Revisiting Public-Key Cryptography for Wireless Sensor Networks

Benjamin Arazi
University of Louisville
Itamar Elhanany, Ortal Arazi, and Hairong Qi
University of Tennessee, Knoxville

Recent advancements in the design and fabrication of low-power VLSI circuitry, along with wireless communications, have broadened the applications prospects for *wireless sensor networks* (WSNs).

These promise to revolutionize our ability to sense and control diverse physical environments using many small, inexpensive devices that integrate sensing, computation, and communication. These sensors can collaborate with each other and perform complex information-gathering and dissemination tasks such as infrastructure security, environment and habitat monitoring, industrial sensing, and traffic control.

The collaborative processing between nodes necessitates the ad hoc formation of node clusters. These clusters typically emerge in the vicinity of a monitored event. Cluster members must be decided upon ad hoc given that the event's location and extent are often unknown in advance.

Many WSN applications that span military and civilian use assume that the sensor nodes will be deployed in hostile environments and thus be prone to a wide variety of malicious attacks. As a result, security becomes a key concern.

An efficient key distribution mechanism can help keep wireless sensor networks secure.

The lack of a fixed infrastructure and the ad hoc nature of WSN deployments suggest that the ability to encrypt and decrypt confidential data among arbitrary sensor nodes while enabling undisputed authentication of all parties will be a fundamental prerequisite for achieving security. To do this, nodes must be able to establish a secret key and know who their counterparts are. Thus, it becomes highly desirable to have a secure and efficient distribution mechanism that allows simple key generation for large-scale sensor networks while facilitating all the necessary authentications.

FEASIBILITY AND OPPORTUNITIES

Although a variety of key-generation methods have been developed, they cannot be directly transplanted to sensor network environments given the unique attributes of WSNs: limited power, memory, and computation resources, as well as a dynamically varying network configuration. A naïve solution for key establishment might use a single network-wide shared key. Unfortunately, capturing even a single node in the network would easily reveal the network's secret key.

Given its steep computational cost, *public-key cryptography* has been dismissed as a pragmatic key-generation technology for WSNs. Until recently, mainstream efforts focused on the design and analysis of random schemes in which the network issues a different set of pre-established keys to each node, thereby reducing the probability that capturing one node will jeopardize the entire network. These schemes offer a partial solution with respect to scalability, cryptographic robustness, and the ability to append and revoke security attributes.

However, the necessity for public-key cryptographic key generation in WSNs cannot be ignored for two main reasons:

- Scalability is compromised if keys are predistributed based on future predictions of node deployment or if a centralized entity manages the key-generation process.
- Given WSNs' ad hoc nature, online central management is impractical.

Recent work has challenged the notion that Diffie-Hellman and public-key-based schemes are infeasible in WSNs. Researchers have demonstrated that basic *elliptic curve cryptographic* (ECC) key generation can in fact be attained and executed on resource-constrained sensor nodes in reasonable time and with predictable improved performance. ECC has thus emerged as a suitable public-key cryptographic foundation that provides high security for relatively small key sizes.

Some, however, challenge the need for asymmetric key generation by arguing that collaborative processing at the application layer can efficiently identify and discard false information fed by adversarial network nodes.

Although this argument is conceptually true, the nodes communicate over a shared untrusted channel, which creates a vulnerability that malicious entities can exploit. For example, an adversary can inject packets that appear to originate from legitimate sources. Even though sensor nodes commonly employ majority-based decisions as part of their collaborative information processing, a single node can cause an entire cluster to generate inaccurate decisions. False-alarm situations can occur easily, which wastes many precious resources. In addition, a node can cause a real emergency situation to go unreported.

In light of this, authentication could play a more important role than confidentiality. This is reinforced by applications that perform collaborative processing, in which validating the nodes' identity could carry more weight than concealing message payload.

Although the past 10 years have seen important advances in intrusion-detection techniques, such methodologies are impractical in the WSN context thanks to the heavy memory and computational resources required. Consider these two possible detrimental scenarios in which an adversary

- compromises a node by capturing and reprogramming it to exploit the network's vulnerabilities, or
- uses a node external to the network in an attempt to insert false information.

In both cases, a denial-of-service attack can be launched through extensive occupation of the shared media. However, these DoS attacks can be identified easily, whereas attacks that involve node imitation typically make localization and isolation difficult. Conducting trusted collaborative processing requires relying on a robust security layer.

A WSN security protocol should incorporate some degree of redundancy to account for potential loss of nodes.

