

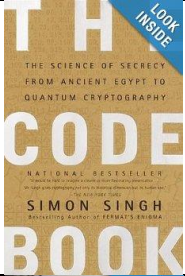
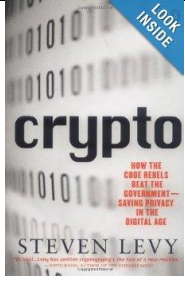
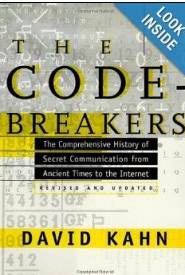

## Cryptography, Symmetric Cryptography

- Block ciphers
  - Block ciphers encrypt bits a block at a time, which is much more efficient than, meaning most encryption uses block ciphers because it is more efficient than stream ciphers that only operate on a single bit at a time.
  - Block ciphers work by mapping a block of bits to a different permutation of a block the same size.
  - For a block size of 9 (a 3-by-3 block of bits), there are  $2^9 = 512$  possible variations. However, the size of all possible mappings of a set of 512 variations to another set of 512 variations is much larger: 512! (<http://www.wolframalpha.com/input/?i=512!>). This is a toy example: modern block sizes are 128 or 256 bits in size.
  - Examples of common block ciphers today are AES, Blowfish, Twofish, and Serpent.
  - DES and 3DES use a 64-bit block size that is considered too small to be secure. See this recent attack on SSL/TLS connections that use 3DES:
    - <https://blog.cryptographyengineering.com/2016/08/24/attack-of-week-64-bit-ciphers-in-tls/>
    - <https://sweet32.info>
  - Video of AES algorithm visualized: <https://www.youtube.com/watch?v=gP4PqVGudtg>
  - Modes of operation.
    - Block ciphers need to be used in a certain way, called a mode of operation.
    - There are many modes of operation that you can read about, including: ECB, CBC, OFB, and more. See section 5.5 of Anderson, *Security Engineering* for more info. Also: [https://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation).
    - Electronic code book (ECB) mode encrypts every block of encryption the same way every time. This is a problem because frequency analysis of the encrypted text can reveal a lot of information.
      - $C_i = \text{Encrypt}(\text{Key}, P_i)$ , for  $i = 1, \dots, k$
    - In contrast, Cipher block chaining (CBC) mode XORs a plaintext with the last encrypted block before encrypting it. This ensures that the same plaintext is encrypted differently every time.
    - CBC mode requires an initialization vector (or IV) to get started, since the first block doesn't have a previous encrypted block to XOR against.
      - $C_i = \text{Encrypt}(\text{Key}, P_i \oplus C_{i-1})$ , for  $i = 1, \dots, k$
- Hashing
  - Hashing provides integrity.
  - This is useful to ensure that files are unchanged in transmission. See this example: <https://www.kali.org/downloads/>
  - Hashes are like digital fingerprints: small pieces of data that serve to identify much larger objects. See "Secrets and Lies," page 94.
  - Block and stream ciphers are two way functions. They are reversible: you can use them to encrypt and decrypt.
  - In contrast, hashes are one-way functions. You can only compute a hash. You can't reverse it.
    - For this reason, hashing uses no key.
  - Another name for a hash is a message digest, because a hash digests an input of any size down to a fixed output.

