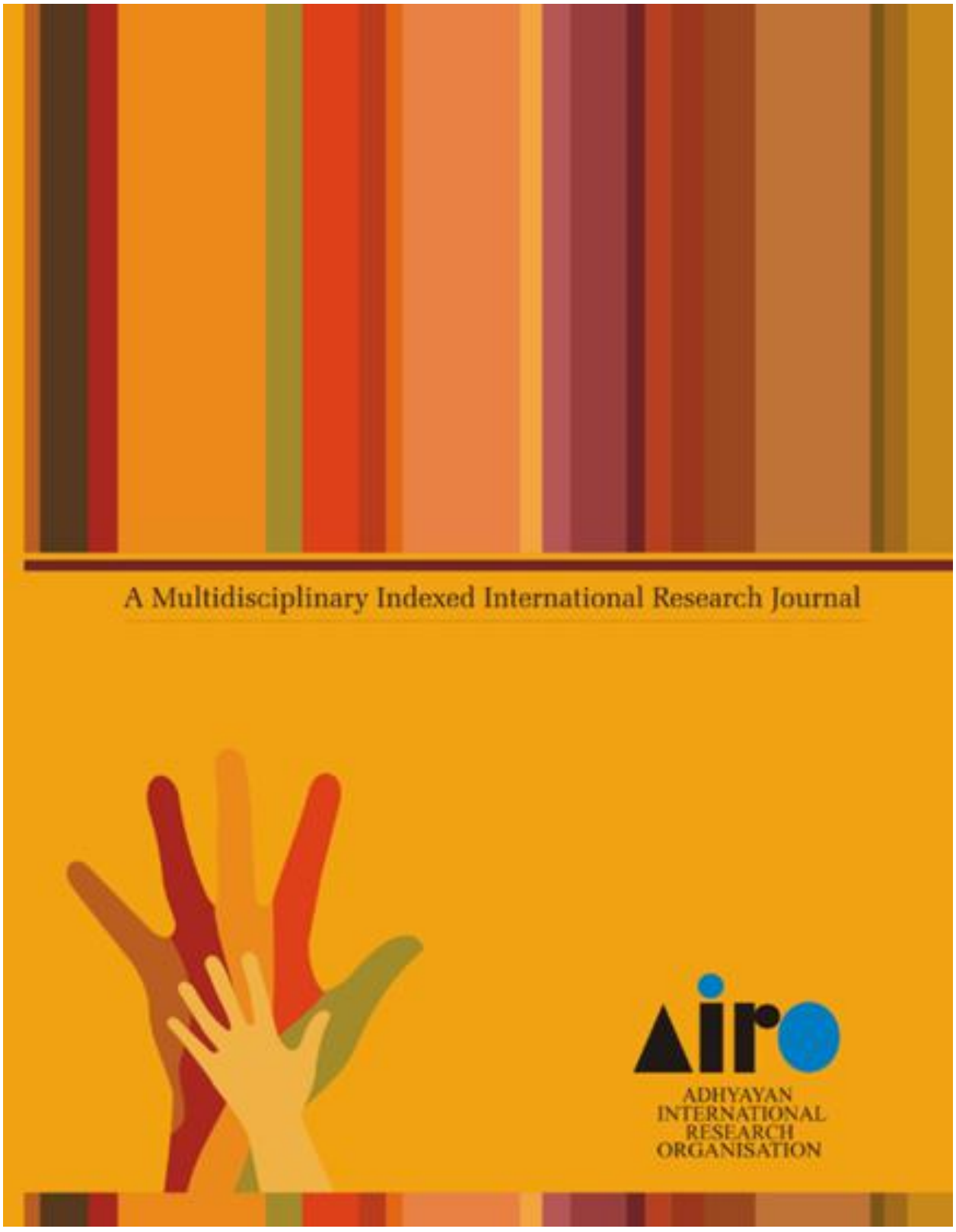
Airo International Research Journal

Volume XIII, ISSN: 2320-3714
November, 2017
Impact Factor 0.75 to 3.19

UGC Approval Number 63012



Airo International Research Journal

Volume XIII, ISSN: 2320-3714
November, 2017
Impact Factor 0.75 to 3.19

UGC Approval Number 63012

A STUDY OF EMBEDDED SYSTEM SAFETY AND ITS ALGORITHMS DESIGN CHALLENGES

Acharya D. Arun Chandra
Research Scholar, OPJS University, Churu Rajasthan
Dr. Amit Jain
Assistant Professor, OPJS University, Churu Rajasthan

