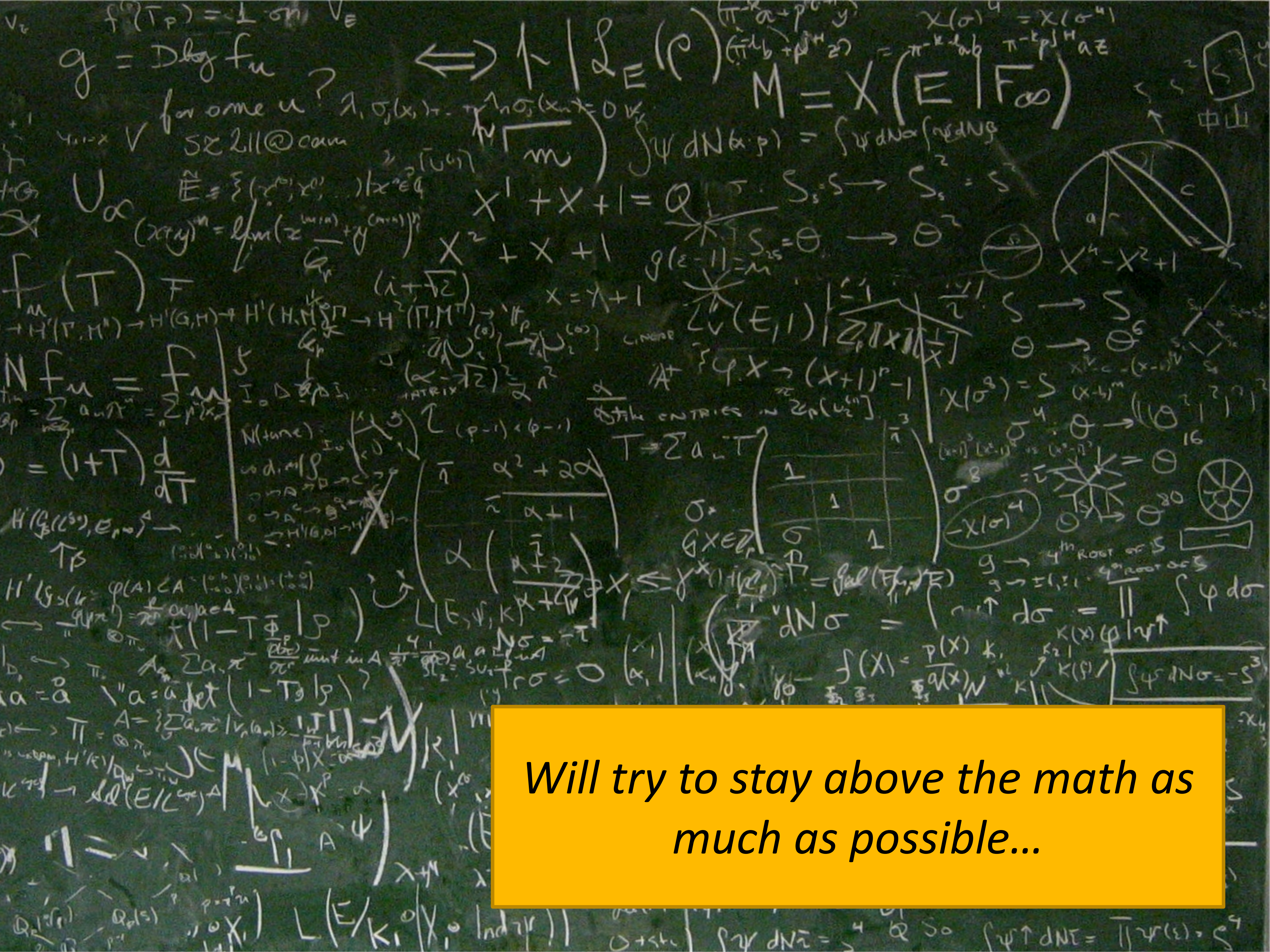
