Network Security

Professor Gaurav S. Kasbekar
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Week - 12

Lecture - 73

Security of the Internet of Things (IoT), Hardware Security: Part 3

Hello, in this lecture, we continue our discussion of the security of Internet of Things and hardware security. An important part of hardware security is tamper resistance and detection. The IoT ecosystem involves a large number of devices deployed in an edge network. They are close to the users, and hence they are prone to tampering by users. So, these IoT devices are from numerous manufacturers and deployed in areas where physical security is difficult.

Hence, they need to be tamper-resistant, and any tampering has to be detected. Two essential security measures in such an environment are tamper resistance and tamper detection. What is the meaning of tamper resistance and tamper detection? Tampering is any unauthorized modification that alters the intended functioning of a system or device that degrades the security it provides. So, by tamper resistance we mean the following.

Tamper resistance is a characteristic of a system component that provides passive protection against an attack. And by tamper detection we mean that these are techniques to ensure that the overall system is made aware of any possible tampering or unwanted physical access. First, we discuss tamper resistance, and then we discuss tamper detection. A common approach to tamper resistance is to use specialized physical construction materials to make tampering with a fog node difficult. Examples are hardened steel enclosures, locks, and security screws.

So, these can be used to make tampering difficult. Another way is tightly packing components and circuit boards within an enclosure, which increases the difficulty of using fiber optics to probe inside the node without opening the enclosure. So, that's another means to achieve tamper resistance. A second category of tamper resistance is deterrence of tampering by ensuring that it leaves visible evidence behind. For example, we can use special seals and tapes that make it obvious when there has been physical tampering.

This tends to deter intruders from trying to tamper with the devices because if they tamper with the devices, then there'll be evidence left behind, such as a seal has been broken or some tapes have been altered, and so on. This is another category of tamper resistance. Now, we will discuss tamper detection. Mechanisms for tamper detection include the following. One is switches.

A variety of switches, such as mercury switches, magnetic switches, and pressure contacts, can detect the opening of a device, the breach of a physical security boundary, or the movement of a device. These switches can help in detecting tampering. So, when some tampering occurs, then the switch flips, and through that we detect the presence of tampering. Then sensors can similarly be used. Temperature and radiation sensors can detect environmental changes.

Voltage and power sensors can detect electrical attacks. Another means for tamper detection is circuitry. It's possible to wrap components with flexible circuitry, resistance wire, or fiber optics so as to detect a puncture or break. So, this concludes our discussion of tamper resistance and detection. Next, we discuss lightweight cryptography, which is another technology very relevant to IoT devices.

We discussed that IoT devices are resource-constrained. They have limited battery power, processing abilities, and storage capabilities. Hence, the cryptography that we use in such devices has to be lightweight. So, lightweight cryptography is focused on developing algorithms which, while secure, minimize execution time, memory usage, and power consumption. Such algorithms are suitable for resource-constrained devices such as those in IoT and small embedded systems, which are used in IoT extensively.

Work on lightweight cryptography is devoted to symmetric key algorithms and cryptographic hash functions. So, lightweight cryptography includes attempts to develop efficient implementations of conventional cryptographic algorithms as well as to design new lightweight algorithms that are specifically designed for IoT context. Let us briefly discuss constrained devices, and then we will discuss lightweight cryptography. A constrained device is one with limited volatile and non-volatile memory. It also has limited processing power and a low data rate transceiver, such as 802.15.4, which can provide data rates up to 250 kbps.

Many devices in the IoT, particularly the smaller, more numerous devices, are resource constrained. Typical constrained devices are equipped with eight- or 16-bit microcontrollers that possess very little RAM and storage capacities. As far as

communication is concerned, resource-constrained devices are equipped with IEEE 802.15.4 radios often, which enable low-power, low-data-rate wireless personal data networks with data rates of, which are very low. For example, 20 to 250 kbps and frame sizes of around 127 bytes. So, compare this with data rates in the case of Wi-Fi, for example, which are very high, several gbps in the latest standards, or several tens of gbps.

And the frame sizes can be several thousands of bytes or tens of thousands of bytes in the case of regular devices such as desktops and laptops. So, in contrast, resource-concerned devices exchange data at the rate of up to 250 kbps in a typical case, and frame sizes are also very small, up to 127 bytes. Now, what are the design trade-offs involved in designing lightweight cryptography techniques? This figure illustrates the trade-offs between security, cost, and performance in designing lightweight cryptographic algorithms. So, if you want high security, then we need to use longer keys.

