

By George Ou

As much as I respect my colleague and mentor David Berlind, who taught me the news business, I have to say that he's got it all wrong when it comes to [Internet mail security](#). Berlind is reeling over his incorrect perception that the Internet still lacks secure e-mail. But the solution has been under his nose all this time, and it really isn't the non-interoperability nightmare he paints it to be. Secure connections between server-to-client, end-to-end, and server-to-server communications have all been around for a long time. The technology is already baked into existing software, and it's simply a matter of installing, configuring, or merely enabling the technology.

Server-to-client encryption is a checkmark away

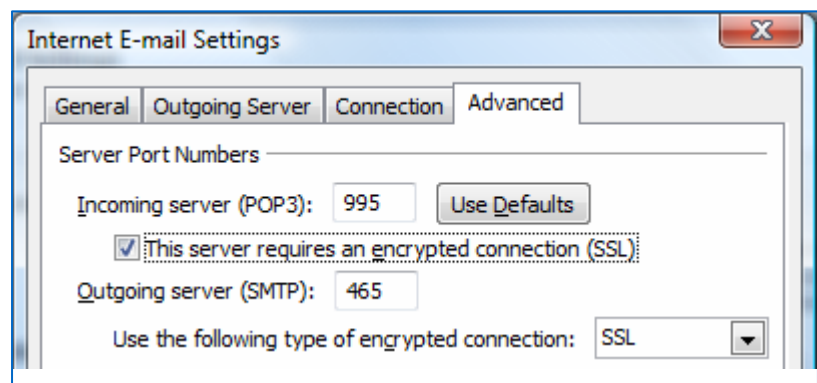
The most likely way to get eavesdropped on is in the last 100 feet, whether that's through a wire (through Layer 2 hijacking) or a wireless LAN connection. To enable server-to-client encryption, you simply check a simple option to enable SSL and type a different port number for your POP3 (inbound) and SMTP (outbound) Mail Server settings in your e-mail client.

My current DSL provider, AT&T, like most ISPs, supports SSL encryption on POP3 and SMTP. It's as simple as a checkmark and using ports 995 for POP3 and 465 for SMTP instead of the usual ports 110 and 25. The problem is that AT&T doesn't disable unencrypted mode, which means the vast majority of users won't use the secure transport mode.

If you happen to be a Gmail kind of guy or gal, simply type in

<https://mail.google.com> (not http://), and

your entire authentication and Web mail session is encrypted with export-restricted grade SSL encryption. The problem with Google (similar to AT&T's not disabling regular POP3 and SMTP) is that it doesn't disable HTTP mode. That means 99 percent of the population will continue using unencrypted http mode. Hotmail doesn't support payload encryption, so Hotmail users are out of luck. **(Are you listening Microsoft?)**



End-to-end e-mail cryptography

End-to-end cryptography, which encompasses authentication, non-repudiation, and encryption, has been baked into every e-mail client in the form of [S/MIME](#) for a decade, with pushbutton simplicity and full interoperability. In fact, an e-mail client without S/MIME support is like a Web browser that doesn't support HTTPS SSL mode. All you need to do is obtain a [FREE Personal Digital Certificate](#) from a certificate authority like Thawte through a Web enrollment process. In that enrollment process, you get to generate your own public and private key pair, and Thawte will digitally sign your public key after an e-mail round trip, where you demonstrate control and possession of your e-mail account.

Once you've obtained that certificate, you can digitally sign any e-mail, and everyone in the whole world using an e-mail client less than a decade old will be able to read and trust that digital signature to have truly come from the purported "from" e-mail address. The other side doesn't even need its own digital certificates to read your signatures. This is the "authentication" component.

Note that I said recipients can trust that it came from the purported e-mail address—I didn't say the digital signature proves the message came from you. If you want to prove it came from you, you're going to have to go to the [WOT \(Web Of Trust\)](#) and get someone acting as a Thawte notary to bind your identity to your e-mail address and digital certificate. With your identity bound to your e-mail and digital certificate, you have non-repudiation, and your signed e-mails can be treated as contracts under the eyes of the law. Non-repudiation means that when you send out a message with your digital signature bound to your e-mail and identity, you can't claim that anyone else falsified that message because only the owner of your private key could have generated that digital signature.

E-mail security has been around forever—you just need to turn it on

To enable encryption, both sides must have their own digital certificates, which also means both sides can digitally sign. Once the digital certificates are installed on each end, you simply need to click the Encrypt button built into your e-mail client. The beauty of this encryption scheme is that it doesn't care if the network and server infrastructure in the middle is trusted or not, because only the end points can decrypt the messages. This, however, does not negate the need for server-to-client encryption. We don't want someone else to be able to take over the e-mail account on either end, even if they can't read the encrypted messages.

S/MIME is so universal that even companies like [PGP Corporation](#), which offers solutions that manage an enterprise's cryptography infrastructure, will support it, in addition to PGP and GPG. PGP and GPG are alternatives to S/MIME, but the technology isn't baked into every e-mail client and must be installed as an add-on.

Server-to-server cryptography

Sniffing traffic between two SMTP servers is REALLY difficult to pull off unless you have access to the ISP's infrastructure, you've hacked into a server on one end, or you've hacked a router or firewall between the company and the ISP. In fact, if you could hijack someone's e-mail server, you could steal their domain name or obtain SSL certificates on behalf of the actual owner. But we can mitigate these types of attacks by enabling server-to-server encryption even if a hacker manages to get between your server and the Internet. Enabling server-to-server encryption is quite simple and requires nothing more than enabling SSL or TLS on the SMTP server and obtaining a \$20 publicly trusted digital certificate on each server. (See this tutorial on [obtaining digital certificates](#)). Unfortunately, not all companies and organizations have enabled server-to-server SSL or TLS communications. But the technology is already there, and the price of obtaining digital certificates has dropped to almost nothing.

The bottom line is that there is a lesson here for any individual or mail administrator who wants secure e-mail. The technology is already baked in and it often costs little or nothing to set it up. You just have to be willing to turn it on.