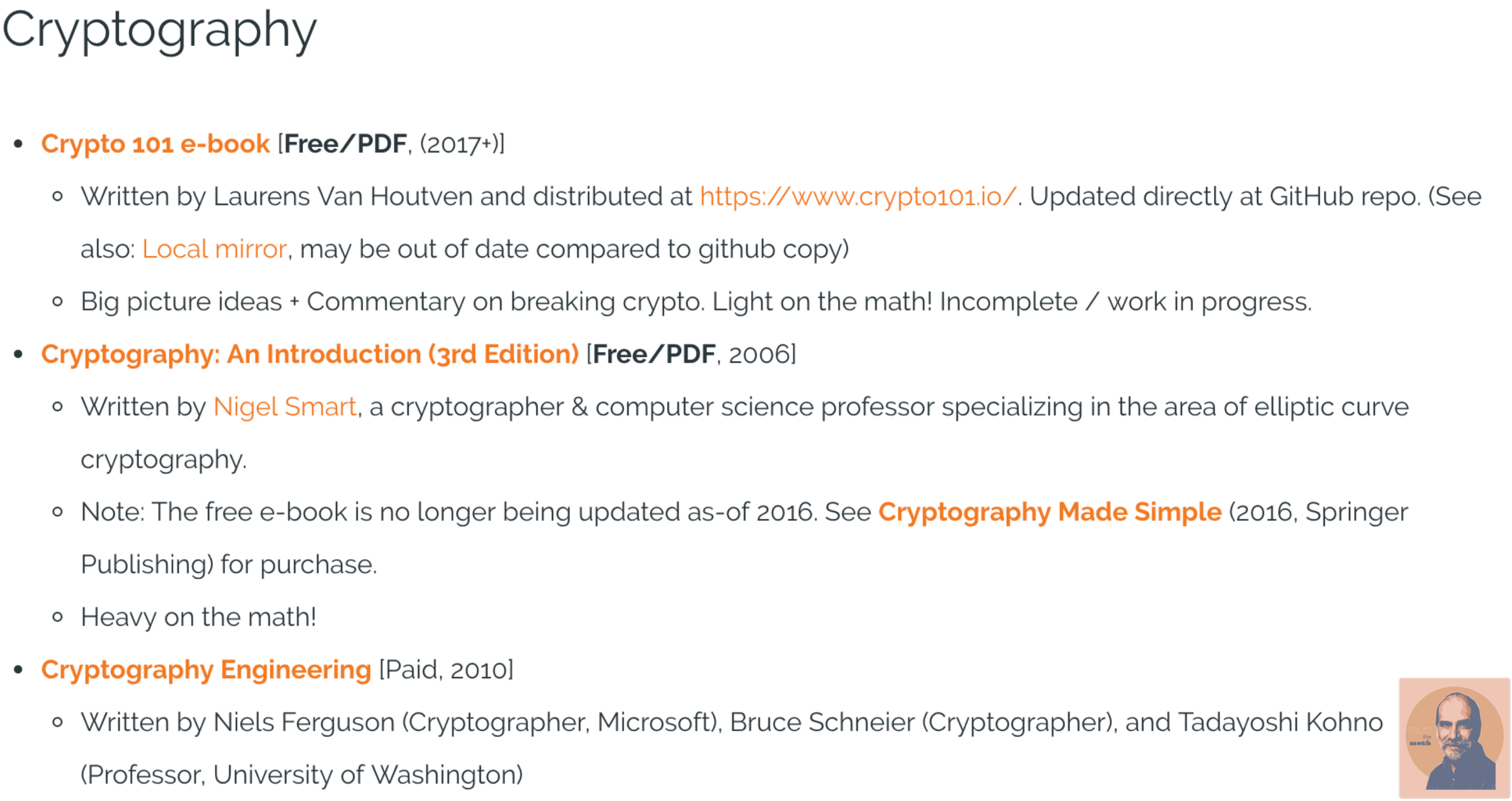
Cryptography



