CNT 4406, Spring 2024

Blockcipher

Viet Tung Hoang

Some slides are based on material from Prof. Mihir Bellare (UCSD) and Prof. Stefano Tessaro (UW)

Agenda

1. **Blockciphers**

2. Birthday Attack

3. App: TCP Sequence Number

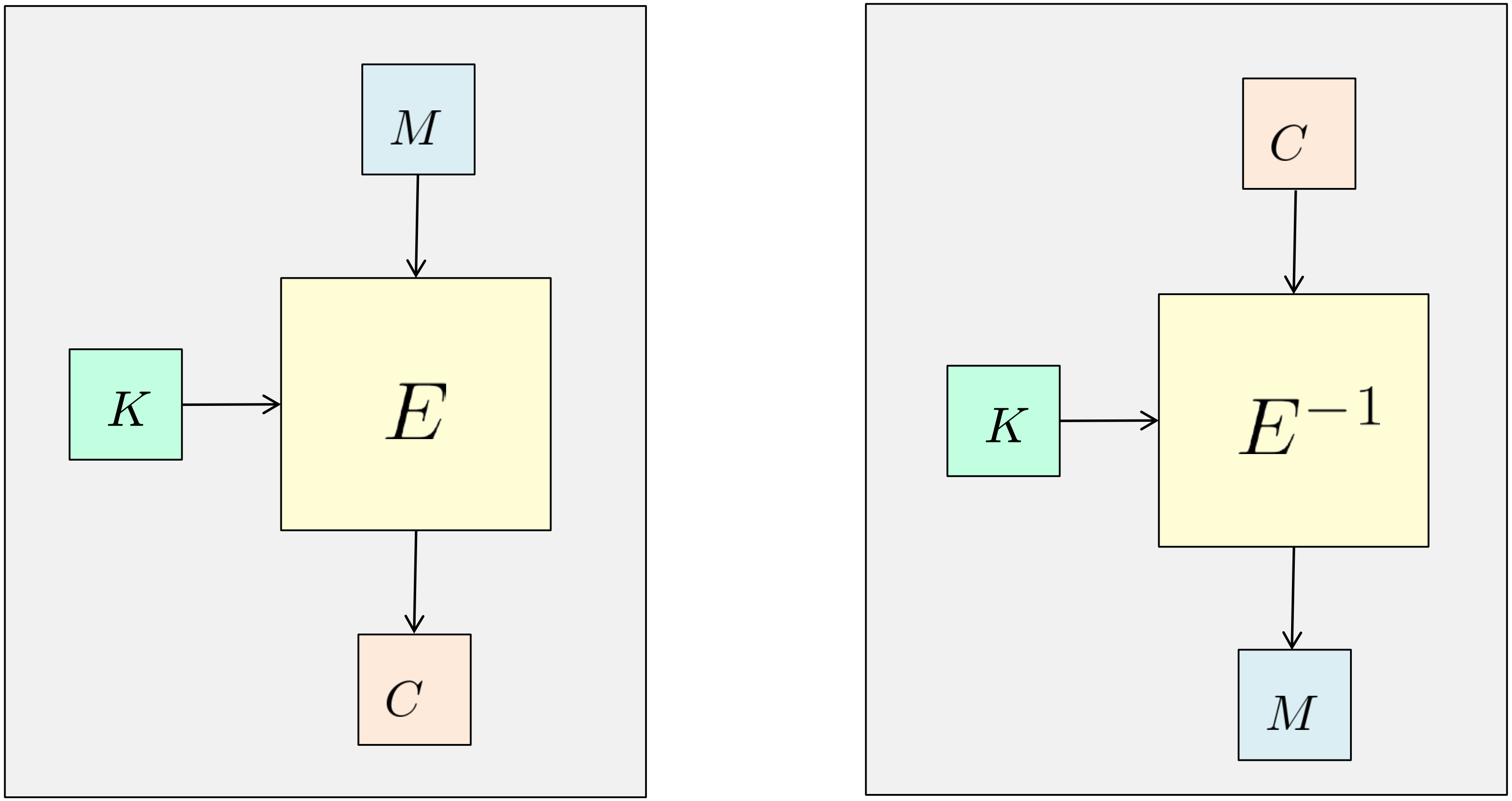
4. App: One-time Password

5. App: Challenge-Response Protocol

Blockcipher

efficiently invertible given the key

$$E : \underbrace{\{0,1\}^k}_{\text{Key space}} \times \underbrace{\{0,1\}^n}_{\text{Domain}} \to \{0,1\}^n$$



Blockcipher Usage

$C_1 \leftarrow E_K(M_1)$ … $C_q \leftarrow E_K(M_q)$

K K

Random key K is known to both parties, but not given to adversary A

Real-world Blockciphers

NIST Special Publication 800-67
Version 1.1

NIST
National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

Recommendation for the Triple Data Encryption Alg
(TDEA) Block Ciphe
Revised 19 May 2008

William C. Barker

3DES, deprecated since 2017 but still in legacy software

$k = 168$, $n = 64$

FIPS 197

Federal Information Processing Standards Publication

Advanced Encryption Standard (AES)

Category: Computer Security **Subcategory:**

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

AES, national standard

$k \in \{128, 192, 256\}, n = 128$

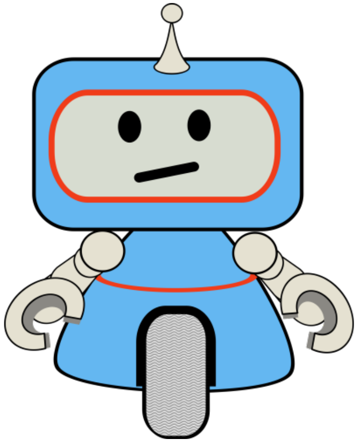
Defining Security for Blockcipher

Possible Properties	**Necessary**	**Sufficient**
Hard to recover the key	Yes	No
Hard to find M given $C \leftarrow E_K(M)$	Yes	No
…		

Want: a single "master" property that is sufficient to ensure security of common usage of blockcipher.

An Analogy: Turing Test

What does it mean for a machine to be “intelligent”?



