Embedded Systems Security

Aspects of secure embedded systems design and implementation

Konstantinos Fysarakis
Dept. of Applied Informatics & Multimedia
Technological Educational Institute of Crete,
Heraklion, Greece

Charalampos Manifavas
Dept. of Applied Informatics & Multimedia
Technological Educational Institute of Crete,
Heraklion, Greece

Konstantinos Rantos
Dept. of Industrial Informatics
Technological Educational Institute of Kavala,
Kavala, Greece

Introduction

Information Security:

- Three main goals:
  - Confidentiality
  - Integrity
  - Availability

A “game” we’ve been playing for a few decades, one that never ends and which is impossible to win (i.e. there is no such thing as a totally secure system)!

Sounds fun? You bet! And it gets even better…

There is this new level (let us call it “Embedded Systems”) in which we are usually asked to play the game with one or, in some cases, both hands tied behind our back!

Let’s see why…

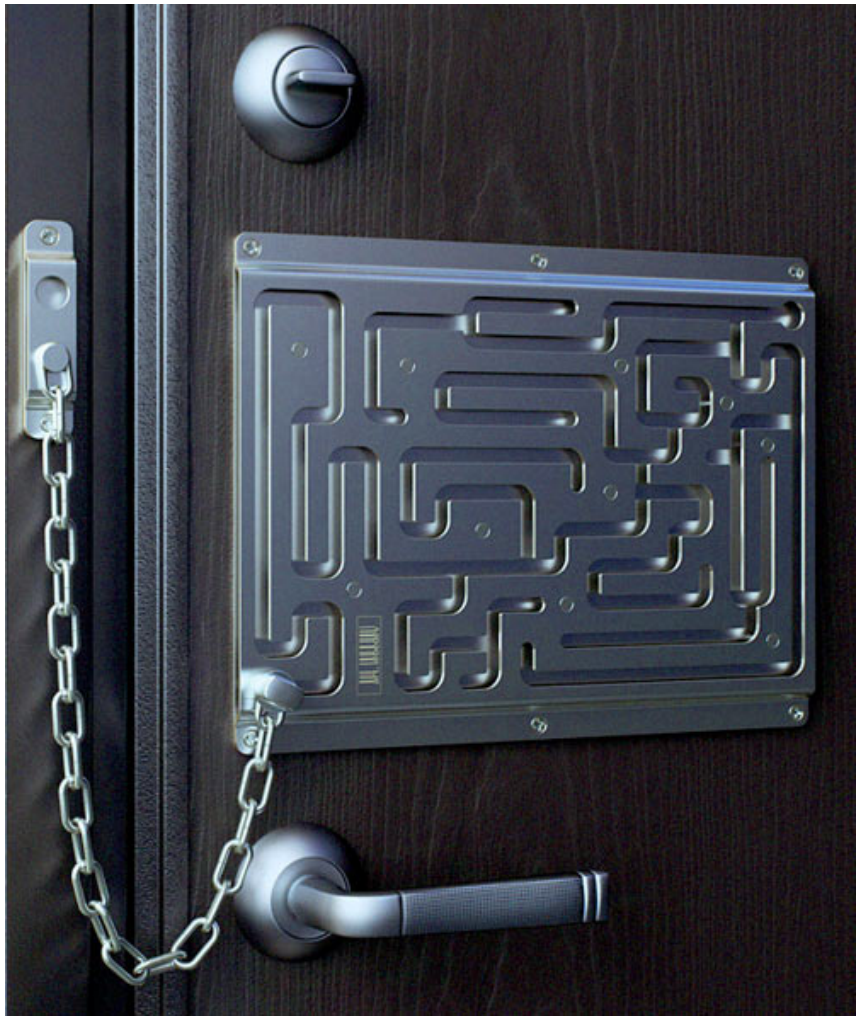
Introduction

Embedded Systems

- becoming available anytime, anywhere…
  - Residential / Home automation
  - Industrial systems
  - Public infrastructures
  - Avionics
  - Military
  - E-textiles
  - Automobiles
  - …
- …and in many different forms
  - small or large
  - visible or invisible
  - attached or embedded
  - simple or complex
  - wired or wireless
  - coordinated or ad hoc
  - …

- How to secure such heterogeneous systems with diverse characteristics?

