Journal of Network and Computer Applications 60 (2016) 130–143

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer

Natassya B.F. Silva [a], Daniel F. Pigatto [a,*], Paulo S. Martins [b], Kalinka R.L.J.C Branco [a]

[a] Laboratory of Critical Embedded Systems (LSEC), Institute of Mathematics and Computer Sciences (ICMC), University of Sao Paulo (USP), Sao Carlos, Sao Paulo, Brazil[1]

[b] University of Campinas, Campinas, Sao Paulo, Brazil

ARTICLE INFO

Article history:
Received 17 October 2013
Received in revised form
28 March 2014
Accepted 25 October 2015
Available online 27 November 2015

Keywords:
Performance evaluation
Cryptographic algorithms
Embedded systems

ABSTRACT

Embedded systems are associations between hardware and software designed to perform a specific function. These systems are usually part of a larger system and their wireless communications are a hallmark. Therefore, it is important to guarantee a secure communication by ensuring the confidentiality of information, which is obtained through cryptography. Security has not traditionally been considered a requirement in embedded systems design and the application of specific security techniques to these devices is still incipient. This paper presents a performance evaluation analysis of cryptographic algorithms in embedded systems (namely RC2, AES, Blowfish, DES, 3DES, ECC and RSA). Parameters considered in the analysis are average processor and memory usage, response time and power consumption. The results show that symmetric and asymmetric algorithms such as Blowfish and ECC have a good performance in embedded systems when properly chosen for each situation.

1. Introduction

Embedded systems are a combination of hardware and software designed to perform a specific function usually as part of a larger system (Barr, 1999). Some of these systems are considered critical because their malfunction or failure may result in large monetary losses or even loss of life, so dependability is essential (Januzaj et al., 2010). Examples of these systems are found in many applications including the avionics and the automotive domains, and specifically in the development of autonomous systems such as unmanned aerial vehicles (UAVs).

One concept that contributed to the dissemination of embedded systems is the issue of mobility. Nowadays most embedded systems are equipped with wireless communication interfaces. Therefore, it is now possible to deploy and move around these systems in scenarios where accessibility was formerly difficult (if not impossible), through the connection to sensor networks that are capable of monitoring a wide range of applications (e.g. forest fires or volcanic activity).

The combined high mobility and the wireless communications have further increased the exposure of these systems to malicious attacks. This new level of communications creates new security concerns since unauthorized users might be able to intrude in order to steal information, disrupt services and even damage physical or logical devices.

It is well known that embedded systems operate on limited resources such as power, processing and memory. Their architecture is usually composed of relatively simple microprocessors, memory and communication units. The development of software for these systems must take into account these constraints. It must also focus on optimizing resource usage while minimizing power consumption to ensure greater availability. These restrictions impose new challenges in providing security for these devices, since there must be a balance between security and processing.

One limitation in the design of embedded systems is that, traditionally, security has not been considered a priority requirement. In general, metrics such as cost, performance and power consumption have largely dominated the design of this class of systems (Ravi et al., 2004; Kocher et al., 2004). Consequently, embedded systems that communicate securely are not yet commonplace, and the immaturity of security in this domain has been emphasized by current literature (Henzinger and Sifakis, 2006; Brändle and Naedele, 2008).

Cryptography is the most common approach to guaranteeing security in computer systems. It can be implemented with different security levels according to application requirements. The use of cryptography can provide confidentiality, authenticity and integrity for these systems. The Diffie–Hellman and RSA algorithms, in addition to being the first publicly known examples of high quality public-key algorithms, have been among the most

* Corresponding author.
[1] URL: http://www.lsec.icmc.usp.br

http://dx.doi.org/10.1016/j.jnca.2015.10.007
1084-8045/

widely used. Other algorithms include the Blowfish, RC2, DES (and 3DES), AES, and various elliptic curve techniques.

In this paper, we evaluate the performance of cryptographic algorithms in the context of embedded systems, in order to allow for more security in this domain. We analyze the behavior of symmetric and asymmetric cryptographic algorithms both in a general purpose computing environment (i.e. laptop) and in a real embedded platform. By comparing both performances, it is possible to infer about the impact that the resource constrained architecture (i.e. embedded system) has on the performance of the algorithms. The work was developed through a set of case studies that analyzed cryptographic algorithms for their power consumption, memory usage and response time.

This paper is organized as follows: in Section 2 we discuss related work; Section 3 introduces the design of experiments; in Section 4 we present the case studies and the results obtained for each case; finally, the final remarks and the conclusion are presented in Section 5.

2. Previous work

In this section we address previous research work in the area of security for embedded systems. Ertaul and Lu (2005) propose the use of Elliptic Curve Cryptography (ECC) to data transfer and the secure sharing of keys in ad hoc networks. They used a set of equations to calculate the average response time of 7 ECC secret sharing algorithms, including El-Gamal and Diffie–Hellman. The goal (i.e. response variables) was to measure the timings of share split before encryption and share split after encryption. Their work compares ECC with RSA and concludes that, for the same level of resistance against the most known types of attacks, a system based on elliptic curves can work with much smaller keys.

