ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РЕСПУБЛИКИ КАЗАХСТАН

ҚазҰТЗУ ХАБАРШЫСЫ

ВЕСТНИК КазНИТУ

VESTNIK KazNRTU

№2 (126)

АЛМАТЫ 2018 МАРТ

Учредитель:

Казахский национальный исследовательский технический университет
имени К.И. Сатпаева

Регистрация:

Министерство культуры, информации и общественного согласия
Республики Казахстан № 951 – Ж “25” 11. 1999 г.

Основан в августе 1994 г. Выходит 6 раз в год

Адрес редакции:

г. Алматы, ул. Сатпаева, 22,
каб. 616, тел. 292-63-46
Nina. Fedorovna. 52 @ mail.ru

Современная классификация методов решения задач на построение:

1.Метод геометрических мест.

2.Метод геометрических преобразований

В ряде случаев полезны те или иные геометрические преобразования. Часто употребляются следующие преобразования:

1) симметрия относительно оси;
2) вращение (в частности, симметрия относительно точки);
3) параллельный перенос;
4) преобразование подобия (в частности, гомотетия);
5) инверсия.

3. Алгебраический метод.

Таким образом, приведенная выше классификация по А. П. Киселеву в принципе соответствует современной классификации задач методов решения на построение на плоскости.

ЛИТЕРАТУРА

[1] Киселев А.П. Элементарная геометрия. Книга для учителя. - М.: Просвещение, 1980.

[2] Энциклопедия элементарной математики, том 4, Геометрия. - М.: Государственное Издательство Физико-Математической Литературы, 1963.

Мұстафин М.А.

Геометриялық салуға есеп әдісі А.П.Киселев классификациясы

Түйіндеме: Мақаланың мақсаты – классификация мәселені қарау геометриялық салуға есеп беру. Геометриялық салуға есеп әдісі алғашқы рет көрнекті орыс кеңес педагог А.П.Киселев ұсынады.

Негізгі сөздер: геометриялық салуға есеп, классификация.

Mustafin M.A.

A.P.Kiselev methods classification of problem solutions on construction

Summary. Given article is devoted to questions of classification of methods of problem solutions of geometric problems on construction. For the first time methods of solving problems on construction were proposed by outstanding Russian Soviet teacher A.P.Kiselev.

Key words: geometric problems on construction, classification.

UDC 004.056.55

A. Pyrkova, Zh.E. Temirbekova
(Al-farabi Kazakh national university, Almaty Republic of Kazakhstan,
temyrbekovazhanerke2@gmail.com)

PERFORMING SYMMETRIC ENCRYPTION MBED PLATFORM

Abstract. This article compiles of the cryptography programming and debugging techniques writing the Arm Mbed TLS platform, the code environment running on BLE Nano kit microcontroller. C++ was used as the core language instead of C, in order to increase programming effectiveness.

Keywords: BLE Nano kit microcontroller, mbed platform, Arm Mbed TLS, cryptography.

Arm Mbed TLS makes it easy for developers to include cryptographic and SSL / TLS functions in their (embedded) products, which makes this functionality easier with a minimum code size. It offers SSL / TLS library with an intuitive API and read the source code and includes a complex set of tests. You can create it from the box in most systems or manually select and configure functions [1].

The Mbed TLS library provides a set of cryptographic components that you can use and compile separately, and include or exclude using a single configuration header file [2]. Mbed TLS also provides a central SSL/TLS module that builds on the cryptographic components, the abstraction layers and the support components to provide a complete protocol implementation for SSL and TLS.

From a functional perspective, the library is split into three major parts:

- The SSL/TLS protocol implementation.
- A cryptographic library.
- An X.509 Certificate handling library.

BLE Nano kit is the smallest Bluetooth 4.1 Low Energy (BLE) development board in the market. The core is Nordic nRF51822 (an ARM Cortex-M0 SoC plus BLE capability) running at 16MHz with ultra low power consumption [3].