Introduction



Mindlessly applying security measures is not a good idea...

Introduction

Security & Embedded Systems

We have to consider their intrinsic and often application-specific characteristics.

E.g.:

- Resource constraints
  - Computational capabilities
  - Memory
  - Power
- Dynamically formulated, remotely managed or unmanaged, networks
- Deployment in “hostile” environment
- Utilization in time-critical applications
- …etc.

…the hand(s) tied behind our backs…

Introduction

- Most security issues are exacerbated by the aforementioned characteristics of ESs.
- Security techniques developed for personal and enterprise computers (and their communication systems) are often inadequate or even inapplicable.
- Replacement, ES-friendly, mechanisms are in many cases non-existent or under development.

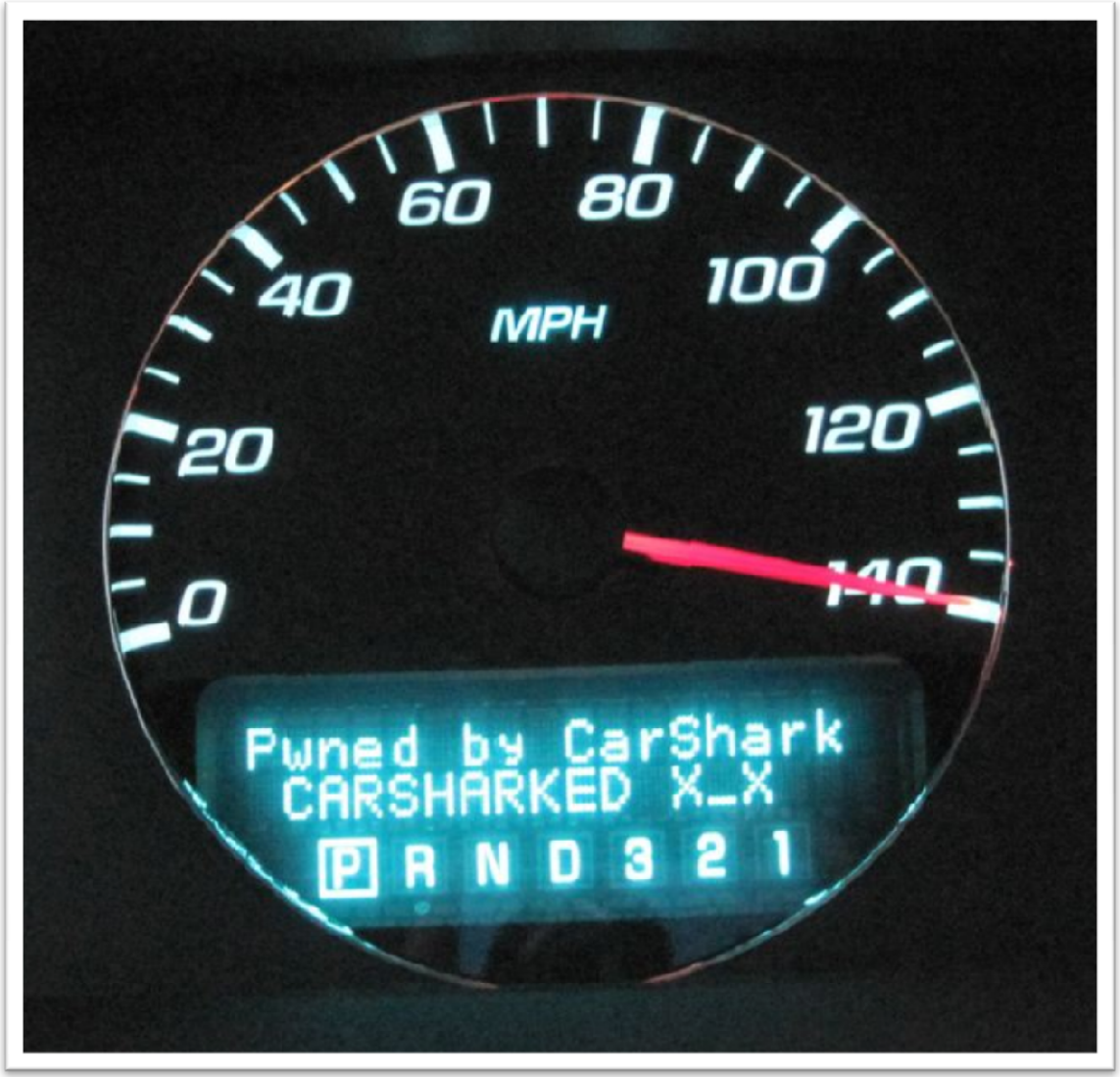
But why should we care?

