Kerckhoffs's principle

In cryptography, **Kerckhoffs's principle** (also called **Kerckhoffs's Desiderata**, **Kerckhoffs's assumption**, **axiom**, or **law**) was stated by Auguste Kerckhoffs in the 19th century: A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

Kerckhoffs's principle was reformulated (or perhaps independently formulated) by Claude Shannon as "The enemy knows the system," *i.e.*, "One ought design systems under the assumption that the enemy will immediately gain full familiarity with them." In that form, it is called **Shannon's maxim**. In contrast to "security through obscurity," it is widely embraced by cryptographers.

Origins

In 1883 Auguste Kerckhoffs[1] wrote two journal articles on *La Cryptographie Militaire*,[2] in which he stated six design principles for military ciphers. Translated from French, they are:[3]

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concourse of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Some are no longer relevant given the ability of computers to perform complex encryption, but his second axiom, now known as Kerckhoffs's principle, is still critically important.

Explanation of the principle

Stated simply, the security of a cryptosystem should depend solely on the secrecy of the key and the private randomizer.[4] Another way of putting it is that a method of secretly coding and transmitting information should be secure even if everyone knows how it works. Of course, despite the attacker's familiarity with the system in question, the attacker lacks knowledge as to which of all possible instances is being presently observed.

Advantage of secret keys

Using secure cryptography is supposed to replace the difficult problem of keeping messages secure with a much more manageable one, keeping relatively small keys secure. A system that requires long-term secrecy for something as large and complex as the whole design of a cryptographic system obviously cannot achieve that goal. It only replaces one hard problem with another. However, if a system is secure even when the enemy knows everything except the key, then all that is needed is to manage keeping the keys secret.

There are a large number of ways the internal details of a widely used system could be discovered. The most obvious is that someone could bribe, blackmail, or otherwise threaten staff or customers into explaining the system. In war, for example, one side will probably capture some equipment and people from the other side. Each side will also use spies to gather information.

If a method involves software, someone could do memory dumps or run the software under the control of a debugger in order to understand the method. If hardware is being used, someone could buy or steal some of the hardware and build whatever programs or gadgets needed to test it. Hardware can also be dismantled so that the chip details can be seen with microscopes.

Maintaining security

A generalization some make from Kerckhoffs's principle is, "The fewer and simpler the secrets that one must keep to ensure system security, the easier it is to maintain system security." Bruce Schneier ties it in with a belief that all security systems must be designed to fail as gracefully as possible:

> Kerckhoffs's principle applies beyond codes and ciphers to security systems in general: every secret creates a potential failure point. Secrecy, in other words, is a prime cause of brittleness—and therefore something likely to make a system prone to catastrophic collapse. Conversely, openness provides ductility.[5]

Any security system depends crucially on keeping some things secret. However, Kerckhoffs's principle points out that the things kept secret ought to be those least costly to change if inadvertently disclosed.

For example, a cryptographic algorithm may be implemented by hardware and software that is widely distributed among users. If security depends on keeping that secret, then disclosure leads to major logistic difficulties in developing, testing, and distributing implementations of a new algorithm: it is "brittle." On the other hand, if keeping the algorithm secret is not important, but only the *keys* used with the algorithm must be secret, then disclosure of the keys simply requires the simpler, less costly process of generating and distributing new keys.

Applications

In accordance with Kerckhoffs's principle, the majority of civilian cryptography makes use of publicly known algorithms. By contrast, ciphers used to protect classified government or military information are often kept secret (see Type 1 encryption). However, it should not be assumed that government/military ciphers must be kept secret to maintain security. It's possible that they are intended to be as cryptographically sound as public algorithms, and the decision to keep them secret is in keeping with a layered security posture.

Eric Raymond extends this principle in support of open source security software, saying, "Any security software design that doesn't assume the enemy possesses the source code is already untrustworthy; therefore, never trust closed source."[6]

Implications for analysis

For purposes of analysing ciphers, Kerckhoffs's principle neatly divides any design into two components. The key can be assumed to be secret for purposes of analysis; in practice various measures are taken to protect it. Everything else is assumed to be knowable by the opponent, so everything except the key should be revealed to the analyst. Perhaps not all opponents know everything, but the analyst should because the goal is to create a system that is secure against any enemy except one that learns the key.

