Securing Mobile Appliances: New Challenges for the System Designer

Anand Raghunathan†, Srivaths Ravi†, Sunil Hattangady‡, and Jean-Jacques Quisquater§
† NEC Laboratories America, Princeton, NJ, USA
‡ Texas Instruments Inc., Dallas, TX, USA
§Universite catholique de Louvain, Louvain-la-Neuve, Belgium

ABSTRACT

As intelligent electronic systems pervade all aspects of our lives, capturing, storing, and communicating a wide range of sensitive and personal data, security is emerging as a critical concern that must be addressed in order to enable several current and future applications. Mobile appliances, which will play a critical role in enabling the visions of ubiquitous computing and communications, and ambient intelligence, are perhaps the most challenging to secure - they often rely on a public medium for (wireless) communications, are easily lost or stolen due to their small form factors and mobility, and are highly constrained in cost and size, as well as computing and battery resources.

This paper presents an introduction to security concerns in mobile appliances, and translates them into challenges that confront system architects, HW engineers, and SW developers, including how to bridge the processing and battery gaps, efficient tamper-proofing of devices, content protection, *etc*. Recent innovations and emerging commercial technologies that address these issues are also highlighted. We envision that, for a large class of embedded systems, security considerations will pervade all aspects of system design, driving innovations in system architecture, software, circuits, and design methodologies.

1. INTRODUCTION

Mobile appliances, including cell phones, PDAs, and smart cards, account for a large segment of the electronics and semiconductor industries. Due to their convenience and ubiquity, it is widely accepted that such mobile appliances will evolve into "personal trusted devices" that pack our identity and purchasing power, benefiting various aspects of our daily lives [1, 2]. On the other hand, mobile appliances are also likely to play an important role in enabling the vision of an intelligent ambience, by collecting and communicating various personal habits and preferences, and enabling our environments to sense and react to us.

Due, in part, to the aforementioned trends, the usage of mobile appliances will frequently involve the storage of, access to, and communication of sensitive information, making security a serious concern. Indeed, the success and adoption of several emerging applications and services are predicated on the ability of mobile appliance manufacturers and service providers to ensure adequate security and gain the trust and confidence of consumers and other parties involved. For example, 2.5G and 3G wireless applications, including mobile commerce (m-commerce), multimedia messaging, mobile content delivery, and mobile office deployment, require high levels of security. In fact, security is cited as the single largest concern among surveys of prospective m-commerce users [3].

Thanks to the evolution of the Internet, information and communications security has already gained significant attention [4, 5, 6, 7]. While the knowledge and experience gained from the wired Internet, including cryptographic algorithms, security protocols, and standards, give us a head start in the quest to secure mobile appliances, there are several challenges unique to mobile appliances that must still be addressed.

- Mobile appliances often use a public transmission medium for (wireless) communication, which implies that the physical signal is easily accessible to eavesdroppers and hackers. Wireless security is a challenging problem, perhaps even more so than wired security in many respects [8, 9, 10, 11, 12], that must be addressed by many mobile appliances.
- Unlike desktop computers, which operate in physically secure environments and have fixed or limited "points of access", mobile appliances are free to operate in far more hostile environments, due to the potentially unlimited points of access, and over a wide range of bearer technologies such as cellular networks (*e.g.*, GSM / GPRS), wireless local area networks (*e.g.*, 802.11a/b), and personal area networks (*e.g.*, Bluetooth).
- Mobile appliances are quite vulnerable to theft, loss, and corruptibility. Security solutions for mobile appliances must, therefore, provide for security under these challenging scenarios.
- Constraints on cost and weight, and the need to operate mobile appliances off batteries, imply that they are quite constrained in their processing capabilities and energy supplies. The processing and energy overhead required to provide sufficient security can be significant, and overwhelm the modest capabilities of mobile appliances [12, 13].

The challenges of securing mobile appliances can be adequately addressed only through measures that span virtually every aspect of their design — hardware circuits and micro-architecture, system architecture, system and application software, and design methodologies. This paper introduces the new challenges that security poses to mobile appliance designers, and surveys technologies that can be used to address them. Despite significant recent interest and notable innovations in this area, many challenges remain that will require further attention and awareness of security among hardware, software, and system designers.

2. BACKGROUND: SECURITY CONCERNS IN MOBILE APPLIANCES

The role of security mechanisms is to ensure the privacy and integrity of data, and the authenticity of parties involved in a transaction. In addition, it is also desirable to provide functionality such as non-repudiation, copy protection, preventing denial-of-service attacks, filtering of viruses and malicious code, and in some cases, anonymous communication [6, 7].

Figure 1 illustrates some of the major security concerns from the perspective of a mobile appliance.