Importance of ESs Security

- Among other things, ES applications often include direct interaction with the physical world.
  - It can be irritating (and costly) having the thermostat in our brand new “smart-home” be controlled by the teenage neighbor who has a lot of time in his hands.
  - More importantly, exploiting the centralized power monitoring and distribution infrastructure of the power company could lead to a spike in power consumption and subsequently a local or total black-out.
  - Most importantly, a security incident might lead to asset damage or even personal injury and death.

Sounds far-fetched?

Case Study - Vehicular Computer Security



K. Fysarakis, C. Manifavas, K. Rantos,
"Embedded Systems Security", AmiEs 2011

Case Study - Vehicular Computer Security

In 2010, Koscher et al. demonstrated that it is feasible to manipulate all critical sub-systems in modern automobiles using a wireless-enabled MP3 player connected to the vehicle's embedded control network.

- Their analysis exposed numerous vulnerabilities in the design and implementation of the Controller Area Network (CAN, ISO 11898) bus protocol, utilized for the communication of the various ECUs (up to 70 of which can be embedded in a modern vehicle) and sensors.
  - CAN is used by BMW, Ford, VW, Honda and GM among other manufacturers.

Attacks included disabling or forcibly activating any of the brakes, consequently compromising the safety of the driver and passengers, as well as injecting malicious code to erase any evidence of tampering after a crash.

  - Alternative bus protocols face, more or less, the same issues since the automobile industry typically consider safety only in terms of car crashes and component failures but not a malicious user attacking the car's electronics.

- Vehicular security is bound to become more critical with the deployment of even "smarter" cars featuring technologies like Vehicle-to-Vehicle and Vehicle-to-Infrastructure ad-hoc networks.

