by S. Lee Henry

HANK OSUNA

Message Authentication

Last month's column was an introduction to authentication. The three major varieties (or uses) of the technology–user identity authentication, message origin authentication and message content authentication–were introduced before we went into more detail on the means by which users are authenticated.

This month, we look at message origin (sender) and message content (integrity) authentication. To begin with, let's briefly examine cryptography, the essential technology behind authentication. Those of us not working with message authentication may wonder about the connection between cryptography and authentication–what it is and how it works to "prove" that a message hasn't been altered or that it is from the purported sender.

O'Reilly & Associates' *Dictionary of PC Hardware and Data Communications Terms* provides succinct and useful definitions of encryption and related terms. It refers to encryption as "the process of changing a digital message (from plain text to cipher text) so that it can be read only by the intended parties (also called enciphering), or to verify the identity of the sender (authentication), or to be assured that the sender really did send that message (nonrepudiation)." The dictionary also clearly distinguishes private from public keys.

Cryptography

Private key (also referred to as "secret" or "symmetric" key) systems use the same key for both the encryption and decryption processes. As a result, the content of files encrypted with this technology can only be read by those who know the encryption key–presumably the sender and chosen recipients.

Public key systems, on the other hand, use two different keys. One of these keys is more or less public, the other is secret. One key is used to encrypt the text, and the other is used to decrypt the text. The two keys have a relationship to each other in that each reverses the process that the other invokes. As a result, either key can be used as the encryption key as long as the other is used for decryption. This enables the technology to be used in two related but fairly opposite ways.

If the private key is used for encryption and the public key is used for decryption, we have a situation in which only the sender of a message could have encrypted it, but anyone (anyone with access to the public key, that is) can decrypt it. Whereas, if the public key is used for encryption and the private key is used for decryption, we have a situation in which anyone

Disclaimer: Regardless of everything that was said last month, proofs of identity based on what a person knows or has (not so much what he is) are spoofable to the extent that what he knows and has can be shared. If we whisper our passwords in our sleep or tell them to our friends because they're oh so clever, and leave our SecurID cards on our desks, no process on our systems will be able to tell the difference between us and the spoofers.

can send an encrypted message to the holder of the secret key, who is the only one who can read it. Each of these approaches is, in fact, quite useful.

What we have described so far suggests that public key systems generally involve the encryption and decryption of entire messages. Actually, this is not the case. It is more common for this technology to be used to generate what is called a digital signature, which is then attached to unencrypted messages as proof of their authenticity. The message itself is generally sent in clear text.

Digital Signatures

What are digital signatures, and how do they work? Well, to begin, let's briefly examine the motivation. One of the primary reasons that cryptography works so well is because the transformations from plain text to cipher text are easy to do but hard to reverse. The underlying mathematics particularly capitalizes on this feature in the functions that make up the encryption tools. At one end of the correspondence, there's some heavy computation going on, more than we'd likely want to encumber for every piece of email. Besides, if it is sender authentication that we're after, we might want to make it *available* to the recipient without making it *mandatory* to go through the throes of proving the sender's authenticity before reading the message. Most of us would prefer to save ourselves the trouble except for our more sensitive electronic correspondence.

With digital signatures, a quick "pass" (similar to generation of a checksum) over the message text creates a profile or "hash" of the message. The range of values is large enough that any changes to the text practically guarantee a change of the resultant hash value.

The hash value is then encrypted with the sender's private key. The result is that anyone can 1) generate the same hash value from the message, 2) decrypt the digital signature using the sender's public key and 3) compare the computed and decrypted hash values to verify the integrity of the message content and the authenticity of the sender.

One of the key benefits of public key systems is that, unlike typical user authentication in which passwords are transmitted over the network in clear text, the private and the public keys are not transmitted (at least not in any context of the message itself) and are, therefore, not susceptible to being sniffed. The flip side of this, of course, is that public keys still have to be managed somehow. They need to be distributed, changed and inactivated as required.

Once public key systems are in popular use, we're likely to see some methodology for locating and verifying public keys as well. It is entirely possible, if strong approaches are not developed for public key management, that fraudulent public keys will be distributed as a means of authenticating impostors.

***S. Lee Henry** is on the board of directors of the Sun User Group and is a security services engineer at Infonet, a provider of virtual private networking, in El Segundo, CA. Email:* `slee@cpg.com`.