Security is the key

As embedded systems communicate more widely, they need to do so securely. By ***Graham Pitcher****.*

The world in which we live is driven – at least for the moment – by a thirst for connectivity. No matter what the product, people expect it to be able to communicate with something else.

The idea is taken almost to an extreme in the archetypal 'internet fridge'. Here, the fridge has intelligence and an IP address, which allows the fridge's user to interrogate it remotely. Conversely, the fridge can 'call home', asking for new supplies.

Thomas Cantrell is a software engineer within Green Hill Software's real time operating systems group. He said: "What we're seeing at the moment is a trend towards putting everything on the internet. It's almost as if people want to see how their washing machine is doing on the web. But the problem is that the internet is not secure."

Embedded systems are experiencing the same demand for connectivity as is found in more obvious market sectors. Even if secure communications is not important to an application, a number of conclusions can be drawn. Firstly, you don't know what's between your application and the one it's trying to connect with. Secondly, you can't guarantee that the data you send over that connection cannot be read by third parties.

There are, of course, solutions to the security aspect. But, according to Cantrell: "When engineers try to do security, things tend to go wrong." He said users tend to do things like send a password over the web. "They sometimes send a scrambled password, which isn't any better, or use something like data encryption standard (DES) encryption, which is still not good enough."

One of the solutions which Green Hills, amongst others, recommends is IPsec. This is a suite of protocols which aims to secure IP communications by authenticating and/or encrypting each IP packet in a data stream. The approach also includes protocols which allow for the establishment of cryptographic keys.

The IPsec protocols work in Layer 3 – the network layer – of the OSI seven layer model. This is in contrast to other security related protocols, which tend to operate in Layer 4 – the transport layer. Because IPsec protocols run in Layer 3, they can be used to protect Layer 4 protocols, including the widely used TCP.

"The good thing about IPsec," Cantrell observed, "is that it's done in the TCP/IP stack, so the applications don't know it's there. And, because it's in the IP layer, the application developer doesn't know it's there either." And, unlike other security protocols, IPsec doesn't need to be designed in.

Combinations

But, like most things in the communications world, IPsec isn't as easy as it would initially appear. "One of the hard things about IPsec is that there are four ways of doing it," Cantrell observed.

These four ways break down into two modes and two protocols. The two modes are tunnel mode and transport mode. The two protocols are authentication header (AH) and encapsulating security payload (ESP).

In transport mode, only the data being transferred is encrypted. In tunnel mode, however, the entire IP packet is encrypted. Where transport mode is generally used for host to host links, tunnel mode is also used for network to network communications. AH provides data integrity and authentication, whilst ESP provides confidentiality, with the option of adding authentication.

Over time, those dealing with IPsec have wondered why the four permutations couldn't be slimmed down to one.

Illustration: Vincent Fraser

Although that hasn't happened officially, Cantrell says 95% of IPsec transmissions use ESP and tunnel mode.

In Cantrell's opinion, one of the interesting things about IPsec is that it is based on cryptography, but it introduces new elements. "You can look at it as a 'black box'," he claimed, "but it helps to know what's going on inside."

Basic IP traffic follows a standard format. Packets are constructed with an IP header, followed by a TCP or UDP header, then data. In IPsec's ESP tunnel mode, a new header is attached, and this is followed by a new ESP header. The original data packet – IPheader, TCP header and data – then follows. The new packet is then terminated with an ESP trailer and authentication.

Cantrell described the operation. "ESP, but IPsec in general, wraps the original packet in order to provide additional security. It does this by providing authentication between two hosts and confidentiality for the data."

Authentication is provided by the use either of shared keys, which could be passwords with hash algorithms, or asymmetric encryption, where public and private keys are employed. Confidentiality, meanwhile, is provided through the use of symmetric encryption. Cantrell noted the approach also provides data integrity checks using hash algorithms and anti replay checks. "What that means," he added, "is that we don't see the same message twice."

The three main cryptographic approaches used with IPsec are symmetric and asymmetric encryption and hash algorithms.

Unlocking the information

Symmetric encryption sees the same key being used to encrypt and decrypt the data. Cantrell said: "The key is the only piece of information needed and the longer the key, the harder it is to crack." Examples of symmetric encryption algorithms include advanced encryption standard (AES) and triple DES. "But people are tending to move away from DES," Cantrell observed, "and if you had to choose, it should be AES."

Asymmetric encryption uses private key and public keys. The private key encodes data, which can be decrypted with the public key. However, data can also be encrypted with a public key and decrypted only with a private key. Examples of such algorithms include RSA and Diffie-Hellman. But an emerging body of algorithms use the elliptic curve approach.

Hash algorithms, meanwhile, take a block of data and produce what's called a message digest, which has a standard length. It's a one way process which allows two pieces of information to be compared. Standard applications include password verification and data integrity checks.

Sitting alongside IPsec is IKE – internet key exchange. "What this allows," said Cantrell, "is IPsec parameters to be exchanged. It supports authentication and negotiation of IPsec parameters transparently and an important thing to note here is that IKE generates the symmetric key needed by both sides of an IPsec link." IKE can also support key rotation, which means the symmetric key is changed periodically for security reasons.

"You can look at (security) as a 'black box', but it helps to know what's going on inside.",

Thomas Cantrell, *Green Hills Software*

In most IPsec implementations, there is an IKE program running in the background, whilst the IPsec stack runs in the kernel where IP packets are processed. Because it is in the user space, IKE has easier access to configuration parameters, whilst IPsec can process packets efficiently. IKE uses UDP packets to establish a secure link, at which point, the key is passed to the IPsec stack.

In Green Hills' view, security and reliability are now intertwined and a system cannot be available and reliable unless it is hardened against attacks. It is for that reason that approaches such as IPsec are now firmly embedded in Green Hills' Platform for Secure Networking. ■

Figure 1: How IPsec works