Case Study – Critical Infrastructure Security

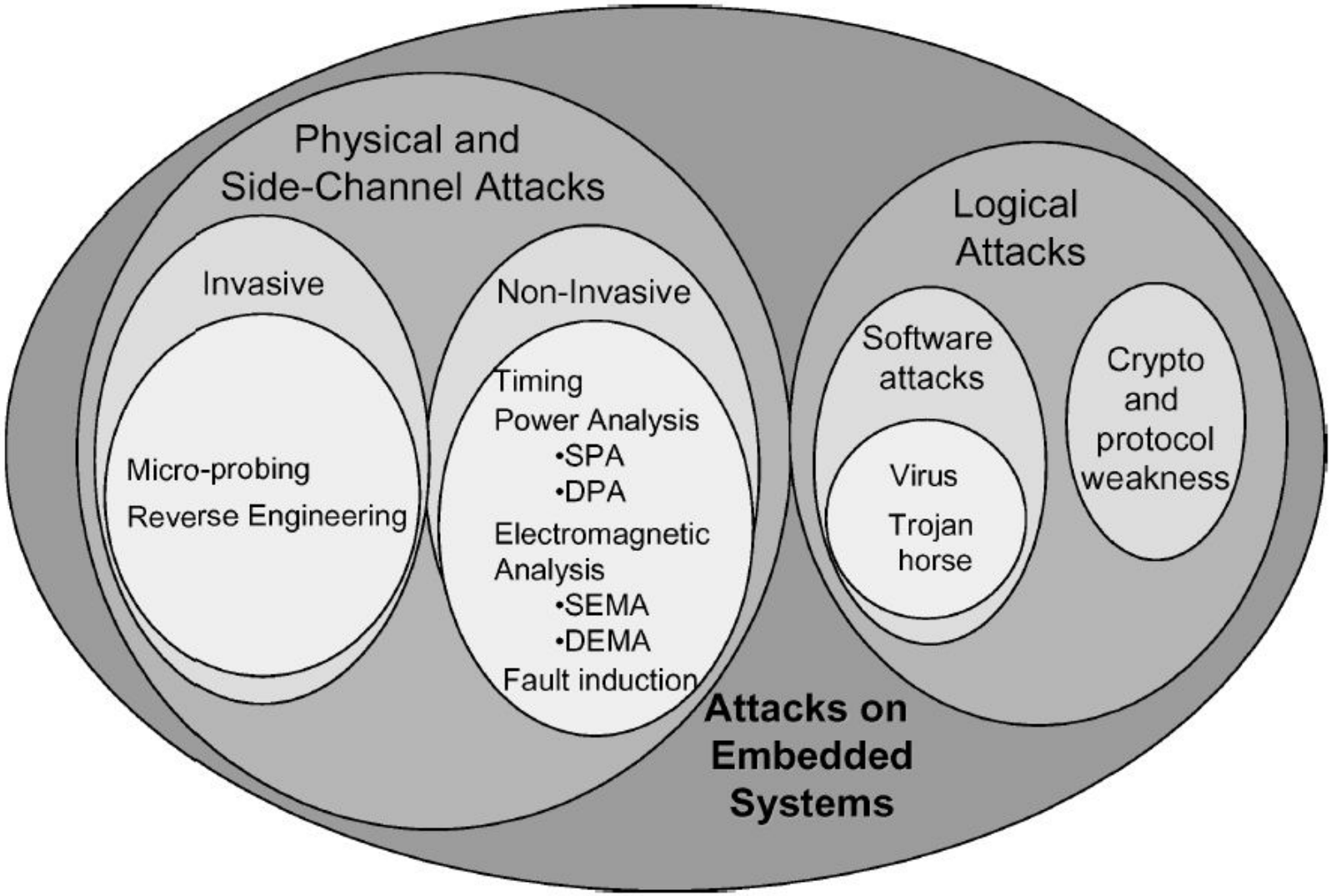
K. Fysarakis, C. Manifavas, K. Rantos,
"Embedded Systems Security", AmiEs 2011

Case Study – Critical Infrastructure Security

STUXNET Worm

- A highly specialized and sophisticated malware, discovered in July 2010.
- Designed to target the specific Siemens supervisory control and data acquisition (SCADA) systems installed in Iran's uranium enrichment infrastructure.
  - The purpose of the worm was to take control of the PLCs causing periodic variations in the uranium enrichment centrifuges' rotor speed, thus destroying the devices.
- Indeed, because of Stuxnet, Iranian scientists were forced to replace approximately 1000 centrifuges over a few months when, prior to the attack, normal failure rates where in the region of 800 per year.

So, now that we (hopefully) agree about the importance of ESs security, let's move on!



K. Fysarakis, C. Manifavas, K. Rantos,
"Embedded Systems Security", AmiEs 2011

Physical Attacks

- Given the often unattended nature of deployed ESs, the risk of device tampering should not be ignored.
- A malicious entity's physical access to a device would enable the launch of:
  - intrusive attacks like micro-probing and reverse-engineering
  - sophisticated Side-Channel Attacks (SCA), like timing attacks, simple power analysis (SPA), differential power analysis (DPA), differential fault attacks (DFA) etc.

Power Supply Issues

- Many embedded systems have inherent energy constraints and are often battery powered. Some might get a daily battery charge but others may be expected to last months on a single charge.
- An attacker who fails to otherwise compromise the system could decide to instead launch a DoS attack (more on that later!) by draining the battery power (e.g. by forcing the device to use its wireless connection or work at full CPU load).
- The power source of an ES should satisfy three key requirements:
  - Provide continuous power, without any unpredicted fluctuations of its output voltage or current levels, ensuring optimal operation of the powered device.
  - Monitor its own state and prevent any power supply issues that might affect the system's operation.
  - Feature fail-safe mechanisms to protect and prevent any further damage to the device in case of failure.
- Most of the above are common in modern high-end Uninterruptible Power Supply (UPS) systems but are a challenge in some applications of ESs where even a backup battery is considered a luxury as it significantly increases size and cost.
- Solutions to be considered:
  - energy scavenging, super-capacitors, micro-solar cells, wireless power transferring schemes
  - fail-safe options (power off non-critical systems, disconnect damaged sub-systems etc.)

