

COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

Previously on COS 433...

Takeaway: Crypto is Hard

Designing crypto is hard, even experts get it wrong

- Just because I don't know how to break it doesn't mean someone else can't

Unexpected attack vectors

- Known/chosen plaintext attack
- Chosen *ciphertext* attack
- Timing attack
- Power analysis
- Acoustic cryptanalysis

Takeaway: Crypto is Hard

Don't design your own crypto

- You'll probably get it wrong
- Use peer-reviewed schemes instead

Actually, don't even implement your own crypt

- Instead, use well studied crypto library built and tested by many experts

Takeaway: Need for Formalism

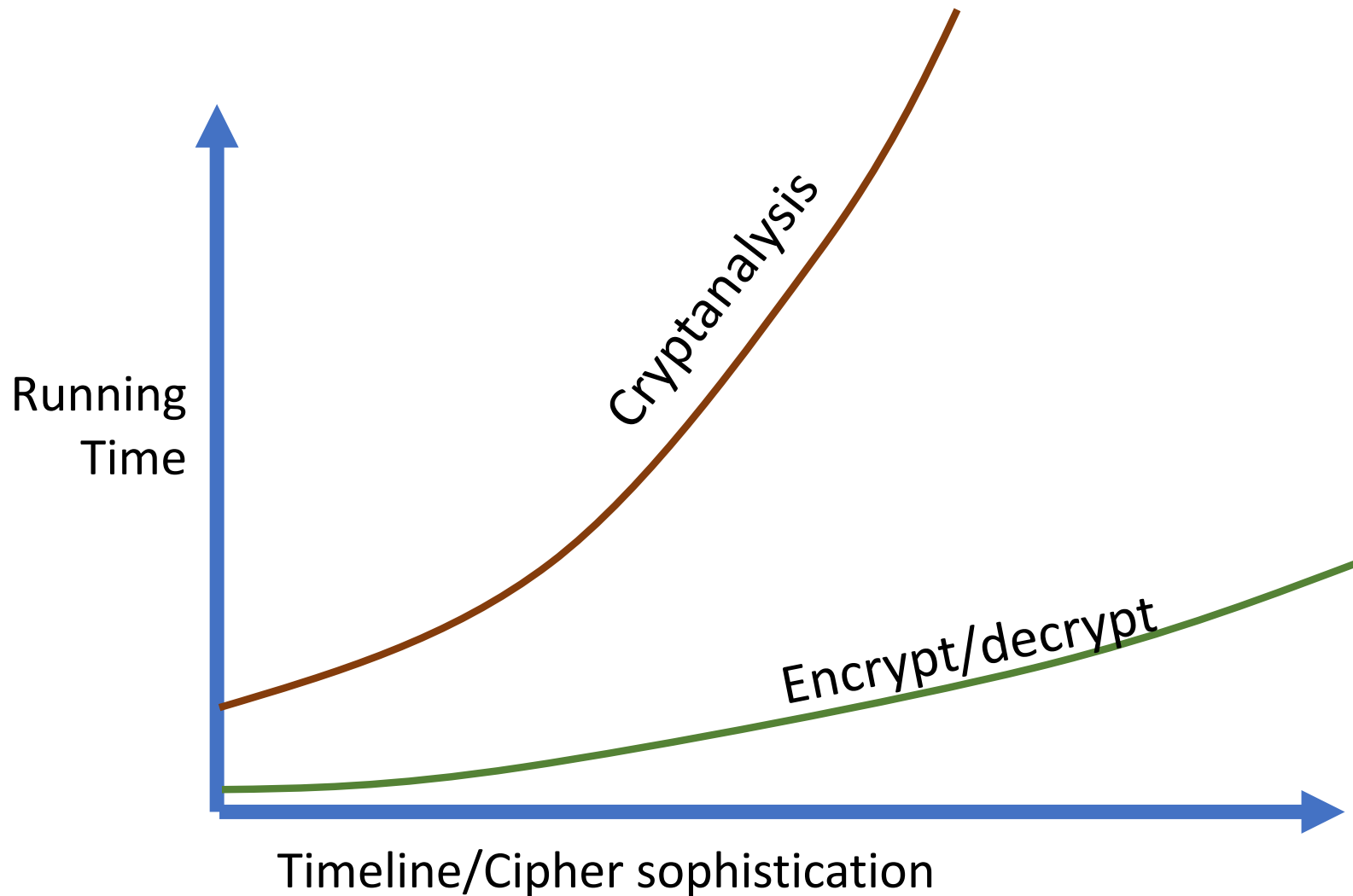
For most of history, cipher design and usage based largely on intuition