Developing a Bluetooth Smart enabled 'appcessory' (accessory device + companion application) is easier than ever. You can quickly produce prototypes and demos target for IoT and other interesting projects. BLE Nano could operate under 1.8V to 3.3V, therefore it works with a lot of electronic components.

Features:

- Smallest BLE development board, only 18.5mm x 21.0mm
- Nordic nRF51822 ARM Cortex-M0 SoC supports both BLE Central and BLE Peripheral roles
- 2.4 G Hz transceiver
- Ultra low power consumption
- Support voltage from 1.8V to 3.3V
- Software development using mbed.org, GCC, Keil or Arduino
- Lots of libraries and examples available
- Easy firmware deployment with the MK20 USB board
- Work with our free Android App and iOS App [4].

BLE Nano microcontroller supports hardware platforms mbed. Microprocessors are a series of mbed ARM microcontroller development boards designed for rapid prototyping.

The mbed platform is a free C/C++ compiler and IDE provided by ARM and suitable for programming most ARM microcontrollers [5]. The IDE is Web based, something which may back some of us down. I was myself skeptical about the practicality of such IDE compilers, however I must say the mbed compiler has proven to be easy to use, fast and (reasonably) friendly, even for a C/C++ compiler.

The many available platform libraries are easy to use and well designed, specially for starters, resembling (vaguely) the simplicity you may have enjoyed when using Arduino libraries. Also, mbed encourages and eases sharing with others your own libraries, or any piece/snippet of code. You have to register a username in the platform from the very beginning, that will allow you to store your projects online, comment, share code and participate in the community forums comfortably [4-6]. Compiling and programming your projects is really easy. You can literally have the blink test application running in a few minutes.

mbed has a dedicated team which develops and maintains a very nice Bluetooth Low Energy API. You can rest assured that using this API for learning to develop BLE appcessories will be much easier (and faster) than trying to do so by using directly the BLE libraries (Soft Device) provided by Nordic [7]. The BLE API includes also several application examples.

Cryptographic library

The mbed TLS library was designed to easily integrate with existing (embedded) applications and provide building blocks for secure communications, cryptography and key management. This tutorial will help you understand the steps that need to be taken when you want to do it.

The cryptographic part of mbed TLS has abstraction levels for public-key cryptography, Hashing (message digests) and symmetric ciphers [8-9]. It also contains standard-based random number generators and an entropy pool.

All cryptographic algorithms are implemented as loosely-coupled modules. You can just take the appropriate header files and source code files and drop them in your project as needed.

mbed TLS provides the most commonly used algorithms, such as AES. mbed TLS supports alternative implementation for most of its cryptography modules. A common use case is for hardware accelerated cryptography engines [10]. There are a couple of methods for alternative implementations: specific function replacement and full module replacement. The configuration file contains the cryptography modules, which you can replace with alternative implementation. These are named MBEDTLS_<MODULE NAME>_ALT. In order to support alternative implementation for a module, uncomment the corresponding *_ALT definition. Function replacement is according to the function name, with the suffix of _ALT [11-13]. To support hardware entropy source, enable MBEDTLS_ENTROPY_HARDWARE_ALT in the configuration file.

Mbed TLS uses a continuous integration system to make sure we maintain the highest possible code quality. Our system checks all committed code on an ever growing set of operating systems and chipsets, covering:

- Regression testing.
- Test vectors.
- Multiple static analyzers.

- Testing for compatibility.
- Behavioural testing.
- Security testing through fuzzing.
- Function and unit testing against known based values.
- Code coverage testing.
- Validation testing .

AES is based on a design principle known as a substitution-permutation network, a combination of both substitution and permutation, and is fast in both software and hardware [14-15]. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits.

In each round, the AES uses the four following operations:

•SubBytes:Each byte of the array is transformed using a nonlinear substitution box called the AES S-Box [16].The S-Box in the AES has been carefully constructed and the cipher uses only one S-Box throughout the encryption.

•ShiftRows: Is a transposition step which ensures that the last threerows of the array are shifted by adifferent number of byte positions.

•MixColumns: Mixes each column in the array to create even more diffusion.

