Keywords: security, encryption, crypto, message, protection, secrecy

APPLICATION NOTE 5973

FUNDAMENTALS OF ELECTRONIC SECURITY: SECURITY IN TRANSIT

By: Ben Smith, Software Manager

Abstract: Any time you employ a medium to transmit your message, you lose control, however temporarily, of that message. This article describes how to protect your message in transit.

A similar version of this article appeared October 6, 2014 on *Embedded*.

There is one way to absolutely, positively guarantee that someone will receive a message intact, unadulterated, authenticated, and observed by no unauthorized party. Just copy the message to a physical medium, lock it in a sturdy briefcase, handcuff the briefcase to your own wrist, and board a plane. Best of luck at the security gate. When you arrive at your destination, remove the briefcase from your wrist, unlock it, and present the message to your intended recipient. You can be assured that nobody else has seen it. Your recipient can be assured that the message is authentic. While you are there, find a comfortable meeting room and discuss the contents of the message, the weather, Italian restaurants—whatever you like. You have a little time before your flight home.

Use any other method for transmitting a message, and your message is at risk. Someone may intercept it and discover its contents. Or intercept your message and substitute it with one of their own. Or, intercept your message and block its transmission.

These three threats—disclosure of a secret message, alteration in transit, or blocking its transmission entirely—are the primary threats to any secure system. Protecting against these threats form what is known as the "CIA Triad."[1] The term has nothing to do with the U.S. Central Intelligence Agency. Instead, the letters stand for confidentiality, integrity, and availability. Any secure messaging system must protect confidentiality, provide integrity, and always be available when needed.

The problem is this: any time you employ a medium to transmit your message, you lose control, however temporarily, of that message. Write a letter, send an email, make a phone call. As soon as the message leaves you, you have relinquished control of it. We understand *in theory* that someone might intervene to take our message, but do we really care?

We now know that many of our electronic messages *are* at risk of routine, low-level snooping. And it is not just governments that do this. Businesses are archiving email messages, and some are routinely scanning inbound and outbound email to ensure that corporate secrets remain secret. Even if it is surprising, none of this is *necessarily* a nefarious thing.

But the time will come when each of us has news that we want to keep private. It is then and there that we care strongly about security. But in fact, the time to think about security is *before* you need it.

This article is the third part in my series on electronic security. In the first part we discussed the basic definition of security, when physical locks are less important than logical and virtual "fences." In the second installment we dissected the meaning and processes of tampering. In this article we do not look at the specific cryptographic algorithms involved in electronic security. We assume, until proven wrong, that properly implemented instances of (for example) AES for encryption and ECDSA for signatures are sufficiently robust to deter essentially all potential opponents. Instead, here we examine security in transit; we assess the "chain" that links the various parts of a secure message path.

Understanding the Networking Model

Before we delve into system security, we have to establish an understanding of the network itself.

In the early 1980s, standards organizations created what became known as the Open Systems Interconnection Reference Model (the "OSI Model" [2]). In this model, data networks were seen as a collection of seven layers, ranging from the physical layer of wires and radio signals at the bottom to the user application at the top. The concepts embodied in the model have proven extremely useful and help us understand what is happening at any point in the journey that a message takes from sender to recipient.

To simplify the discussion, we can condense the seven layers into just four, ranked from bottom to top:[3]

- **The physical and data link layer**. This layer is concerned with how a network-connected device talks and listens over the physical medium. For example, in a digital radio system this layer would control when the station can transmit; what transmit power, frequencies, and modulation schemes may be used; what packet structure must be used; and how to address other stations that can be directly contacted.
- **The network layer**. This layer is concerned with how one system talks to another system on the network. Systems on the network may not all be the same; some are connected by wireless links, some by wired Ethernet, and others by modems or other, older technology. These distinctions do not matter to the network layer. The network layer provides a consistent addressing scheme that gives every system running on it a unique network address. The most widely used network layer protocol today is Internet Protocol (IP).
- **The transport layer**. This layer is the first to "understand" the concept of a connection. The network layer just sends and receives packets. The data link layer takes the data and transmits it over the physical medium without any analysis. But the transport layer is the first to "make sense" of the set of packets, and arrange them as though they belong to a group. The most widely used transport layer protocol today is the Transmission Control Protocol, or TCP, and it provides the concept of a "connection." Thus, the software asks TCP to make a connection to a port on a remote machine. (The web, for example, operates on TCP port 80.) Its companion protocol, the User Datagram Protocol (UDP) is similar, but simply passes packets individually through to the higher layers without regard to an established connection. UDP is termed connectionless for that reason.

