Easy Email Encryption

Most security people would like to see more use of end-to-end encrypted email. But achieving this would require overcoming many obstacles, most notably usability and the absence of a suitable public-key infrastructure (PKI). With a bit of work, I think these can be overcome. In fact, the five biggest Internet companies—Apple, Google, Amazon, Facebook, and Microsoft—already have the skills, the technology, the information, and the scale to do what I dub “auto-cert.”

The PKI is the easiest part. A PKI is, in essence, a set of certified bindings between an identity and a cryptographic public key. “Identity” doesn’t have to be a name: it can be an email address or anything else meaningful to an email user. Any large company that runs an email service could easily create a PKI, and the big five could easily get their certificate authority (CA) operation trusted by major OSs. Facebook and Google have a pretty good handle on people’s real names as well. All of these companies either require or strongly encourage users to log in, often with a working email address.

The hard part, though, is usability from top to bottom: getting certificates, configuring mailers to use them, sending and receiving email, sharing everything across multiple devices, searching, and more. Let’s start with the simplest issue: obtaining a usable certificate. Web browsers have long been able to generate key pairs; the `<KEYGEN>` tag uploads the public key for signing and the web server can then construct a certificate. The problem is getting the private key and the certificate from the browser to the mailer. Apple and Microsoft control not just browsers but also mailers and the underlying OSs. If they want, their auto-cert mailers could request the certificates themselves.

Webmail systems are more challenging because there’s no local, trusted code base; you don’t know what downloaded JavaScript will do next time. This problem is easily solved with a browser extension, and enough of those have been written that we can regard this as solved. This also solves the problem of moving the key pair to the mailer: the browser is the mailer.

Today, of course, most people use multiple devices. Moving the private key to all devices has long been one of the trickier problems to solve. But these five companies know how to run large-scale, secure cloud-storage systems. That’s how to do it: store encrypted private keys online and download them to every device on which you want to read mail. Apple actually has a cloud service expressly designed for secure key storage; any of the other companies could easily create their own.

Search seems challenging, but there are two simple solutions. If the end systems are secure enough, they could store downloaded email in the clear. If not, encrypted search technology has become quite efficient.

As for key distribution, mailers—since Apple, Microsoft, and to some extent Google control software suites—could append users’ certificates to all emails they send and perhaps sign the emails. Gradually, their correspondents will learn these certificates. There remains the problem of people with some devices that are auto-cert-capable and some that aren’t, but that’s a problem that will resolve itself as older devices age out. In the meantime, auto-cert-capable devices will continue learning: if they see a mix of email coming from someone, they’ll know to be cautious about encrypting; if they see only mail with certificates, they can assume that all of that person’s devices are auto-cert capable. For that matter, all of these companies could easily run key servers, and a federated key server (one that could query all of them) would make life even easier.

The problem of standards remains. We need a standardized way of talking to key servers, passing around private keys, and deciding when and when not to send encrypted email. But the industry already knows how to do standards.

We can thus have secure email with little effort. These five companies could do it. It might be easiest for Apple because they already have secure key storage, but the others could easily replicate it. Will the industry do it?

Steven M. Bellovin is a professor of computer science at Columbia University. Contact him via www.cs.columbia.edu/~smb.