ABSTRACT

Security is an important aspect of embedded system design. The characteristics of embedded systems give rise to a number of novel vulnerabilities. A variety of different solutions are being developed to address these security problems. In this paper, we provide a brief overview of important research topics in this domain. The paper discusses the hardware and software security requirements in an embedded device that are involved in the transfer of secure digital data. The paper gives an overview on the security processes like encryption/decryption, key agreement, digital signatures and digital certificates that are used to achieve data protection during data transfer. The paper also discusses the security requirements in the device to prevent possible physical attacks to expose the secure data such as secret keys from the device. The paper also briefs on the security enforced in a device by the use of proprietary security technology and also discusses the security measures taken during the production of the device. The central task of HSCD is hardware/software partitioning which is concerned with deciding which functions are to be implemented in Hardware (HW) and which ones in Software (SW). It aims at finding an optimal trade-off between conflicting requirements on area and execution time. The problem of partitioning is also encountered in circuit layout. The layout may be generated automatically using placement and routing algorithms.

KEYWORDS: *Security, embedded system design, hardware and software security, device, technology, routing algorithms.*

INTRODUCTION

The embedded or handheld devices are getting increasingly connected and are more and more involved in network communications. The users of these devices are now able to execute almost all the network/internet applications that run in a

PC on these devices. These devices are also increasingly involved in transfer of secure data through public networks that needs protection from unauthorized access and thus the security requirements in embedded devices have become critical. The secure data falls in different categories requiring different levels of security. According to whose interest the protection of the data is, the secure data can be classified as two: the user's private data and the user restricted data. The users private data are those data which when its security is compromised impacts directly on the user. A simple example of compromising such security is having access to a user's internet banking password. But in case of user restricted data, it's not the user but the content (data) provider who suffers direct loss on compromising the security of that data. The examples of such data are digital multimedia content such as copyrighted digital photos, audio and video contents. The secure data not only requires protection during data transfer but also while handling the data at the end user devices. Vulnerability at the end user device, like easy access to the secret keys that are used to encrypt or decrypt the data, can easily turn down the entire security measures. Today, an increasing number of embedded systems need to deal with security in one form or another—from low-end systems such as wireless handsets, networked sensors, and smart cards, to high-end systems such as network routers, gateways, firewalls, and storage and web servers. Technological advances that have spurred the development of these electronic systems have also ushered in seemingly parallel trends in the sophistication of attacks they face. It has been observed that the cost of insecurity in electronic systems can be very high.

The protocol involved for the secure transmission of either of the above mentioned contents through a public network uses more or less the same techniques but the handling of the user restricted data at the user's end involves much more care as the content is protected from the user itself! Thus an embedded device must implement methods or protocol for secure data transfer and also should implement security methods to defeat attempts of unauthorized access of secure data from the device. The security needs for an embedded device thus can be classified into two:

- Security needs for data transfer and
- Security needs within the device

REVIEW OF LITERATURE

In (B. Schneier 2014,), the uniqueness of embedded systems security and possible countermeasures to software and hardware attacks are elaborated. Various attack scenarios are also discussed. Vulnerabilities in general computer and IT systems are studied in (D. Kleidermacher and M. Kleidermacher 2012). An empirical study focusing on embedded systems vulnerability is included in (D. N. Serpanos 2013).

For attack taxonomy for cyber-physical systems, (G. Hernandez 2014) provided taxonomy of cyber-attacks on Supervisory Control and Data Acquisition (SCADA) systems. Cyber-attacks are classified according to their targets: hardware, software and the communication stack. The attacks on software are grouped into exploitation of embedded operating systems without privilege separation, buffer overflow, and SQL injection. The attacks on communication stack are classified into network, transport, and application layer, as well as the implementation of the protocols. In their identification of potential attacks on avionics embedded systems, (H. Holm 2013) provided a categorization of attacks against onboard aerospace systems. Attacks are categorized into two major classes: attacks against core functions and against fault-tolerance mechanisms. For each subcategory, the authors provide examples and emphasize the impact of such attacks. (L. Bilge and T. Dumitras 2012) placed their emphasis on how cyber-attacks might influence the physical space and proposed a taxonomy that could be used to categorize cross-domain attacks as well. Their proposed taxonomy has six dimensions, organized in three groups, i.e. targets, effects, and attacks.

