CS 88: Security and Privacy

13: Symmetric Key Cryptography

03-07-2023

slides adapted from Dave Levine, Jonathan Katz, Kevin Du

SWARTHMORE COLLEGE

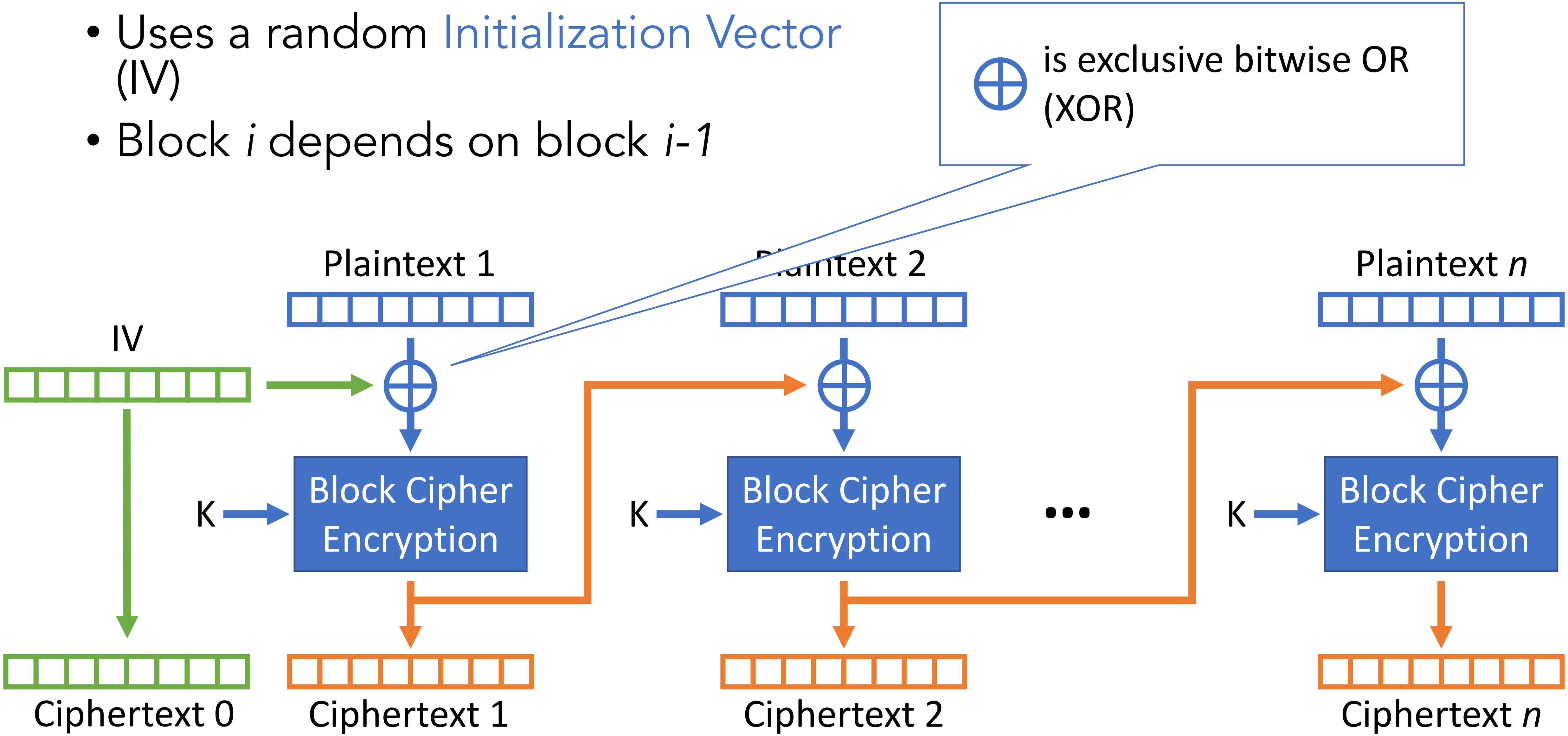
Chosen Ciphertext Attack (CCA – Security)

- In the definition of CCA-security, the attacker can obtain the decryption of any ciphertext of its choice (besides the challenge ciphertext)
  - Is this realistic?

- We show a scenario where:
  - *One bit* about decrypted ciphertexts is leaked
  - The scenario occurs in the real world!
  - It can be exploited to learn the entire plaintext

