

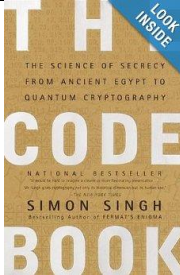
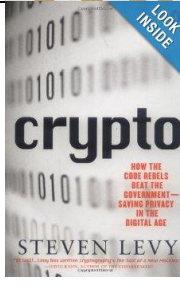
## Asymmetric Cryptography, 2/16/2021

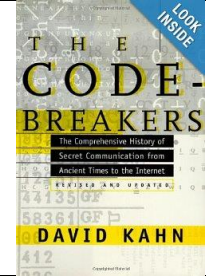

- Key sharing problem:
  - In order to confidentially communicate, you must first create a “secure channel.” However, this requires that you first somehow exchange secret encryption keys.
  - This required in-person meetings, physical couriers, etc.
  - Also, Metcalf’s Law explains that in a group in which each member must be able to communicate secretly with each other member of the group, as the group size grows, the number of secret keys required increases exponentially.  
[https://en.wikipedia.org/wiki/Metcalf%27s\\_law](https://en.wikipedia.org/wiki/Metcalf%27s_law)
  - This problem is as old as cryptography, but it all changed in 1976, with the invention of public key cryptography, in the form of the Diffie-Hellman key exchange.
    - <http://math.boisestate.edu/~liljanab/MATH308/NewDirectionsCryptography.pdf>
    - 2016 Turing award:  
<http://www.nytimes.com/2016/03/02/technology/cryptography-pioneers-to-win-turing-award.html>
- Diffie-Hellman
- Discrete logarithm problem
- Video explaining how it works: <https://www.youtube.com/watch?v=YEBfamv-do>
- Example 1:
  - $p = 23, g = 5$ 
    1. Choose  $x$ , a number between 1 and  $p - 1$ .
    2. Compute  $g^x \bmod p$ .
    3. Tell this number to your partner. Your partner will tell you  $(g^{\text{secret}} \bmod p)$ . Call this  $y$ .
    4. Compute the key as  $y^x \bmod p$ .
    5. Eve reveals her derived key.
    6. Alice and Bob reveal their key.
- Example 2:
  - $p = 31, g = 3$
  - Same steps as before.
- Problems with Diffie-Hellman:
  - Have to negotiate a key for every new person.
  - No authentication.
  - These problems were solved with another groundbreaking invention the following year in 1977, the RSA algorithm.
- RSA
  - Named for inventors Ron Rivest, Adi Shamir, and Leonard Adelman.
  - Original article: <http://case.ntu.edu.tw/blog/wp-content/uploads/2013/10/p120-rivest.pdf>
  - 2002 Turing award: <http://www.ams.org/notices/200307/comm-turing.pdf>
  - Ron Rivest on RSA and RSA-129: <https://youtu.be/YQw124CtvOQ>
  - RSA numbers: [https://en.wikipedia.org/wiki/RSA\\_numbers](https://en.wikipedia.org/wiki/RSA_numbers)
  - Symmetric encryption: use the same keys to encrypt and decrypt.
  - Asymmetric encryption: use different keys to encrypt and decrypt.
  - Provides authentication
    - $\text{privatekey}(m) = c$
    - $\text{publickey}(c) = m$
    - Where  $m = \text{message in plaintext}$ ,  $c = \text{ciphertext}$
  - Non-repudiation

- Can be used to create digital signature
  1. hash(message)
  2. privatekey(hash) = c
  3. publickey(c) = hash
  4. Recalculate the hash
- Asymmetric cryptography is 1000 times slower than symmetric cryptography.
- For this reason, asymmetric cryptography is typically only used to exchange a symmetric key (such as a 128-bit AES key), at which time the communicating parties switch to using symmetric encryption. This is called a *hybrid crypto system*.
- For asymmetric cryptography, good key lengths are 2048, 4096-bits. This is because the strength of the key rests not in how long it would take to guess the correct key out of all possible keys, but rather in the difficulty of solving the discrete logarithm problem or factoring the product of two large prime numbers.
- Perfect forward secrecy (PFS):
  - Also known as key erasure
  - Google started using PFS in 2011. Now this is a best practice.
    - <https://security.googleblog.com/2011/11/protecting-data-for-long-term-with.html>
  - Heartbleed SSL attack leaked private keys
    - <https://www.eff.org/deeplinks/2014/04/why-web-needs-perfect-forward-secrecy>
  - Example: Lavabit
    - <https://www.wired.com/2016/03/lavabit-apple-fbi/>
    - <https://www.wired.com/2016/03/government-error-just-revealed-snowden-target-lavabit-case/>
    - <http://www.wired.com/2014/04/lavabit-ruling/>

## 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>