Access Control Issues

- Access Control mechanisms are essential to prevent unauthorized/malicious entities from accessing the system resources, physical or otherwise.
  - Some often-used methods include: Profile Authentication, Access Code, Predefined Topology.
- The commonly used authentication schemes, typically password-based, can be impractical or even insecure when considering the heterogeneous nature ES networks can demonstrate and the scalable remote manageability often required
- Even in wired embedded networks and in industries like automotive and aviation, most control networks utilized (e.g. Controller Area Network, Time-Triggered Protocol, FlexRay) are designed with safety and reliability in mind and do not feature any built-in security mechanisms like node authentication, data encryption or prevention of Denial of Service (DoS) attacks.

DoS and DDoS Attacks

- Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks aim to compromise the availability of a node or network of nodes
  - preventing authorized entities from accessing system resources and services or delaying system operations and functions.
- Many ways to achieve this since DoS attacks can essentially be mount on all different layers (e.g. exploiting software and/or firmware vulnerabilities, flooding the node with traffic, misrouting, exploiting protocol vulnerabilities etc.), the end goal being to consume the nodes' CPU cycles, memory, network bandwidth and power or even physically destroy the node.

DoS and DDoS Attacks

- The unattended nature of many ESs necessitates the implementation of remote management features. These features can be exploited to launch remote attacks.
- E.g.: Phlashing (i.e. exploiting remote firmware upgrade mechanisms to corrupt flash memory)
  - Relatively easy since:
    - mechanisms are turned “ON” by default
    - firmware binaries are freely available on the Internet
    - the protection mechanisms are typically elementary (the process is designed with error detection, not malicious attacks, in mind).
- The outcome of a successful attack can be the permanent destruction of a node (PDoS or Bricking), requiring out-of-band hardware re-initialization or the installation of a new node in order to restore service.

DoS and DDoS Attacks

- A particularly challenging issue to address, especially in the context of ESs. Some guidelines:
  - Employing secure node firmware deployment and software updates
  - Choosing network protocols resilient to (D)DoS attacks, sound authentication, access control and resource allocation mechanisms
  - Employing fail-safe mechanisms, even hardware redundancy if cost allows, esp. in applications where high-dependability is needed (e.g. avionics, military)
  - Using runtime reconfiguration (where hardware allows – FPGAs)
    - Self-reconfigurability can, for example, make a node more secure against side-channel attacks through measurement of EM radiation
  - Self-recovery mechanisms could reallocate functional blocks to mark and replace faulty resources, through device reprogramming in the case of self-reconfigurable nodes or through controlled degradation of service techniques in less “intelligent” devices.

Cryptographic Mechanisms

- A Trusted Platform Module (TPM), is an example of a component that can be integrated on ESs to provide tamper resistant hardware and software security functions. The software embedded on such a cryptographic component has direct impact on its:
  - Size: Memory elements constitute a significant part of the module's surface.
  - Costs: Directly linked to the surface of the component.
  - Speed: Optimized code provides results faster.
  - Power Consumption: The quicker a set of instructions is executed, the quicker the module can return to an idle state or be put in sleep mode where power consumption is minimal.

Cryptographic Mechanisms

Lightweight Cryptography

- Refers to algorithmic designs and implementations best suited for deployment in devices with inherent limitations in terms of processing power, memory, storage and energy.
  - The aim is to maintain the level of security “traditional” algorithms and implementations offer while narrowing what is often referred to as “battery gap”, i.e. the very high energy consumption overheads of supporting security on battery constrained systems.
- Lightweight symmetric algorithms:
  - DESL, Present. Researchers also focus on developing lightweight hardware and/or software implementation of existing and well-established algorithms like AES, IDEA etc.
    - E.g. Feldhofer et al. presented a hardware implementation of the AES algorithm, supporting encryption, decryption and key setup and which occupies an area of 0.25mm^2, the size of a small grain of sand, and draws only 3.0μA of current at 1.5V

