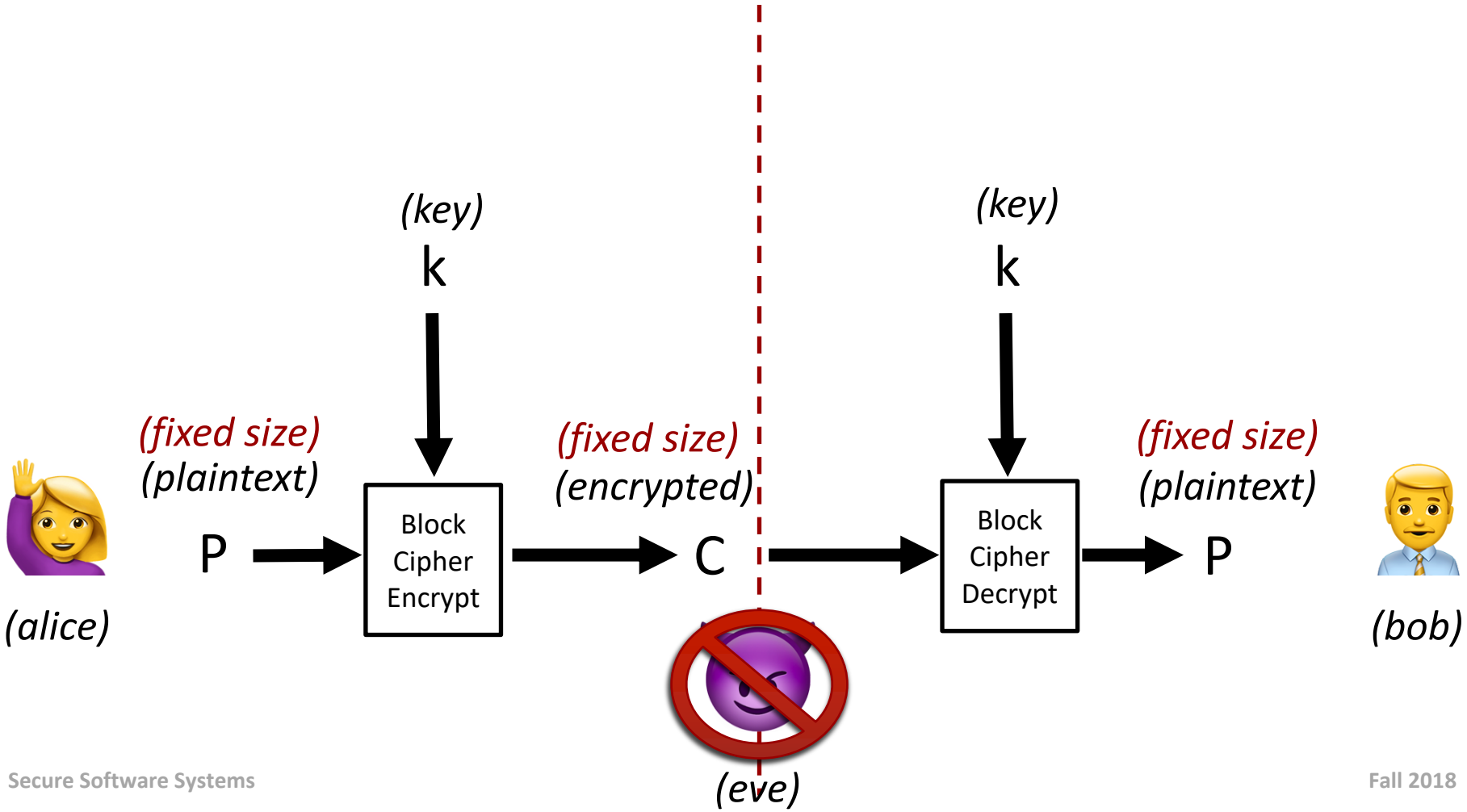
Block Ciphers

Block Cipher

- Encryption function *E*
  C = E(k, P)
- Decryption function *D*
  P = D(k, C)
- **Symmetric-key encryption**
  - Same key is used for both encryption and decryption
- Operates not bit-by-bit but block-by-block
  - Block is a collection of bits
  - Block size is *fixed* by the cipher

