype, we decided to use a chip, based on AVR. There are several AVR chips [10], but the final choice for prototype is ATmega328P chip. This solution has enough resources and performance to execute all mathematical operations in a short time. Below is a structural diagram of this chip and the characteristics [7].

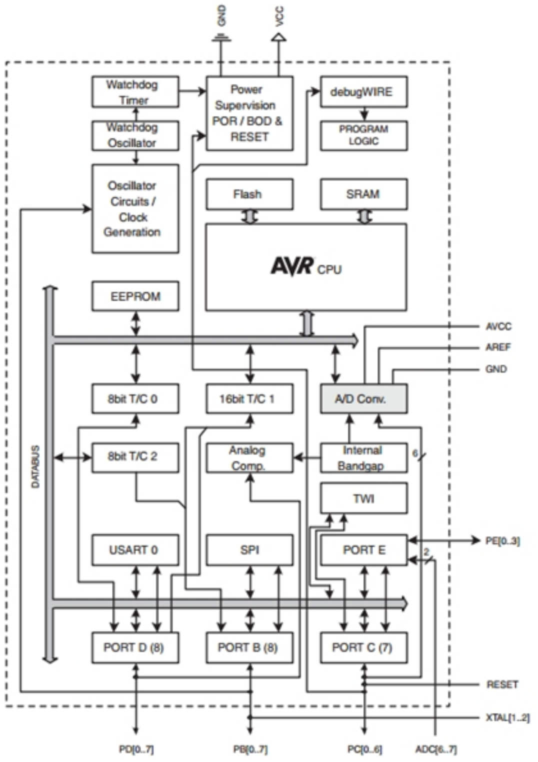

Fig. 1. Structural diagram of ATmega328p.

Characteristics:

- 32 kB Flash
- 2 kB RAM
- 1 kB EEPROM
- 8-bit ATMEL AVR architecture, up to 20MHz, in Arduino works at 16MHz. 1 MIPS / MHz
- Supply voltage 5V or 3.3V
- I2C bus
- SPI bus
- 1 UART
- 1x 16bit timer
- 2x 8bit timer

Due to the development environment of Arduino IDE, the programming process is simplified as much as this environment eliminates a lot of settings and additional project configurations.

After successful tests of the work done on the prototype, it is planned to change the development platform to a more serious one, namely STM. These controllers have more computing resources in comparison with AVR. Another important point is that STM is cheaper than AVR several times and has better performance results. The disadvantage is the complexity of a process of developing projects because of the specific nature of existing development environments. There is a difficulty in setting up [15] and creating a configuration file. Below is a structural diagram of STM32f103c8 chip and the characteristics [8]. Later this chip can be changed to another STM-platform based [11].

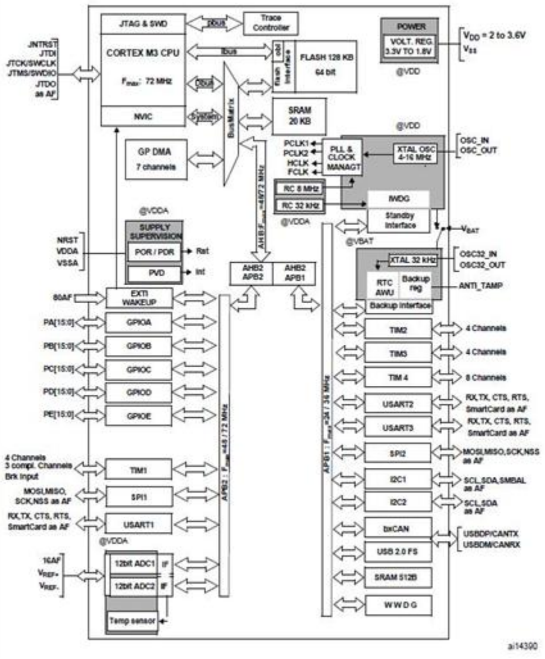

Fig. 2. Structural diagram of STM32f103c8.