Ramachandran et al. (2007) use Pocket PCs and sensors as test environment to develop secure communication protocols based on elliptic curve cryptography. Performance tests were developed to study the computational ability of these devices to process cryptographic functions. They used a benchmark to evaluate AES, SHA, MD5 and Pairings in a Pocket PC and a wireless sensor. The response variable was the algorithm response time (encryption and decryption) and for some experiments it was also calculated the time taken for primitive operations. Among the results, an important point emphasized by the authors is the need to optimize some math functions, especially those related to multiplications. Another important consideration is the difficulty in performing computationally intensive operations in sensors due to their severe limitations.

Hyncica et al. (2011) compare the generated code-size and the speed of encryption and decryption for fifteen symmetric ciphers on three different embedded microcontroller platforms. The embedded platform uses the 8-bits microcontrollers Freescale HCS08, Atmel ATmega128 and the 16-bits Texas Instruments MSP430. The implementation of the algorithms was taken from free open sources libraries and are AES, XTEA, RC5, Skipjack, SAFER, DES, Anubis, Twofish, CAST5, Kasumi, Kseed, Noekeon, RC2, RC6, Blowfish. They concluded that the highest throughput algorithms were AES, Twofish and SAFER and that most of the algorithms require less than 15% from the memory.

Limited resources such as power are also a concern when secure protocols are employed in these systems. Potlapally et al. (2003) have assessed the impact of secure communication protocols on power consumption for devices with limited resources. They presented the energy measurement and analysis for both the SSL (Secure Socket Layer) protocol and the cryptographic algorithms of a battery-powered wireless communication system. All experiments have evaluated the energy consumption given in Joules. The first experiment was performed with the symmetric algorithms DES, 3DES, IDEA, CAST, AES, RC4, RC5 and Blowfish. It showed that AES is the least power consuming algorithm and Blowfish is the one with the largest consumption. The second experiment included the hash algorithms MD2, MD4, MD5, SHA, SHA1 and HMAC. It concluded that MD2 and HMAC were more compute-intensive than the rest of the algorithms. The third experiment used asymmetric algorithms to generate and verify signatures (RSA, DSA and ECDSA). It showed that ECDSA consumes less energy. The final conclusion is that symmetric algorithms consume less energy than the asymmetric algorithms. On the other hand, hash algorithms lie at the very bottom in terms of energy consumption.

AL-Rousan et al. (2009) introduced a security scheme for exchanging information between wireless sensors with low power consumption. The scheme uses a one-way hash function and a mix of symmetric and asymmetric functions. The computational and communication overheads are calculated using the Direct Diffusion protocol. The ECC is the asymmetric cryptography algorithm chosen and RC5 is the symmetric one. The energy consumption of the scheme is calculated and compared with others, based on the execution of both algorithms in a microprocessor of 16 MHz. They concluded that their scheme consumes less energy than the public key cryptography. It is also a slightly better energy saving approach than the symmetric key cryptography.

Minaam et al. (2010)discuss power consumption for different symmetric algorithms. The experiment is carried out in a laptop with 1.5 GHz and the response variables are power consumption, encryption time, CPU process time and CPU clock cycles. Six algorithms, namely AES, DES, 3DES, RC2, RC6 and Blowfish, are used to encrypt different types of data such as audio, video and pictures (factor type of file), with different sizes (varying the size of file with levels that depends on the type of the file). They conclude that Blowfish encryption and decryption outperforms the other algorithms regarding power consumption and processing time. Another observation is that the type of file does not influence the performance of cryptography algorithms.

This paper presents five case studies describing the performance evaluation of cryptographic algorithms implemented in an embedded system. A major difference from previous works is the use of statistical analysis to ensure reproducible results. Moreover, unlike previous work, we also consider the analysis of the response time (including encryption and decryption), memory usage and power consumption for asymmetric algorithms.

3. Experimental setup

Before presenting the analysis, in this section we describe the experimental setup. We present the platform used to evaluate both symmetric and asymmetric cryptography algorithms, and the performance evaluation requirements.

3.1. Platform

The hardware platforms used in the following case studies are an embedded system and a general purpose environment. Both platforms were used to obtain a comparison of how the characteristics of the cryptography algorithm are altered in an embedded system.

The embedded system is a kit developed by Gumstix called Overo EVM pack that contains an Overo Fire computer-on-module with the ARM Cortex-A8 32-bit OMAP3530 720 MHz. It supports MicroSD memory card, Bluetooth and Wireless connections and a Chestnut43 expansion board with Ethernet, USB and serial console via mini-USB. This system and configuration was chosen because it