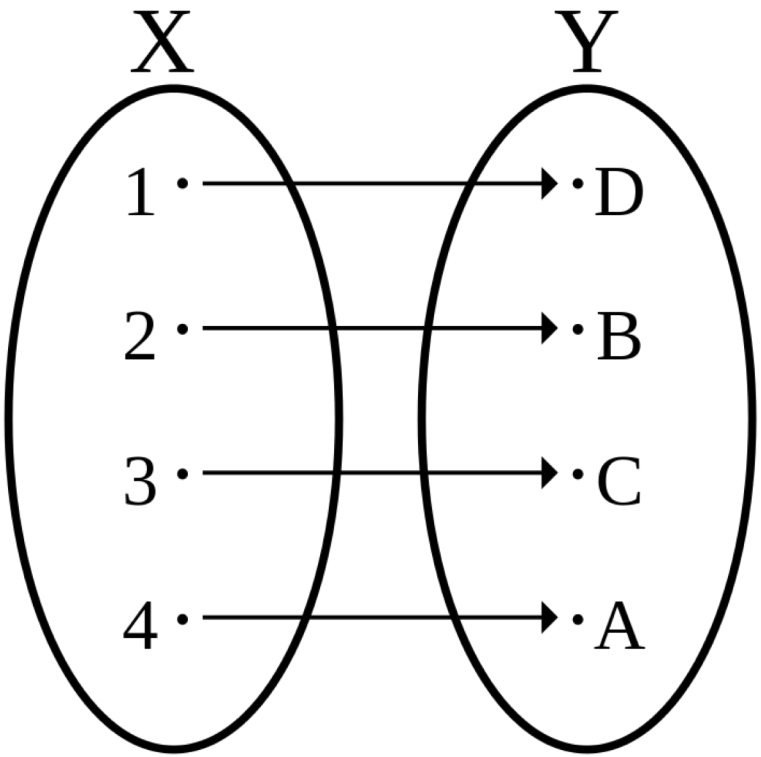
Block Cipher



Block Cipher

- What's in the box?
- **Keyed Permutation**
  - Permutation – every input block is mapped to a unique output block
    - Reversible!
  - Keyed – The key determines the exact mapping
    - *Mapping varies by the key*