Cipher Block Chaining (CBC) Mode

- Uses a random Initialization Vector (IV)
- Block *i* depends on block *i-1*

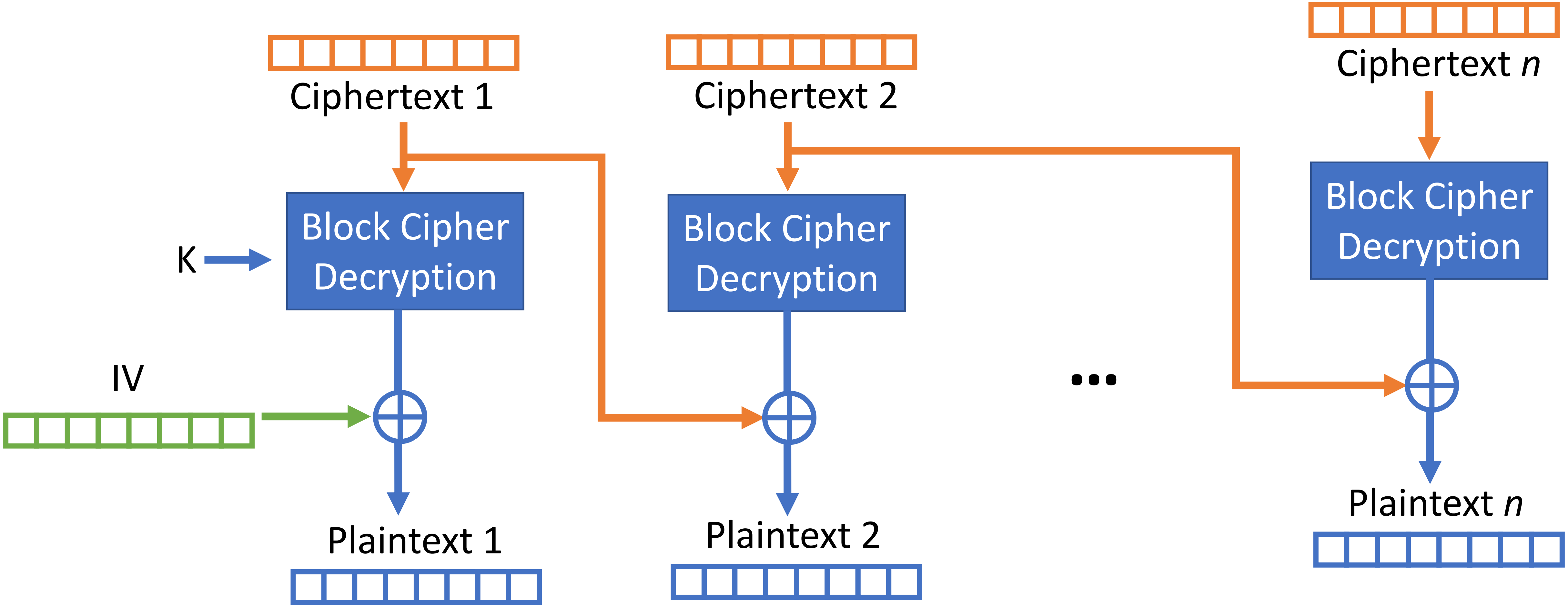


CBC-mode decryption

Decryption

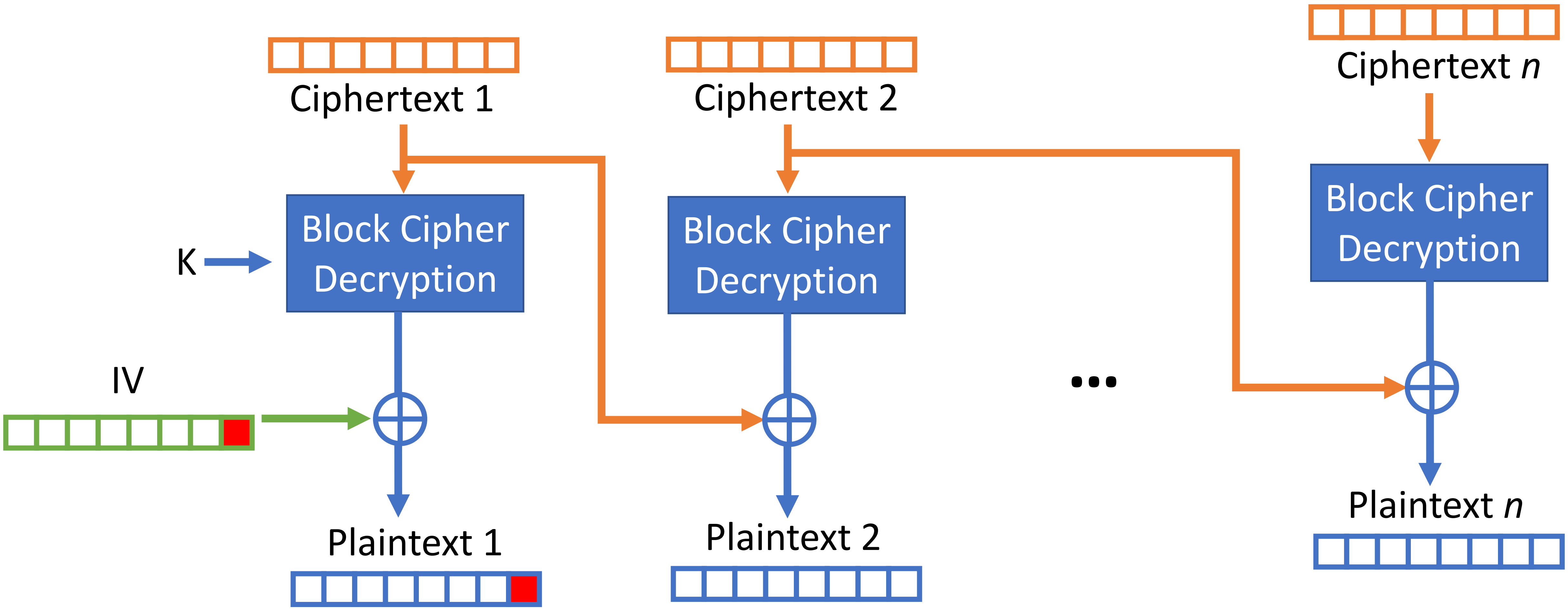
input: ciphertext c,
key k,
initialization vector IV