Characteristics:

- 64 kB Flash
- 20 kB RAM
- 32bit ARM CORTEX-M3 architecture, up to 72MHz, 1.25MIPS / MHz
- Power supply 3.3V (2.7-3.6)
- 3x USART

- 2x I2C
- 2x SPI (18Mbit/s)
- 1 x CAN 2.0B
- USB 2.0 FS (FullSpeed - 12Mbit)
- 3x 16bit timer + 1 PWM timer
- DMA - 7 channels (ADC, SPI, I2C, USART)
- RTC - real time clock

To transfer information between the controllers, the HC-12 (NS-12) radio transmitter was chosen. This is a half-duplex wireless serial communication module with 100 channels in the range 433.4-473.0 MHz, capable of transmitting data for a distance of up to 1 km [9]. It is planned to create software modules capable of recreating a self-organizing network analogous to ZigBee [14] or XBee [12] on the basis of this module. Below are the characteristics of this transceiver.

Characteristics:

- Operating frequency - 433.4 - 473.0 MHz
- External antenna usage only
- Transmission range - up to 1000 - 1800 m in the open space, depending on the mode of operation
- Transmitter power - up to 100 mW (settings for 8 power levels are available)
- Number of data transmission channels - 100
- Four operating modes
- Built-in microcontroller (present on the module) STM8S003F3
- Interface for communication with external devices - UART
- Current consumption - from 3.6 mA to 16 mA, depending on the mode of operation
- Peak current consumption - up to 100 mA (data transfer)
- Current consumption in standby mode - 80 µA
- Supply voltage - from 3.2 V to 5.5 V.

3. Cryptography

Currently, cryptographic protection of data is not as widespread in the Internet of things, as it is required by the current state of affairs [20], in most existing solutions it is either not available, or implementations in which it is available are much more expensive. To ensure the safe communication of devices in the project it is decided to use a cryptographic protection of data. At the moment, the symmetric cryptographic algorithms such as AES128, AES256, DES, RC4 and the algorithm with the public key RSA are located at the stage of implementation and optimization of the program code. After the implementation of the software is done, it is planned to do an evaluation of the efficiency in terms of the time spent on processing the data flow, cryptographic strength, and power consumption on the prototype hardware. In case of an unsatisfactory result, further optimization of the program code will be continued.

The system of cryptographic protection of data in this project is constructed as follows: the flow of processed data is encrypted using a symmetric algorithm, for example, AES256 (the RC4 algorithm is used in the process of generating a secret key with taking into account all requirements that increase its cryptographic strength [16]), and the transmission of the encryption key occurs via an asymmetric algorithm, for example, RSA. The explanation for the choice of such encryption system configuration is that symmetric algorithms take less time to encrypt and decrypt information than asymmetric ones [6], and the level of cryptographic strength of certain ones, for example, the same AES256, is quite high [17]. Asymmetric cryptographic algorithms spend much more resources, but they allow you to transmit encrypted information over unprotected channels of information transmission safely. A specific example is transmission of an encryption key for a symmetric algorithm.

When using the above-mentioned configuration of the data encryption system, the optimal resource consumption and a sufficient level of protection of the transmitted data are ensured.

The results of Sauter and Treytl show that the calculation of the Diffie-Hellman function, used in asymmetric crypto algorithms, requires significant computational power, as a result of which the encryption operation is performed a thousand times slower on the controller than encoding with a symmetric algorithm. Therefore, it is preferable to use a symmetric key infrastructure only.

As an encryption algorithm, we propose to use ARC4 (alleged RC4). This is one of the most easy to implement encryption algorithms, which does not require significant computing resources [19, 18]. However, this algorithm should be used with great caution due to the large number of potential vulnerabilities. An additional research of potential vulnerabilities of the algorithm during the implementation of symmetric key infrastructure is required.

4. Conclusion

This article discusses the technology for ensuring secure communication between low-power devices. This article is a proposal of such technology - it is proposed to use a specific hardware and crypto algorithm. In the course of further research, it is planned to develop software for processing transmitted and received data and to check the state of security of data transmission using the ARC4 algorithm.

Acknowledgements

The research is supported by the Russian Fund for Basic Research, grant #17-07-00351.