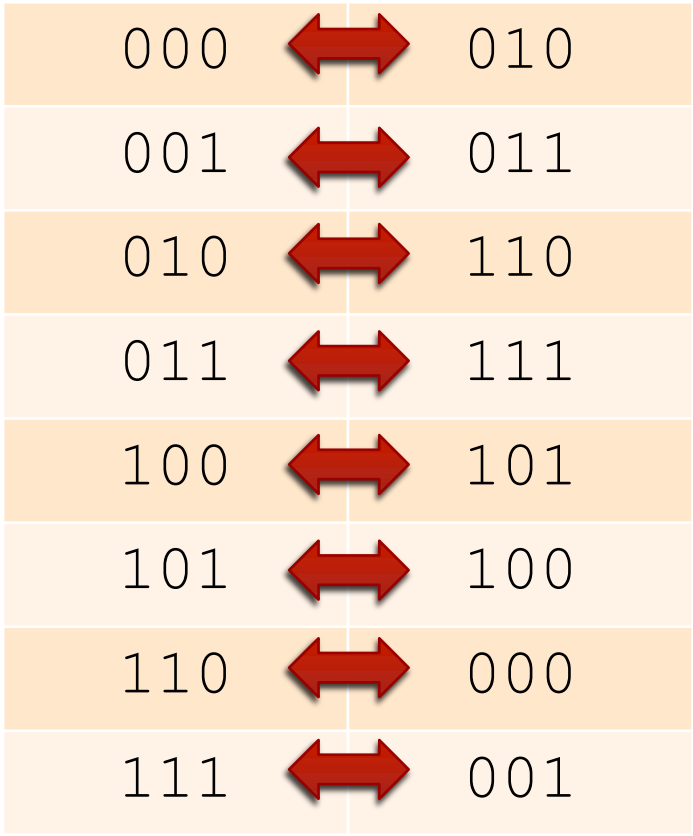
Bijection



- **Bijective function**
- Function between the elements of two sets
  - Each element of set X is paired with exactly one element of set Y
  - Each element of set Y is paired with exactly one element of X
  - **There are no unpaired elements**

Block Cipher

- **Example Block Cipher**
  - 3 bit block size (tiny!)



Block Cipher

- Assuming no fatal flaws in mapping, brute forcing key is only path for attacker
  - Must try every potential keys
  - $2^3 = 8$ possible blocks
  - $2^{128} = 3.4 \times 10^{38}$ possible blocks
- $n!$ permutations of an n element set
  - 128 bit key?
    $(2^{128})!$ permutations

Block Cipher

Pros

- Key are short / fixed length
  - 128-256 bits
- Much easier to communicate to recipient than a One-Time Pad!

Cons

- What happens if the **plaintext is larger than the block size**?
  - Example: AES block size is 128 bits (16 bytes)
- How do the sender and receive agree on the same key over an **insecure channel**?

Block Cipher Examples

- AES – Advanced Encryption Standard
- Blowfish/Twofish/Threefish
- DES/3DES

Block Cipher Examples - AES

- Public peer-reviewed competition to select next-generation block cipher (AES)
  - Sponsored by NIST, 1997-2001
  - Evaluated on: security, licensing, implementation
- **Security**
  - "The extent to which the algorithm output is indistinguishable from a random permutation on the input block"
  - "Soundness of the mathematical basis for the algorithm's security"
  - "Other security factors **raised by the public** during the evaluation process, including any attacks which demonstrate that the actual security of the algorithm is less than the strength claimed by the submitter"
- **Licensing** (worldwide, non-exclusive, royalty-free)
- **Algorithm / Implementation**
  - Flexibility (e.g., additional key sizes, additional block sizes, stream cipher, …)
  - Computational efficiency / Memory requirements
  - Hardware and software implementation suitability
  - Simplicity ☺

https://competitions.cr.yp.to/aes.html

Block Cipher Examples - AES

- **AES – Advanced Encryption Standard**
  - Winning cipher: Rijndael (pronounced “Rhine Dahl”)
    - Developed by two Belgian cryptographers: Vincent Rijmen and Joan Daemen
    - Block size: 128 bits
    - Key size: 128, 192, 256 bits
  - Standardized: FIPS PUB 197 and ISO/IEC 18033-3

Block Cipher Examples - *Fish

- Three algorithms created by Bruce Schneier
- **Blowfish** (1993) – **Don't use**!
- **Twofish** (1998)
  - One of five finalists in the AES competition
  - "I have nothing but good things to say about NIST and the AES process."
    ~ Bruce Schneier
- **Threefish** (2008)

Block Cipher Examples – DES/3DES

- **DES** – Data Encryption Standard – **Don't use**!
  - Designed by IBM in 1975
  - Algorithm is "basically sound" (unless you have a large number of plaintexts to feed through system), but 56 bit key is too small on modern hardware
    - Brute forced in under a day using 1999 hardware
  - Withdrawn as standard by NIST (Replaced by AES)
- **3DES** ("triple DES") – **Don't use**!
  - Run DES three times….
  - Secure, but very very slow (10x worse than AES)
    - Only bother if dealing with legacy systems ☹