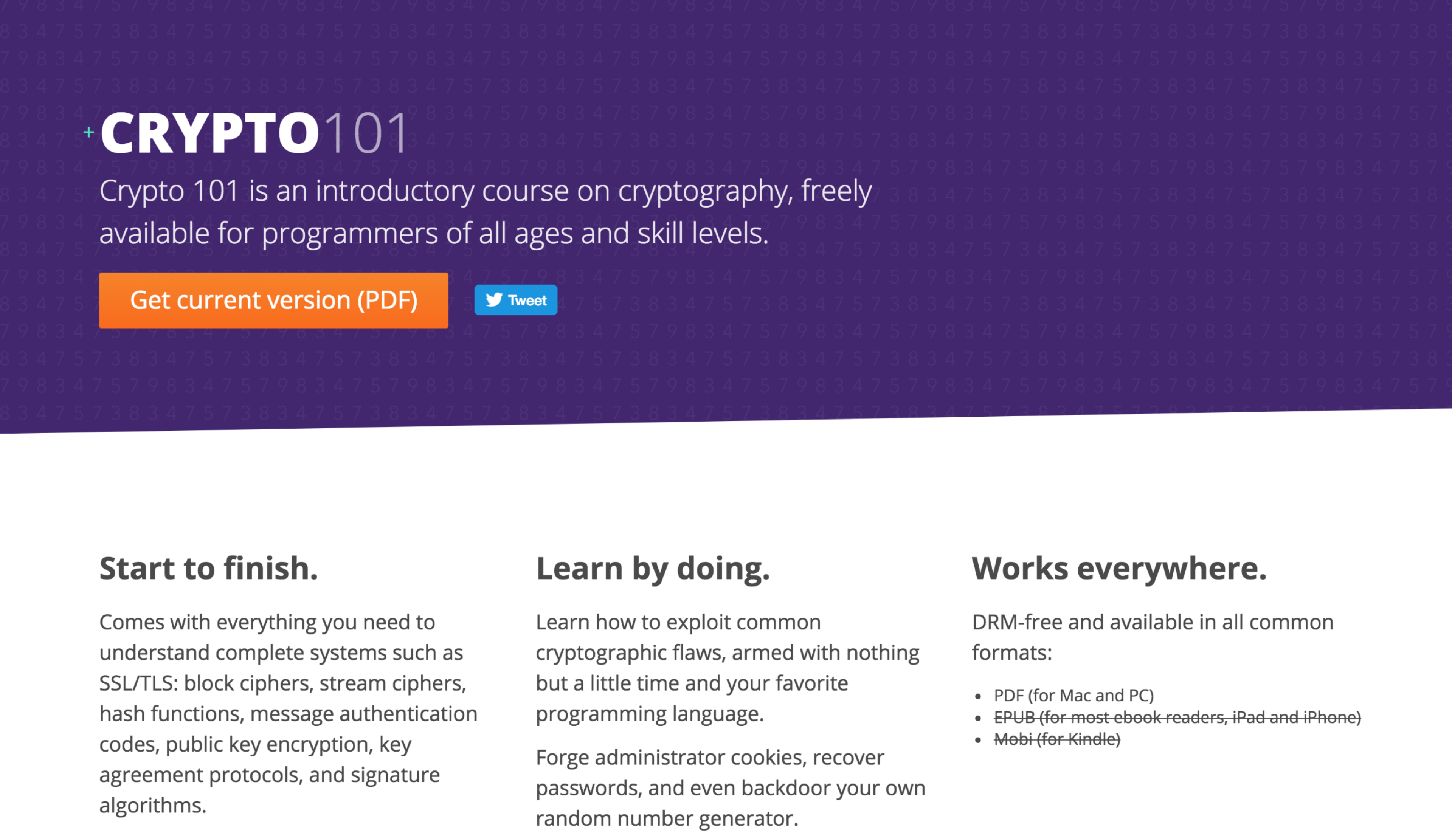
Will try to stay above the math as much as possible…