m[i] = D(k, c[i]) ⊕ c[i−1]

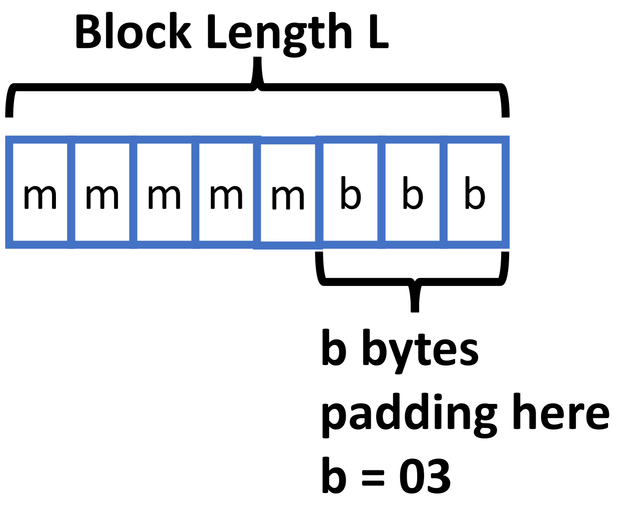


Observation

If an attacker modifies c_{i-1}, this causes a predictable change to m_i



Arbitrary-length messages?



- Message → encoded/padded data → ciphertext
- PKCS #5 encoding:
  - Assume message is an integral number of bytes
  - Let L be the block length (in bytes) of the cipher
  - Let b ≥ 1 be # of bytes that need to be appended to the message to get length a multiple of L
    - 1 ≤ b ≤ L; note b ≠ 0
  - Append b (encoded in 1 byte), b times
    - I.e., if 3 bytes of padding are needed, append 0x030303

Decryption?

- Use CBC-mode decryption to obtain encoded data
- Let’s say the final byte of encoded data has value b
  - If b=0 or b > L, return “error”

AB	01	4F	21	00	7C	04	00

  - If final b bytes of encoded data are not all equal to b, return “error”

AB	01	4F	21	00	7C	03	03

  - Otherwise, strip off final b bytes of the encoded data, and output what remains as the message

Example (L=8)

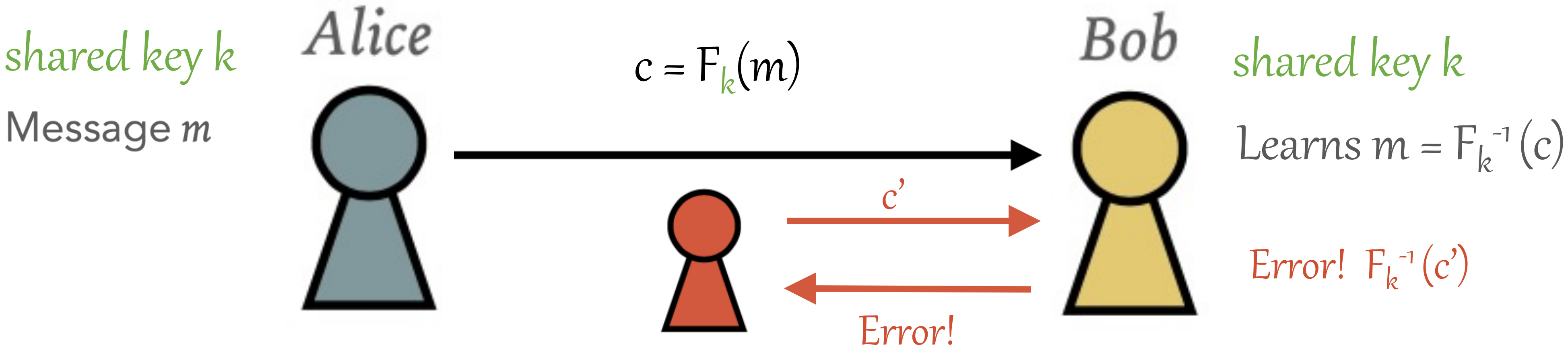
Strip off final b bytes of the padded data, and output what remains as the message

AB	01	4F	21	00	7C	02	02

↓

AB	01	4F	21	00	7C	02	02

Chosen Ciphertext Attack (CCA – Security)



Padding oracle attack!