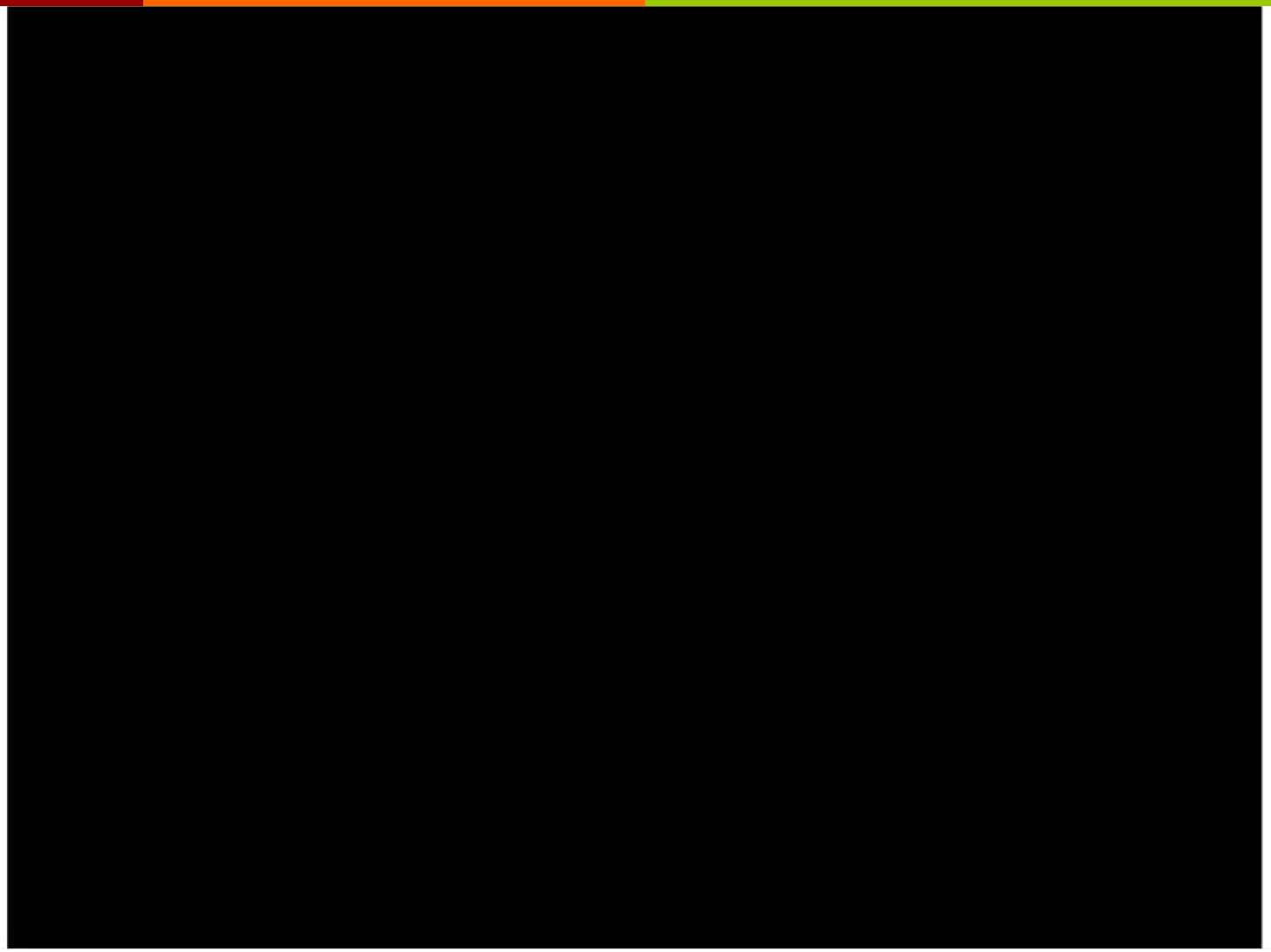
Deep Dive – AES Rijndael Cipher

AES Rijndael Cipher

- Design: **Substitution-Permutation Network**
- Resource: Interactive AES calculator (change bytes in plaintext or key, see how they affect the result)
  - https://www.cryptool.org/en/cto-highlights/aes