Useful References on Class Website



Cryptography

- **Crypto 101 e-book** [**Free/PDF**, (2017+)]
  - Written by Laurens Van Houtven and distributed at https://www.crypto101.io/. Updated directly at GitHub repo. (See also: Local mirror, may be out of date compared to github copy)
  - Big picture ideas + Commentary on breaking crypto. Light on the math! Incomplete / work in progress.
- **Cryptography: An Introduction (3rd Edition)** [**Free/PDF**, 2006]
  - Written by Nigel Smart, a cryptographer & computer science professor specializing in the area of elliptic curve cryptography.
  - Note: The free e-book is no longer being updated as-of 2016. See **Cryptography Made Simple** (2016, Springer Publishing) for purchase.
  - Heavy on the math!
- **Cryptography Engineering** [Paid, 2010]
  - Written by Niels Ferguson (Cryptographer, Microsoft), Bruce Schneier (Cryptographer), and Tadayoshi Kohno (Professor, University of Washington)



Crypto 101 is an introductory course on cryptography, freely available for programmers of all ages and skill levels.

Start to finish.

Comes with everything you need to understand complete systems such as SSL/TLS: block ciphers, stream ciphers, hash functions, message authentication codes, public key encryption, key agreement protocols, and signature algorithms.

Learn by doing.

Learn how to exploit common cryptographic flaws, armed with nothing but a little time and your favorite programming language.

Forge administrator cookies, recover passwords, and even backdoor your own random number generator.

Works everywhere.

DRM-free and available in all common formats:

- PDF (for Mac and PC)
- ~~EPUB (for most ebook readers, iPad and iPhone)~~
- ~~Mobi (for Kindle)~~

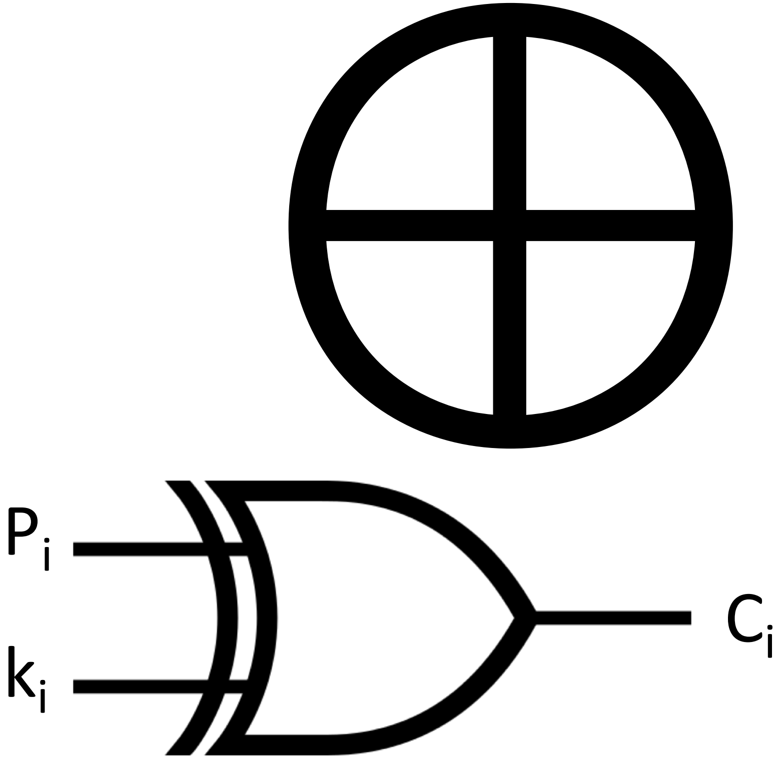
https://www.crypto101.io/

Has a video too!

https://www.youtube.com/watch?v=3rmCGsCYJF8

XOR & One Time Pads

XOR



P_i	k_i	C_i
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

P_i = Plaintext (bit i)

K_i = Key (bit i)