- **The application layer**. This protocol layer is the workhorse that accomplishes the task intended by the user. If the user requests a webpage, it is transmitted over the Hypertext Transport Protocol (HTTP). If the user sends an email, it may be carried over the Simple Mail Transfer Protocol (SMTP). Everything that one might wish to do on the network is mediated by an application layer protocol.

When, for example, a user requests a webpage, the request makes its way over the network until, finally, it lands on the router connected, probably by an Ethernet cable, to the web server. The Ethernet card receives the packet and unwraps the first layer (the data link layer), thereby terminating that layer. The Ethernet interface passes the data contained therein to the computer. Network software running on the computer unpacks the IP header and extracts the destination address ("Yes, it is for me!") and the source address ("Oh, so *that* computer is calling!") and determines the transport protocol ("It is TCP."). The software terminates the network layer and passes the payload to the transport layer.

The transport layer verifies that the packet is transmitted in the proper sequence and that it is bound for a port which some software task has claimed. If the requested port is port 80, the packet is destined for the web server task. The code running the transport layer removes the transport information, terminating the transport layer, and passes what is (most likely) HTTP data to the web server.

So now you know the network model…in an abbreviated way. The important thing here is that, from end to end, only the application layer remains intact. The message may originate on a cell phone with a wireless physical/link layer. It may pass through message handlers that terminate and reestablish the network and transport layers; and finally, it may terminate the message on a desktop computer with a wired connection. While each segment of the journey may (or may not) be encrypted, at the borders between segments, unless you take specific action, *your message can be read by anyone with access to those segment breaks*.

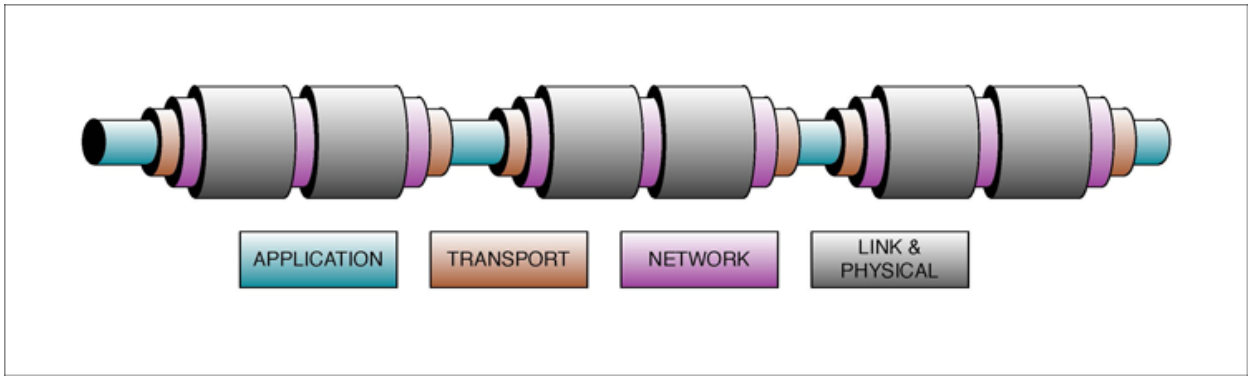

Figure 1. The application information may pass through many different transport, network, link, and physical layers before reaching its destination.

"But I Thought the Network Was Safe!"

Figure 1 shows how application data (i.e., the blue pipe) proceeds from the originator of the message to the receiver, even through different physical media, different types of networks, and different transport mechanisms carrying the message. Each concentric cylinder represents a network protocol. Where each cylinder begins and ends represents the points at which one protocol layer is established and another terminated.

Now, assume that you are using a Wi-Fi® connection to send a message. You are not concerned about security, because the Wi-Fi connection is secure. You are using WPA2 encryption, so no one can snoop on your message. How far does that security actually extend?

The answer is, not very far (**Figure 2**). Because WPA2 is part of the link layer protocol, as soon as the packets are received at the wireless access point, the encryption envelope is terminated. Hopefully, the other network elements will, in turn, encrypt the network traffic. But when dealing with security, we cannot depend on hope.