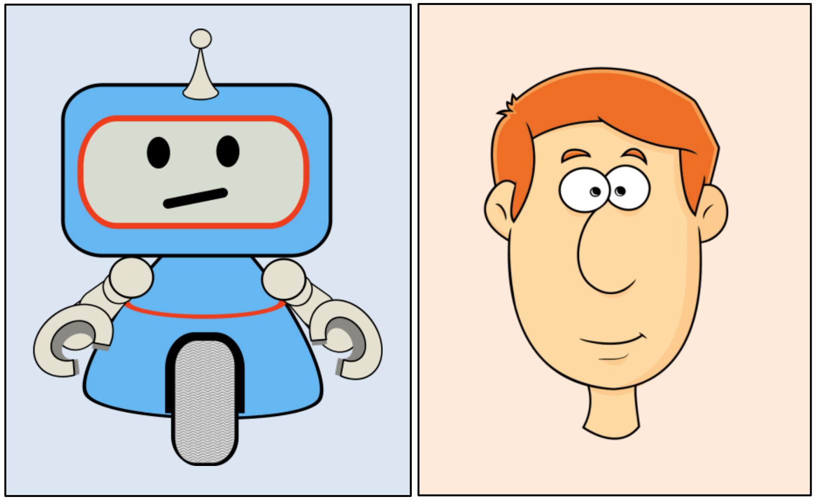
Possible Answers
It can be happy
It recognizes pictures
…

But no such list is satisfactory

An Analogy: Turing Test

A ← interaction →

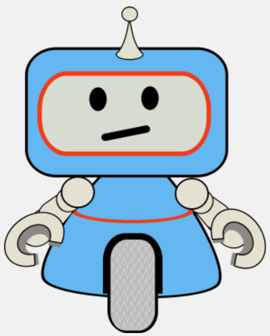
b



1 **0**

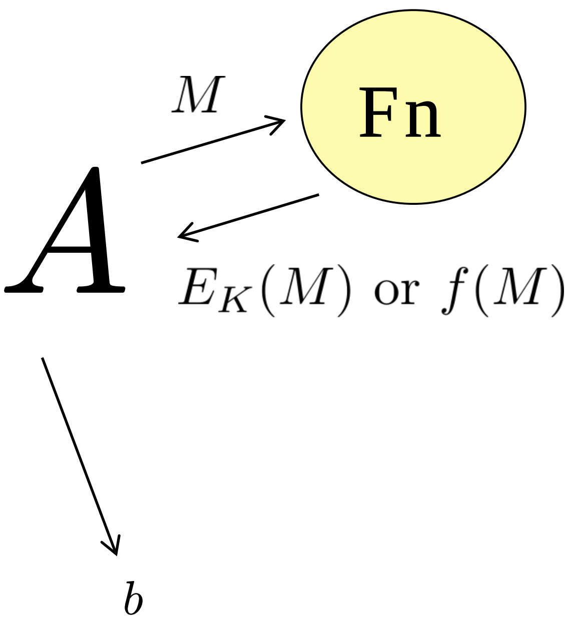
Man (0) or Machine (1)?

