Bruce Schneier

Why Cryptography Is Harder Than It Looks

by Bruce Schneier

CTO and Founder
Counterpane Internet Security, Inc.
e-mail: schneier@counterpane.com

You can download this essay in Postscript or PDF (Acrobat) format, or as a PalmPilot DOC.

From e-mail to cellular communications, from secure Web access to digital cash, cryptography is an essential part of today's information systems. Cryptography helps provide accountability, fairness, accuracy, and confidentiality. It can prevent fraud in electronic commerce and assure the validity of financial transactions. It can prove your identity or protect your anonymity. It can keep vandals from altering your Web page and prevent industrial competitors from reading your confidential documents. And in the future, as commerce and communications continue to move to computer networks, cryptography will become more and more vital.

But the cryptography now on the market doesn't provide the level of security it advertises. Most systems are not designed and implemented in concert with cryptographers, but by engineers who thought of cryptography as just another component. It's not. You can't make systems secure by tacking on cryptography as an afterthought. You have to know what you are doing every step of the way, from conception through installation.

Billions of dollars are spent on computer security, and most of it is wasted on insecure products. After all, weak cryptography looks the same on the shelf as strong cryptography. Two e-mail encryption products may have almost the same user interface, yet one is secure while the other permits eavesdropping. A comparison chart may suggest that two programs have similar features, although one has gaping security holes that the other doesn't. An experienced cryptographer can tell the difference. So can a thief.

Present-day computer security is a house of cards; it may stand for now, but it can't last. Many insecure products have not yet been broken because they are still in their infancy. But when these products are widely used, they will become tempting targets for criminals. The press will publicize the attacks, undermining public confidence in these systems. Ultimately, products will win or lose in the marketplace depending on the strength of their security.

Threats to computer systems

Every form of commerce ever invented has been subject to fraud, from rigged scales in a farmers' market to counterfeit currency to phony invoices. Electronic commerce schemes will also face fraud, through forgery, misrepresentation, denial of service, and cheating. In fact, computerization makes the risks even greater, by allowing attacks that are impossible against non-automated systems. A thief can make a living skimming a penny from every Visa cardholder. You can't walk the streets wearing a mask of someone else's face, but in the digital world it is easy to impersonate others. Only strong cryptography can protect against these attacks.

Privacy violations are another threat. Some attacks on privacy are targeted: a member of the press tries to read a public figure's e-mail, or a company tries to intercept a competitor's communications. Others are broad data-harvesting attacks, searching a sea of data for interesting information: a list of rich widows, AZT users, or people who view a particular Web page.

Criminal attacks are often opportunistic, and often all a system has to be is more secure than the next system. But there are other threats. Some attackers are motivated by publicity; they usually have significant resources via their research institution or corporation and large amounts of time, but few financial resources. Lawyers sometimes need a system attacked, in order to prove their client's innocence. Lawyers can collect details on the system through the discovery process, and then use considerable financial resources to hire experts and buy equipment. And they don't have to defeat the security of a system completely, just enough to convince a jury that the security is flawed.

Electronic vandalism is an increasingly serious problem. Computer vandals have already graffitied the CIA's web page, mail-bombed Internet providers, and canceled thousands of newsgroup messages. And of course, vandals

Security First Innovations, LLC, Exhibit 2006
Page 2006 - 1
IPR2024-00212, Google LLC v. Security First Innovations, LLC

and thieves routinely break into networked computer systems. When security safeguards aren't adequate, trespassers run little risk of getting caught.

Attackers don't follow rules; they cheat. They can attack a system using techniques the designers never thought of. Art thieves have burgled homes by cutting through the walls with a chain saw. Home security systems, no matter how expensive and sophisticated, won't stand a chance against this attack. Computer thieves come through the walls too. They steal technical data, bribe insiders, modify software, and collude. They take advantage of technologies newer than the system, and even invent new mathematics to attack the system with.