Cryptographic Mechanisms

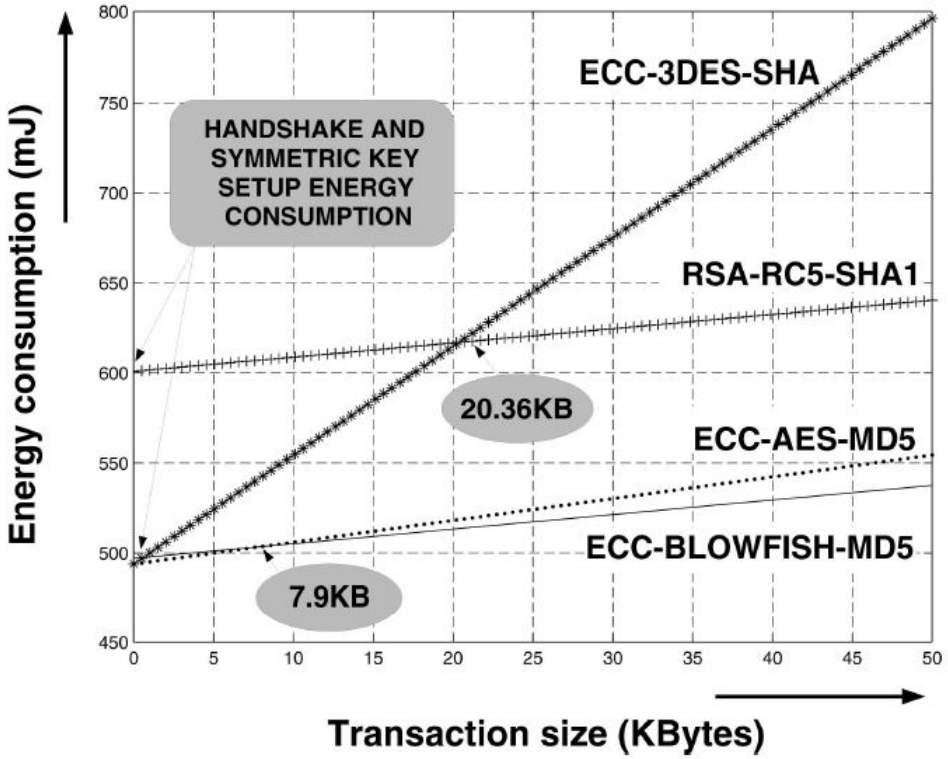
Lightweight Cryptography - Issues

- Several mature block ciphers are available and their characteristics (i.e. performance and security) are well understood and documented but stream cipher designs are lacking in comparison.
- Hash functions design is another area where further research is required since existing solutions are not sufficiently lightweight. Hashes based on block ciphers seem to still have an advantage.
- Developing lightweight versions of asymmetric algorithms and protocols is also an elaborate task.
  - Asymmetric ciphers are computationally far more demanding than their symmetric counterparts. The performance gap is even more obvious on constrained devices such as 8-bit microcontrollers.

Cryptographic Mechanisms

Lightweight Cryptography

- Typically, since asymmetric ciphers come with such intrinsic performance issues they are mainly used for key-management facilities and non-repudiation, whereas integrity checks, entity authentication and the encryption are provided by symmetric primitives.
- Elliptic Curve Cryptography (ECC) is considered the most attractive option in ESs because of its small operand length and relatively low processing requirements.
  - HyperElliptic Curve Cryptography (HECC) is another type of curve-based cryptography that has been recently revisited.



•This graph (Potlapally et al.) shows the impact of cipher suite selection on energy consumption during the SSL handshake and record stages.

•Let's consider two cipher suites, RSA-RC5-SHA and ECC-3DES-SHA.

•In this hybrid scheme the public-key algorithm (RSA or ECC) is used in the SSL handshake stage and the symmetric key algorithm (RC5 or 3DES) is used for bulk encryption in the SSL record stage.

•For data sizes smaller than 21Kb, ECC-3DES-SHA is more energy-efficient because ECC is simpler than RSA (and asymmetric energy consumption dominates that of small data transactions).