- Conceptually, this is the opposite of a pseudo-random number generator that accepts a seed of a fixed size and then generates random data of variable size.
- With hashing, a very small change in the file causes what's known in cryptography as the "avalanche effect" a very large, unpredictable change in the hash results.
- Hashes should be resistant to collisions. With a fixed number of possible hashes, and an infinite number of possible inputs, collisions are inevitable. However, it should be difficult (i.e., practically impossible) to find collisions on purpose.
  - This should be hard to find:  $\text{hash}(\text{Input}_1) = \text{hash}(\text{Input}_2)$ .
- The birthday theorem makes collision even easier, which is for a set of  $2^n$ , you only need  $2^{n/2}$  to find a collision.
- Current hashing algorithms:
  - MD5
  - Still very common
  - 128-bit hash, but only need  $2^{128/2} = 64$  to find a collision. This is not strong enough for modern computers.
  - MD5 is broken.
  - Finding this collision is trivial: Find  $m_1$  and  $m_2$  such that:  $H(m_1) = H(m_2)$
  - This collision is hard, but possible for a supercomputer: Given  $p_1$  and  $p_2$ , find  $m_1$  and  $m_2$  such that:  $H(p_1 \parallel m_1) = H(p_2 \parallel m_2)$ .
    - The NSA malware Flame performed a MD5 hash collision to hijack Microsoft Windows Update to spread:
      - <http://arstechnica.com/security/2012/06/flame-crypto-attack-may-have-needed-massive-compute-power/>
      - [https://www.washingtonpost.com/world/national-security/us-israel-developed-computer-virus-to-slow-iranian-nuclear-efforts-officials-say/2012/06/19/gJQA6xBPoV\\_story.html](https://www.washingtonpost.com/world/national-security/us-israel-developed-computer-virus-to-slow-iranian-nuclear-efforts-officials-say/2012/06/19/gJQA6xBPoV_story.html)
- SHA-1 uses 160-bits, but attacks already exist. This is way major software companies will no longer accept encryption that uses SHA-1 beginning in 2017.
- <https://securityintelligence.com/browser-vendors-are-shutting-down-sha-1-digital-certificates/>
- <https://security.googleblog.com/2015/12/an-update-on-sha-1-certificates-in.html>
- <https://shaaaaaaaaaaaaaaaa.com/>
- SHA1 Crack, "SHAttered": <https://shattered.it>
- Announcement: <https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html>
- <https://arstechnica.com/security/2017/02/at-deaths-door-for-years-widely-used-sha1-function-is-now-dead/>
- SHA-2 uses 224, 256, 384, and 512-bit hashes, but it is built off of the same design as SHA-1, and is therefore prone to the same weaknesses. It's believed to be a matter of time before SHA-2 is also exploited.
- SHA-3 was just ratified last year by NIST, the U.S. National Institute of Standards and Technology. It was the result of a six-year hashing competition.
  - <https://en.wikipedia.org/wiki/SHA-3>

## Lifelong Learning

To learn more about cryptography, the following are some of the best and most readable books.

	<p>“The Code Book” by Simon Singh.</p> <p>This is a very interesting and gripping book about the history and intrigue of cryptography and cryptanalysis.</p> <p>BYU Library: <a href="http://search.lib.byu.edu/byu/id:byu2657194">http://search.lib.byu.edu/byu/id:byu2657194</a>  Amazon: <a href="http://amzn.com/0470474246">http://amzn.com/0470474246</a></p>
	<p>“Crypto: How the Code Rebels Beat the Government Saving Privacy in the Digital Age” by Steven Levy.</p> <p>A very engaging look at the modern history of cryptography, including the development of DES, RSA, and PGP. Also, it describes the fight in the 1990’s to legalize the use of strong cryptography.</p> <p>BYU Library: <a href="http://search.lib.byu.edu/byu/id:byu2931203">http://search.lib.byu.edu/byu/id:byu2931203</a>  Amazon: <a href="http://amzn.com/0140244328">http://amzn.com/0140244328</a></p>
	<p>“The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet” by David Kahn.</p> <p>The bible of the history of Cryptography. A deep dive (at 1,200 pages) but still readable.</p> <p>BYU library: <a href="http://search.lib.byu.edu/byu/id:byu2089306">http://search.lib.byu.edu/byu/id:byu2089306</a>  Amazon: <a href="http://amzn.com/0684831309">http://amzn.com/0684831309</a></p>
	<p>“Cryptography Engineering: Design Principles and Practical Applications” by Niels Ferguson, Bruce Schneier, Tadayoshi “Yoshi” Kohno.</p> <p>The updated successor to the classic “Applied Cryptography” by Bruce Schneier. This book explains how modern crypto works and how to implement it in your systems.</p> <p>BYU Library: <a href="http://search.lib.byu.edu/byu/id:byu4424191">http://search.lib.byu.edu/byu/id:byu4424191</a>  Amazon: <a href="http://amzn.com/0470474246">http://amzn.com/0470474246</a></p>