The odds favor the attacker. Bad guys have more to gain by examining a system than good guys. Defenders have to protect against every possible vulnerability, but an attacker only has to find one security flaw to compromise the whole system.

What cryptography can and can't do

No one can guarantee 100% security. But we can work toward 100% risk acceptance. Fraud exists in current commerce systems: cash can be counterfeited, checks altered, credit card numbers stolen. Yet these systems are still successful because the benefits and conveniences outweigh the losses. Privacy systems--wall safes, door locks, curtains--are not perfect, but they're often good enough. A good cryptographic system strikes a balance between what is possible and what is acceptable.

Strong cryptography can withstand targeted attacks up to a point--the point at which it becomes easier to get the information some other way. A computer encryption program, no matter how good, will not prevent an attacker from going through someone's garbage. But it can prevent data-harvesting attacks absolutely; no attacker can go through enough trash to find every AZT user in the country. And it can protect communications against non-invasive attacks: it's one thing to tap a phone line from the safety of the telephone central office, but quite another to break into someone's house to install a bug.

The good news about cryptography is that we already have the algorithms and protocols we need to secure our systems. The bad news is that that was the easy part; implementing the protocols successfully requires considerable expertise. The areas of security that interact with people--key management, human/computer interface security, access control--often defy analysis. And the disciplines of public-key infrastructure, software security, computer security, network security, and tamper-resistant hardware design are very poorly understood.

Companies often get the easy part wrong, and implement insecure algorithms and protocols. But even so, practical cryptography is rarely broken through the mathematics; other parts of systems are much easier to break. The best protocol ever invented can fall to an easy attack if no one pays attention to the more complex and subtle implementation issues. Netscape's security fell to a bug in the random-number generator. Flaws can be anywhere: the threat model, the system design, the software or hardware implementation, the system management. Security is a chain, and a single weak link can break the entire system. Fatal bugs may be far removed from the security portion of the software; a design decision that has nothing to do with security can nonetheless create a security flaw.

Once you find a security flaw, you can fix it. But finding the flaws in a product can be incredibly difficult. Security is different from any other design requirement, because functionality does not equal quality. If a word processor prints successfully, you know that the print function works. Security is different; just because a safe recognizes the correct combination does not mean that its contents are secure from a safecracker. No amount of general beta testing will reveal a security flaw, and there's no test possible that can prove the absence of flaws.

Threat models

A good design starts with a threat model: what the system is designed to protect, from whom, and for how long. The threat model must take the entire system into account--not just the data to be protected, but the people who will use the system and how they will use it. What motivates the attackers? Must attacks be prevented, or can they just be detected? If the worst happens and one of the fundamental security assumptions of a system is broken, what kind of disaster recovery is possible? The answers to these questions can't be standardized; they're different for every system. Too often, designers don't take the time to build accurate threat models or analyze the real risks.

Threat models allow both product designers and consumers to determine what security measures they need. Does it makes sense to encrypt your hard drive if you don't put your files in a safe? How can someone inside the company defraud the commerce system? Are the audit logs good enough to convince a court of law? You can't design a secure system unless you understand what it has to be secure against.

System design

Security First Innovations, LLC, Exhibit 2006
Page 2006 - 2
IPR2024-00212, Google LLC v. Security First Innovations, LLC

Design work is the mainstay of the science of cryptography, and it is very specialized. Cryptography blends several areas of mathematics: number theory, complexity theory, information theory, probability theory, abstract algebra, and formal analysis, among others. Few can do the science properly, and a little knowledge is a dangerous thing: inexperienced cryptographers almost always design flawed systems. Good cryptographers know that nothing substitutes for extensive peer review and years of analysis. Quality systems use published and well-understood algorithms and protocols; using unpublished or unproven elements in a design is risky at best.

Cryptographic system design is also an art. A designer must strike a balance between security and accessibility, anonymity and accountability, privacy and availability. Science alone cannot prove security; only experience, and the intuition born of experience, can help the cryptographer design secure systems and find flaws in existing designs.