•However, for transactions where there are significant bulk data (greater than 21 Kb) to encrypt, RSA-RC5-SHA consumes less energy, because for large data transfers energy consumption of symmetric ciphers dominates the total energy spent, and RC5 is much simpler than 3DES.

Careful choice of the cryptographic suite, i.e. one that takes into consideration the size of data that will be typically processed and transferred, can greatly reduce the amount of energy consumed.

Cryptographic Mechanisms

- The need for lightweight cryptography introduces major multi-dimensional challenges in cryptographic algorithms design. Some key points:
- Hardware and software co-design seems to offer the best results in terms of speed/size ratio for many ubiquitous computing applications.
- Regarding primitives that cannot yet be effectively implemented (e.g. hashes) alternatives could be investigated so that the protocols which are based upon them can be researched further and, perhaps, employed.
- Special care should be taken during the development of optimized implementations so that they do not introduce new leakage channels which could be exploited by Side-Channel Attacks (SCA).

Network Protocol & Management Issues

- Certain applications of embedded systems, like Wireless Sensor Networks, rely on the integrity of the platform for providing trustworthy services (e.g. measurements taken by a sensor).
  - It is essential to have a method of validating this integrity and assuring that system components have not been compromised.
  - The integrity of the service requester platform, i.e. control node, must also be validated before allowing it to allocate resources to the nodes it controls or receive the data these nodes have collected.
  - It should also be established that these secure resource management mechanisms will not act as a bottleneck in service performance.
- Employing TPM could be an option for the above.

Network Protocol & Management Issues

- Inspecting the problem from a higher level, middleware resources should be managed by:
  - monitoring their availability
  - enforcing a policy based on which these resources are assigned
  - implementing a secure model for the identification and authorization of requests
  - Implementing an account system to track resource usage.
- Most of the above can be found in protocol Diameter, successor to RADIUS.

Network Protocol & Management Issues

Reputation-based Schemes

- A relatively novel paradigm for enhancing security in various applications, including secure routing and intrusion detection systems for Mobile Ad Hoc Networks (MANETS).
  - These systems are secure, easy to implement and lightweight.
- The basic concept is inspired from social behavior and relies on the cooperation of the nodes. Much like human interaction, each entity decides to trust or ignore a new, unknown entity based on the opinion of his/her peers about the individual in question.
- The three main goals identified for reputation systems are:
  - To provide the required information in order to distinguish between a trustworthy principal and an untrustworthy one.
  - To encourage principals to act in a trustworthy manner.
  - To discourage untrustworthy principals from participating in the service.

Network Protocol & Management Issues

Anonymity and Location Privacy

- Location-based applications are a relatively new and rapidly expanding market, owing to the widespread use and advances both in mobile devices and positioning systems.
  - Applications: Enhanced Reality services, location-aware emergency response, entertainment, targeted advertising etc.
- The location of an individual constitutes sensitive personal data.
  - It can reveal information about his/her personal relationships, political affiliations, medical issues etc.
- Disclosure of such information can enable a malicious user to harass, blackmail, enter the individual's residence (e.g. when he/she is away) etc.
- Such information needs to be handled accordingly and there is on-going research on the subject (mostly on k-Anonymity schemes i.e. in databases with personal data, attributes are suppressed or generalized until each row is identical with at least k-1 other rows).

In Conclusion

- ESs Security is important!

  ...not merely because of the potentially dire consequences a successful security attack might have in the case of critical systems but also because these attacks are bound to become more common as ESs become an even more integral part of our lives, with the widespread adoption of smart devices in our homes, cars, clothes etc.

- It is essential to consider the intrinsic and often application-specific characteristics of ESs and their particular requirements which:
  - introduce new vulnerabilities
  - exacerbate existing ones
  - limit the efficacy of established computer security techniques and mechanisms (including access control mechanisms, cryptographic primitives and network protocols).

In Conclusion

- Further research is required on various ES-friendly security mechanisms.
- Equally important to the research and development of such security mechanisms is that they are considered at the design phase, constituting an integral part of a developed system and not patched-in additions to an existing insecure one.

Security performance is going to be one of the next product differentiators in embedded products and services!

Thank you!

Questions?