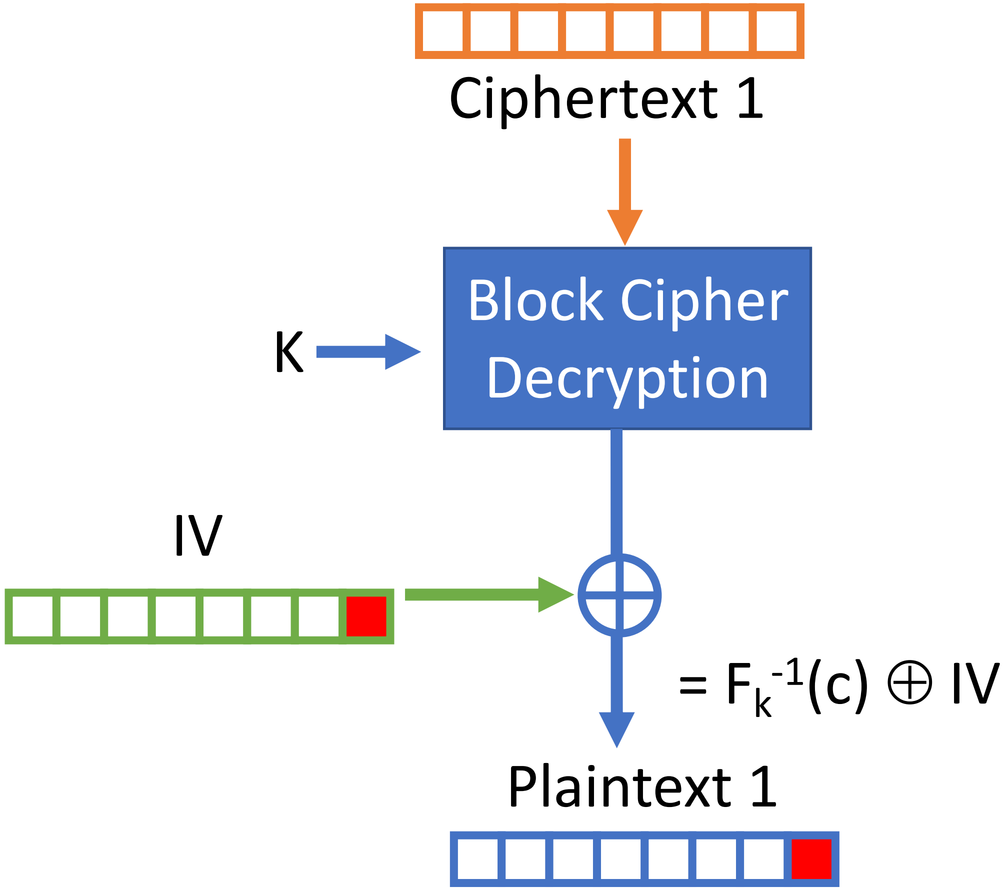
Padding oracles

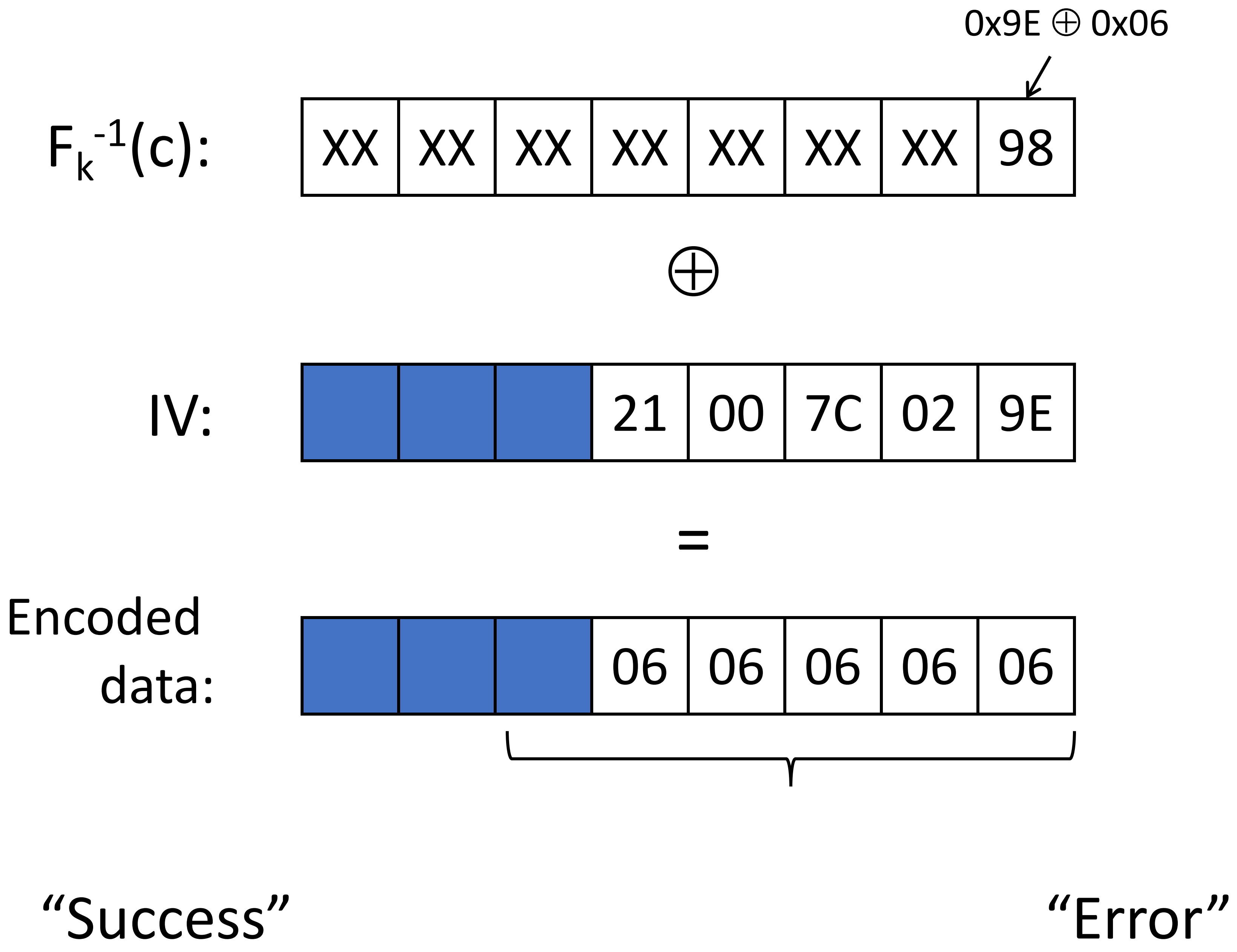
- Padding oracles are frequently present in, e.g., web applications