Real versus Ideal

Notion	Real object	Ideal object
Intelligence		
PRF	E_K	Random function

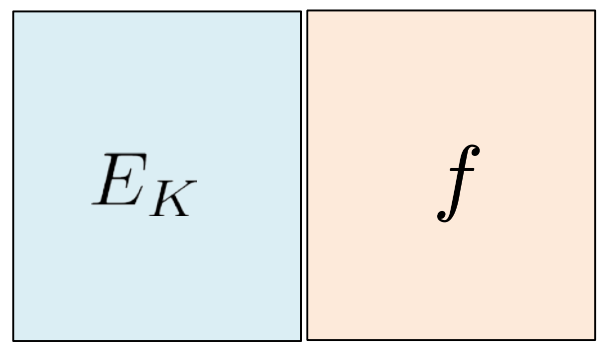
Informal View of PRF Security

$$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$



Sample random $f : \{0,1\}^n \to \{0,1\}^n$

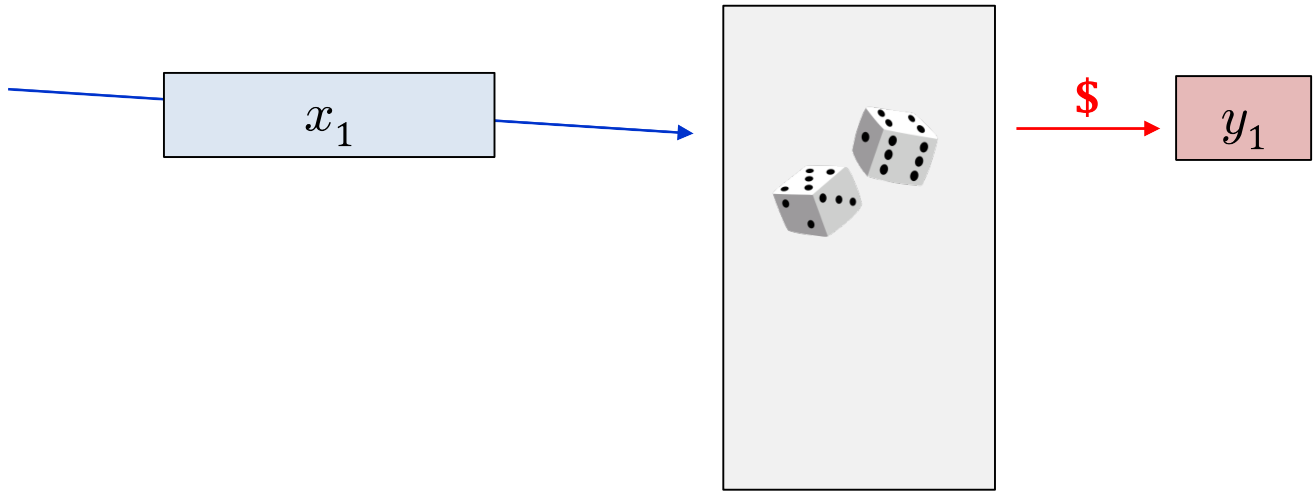
$K \xleftarrow{\$} \mathcal{K}$



Adversary doesn't know K or f

Defining Random Function: Lazy Sampling

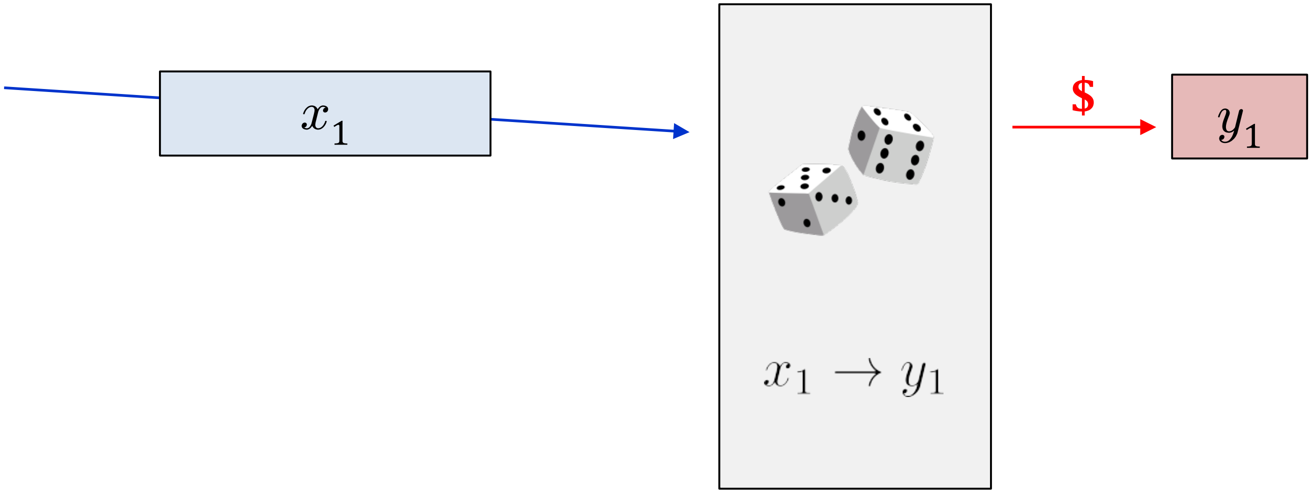
Want: a random function $f : \{0,1\}^n \to \{0,1\}^m$



Pick a fresh random answer for a new query, and remember the answer

Defining Random Function: Lazy Sampling

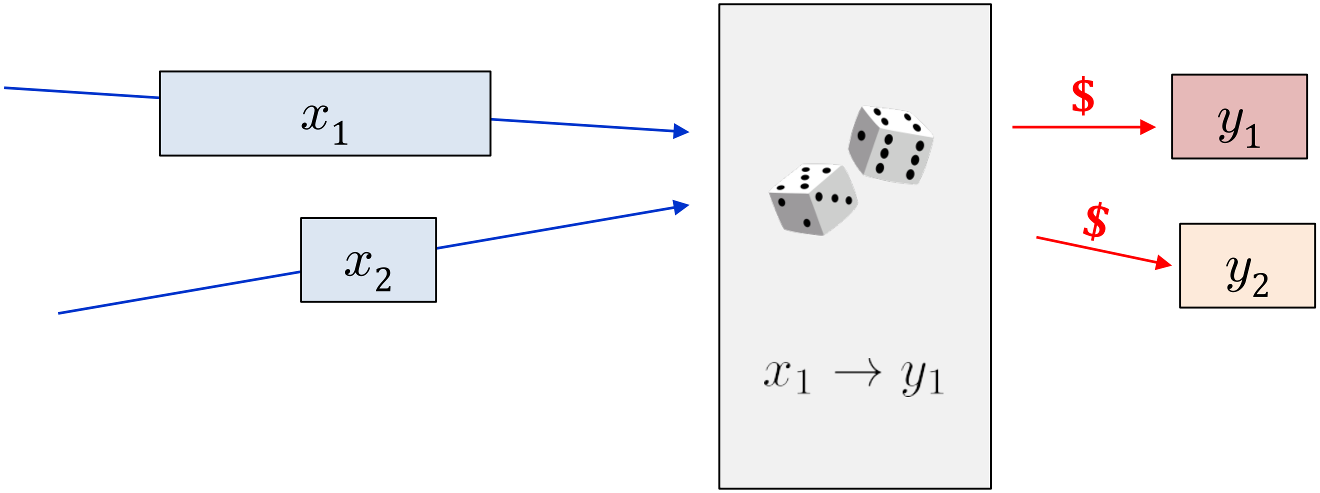
Want: a random function $f : \{0,1\}^n \rightarrow \{0,1\}^m$



Pick a fresh random answer for a new query, and remember the answer

Defining Random Function: Lazy Sampling

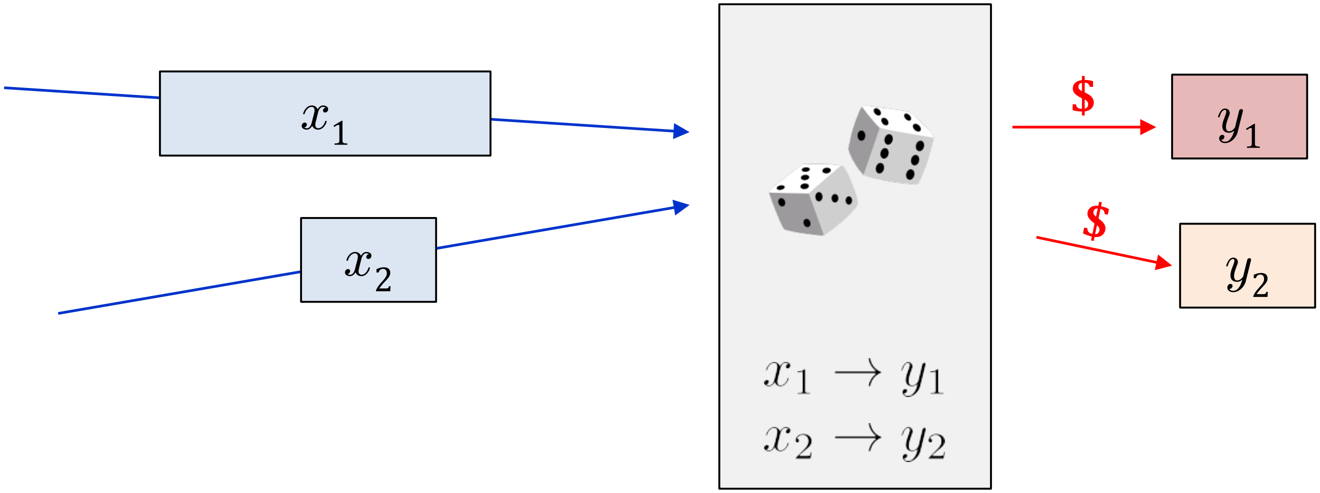
Want: a random function $f : \{0,1\}^n \rightarrow \{0,1\}^m$



