Inse 6110

Jeremy Clark

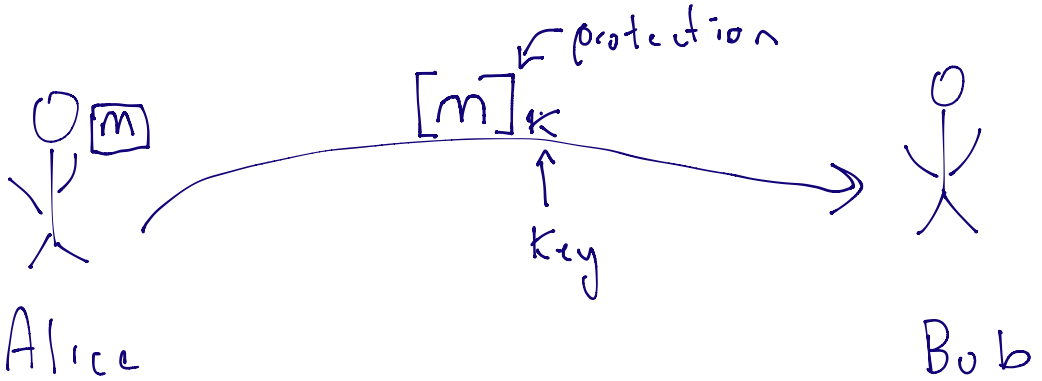
Fall 2017

Cryptography and its relation to security: (CIA)

* Confidentiality
* Integrity
* Availability

Confidentiality

* Crypto plays a large role
* e.g., encryption



Two types of encryption:

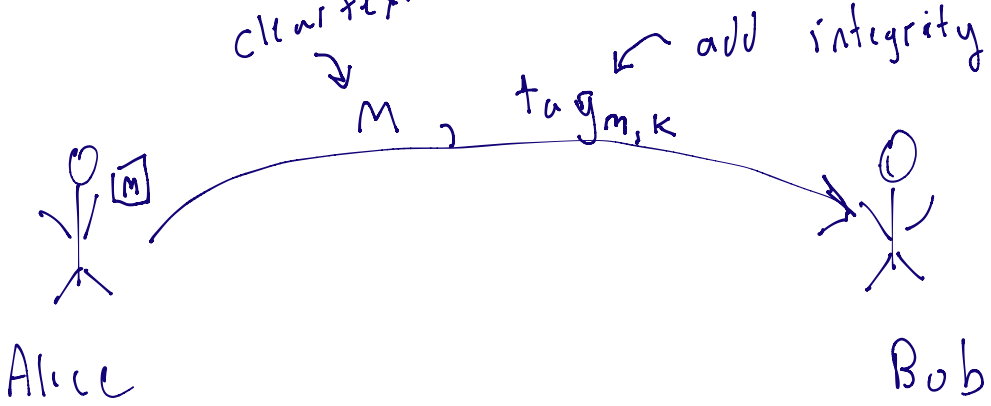
* Symmetric key → Alice & Bob use same key.

slow → * Asymmetric Key (Public key)

↳ Alice uses Bob's public key to encrypt, Bob uses his private key to decrypt.

Integrity

* Assurance that the message we sent/stored is identicle to the one received/retrieved



→ Protection against random errors
  ↳ reliability.
    ↳ error correcting code

→ We want: protection against an intelligent adversary.
  ↳ use a keyed function

↳ Symmetric case: MACs (message authentication codes)

slow ↳ Asymmetric case: digital signatures

Availability

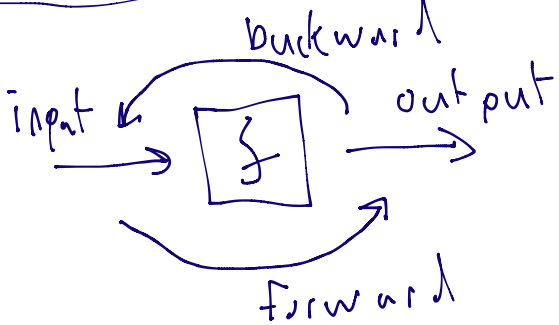
↳ No crypto here

Summary

	Keys	C	I	A	BB	Forward	Backwards
Hash	0				×	Anyone	No-one
PRG	0				×	Anyone	No-one
Extractors	0				×	Anyone	No-one
Encryption	1	×				key	key
MACs	1		×			key	—
key Exchange	2				×	—	—
Encryption	2	×				Anyone (encrypt)	Key (decrypt)
Signatures	2		×			Key (sign)	Anyone (Verify)

Building Block

Forward / Backward Functions.



