Security of Distributed, Ubiquitous, and Embedded Computing Platforms

Anthony D. Wood, University of Virginia

John A. Stankovic, University of Virginia

Keywords

Ad Hoc Network, Wireless Sensor Network, RFID, Embedded, Security, Privacy

Abstract

As embedded computer systems continue to explode in number and capability, security and privacy challenges abound. We review desirable security properties and the design constraints posed by these systems that make security difficult. We summarize current research by focusing on solutions for ad hoc networks, wireless sensor networks, and RFID tags as representative of the design space. State of the art protocols and approaches for defeating or mitigating attacks at the physical, network and middleware layers are presented. Critical application areas and research needs are identified, as are possible funding sources.

Computer systems and networks are becoming more capable—and more vulnerable—as they are embedded more deeply into our environment. In this article we describe security challenges faced by ubiquitous distributed systems: ad hoc networks of handheld computers, sensor networks for directly interacting with the world, and radio-frequency identification (RFID) tags which instantiate real-world objects with elements in our virtual computer systems. We review promising research approaches, and identify important future directions in these application areas.

Scientific Overview

The confluence of wireless networking, increasing transistor densities (Moore's Law), and miniaturization of manufacturing processes has accelerated the deployment of computer networks. Computing devices are now lightweight, portable, unobtrusive, powerful, and more well-connected than ever. Adding environmental and biological sensors tightens the connection with the real world, so that computing is not just embedded in non-computing devices (like the proverbial Internet toaster), but is embedded in our living spaces.

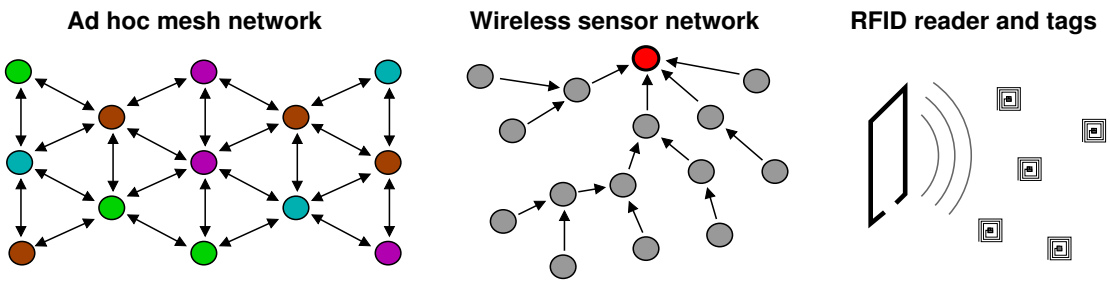

Figure 1: Example network connectivity for an ad hoc network, sensor network, and RFID reader and tags.

We focus on three developing technology areas, represented in Figures 1 and 2: ad hoc networks, wireless sensor networks, and RFID tags. Their applications range widely and are expanding, including military battlefield awareness, airport surveillance, emergency medical care, disaster response, critical infrastructure monitoring, container tracking, facilities access control, firearm and vehicle immobilizers, currency and travel document fraud detection, and border enforcement.

Security requirements are unique for each application, but overall they are becoming increasingly significant due to several factors. The systems being monitored, controlled, or protected are often critical for economic or safety reasons. Technological societies are becoming more dependent on their proper operation and real-time response. The networks are pervasive in many environments, where they are easily accessible and, therefore, exposed to greater threats. For example, wireless accessibility, while a great convenience, also makes it easier for attackers to find and interact with devices. Finally, the deepening of familiarity with and acceptance of computing devices extends to the unscrupulous, as well. The constant attacks that occur daily on the Internet, from ego-boosting web defacements to vengeful distributed denial of service botnets, may eventually be the norm on any accessible network.

