Proceedings of 30th Indian Engineering Congress, the 21st Century Engineering: The Make in India Pathway, *December 2015*, pp. 21-26

Reliability and Security of Embedded Systems: A Study from End User Perspective

O. P. Roy* and Abhijeet Singh

Department of Electrical Engineering, NERIST, Nirjuli, Arunachal Pradesh – 791109, India; oproy61@yahoo.com, abhijeet0001@gmail.com

Abstract

The paper provides an end user perspective of reliability and security issues of embedded systems being used in 21st century. Now-a-days, embedded systems controlled devices are increasingly popular in the industry as well as among the customers. Some of the very useful domains of its use are medical sciences, aviation, rail transport (metro), driverless car, critical data transfer etc. Due to complex nature in some of the embedded system applications, it is found that the defects in the system can cause life-threatening situations, delays can create huge financial impacts, and insufficient productivity can impact entire economies. It is increasingly sought a reliable and better performance embedded system to meet objective with desired level of accuracy and embedded system should be adoptable to environmental change by using artificial intelligence. This paper starts with the basic ideas of embedded system, its classifications and important applications with an emphasis on reliability and security of the embedded systems from the point of view of end users.

Keywords: Embedded System, Message Authentication Codes (MACs), Reliability, Security, Systems on Chips (SoCs)

1. Introduction

Most of the everyday products used in 21st century in our day to day activity has a computer system embedded in it. These types of embedded systems are found in every sphere of life as from cars to cell phones, video equipment to MP3 players and dishwashers to home thermostats. Till now, these systems had some dedicated functionality. For example, an embedded system in a microwave oven may be a simple programed to turn on the magnetron for a set amount of time and to monitor if the door is open or close for security reason.

Now, embedded systems are becoming increasingly sophisticated and interconnected, both to each other and to the Internet. Unfortunately, it appears that the security implications due to complexity and connectivity have mostly been overlooked, even though ignoring security could have disastrous consequences. Since embedded systems control much of our environment, compromising systems could lead to inflict physical harm. It has already seen a malware developed to do so. For example, *Stuxnet* which reprogrammed the firmware of Iran's uranium enrichment centrifuge controllers, causing the centrifuges to self-destruct[1]. Even without physical harm, massively-exploited devices can irreparably damage the manufacturer's

* *Author for correspondence*

reputation, as was almost the case just before Microsoft announced their Trust worthy Computing initiative[2,3].

But reliability and security for these systems is an open question and could prove a more difficult long-term problem than it does today. Security issues are nothing new for embedded systems. In 2001, Peter Shipley and Simson L. Garfinkel reported an unprotected modem line to a system that controlled a high voltage power transmission line[4].

However, as more embedded systems are connected to the Internet, the potential damages from such vulnerabilities scale up dramatically. This issue is already upon us. Some hospitals use wireless IP networks for patient care equipment. Cars will inevitably have indirect Internet connections – via firewall or two – to safety – critical control systems. There have already been proposals for using wireless roadside transmitters to send real – times peed limit changes to engine control computers. There is even a proposal for passenger jets to use IP for their primary flight controls, just a few firewalls away from passengers surfing the Web[5].

2. Embedded System

Embedded systems are now the dominant form of the computer system, vastly outnumbering "traditional" computers such as PCs. However, there is a huge amount of diversity among embedded systems. Some use simple 8-bit microcontrollers with less than 1 kB of memory. Some use 64-bit multicore processors with huge memory. However, there are some salient features that embedded systems share. First, they are generally special-purpose devices, as opposed to general-purpose PCs[6]. Second, they are tightly coupled with their environment – taking input from a variety of sensors and affecting the environment through its outputs.

Many embedded systems are implemented with microcontrollers, as opposed to microprocessors. Where as microprocessors need external components to operate (such as RAM and I/O devices), microcontrollers integrate a CPU with these components onto a single chip. Pressure to make devices smaller and more capable has led to many microcontrollers implementing a huge variety of peripherals, such as audio, video, and USB interfaces into them. A single highly – capable microcontroller is sufficient to build an entire system, for this reason, such microcontrollers are often called Systems on Chips (SoCs).

An embedded system typically consists of a collection of digital programs that interact with each other and with an analog environment. Examples of embedded systems include manufacturing controllers, automotive controllers, engine controllers, avionics systems, medical devices, micro-electromechanical systems, robots, etc. As computing tasks performed by embedded devices, this makes the system more sophisticated and the need for a sound discipline for writing embedded system application specific software becomes more apparent. An embedded system consisting of sensors, actuators, plant, and control software is best viewed as a hybrid system. The major application areas of embedded systems are shown in Figure 1.

3. Reliability Concerns of Embedded Systems

While mission-critical embedded applications raise reliability concerns from the customers, unexpected or premature failures in even noncritical applications such as game boxes and portable video players can erode a manufacturer's reputation and greatly diminish widespread acceptability of new devices.