SECURITY REQUIREMENTS OF EMBEDDED SYSTEMS

Embedded systems often provide critical functions that could be sabotaged by malicious entities. Before discussing the common security requirements of embedded systems, it is important to note that there are many entities involved in a typical embedded system design, manufacturing, and usage chain. Security requirements vary depending on whose perspective we consider. For example, let us consider a state-of-the-art cellular handset that is capable of wireless voice, multimedia, and data communications. Figure 1 illustrates security requirements from the viewpoint of the provider of HW/SW components inside the cell phone (e.g., baseband processor, operating system), the cell phone manufacturer, the cellular service provider, the application service provider (e.g., mobile banking service), the content provider (e.g., music or video), and the end user of the cell phone. The end user's primary concerns may include the security of personal data stored and communicated by the cell phone, while the content provider's primary concern may be copy protection of the multimedia content delivered to the cell phone, and the cell phone manufacturer might additionally be concerned with the secrecy of proprietary firmware that resides within the cell phone. For each of these cases, the set of untrusted (potentially malicious) entities can also vary. For example, from the perspective of the content provider, the end user of the cell phone may be an untrusted entity. While this study outlines broad security requirements typical of embedded systems, the security model for each embedded system will dictate the combination of requirements that apply.

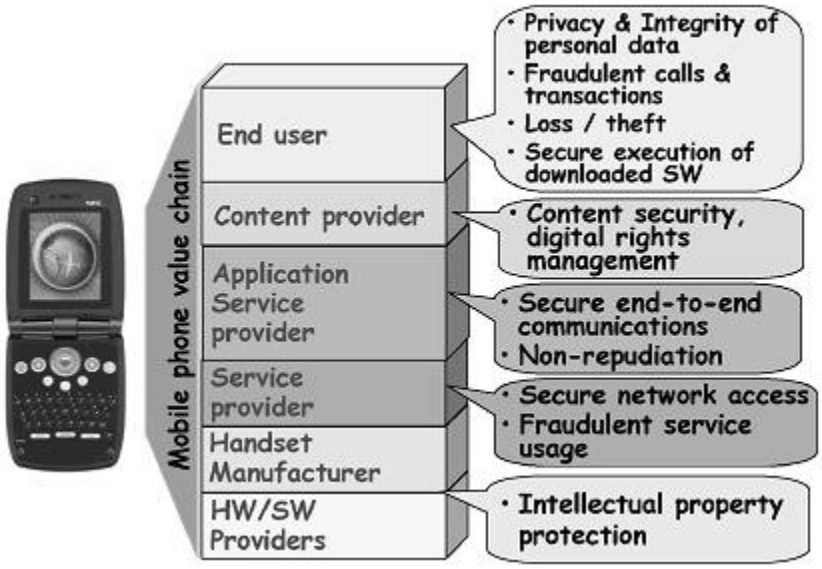


Fig. 1. Security requirements for a cell phone

Figure 2 lists the typical security requirements seen across a wide range of embedded systems, which are described as follows:

- User identification refers to the process of validating users before allowing them to use the system.
- Secure network access provides a network connection or service access only if the device is authorized.
- Secure communications functions include authenticating communicating peers, ensuring confidentiality and integrity of communicated data, preventing repudiation of a communication transaction, and protecting the identity of communicating entities.
- Secure storage mandates confidentiality and integrity of sensitive information stored in the system.
- Content security enforces the usage restrictions of the digital content stored or accessed by the system.
- Availability ensures that the system can perform its intended function and service legitimate users at all times, without being disrupted by denial-of service attacks.