AES Rijndael Cipher

- KeyExpansions – expand single input key into multiple keys (1 per round) using Rijndael's key schedule
- Initial Round
  - **AddRoundKey** – each byte of the state is combined with a block of the round key using bitwise xor
- Rounds 2-*n*
  - **SubBytes** – non-linear substitution step where each byte is replaced with another according to a lookup table.
  - **ShiftRows** – transposition step where the last three rows of the state are shifted cyclically a certain number of steps.
  - **MixColumns** – mixing operation which operates on the columns of the state, combining the four bytes in each column.
  - **AddRoundKey**
- Final Round (no MixColumns)
  - SubBytes
  - ShiftRows
  - AddRoundKey

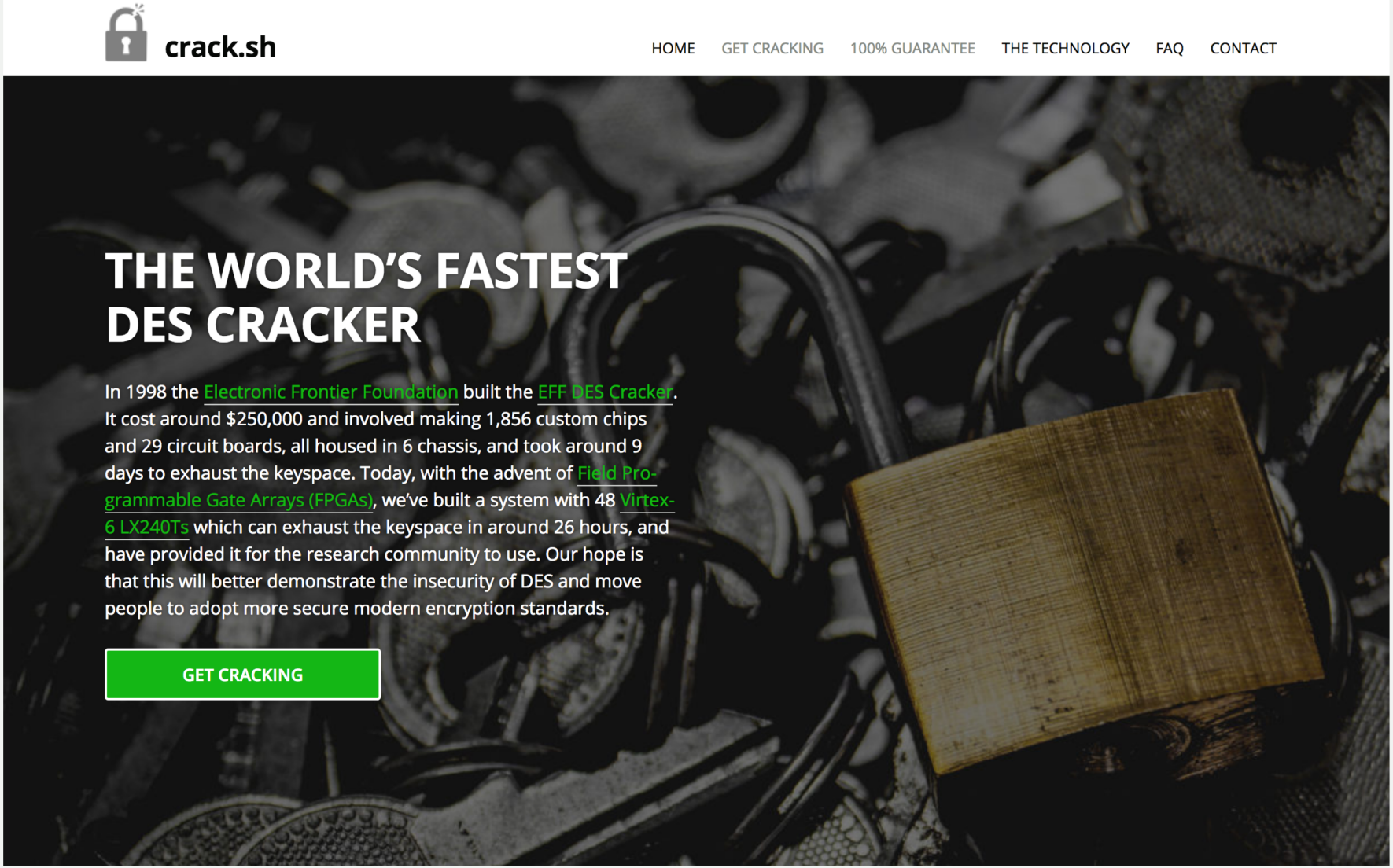


https://www.youtube.com/watch?v=mlzxpkdXP58

Breaking Cryptography (AES/DES)

Breaking Cryptography

- General methods
  - **Brute force**
    - Time / $$$
    - Cryptographic “break” is any attack *faster* than brute force
  - **Attack the cipher** (“Smart mathematicians”)
    - Reduce the search space for brute force attacks
    - Make repeated observations and intelligent guesses about keys