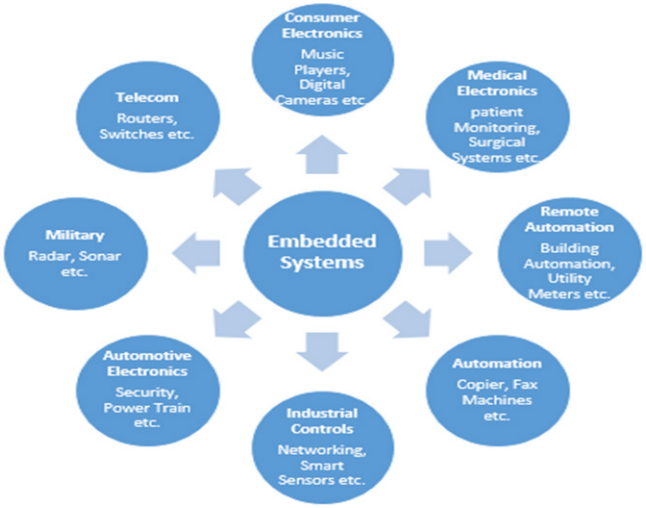

Figure 1. Major Application Areas of Embedded Systems.

The advent of more sophisticated embedded systems that support more powerful functions and the reliance on deep submicron process technologies for their fabrication, have brought reliability concerns to the forefront.

The three important categories of reliability concerns facing the design of embedded systems hardware are given as follows:

- Variability in fabricated circuit primitive parameters due to the statistical nature of manufacturing process,
- Signal integrity issues arising from internal and external noise sources, and
- Accelerated aging of the devices.

While these concerns also affect other types of computer system but embedded systems pose unique challenges as given below:

- These systems are cost sensitive and often work with limited resources such as smaller memory size or diskless designs. These constraints make it difficult to apply many traditional computer design methodologies for improving embedded systems reliability.
- Providing reliable embedded systems operation while satisfying other stringent constraints such as power consumption and real-time throughput is essential.
- Embedded systems often have reduced noise margins.

In addition to hardware reliability concerns, the increasing amount of software content in embedded systems also poses a major challenge. Many documented cases of embedded systems, failures have been attributed to software malfunction.

4. Security Risks of Embedded Systems

Internet connections expose applications to intrusions and malicious attacks. Unfortunately, security techniques developed for enterprise and desktop computing might not satisfy embedded application requirements.

Many embedded systems interact with the real world. Thus, a security breach can result in physical side effects, including property damage, personal injury, and even death. Backing out

financial transactions can repair some enterprise security breaches, but reversing a car crash isn't possible. Unlike transaction-oriented enterprise computing, embedded systems often perform periodic computations to run control loops with real-time deadlines. Speeds can easily reach 20loops per second even for mundane tasks. When a delay of only a fraction of a second can cause a loss of control loop stability, systems become vulnerable to attacks designed to disrupt system timing. Embedded systems often have no real system administrator. For Example, who's the System administrator for an Internet-connected washing machine? Who will ensure that only strong passwords are used? How is a security update handled? What if an attacker takes over the washing machine and uses it as a platform to launch distributed denial – of – service(DoS) attacks against a government agency?

There is a crisis point now with regard to the security of embedded systems, where computing is embedded into the hardware itself as with the Internet of Things. These embedded computers are riddled with vulnerabilities, and there's no good way to patch them.

It is not like what happened in the mid 1990s,when the insecurity of personal computers was reaching crisis levels. Software and operating systems were riddled with security vulnerabilities and there was no good way to patch them. Companies were trying to keep vulnerabilities secret and not releasing security updates quickly. And when updates were released, it was hard if not impossible to get users to install them. This has changed over the past twenty years, due to a combination of full disclosure publishing vulnerabilities to force companies to issue patches quicker and automatic updates.

But this time the problem is much worse, because the working environment is different. All of these devices are connected to the Internet. The computers connected to routers and modems are much more powerful than the PCs of the mid1990s and the Internet will put computers into all sorts of consumer devices. The industries producing these devices are even less capable of fixing the problem than the PC. If we don't solve this soon, this will be a security disaster as hackers figure out that it is easier to hack routers than computers. At a recent Defense Conference, it has been noticed that a researcher looked at thirty home routers and broke into half of them including some of the most popular and common brands.

To understand the problem, we need to understand the embedded systems as a whole and their market. Typically, these systems are powered by specialized computer chips made by companies such as Broadcom, Qualcomm and Marvell. These chips are cheap and the profit margins are comparatively less. Aside from price, the way the manufacturers differentiate themselves from each other is by features and bandwidth. They typically put a version of the Linux operating system onto the chips, as well as a bunch of other open source and proprietary components and drivers. They do as little engineering as possible before shipping and there is little incentive to update their "board support package" until absolutely necessary. The system manufacturers' usually original device manufacturers (ODMs) who often don't get their brand name on the finished product but choose a chip based on price and features.