- Even if an error is not explicitly returned, an attacker might be able to detect differences in timing, behavior, etc.

Main idea of the attack

- Consider a two-block ciphertext IV, c
  - Padded data = $F_k^{-1}(c) \oplus IV$
  - Goal is to learn the encoded data
- Main observation: If an attacker modifies (only) the *i*th byte of IV, this causes a predictable change (only) to the *i*th byte of the padded message.




0x9E ⊕ 0x06
Fk-1(c):
XX XX XX XX XX XX XX 98
⊕
IV:
21 00 7C 02 9E
=
Encoded data:
06 06 06 06 06
"Success"
"Error"

$F_k^{-1}(c)$:

XX	XX	XX	XX	XX	XX	XX	98

$\oplus$

0x02 ⊕ 0x06 ⊕ 0x07
0x98 ⊕ 0x07

IV:

AB	41	4E	20	01	7D	03	9F

=

Encoded data:

XX	07	07	07	07	07	07	07

"Success!"

$XX \oplus 0x41 = 0x07$

$\Rightarrow XX = 0x41 \oplus 0x07$

$\Rightarrow$ plaintext byte $= XX \oplus 0x01 = 0x47$

Attack complexity?

- ≤ L tries to learn the # of padding bytes

- ≤ 2^8 = 256 tries to learn each plaintext byte

CCA-security: a summary

- Chosen-ciphertext attacks are a significant, real-world threat
  - Modern encryption schemes are designed to be CCA-secure

- None of the schemes we have seen so far is CCA-secure!

BLACKBOX #3:
HASH FUNCTIONS

Hash Function Properties

- Very fast to compute
- Takes arbitrarily-sized inputs, returns fixed-sized output
- Pre-image resistant:

  Given H(m), hard to determine m
- Collision resistant

  Given m and H(m), hard to find m′ ≠ m s.t. H(m) = H(m′)

Good hash functions: SHA family (SHA-256, SHA-512, …)

Hash Functions

Cryptographic hash function: maps arbitrary length inputs to a short, fixed-length digest.

Collision-resistance

- Let H: $\{0,1\}^* \rightarrow \{0,1\}^n$ be a hash function
- A *collision* is a pair of distinct inputs x, x' such that H(x) = H(x')