Pick a fresh random answer for a new query, and remember the answer

Defining Random Function: Lazy Sampling

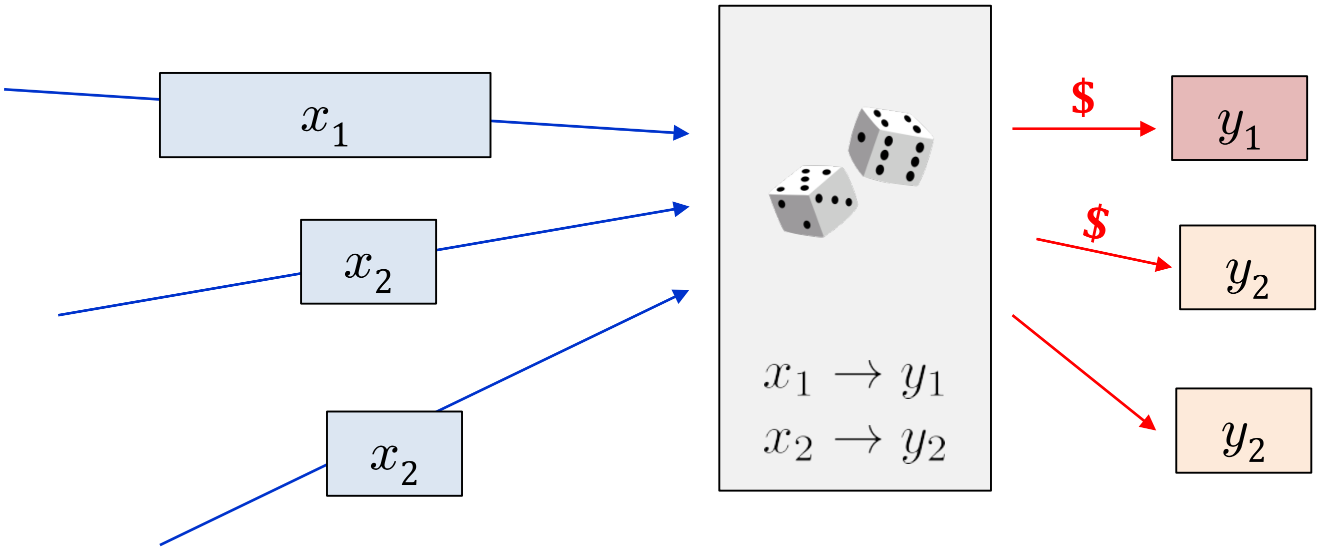
Want: a random function $f : \{0,1\}^n \to \{0,1\}^m$



Pick a fresh random answer for a new query, and remember the answer

Reuse **Prior Answer** for Old Query

Want: a random function $f : \{0,1\}^n \to \{0,1\}^m$



Putting Things in Code

Game Real_E

procedure Initialize()

$K \leftarrow\$ \mathcal{K}$

procedure Fn(M)

return $E_K(M)$

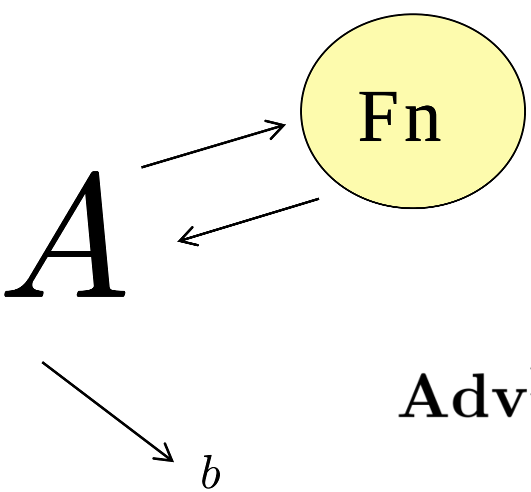
Game Rand_E

string array $T = \{\}$ // Global variable

procedure Fn(M)

If $T[M] = \perp$ then $T[M] \leftarrow\$ \{0,1\}^n$

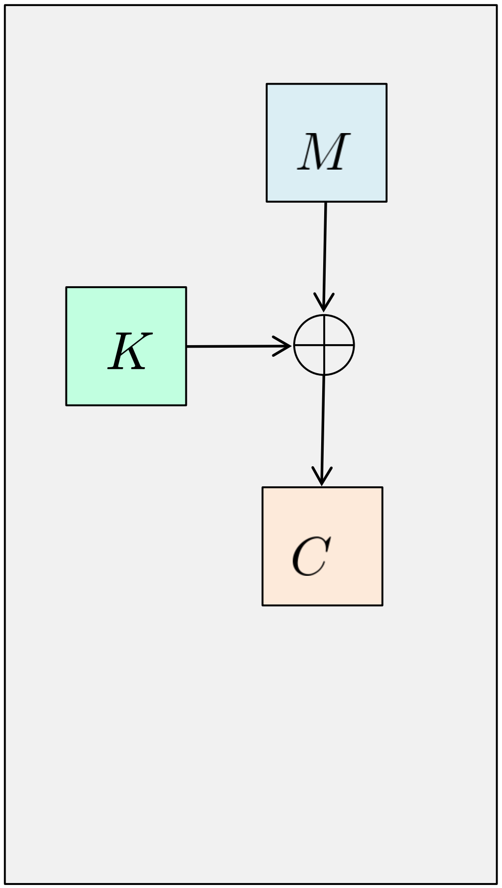
return $T[M]$



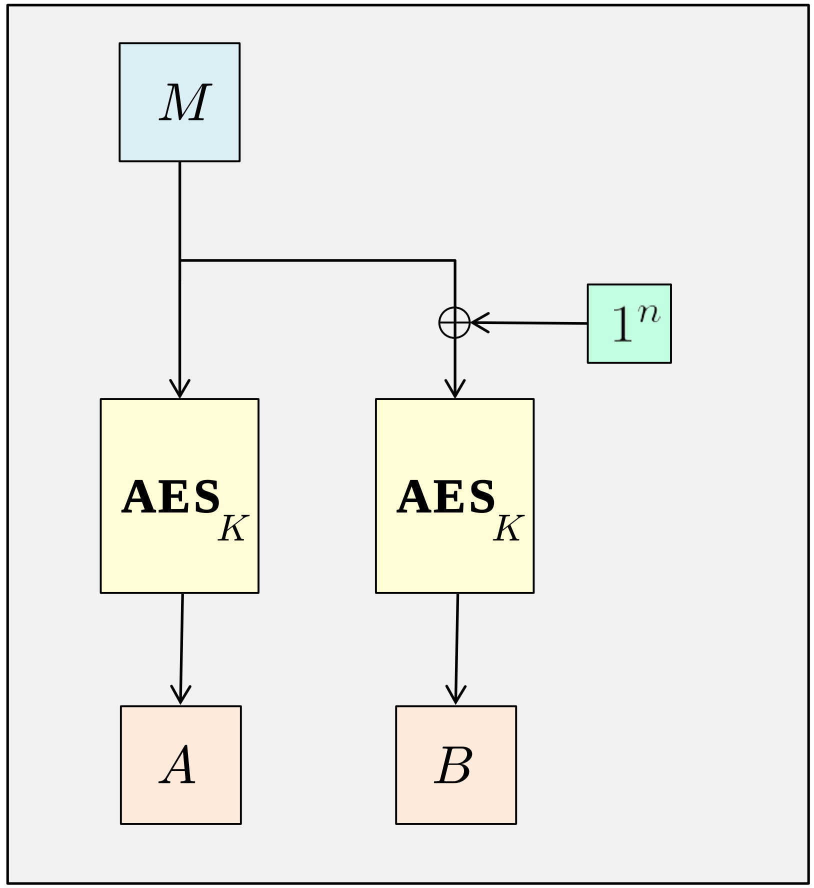
$$\mathbf{Adv}^{\text{prf}}_E(A) = \Pr[\text{Real}^A_E \Rightarrow 1] - \Pr[\text{Rand}^A_E \Rightarrow 1]$$

