Suppose your doctor said, “I realize we have antibiotics that are good at treating your kind of infection without harmful side effects, and that there are decades of research to support this treatment. But I’m going to give you tortilla-chip powder instead, because, uh, it might work.” You’d get a new doctor.

Practicing medicine is difficult. The profession doesn’t rush to embrace new drugs; it takes years of testing before benefits can be proven, dosages established, and side effects cataloged. A good doctor won’t treat a bacterial infection with a medicine he just invented when proven antibiotics are available. And a smart patient wants the same drug that cured the last person, not something different.

Cryptography is difficult, too. It combines mathematics, computer science, sometimes electrical engineering, and a twisted mindset that can figure out how to get around rules, break systems, and subvert the designers’ intentions. Even very smart, knowledgeable, experienced people invent bad cryptography. In the crypto community, people aren’t even all that embarrassed when their algorithms and protocols are broken. That’s how hard it is.

Reusing Secure Components

Building cryptography into products is hard, too. Most cryptography products on the market are insecure. Some don’t work as advertised. Some are obviously flawed. Others are more subtly flawed. Sometimes people discover the flaws quickly, while other times it takes years (usually because no one bothered to look for them). Sometimes a decade goes by before someone invents new mathematics to break something.

This difficulty is made even more serious for several reasons. First, flaws can appear anywhere. They can be in the trust model, the system design, the algorithms and protocols, the implementations, the source code, the human-computer interface, the procedures, the underlying computer system. Anywhere.

Second, these flaws cannot be found through normal beta testing. Security has nothing to do with functionality. A cryptography product can function normally and be completely insecure. Flaws remain undiscovered until someone looks for them explicitly.

Third, and most importantly, a single flaw breaks the security of the entire system. If you think of cryptography as a chain, the system is only as secure as its weakest link. This means that everything has to be secure. It’s not enough to make the algorithms and protocols perfect if the implementation has problems. A great product with a broken algorithm is useless. And a great algorithm, protocol, and implementation can be ruined by a flawed random-number generator. If there is a security flaw in the code, the rest of it doesn’t matter.

Given this harsh reality, the most rational design decision is to use as few links as possible, and as high a percentage of strong links as possible. Since it is impractical for a system designer (or even a design team) to analyze a completely new system, a smart designer reuses components that are generally believed to be secure, and only invents new cryptography where absolutely necessary.

TRUSTING THE KNOWN

Consider IPSec, the Internet IP security protocol (described in the sidebar “About IPSec”). Beginning in 1992, it was designed in the open by committee and was the subject of considerable public scrutiny from the start. Everyone knew it was an important protocol, and people spent a lot of effort trying to get it right. Security technologies were proposed, broken, and then modified. Versions were codified and analyzed. The first draft of the standard was published in 1995. Aspects were debated on security merits and on performance, ease of implementation, upgradability, and use.

In November 1998, the committee published a pile of RFCs—one in a series of steps to make IPSec an Internet standard. And it is still being studied.
Cryptographers at the Naval Research Laboratory recently discovered a minor implementation flaw. The work continues, in public, by anyone and everyone who is interested.

On the other hand, Microsoft developed its own Point-to-Point Tunneling Protocol (PPTP) to do much the same thing. They invented their own authentication protocol, their own hash functions, and their own key-generation algorithm. Every one of these items was badly flawed. They used a known encryption algorithm, but they used it in such a way as to negate its security. They made implementation mistakes that weakened the system even further. But since they did all this work internally, no one knew that their PPTP was weak.

Microsoft fielded PPTP in Windows NT and 95, and used it in their virtual private network (VPN) products. It wasn’t until summer of 1998 that Counterpane Systems published a paper describing the flaws we found. Microsoft quickly posted a series of fixes, which we have since evaluated and found wanting. They don’t fix things nearly as well as Microsoft would like people to believe. And then there is a company like TriStrata, which claimed to have a proprietary security solution without telling anyone how it works (because it’s patent pending). You have to trust them. They claimed to have a new algorithm and new set of protocols that are much better than any that exist today. And even if they make their system public, the fact that they’ve patented it and retain proprietary control means that many cryptographers won’t bother analyzing their claims.

LEVERAGING THE COLLECTIVE STRENGTH

You can choose any of these three systems to secure your virtual private network. Although it’s possible for any of them to be flawed, you want to minimize your risk. If you go with IPSec, you have a much greater assurance that the algorithms and protocols are strong. Of course, the product could still be flawed—there could be an implementation bug or a bug in any of the odd little corners of the code not covered in the IPSec standards—but at least you know that the algorithms and protocols have withstood a level of analysis and review that the Microsoft and TriStrata options have not.

Choosing the TriStrata system is like going to a doctor who has no medical degree and whose novel treatments (which he refuses to explain) have no support from the AMA. Sure, it’s possible (although highly unlikely) that he’s discovered a totally new branch of medicine, but do you want to be the guinea pig?

The point here is that the best security methods leverage the collective analytical ability of the cryptographic community. No single company (outside the military) has the financial resources necessary to evaluate a new cryptographic algorithm or shake the design flaws out of a complex protocol. The same holds true in cryptographic libraries. If you write your own, you will probably make mistakes. If you use one that’s public and has been around for a while, some of the mistakes will have been found and corrected.

It’s hard enough making strong cryptography work in a new system; it’s just plain lunacy to use new cryptography when viable, long-studied alternatives exist. Yet most security companies, and even otherwise smart and sensible people, exhibit acute neophilia and are easily blinded by shiny new pieces of cryptography.