- H is *collision-resistant* if it is infeasible to find a collision in H

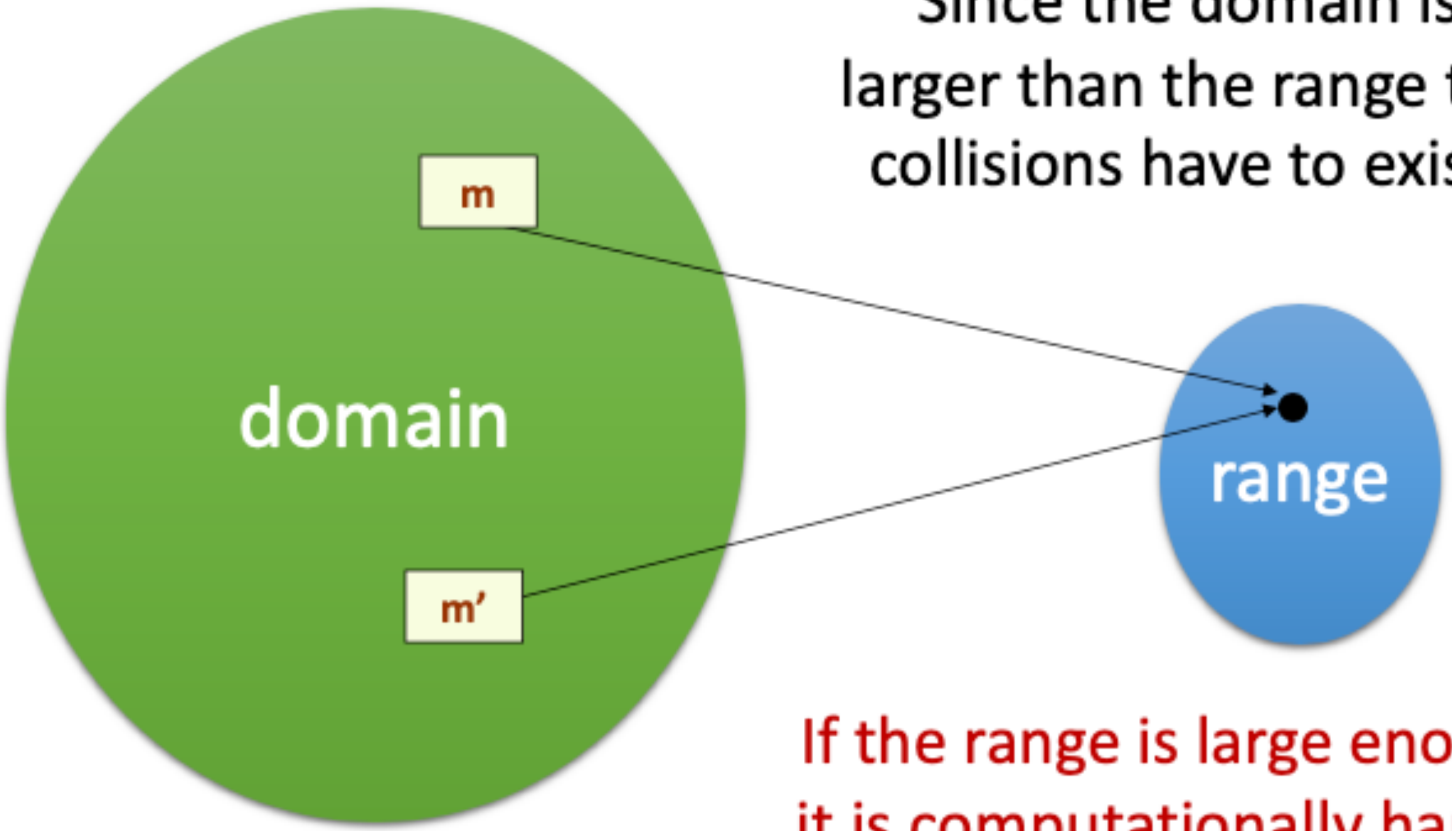
Cryptographic Hash Functions

- Deterministic: H(x) is always the same
- High entropy:
  - md5(‘security’) = e91e6348157868de9dd8b25c81aebfb9
  - md5(‘security1’) = 8632c375e9eba096df51844a5a43ae93
  - md5(‘Security’) = 2fae32629d4ef4fc6341f1751b405e45
- Collision resistant
  - Locating x’ such that H(x) = H(x’) takes a long time
  - Example: 221 tries for md5

Generic hash-function Attacks

- What is the best "generic" collision attack on a hash function
  $H: \{0,1\}^* \rightarrow \{0,1\}n$?
- If we compute $H(x_1)$, ..., $H(x_{2n + 1})$, we are guaranteed to find a collision
- Is it possible to do better?