We discussed that symmetric key algorithms, for example, they use keys, so the key lengths have to be large for high security. And we discussed that often symmetrically algorithms operate through rounds. So, in each round we pass the data through some P boxes and S boxes. So, more such rounds have to be performed for better security. Also, increased silicon area is required so that we can process in parallel, for example.

Also, a lot of power consumption is required and reduced throughput. But these all the objectives are opposed to those required for achieving a low cost. For low cost, shorter keys are required, and fewer rounds and reduced silicon area and so on. So, we can see that these objectives for low cost are, many of them are different from those for achieving high security. And many of these objectives for high security also conflict with those required for high performance; for example, again for high performance, we require shorter keys, fewer rounds, and so on.

Hence, there is often conflict between these three objectives: security, low cost, and performance. So, in general longer the key and more the rounds the greater the security. In our discussion of symmetric key cryptography, we discussed an example cipher, and several rounds of the cipher are performed. So, the data is passed through the cipher n times. So, the larger the value of n, that is, more the number of rounds, the greater the security, and similarly, longer the key, the greater is the security in general, but this implies a reduced throughput in terms of the amount of plain text that is processed per time unit, as well as it results in increased power consumption.

Another trade-off is that the more complex an algorithm or its implementation is, the more security it can provide, but this generally requires increased silicon area, either for hardware implementation or software implementation. So, we can have a lot of hardware which performs the processing in parallel, which requires greater silicon area, or we require a powerful processor and do software implementation, which again requires larger silicon area. Hence, achieving greater security can degrade either cost or performance objectives, or both. So, that summarizes the trade-off between security, low cost, and performance. To meet the requirements of lightweight cryptography, a number of new algorithms have been proposed.

The typical characteristics of such lightweight algorithms are, they perform many iterations of simple rounds. So, computations that are performed in a round are quite simple compared to regular cryptography, which is used in devices such as desktops and laptops. Another characteristic is that simple operations such as XORs, rotation, 4×4, S-boxes, and bit permutations are performed. Smaller block sizes are used, such as 64 or 80 bits, in comparison to, for example, 128 bits or 256 bits for regular devices. Smaller key sizes are used, for example, 96 or 112 bits.

- Typical characteristics include:
  - ❑ Many iterations of simple rounds
  - ❑ Simple operations like XORs, rotation, 4×4 S-boxes, and bit permutations
  - ❑ Smaller block sizes (e.g., 64 or 80 bits)
  - ❑ Smaller key sizes (e.g., 96 or 112 bits)

And also smaller security margins are used by design. What is security margin of a cipher? It is the difference between the number of rounds in the complete implementation of the cipher and the maximum number of rounds that are known to be breakable using the best-known real-world attack. Consider a symmetric key cipher which uses n rounds. There are typically some real-world attacks which can break the cipher if n is less than a certain value.

So, the difference between the actual value of n used and the value of n which is breakable, so that's the security margin. So, the value of n that is typically used is more than the minimum value of n, which is safe against the best-known real-world attack. So, that is the security margin of a cipher. So, lightweight cryptography techniques typically use smaller security margins compared to regular cryptography. These design choices yield smaller security margins compared to established algorithms such as advanced encryption standard and secure hash algorithm too.

We now discuss some examples of lightweight block ciphers and cryptographic hash functions. An example of a lightweight cryptographic block cipher is the Scalable Encryption Algorithm (SEA). So, this is an example block cipher, which is symmetric key-based. Two ways in which hash functions differ from more traditional ones are: one, they have a smaller internal state and output sizes. Recall that we discussed the functioning of the SHA-1 cryptographic hash function earlier. It maintains an internal state, and then the final output is derived from that internal state.

- An example of a lightweight cryptographic block cipher is the Scalable Encryption Algorithm (SEA)
- Two ways in which lightweight hash functions differ from more traditional ones are:
  - ❑Smaller internal state and output sizes
  - ❑Smaller message (input) size
- An example of a lightweight cryptographic hash function is PHOTON

In the case of lightweight hash functions, the internal state is smaller than in the case of traditional hash functions, and also the output size is smaller. These lightweight hash functions also use a smaller message or input size. An example of a lightweight cryptographic hash function is PHOTON. We will not discuss the details of SEA or PHOTON in the interest of time. You can read up about them if you are interested.

So, in summary, we discussed different aspects of IoT security and hardware security, such as, for example, tamper resistance and tamper detection. We also discussed lightweight cryptography and some examples of symmetric key and hash functions, which are lightweight by design. This concludes our discussion of IoT security and hardware security. Thank you.