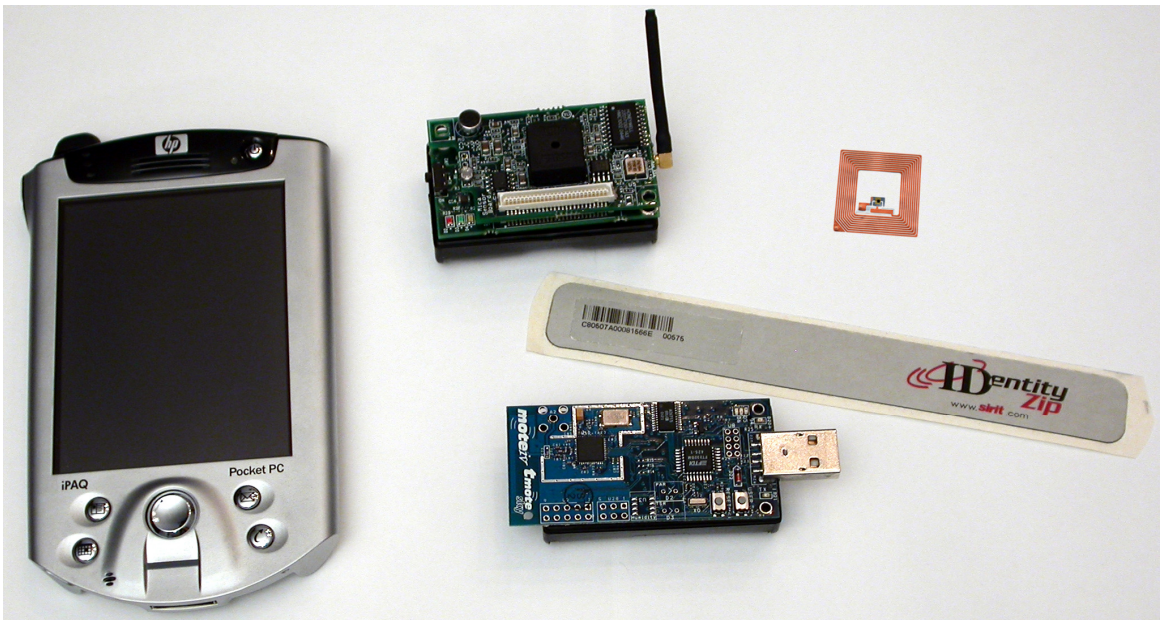

Figure 2: Examples of embedded devices: PDA, sensors, and RFID tags.

Security Properties

Security properties can be distilled to a core set, many of which may be important for a given application. Many others are defined in the literature [1], and here we describe those discussed in this article, giving brief examples of their use.

- Confidentiality: secrecy of communication between parties. Exposure of communications in wireless networks makes eavesdropping a constant threat.
- Integrity: assurance that data has not been modified by an unauthorized party. This applies to messages in transit, records stored in databases, and even data possessed by attackers (such as on stolen smart cards).
- Authenticity: assurance that a message originated from a known other party. Among others, command and control systems require high confidence that actions with large or dangerous effects have been issued by appropriate means. Message Authentication Codes (MACs) are often appended to protocol messages to provide this property.
- Identification: determination of a contextually unique label for a party. It enables authentication of a party and authorization of actions it may take. Also, a persistent ID allows goods to be tracked through supply chains, from manufacturers to shelves.
- Authorization: determination of privileges from a party's identity. System designs may change the authorized set of actions a party may take based on environmental contexts, for example, granting additional access during medical emergencies.

- Access Control: limitations on exposure or modification of protected resources to authorized parties. An RFID token that serves as a "key" for an automobile is a form of access control.
- Availability: a service or system performs its function in a timely manner for legitimate users. Denial of service attacks may crash a system completely, or may only slow it down enough to cause significantly disrupted service.
- Auditability: logging of security-relevant actions or events for later analysis. Many attacks can not be reliably detected in real-time, but can be analyzed after the fact to help with future defenses.
- Tamper Resistance: ability of a device's packaging and design to withstand physical modification or interrogation. Smart cards, though in public possession, often contain secret keys which must remain secret to prevent changing credit balances.
- Non-Repudiation: inability of a user or device to deny participation in a protocol or performance of an operation after the fact. This is often related to auditability.

Constraints on the Design Space

Constraints on design are imposed by considerations such as available power, cost to manufacture and maintain, form factor and size, tamper-resistance, development effort, the ability to dynamically reprogram, and intended architectures for deployment. Devices in ubiquitous embedded networks form a spectrum of capabilities, from PDAs to passive RFID tags, and are connected together in varying ways.

Ad hoc networks [2] connect (frequently) mobile devices together in a relatively flat mesh and usually depend on peer routing for connectivity (see Figure 1). Hardware typically consists of cell phones, mobile handheld computers (PDAs) and laptop computers, with relatively powerful processors such as the Intel PXA255 running at 400MHz. They may use networks with high bandwidth, e.g. IEEE 802.11b/g, to deliver multimedia. Storage on internal and removable flash drives with capacity up to 2GB is common.

