Security Goes Beyond Cryptography

Jon Titus, Senior Technical Editor

Encryption alone doesn't keep secrets. Engineers must assess overall system security needs.

(Click here to go to web only content.)

Anyone who thinks communication security applies mostly to financial, medical, and personal transactions should view a short video of the Aurora test CNN aired in September 2007 [1]. This video shows the destruction of a large electric generator that resulted from a controlled hacker attack on a power-plant control system. In another hack, a teenager in Lodz, Poland compromised the train system and caused numerous injuries when four trains derailed or stopped abruptly. (See: For further reading.)

“When engineers start to network devices, security becomes a top design requirement,” said Tim Stapko, lead software engineer at Digi International. “But many designers of embedded systems just don’t think about security. When they do, they might consider security as an add-in option or think of security as simply encrypting communications.”

Is Your Application “Authentic”?

Unfortunately, encryption does not offer enough protection. “Some time ago, people ‘cracked’ the Wired Equivalency Privacy [WEP] encryption scheme used in many WiFi connections,” said Vishal Kakkad, senior software engineer at Lantronix. “To enhance security, engineers must include authentication in their application. A receiving device must know whether data originates from a valid source, otherwise it could try to decrypt ‘garbage data’ sent by some random unit. So, your devices must know if they may connect to another device and accept packets from it.”

According to Stapko at Digi International, you can’t achieve a 100-percent secure system. “You simply make your device and its communications hard enough to break into so the bad guys go elsewhere. But you can overdo security and make your application unusable. Then security doesn’t help much.” To start, you must analyze the value of what you want to protect. Rudan Bettelheim, product Manager at Freescale Semiconductor noted, “If you have a vending machine that holds $1,000 of inventory, do you pay millions to protect it? But ask yourself, ‘What else is at risk?’ Communication keys and financial information stored in the machine may have more value than its inventory.” Bettelheim explained that criminals don’t steal ATMs for their money: they want security keys, credit- and debit-card information, and PINs. That information has more value than the cash.

Different Levels of System Security

Digi’s Stapko divides embedded-system security into several levels. “At the lowest level, you have a small microprocessor with limited CPU power and memory space, so you might use a simple redundancy check. You create a ‘hash’ — basically a checksum — for the data and send the hash value with the data. Then you produce a hash value at the receiving end and compare it to the sent hash value to verify that nothing tampered with the data in transit. Someone can look at everything that

goes down the line, but they can't change it without you knowing about it." The actual hashing technique gets more involved, but it follows this basic idea.

The next higher step involves using a pre-shared key and a simple encryption algorithm. The sender and the receiver both have the same key, and they use the same encryption/ decryption algorithm such as the Advanced Encryption Standard [AES] to send messages back and forth. This general technique goes by the name symmetric-key or secret-key cryptography. Here, symmetric refers to the use of the same secret key and algorithm at each end of a communication. "AES provides pretty good security but you must get the secret key securely from point A to point B," said Stapko. "When you have devices going into the field, you can program them all with a single key. But if someone attacks a device and obtains its key, then the attacker has the key for all your devices."

If that happens, you have to figure out how to redistribute a new key to each remote device. And if you have the same key in every device, you should change the key often. Better yet, devise a way to ensure each device has a unique key. "Continually updating the keys is good and everybody having a different key is good," said Stapko. "But you still need a way to distribute the keys so they are secure during that step." At the next higher security level, you can employ a Secure Sockets Layer [SSL] or Transport Layer Security [TLS] algorithm. "Both offer a complete protocol with authentication and integrity checks that take place over an encrypted channel," noted Stapko. SSL and TLS use public-key, or asymmetric-key, cryptography that does not require the pre-distribution of a secret key to each device. "The digital certificates used by SSL and TLS algorithms provide for authentication that verifies the identification of the connected devices," said Stapko. "After authenticating each other and exchanging secret keys using public-key encryption, the devices can use symmetric-key encryption to transfer large amounts of information." (See: "Get the Proper Security Certificate [2]")

Keys and Cryptography

"Symmetric-key cryptography works well because encrypting and decrypting operations can take place about 1,000 times faster than when you use a public-key algorithm," explained Freescale's Bettelheim. "But you must have the same secret key at both ends, so how do you get the keys securely to both ends? You use public-key cryptography to exchange the key." (A discussion of crypto techniques goes beyond the scope of this column; see some of the references in For further reading.)

"Even if a remote device has had no prior contact with a local device or server, the two can establish a secure communication link because a remote device can generate its own public/private key pair," said Bettelheim. "It keeps its private key secret and sends its public key to the other device. Say you want someone to send you something valuable. You send them an open padlock and keep the key. They secure your valuable item and when you receive it, your key unlocks it. They don't need the key and only you can unlock the padlock."

(Web only content starts here.)

Digi's Stapko places the type of security used in Wi-Fi communications at the top of his security list for embedded systems. "Wi-Fi applications use TLS to establish an encrypted connection over which devices can authenticate each other. Then they

use a standard method such as Wi-Fi Protected Access (WPA) or WPA2, or one of many non-standard or proprietary methods."

Digi's Stapko places the type of security used in WiFi communications at the top of his security list for embedded systems. "WiFi application use SSL to establish an encrypted connection over which devices can authenticate each other. Then they use a standard method such as WiFi Protected Access (WPA) or WPA2, or one of many non-standard or proprietary methods."