•Addkey:Using bitwise XOR, each byte of the array is mixed witha byte of a sub-key material, alsocalled round-key [17-18]. The sub-key ismade by "key expansion" and is derived from the main cipher key using a Rijndael key-schedule.

The overall structure of the cipher begins with a key expansion before going in to the pre-whitening which only includes AddKey. Then the rounds performing Sub-Bytes, ShiftRows, MixColumns and Add-Key will loop for $n-1$ rounds where n is the numbers of rounds. The n-th round or final round does not include MixColumns and produce the output ciphertext.

AES context structure. buf is able to hold 32 extra bytes, which can be used:

- for alignment purposes if VIA padlock is used.
- to simplify key expansion in the 256-bit case by generating an extra round key.

AES-CBC buffer encryption/decryption. Length should be a multiple of the block size (16 bytes).

Upon exit, the content of the IV is updated so that you can call the function same function again on the following block(s) of data and get the same result as if it was encrypted in one call. This allows a "streaming" usage [19]. If on the other hand you need to retain the contents of the IV, you should either save it manually or use the cipher module instead. param ctx-AES context, param mode-MBEDTLS_AES_ENCRYPT or MBEDTLS_AES_DECRYPT, param length-length of the input data, param iv -initialization vector (updated after use), param input-buffer holding the input data, param output-buffer holding the output data, return 0 if successful, or MBEDTLS_ERR_AES_INVALID_INPUT_LENGTH.

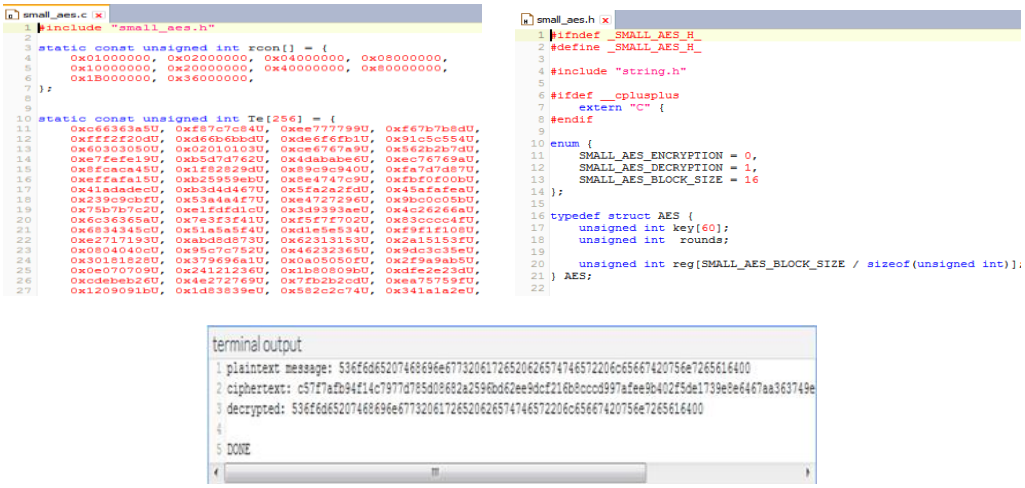

Figure 3. Performance of ciphers in mbed platform.

AES also has one advantage over other encryption algorithms. As the name suggests, AES is an industry standard, it is not patented and is not free for everyone. One of the attributes of the AES algorithm is that it must work reliably on all platforms as hardware and software, and therefore AES is well documented and distributed around the world [20-21]. This means that if the AES breaks, regardless of whether the headers do not touch the company that used the hacked AES, they will touch the flags in something that many companies and people use.

Security is fundamental to the successful implementation of the Internet of things. Currently, Edgenodes are the weakest link in securing IoT security and protecting the cryptographic key. The best way to achieve a lock is protected equipment. This is the only way to remove these keys and other secrets from prying eyes. The IoT device can only be protected as the weakest link.

REFERENCES

[1] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, “The most dangerous code in the world: validating SSL certificates in non-browser software,” in Proceedings of the 2012 ACM conference on Computer and Communications Security. ACM, - 2012, - pp. 38–49.