5. Common Security Requirements and Counter-Measures for Embedded Systems

Embedded systems often provide critical functions

that could be sabotaged by malicious parties. When they send or receive sensitive or critical information using public networks or communications channels accessible to potential attackers, they should ideally provide basic security functions such as data confidentiality, data integrity, and user authentication.

Several functional security primitives have been proposed in the context of network security. These include various cryptographic algorithms used for encrypting and decrypting data and for checking the integrity of data. Most cryptographic algorithms fall into one of three classes given as follows:

5.1 Symmetric Ciphers

This requires the sender to use a secret key to encrypt data (the data being encrypted is often referred to as plain text) and transmit the encrypted data (usually called the cipher-text) to the receiver. On receiving the cipher-text, the receiver then uses the same secret key to decrypt it and regenerates the plaintext.

5.2 A symmetric Ciphers

This typically uses a private (secret) key for decryption, and a related public (non-secret) key for encryption. Encryption requires only the public key, which is not sufficient for decryption.

5.3 Hashing Algorithms

This converts arbitrary messages into unique fixed-length values, thereby providing unique "thumbprints" for messages for example MD5 and SHA. Hash functions are often used to construct Message Authentication Codes (MACs), such as HMAC-SHA, which additionally incorporate a key to prevent adversaries who tamper with data from avoiding detection by re-computing hashes.

Various attacks on embedded and computing systems have observed that hackers rarely take on the theoretical strength of well-designed functional security measures or cryptographic algorithms. Instead, they rely on exploiting security vulnerabilities in the software or hardware components of the implementation. It is observed that unless security is considered throughout the design cycle, embedded system implementation weaknesses can easily be exploited to bypass or weaken functional security measures. The major security requirements for embedded systems are shorn in Figure 2.

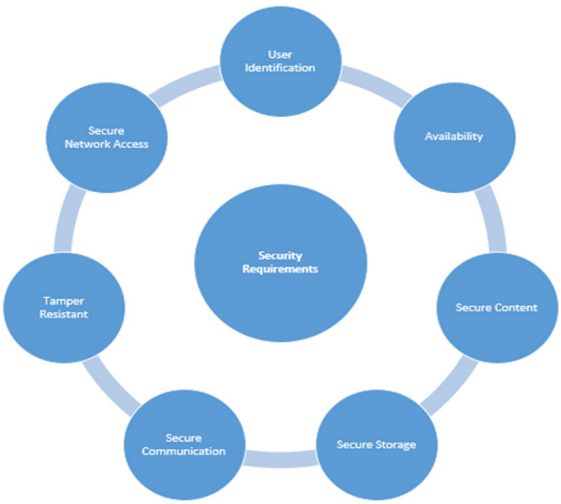

Figure 2. Common Security Requirements for Embedded Systems.

6. Conclusion

For reliable and secure embedded system design, various issues have been analysed in the context of network security and cryptography. The challenges imposed by the process of securing emerging environments or networks of embedded systems compel to take a fresh look at the problem. The good news is that unlike the problem of providing security in cyberspace, securing the application specific world of embedded systems is more likely to succeed in the near period. However, the constrained resources of embedded devices pose significant new challenges to achieving desired levels of security. It is believed that a combination of advances in architectures and design methodologies would enable to scale the next frontier of embedded system design, where in, embedded systems will be secure and reliable in every sense. To realize this goal, it should be looked beyond the basic security functions of an embedded system and provide defenses against broad classes of attacks without compromising performance, area, energy consumption, cost and usability.

7. References

1. Sanger DE. Obama Order Sped Up Wave of Cyber-attacks against Iran. The New York Times. 2012 Jun.
2. Markoff J. Stung by Security Flaws, Microsoft Makes Software Safety a Top Goal. The New York Times. 2002 Jan.
3. Mundie C, Pierre de V, Haynes P, Corwine M. 2002. Available from http://download.microsoft.com/download/a/f/2/af22fd56-7f19-47aa-8167-4b1d73c-d3c57/twc_mundie.doc.
4. Shipley P, Garfinkel SL. An Analysis of Dial-Up Modems and Vulnerabilities. 2001. Available from www.dis.org/filez/Wardial_ShipleyGarfinkel.pdf
5. Aeronautical Radio Inc. Draft 1 of Project Paper 664, Aircraft Data Networks, Part 5. Network Interconnection Devices. 2001 May. AEEC Letter 01-112/SAI/742
6. Kocher P, Lee R, Mc Graw G, Raghunathan A, Ravi S. Security as a New Dimension in Embedded System Design, DAC 2004. 2004 Jun 7–11.