A Modular Test Platform for Evaluation of Security Protocols in NFC Applications

Geoffrey Ottoy[1,2], Jeroen Martens[1], Nick Saeys[1], Bart Preneel[2],
Lieven De Strycker[1,3], Jean-Pierre Goemaere[1,3], and Tom Hamelinckx[1,3]

[1] KAHO Sint-Lieven, DraMCo research group,
Gebroeders de Smetstraat 1, 9000 Gent, Belgium
{geoffrey.ottoy,dramco}@kahosl.be
http://www.dramco.be/
[2] K.U. Leuven, COSIC research group,
Kasteelpark Arenberg 10, bus 2446, 3001 Leuven-Heverlee, Belgium
http://www.esat.kuleuven.be/cosic
[3] K.U. Leuven, TELEMIC research group,
Kasteelpark Arenberg 10, bus 2444, 3001 Leuven-Heverlee, Belgium
http://www.esat.kuleuven.be/telemic/

Abstract. In this paper we present the advantages and possibilities of a modular test platform for the evaluation of security protocols in NFC applications. Furthermore, we also depict some practical implementation results of this modular system. The scope of the platform is to provide a highly modular system. Adding or removing certain functionality can be done without the need of rebuilding the entire system. Security measures in hardware as well as software can be tested and evaluated with this platform. It can serve as a basis for a broad range of security related applications, NFC being our domain of interest, but even so in other domains.

Keywords: NFC, Smart Card, FPGA, Test Platform, Security

1 Introduction

Near Field Communication (NFC) is a rapidly emerging technology that enables the setup of a wireless connection between electronic devices over a short distance, in a secure and intuitive way. This technology is gaining more and more popularity every day [1]. New developed applications, ranging from mobile payments and ticketing [2, 3] to advanced access control [4, 5], require enhanced security measures. To quickly develop hardware support for security and to test and evaluate the impact of these security measures, a test platform can be of great value.

We developed a platform that offers flexibility, modularity, reliability and reduced development time. With the system described in this paper, a wide scope of hardware and software, to support security protocols for NFC applications, can be tested and evaluated. Though primarily developed for NFC applications, the platform can be easily modified to cope with other communication technologies.

At this moment, the core components of an FPGA-based test platform are built. A kernel is running on the FPGA's embedded processor and there's the ability to run multiple threads within this kernel. Currently three system services are finished. A logging service is used in order to monitor all system and application events; another service is responsible for the NFC communication and finally a web server is used for remote connections to the system. With respect to the security side of the test platform, an AES [6, 7] cryptographic core (hardware) is implemented and has proven its reliability. Further development and integration of security and interface modules on the platform is ongoing.

In the next chapters we discuss the technical aspects of a modular test platform. First, in section 2, we provide some background about NFC and security within NFC. Then, in section 3, we talk about the advantages and the possibilities of a modular test platform and we clarify the choices made in the development of our system. In section 4, a more thorough discussion the current status of our platform is presented, whereas section 5 highlights some remarkable results. Finally, in section 6, we state some conclusions and we present our objectives for the future.

2 NFC and Security

Near Field Communication is a short range, wireless connectivity technology that allows a fast, short range, bidirectional communication between two electronic devices. This technology is based on the RFID[4] technology and works on the same communication frequency of 13.56 MHz (HF). The NFC technology is standardized under the ISO/IEC 18092 (NFCIP-1) and ISO/IEC 21481(NFCIP-2) standards and it is backward compatible with other wireless communication Protocols such as Mifare (NXP, ISO 14443-A) and Felica (Sony, ISO 14443-B).

Using NFC technology, consumers can perform contactless payments and transactions, access digital content and share data between NFC-enabled devices, all with just one simple *touch*[5]. NFC is also typically used to setup communication of other wireless protocols with a higher bandwidth, such as WiFi and Bluetooth [8].

Typical security threats for NFC, as described in [8] en [9], are Eavesdropping, Data Modification, Relay Attack or the more general Man-in-the-middle attack, even though chances for the latter are low due to the RF characteristics and short distance of NFC communication [10]. This implies that the setup of a secured channel is quite easy. Ensuring entity and data authentication further decreases the risk of most attacks. Once the secured channel is established, dedicated cryptographic algorithms should be used to further fulfill the needs of the application (e.g. a ticketing or payment application). Security algorithms can be built in hardware or software, depending on the requirements.