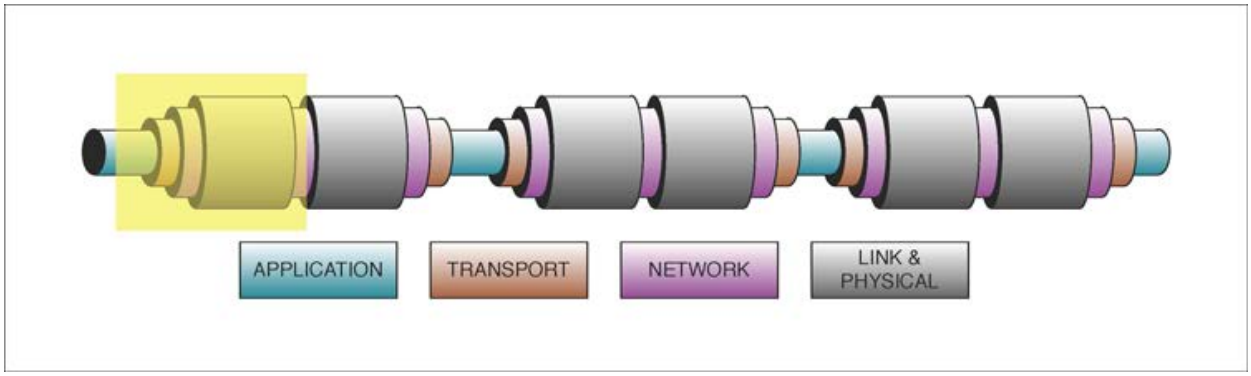

Figure 2. WPA2 protection over Wi-Fi extends only to the end of the first link layer.

Well, what about the Secure Sockets Layer, or Transport Layer Security (SSL/TLS)? Most of us have seen the little lock in the browser window when we are doing sensitive work, like online banking.

We have been trained to interpret this lock icon as “secure;” it means that the transaction is secure and the information encrypted so that nobody can access the information except the intended recipient. So, how far does that security extend? Transport layer security only survives as long as the original transport layer is in place (**Figure 3**). If the message is handed off to another transport layer, security has to be reestablished… or not.

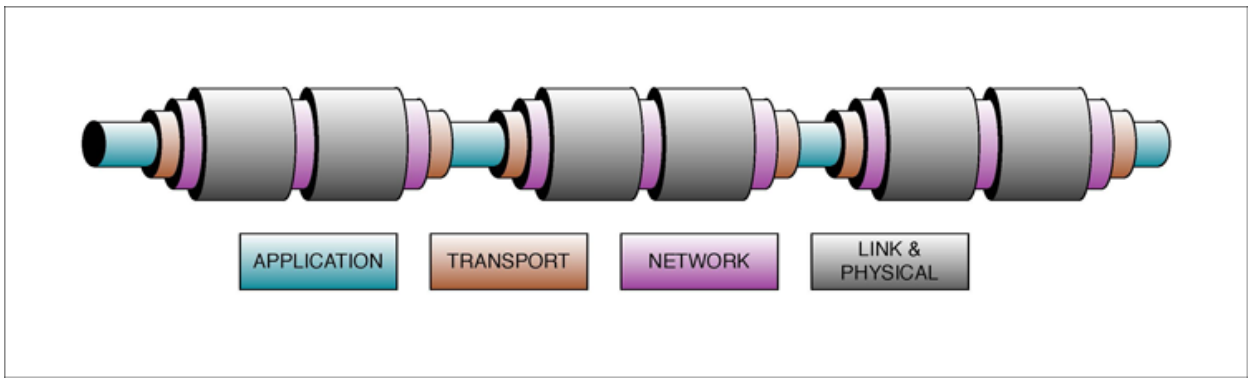

Figure 3. Transport layer security lasts until the transport layer is terminated—usually at the server. The rest of the message path may be unprotected.

In most cases, if you are dealing with a first-party website, the transport layer security is enough. This is because there is only one transport layer from your computer, tablet, or cell phone to the web server that manages the transaction; the information always stays in the security envelope from start to finish. But in the case of email, that security envelope goes from your computer only to the SMTP server receiving your outbound email message, *and no further*. From that point on, your email can be read by anyone who can access the servers along the way.

These examples are not intended to create fear, but they do serve as a warning: once a message leaves the safety of its security envelope, it is vulnerable to parties whose privacy and security concerns may not align with yours. Fortunately, there are other ways to ensure the security and privacy of a message.