John Savard describes the widespread acceptance of this idea:

> That the security of a cipher system should depend on the key and not the algorithm has become a truism in the computer era, and this one is the best-remembered of Kerckhoffs's dicta. ... Unlike a key, an algorithm can be studied and analyzed by experts to determine if it is likely to be secure. An algorithm that you have invented yourself and kept secret has not had the opportunity for such review.[7]

Security through obscurity

It is moderately common for companies and sometimes even standards bodies as in the case of the CSS encryption on DVDs – to keep the inner workings of a system secret. Some argue this "security by obscurity" makes the product safer and less vulnerable to attack. A counter argument is that keeping the innards secret may improve security in the short term, but in the long run only systems that have been published and analyzed should be trusted.

Steve Bellovin commented:

> The subject of security through obscurity comes up frequently. I think a lot of the debate happens because people misunderstand the issue.
>
> It helps, I think, to go back to Kerckhoffs's second principle, translated as "The system must not require secrecy and can be stolen by the enemy without causing trouble," per http://petitcolas.net/fabien/kerckhoffs/). Kerckhoffs said neither "publish everything" nor "keep everything secret"; rather, he said that the system should still be secure *even if the enemy has a copy*.
>
> In other words – design your system assuming that your opponents know it in detail. (A former official at NSA's National Computer Security Center told me that the standard assumption there was that serial number 1 of any new device was delivered to the Kremlin.) After that, though, there's nothing wrong with trying to keep it secret – it's another hurdle factor the enemy has to overcome. (One obstacle the British ran into when attacking the German Enigma system was simple: they didn't know the unkeyed mapping between keyboard keys and the input to the rotor array.) But – *don't rely on secrecy*.[8]

References

This article incorporates material from the Citizendium article "Kerckhoffs' Principle", which is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License but not under the GFDL.

[1] Kahn, David (second edition, 1996), *The Codebreakers: the story of secret writing*, Scribners p.235

[2] Peticolas, Fabien, *electronic version and English translation of "La cryptographie militaire"* (http://petitcolas.net/fabien/kerckhoffs/),

[3] Auguste Kerckhoffs, "La cryptographie militaire" (http://www.petitcolas.net/fabien/kerckhoffs/) *Journal des sciences militaires*, vol. IX, pp. 5–83, January 1883, pp. 161–191, February 1883.

[4] Massey, James L (1993), *Cryptography: Fundamentals and Applications, course notes* p.2.5

[5] Mann, Charles C. (September 2002), "Homeland Insecurity" (http://www.theatlantic.com/issues/2002/09/mann.htm), *The Atlantic Monthly* **290** (2), .

[6] Raymond, Eric S. (May 17, 2004). "If Cisco ignored Kerckhoffs's law, users will pay the price" (http://lwn.net/Articles/85958/). LWN.net. .

[7] Savard, John J. G., "The Ideal Cipher" (http://www.quadibloc.com/crypto/mi0611.htm), *A Cryptographic Compendium*,

[8] Bellovin, Steve (June, 2009), "Security through obscurity" (http://catless.ncl.ac.uk/Risks/25.71.html#subj19), *Risks Digest*,

External links

- John Savard article discussing Kerckhoffs's design goals for ciphers (http://www.quadibloc.com/crypto/mi0611.htm)
- Reference to Kerckhoffs's original paper, with scanned original text (http://petitcolas.net/fabien/kerckhoffs/)

Article Sources and Contributors

Kerckhoffs's principle *Source*: http://en.wikipedia.org/w/index.php?oldid=540296535 *Contributors*: Angela, BD2412, Banana04131, Bruce1ee, Burn, Cybercobra, Daedelus, Darekun, David Shay, Dougher, Elwikipedista, EoGuy, Fredrik, Frédérick Lacasse, Gene.arboit, Hairy Dude, Hullo exclamation mark, Inkling, Jdimpson, Kesla, LapoLuchini, Ldo, Llavigne, Matt Crypto, Mauls, Michael Hardy, Michael miceli, Mike 7, Murtasa, Nealmcb, Nihil novi, Ninly, Noel Bush, NuclearWarfare, Officiallyover, Pacaro, Paul Richter, PierreAbbat, Piet Delport, Psinu, Quondum, Roentgenium111, Samuel Sol, Sanilunlu, Sinar, Sliwers, The Anome, The wub, Toh, Tregoweth, Tristan Schmelcher, Tyler Oderkirk, Wavelength, Wik, Wikih101, Wilsonjohna, Wonderstruck, Ww, 62 anonymous edits

License