[4] Radio Frequency Identification

[5] NFC jargon, meaning two devices are brought into each others proximity

3 The Modular Test Platform

If we want to develop a test platform for secure NFC applications, we can state several system requirements: modularity in hardware as well as in software, support for security and several debugging and interfacing options. As an extra feature for our platform we also opted for a network connection. The main reason for this is, that a network connection (like Ethernet) is of great surplus value for a lot of applications (e.g. access control). Another advantage is that the platform can be accessed and monitored remotely (e.g. for field testing). With these requirements in mind, we can make a general description of the platform.

First of all the required security policies need to be supported by the lowest level in the design, i.e. the hardware [11]. This means that the proper countermeasures against hardware attacks need to be taken. The Side-Channel Analysis Attacks (SCA) are especially dangerous because they exploit the characteristics of the hardware design to extract the secret parameters. In [12], a number of possible attacks on hardware implementations are described. Today's secure hardware should be resistant to (Differential) Power Analysis (DPA) [13, 15], Timing Analysis (TA) [14] and Electromagnetic Attacks [15]. We will not go into detail about this, but it should be taken into account when designing the security-related hardware and software.

For easy (and fast) prototyping, modularity and expandability of the base design is required. We are designing a test platform so if we want to add or remove a certain functionality, it is preferable that this can be done without the need to redesign the entire system. By building the system out of several blocks, adding or removing functionality, in hardware as well as in software, can be done easily. In hardware we implement a bus structure (Fig. 1). The bus forms the backbone where all the other hardware blocks are attached to. Each hardware block is responsible for a designated task, e.g. memory management, interrupt handling, cryptographic operations, etc. A controller or processor will be used to control the hardware. Instead of working with a single application or thread, we choose to deploy an Operating System (OS). This further increases the flexibility and makes it more easy to add applications (in the form of threads) to the system. Furthermore, it enables the software designers to write code on a more abstract level, rather than having to know the underlying hardware.

A last topic is the need for I/O-functionality. In our case we need at least an NFC connection and an Internet connection, but several other interfaces can prove useful when using the test platform: USB or compact flash for logging, a display, I^2C, etc. It is clear that this architecture can be used in several other applications where testing of security measures is involved. A change of interface (e.g. a GPRS connection instead of NFC) can be done easily because of the modularity of the design.

4 The Platform in Detail

In figure 1, the block diagram of the modular test platform is shown. The heart of the system is an FPGA. We have opted to work with an FPGA because of its flexibility and the rapid development results. In practice, we use a Virtex II Pro Development System [6]. The Virtex II Pro FPGA has two embedded PowerPC's and enough configurable logic to implement the crypto cores, memory controllers and I/O controllers. The development board provides the necessary hardware to implement a complete system (RAM, Flash, Ethernet PHY, RS-232, etc.) and to expand it using the several I/O connectors.

By using a Hardware Description Language (HDL), we can customize the system to our needs. Every component can be described separately and added to or removed from the system when necessary. Because of the bus structure, adding hardware that has been previously described in HDL (e.g. a cryptographic core) is quite a straight-forward thing to do. As an example, we implemented an AES-128 (ECB) core.