- *User identification* attempts to ensure that only authorized entities can use the appliance.
- *Secure storage* addresses the security of sensitive information such as passwords, PINs, keys, certificates, *etc.*, that may reside in secondary storage (*e. g.*, flash memory) of the mobile appliance.
- A *secure software execution environment* is necessary to ensure that attacks from malicious software such as viruses or trojan horses are prevented.
- A *tamper-resistant system implementation* is required to ensure security of the hardware implementation from various physical and electrical attacks.
- *Secure network access* ensures that only authorized devices can connect to a network or service.
- *Secure data communications* considers the privacy and integrity of data communicated to/from the mobile appliance.
- *Content security* refers to the problem of ensuring that any content that is downloaded or stored in the appliance is used in accordance with the terms set forth by the content provider (*e. g.*, read only, no copying, *etc.*).

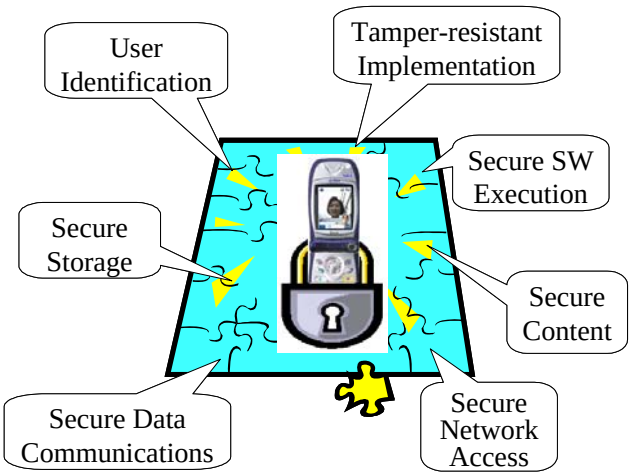

Figure 1: Security concerns in a mobile appliance

We illustrate mobile appliance security concerns through the example of secure data communications on a wireless handset (*e.g.*, a cell phone or PDA). Wireless data communications can be secured by employing security protocols that are added to various layers of the network protocol stack, or within the application itself. Security protocols utilize cryptographic algorithms (asymmetric or public-key ciphers, symmetric or private-key ciphers, hashing functions, *etc.*) as building blocks in a suitable manner to achieve the desired objectives (peer authentication, privacy, data integrity, *etc.*).

Different security protocols have been developed and employed in cellular technologies such as CDPD [14] and GSM [15, 16], wireless local area network (WLAN) technologies such as IEEE 802.11 [17], and wireless personal area network technologies such as Bluetooth [18]. Many of these protocols address only network access domain security, *i.e.*, securing the link between a wireless client and the access point, base station, or gateway. Several studies have shown that the level of security provided by most of the above security protocols is insufficient, and that they can be easily broken or compromised by serious hackers [19, 20, 21, 22, 23, 24, 25]. While some of these drawbacks are being addressed in newer wireless standards such as 3GPP [26, 27] and 802.11 enhancements [17], it is generally accepted that they need to be complemented through the use of security mechanisms at higher protocol layers.

With the push to bring wired Internet data and applications to wireless handsets, and to enhance the wireless data experience, conventional Internet protocols are being increasingly used in wireless networks, by overlaying them on top of the underlying "bearer" technologies. In the wired Internet, the most popular approach is to use security protocols at the network or IP layer (IPSec), and at the transport or TCP layer (TLS/SSL) [6, 7].

Recognizing the need to provide protocols optimized for the wireless environment, the Wireless Application Protocol (WAP) standard [28] defines protocols that can be overlaid on top of existing wireless bearer technologies, such as GSM, CDPD, CDMA, *etc.* In the WAP architecture, wireless handsets run the WAP protocol stack, and a WAP gateway translates traffic to/from the wireless handset to conventional Internet protocols (HTTP/TCP/IP), thereby facilitating inter-working with existing (wired) Internet servers.

The WAP architecture allows for the use of security schemes at multiple layers of the protocol stack.

- Security protocols provided in the bearer technologies (such as CDPD, GSM, CDMA, *etc.*) may be used to provide network access domain security, including user authentication to the serving network, as well as a basic level of confidentiality and integrity over the wireless link.
- The WAP protocol stack includes a transport-layer security protocol, called WTLS, which provides higher layer protocols and applications with a secure transport service interface and secure connection management functions.
- Finally, specific applications may decide to directly employ security mechanisms instead of, or in addition to, the aforementioned options (through an application-level security protocol such as SET [6], or to provide additional functionality, such as non-repudiation, that is not provided in the transport-layer security protocol).

The need to support security protocols and mechanisms, such as those described above, translates to various challenges in the design of the mobile appliance. The rest of the paper focuses specifically on these system design challenges and solutions that address them.

3. SECURE MOBILE APPLIANCE DESIGN CHALLENGES

In this section, we describe the various challenges and considerations involved in supporting security on mobile appliances. Section 3.1 first discusses the diversity and evolutionary nature of security protocols and cryptographic algorithms, and the consequent need for flexibility in the security processing architecture of a mobile appliance. Section 3.2 analyzes the computational requirements of security processing, while Section 3.3 examines the impact of security processing on battery life. Finally, Section 3.4 tackles the important problem of securing the system implementation and the resultant need for building in attack resistance features.