Extending the Envelope

For better or worse, we still think in old patterns. For example, the Save icon in most programs still bears an image of a floppy disk, even though most of us have not used one of those for 20 years. Old ways of thinking really are hard to break.

In times past when we received a letter, we knew that the sender had sealed the envelope and that we were the first to see the message. (Oh, yes, the envelope might have been steamed open and resealed, but that was a lot of trouble. And besides, who would want to do that to me or you on a bulk basis?) But as we have seen, modern digital envelopes do not persist very far in the message chain. To intercept a digital message, no steam is required. The opponent need only capture the message after the encryption envelope has been removed at some stage during the lifespan of the message.

The solution to this vulnerability is conceptually simple, although challenging in practice: extend the envelope. If you apply the encryption at the *application layer* then it survives the setup and teardown of all lower layers. In brief, it looks like the illustration in **Figure 4**.

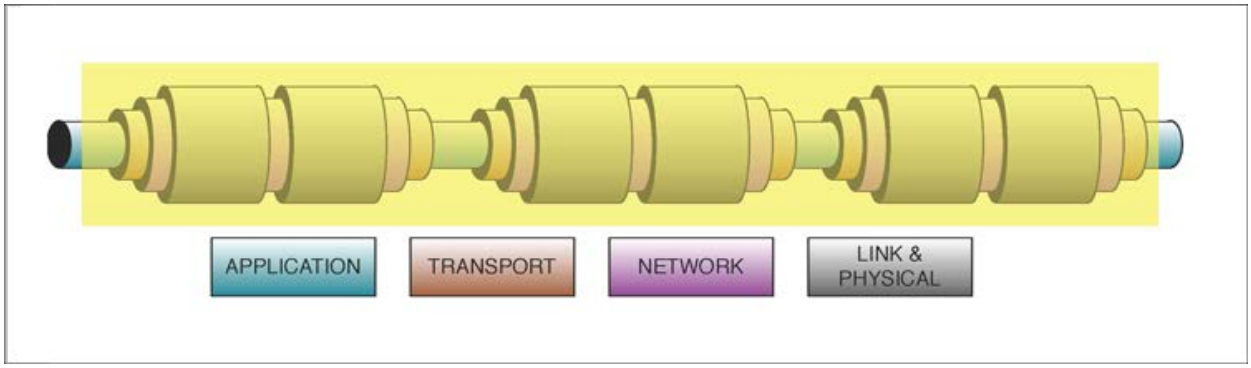

Figure 4. Application layer security extends from the message source all the way to the destination.

If you present the transport layer with already encrypted data, it does not matter how it treats the data, as long as it faithfully transfers the data to the endpoint. It can apply—and probably will apply—a TLS envelope to the already encrypted data, and that is fine. That encryption layer will terminate when the transport layer terminates. Wi-Fi encryption will survive only until the Wi-Fi signal is terminated at the access point—but the application layer will still be encrypted.

But How?

If you are a designer creating a new messaging system, the lesson is clear. Unless there is just one transport pipe between the originator and receiver of a message, you must not rely just on the SSL/TLS libraries and subsystems to secure your users' messages. Consider, for example, a store-and-forward message system. It establishes a transport pipe when a user sends a message and a second transport pipe when the intended recipient retrieves a message. This message transfer is not completely secure even if SSL/TLS is used on both the sending and receiving links.

But suppose that you are not a designer? What if you want to provide better security for messages that you or your company sends?

Begin with one fundamental truth: if you are dealing with a first-party website, they define the security parameters, and you most certainly do not. The obvious first-party web operations over which we might predictably have security concerns are social media, but they are not the only ones. Banks, credit card companies, and any firm which has day-to-day contact with users also get to set the rules about security. The good news is that since the entire transaction takes place over a single transport-layer pipe, the transport layer security is generally good enough for these parties. The bad news is that the data you access comes from a back-end database, and there is absolutely no transparency about the security of those back-end links.

But there are areas where we can take control of our own security. For the remainder of this article, we will look at one in particular: email security.

End-to-End Security of Email

To perform end-to-end encryption of email messages, you need an encryption certificate from a certificate authority. Some certificate authorities provide a certificate valid for one year free for personal use; longer-term certificates or certificates intended for business use are available on a paid basis. There are two types of certificates: class 1 certificates only guarantee that the email address used to obtain the certificate is the originator of the email message; class 2 certificates actually guarantee the identity of the sender and require more extensive background checks by the certificate authority.