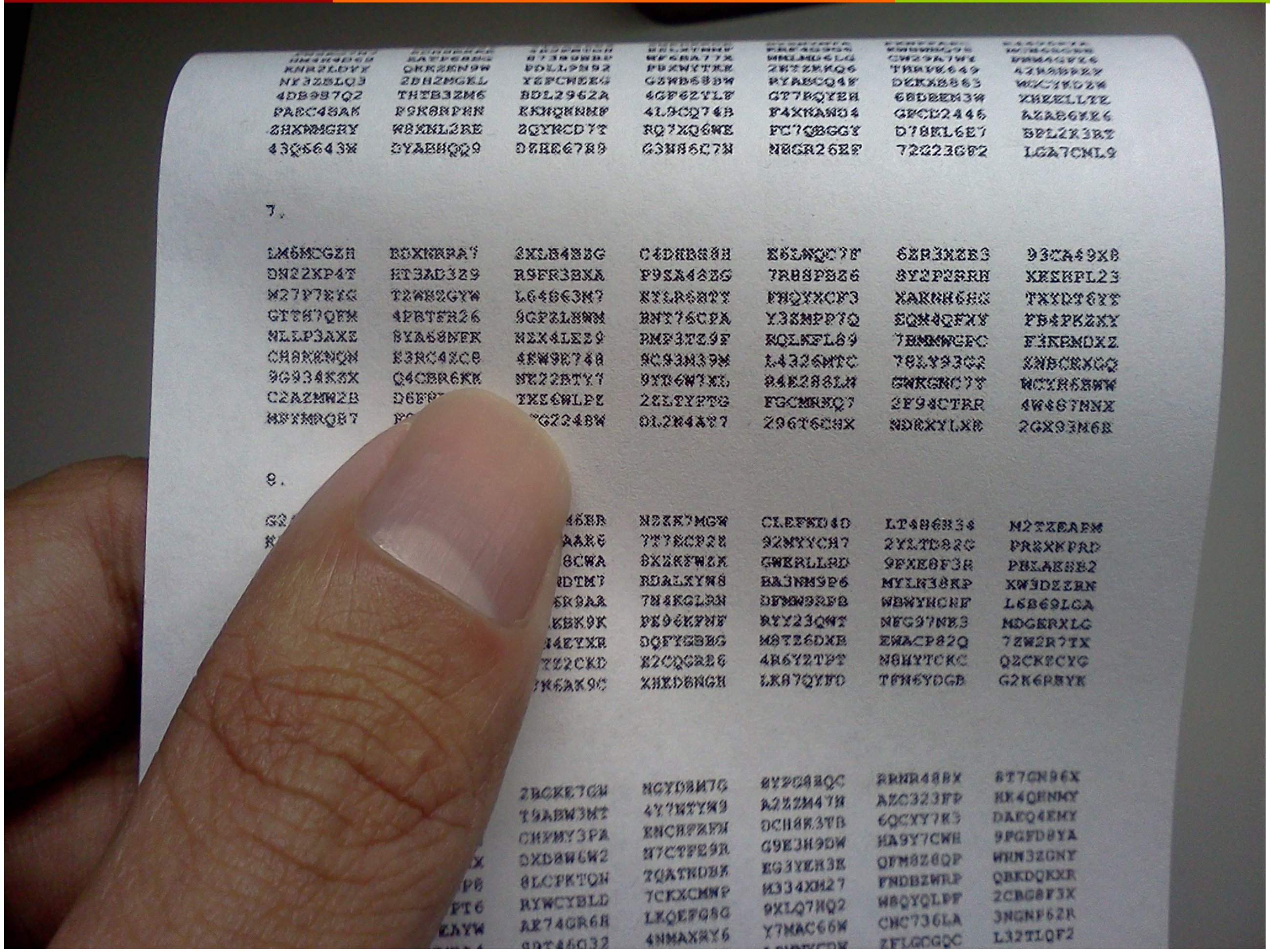
C_i = Ciphertext (bit i)

Useful XOR Properties

- Associative: Apply XOR in any order:
  $a \oplus (b \oplus c) = (a \oplus b) \oplus c$
- Commutative: Flip operands around:
  $a \oplus b = b \oplus a$
- Useful property for encryption:

$$\begin{aligned} a \oplus b \oplus a &= a \oplus a \oplus b \\ &= 0 \oplus b \\ &= b \end{aligned}$$