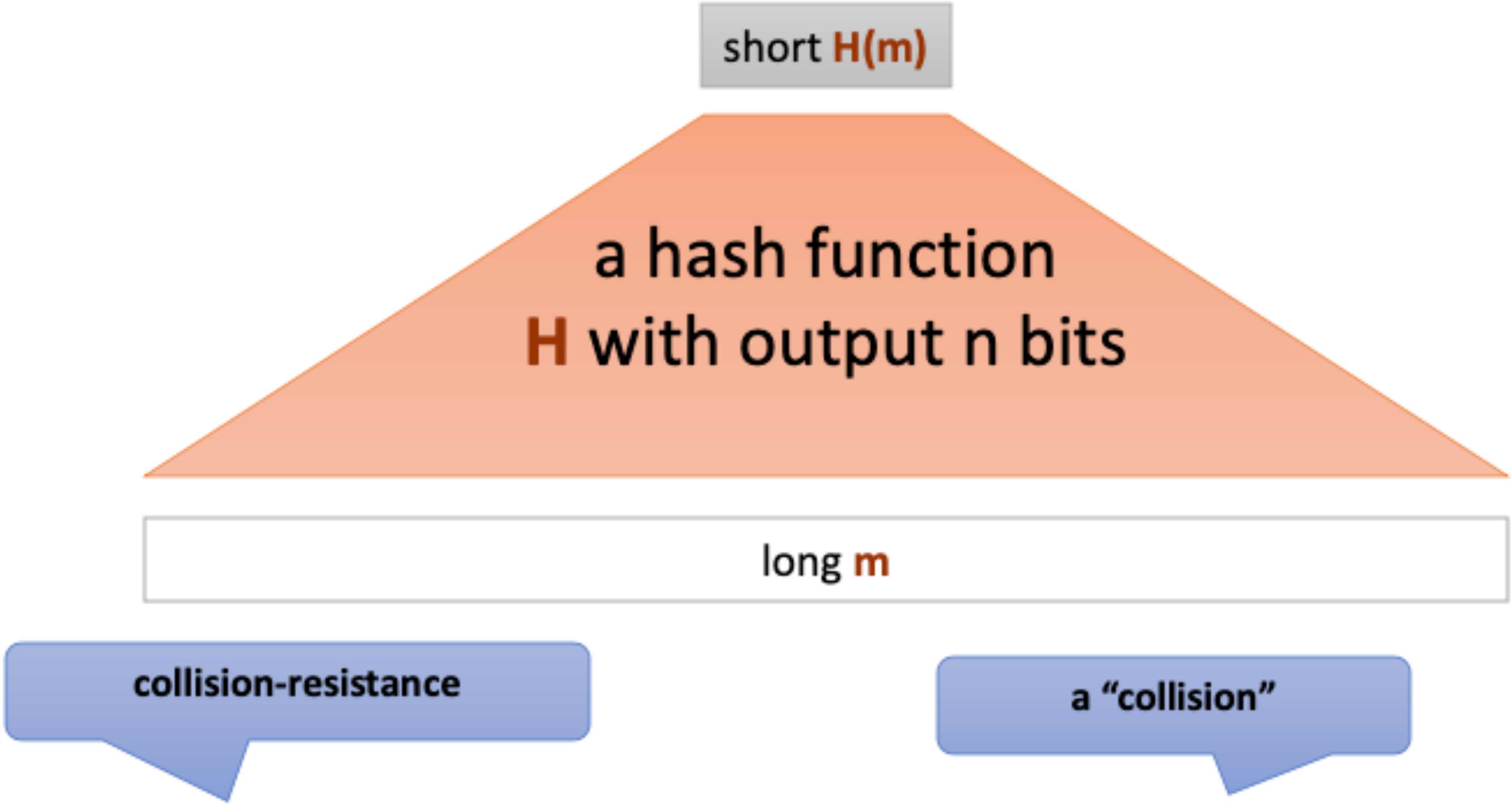
Collisions always exist



Since the domain is larger than the range the collisions have to exist.

If the range is large enough, it is computationally hard to find collisions.

Collision-resistant hash functions



Requirement: it should be hard to find a pair (m,m') such that H(m) =H(m')

"Birthday" attacks

- "Compute $H(x_1), \ldots, H(x_{2n/2})$
  - What is the probability of a collision?
- Related to the so-called birthday paradox
  - How many people are needed to have a 50% chance that some two people share a birthday?

Birthday paradox

- If we choose q elements $y_1, \dots y_q$ at random from {1,…,N}, what is the probability that there exists i and j such that $y_i = y_j$?

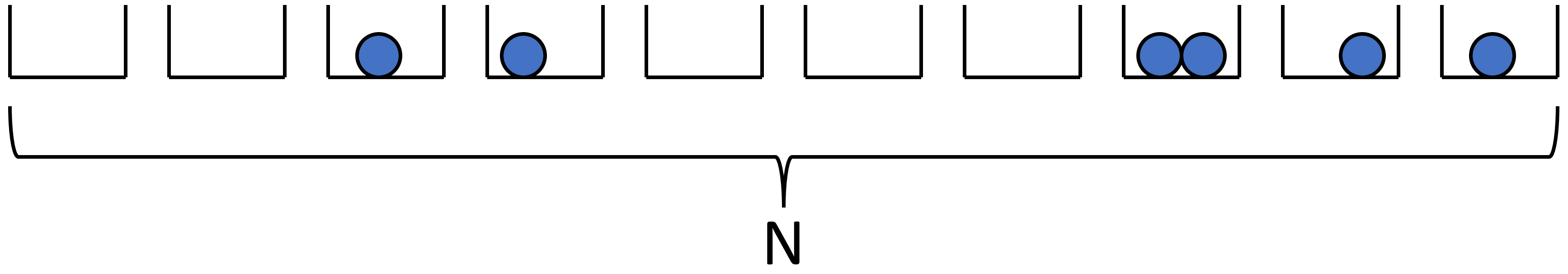
N=365
possible days

- What is the probability that two people have the same birthday?
- When is this probability higher than 0.5?

Bins: days of the year (N=365)
Balls: k people

Bins: values in $\{0,1\}^{\ell}$ ($N = 2^{\ell}$)
Balls: k hash-function computations

How many balls do we need
to have a 50% chance of a collision?