Once you obtain a certificate, you must install it in your email client. This is a nontrivial operation best handled by an IT professional. It can, however, be done by an individual willing to read and follow directions and perform the requisite follow-up testing.

Once installed, the certificate directs the email client to transmit a signature attachment with each email. Properly configured email clients receiving a message with a signature attachment can verify the message sender by following the certificate's "chain of trust." The signature asserts, "This message was signed by (for example) Alice, and here is her certificate." The certificate asserts, "This is Alice's certificate, and this certificate is signed by (once again, for example) Bob's certificate authority, who vouches for it." Bob's certificate authority also has a certificate signed by a higher level certificate authority, and so forth, until a root certificate known to the receiving email client is reached.

At the end, the reasoning embodied in the chain of trust works like this: "I know Doug, and Doug says he knows Cindy, and Cindy knows Bob, and Bob knows Alice, so I can trust that this message came from Alice." In this way, Alice can prove that she, and only she, could have sent the message since she is the only one who possesses the signing key for which the chain of trust vouches. The signature also proves

that the message has not been corrupted in transit. In addition, the signature block will include the sender's public encryption key, so that the return message can be automatically encrypted.

After you—and those with whom you correspond—have installed certificates, the mail client will automatically encrypt and sign messages between you and your associates. This encryption has both benefits and disadvantages. The benefits are obvious: your message is secure and probably comes from you; your message cannot be scanned for keywords as it passes through commercial email providers; consequently, your message cannot be used for future advertising or demographic analysis.

But there is a disadvantage to this encryption: the message cannot be scanned for dangerous attachments or viruses. Corporate firewalls are useless against harmful content that may be present in encrypted messages. You, as the sender or receiver of the message, must be particularly vigilant about opening attachments within encrypted messages.

Conclusion

I am not the first to observe that transmitting sensitive information over public data networks is like writing your most personal secrets on postcards. Probably nobody will take the time to read a postcard as it moves through the postal system, but what if they do? And in a world where everyone writes their most personal secrets on postcards, what can we say about someone who wants an envelope? Is that person a spy? A criminal? Maybe worse?

Right now, people who rely on SSL/TLS and other electronic transport security mechanisms for end-to-end email security are just like someone who writes secrets on postcards and then insists that the mail trucks be armored. The journey from your home to the local post office is absolutely secure, but the real security risk is when the postcards are unloaded off the armored truck. And what about the odd, occasional message that actually uses an envelope? We'll set that aside for "special processing…."

Today, we rarely use postcards. Even the most mundane messages are placed in envelopes. In the real world, end-to-end security of postal mail is the default; in the digital world, end-to-end security of email messages is the exception, difficult to set up and perhaps even makes the user of such technology a suspect.

We designers should consider end-to-end security as a base function and not as an ancillary feature of our systems, all systems and not just email. And for everyone else, perhaps it is time to invest in a box of envelopes.

References

1. For more background on the CIA Triad, see Perrin, Chad, "The CIA Triad," TechRepublic, June 30, 2008, www.techrepublic.com/blog/it-security/the-cia-triad/. Also background on CIA Triad and general security under "Information Security" at Wikipedia, http://en.wikipedia.org/wiki/Information_security.
2. General information on the OSI Model is found at http://en.wikipedia.org/wiki/OSI_model.
3. We are not discussing everything about each layer. For a more detailed presentation on the Microsoft® website, see "The OSI Model's Seven Layers Defined and Functions Explained," at http://support.microsoft.com/kb/103884.

Microsoft is a registered trademark and registered service mark of Microsoft Corporation.
Wi-Fi is a registered certification mark of Wi-Fi Alliance Corporation.

Related Parts		
MAX71637	Energy Measurement SoCs	Free Samples
MAXQ1103	DeepCover Secure Microcontroller with Rapid Zeroization Technology and Cryptography	
MAXQ1850	DeepCover Secure Microcontroller with Rapid Zeroization Technology and Cryptography	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/en/support
For Samples: http://www.maximintegrated.com/en/samples
Other Questions and Comments: http://www.maximintegrated.com/en/contact

Application Note 5973: http://www.maximintegrated.com/en/an5973
APPLICATION NOTE 5973, AN5973, AN 5973, APP5973, Appnote5973, Appnote 5973