Wireless sensor networks [3] may also use node-to-node ad hoc connectivity, perhaps organized hierarchically with one or more nodes to act as sinks for generated data. Devices are primarily constrained by size, cost, and power. For example, the Crossbow Mica2 family of motes uses the 8-bit Atmel ATMEGA128 processor operating at 8MHz, with 4KB RAM and 128KB flash. Simple FSK modulation at 900MHz, IEEE 802.15.4, or Bluetooth radio communication is common.

RFID tags [4] are even more limited. Most are completely passive, using the energy of a reader's transmission to briefly power the tag's processing circuit. The tag communicates by modulating the reader's transmission. Tags may be

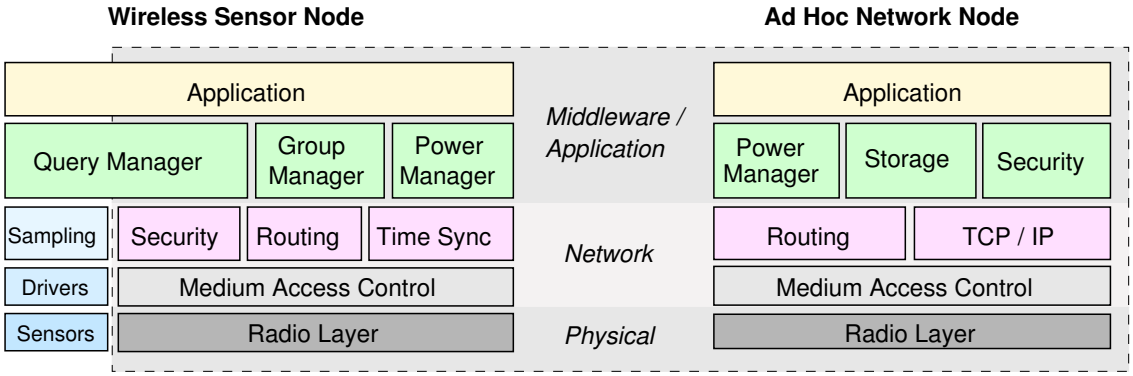

Figure 3: Typical software stacks for wireless sensor network and ad hoc network devices. A dashed box surrounds the communication layers, which contain various services often found in the networks.

smaller than 1-2mm (without the antenna), and operate in the high-frequency (HF) 13.56 MHz band for intermediate range.

Security comes at the cost of memory, computation, and messaging [5, 6]. Ad hoc network devices may be able to afford expensive asymmetric cryptography, storing 1024-bit keys for their neighbors, and participating in multi-round key establishment protocols. Sensor network devices cannot afford this computation and storage expense, unless very efficient elliptic-curve cryptographic (ECC) methods are used only infrequently. Instead, most researchers focus on lighter-weight symmetric cryptography and hashing in this context. Many RFID tags provide no security at all. Those that do may use only hashing or very efficient symmetric methods.

Hence, there are considerable differences in the security approaches that are practical and possible in distributed, embedded, and ubiquitous networks. Next, we describe the state of important research areas in security for these types of systems.

Solution Approaches

Distributed devices typically use layering to modularize hardware and software. Figure 3 shows generic software stacks for ad hoc and wireless sensor devices, and how services may be classified by layer. Due to their limited capabilities, RFID tags may be considered to have only a couple of layers. For this discussion we abstract away many details unique to each network type.

Strong security mechanisms at higher layers may be completely subverted by design or coding flaws at lower layers. Nowhere is this more evident than at the physical layer. Therefore, we describe attacks and defenses proposed in the state of the art by focusing on solutions to securing services at the critical physical, network, and middleware/application layers of the stack, starting at the bottom. We discuss ad hoc, sensor, and RFID networks together in each layer.

Physical Layer

The ubiquity of network devices means that they are easily inspected and probed by attackers. There is by definition no physical access control to sensors that are deployed throughout a public building, in parks, forests, or other open spaces. RFID tags attached to books, clothes, or supplies are necessarily as accessible as the asset they help to track.

The simplest attack is to destroy or disable the devices entirely, creating a denial of service. This is as low-tech as briefly putting a bank note or passport in a microwave oven. However, a destruction attack can often be mitigated using fault-tolerant protocols. For example, a mesh network can continue to operate despite some fraction of the devices being destroyed—remaining connected nodes take over routing. Or, if a passport's RFID tag is destroyed, backup procedures such as optical scanning of barcodes may be used instead.