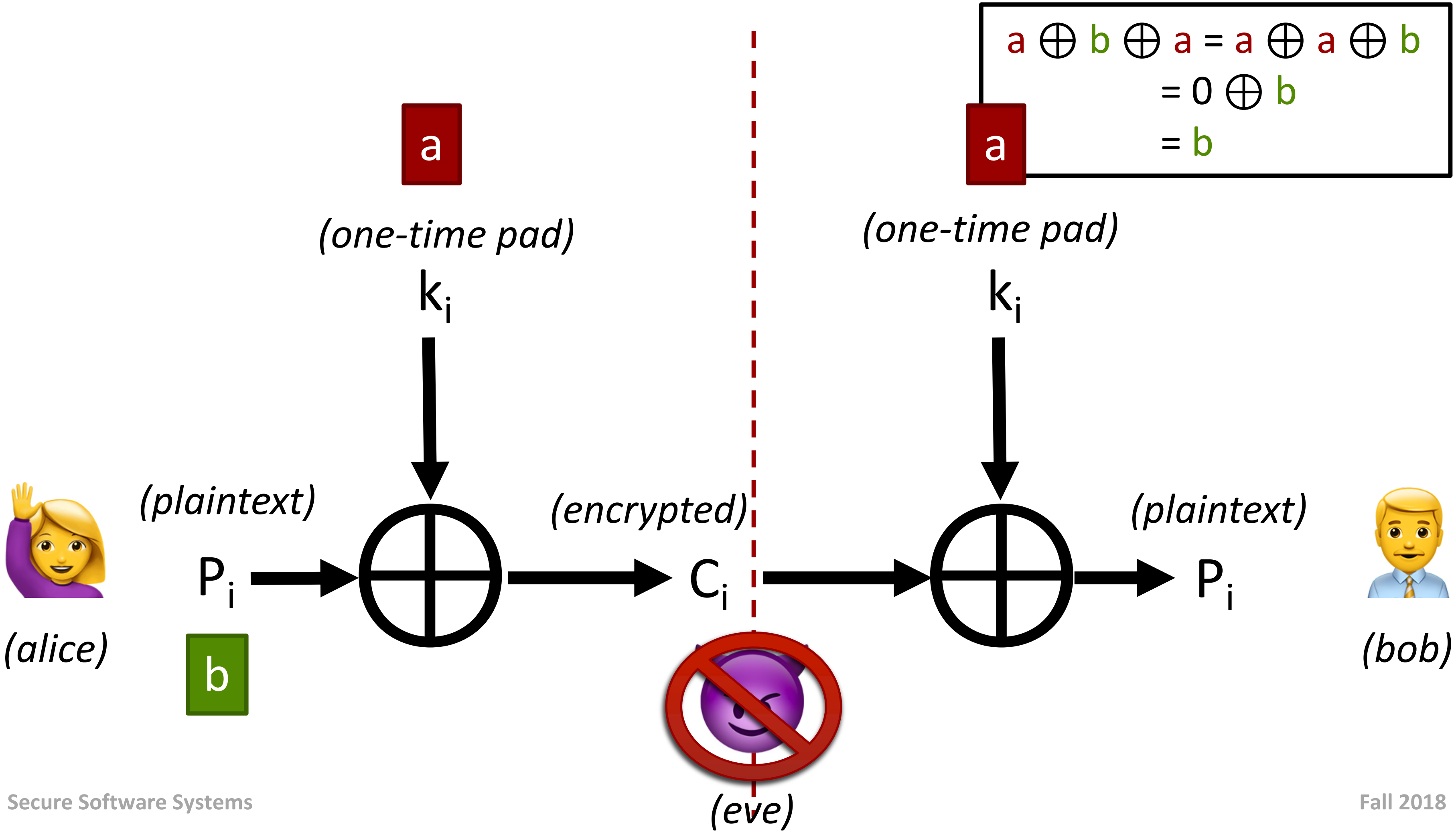
One-Time Pad (OTP)



Pencil and paper technique

Dates back to 1882 (Frank Miller, designing for telegraph machines)

One-Time Pad (OTP)



$$
\begin{aligned}
a \oplus b \oplus a &= a \oplus a \oplus b \\
&= 0 \oplus b \\
&= b
\end{aligned}
$$

One-Time Pad (OTP)

- Ciphertext provides no information about the original message (beyond length)
  - Thus, OTP has property of **Perfect Secrecy** ☺
  - Eve has no idea if P_i was 0 or 1 - Cannot be broken!
- But, Perfect Secrecy property is **only** true **if:** ☹
  - **….**

One-Time Pad Problems

- One-Time Pad must be used only **<u>once</u>**!
  - Problem: Repeated use allows statistical comparisons – easy to break
- One-Time Pad must be (cryptographically) random
  - Problem: Random number generation is hard/slow
- One-Time Pad must be shared between parties
  - Problem: Sharing must occur in advance of communication, out of band?
  - Both parties must keep pad secret
- One-Time Pad must equal length of plaintext
  - Problem: Length!

Beyond the One-Time Pad

- What do we use instead? **Ciphers!**
  - Manageable key sizes ☺
  - Methods to negotiate keys over the Internet with parties we've never communicated with before ☺
  - Systems are more complex and lack the theoretical Perfect Secrecy property of OTPs ☹