  - **Attack the implementation** (“Smart engineers”)
    - Side channel attacks (Cache timing, power usage, software-initiated attacks like rowhammer, …)



https://crack.sh/

Breaking Crypto – DES with crack.sh

- 48 Xilinx Virtex-6 LX240T FPGAs
  - 40 DES cores
  - 400MHz
  - 16×10^9 keys/sec/FPGA
  - 768×10^9 keys/sec total
- **Exhaustively search 56-bit DES key-space in:**
  2^{56} / 768,000,000,000
  = ~26 hours
- Formats supported (Single-DES)
  - NETLM/NETNTLMv1 Authentication
  - PPTP VPNs
  - WPA-Enterprise
  - des_crypt() Hashes
  - Kerberos5 DES
  - Known Plaintext DES
- Price: $20-$100

Breaking Crypto - AES

- AES cipher has been "broken"
  - Key recovery faster than brute force by a factor of 4
    - 128 bit key now offers security of 126 bits
    - 256 bit key now offers security of 254
  - Good math work, but *yawn…*
    - *Time to brute force before: ∞*
    - *Time to brute force now: ∞ / 4*

Research paper

A. Bogdanov, D. Khovratovich, C. Rechberger, "Biclique Cryptanalysis of the Full AES". IACR Cryptology ePrint Archive. 344-371. 10.1007/978-3-642-25385-0_19, **2011**