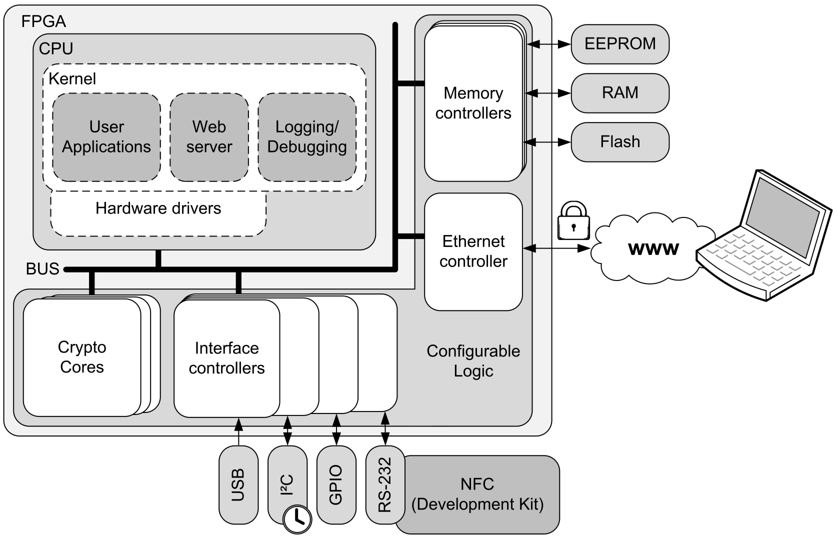

Fig. 1. Block diagram of the modular test platform

The two PowerPC's of the Virtex II Pro are hard core processors. Of course, a soft core processor (described in HDL) could be used as well. As stated in section 3, we choose to work with an operating system. On top of the embedded processor, a kernel is running. We use the Xilkernel [16] on our system. It is fully functional on our platform, outfitted with the necessary hardware drivers and provides threading support under the form of POSIX threads.

For the NFC communication, we use an NFC development kit [17]. This kit has been chosen because it supports several protocols (e.g. Mifare, Felica), but

[6] Documentation at: `http://www.xilinx.com/products/devkits/XUPV2P.htm`.

it also supports both modes of NFC operation: initiator mode and target mode. The NFC development kit communicates with the FPGA over RS-232.

Several other interface controllers have been implemented. Most notably a JTAG debug module for programming and debugging the processor over a USB connection and an I^{2}C controller. The latter is used to interface with a Real-Time Clock (RTC). Another important interface controller is the Ethernet controller (depicted separately in Fig. 1). A special thread is responsible for implementing the web server. In this way, remote connections to the test platform are supported. The web server is able to handle HTTP requests as well as manage single TCP connections. TLS [18] is used to protect the TCP connections.

In a test platform as presented here, it is a great asset if there is a service that logs all system and application events that occur. Therefore, a so called *logging service* has been written. This service makes a chronological registration of all events that occur, together with the date and time (by using the RTC). The logging records are directly written to a Compact Flash memory. The records on the Compact Flash card can be read on a PC, so the system design engineer can track which events did or did not occur, where an application crashes, how long it takes to complete a certain operation, etc.

5 Results

A first practical result is the fact that the TCP/IP stack takes in a lot of processing time. We even noticed that increasing the activity of other processes, results in timing failure of TCP/IP-related threads. Even more, the TCP/IP stack[7] intertwines with the kernel in such a manner that the POSIX API is lost. Therefore we have chosen to reserve one processor for the web server and another processor for the rest of the functionality. In this way, the web server threads are not interrupted by other processor or interrupt signals, which increases the stability of the network application. Communication between the two processors can be done by using shared memory or a hardware mailbox[8]. It is clear that dividing the functionality over the two processors, helps to control the complexity of the system. Furthermore increases the modularity of the system, because the web-part can easily be omitted now. This can be interesting when developing an off-line terminal or stand-alone platform.

As a next result we would like to present the time needed to develop an NFC application. We have chosen to implement an existing protocol, developed by the MSEC research group[9]. It uses an NFC enabled smart card and a terminal. The smart card can log on to the terminal if it's ID is in the database. As a protection against eavesdropping, the ID is encrypted using AES-128. To protect the Master Secret Key (MK) and to randomize the encrypted ID, a Session Key

[7] We use the lwIP, an open-source lightweight TCP/IP stack

[8] Both are supported by the Xilinx design tools

[9] `www.msec.be`

(SK) is generated based on two random challenges. The protocol is depicted in Fig. 2.

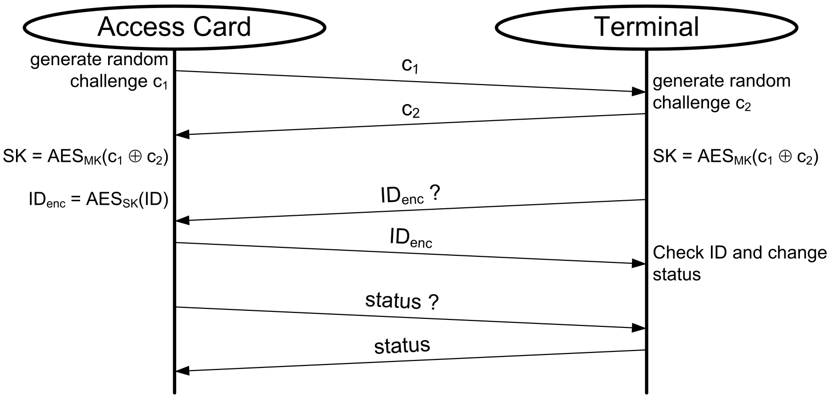

Fig. 2. Secure NFC ID interchange protocol developed by MSEC

One can see that the protocol involves some NFC data transactions and some AES operations. Because of the available API for both the NFC hardware as for the AES core and with the use of a library with NFC commands, the development of this application took a few hours (1 workday). It is clear that minor changes in the protocol can easily be implemented and evaluated in short period of time.