Implementation

There is an enormous difference between a mathematical algorithm and its concrete implementation in hardware or software. Cryptographic system designs are fragile. Just because a protocol is logically secure doesn't mean it will stay secure when a designer starts defining message structures and passing bits around. Close isn't close enough; these systems must be implemented exactly, perfectly, or they will fail. A poorly designed user interface can make a hard-drive encryption program completely insecure. A false reliance on tamper-resistant hardware can render an electronic commerce system all but useless. Since these mistakes aren't apparent in testing, they end up in finished products. Many flaws in implementation cannot be studied in the scientific literature because they are not technically interesting. That's why they crop up in product after product. Under pressure from budgets and deadlines, implementers use bad random-number generators, don't check properly for error conditions, and leave secret information in swap files. The only way to learn how to prevent these flaws is to make and break systems, again and again.

Cryptography for people

In the end, many security systems are broken by the people who use them. Most fraud against commerce systems is perpetrated by insiders. Honest users cause problems because they usually don't care about security. They want simplicity, convenience, and compatibility with existing (insecure) systems. They choose bad passwords, write them down, give friends and relatives their private keys, leave computers logged in, and so on. It's hard to sell door locks to people who don't want to be bothered with keys. A well-designed system must take people into account.

Often the hardest part of cryptography is getting people to use it. It's hard to convince consumers that their financial privacy is important when they are willing to leave a detailed purchase record in exchange for one thousandth of a free trip to Hawaii. It's hard to build a system that provides strong authentication on top of systems that can be penetrated by knowing someone's mother's maiden name. Security is routinely bypassed by store clerks, senior executives, and anyone else who just needs to get the job done. Only when cryptography is designed with careful consideration of users' needs, and then smoothly integrated, can it protect their systems, resources, and data.

The state of security

Right now, users have no good way of comparing secure systems. Computer magazines compare security products by listing their features, not by evaluating their security. Marketing literature makes claims that are just not true; a competing product that is more secure and more expensive will only fare worse in the market. People rely on the government to look out for their safety and security in areas where they lack the knowledge to make evaluations--food packaging, aviation, medicine. But for cryptography, the U.S. government is doing just the opposite.

When an airplane crashes, there are inquiries, analyses, and reports. Information is widely disseminated, and everyone learns from the failure. You can read a complete record of airline accidents from the beginning of commercial aviation. When a bank's electronic commerce system is breached and defrauded, it's usually covered up. If it does make the newspapers, details are omitted. No one analyzes the attack; no one learns from the mistake. The bank tries to patch things in secret, hoping that the public won't lose confidence in a system that deserves no confidence. In the long run, secrecy paves the way for more serious breaches.

Laws are no substitute for engineering. The U.S. cellular phone industry has lobbied for protective laws, instead of spending the money to fix what should have been designed correctly the first time. It's no longer good enough to install security patches in response to attacks. Computer systems move too quickly; a security flaw can be described on the Internet and exploited by thousands. Today's systems must anticipate future attacks. Any comprehensive system--whether for authenticated communications, secure data storage, or electronic commerce--is likely to remain in use for five years or more. It must be able to withstand the future: smarter attackers, more

Security First Innovations, LLC, Exhibit 2006
Page 2006 - 3
IPR2024-00212, Google LLC v. Security First Innovations, LLC

computational power, and greater incentives to subvert a widespread system. There won't be time to upgrade them in the field.

History has taught us: never underestimate the amount of money, time, and effort someone will expend to thwart a security system. It's always better to assume the worst. Assume your adversaries are better than they are. Assume science and technology will soon be able to do things they cannot yet. Give yourself a margin for error. Give yourself more security than you need today. When the unexpected happens, you'll be glad you did.

Security First Innovations, LLC, Exhibit 2006
Page 2006 - 4
IPR2024-00212, Google LLC v. Security First Innovations, LLC