Probing of the physical device to deconstruct its internals is more powerful and damaging. By reading the contents of memory cells, the secret keys are recovered and then programmed into another device which can fully masquerade as the original—yet is under attacker control. Messages originated by the clone are fully authentic, and the device can actively participate in formerly inaccessible transactions, as between a smart card and a payment terminal [7].

In addition to invasive techniques that usually require partial depackaging of a device, various physical properties of the circuits can be inspected without leaving a trace. Data-dependent computation affects the power consumption and timing of circuits, which can be analyzed statistically over many trials to determine bit patterns of a key [8]. Faults may be injected using heat or radiation, while the observed behavior is compared with correct behavior. Electromagnetic emissions may be inspected similarly to power consumption.

Proposed solutions include tamper-resistant packaging [7], better attack detection, fault recovery mechanisms, and reducing trust in external components [9]. For example, if a device can detect that it is being tampered with, it may erase its memory to prevent disclosure of sensitive data. Circuits may be shielded by distributing logical components across the die, bus traffic may be encrypted, and data values and timing can be randomized or "blinded" to thwart side-channel attacks.

Devices' use of wireless communication leaves them vulnerable to denial of service by radio jamming, which can be perpetrated at large distances and unobtrusively. Xu et al. propose channel hopping and "retreats" [10] to physically move away from the jammer. This is most appropriate for ad hoc networks, as it may be too energy consuming for sensor devices. Law et al. propose data blurting and changing transmission schedules as countermeasures [11]. Another approach, when the jamming cannot be avoided, is for nodes to determine the extent of the jammed area in a wide-scale network by collaboratively mapping and avoiding the region [12].

Networking Layers

We group the networking layers of the stack together to examine the security needs and vulnerabilities created when connecting multiple devices in networks. Services provided include channel arbitration, link establishment, one-hop data transmission, routing, and end-to-end data transport.

A primary concern is keeping data private, given the innate vulnerability to eavesdropping of wireless networks. Ad hoc networks often use protocols developed for the Internet, such as IPSec [13] and SSL/TLS [14]. Both of these protocols allow the use of suites of cryptographic mechanisms to provide authenticity, integrity, and confidentiality of messages. IPSec is commonly used to establish a secure virtual private network (VPN) connection between peers. SSL/TLS operates end-to-end, above an existing TCP connection, and further allows the client and server to negotiate a common set of capabilities. Though asymmetric methods may be used to establish keys and authenticate certificates, symmetric cryptographic protocols are used for data transfer.

In wireless sensor networks and RFID devices, symmetric mechanisms are encapsulated in lightweight protocols to provide data security. SPINS [15] provides two-party confidentiality and authenticity with the SNEP protocol. TinySec [16] similarly provides these properties using Skipjack or RC5 ciphers, in a fully-implemented and compact form with low overheads.

Due to energy constraints, sensor devices cannot use asymmetric cryptographic operations often. TinyPK [17] is an implementation of the relatively less demanding signature verification and key agreement for sensor devices. Though processing times for a single message may be over a minute (depending on the key length), they argue that it is acceptable for rare events, like code updating. Recent elliptic-curve implementations [18] improve efficiency, making slightly more frequent use of public-key infrastructures possible. For RFID tags with modest resources, researchers have proposed simple authentication and encryption to prevent "skimming", or physical proximity-based interrogation of tags [4].

In addition to neighbor-to-neighbor communication, many networks require secure broadcast and multicast communication. Often a control station must broadcast parameter changes to an entire network, and authentication of these messages is imperative. Both TESLA [19] (for ad hoc networks) and uTESLA [15] (for sensor networks) provide broadcast authentication. A base station commits to a chain of one-way hashes, and then uses each in reverse sequence as a key to authenticate a message. After the message has been distributed, the next key in the chain is released. Network nodes validate that $K_i = H(K_{i-1})$ and deliver the message. If all keys in the chain are exhausted, the base station must again securely distribute the commitment (last value) for a new chain to every network node.

LKHW [20] merges Logical Key Hierarchy (LKW) with directed diffusion, to provide secure multicast for groups in sensor networks. Directed diffusion is a routing protocol in which sinks diffuse interests for events, and sources send

messages along "interest gradients" that find all sinks. LKHW allows group membership to change and provides backward and forward secrecy.