Don't Overlook These Important Issues

Even when engineers understand security issues, they can fail to consider important aspects of security:

1. Engineers should know about the types of network attacks their equipment could experience. "They could see a ping flood or a broadcast storm, for example, on a network," said Kakkad of Lantronix. "Usually a security-hardened communication stack will handle those types of attacks. But engineers must investigate networking hardware and software to ensure they provide the needed protection and that tests show they can withstand any common attacks."

2. A secure device does not require a proprietary protection method. "Engineers may think when they implement home-grown security protocols attackers will know nothing about it so they can't crack it," said Kakkad. "That's a bad assumption. Use a standard security method that people have researched and tested many times to ensure it provides the needed security."

3. Shut down unused communication ports. "When engineers buy a communication module such as our MatchPort b/g Pro, it comes with standard services such as Telnet, FTP, and Web access," explained Kakkad. "We leave these ports 'open' to ease setup by designers. But if you will not use these services, shut them down. You can still access the module to configure it via secure protocols such as SSH and HTTPS (SSL)."

4. Don't modify security algorithms. According to Digi's Stapko, an engineer might decide to modify the OpenSSL code to do something faster. "Suppose you use a static number as the random-number seed used to generate keys. It takes less CPU time to use a constant, say, 12345, than to generate a random number, but that breaks the security provided by SSL. Don't try to 'optimize' standard security methods."

5. Use stream ciphers with caution. A stream cipher, such as RC-4, uses a cipher bitstream created through exclusive-OR (XOR) operations. When used improperly, a stream cipher can reveal plain-text (unencrypted) information. "If you use the same key to encrypt two different messages, you can exclusive-OR the output and get the text back," noted Stapko. To prevent an XOR attack you must use an initialization vector; a secondary key that is a cryptographically secure random number. Researchers used the XOR technique to break WEP encryption, the original Wi-Fi encryption standard that preceded WPA."

6. Make sure the method you plan to use still provides security. “We consider WEP for Wi-Fi insecure, so you probably shouldn’t use it,” cautioned Stapko. “And SSL Version 2 is also considered insecure. Keep in mind your security method faces hundreds of PhD students trying to make a name for themselves by breaking security algorithms. Practically speaking, however, both WEP and SSL version 2 can be better than nothing, but don’t let that lull you into thinking your application is truly ‘secure’.” The Computer Emergency Response Team (www.cert.org [3]) at Carnegie Mellon University publishes advisories that cover current Internet security concerns. Stapko also recommends the Webpage maintained by cryptography guru Bruce Schneier (www.schneier.com [4]) as a source of security information.

Address Security In Every Aspect of Product Development

Although security efforts can prevent the extraction of useful information from network communications, engineers also must ensure the security of every aspect of product development. “Often you have people who develop code in some far-off place,” said Freescale’s Bettelheim. “They could insert rogue code into a library and with a sufficiently large build, it might go unnoticed. Although you take care to implement secure communications and keep keys secret, that rogue code might communicate internal information to an unauthorized location.” Bettelheim also explained that someone could insert rogue code or malware during field upgrades or maintenance.

In the embedded world, most suppliers do not provide tools that make password management easy and the task often this task falls upon an IT department. “But technicians who maintain cell-phone base stations, for example, don’t want to get involved with the IT department,” said Bettelheim. “So they use one password and they give it to everyone or they leave their systems set with the default password.” So security must involve more than just locking down communications.

An incompletely-secure system can cause future problems, too. “Apple’s first iPhone came with low-level security,” said Bettelheim. “The latest firmware significantly improved the level of protection. But hackers have loaded the previous versions of the firmware into the new iPhones so they can then install their own small routines. Then, they upgrade to the latest version of the firmware and use their loaded routines to hack the new software.”

For further reading

Schneier, Bruce, “Applied Cryptography: Protocols, Algorithms, and Source Code in C,” 2nd ed., Wiley. 1996.

Singh, Simon, “The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography,” Anchor. 2000.

Stapko, Tim, “Practical Embedded Security,” Newnes Press. 2008.

Aurora Generator Test: http://www.cnn.com/2007/US/09/26/power.at.risk/index.html [1]

Lodz, Poland Train Derailing: http://hackedgadgets.com/2008/01/24/simple-ir-hack-derails-polish-trains/ [5]

Internet Engineering Task Force, http://www.ietf.org/ [6] See the “Request for Comments” (RFC) section.

“Deploying Wi-Fi Protected Access (WPA) and WPA2 in the Enterprise,” Wi-Fi Alliance, March 2005. www.wi-fi.org/files/wp_9_WPA-WPA2 Implementation_2-27-05.pdf [7]

WPA2 (Wi-Fi Protected Access 2): www.wi-fi.org/knowledge_center/wpa2/ [8]

Extended EAP (Extensible Authentication Protocol): http://www.wi-fi.org/knowledge_center/eap/ [9]

Links:
[1] http://www.cnn.com/2007/US/09/26/power.at.risk/index.html
[2] http://www.ecnmag.com/article-security-certificate.aspx
[3] http://www.cert.org/
[4] http://www.schneier.com/
[5] http://hackedgadgets.com/2008/01/24/simple-ir-hack-derails-polish-trains/
[6] http://www.ietf.org/
[7] http://www.wi-fi.org/files/wp_9_WPA-WPA2%20Implementation_2-27-05.pdf
[8] http://www.wi-fi.org/knowledge_center/wpa2/
[9] http://www.wi-fi.org/knowledge_center/eap/