[2] Red Bear Company, http://redbearlab.com/blenano/, Retrieved: 18th January, 2018.

[3] Jose Angel. BLE Nano hardware development kit for Bluetooth Low Energy – 2015. – pp. 109

[4] Atmel. http://www.atmel.com/dyn/products/devices.asp?family id=607 Retrieved: 20th January, 2018.

[5] Eide, E., Regehr, J. Embedded C++ Technical Committee. “The embedded C++ programming guide lines”. http://www.caravan.net/ec2plus/guide.html Retrieved: 20th January, 2018

[6] Tsai, C., Lai, C., &Vasilakos, V. Future internet of things: Open issue and challanges. ACM/SpringerWirelessNetworks,. – 2014. pp. 225

[7] Li, S.; Xu, L.D. and Zhao, S., "The internet of things: a survey", Information Systems Frontiers, vol. 17, No.2, - 2015. - pp 243-259

[8] https://tls.mbed.org/ Retrieved: 19th January, 2018.

[9] Simon N. FoleyDieter GollmannEinar Snekkenes. Computer Security – ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part 1, - p 286

[10] A.L. Fardan, N.J. Paterson, Breaking the TLS and DTLS record protocols. – 2013. pp. 201

[11] Katherine Q. Ye, Princeton U., Carnegie Mellon U, Verified Correctness and Security of mbedTLS HMAC-DRBG, - 2017. pp. 198

[12] Karthikeyan Bhargavan, Cédric Fournet, Markulf Kohlweiss, Alfredo Pironti, and Pierre-Yves Strub. Implementing TLS with veri ed cryptographic security.In 2013 IEEE Symposium on Security and Privacy. IEEE, - 2013. - pp. 445–459.

[13] A.R.M. Ltd: mbed TLS buys leading IoT Security company offspark as it Expands its mbed platform (2018).

[14] Kubilay Atasu, Luca Breveglieri, and Marco Macchetti. Efficient AES implementations for ARM based platforms. In Proceedings of the 2004 ACM Symposium on Applied Computing, - 2004, - pp. 841–845.

[15] Daniel J. Bernstein. Cache-timing attacks on AES. https://cr.yp.to/antif orgery/cachetiming-20050414.pdf, - 2005. pp. 105-109

[16] Viega, John, and Matt Messier: Secure Programming Cookbook for C and C++ (O’Reilly & Associates, 2003) -2003 - pp. 231-232

[17] K. Nyberg, "Differentially uniform mappings for cryptography, Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765, T. Helleseth, Ed., Springer-Verlag, - 1994, - pp. 55-64.

[18] E. Biham, "New types of cryptanalytic attacks using related keys," Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765, T. Helleseth, Ed., Springer-Verlag, - 1993. - pp. 398-409.

[19] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, - 1999. pp. 25-26

[20] J. Daemen and V. Rijmen, The block cipher Rijndael, Smart Card research and Applications, LNCS 1820, Springer-Verlag, - 2010. - pp. 288-296.

[21] Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael" (PDF). National Institute of Standards and Technology. p. 1. - 2013. – pp. 105-107.

Пыркова А.Ю., Темирбекова Ж.Е.

Выполнение симметричного шифрования в mbed платформе

Аннотация. В этой статье собраны методы программирования и отладки криптографии, написанные на платформе Arm Mbed TLS, среда кода, работающая на микроконтроллере BLE Nano**.** Использовался как основной язык C ++ вместо C, чтобы повысить эффективность программирования.

Ключевые слова: микроконтроллер BLE Nano kit, платформа mbed, Arm Mbed TLS, криптография.

Пыркова А.Ю., Темирбекова Ж.Е.

Симметриялық шифрлеуді mbed платформасында жүзеге асыру

Түйіндеме. Бұл мақала криптографиялық бағдарламалау және BLE Nano жиынтығы микроконтроллерінде жұмыс істейтін кодтық ортаны құрайтын Arm Mbed TLS платформасы туралы жазылған. Бағдарламасының тиімділігін арттыру үшін С орнына негізгі тіл ретінде С ++ пайдаланылды.