Any protocol that uses cryptographic protection for confidentiality, integrity, or authentication relies on the presence of shared keys. Many approaches for creating and distributing these keys have been proposed. For public-key algorithms, a traditional centralized key distribution architecture may be used, such as Kerberos [21]. Centralized key distribution centers can become performance bottlenecks and attractive targets for attacks, however. By using threshold cryptography, the certification function is distributed among multiple authorities, such that at least *k* out of *n* are required to grant certificates [22]. This is more resistant to compromise than a centralized approach, but has higher overhead.

Ad hoc network devices often must collaborate together in groups, using secure multicast communication. In the Group Key Management Protocol (GKMP) [23], a centralized controller for each group generates and distributes pairwise keys to the other members. The Secure Spread service [24] provides multi-party key creation using Group Diffie-Hellman, in which each member contributes to the key.

Any scheme that requires cryptography also requires keys. Much research on key distribution in wireless sensor networks has focused on distributing secrets prior to deployment [25, 26, 27, 28]. Nodes are pre-loaded with multiple keys from a large key space. After deployment, nodes discover neighbors with whom they share keys, and use these paths to indirectly establish keys with other neighbors. Adding the requirement for nodes to share *q* common keys improves the protocols' resistance to compromise. Other protocols, such as LEAP [29], use a globally shared key to create pairwise-shared keys with neighbors during a short initialization period. The network is assumed to be free from compromises during this time, and the global key is erased thereafter.

RFID tags only interact with readers and certain special purpose tags, so the security concerns mostly center on identification and authentication. To prevent cloning attacks, a tag may implement lightweight symmetric cryptography or hashing and be programmed with a unique key. A challenge-response protocol prevents replay attacks, and provides simple identification or authentication. With the most constraints on size and cost, RFID tags are often vulnerable to physical attacks such as those described in the previous section.

Tags that respond to any reader or that respond using the same ID or key pose privacy risks. Weis et al propose using key-search techniques to conceal a tag's identity from any except legitimate readers [30]. The reader receives $H(k_i, N)$ from a tag, for key k_i and nonce N, and searches through all keys known to the reader for a match, identifying tag i. This is expensive for large numbers of tags, however. Others propose tree-based and synchronization-based schemes to limit the searching necessary, for example, by computing outputs based on an increasing counter.

Ad hoc and sensor networks use devices connected together wirelessly for multi-hop routing. The use of redundant, dynamic routing paths provides protection against link failures, but it increases the risk of relying on a compromised or adversarial node.

Approaches to securing ad hoc routing have focused on retrofitting existing protocols, or creating new ones to provide desirable properties. SAODV [31] provides authentication, non-repudiation, and integrity by means of a protocol extension to AODV that relies on digital signatures and hashing.

SEAD [32] also addresses security in distance-vector routing protocols, which are suitable for networks with limited mobility. It uses hash chains to secure routing updates, an approach that is more computationally efficient than SAODV and which provides some defense against denial of service attacks.

Protocols such as SEAD and SAODV rely on periodic routing updates, which have high overhead or poor performance when node mobility is high. In these networks, on-demand protocols like Ariadne [33] may be more suitable. Ariadne provides secure on-demand routing based on the DSR protocol, and requires one of: network-wide pairwise-shared keys, neighborhood pairwise-shared keys and broadcast authentication (such as TESLA), or digital signatures.

Wireless sensor networks require very efficient routing mechanisms, since radio transmission consumes so much of their energy budget. Their unique characteristics also pose special difficulties for secure routing [34]. Addressing the many attacks given the constraints of low-end sensor devices is problematic.

Aggregation of information to a centralized base station is a common communication pattern in WSNs. The authors of SPINS [15] suggest using underlying secure unicast and broadcast links (SNEP and uTESLA, respectively) to form routing trees from nodes back to base stations. LKHW targets communication within groups of collaborating devices (as described above), and secures directed diffusion for routing.

SIGF [35] is a family of routing protocols for WSNs, that allows very lightweight operation when no attacks are present, and stronger defenses—at the cost of overhead—when more attacks are detected. It is a form of on-demand routing based on geographic forwarding, where the message destination is specified as a location toward which each hop makes progress. The set of candidates considered at each hop may be increased, and their selection is randomized to prevent persistent selection of neighboring compromised nodes.

INSENS [36] is an intrusion tolerant protocol for WSNs that need little or no sensor-to-sensor communication, but which have well-defended base stations. Network topology is collected from sensor devices by the base station. Routes are computed centrally and securely distributed to sensors using one-way hash chain sequence numbers, similarly to uTESLA.