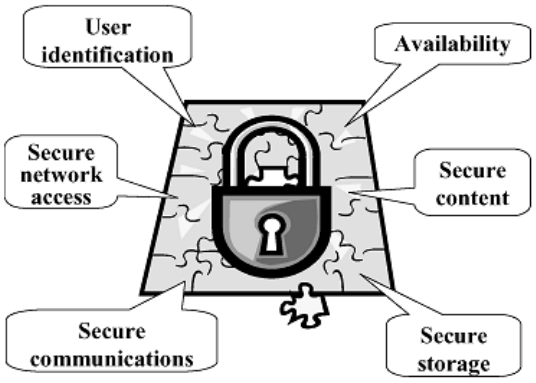


Fig. 2: Common security requirements of embedded systems

SECURITY NEEDS WITHIN THE DEVICE

Whether it is the private-key of any public-key algorithm as discussed or it is any previously negotiated shared secret between the devices, the security of data transferred depends in the secrecy of these keys. To enforce additional security, some cryptographic algorithms may also specify a set of constant values that should not be disclosed from the device. These secret keys and secret values stored in the device that requires protection from unauthorized exposure are referred as 'secret keys' in this document. The secret keys are stored inside the device, some even for the lifetime of the device. Hardware and software security measures implemented in the device must defeat any attempts of unauthorized access to retrieve these secret keys (A. Costin and A. Francillon 2012). Also, there are data such as the Root CA Certificate in the device that can be disclosed but should be prevented from unauthorized modification. If Root CA certificate can be modified, then the attacker can make the device to accept any certificate by substituting a fake root CA certificate and thus defeating the purpose certificate and secured communication. It is therefore also important that the security in the device is such that the data such as Root CA Certificates in the device is not subjected to unauthorized modification. The level of security within the device varies depending on the nature of the protected content. The need for device security is more in the case of device handling user restricted data like copy-protected* video than in the case of user's private data like personal files or bank transactions. This is mainly because, in the case of user's private data since the user will suffer the direct loss on compromising such data, he/she will be responsible for restricting the physical access to the secret keys and other secured contents stored in the device. Also, the general implementation of secure data transfer protocols recommends a unique secret key for each device. Therefore if the hardware security of any of the device is compromised, it doesn't affect the security of other device in the network. But

in the case of user restricted data, compromising the secret key of a single device results in the compromise of the security of all the copy-protected content handled by that device. One vulnerable device can thus results in helping an unauthorized device to access the copy protected content, decrypt it and distribute countless copy of the copy protected content. The example of prototype SoC to discuss the hardware and software support required to enforce the security within the device and thereby defeating the physical attack that compromises the security of the device.

SECURE EMBEDDED SYSTEM DESIGN CHALLENGES

Designers of a large and increasing number of embedded systems need to support various security solutions in order to deal with one or more of the security requirements described earlier. These requirements present significant bottlenecks during the embedded system design process, which are briefly described below:

Processing Gap: Existing embedded system architectures are not capable of keeping up with the computational demands of security processing, due to increasing data rates and complexity of security protocols. These shortcomings are most felt in systems that need to process very high data rates or a large number of transactions (e.g., network routers, firewalls, and web servers), and in systems with modest processing and memory resources (e.g., PDAs, wireless handsets, and smartcards). In this paper, we will examine the two sides of the processing gap issue (requirements and availability) and study various solutions proposed to address this mismatch.

Battery Gap: The energy consumption overheads of supporting security on battery-constrained embedded systems are very high. Slow growth rates in battery capacities (5–8% per year) are easily outpaced by the increasing energy requirements of security processing, leading to a battery gap. Various studies (R. Santamarta 2014) show that the widening battery gap would require designers to make energy-aware design choices (such as optimized security protocols, custom security hardware, and so on) for security.

Flexibility: An embedded system is often required to execute multiple and diverse security protocols and standards in order to support (i) multiple security objectives (e.g., secure communications, DRM, and so on), (ii) interoperability in different environments (e.g., a handset that needs to work in both 3G cellular and wireless LAN environments), and (iii) security processing in different layers of the network protocol stack (e.g., a wireless LAN enabled PDA that needs to connect to a virtual private network, and support secure web browsing may need to execute WEP, IPSec, and SSL). Furthermore, with security protocols being constantly targeted by hackers, it is not surprising that they keep continuously evolving. It is, therefore, desirable to allow

the security architecture to be flexible (programmable) enough to adapt easily to changing requirements. However, flexibility may also make it more difficult to gain assurance of a design's security

Tamper Resistance: Attacks due to malicious software such as viruses and trojan horses are the most common threats to any embedded system that is capable of executing downloaded applications. These attacks can exploit vulnerabilities in the operating system (OS) or application software, procure access to system internals, and disrupt its normal functioning. Because these attacks manipulate sensitive data or processes (integrity attacks), disclose confidential information (privacy attacks), and/or deny access to system resources (availability attacks), it is necessary to develop and deploy various HW/SW countermeasures against these attacks.