References

1. Number of smart homes in Europe and North America [In Russian] URL: https://iot.ru/gorodskaya-sreda/kolichestvo-umnykh-domov-v-evrope-i-severnoy-amerike-v-2016-godu-dostiglo-30-3-mln (Visited on 17.05.18)
2. Nian Liu, Jinshan Chen, Lin Zhu, Jianhua Zhang, Yanling He, A Key Management Scheme for Secure Communications of Advanced Metering Infrastructure in Smart Grid, *J. IEEE Trans. on Industrial Electronics*, vol. 60, no. 10 (Oct. 2013) pp. 4746-4756.
3. S. Fuloria, R. Anderson, F. Alvarez, K. McGrath, Key management for substations: Symmetric keys, public keys or no keys?, *J. IEEE/PES Power Systems Conference and Exposition (PSCE)* (20-23 March 2011) pp. 1-6.
4. A. Treytl, T. Sauter, "Security Concept for a Wide-Area Low-Bandwidth Power-Line Communication System," in *Int. Symp. on Power Line Communications and its Applications (ISPLC)* (Vancouver, 2005) pp. 66-70.
5. Sungwook Kim, Eun Young Kwon, Myungsun Kim, Jung Hee Cheon, Seong-ho Ju, Yong-hoon Lim, Moon-seok Choi, A Secure Smart-Metering Protocol Over Power-Line Communication, *J. IEEE Trans. on Power Delivery*, vol.26, no.4 (Oct. 2011) pp. 2370-2379.
6. Hierarchical Key Management for Smart Grids [In Russian]. URL: https://ieeexplore.ieee.org/document/7302803/ (Visited on 04.05.18).
7. ATmega 328P Datasheet [In Russian]. URL: http://mkprog.ru/wpcontent/uploads/2017/09/ATmega328-328P_Datasheet.pdf (Visited on 05.05.18).
8. STM32F405 Datasheet [In Russian]. URL:http://www.st.com/resource/en/datasheet/dm00037051.pdf (Visited on 29.04.18).
9. HC-12 Receiver Datasheet [In Russian] URL: http://avrproject.ru/112/rf_hc12/2016-01-14_122335_HC-12_v2.3B.pdf (Visited on 29.04.18).
10. ATmega 2560 Dtasheet [In Russian] URL: http://www.alldatasheet.com/datasheet-pdf/pdf/107092/ATMEL/ATMEGA2560.html (Visited on 29.04.18).
11. STM32F4Discovery Datasheet [In Russian]. URL:http://www.st.com/en/evaluation-tools/stm32f4discovery.html (Visited on 25.04.18).
12. XBee Datasheet [In Russian] URL: https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf (Visited on 20.04.18).
13. A Glossary of Cryptographic Algorithms [In Russian] URL: https://www.globalsign.com/en-sg/blog/glossary-of-cryptographic-algorithms/ (Visited on 01.05.18)
14. ZigBee: Looking Deeper [In Russian] URL: http://www.kit-e.ru/articles/wireless/2005_4_144.php (Visited on 22.04.18).
15. How to start working with STM32 [In Russian] URL: https://geektimes.com/post/255334/ (Visited on 19.04.18).
16. RC4 Vulnerabilities [In Russian] URL: https://paginas.fe.up.pt/~ei10109/ca/rc4-vulnerabilities.html (Visited on 23.04.18).
17. Why 256 bits is enough forever [In Russian] URL: https://xakep.ru/2012/12/28/59888/ (Visited on 03.05.18).
18. Properties of software-implemented stream ciphers for the example RC4, GI, Vesta [In Russian] URL: http://www.dissercat.com/content/svoistva-programmno-realizuemykh-potochnykh-shifrov-na-primere-rc4-gi-vesta (Visited on 25.04.18).
19. What is RC4 [In Russian] URL: https://paginas.fe.up.pt/~ei10109/ca/rc4.html (Visited on 23.04.18).
20. Safe smart home: a complex technology, useful to everyone [In Russian] URL: http://news.ifmo.ru/ru/startups_and_business/startup/news/5832/ (Visited on 14.05.18).