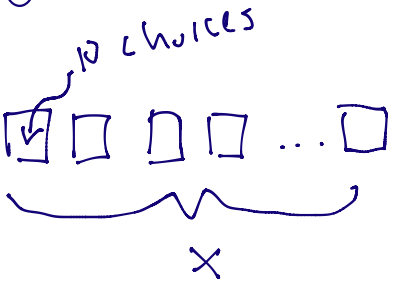
Exhaustive Search / Brute-force

(1) Assume you have a 4 digit PIN. How many guesses does it take to find the PIN?

* There are 10,000 unique pins (0000 → 9999)
* Probality to guess it: 1/10,000

(2) Assume you have an X digit PIN:

How many PINS: 10^x



Probability?

$Pr[guess] = 1/10^x = 10^{-x}$

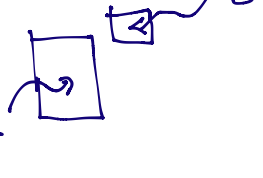
(3) Assume you have an X bit PIN:

How many: 2^x

Probability: 2^{-x} $(1/2^x)$

(4) Assume an 8 character password using lower-case, upper-case, numbers, special characters (33)

How many?:



$95^8 = 6,634,204,312,890,625$

$\simeq 2^{53}$

2^{53} is feasible to brute-force:

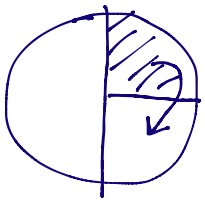
* Hardware AntMiner ($2500)
  ↳ 8 minutes.

Computational infeasibility

2^{53} → feasible
↳ corresponds to a 53-bit secret

2^{54} → Twice as hard

2^{70} ⇝ threshold between infeasible and feasible



2^{112} ⇝ NIST standard for infeasibility (with safety

margin)

↳ 112-bit key

↳ can't brute-force it

↳ $Pr[\text{guess}] = 2^{-112}$

$= 1/2^{112}$

Aside

Presentation of functions for this class:

(1) Blackbox description: Properties

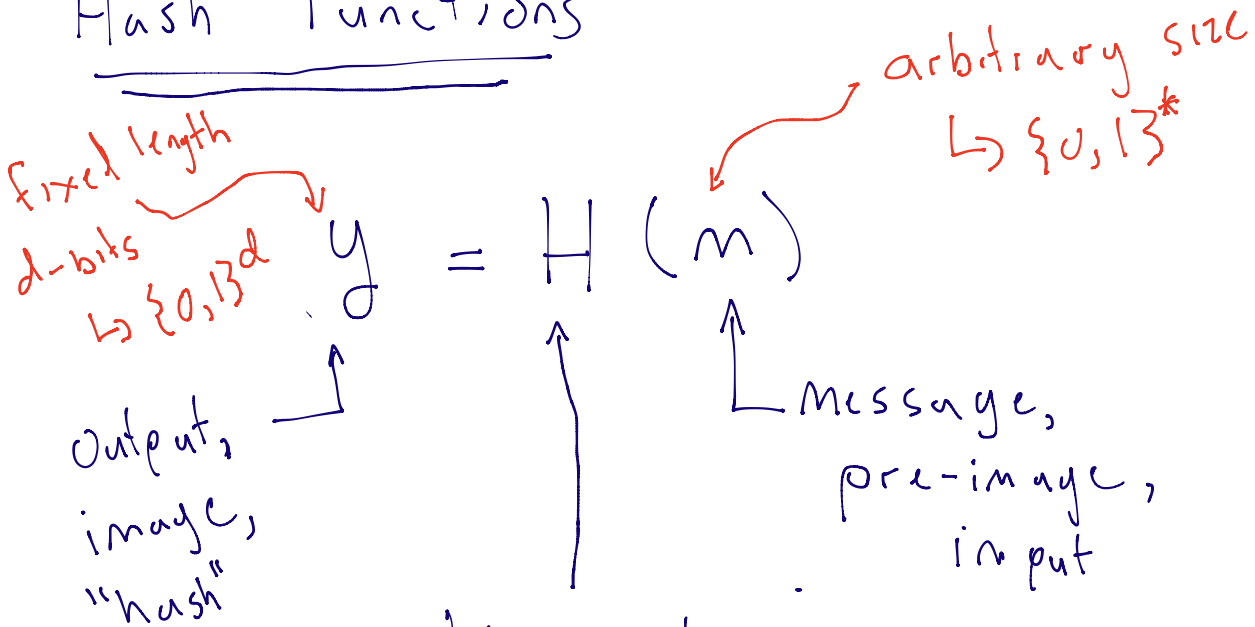
(2) Greybox: some major structure

(3) Whitebox: full view, end-to-

end

* Not tested
* Obsolete versions

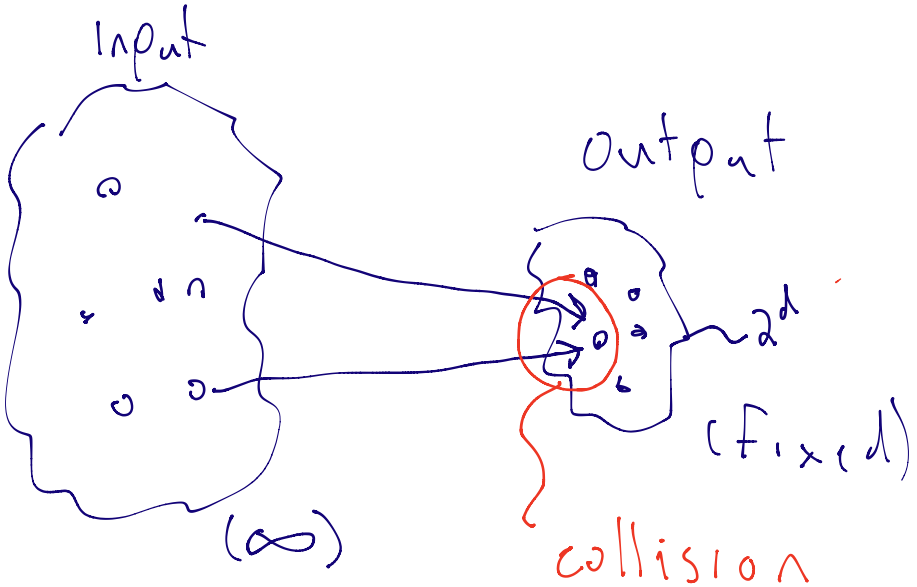
Hash Functions



What does a hash function do?

* Creates a unique "fingerprint" for the input message/data.

* input can be any bitstring
  ↳ files, images, movies, etc.



Known hashes: MD4, MD5, SHA-1, SHA-2, SHA-3
X X X ✓ ✓

"Secure" hash functions has 3 properties:

(1) Pre-image resistance

(2) Collision resistance

(3) Weak collision resistance

Pre-image Resistance (PR)

Given y, where y is d-bits long, it is infeasible to find any x s.t. $H(x) = y$.

↓ collisions exist

↳ intuition: one-way or non-invertible.

How hard is it to break PR?

* Has something to do with d

Extreme case: $d = 1$

↳ You will find <u>an</u> x that produces the same output y in ~ 2 attempts.

General case: d

Given output y, what is the probability that an arbitrary x will give $H(x) = y$?

$$Pr[H(x) = y] = 1/2^d = 2^{-d}$$

↳ d-bits of security

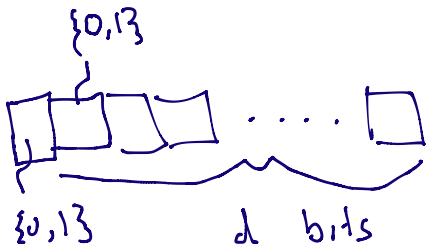
↳ 2^d work to break PR

How big should d be for NIST level in terms of PR?

112 bits

Note: this analysis assumes the hash function maps inputs to outputs with uniform randomness.

Hash output:



↳ $Pr(bit_0 = 0)$ 50%

$Pr(bit_0 = 1)$ 50%

$$Pr(\text{output} \overset{\text{matches}}{=} \text{given output}) = 1/2^d$$

↑ for a new input

Attack to break PR is to choose inputs (N in total) until the output matches y (given output of the hash):

$$E[\#\text{ of pre-images}] = (\text{number of chances}) \cdot (\text{Probability of one chance being favorable})$$

expect a pre-image

$1 = N \cdot 1/2^d$

$N = 2^d$

↑ infeasible?

↳ $N \geq 2^{112}$

$2^d \geq 2^{112}$

$d \geq 112$ bits.

∴ $d \geq 112$ bits for H to be PR.