Middleware and Applications

Above the networking layers, which are concerned with relatively low-level details of inter-connection, middleware and application-layer software provide rich and

varied services. Networks connected to the physical world must provide mechanisms for extracting important data to authorized parties for analysis and manipulation. Several protocols have been proposed for querying, aggregating, and validating sensor data collected by WSNs.

Secure Information Aggregation (SIA) [37] uses special nodes in the network to aggregate sensor data. As data are collected and aggregated, results are reported to the base station along with a commitment to the data. Commitments are formed using a binary Merkle hash tree, which reduces the size of the verification information. The base station may then request particular sensor values from the aggregator in an interactive proof, until results are verified with a desired probability.

For large-scale networks where events of interest are witnessed by multiple sensors, Ye at al. propose Statistical En-route Filtering (SEF) of injected false data [38]. Nodes compute message authentication codes which are aggregated and sent with the reported data. Intermediate nodes check the MACs probabilistically, dropping incorrect messages. A Bloom filter is used to decrease the cost of aggregating multiple MACs.

Reprogramming widely distributed systems is expensive if it requires manual retrieval and manipulation of unattended devices. Over the network reprogramming alleviates this practical difficulty, but presents significant security concerns. All other hardware and software defenses may be subverted by a flaw that allows an attacker to replace nodes' programs with custom code.

Deng et al. [39] propose related schemes for securely distributing code in WSNs. The first uses a chain of hashes, where each message contains segment *i* of code and a hash of segment *i+1*. Upon receipt of a message, the previous code segment can be immediately and efficiently verified. To bootstrap the chain, an ECC signature of the first hash value is computed using the base station's private key. This method is suitable when there is little message loss and packets are received mostly in order. The second scheme uses a hash tree to distribute all the hashes in advance, so that out-of-order packets can be quickly checked. Resistance to denial of service is improved since packets need not be stored if they are corrupt.

SCUBA [40] is a protocol for detecting and recovering compromised nodes in sensor networks. Base stations verify code images on nodes using an Indisputable Code Execution (ICE) facility, which ensures that unmodified self-checksumming code runs on the target in the expected time. The ICE code computes checksums over the ROM, ICE function, and main executable. Incorrect checksums or executions that take too long indicate that malicious code is interfering with proper operation of the device. The result of the full SCUBA protocol is a repaired or blacklisted node.

Many applications may be built upon the foundations provided by the protocols we have reviewed: physical and radio-layer protections, secure node to node communication, multi-hop routing, data aggregation, and code updating. System designers must determine the attack model most appropriate for their application

domain and deployment environment, carefully choosing protocols that defend against possible attacks and which do not create additional points of vulnerability.

Global Research and Funding

NSF's Embedded and Hybrid Systems (EHS) Program [41] supports research in many aspects of embedded systems technology. A pervasive theme of the EHS program is the high-confidence integration of real-time and other service guarantees with the coordination requirements of next-generation complex, secure, networked, embedded systems.

NSF's Cyber Trust (CT) Program [42] envisions computer networks that are more reliable, accountable, and resistant to attacks, and a workforce that is well-trained and educated to operate them. Research proposals are solicited that will target security for applications, security for computer systems, security for networks, and new security foundations. The entire system life cycle may be considered, and multi-disciplinary projects with behavioral and social science participation are encouraged.

The European ARTIST2 Consortium [43] supports the Network of Excellence on Embedded Systems Design, which intends to strengthen European research in this area. The Testing and Verification cluster targets verification of security properties in designs.

Critical Needs Analysis

Embedded systems have already become ubiquitous, but their composition into large-scale systems for monitoring and controlling the physical world is nascent. Advancements in this field will enable many advanced applications, such as: secure communication for emergency personnel, disaster-site coordination, border patrol, container tracking and inspection, biological and radiological sensing, traffic control, and civil infrastructure monitoring. Realization of these critical applications will be subject to research progress on many technical fronts, including security and privacy.

Research Directions

Physical-layer security is often a weak spot in embedded devices even when higher layers are provably sound. Tamper-resistant packaging and designs for smart card, RFID, and sensor devices will be necessary for ubiquitous deployments and deserve more research.

Wireless devices expose the system to monitoring by and remote interaction with attackers. More research in resistance to denial of service attacks by jamming, flooding, and invoking expensive computations is needed to enable continued operation of critical components even while attacks are ongoing.

Connecting virtual and physical worlds raises many privacy concerns. Controversies over RFID-enabled passports and banknotes, urban camera networks, tracking of consumer-products post-sale, and disclosure of aggregated

data by companies and government agencies all portend a complex future of interdependent technical, legal, and political effects on personal privacy. More fundamental research is needed in ways to preserve privacy despite the collection of unprecedented amounts of data in the public and private sectors.