6 Conclusions

In this article, we described the development of a test and development platform for NFC applications. We highlighted the design choices and commented on the elaborated parts. As explained in the previous paragraphs the modularity of our platform has several advantages. From a hardware point-of-view, it is straightforward to add new cores. Even more, they can be compared with other cores or with an approach in software. Using an OS, further eases the work for the software design engineer. This makes this platform extremely fit for testing the performance of different approaches of implementing a protocol, as well as an ideal tool for quick prototyping and evaluating the feasibility of new protocols.

For the next period, we will test and implement new cryptographic cores (e.g. a cryptographic hash function) to further expand the hardware capabilities of the system. Another focus will be on the evaluation of new protocols and applications.

References

1. Madlmayr, G., Langer, J., Scharinger, J.: Managing an NFC Ecosystem, In: Mobile Business, 2008. ICMB '08. 7th International Conference on, pp.95-101 (2008).
2. Van Damme G., Wouters K., Karahan H., Preneel B.: Offline NFC Payments with Electronic Vouchers, In: Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds (MobiHeld 2009), ACM, 6 pages, (2009).
3. Juntunen A., Luukkainen S., Tuunainen V.K.: Deploying NFC Technology for Mobile Ticketing Services Identification of Critical Business Model Issues, Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR), 2010 Ninth International Conference on , pp.82-90, 13-15 June 2010
4. Peeters R., Singele D., Preneel B.: Threshold-Based Location-Aware Access Control, International Journal of Handheld Computing Research 2(2), 17 pages, (2011).
5. Madlmayr G., Langer J., Kantner C., Scharinger J.: NFC Devices: Security and Privacy, Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, pp.642-647, 4-7 March 2008
6. Daemen J., Rijmen V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag New York, Inc. Secaucus, NJ, USA (2002).
7. Federal Information Processing Standards Publication 197. Specification for the Advanced Encryption Standard (AES). 2001.
8. Breitfuß, K., Haselsteiner, E.: Security in Near Field Communication (NFC) (2006)
9. Van Damme, G., Wouters, K.: Practical Experiences with NFC Security on mobile Phones (2009)
10. ECMA International, NFC-SEC NFCIP-1 Security Services and Protocol Cryptography Standard using ECDH and AES (white paper), Dec. 2008, Online: http://www.ecma-international.org/activities/Communications/tc47-2008-089.pdf
11. Manninger M.: Smart Card Technology. In: Sklavos N., Zhang X. (eds.) Wireless Security and Cryptography - Specifications and Implementations, Cha. 13, p. 364, CRC Press (2007).
12. Örs S. B., Preneel B., Verbauwhede I.: Side-Channel Analysis Attacks on Hardware Implementations of Cryptographic Algorithms. In: Sklavos N., Zhang X. (eds.) Wireless Security and Cryptography - Specifications and Implementations, Cha. 7, pp. 213-247, CRC Press (2007).
13. Coron J.-S.: Resistance Against Differential Power Analysis For Elliptic Curve Cryptosystems, In: Koç C., Paar C. (eds.) Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, pp. 292-302 Springer-Verlag Berlin / Heidelberg (1999)
14. Kocher P. C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology (1996).
15. De Mulder E., Örs S. B., Preneel B., Verbauwhede I.: Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems, In: Computers and Electrical Engineering 33(5-6), pp. 367-382, 2007
16. Xilinx Inc., Xilkernel, Dec. 2006, Online: http://www.xilinx.com/ise/embedded/edk91i_docs/xilkernel_v3_00_a.pdf
17. NXP: UM0701-02 PN532 User Manual, Rev. 02 (2007), Online: http://www.nxp.com/documents/user_manual/141520.pdf
18. Stallings W., Transport Layer Security, In: Cryptography and Network Security - Principles and Practice, Fifth ed., Pearson Prentice Hall, Cha. 16 (2003).