Exercise: PRF Attacks

$E_K(M) = M \oplus K$



$E_K(M) = \mathbf{AES}_K(M) \| \mathbf{AES}_K(\overline{M})$



Agenda

1. Blockciphers

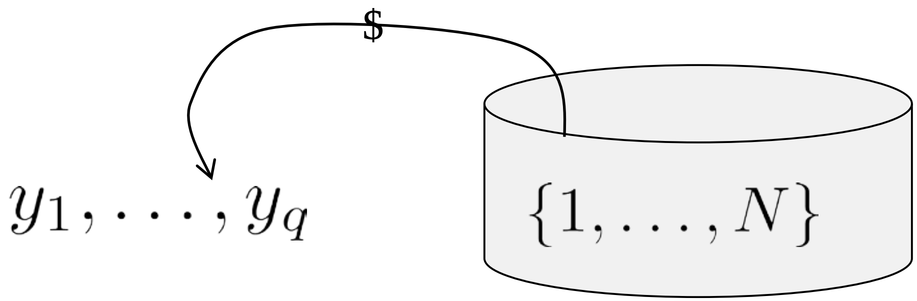
2. Birthday Attack

3. App: TCP Sequence Number

4. App: One-time Password

5. App: Challenge-Response Protocol

Birthday Problem

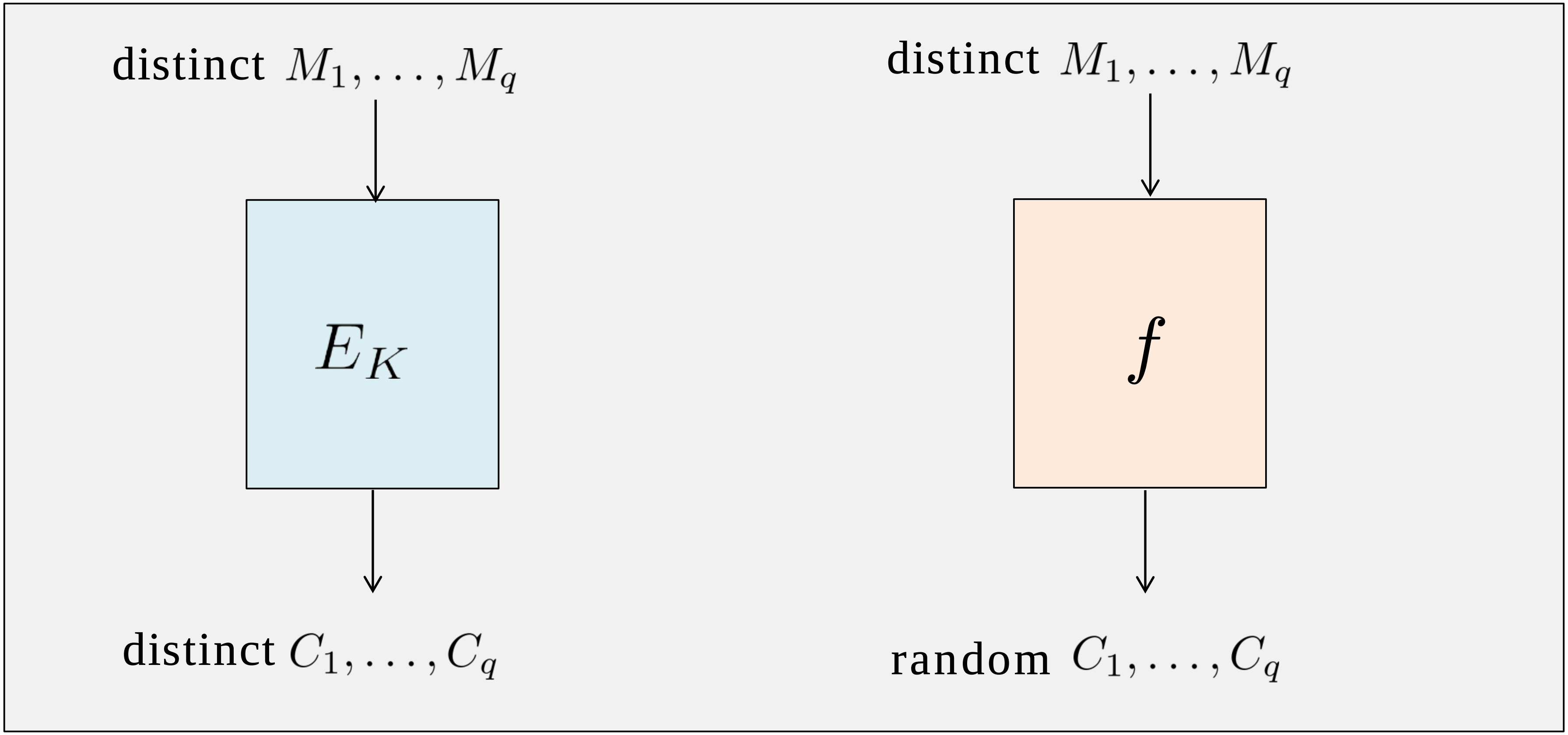


$$C(N,q) = \Pr[y_1, \ldots, y_q \text{ not distinct}]$$

Fact: For $q \leq \sqrt{2N}$,

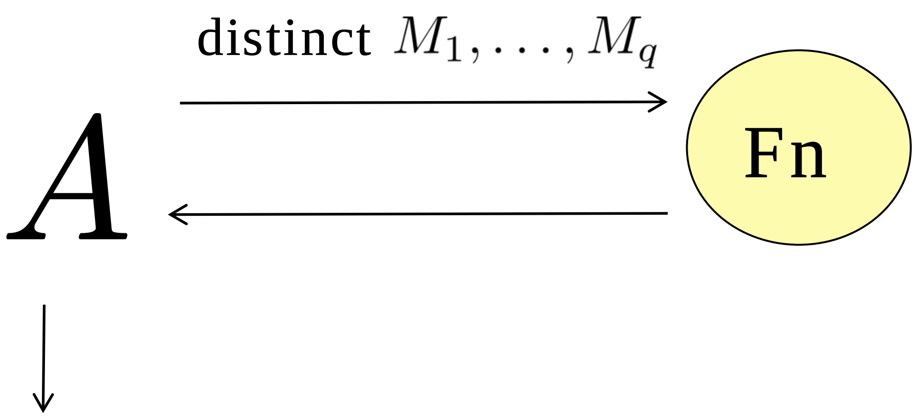
$$\frac{q(q-1)}{4N} \leq C(N,q) \leq \frac{q(q-1)}{2N}$$