Assurance Gap: It is well known that truly reliable systems are much more difficult to build than those that merely work most of the time. Reliable systems must be able to handle the wide range of situations that may occur by chance. Secure systems face an even greater challenge: they must continue to operate reliably despite attacks from intelligent adversaries who intentionally seek out undesirable failure modes. As systems become more complicated, there are inevitably more possible failure modes that need to be addressed. Increases in embedded system complexity are making it more and more difficult for embedded system designers to be confident that they have not overlooked a serious weakness.

Cost: One of the fundamental factors that influence the security architecture of an embedded system is cost. To understand the implications of a security related design choice on the overall system cost, consider the decision of incorporating physical security mechanisms in a single-chip cryptographic module. The Federal Information Processing Standard (FIPS 140-2) [FIPS] specifies four increasing levels of physical (as well as other) security requirements that can be satisfied by a secure system. Thus, we can choose to provide increasing levels of security using increasingly advanced measures, albeit at higher system costs, design effort, and design time. It is the designer's responsibility to balance the security requirements of an embedded system against the cost of implementing the corresponding security measures.

CONCLUSION

Security is critical to enabling a wide range of applications involving embedded systems. While some aspects of security have been addressed in the context of traditional general-purpose computing systems, embedded systems usher in many new challenges. This paper highlighted the security-related problems faced by designers of embedded systems, and outlined recent technological developments and innovations to address them. Several issues, however, remain open at the intersection of security

and embedded system design. The more the hardware security measures implemented in a device to protect its secret keys and other secure data, the more costly the device will be. Thus the hardware security measures implemented in the device are a trade of between the cost of implementation and the cost of the data protected. Achieving a cost effective yet foolproof method to protect the secret keys and secure data within the device will be a boon to the owner of the contents that needs security, especially to the content provider of copy-protected digital contents.

REFERENCES:

[1] B. Schneier, "Security risks of embedded systems," https: //www.schneier.com/blog/archives/2 014/01/security risks 9.html, January 2014.

[2] D. Kleidermacher and M. Kleidermacher, Embedded systems security: practical methods for safe and secure software and systems development. Elsevier, 2012.

[3] D. N. Serpanos and A. G. Voyiatzis, "Security challenges in embedded systems," ACM Transactions on Embedded Computing Systems (TECS), vol. 12, no. 1s, p. 66, 2013.

[4] H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of systemlevel vulnerability metrics through actual attacks," Dependable and Secure Computing, IEEE Tran. on, vol. 9, no. 6, pp. 825–837, 2012.

[5] L. Bilge and T. Dumitras, "Before we knew it: an empirical study of zero-day attacks in the real world," in Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012, pp. 833–844.

[6] R. Santamarta, SATCOM Terminals: Hacking by Air, Sea, and Land, IOActive, Inc., 2014. [Online]. Available: https://www.defcon.org/images/defc on-22/dc-22-presentations/Cerrudo/ DEFCON-22-Cesar-Cerrudo-Hacking-Traffic-Control-Systems-UPDATED. pdf

[7] A. Costin and A. Francillon, "Ghost in the air(traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices," 2012. [Online]. Available: http://s3.eurecom.fr/docs/bh12us costin.pdf

[8] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, "Smart nest thermostat: A smart syp in your home," ser. Black Hat, 2014

[9] A. Costin and A. Francillon, "Short paper: A dangerous 'pyrotechnic composition': Fireworks, embedded wireless and insecurity-by-design," in Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks, ser. WiSec '14. New York, NY, USA: ACM, 2014, pp. 57–62.

[10] S. Hanna, R. Rolles, A. Molina-Markham, P. Poosankam, K. Fu, and

D. Song, “Take two software updates and see me in the morning: The case for software security evaluations of medical devices,” in Proceedings of the 2Nd USENIX Conference on Health Security and Privacy, ser. HealthSec’11. Berkeley, CA, USA: USENIX Association, 2011, pp. 6–6.