

Volume Editor

Hans Weghorn  
Faculty of Mechatronics  
BA-University of Cooperative Education, Stuttgart  
Germany

Proceedings of the 4<sup>th</sup> Annual Meeting on  
Information Technology and Computer Science – ITCS, Volume 2008  
Stuttgart, Germany, February 2008  
Hans Weghorn (Ed.)

Copyright © 2008  
All rights reserved

Printed in Stuttgart, Germany

ISSN 1614-2519

# Algorithms and Protocols for Secure Embedded Networking

Axel Sikora

University of Cooperative Education, 79539 Lörrach, Germany  
sikora@ba-loerrach, <http://www.ba-loerrach.de/>

**Abstract.** Albeit security is a major challenge for embedded systems in industrial, building and consumer applications, to-day most of the networked embedded devices run at a terribly low security level. This is not mainly due to the restricted resources on embedded systems, but appears to be a question of perception and knowledge of embedded designers. For the integration of security measures into embedded systems, a variety of additional requirements must be taken into account. Those are presented and discussed in this contribution with a special focus on embedded (inter)-networking. Selected solutions of the authors are presented, i.e. for embedded SSL.

## 1 Introduction

The integration of Internet connectivity and web services into embedded systems brings many potential advantages with it [1]. It combines two enabling technologies, both with low cost and high efficiency. It is crucial to understand that all use cases can be solved with legacy technologies as well. However, the major driver to use embedded web services lies in the huge potential to save cost, as they allow the integration of applications and communication into one device, the use of well known TCP/IP (and HTTP) Internet protocols with the low-cost and seamless integration with Ethernet, the quick, easy and solid development of HTML-based graphical user interfaces that can be deployed locally or remotely, and the use of standard web browsers for remote monitoring, diagnose and control. However, a number of aspects should be regarded when using embedded systems in networks:

- Security is a major issue: Many embedded web servers may give access not only to data but may allow the control of devices, machines, and factories [5]. This results in the challenge to provide the highest possible security level at lowest cost [4].
- The risk of being attacked comes with the mere connectivity to the Internet [6] as there is a large number of systematic port scanners and as the data is transported via the public Internet. Therefore, the risk only partially depends on the application itself.
- Scalability is another issue. On the one hand, Internet protocols allow an optimum degree of scalability, as computing performance may vary from an 8 bit microcontroller to a 64 bit high end enterprise server without changing communication protocols or access mechanisms. Although the communication protocols remain the same for all these levels, the approach for security may differ significantly.

- Cryptography of course is a major means to tackle the security issue. The implementation of cryptography brings some peculiarities with it [3].
- The evolution of security threats and new means of deployment and therefore to provide possibilities to update an embedded system.

This contribution demonstrates how embedded SSL can help to overcome these issues (cf. ch. 2). It then discusses the advances in microcontroller technology (cf. ch. 3) presents one implementation (cf. ch.4).

## 2 SSL – pro’s and con’s

Secure Socket Layer (SSL)-based Virtual Private Networks (VPN) promise many advantages to embedded systems, such as:

- Embedded system can be accessed with standard software such as a web browser.
- SSL allows client authentication without storing passwords on the embedded system. This is an essential benefit since embedded systems might operate in remote, inaccessible locations, where the exchange of secured data storage is not possible.
- SSL provides a variety of encryption standards that might be implemented mandatory. System designers find a balance between security risk, capabilities of the system and the strength of the encryption.
- SSL provides client as well as server authentication.

One major drawback of SSL proves to be the symmetric key generation that is computation intense and can not be easily and efficiently sourced out in hardware. On the other hand asymmetric encryption would increase the load during communication so that the drawback of the intense key generation must be taken into account.

The effort to provide client authentication must be considered carefully. Even though the authentication of clients on the server does not add a significant effort on the design, it requires the use of a PKI to distribute certificates to authorized clients. This effort has to be considered carefully when designing a system. However, these advantages help to overcome practical problems as with [2].

## 3 Advances in microcontroller technology

In terms of long term security the use of configurable hardware such as Field Programmable Gate Array (FPGA) is the best choice. This enables a scalable hardware – software design. The use of encryption/decryption functions in hardware relieves the CPU significantly. Renowned manufacturers of FPGAs with optimized 32 bit CPU cores are Altera with its NIOS II platform [8] and Xilinx [9].

The drawback of FPGAs in comparison with standard microcontrollers is increased power consumption and higher prices per logic function (which might be alleviated through monolithic system integration), an increased complexity and the need for hardware synthesis tools.

As a result for a major number of embedded applications the choice lies still at standard microcontrollers. The following trends among standard microcontrollers can be observed:

- Increase of performance: 32 bit microcontrollers are more and more used with internal CPU frequency in the range of 50 MHz and higher.
- Increase of monolithic integration of communication peripherals such as Ethernet AC and PHYs.
- Increase of monolithic integration of hardware support for crypto algorithms such as ES, RSA, DES/3DES, or random number generation [10].
- Increasing memory capacities on the chip.

The requirements to the software in regards of the still limited resources of embedded systems and of the vast variety of platforms can be seen in:

- Efficient implementation in terms of execution speed and memory usage.
- Portability can be achieved by a smart software interface design.