Birthday Attack on PRF Security



Birthday Attack on PRF Security

$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$



Output 1 if $C_1, \ldots, C_q$ are distinct

$$\mathbf{Adv}_E^{\mathrm{prf}}(A) = C(2^n, q) \approx \frac{q^2}{2^n}$$

Need $2^{n/2}$ queries to break PRF security

Blockcipher	n	$2^{n/2}$	**Status**
3DES	64	2^{32}	Insecure
AES	128	2^{64}	Secure

Does It Matter In Practice?

Sweet32: Birthday Attacks on 64-bit Blockciphers in TLS and OpenVPN [Bhargavan, Leurent 16]

HTTPS encryption via 3DES

Recover cookie after capturing 785GB

Agenda

1. Blockciphers
2. Birthday Attack
3. **App: TCP Sequence Number**
4. App: One-time Password
5. App: Challenge-Response Protocol

TCP Recap

Handshake

Session data

Termination

TCP Handshake

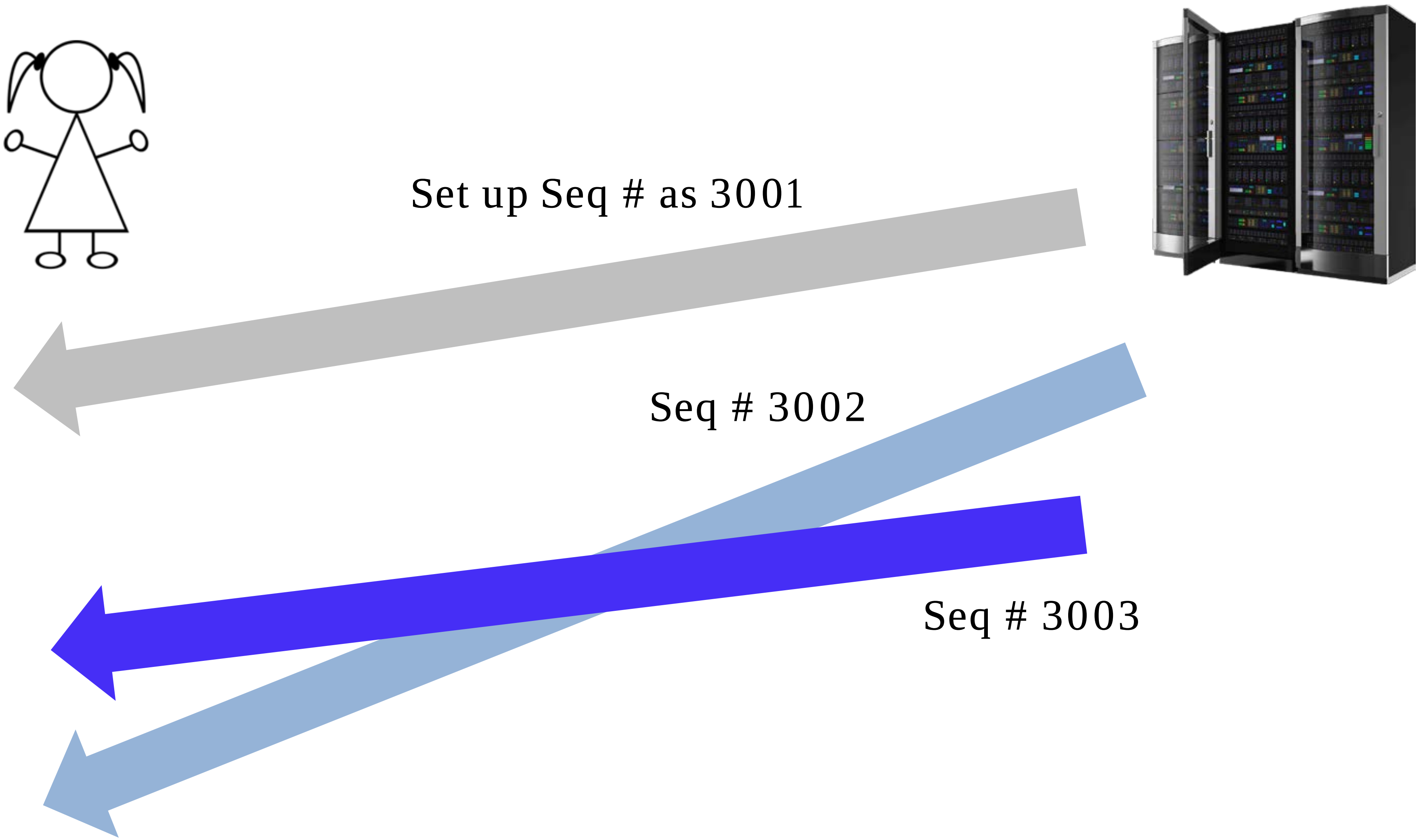
SYN: Alice's Seq # is X

ACK: Your Seq # is X

SYN: Server's Seq # is Y

ACK: Your Seq # is Y

The Use of Sequence Numbers



Can still reorder out-of-order packets