3.1 Flexibility

A fundamental requirement of a mobile appliance is the ability to cater to a wide variety of security protocol standards in order to facilitate interoperability in different environments. For example, an appliance that needs to work in both 3G cellular and wireless LAN environments would need to execute security algorithms specified by 3GPP [26, 27] as well as the Wired Equivalent Privacy (WEP) algorithm specified by the 802.11 standard [17]. Additionally, a device is often required to support distinct security processing standards at different layers of the network protocol stack. For example, a wireless LAN enabled PDA that supports secure web browsing may need to execute both WEP (Link Layer) and SSL (Transport Layer), while the same PDA, if required to connect to a virtual private network (VPN), may additionally need to support IPSec (Network Layer).

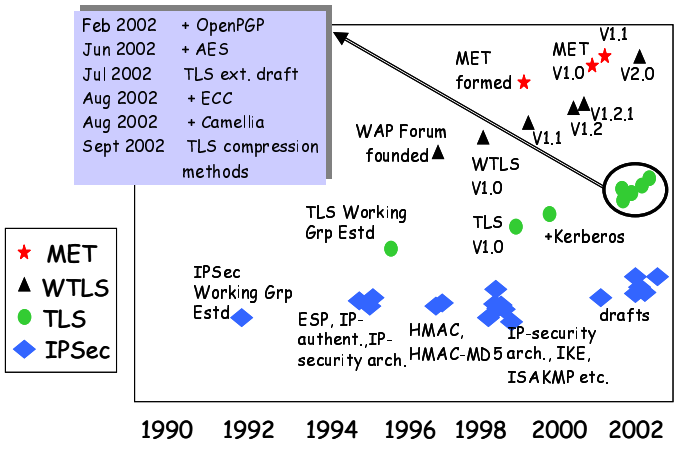

Figure 2: Evolution of security protocols

Complicating the above picture is the specification of any security protocol standard, which typically allows for the usage of a wide range of cryptographic algorithms. To illustrate this scenario, let us consider the SSL protocol [29], which supports the use of different ciphers for its operations (authenticating the server and client, transmitting certificates, establishing session keys, *etc.*). For key exchange, cryptographic algorithms such as RSA and KEA are possible choices. For symmetric encryption, an RSA key exchange based SSL cipher suite would need to support 3-DES, RC4, RC2 or DES, along with the appropriate message authentication algorithm (SHA-1 or MD5). Since the mobile appliance may have to communicate with a server/client that uses a specific combination of cipher suite and key exchange algorithm, it is desirable to support all the allowed combinations so as to inter-operate with the widest possible range of peers.

Finally, security protocols are not only diverse but also are continuously evolving. This has been and is still witnessed in the wired domain, wherein, protocol standards are revised to enable new security services, add new cryptographic algorithms or drop weaker ciphers. Figure 2, for example, tracks

the evolution of popular security protocols in the wired domain IPSec [30] and SSL/TLS [31]. We can see that even a well-established protocol such as TLS is subject to constant modifications (*e.g.*, in June 2002, TLS was revised to accommodate the proposed replacement to the DES standard, AES).

The evolutionary trend is much more pronounced today in the wireless domain, where security protocols can be termed to be still in their infancy. Figure 2 also outlines the evolution of the wireless security protocols, WTLS [32] and MET [33]. Many of the security protocols used in the wireless domain are adaptations of the wired security protocols. For example, WTLS bears a close resemblance to the SSL/TLS standards. However, it is anticipated that future security protocols would be specifically tailored from scratch for the wireless environment. This presents a formidable challenge to the design of a security processing architecture, since flexibility and ease-of-adaptation to new standards become equally important design considerations as traditional objectives such as power, performance, *etc.*

3.2 Computational Requirements of Security Processing

The computational power available in a mobile appliance is significantly limited compared to the processing capabilities of a desktop computer. To understand the difference, compare the MIPS ratings of a 2.6 GHz Pentium 4 processor powered desktop and a state-of-the-art PDA featuring the Intel StrongARM 1100 processor. While the former is capable of delivering roughly 2890 MIPS, the latter can supply only 235 MIPS at its fastest (206MHz) [34]. The above scenario actually represents the higher end of the embedded processor spectrum. At the other end of this spectrum, we have the Motorola 68EC000 DragonBall processor core used in Palm OS products rated at approximately 2.7 MIPS [35], while the ARM7/ARM9 central CPU used in cell phones typically deliver 15 to 20 MIPS processing power running at speeds of 30 to 40 MHz.

While power dissipation and size requirements of mobile appliances restrict the processor architectures and, hence, their MIPS ratings, the level of security desired in data communicated by the mobile appliance remains unchanged or even increases! As a consequence, the computational requirements of standard security protocols tend to be significantly higher than the embedded processor capabilities [12, 13]. Data presented in [12] reveal that the total processing requirements for a security protocol that uses 3DES for encryption/decryption and SHA for message authentication at 10 Mbps (current and emerging data rates for wireless LAN technologies are said to be in the range of 2-60 Mbps) is around 651.3 MIPS. A similar trend has also been observed for RSA based connection set-ups performed in client/server handshake phase of the SSL protocol. A 235 MIPS embedded processor can be used to establish connection latencies at 0.5*sec* or 1*sec*, but not at 0.1*sec*. Thus, there exists a clear mismatch between the security processing requirements and the available processor capabilities, even if the workload of the appliance is assumed to be completely dominated by security processing. In other words, this mismatch is likely to be worse in reality since the processor is typically burdened by a workload that also includes other application software, network protocol and operating system execution.