Data collected by wireless sensor networks will be useful for many purposes, but may inadvertently disclose sensitive information—even if the data payloads in-network are encrypted. Traffic analysis or simple radio-activity detection may reveal to an attacker whether a home is occupied, the nationality of a traveler in a crowd, or the location of important control devices. Comprehensive research that crosses traditional functional layers and includes non-cryptographic approaches to information hiding is needed.

References

[1] NIST. Glossary of Key Information Security Terms. Kissel R, ed. NIST IR 7298. April 25, 2006.

[2] Royer, E, Toh, C. A. 1999. Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. IEEE Personal Communications 6 (2): 46-55.

[3] Akyildiz, IF, Su, W, Sankarasubramaniam, Y, Cayirci, E. 2002. Wireless Sensor Networks: A Survey. Computer Networks, 38(4): 393-422.

[4] Juels A. 2005. RFID Security and Privacy: A Research Survey. Technical Report. RSA Laboratories. pp. 1-19.

[5] Perrig, A, Stankovic, JA, and Wagner, D. 2004. Security in wireless sensor networks. Communications of the ACM 47 (6): 53-57.

[6] Ravi, S, Raghunathan, A, Kocher, P, and Hattangady, S. 2004. Security in embedded systems: Design challenges. Trans. on Embedded Computing Sys. 3 (3): 461-491.

[7] Anderson, R, and Kuhn, 1996. M. Tamper Resistance - A Cautionary Note. Usenix Workshop on Electronic Commerce. pp. 1-11.

[8] Ravi, S, Raghunathan, A, Chakradhar, S. 2004. Tamper Resistance Mechanisms for Secure, Embedded Systems. In Proc. of 17th International Conference on VLSI Design. p. 605.

[9] Suh, G, Clarke, D, Gassend, B, van Dijk, M, and Devadas, S. 2003. AEGIS: Architecture for Tamper-Evident and Tamper-Resistant Processing. In Proc. of ICS. pp. 168-177.

[10] Xu, W, Trappe, W, Zhang, Y, Wood, T. 2005. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In Proc. of MobiHoc. pp. 46-57.

[11] Law, YW, Hartel, PH, den Hartog, JI, and Havinga, PJM. 2005. Link-layer jamming attacks on S-MAC. In Proc. of EWSN. pp. 217-225.

[12] Wood, AD, Stankovic, JA, and Son, SH. 2003. JAM: A Jammed-Area Mapping Service for Sensor Networks." In Proc. of IEEE RTSS. pp. 286.

[13] Kent, S, and Seo, K. 2005. Security Architecture for the Internet Protocol. IETF RFC-4301.

[14] Dierks, T, Allen, C. 1999. The TLS Protocol, Version 1.0. IETF RFC-2246.

[15] Perrig, A, Szewczyk, R, Wen, V, Culler, D, and Tygar, JD. 2001. SPINS: Security protocols for sensor networks. In Proc. of MobiCom, pp. 189-199.

[16] Karlof, C, Sastry, N, and Wagner, D. 2004. TinySec: A link layer security architecture for wireless sensor networks. In Proc. of SensSys. pp. 162-175.

[17] Watro, R, Kong, D, fen Cuti, S, Gardiner, C, Lynn, C, and Kruus, P. 2004. TinyPK: securing sensor networks with public key technology. In Proc. of SASN, pp. 59-64.

[18] Malan, DJ, Welsh, M, and Smith, MD. 2004. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In Proc. of SECON.

[19] Perrig, A, Canetti, R, Tygar, D, and Song, D. 2002. The TESLA broadcast authentication protocol. In RSA Cryptobytes, v. 5.

[20] Pietro, RD, Mancini, LV, Law, YW, Etalle, S, and Havinga, PJM. 2003. LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks. In 32nd International Conference on Parallel Processing Workshops (ICPP 2003 Workshops).

[21] Steiner, JG, Neuman, C, and Schiller, JI. 1988. Kerberos: An authentication service for open network systems. In Proc. of USENIX, pp. 191-200.

[22] Zhou, L, Haas, Z. 1999. Securing ad hoc networks. IEEE Network. 13 (6): 24-30.

[23] Harney, H, and Muckenhirn, C. 1997. Group Key Management Protocol (GKMP) Architecture. IETF RFC 2094.