Security Issue: Denial-of-Service Attack

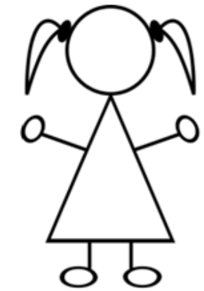
Session data

Need to guess seq #, but it's easy

in early TCP implementations

Reset, source=Alice

Security Issue: Connection Spoofing



SYN: Alice's Seq # is X,

source IP = Alice

ACK: Your Seq # is X

SYN: Server's Seq # is Y

ACK: Your Seq # is Y

Now execute command on Alice's behalf

First Attempt: Random Sequence Number

Backward Compatibility Issue

End with seq # X

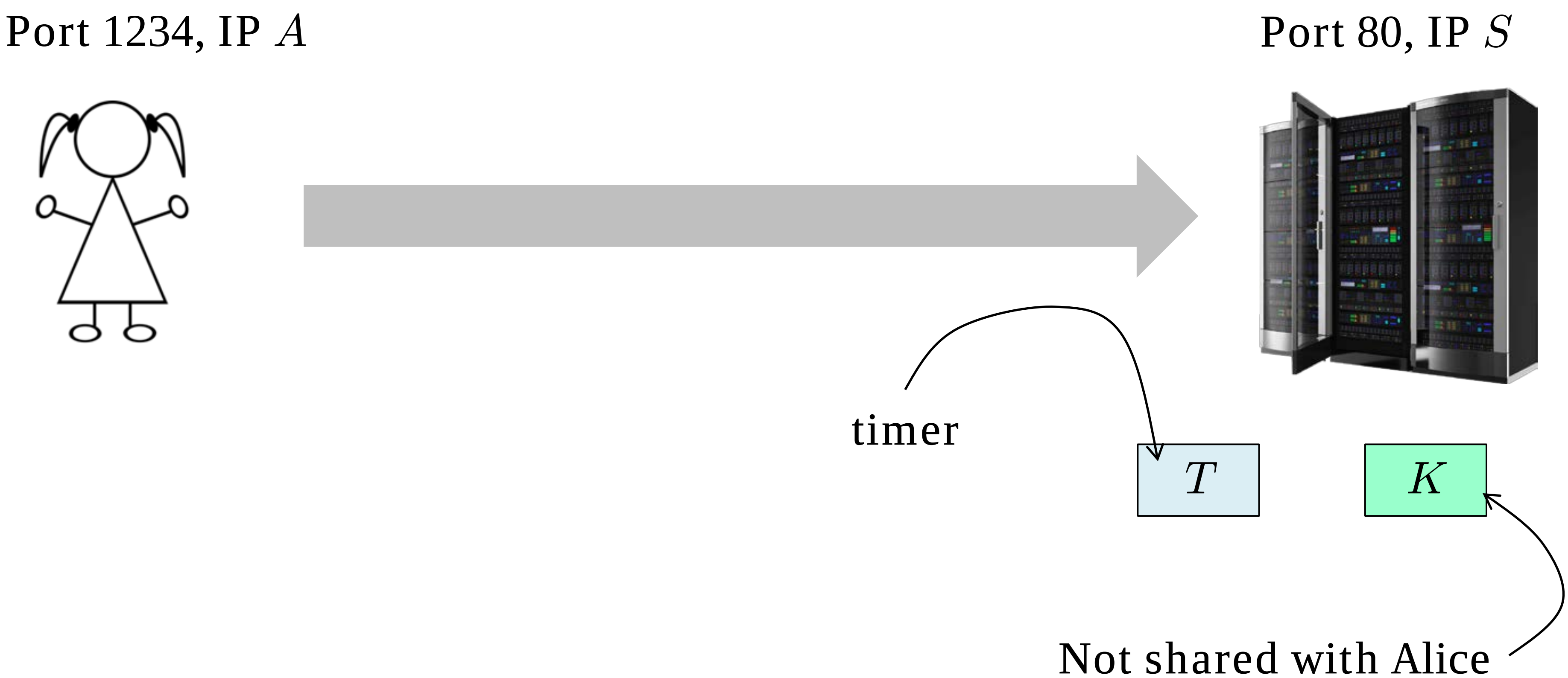
Port 1324 → Port 80

Start with seq # Y

Port 1324 → Port 80

Requirement: If two connections of same IP addresses and ports are within a small window, must have $X < Y$ to avoid interference from delayed packets

Generating TCP Sequence Numbers with PRF

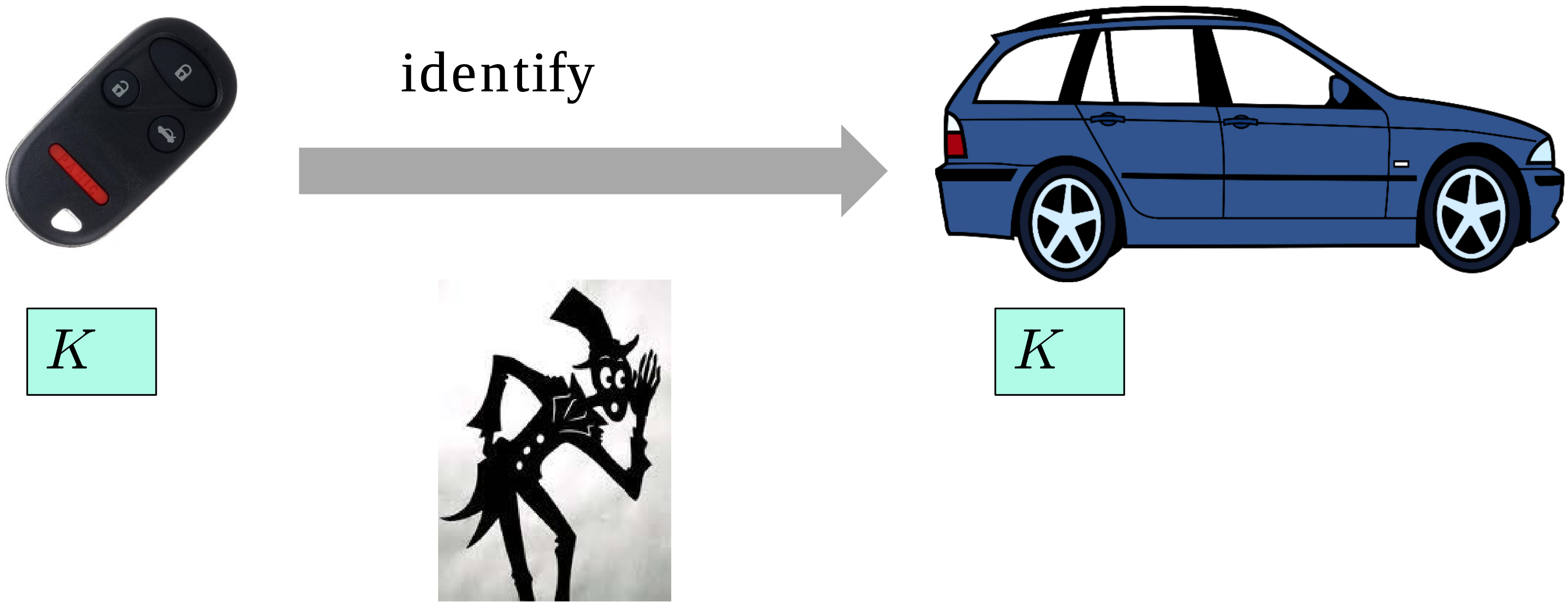


$$\text{Server's Seq \#} = T + F_K(A \| 1234 \| S \| 80)$$

Agenda

1. Blockciphers
2. Birthday Attack
3. App: TCP Sequence Number
4. **App: One-time Password**
5. App: Challenge-Response Protocol