The effective computational requirements of a typical security protocol that performs RSA based connection set-up, 3DES-based data encryption and SHA-based integrity are shown in Figure 3 for various combinations of connection latencies and data rates. The processing capabilities of an embedded processor can be represented as a plane in the 3-dimensional space (see, for example, the 300 MIPS plane). Clearly, the processing requirements above the plane (and, hence, the corresponding combinations of connection latencies and data rates) can not be supplied by the embedded processor, leading to a *wireless security processing gap*. While embedded processor performance can be expected to increase due to improvements in fabrication technologies and innovations in processor architecture, the increase in data rates (due to advances in wireless communication technologies), and the use of stronger cryptographic algorithms (to stay beyond the extending reach of malicious entities) threaten to further widen the wireless security processing gap.

3.3 Battery Life

The computational requirements of security protocols stemming from the inherent complexity of cryptographic algorithms suggest that the energy consumption of these algorithms will be high. For battery powered mobile appliances, the energy drawn from the battery directly impacts the system's battery life, and, consequently, the duration and extent of its mobility and its overall utility. To illustrate the impact of security processing on battery life, consider the following case-study based on data taken from [36]. The energy consumed when a sensor node containing a Motorola DragonBall

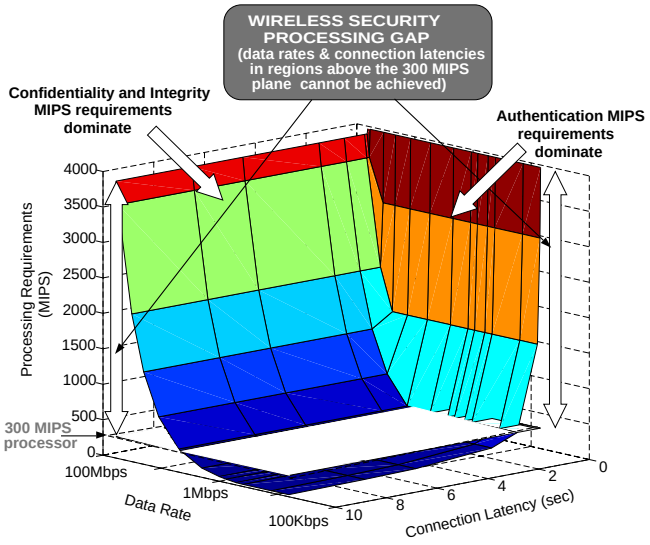

Figure 3: The wireless security processing gap (Source: [12])

MC68328 processor transmits and receives data at a data rate of 10 Kbps are $21.5mJ/KB$ and $14.3mJ/KB$, respectively. When the sensor node operates in the secure mode, it performs RSA-based data encryption as a part of its security protocol and incurs an additional energy overhead of $42mJ/KB$. Given a typical battery capacity of 26 KJ in the sensor node, we can, therefore, estimate that the number of 1KB transactions that can be completed in the secure mode by the sensor node before the battery runs out of power is less than half the corresponding number in the un-encrypted mode (see Figure 4).

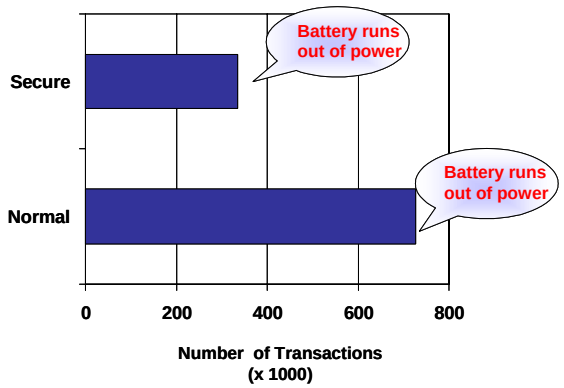

Figure 4: The impact of security processing on battery life

While the energy requirement for security will only increase in the future, the supply side also leaves much to be desired. There has only been a slow growth (5-8% per year) in the battery capacities [37]. With the increasing complexity of functions that a mobile appliance supports, the energy requirements, even in the absence of security processing, seem to be outpacing the much slower evolution of battery technologies. Since the addition of security processing to the existing workload only threatens to widen this gap, it becomes very important to consider battery-aware system design techniques while embedding security in a mobile appliance.

3.4 Tamper-Resistance

Most security protocols and mechanisms address security of a mobile appliance without regard to the specifics of the implementation. Theoretical analyses of the strength of cryptographic algorithms assume that malicious entities do not have access to the implementation (classical cryptanalysis). Here, a cryptographic primitive is viewed as an abstract mathematical object, that is, a mapping of some inputs into some outputs parameterized by a

secret value, called the key. An alternative view of the cryptographic primitive comes from its implementation. Here, the primitive manifests itself as hardware circuit or as a program that will run on a given embedded processor, and will thus present very specific characteristics. Such a view implies that security protocols and cryptographic algorithms can simply be broken by observing properties of the implementation (for example, "side-channel information", such as timing, power, behavior in the presence of faults, *etc.*). Sensitive data can also be compromised, while it is being communicated between various components of the system through the on-chip communication architecture, or, even when simply stored in the mobile appliance (in secondary storage like Flash memory, main memory, cache, or even CPU registers).