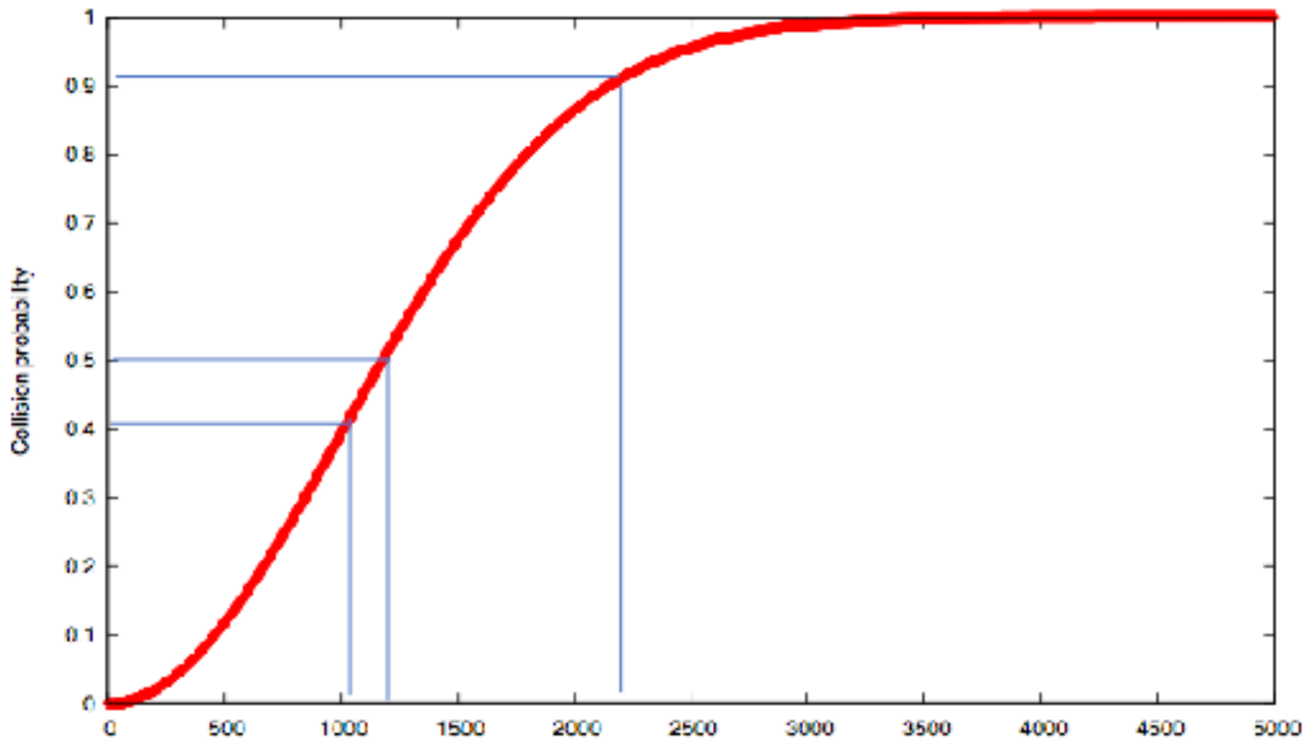
“Birthday” attacks

- **Theorem:** When the number of balls are $O(N^{1/2})$ the probability of a collision is ≈ 50%
  - Birthdays: 23 people suffice!
  - Hash functions: $O(2^{n/2})$ hash-function evaluations

- Need 2n bit output length to get security against attackers running in time 2^n
  - Note: *twice as long* as symmetric keys (e.g., block-cipher keys or PRG seeds) for the same security

Collision probability

$N = 10^6$



samples
n

- If $q = \Theta(\sqrt{N})$ items, then probability of collision is approx. ½
- Birthday paradox
  - N = 365, q = 23
- Hash functions
  - $N = 2^{256}, q = 2^{128}$
- Implies n/2 level of security for n-bit hash function in best case

“Birthday bound”

- The birthday bound comes up in many other cryptographic contexts

- Example: IV reuse in CTR-mode encryption
  - If k messages are encrypted, what are the chances that some IV is used twice?
  - Note: this is much higher than the probability that a *specific* IV is used again

History of hash functions

H is a collision-resistant hash function if it is “practically impossible to find collisions in H”.

- 1991: MD5
- 1995: SHA1
- 2001: SHA2 -- SHA-256 and SHA-512
- 2004: Team of Chinese researchers found collisions in MD5
- 2007: NIST competition for new SHA3 standard
- 2012: Winner of SHA3 is Keccak

The Future: SHA3

- 2007: NIST opens competition for new hash functions
- 2008: Submission deadline, 64 entries, 51 make the cut
- 2009: 14 candidates move to round 2
- 2010: 5 candidates move to round 3
- 2011: final round of public comments
- 2012: NIST selects keccak (pronounced “catch-ack”) as SHA3
- Created by Guido Bertoni, Joan Daemen, Gilles Van Assche, Michaël Peeters