Motivation



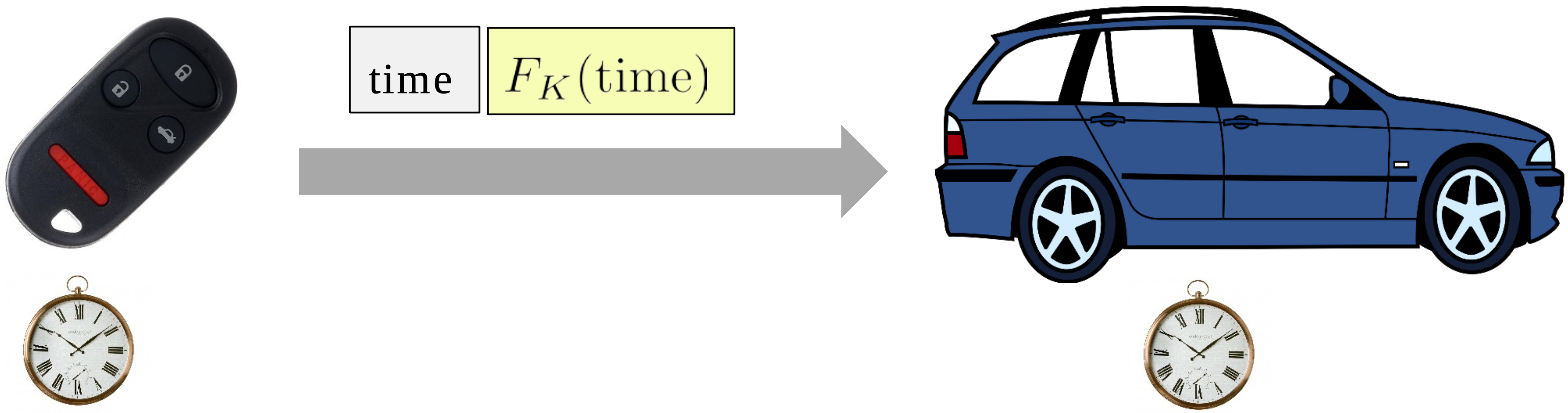
Goal: An eavesdropper cannot later open the car

A Wrong Solution

Question: Why is it bad?

One-Time Password Via PRF

https://tools.ietf.org/html/rfc6238



Should allow time drift, and accept for slightly outdated time

(Stateful) alternative: Run the PRF on a synchronized counter

https://tools.ietf.org/html/rfc4226

A Real-world Example: RSA's SecureID


The Register

This article is more than **1 year old**

SecurID breach cost RSA $66m

In 2nd quarter alone

Dan Goodin

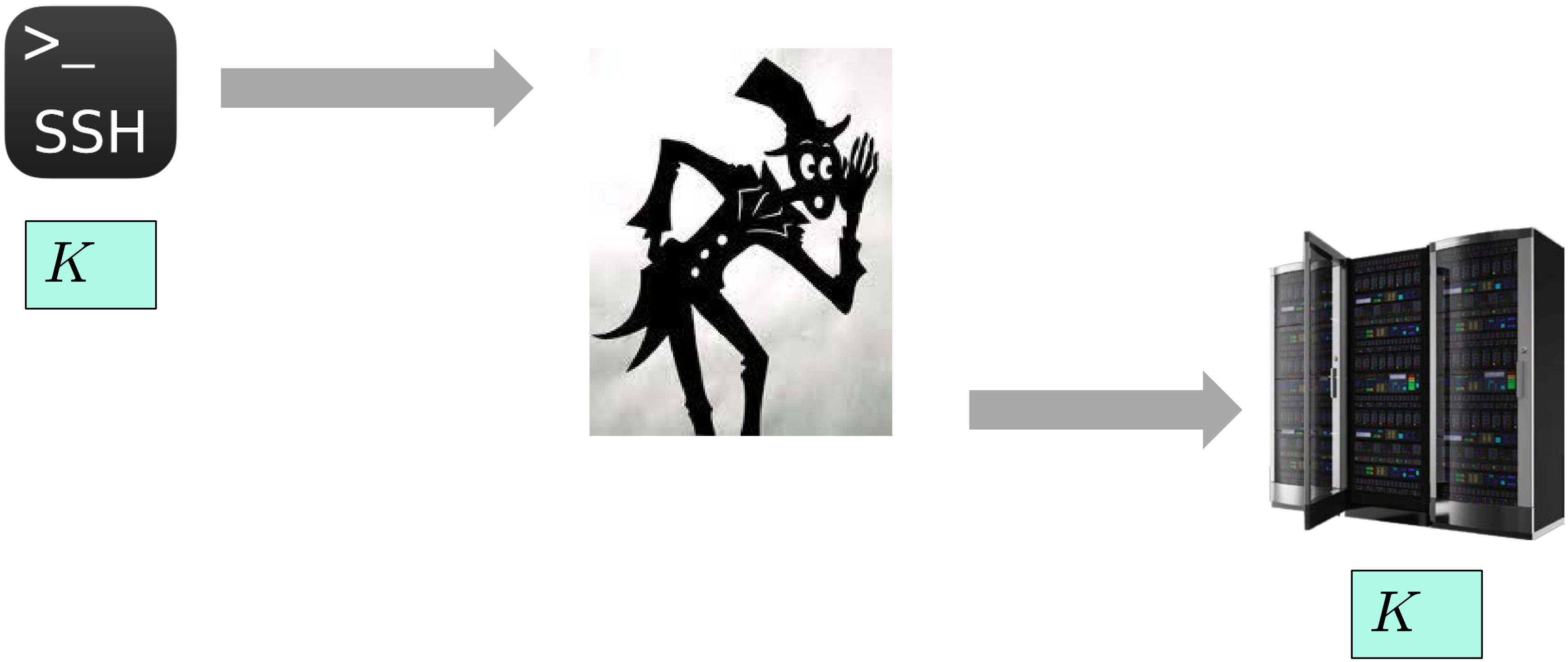
Wed 27 Jul 2011 // 17:17 UTC

But it's disastrous if the key is stolen

Agenda

1. Blockciphers
2. Birthday Attack
3. App: TCP Sequence Number
4. App: One-time Password
5. **App: Challenge-Response Protocol**

Motivation: Man-In-The-Middle Attack



Question: Does one-time password work here?

Solution: Challenge-Response

Nonce: a string that should never repeat

>_ SSH

N

$F_K(N)$