Thus, secure design of the the HW/SW system architecture becomes as important as secure data communications. The first step in this process is to understand the various ways in which a mobile appliance can be attacked. We will now give a brief overview of the common techniques that can be used to "attack" a mobile appliance. The techniques are classified into two broad categories: *physical and side-channel attacks*, and, *software attacks*.

- *Physical and side-channel attacks* refer to attacks that exploit the system implementation and/or identifying properties of the implementation. It is not surprising that the first target of these attacks [38, 39, 40, 41] are mobile devices such as smart cards. For concreteness, the discussion here will be put in that context, although most of it applies to other (cryptographic) devices as well. Physical and side-channel attacks are generally classified into invasive and non-invasive attacks. Invasive attacks such as micro-probing techniques involve getting access to the silicon to observe, manipulate and interfere with the system internals. Since invasive attacks typically require relatively expensive infra-structure, they are much harder to deploy. Non-invasive attacks, on the other hand, do not require the device to be opened. While these attacks require knowledge of the system, they tend to be cheap and scalable (compared to invasive attacks).

  There are many forms of non-invasive attacks. Fault induction techniques manipulate the environmental conditions of the system (voltage, clock, temperature, radiation, light, eddy current, *etc.*) to generate faults and to observe the related behavior [42, 43]. Eavesdropping techniques attempt to deduce information by monitoring any accessible system resources such as the supply and interface connections. The most common form of this attack involves analyzing the power consumption of the system [44, 45]. Other possibilities involve analyzing the electromagnetic radiation around the device [46]. Another important class of attacks is the timing attack [47, 48], which exploits the observation that the computations performed in some of the cryptographic algorithms often take different amounts of time on different inputs. A well-known example is the implementation of the RSA public-key cryptosystem using the Chinese Remainder Theorem (CRT) for improving the performance. Other attacks targeting symmetric encryption schemes such as DES have also been used.

- *Software attacks* are based on malicious software being run on the mobile appliance, that exploits weaknesses in security schemes and the system implementation. The likelihood of software attacks tends to be high in systems such as mobile terminals, where application software is frequently down-loaded from the Internet. The down-loaded software may originate from a non-trusted source and, hence, can be be used to implement attacks. Compared to physical attacks, software attacks typically require infrastructure that is substantially cheaper and easily available to most hackers, making them a serious immediate challenge to secure system design.

  Popular examples of software attacks include viruses and trojan-horse applications that exploit OS weaknesses and procure access to the system internals. There are many categories of software attacks. While *integrity attacks* can manipulate sensitive data or processes, *privacy attacks* lead to disclosure of private or confidential information. *Availability attacks*, on the other hand, can prevent the secure functioning of the system by denying access to system resources. Building attack resistance especially into software [49, 50, 51] would necessitate one or more of the following measures: (i) finding a means to ascertain the operational correctness of protected code and data, before and during run-time, (ii) providing protection against trojan horse applications trying to steal data (*e.g.*, cryptographic keys) from a security application that is run on behalf of the user, (iii) enforcing that application content can remain secret (*digital rights management*), and (iv) protecting against probing (looking at the memory used by secure applications) and reverse engineering (de-compilation, flow analysis, profiling *etc.*).

4. SECURE HW/SW PLATFORM ARCHITECTURES

In this section, we describe approaches to design secure mobile appliance architectures that address some of the challenges described in Section 3.

4.1 Elements of a secure mobile appliance architecture

Security challenges are usually complex even when viewed in a limited perspective (handset). Thus from a systems perspective, it is imperative to take a hierarchical approach where each layer of security provides a foundation for the one above it (Figure 5).

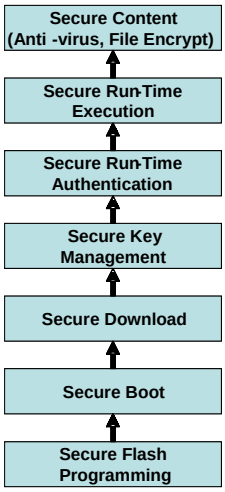

Figure 5: A layered hierarchical approach to security

A base platform architecture must be flexible and scalable to meet the needs of each stratum in the marketplace. Flexible and scalable base platform architectures simplify the development and deployment of new applications and services and the associated security requirements. Figure 6 shows an example of such a base architecture. At the core is a powerful crypto engine surrounded by firmware and an application-programming interface (API) which speeds the integration of various security applications and peripherals.

Figure 6: A modular base architecture for secure mobile appliances

Combining hardware and software crypto components plays a significant role in providing a strong crypto foundation that meets basic security requirements such as authentication, confidentiality, integrity, and non-repudiation.