[24] Amir, Y, Kim, Y, Nita-Rotaru, C, Schultz, JL, Stanton, J, and Tsudik, G. 2004. Secure Group Communication Using Robust Contributory Key Agreement. IEEE Transactions on Parallel and Distributed Systems, 15 (5): 468-480.

[25] Du, W, Deng, J, Han, YS, and Varshney, PK. 2003. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In Proc. of ACM CCS. pp. 42-51.

[26] Chan, H, Perrig, A, and Song, D. 2003. Random key predistribution schemes for sensor networks. In IEEE Symposium on Security and Privacy. pp. 197-213.

[27] Liu, D, and Ning, P. 2003. Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks. In Proc. of NDSS. pp. 263-276.

[28] Eschenauer, L, and Gligor, VD. 2002. A key-management scheme for distributed sensor networks. In Proc. of ACM CCS. pp. 41-47.

[29] Zhu, S, Setia, S, and Jajodia, S. 2003. LEAP: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks. In Proc. of ACM CCS.

[30] Weis, S, Sarma, S, Rivest, R, and Engels, D. 2003. Security and privacy aspects of low-cost radio frequency identification systems. In Hutter, D, Mueller, G, Stephan, W, and Ullmann, M, eds. International Conference on Security in Pervasive Computing (SPC 2003). v. 2802 of Lecture Notes in Computer Science. pp. 454-469. Springer-Verlag.

[31] Zapata, MG. 2001. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. IETF Internet Draft, draft-guerrero-manet-saodv-00.txt.

[32] Hu, Y-C, Johnson, DB, and Perrig, A. 2002. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In Proc. of the Fourth IEEE Workshop on Mobile Computing Systems and Applications.

[33] Hu, Y-C, Perrig, A, and Johnson, DB. 2002. Ariadne: A secure on-demand routing protocol for ad hoc networks. In Proc. of ACM MobiCom.

[34] Karlof, C, and Wagner, D. 2003. Secure routing in wireless sensor networks: attacks and countermeasures. In First IEEE International Workshop on Sensor Network Protocols and Applications.

[35] Wood, AD, Fang, L, Stankovic, JA, and He, T. 2006. SIGF: A Family of Configurable, Secure Routing Protocols for Wireless Sensor Networks. In Proc. of SASN.

[36] Deng, J, Han, R, and Mishra, S. 2005. INSENS: Intrusion-Tolerant Routing For Wireless Sensor Networks. Elsevier Journal on Computer Communications, Special Issue on Dependable Wireless Sensor Networks. 29 (2): 216-230.

[37] Przydatek, B, Song, D, and Perrig, A. 2003. SIA: Secure information aggregation in sensor networks. In Proc. of ACM SenSys.

[38] Ye, F, Luo, H, Lu, S, and Zhang, L. 2004. Statistical en-route detection and filtering of injected false data in sensor networks. In Proc. of IEEE INFOCOM.

[39] Deng, J, Han, R, and Mishra, S. 2006. Secure Code Distribution in Dynamically Programmable Wireless Sensor Networks. In Proc. of ACM/IEEE IPSN. pp. 292-300.

[40] Seshadri, A, Luk, M, Perrig, A, van Doorn, L, and Khosla, P. SCUBA: 2006. Secure Code Update By Attestation in Sensor Networks. In Proc. ACM WiSe. pp. 85-94.

[41] NSF's Embedded and Hybrid Systems (EHS) Program, URL: http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=5139

[42] NSFs Cyber Trust (CT) Program, URL: http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=13451

[43] European ARTIST2 Consortium, URL: http://www.artist-embedded.org/FP6/

Further reading list

Anderson, R. 2001. Security Engineering. John Wiley & Sons, New York.

Karl, H, and Willig, A. 2005. Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, England.

Zhao, F, and Guibas, L. 2004. Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, San Francisco, CA.

Murthy, CSR, and Manoj, BS. 2004. Ad Hoc Wireless Networks: Architectures and Protocols. Prentice Hall Ptr, Upper Saddle River, New Jersey.

Basagni, S, Conti, M, Giordano, S, and Stojmenovie, I, eds. 2004. Mobile Ad Hoc Networking. Wiley-IEEE Press.

Cross-references

RA40 Physical security: models and countermeasures

SD11 Protecting the protection systems

CS32 Protocol security

CS34 Wireless security

AG48 Tracking and tracing

Glossary terms

Side-channel, jamming, skimming, challenge-response