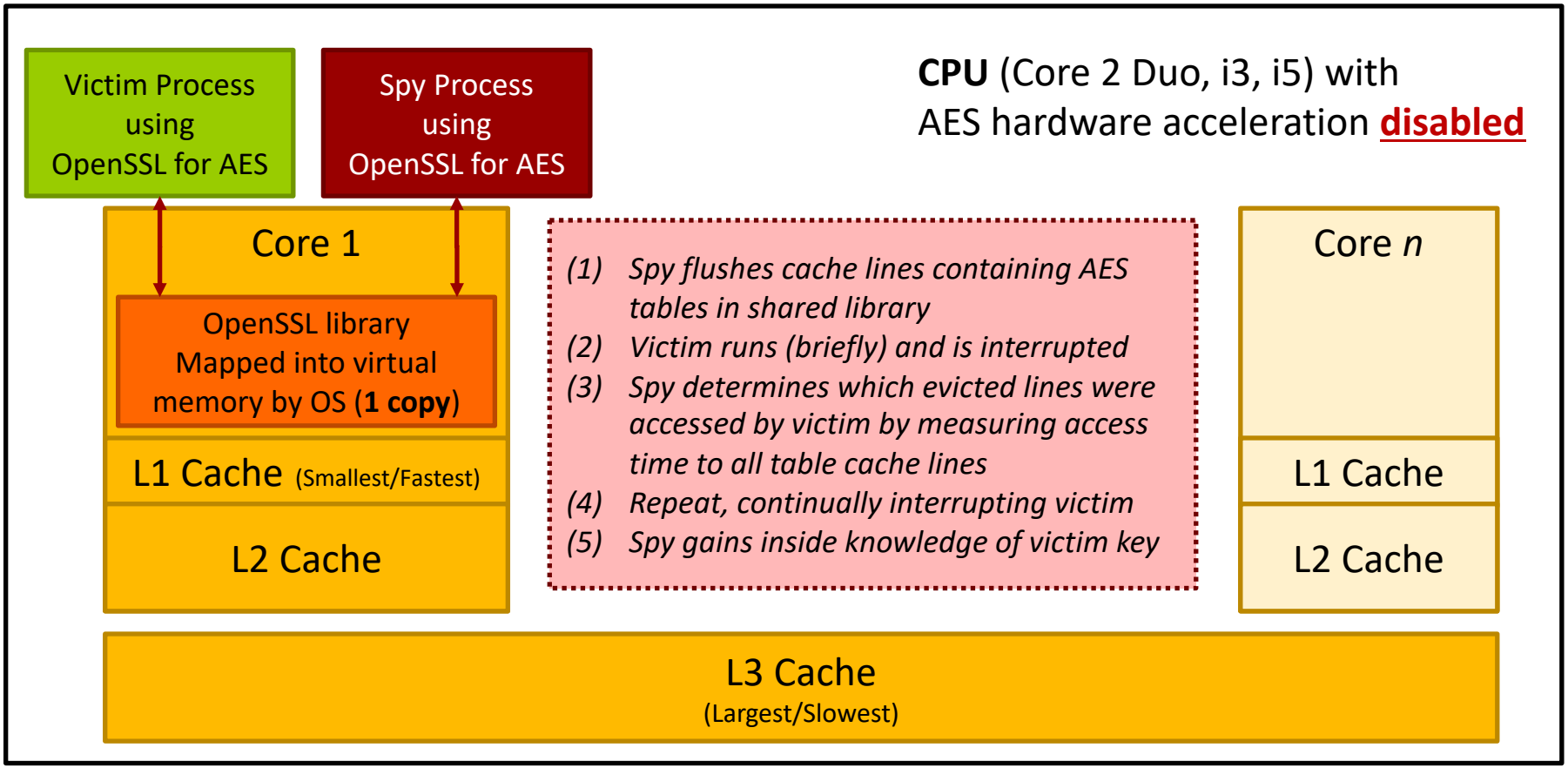
Breaking Crypto - AES

- AES implementation has been “broken”
  - Demonstrated recovery of the full AES key given only about 6 - 7 blocks of plaintext or ciphertext
  - Requires *spy process* running on same machine as AES software. No privilege escalation required.
    - **Side channel attack** via processor cache timing
    - Runs in Python in under a minute!

Research paper

A. C, R. P. Giri and B. Menezes, "Highly Efficient Algorithms for AES Key Retrieval in Cache Access Attacks," *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Saarbrucken, **2016**

Side Channel Attack on AES



Countermeasures

- Use hardware AES acceleration
  - Hardware method does not use memory-resident tables for cipher
  - AES-NI instruction set in x86-64
- Use a cipher and implementation that is resistant to timing attacks
  - Idea: All operations should take constant time
    - No conditional branches with conditions based on secret information.
    - All loop counts are predictable in advance
    - No array lookups with indices based on secret information. *The pattern of memory access must be predictable in advance*
  - *Like NaCL library…*

Have I mentioned that writing robust crypto libraries is hard?