The foundation of secure crypto operations includes true random number generation, which may be provided for with a HW-based random number generator. Crypto HW accelerators are one method to provide significant performance and power efficiencies to critically used algorithms such as DES/3DES, AES, SHA1/MD5, and public key operations (RSA/DH) necessary in the mobile environment. A low-power DSP in a dual-core processor, such as TI's OMAP1510 processor [52], accelerates critical and performance intensive crypto operations, freeing up much-needed headroom on the main applications processor.

Additionally, HW components such as secure RAM and secure ROM in conjunction with HW-based key storage and appropriate firmware can enable an optimized 'secure execution' environment where only trusted code can execute. A secure execution mode can be used for critical security operations such as key storage/management and run-time security to provide a strong security foundation for applications and services.

Another security challenge stems from most of today's devices relying on the authentication of the client device. The lack of end-user authentication is thus a weak link. Biometric technologies such as finger print recognition and voice recognition are emerging as important elements in enabling a secure wireless environment with minimal actions or understanding required from end-users.

4.2 Security processing architectures

Security processing refers to computations that need to be performed specifically for the purpose of security. For example, in a secure wireless data transaction, security processing includes the execution of any security protocols that are utilized at all layers of the protocol stack (and the cryptographic algorithms employed therein). As demonstrated in Section 3, the computational requirements of security processing place a significant burden on the embedded processors used in mobile appliances, and can lead to significant degradations in battery life. Recognizing these issues, various technologies have been developed in order to enable efficient security processing in mobile appliances. They include enhancements of embedded processors for security, cryptographic hardware accelerators, and programmable security protocol engines.

4.2.1 Embedded processor enhancements for securing processing

There have been several attempts to improve the security processing capabilities of general purpose processors. Since most microprocessors today are word-oriented, researchers have targeted accelerating bit-level arithmetic operations such as the permutations performed in DES/3DES. Multimedia instruction set architecture (ISA) extensions such as those in PA-RISC's Max-2 [53] or IA-64 [54] already incorporate instructions for permutations of 8-bit or larger sub-words. For arbitrary bit-level permutations, only recently have efficient instructions been proposed [55]. Instruction set extensions have also been proposed for other operations such as substitutions, rotates and modular arithmetic present in different private-key algorithms [56].

Many such extensions have already been applied to embedded processors used in the wireless handset domain. For example, the SmartMIPS [57] cryptographic enhancements extend the basic 32-bit MIPS ISA to speed up security processing. Similar features are also found in the ARM SecureCore family [58]. The security processing capabilities of SecureCore processors can also be further extended by adding custom-designed cryptographic processing units through a co-processor interface. This is useful for delivering efficient performance on new and proprietary cryptographic algorithms without having to re-design the basic processor core.

4.2.2 Cryptographic hardware accelerators

Highest levels of efficiency in processing are often obtained through custom hardware implementations. Since cryptographic (asymmetric, symmetric, hash) algorithms form a significant portion of security processing workloads, various companies offer custom hardware implementations of these cryptographic algorithms suitable for mobile appliances including smart cards and wireless handsets [59, 60]. Several vendors also offer integrated micro-controllers that contain embedded processors, cryptographic accelerators, and other peripherals [61, 62].

4.2.3 Programmable security protocol engines

While cryptographic accelerators alleviate the performance and energy bottlenecks of security processing to some extent, achieving very high data rates or extreme energy efficiency requires a holistic view of the entire security processing workload. In addition to cryptographic algorithms, security protocols often contain a significant protocol processing component, including packet header/trailer allocation parsing, *etc*. Security protocol engines (*e.g.*, the IPSec packet engine from Safenet Inc. [60]) accelerate all or most of the functionality present in a security protocol, resulting in higher efficiency than cryptographic accelerators.

As mentioned in Section 3, security standards for mobile appliances are in their infancy, and are expected to evolve significantly [32, 33, 63, 64, 65]. Hence, it is desirable to provide sufficient flexibility in security processing architectures so that they can be re-used, or adapted with minimal effort, to support new standards or enhancements of existing standards. Programmable security protocol engines, such as the MOSES platform developed at NEC [66, 67, 68] combine the benefits of flexibility and efficiency for security processing.

5. CONCLUSIONS

Security is critical to enabling a wide range of applications involving mobile appliances. While security has been addressed in the context of traditional computing systems and the wired Internet, mobile appliances usher in many new challenges. This paper highlighted the problems faced by designers of mobile appliances, and outlined recent technological developments and commercial innovations to address them. Security concerns are not limited to a specific application domain, but cut across a wide range of electronic systems. Hence, we believe that security will increasingly impact various aspects of the system design process, including hardware circuits and micro-architecture, software, system architecture, and design methodologies.

6. REFERENCES

[1] *MeT PTD definition (version 1.1)*. Mobile Electronic Transactions Ltd. (`http://www.mobiletransaction.org/`), Feb. 2001.

[2] P. Flavin, *Who needs a credit card when you have a mobile?* `http://www.btignitesolutions.com/insights/visionaries/flavin_mobile.htm`, (accessed Dec. 2002).