Түйін сөздер: BLE Nano kit микроконтроллері, Mbed платформасы, Arm Mbed TLS, криптография.

UDC 004.056.55

A. Pyrkova, Zh.E. Temirbekova
(Al-farabi Kazakh national university, Almaty, Republic of Kazakhstan,
temyrbekovazhanerke2@gmail.com)

POSSIBILITIES OF USING A BLE NANO KIT MICROCONTROLLER TO DEVELOP CRYPTOGRAPHIC LIBRARIES

Abstract. Nowadays, the number of IoT (Internet of Things) devices are growing rapidly around the world with wide range of purposes. Many types of these devices collect data and depending on their purpose, some of the data can be highly sensitive for the user. This calls for security on the device which has to secure the collected data and send it to a server through the Internet. Using TLS (Transport Layer Security) is a great way to provide such a security and one part of it is the data encryption which is the main focus of this thesis. The problem is that IoT devices are often made out of microprocessors having a low computational power and performance.

This article compiles some of the programming and debugging techniques writing the mbed platform, the code environment running on our BLE Nano kit microcontroller. C++ was used as the core language instead of C, in order to increase programming effectiveness. This turned out to be a good choice, but came at the cost of increased program size. Thus, a big part of optimizing was spent on reducing program and data size. Then demonstrate.The main goal of this thesis is to determine AES (Advanced Encryption Standard) symmetric encryption algorithms, security and space complexity using a specific encryption library from ARM mbed.

Keywords: IOT, BLE Nano kit microcontroller, C++, mbed platform, Arm Mbed TLS, AES cryptography.

BLE Nano kit is the smallest Bluetooth 4.1 Low Energy (BLE) development board in the market [1] . The core is Nordic nRF51822 (an ARM Cortex-M0 SoC plus BLE capability) running at 16MHz with ultra low power consumption.

Developing a Bluetooth Smart enabled 'appcessory' (accessory device + companion application) is easier than ever [2]. You can quickly produce prototypes and demos target for IoT and other interesting projects. BLE Nano could operate under 1.8V to 3.3V, therefore it works with a lot of electronic components.

Features:

- Smallest BLE development board, only 18.5mm x 21.0mm
- Nordic nRF51822 ARM Cortex-M0 SoC supports both BLE Central and BLE Peripheral roles
- 2.4 G Hz transceiver
- Ultra low power consumption
- Support voltage from 1.8V to 3.3V
- Software development using mbed.org, GCC, Keil or Arduino
- Lots of libraries and examples available
- Easy firmware deployment with the MK20 USB board
- Work with our free Android App and iOS App

BLE Nano kit microcontroller supported hardware platforms on mbed. The mbed Microcontrollers are a series of ARM microcontroller development boards designed for rapid prototyping [3-4].

The mbed platform is a free C/C++ compiler and IDE provided by ARM and suitable for programming most ARM microcontrollers. The IDE is Web based, something which may back some of us down. I was myself skeptical about the practicality of such IDE compilers, however I must say the mbed compiler has proven to be easy to use, fast and (reasonably) friendly, even for a C/C++ compiler [5-6].

The many available platform libraries are easy to use and well designed, specially for starters, resembling (vaguely) the simplicity you may have enjoyed when using Arduino libraries. Also, mbed

Химия-металлургия ғылымдары

Химико-металлургические науки

Chemical and metallurgical sciences

Редакторы:
Н.Ф. Федосенко
Верстка на компьютере:
Л.Т. Касжанова

Подписано в печать 29.03.2018 г.
Формат 60х84 $^{1}/_{8}$. Усл. п.л 37,8.
Тираж 500 экз. Заказ № 207.

Адрес редакции:
ул. Сатпаева, 22, КазНИТУ каб. 616, тел. 292-63-46 ,Nina.Fedorovna. 52 @ mail.ru
Департамент маркетинга и коммуникаций КазНИТУ
Казахского национального исследовательского технического университета имени К.И. Сатпаева