PRAGMATIC CONSIDERATIONS

In contrast to wired networks, wireless networks might consist of links with different characteristics. The nature of sensor networks suggests that geographical deployment of the nodes can significantly impact security requirements. Moreover, advanced applications could necessitate heterogeneous sensor networks that consist of nodes with vastly different resources. Consequently, while large key sizes of, for example, 163 bits in ECC, are a stringent requirement in many wired network applications, a mixture of large and smaller keys might be appropriate in WSNs. This approach yields increased speed and reduced power and memory consumption. In such cases, some nodes could manage keys of different lengths.

Although a smaller key offers a less secure link, employing ephemeral- as opposed to fixed-key generation can help boost security. In ephemeral-key generation, specific nodes generate a different session key whenever they attempt to establish a secure link. This renders the process of uncovering confidential information considerably harder. Whereas relatively small session keys can be appropriate in many cases, system keys like the certifying authority's private key should be kept large and protected accordingly. To that end, processing large and small keys within unified mathematical expressions becomes a principal issue.

Once the feasibility of executing core ECC operations in WSNs has been demonstrated, a natural next step would be to identify the distinct attributes of sensor networks that can be exploited to further enhance efficient implementation. One approach might involve taking advantage of the collaborative processing ability, which efficiently partitions the computational load into disjoint calculations that can be distributed among several nodes. However, given that no entity in the network—let alone the communications channel—can be trusted, special care must be given to protecting the nodes from potential attackers that exploit such offloading techniques. Recent work suggests that calculations involved in ephemeral-key generation can indeed be partitioned into secure and nonsecure calculations, thereby opening the door to potential efficient realizations.

Sensor node vendors have begun working to incorporate security primitives into CPUs embedded in next-generation nodes. Such primitives offer the potential to further optimize the overall key-generation process by facilitating, for example, efficient execution of core ECC operations—such as scalar-point multiplication—beyond performances published so far. Cost-efficiency will play a key role in determining the extent to which security primitives will be incorporated in next-generation sensor nodes.

Fault tolerance is yet another crucial aspect of any security protocol. Given the nature of their use, sensor nodes are inherently susceptible to physical impairment. This means that whatever security protocol is devised, it should incorporate some degree of redundancy

to account for potential loss of nodes. A cluster of nodes that shares a joint secret key must be robust to the loss of one or more nodes. It is expected that a computation and robustness tradeoff will emerge in practical protocol designs.

EFFECIENCY THROUGH SELF-CERTIFICATION

In self-certified public-key cryptographic applications, a user submits an ID along with its public key, but does not submit an explicit certificate. This reduces communication and management overheads, a vital consideration in WSNs. The validity of a user's public key, specifically that the public key is associated with the user's ID, can be established through implication. As with any public-key cryptographic application, self-certification requires an explicit reference to the public key of a certifying authority or agent.

Current efforts focus on exploiting computational offloading in the context of self-certified key generation. Group self-certification introduces further intricacies, as it requires multiple identity validations while retaining robustness with respect to the potential loss of member nodes.

The issue of managing the keys, in concert with validating the identity of the communicating parties, becomes even more intricate for asymmetric nodes, such as sensor nodes and sink-based nodes. An efficient security protocol design would exploit the abundant resources of sink-based nodes to relieve resource-constrained entities as much as possible.

On both the hardware and software fronts it appears that much work remains before WSN applications can widely utilize resource-efficient public-key cryptography. Nevertheless, the potential gains are too compelling to be ignored, thus they merit further research. Such work must strive for innovative techniques and methodologies that exploit the unique attributes of WSNs in an effort to make a public-key infrastructure a reality. ■

Benjamin Arazi *is a visiting professor in the Computer Engineering and Computer Science Department at the University of Louisville. Contact him at ben.arazi@louisville.edu.*

Itamar Elhanany *is an assistant professor in the Electrical and Computer Engineering Department at the University of Tennessee, Knoxville. Contact him at itamar@ ieee.org.*

Ortal Arazi *is a PhD student in the Electrical and Computer Engineering Department at the University of Tennessee, Knoxville. Contact her at ortal@ ieee.org.*

Hairong Qi *is an associate professor in the Electrical and Computer Engineering Department at the University of Tennessee, Knoxville. Contact her at hqi@utk.edu.*

Editor: Wayne Wolf, Dept. of Electrical Engineering, Princeton University, Princeton NJ; wolf@princeton.edu