Portability with the definition of software interfaces unfortunately contradicts with efficient implementations. However, only portable (and thus reusable) software can be maintained at reasonable costs. The implementation of the crypto wrapper is of high importance. It provides the means of generating target specific code. The network security layer must be able to access the same functions on different platforms. The implementation of the crypto library, however, may vary from one platform to another. If a platform provides hardware support then the crypto wrapper is configured to access those functions in order to offload the CPU. Only the functions that are not available in hardware are then run completely in software. Furthermore this enables the implementation of time critical functions in CPU optimized code such as assembly language.

## 4 Realization

### 4.1 Implementation

The emBetter TCP/IP stack [7] has been developed since many years at the institute of the author. It boasts a number of enhancements, e.g. high-speed and real-time extensions. Now, a fully standard compliant SSL-module has been developed, which allows

- Diffie-Hellman-(DH)- and RSA-based key exchange
- symmetric encryption using 3DES, AES or RC4.

The emBetterSSL is already used in various projects that require legal certification for their security mechanisms, e.g. money gambling machines and health card readers. Those are either based on Altera-NIOS [8] or Atmel-ARM9 [10]-microcontrollers.

### 4.2 Memory Management

The memory management of the emBetter suite is completely kept static. This is an absolute requirement to all microprocessor systems without a memory management unit (MMU). As a consequence, memory fragmentation does not become an issue.

In addition, care must be taken concerning the dynamic execution of program code, i.e. stack behavior. Whereas this issue can be handled with a reasonable margin, the coop-

eration with the application software tends to be more error-prone. No single solution can be provided, except extensive testing (cf. ch. 4.6).

### 4.3 Mutual Authentication

In legacy computing systems, e.g. business IT systems, servers communicate with many clients. One-side authentication seems to be sufficient for the computing devices, as authentication can be ensured by user related credentials. In addition, as servers can be physically secured, user-related information may be stored on them.

In embedded computing, machine-to-machine communication is much more relevant. Thus, devices itself must be authenticated. This should be done on a certificate level. Mutual certificate-based authentication is supported by emBetterSSL.

### 4.4 Physical Security

Legacy computing systems are separate machines, which can be centralized and secured in dedicated premises, whereas embedded systems are applied in their final system - and thus cannot be mechanically secured with major efforts to keep cost at a reasonable level. Major means to ensure the integrity of the systems are:

- use of microcontrollers with internal flash memory,
- use of internal flash memory, which can be secured against unauthorized read-out,
- use of additional physical measures to secure the microcontroller or an attached chip card.

It should be clear that all of these measures can be tempered at the attacker's expense. emBetterSSL comes with different solutions to this issue. However, they strongly depend on the target system and application.

### 4.5 Update

Running machines for years and decades also implies the necessity of updates. They must be secured so that

- the update can be accessed only after a successful mutual authentication,
- the update is stable, before the older version is removed.

emBetterSSL may use the mutually authenticated SSL-secured communication channel via http file upload. Security against loss of data or interrupt of flashing process is achieved via flash mirroring, i.e. running double program memory.

### 4.6 Testing

Albeit theoretical analyses are the fundament for a reasonable long-term stability and security, testing remains a major practical means to find weaknesses and loopholes.

emBetterSSL comes with an extended test-suite covering the functional use cases. In addition, further experience is to be collected with port-scanner like Nessus [11]. Due to non-standard implementation and no OS-related loopholes, the results are very relaxing. However, one should not forget about system specific weaknesses.

## 5 Summary

Long term security for embedded systems is closely related with the use of encryption technologies. More detailed discussion can be found in [12].

The use of standard protocols vs. proprietary protocols promises the use of standard software on remote systems and thus cost reduction on the overall system. On the other hand there is no standard hardware platform on an embedded system and the variety of implementations is big. In some cases the embedded system must provide a continuous uptime so that the update of software is only partially possible.

The use of SSL or TLS on embedded systems is of special interest since it does not require passwords stored within the embedded device. Even though SSL requires significant resources embedded systems are capable to apply it due to increased processor performance, the break through of 32bit platforms and the implementation of hardware accelerators such as crypto co-processors.

## References

1. Sikora, A, Bruegger, P.: Virtual Private Infrastructure - An Industry Consortium for Unified and Secure Web Control with Embedded Devices. In: Proc. 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 003), Lisbon, 2003.
2. Riedmueller, S., Brecht, U., Sikora, A.: IPsec for Embedded Systems, in: H. Weghorn (Ed.), Proceedings of the 2<sup>nd</sup> Annual Meeting on Information Technology and Computer Science at the BA-University of Cooperative Education, ITCS 2005, Stuttgart, 3. Mai 2005.
3. Straub, T., Sikora, A.: Cryptography on Embedded Systems, embedded world 2005 Conference, Nürnberg, S. 517-536.
4. Sikora, A.: Embedded Security for Ambient Intelligence, VDE-Kongress 2004, Berlin, 2004.
5. Byres, E., Lowe, J., Real world cyber security risks for industrial control systems, The Industrial Ethernet Book, September 2004, pp. 30-33.
6. Eichin, M.: Security Issues in Embedded Networking, Embedded Systems Conference 1993, Apr. 1994.
7. <http://www.embetter.de>
8. <http://www.altera.com>
9. <http://www.xilinx.com>
10. [http://www.atmel.com/dyn/resources/prod\\_documents/doc1593.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc1593.pdf)
11. <http://www.nessus.org/>
12. Braun, N., Sikora, A.: Design Strategies for Secure Embedded Networking, Workshop Long-term Security, Emerging Trends in Information and Communication Security (ETRICS'06), 6.- 9. Juni 2006, Freiburg, Germany.