[3] *ePaynews - Mobile Commerce Statistics*. `http://www.epaynews.com/statistics/mcommstats.html`.

[4] U. S. Department of Commerce, *The Emerging Digital Economy II*. `http://www.esa.doc.gov/508/esa/TheEmergingDigitalEconomyII.htm`, 1999.

[5] World Wide Web Consortium, *The World Wide Web Security FAQ*. `http://www.w3.org/Security/faq/www-security-faq.html`, 1998.

[6] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 1998.

[7] B. Schneier, *Applied Cryptography: Protocols, Algorithms and Source Code in C*. John Wiley and Sons, 1996.

[8] S. K. Miller, "Facing the Challenges of Wireless Security," *IEEE Computer*, vol. 34, pp. 46–48, July 2001.

[9] P. Ashley, H. Hinton, and M. Vandenwauver, "Wired versus wireless ecurity - The Internet, WAP and iMode for e-commerce," in *Proc. 17th Annual Computer Security Applications Conf.*, Dec. 2001.

[10] *Wireless Security Basics*. Certicom (`http://www.certicom.com/about/pr/wireless_basics.html`).

[11] S. Hattangady and C. Davis, *Reducing the Security Threats to 2.5G and 3G Wireless Applications*. Texas Instruments Inc. (`http://focus.ti.com/pdfs/vf/wireless/securitywhitepaper.pdf`).

[12] S. Ravi, A. Raghunathan, and N. Potlapally, "Securing wireless data: System architecture challenges," in *Proc. Intl. Symp. System Synthesis*, pp. 195–200, October 2002.

[13] D. Boneh and N. Daswani, "Experimenting with electronic commerce on the PalmPilot," in *Proc. Financial Cryptography*, pp. 1–16, Feb. 1999.

[14] *Cellular Digital Packet Data System Specification, Release 1.1*. CDPD Forum, Jan. 1995.

[15] *European Telecommunication Standard GSM 02.09*. Digital Cellular Telecommunications System (Phase 2+): Security Aspects.

[16] C. Brookson, "GSM security: A description of the reasons for security and the techniques," in *Proc. IEE Colloqium on Security and Cryptography Applications to Radio Systems*, pp. 2/1–2/4, June 1994.

[17] *IEEE 802.11 Wireless LAN Standards*. IEEE 802.11 Working Group (`http://grouper.ieee.org/groups/802/11/`).

[18] *Bluetooth security white paper*. Bluetooth SIG Security Expert Group (`http://www.bluetooth.com/`), Apr. 2002.

[19] Y. Frankel, A. Herzberg, P. A. Karger, H. Krawczyk, C. A. Kunzinger, and M. Yung, "Security issues in a CDPD wireless network," *IEEE Personal Communications*, vol. 2, pp. 16–27, August 1995.

[20] S. Patel, "Weaknesses of North American wireless authentication protocol," *IEEE Personal Communications*, vol. 4, pp. 40–44, june 1997.

[21] J. R. Walker, *Unsafe at any key size: An analysis of the WEP encapsulation*. IEEE document 802.11-00/362 (`http://grouper.ieee.org/groups/802/11/Documents/`), Oct. 2000.

[22] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: The insecurity of 802.11," in *Proc. ACM Int. Conf. Mobile Computing and Networking*, pp. 180–189, July 2001.

[23] W. A. Arbaugh, *An inductive chosen plaintext attack against WEP/WEP2*. IEEE document 802.11-01/230 (`http://grouper.ieee.org/groups/802/11/Documents/`), May 2001.

[24] A. Mehrotra and L. S. Golding, "Mobility and security management in the GSM system and some proposed future improvements," *Proceedings of the IEEE*, vol. 86, pp. 1480–1497, July 1998.

[25] ISAAC group, U. C. Berkeley, *GSM cloning*. `http://www.isaac.cs.berkeley.edu/isaac/gsm.html`.

[26] *3GPP Draft Technical Specification 33.102, 3G Security Architecture*. `http://www.3gpp.org`.

[27] C. W. Blanchard, "Wireless security," *BT Technology Journal (`http://www.bt.com/bttj/`)*, vol. 19, pp. 67–75, July 2001.

[28] *Wireless Application Protocol 2.0 - Technical White Paper*. WAP Forum (`http://www.wapforum.org/`), Jan. 2002.

[29] *SSL 3.0 Specification*. `http://wp.netscape.com/eng/ssl3/`.

[30] *IPSec Working Group*. `http://www.ietf.org/html.charters/ipsec-charter.html`.

[31] *TLS Working Group*. `http://www.ietf.org/html.charters/tls-charter.html`.

[32] *Open Mobile Alliance*. `http://www.wapforum.org/what/technical.htm`.

[33] *Mobile Electronic Transactions*. `http://www.mobiletransaction.org/`.

[34] *Intel StrongARM SA-1110 Microprocessor Brief DataSheet*. `http://www.intel.com/design/strong/datashts/278241.htm`.

[35] *The DragonBall processor family*. `http://www.motorola.com`.

