ArticSoft

The problems with secure e-mail

Summary
The ideal system that everyone is searching for – the silver bullet, is to have top security automatically regardless of who you are sending to and what product(s) they happen to be using. The reality is that many e-mail packages are not themselves secure, and do not interoperate cleanly with anything but their own products.

For the time being you are better off keeping your security outside of your e-mail or word processing package, and exchanging attachments that are fully protected and not relying upon any of the different systems that people are using. That way you increase the security of the result and do not have to rely on complex interactions between proprietary systems.

It may not be as elegant, but it will take you a lot further than relying on a specific e-mail service and will give you, for the time being, a much more secure result.

Introduction

For the last ten years or so we have become increasingly reliant on e-mail. It is ubiquitous, and unlike real mail it can chase us from continent to continent in seconds. For better or worse we now have the ability to conduct the next worst thing to conversation, but in writing.

Of course, and despite all the advice, we treat this ability as if it were the same as personal conversation. Private. Off the record. We also assume that no-one else is going to be able to read it, and that it can't ever get into the wrong hands.

Slowly but surely we are finding out, the hard way, that, as in the words of the song, "It ain't necessarily so." What we are doing is like sending picture postcards through the mail. It appears that everyone from our e-mail administrator to half the hacking community can pick up what we are doing, even off the internal network.

Enter the answer – secure e-mail (Se-mail?). Run it just like ordinary mail but click on the secure button and you're done. Shangri-La! But is it for real or is it yet another of the IT pipe dreams?

Silver Bullet Syndrome

This is not a new disease. Far from it. This is a regular epidemic every time someone goes near the IT security allergy. Somehow or other it seems obvious to anyone that the immense complexity of the computer can be made safe and secure by a single act (the laying on of hands perhaps?). Despite the fact that every day experience teaches us how difficult it is to get a computer to anything without us making a significant contribution, security is supposed to happen without any thought or planning (even less than putting something in a brown envelope rather than a see-through folder).

Manufacturers have been quick to recognize two things. The first has been that they need to service their customers more so that they can charge more. The second is that despite all the claims about standards in security, the cold hard reality is that there are hardly any.

What, no standards?
Well, almost none. We have S/MIME (version 2 or 3?) to sort out how you might sign and encrypt streams going from one e-mail client to another. That's fine except that you need 'PKI' standards sitting behind S/MIME to make it useful, and there seem to be more of those than you can shake a stick at. This is a case where there are so many different standards (and even more interpretations of them) that in effect you have no standards.

If you want to think about standards in terms of manufacturer's products (after all, dominant suppliers and monopolies set standards of a kind) then the picture is more like this. We have Outlook Express and Outlook (not the same thing even if they are from the same stable) and HotMail. To that we must add Eudora, Lotus Notes and AOL (Compuserve). We have an increasing number of web-mail products such as Yahoo and Lycos, just in case the others weren't enough. And we haven't yet begun to mention all the various brands of 'secure' mail that exist, including PGP. Can you believe that all of these interoperate smoothly and seamlessly with each other?

So we can conclude that standards are not yet in a position to help us.

Our objectives
Somewhere in the security debate, you lose, as we seem in danger of doing, sight of what your objective actually is because the technology debate is so much more confusing.

The objective for the user might be summarized as follows (borrowing from the paper world):

- to be certain what they send goes to the right person/place;
- to be certain that the right person/place can read the information;
- to be able to use signed information as proof to a court or other body;
- to stop the wrong people from reading personal and private information.

Some of these wishes are more difficult than others. Just as in the paper world, you can't stop anyone seeing the address on the outside of a letter, the same is true of e-mail. If someone alters that address, it doesn't go to the right place, and if someone alters the return address (in many countries it is written on the back of the envelope) the recipient may not know where it has come from or it may not, if delivery fails, be returned to the correct sender.

We are familiar with the paper world and it has some benefits. You can usually see if someone has already opened your mail. The Post Office can often cope with wrong addressing and still get it to the right place. You believe that the delivery service is going to behave in the way that you expect and you know that a proof of delivery from them is accepted by the authorities.

E-mail is rather different. There is no way of telling who reads the mail unless you take actual steps to make it impossible. The e-mail Post Office can't cope with any address errors whatsoever. It has no idea if any of the addresses on the mail are correct and can't tell if they have been altered. There is no plain envelope to stop people reading the contents and it is possible for hackers, government agencies and almost anyone else to read the mail. Proof of delivery is worth the paper it is printed on.

An impossible dream?
No. E-mail can be made secure, but you have to take a few things into account.

The first thing to understand is that you can't do much about the addresses, or the subject line. Nothing about these can be made secure. Don't ever believe them when you read them.

Different systems may allow you to secure the message text of the e-mail, but you have to be very certain what that security is, when it is added, when it is removed, and how you would prove it had been secured afterwards. These are fundamental to you if you are going to rely on the security mechanisms later as proof that something happened.

The second thing to understand is that you can never (with current systems) send anything secret to someone you don't know. It's not possible. You have to have a 'public key' of theirs before it can be done. You can't, with conventional systems, send information to 'anyone' in a particular group, function or business. You have to send to specific individuals.

The third thing to understand is that the protection that you apply to an e-mail has to be something that the recipient can deal with. E-mail systems don't currently relate the keys used for information protection to the recipients of the e-mail, and don't know what algorithms the recipient is likely to have. This is because there are far too many unnecessary choices forced onto users of these systems and services (or set by administrators who are making choices based upon their own prejudices rather than looking at usability). If you use something the recipient can't process you are wasting your time. But you can't afford the time needed to sort this kind of problem out.

Problem solving strategies
Most of the difficulties identified can be avoided by ignoring the e-mail systems completely and concentrating instead on the information to be sent. This could be anything – a Word document, a text file, some HTML, a graphic or even a video. Whatever you do should not alter its content, and it should not be possible to remove your security before the information is securely in the computer of the recipient.

This means that your protection software is going to have to protect the file in such a way that an attacker cannot remove the protection without you being able to detect it. (That's not the same as pretending a fake document is real. Since much of the information you get is not protected, today you make value judgments on what is 'right' based upon your own feelings, or you 'phone the sender and ask them to confirm what they actually sent. So removing the protection and making subtle changes to documents that you might then believe is perfectly feasible.)

The recipient is then in a position where their first step is to check the authenticity of the file they have received. That avoids any possibility of misunderstanding what is protected and what is not. The file is the thing that is protected, and not other parts of the e-mail that may, or may not be correct.

Once the recipient has checked that the file is authentic they can go ahead and use a copy of it that has had the protection removed. This is an essential step, because they must not be able to alter, or add to, the file that they received and still have it claim that it was ever authentic (unless, of course, you have some system that maintains a copy of each different thing in the file, protected by each person that has altered or added to it).

This approach may not seem as 'elegant' as having everything automated, but it is a lot more secure, and prevents any mistakes or misunderstandings about who has signed what, and therefore what can be relied upon.