FOLLOWING THE CROWD

At Counterpane Systems, we analyze dozens of products a year. We review all sorts of cryptography, from new algorithms to new implementations. We break the vast majority of proprietary systems, and, with no exception, the best products are the ones that use existing cryptography as much as possible. Not only are the conservative choices generally smarter, but they mean we can actually analyze the system. We can review a simple cryptography product in a couple of days if it reuses existing algorithms and protocols, in a week or two if it uses newish protocols and existing algorithms. If it uses new algorithms, a week is barely enough time to get started.

This doesn’t mean that everything new is lousy. What it does mean is that everything new is suspect. New cryptography belongs in academic papers, and then in demonstration systems. If it is truly better, then eventually cryptographers will

About IPSec

IPSec is a set of protocols being developed by the IETF to support secure packet exchange at the IP layer. Once it’s completed, IPSec is expected to be deployed widely to implement Virtual Private Networks. The current IPSec standards include three algorithm-independent base specifications that are currently standards-track RFCs. These three RFCs are in the process of being revised (according to the usual IETF procedures), and the revisions will take into account a number of security issues with the current specifications.

The IP Security Architecture (ftp://ds.internic.net/rfc/rfc1825.txt) defines the overall architecture and specifies elements common to both the IP Authentication Header and the IP Encapsulating Security Payload.

The IP Authentication Header (ftp://ds.internic.net/rfc/rfc1826.txt) defines an algorithm-independent mechanism for providing exportable cryptographic authentication without encryption to IPv4 and IPv6 packets.

The IP Encapsulating Security Payload (ftp://ds.internic.net/rfc/rfc1827.txt) defines an algorithm-independent mechanism for providing encryption to IPv4 and IPv6 packets.

For more information on IPSec, see the IETF’s IPSec charter (http://www.ietf.cnri.reston.va.us/html.charters/ipsec-charter.html).
Cambio+ products and services is defined. This results in a continuous supply and feedback loop, as shown in Figure 2.

As indicated in the figure, Cambio+ and the customer supply resources (including personnel) and deploy them according to a set of ISO/IEC 12207-defined cooperative processes. This creates a product version, which is then verified and validated according to a set of product quality attributes following ISO/IEC 9126. After product deployment, and based on experience with the product, the customer and Cambio+ jointly evaluate the product and establish new goals for improving the existing version as well as adding new functionality. This carries them along in the development cycle, revisiting the resource aspect, the processes, and the product quality.

The cooperative processes are based on a first-level tailoring of relevant ISO/IEC 12207 processes down to the activity level, which concretely identifies the project roles and responsibilities of Cambio+ and the customer. This tailoring serves as a firm basis for negotiation between Cambio+ and their customers. Further tailoring of the activities and task definition then provides a firm basis for project scheduling. Cambio+ can then employ the six major product quality attributes of ISO/IEC 9126 in verifying and validating the product: functionality, reliability, usability, efficiency, maintainability, and portability.

Cambio+ has, by using the ISO/IEC 12207 and 9126 standards, developed the backbone of a viable business concept that is gaining wide acceptance among its customers and potential customers.

T

hese examples demonstrate that learning to use ISO/IEC 12207 constructively provides important value-added returns on investment. Already translated into French, Japanese, and Portuguese, the standard will soon be available in Swedish. A US adaptation, the standard has now been officially approved as IEEE/EIA 12207. In addition to containing the ISO/IEC standard, further guidance is provided in respect to documentation and utilization. The 12207 standard will continue to evolve, and the standard working group is scheduled to amend it in 2000 and then revise a few years after. In addition, the same working group is developing ISO/IEC 15288 (System Life Cycle Processes), which provides similar guidance for the life cycles of complex systems involving hardware, software, and humans. Given the fact that quality management systems based on processes occupy the current line of thinking for ISO 9001:2000, both ISO/IEC 12207 and eventually 15288 will play an increasingly important role in defining processes to be evaluated and certified.

Harold W. (Bud) Lawson, an IEEE and ACM Fellow, is an independent consultant in Stockholm and head of the Swedish delegation to ISO/IEC JTC1 SC7WG7. He thanks Raghu Singh, editor of ISO/IEC 12207, and the Swedish National Technical Development Board and the Swedish National Telephone Administration for their support. Contact him at bud@lawson.se.

And beware the doctor who says, “I invented and patented this totally new treatment that consists of tortilla-chip powder. It has never been tried before, but I just know it is much better and I’m going to give it to you.” There’s a good reason we call new crytpography “snake oil.”

Bruce Schneier is president of Counterpane Systems, a cryptography and computer security consulting company. He is the author of Applied Cryptography (John Wiley & Sons, 1995) and the inventor of the Blowfish and Twofish encryption algorithms. He thanks Matt Blaze for the analogy that opened this column. You can subscribe to his free cryptography e-mail newsletter at http://www.counterpane.com.

Resources

Copies of the ISO/IEC 12207 standard are available from national standards bodies, such as ANSI (Customer Services, 11 West 42nd St., New York, NY 10036), from ISO (1, rue de Varembe, CH-1211 Geneva 20, Switzerland), or from IEA (3, rue de Varembe, CH-1211 Geneva 20, Switzerland).


The following book thoroughly presents software engineering standards, including 12207:


Internet Watch

Continued from page 109