[36] D. W. Carman, P. S. Krus, and B. J. Matt, "Constraints and approaches for distributed sensor network security," Tech. Rep. #00-010, NAI Labs, Network Associates, Inc., Glenwood, MD, Sept. 2000.

[37] K. Lahiri, A. Raghunathan, and S. Dey, "Battery-driven system design: A new frontier in low power design," in *Proc. Joint Asia and South Pacific Design Automation Conf. / Int. Conf. VLSI Design*, pp. 261–267, Jan. 2002.

[38] R. Anderson and M. Kuhn, "Tamper resistance - a cautionary note," 1996.

[39] R. Anderson and M. Kuhn, "Low cost attacks on tamper resistant devices," in *IWSP: International Workshop on Security Protocols, LNCS*, 1997.

[40] O. Kommerling and M. G. Kuhn, "Design principles for tamper-resistant smartcard processors," in *Proc. USENIX Wkshp. on Smartcard Technology (Smartcard '99)*, pp. 9–20, May 1999.

[41] J. J. Quisquater and D. Samyde, "Side channel cryptanalysis," in *Proc. of the SECI*, pp. 179–184, 2002.

[42] D. Boneh, R. DeMillo, and R. Lipton, "On the importance of checking cryptographic protocols for faults," *Springer-Verlag Lecture Notes in Computer Science (Proceedings of Eurocrypt'97)*, vol. 1233, pp. 37–51, 1997.

[43] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," *Lecture Notes in Computer Science*, vol. 1294, pp. 513–525, 1997.

[44] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Springer-Verlag Lecture Notes in Computer Science*, vol. 1666, pp. 388–397, 1999.

[45] J. J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and counter-measures for smart cards," *Lecture Notes in Computer Science (Smartcard Programming and Security)*, vol. 2140, pp. 200–210, 2001.

[46] W. van Eck, "Electromagnetic radiation from video display units: an eavesdropping risk?," *Computers and Security*, vol. 4, no. 4, pp. 269–286, 1985.

[47] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," *Springer-Verlag Lecture Notes in Computer Science*, vol. 1109, pp. 104–113, 1996.

[48] E. English and S. Hamilton, "Network security under siege: the timing attack," *IEEE Computer*, vol. 29, pp. 95–97, March 1996.

[49] D. Aucsmith, "Tamper Resistant Software: An Implementation," *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 317–333, 1986.

[50] M. Blum and S. Kannan, "Designing programs that check their work," in *Proc. ACM Symposium on Theory of Computing*, pp. 86–97, 1989.

[51] C. S. Collberg and C. Thomborson, "Watermarking, tamper-proofing, and obfuscation - tools for software protection," *IEEE Transactions on Software Engineering*, vol. 28, pp. 735–746, August 2002.

[52] *OMAP Platform - Overview*. Texas Instruments Inc. (`http://www.ti.com/sc/omap`).

[53] R. B. Lee, "Subword Parallelism with Max-2," *IEEE Micro*, vol. 16, pp. 51–59, Aug. 1996.

[54] Intel Corp., *Enhancing Security Performance through IA-64 Architecture*. http://developer.intel.com/design/security/rsa2000/itanium.pdf, 2000.

[55] R. B. Lee, Z. Shi, and X. Yang, "Efficient Permutations for Fast Software Cryptography," *IEEE Micro*, vol. 21, pp. 56–69, Dec. 2001.

[56] J. Burke, J. McDonald, and T. Austin, "Architectural Support for Fast Symmetric-Key Cryptography," in *Proc. Intl. Conf. ASPLOS*, pp. 178–189, Nov. 2000.

[57] *SmartMIPS*. http://www.mips.com.

[58] *ARM SecurCore*. http://www.arm.com.

[59] *Cryptocell™*. Discretix Technologies Ltd. (`http://www.discretix.com`).

[60] *Safenet EmbeddedIP™*. Safenet Inc. (`http://www.safenet-inc.com`).

[61] *SLE 88 family*. Infineon Technologies (`http://www.infineon.com`).

[62] *ST19 smart card platform family*. STMicroelectronics Inc. (`http://www.st.com`).

[63] *Mobey Forum*. `http://www.mobeyforum.org/`.

[64] *Mobile Payment*. `http://www.mobilepaymentforum.org/`.

[65] *Consortium for efficient embedded security*. `http://www.ceesstandards.org/`.

[66] S. Ravi, A. Raghunathan, N. Potlapally, and M. Sankaradass, "System Design Methodologies for a Wireless Security Processing Platform," in *Proc. ACM/IEEE Design Automation Conf.*, pp. 777–782, June 2002.

[67] N. Potlapally, S. Ravi, A. Raghunathan, and G. Lakshminarayana, "Optimizing Public-Key Encryption for Wireless Clients," in *Proc. IEEE Int. Conf. Communications*, pp. 1050–1056, May 2002.

[68] N. Potlapally, S. Ravi, A. Raghunathan, and G. Lakshminarayana, "Algorithm exploration for efficient public-key security processing on wireless handsets," in *Proc. Design, Automation, and Test in Europe (DATE) Designers Forum*, pp. 42–46, Mar. 2002.