- Intuition in many cases false

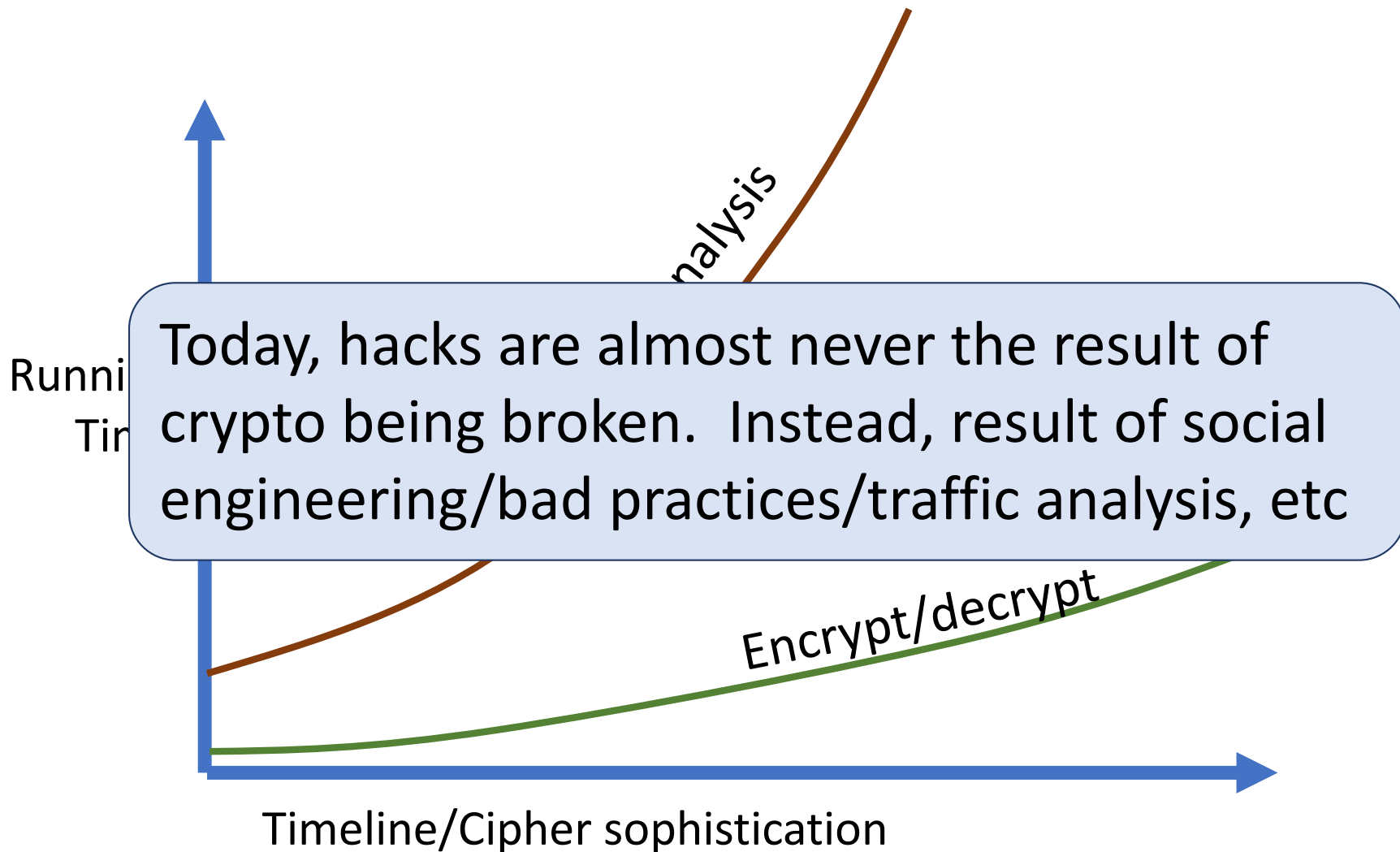
Instead, need to formally define the usage scenario

- Prove that scheme is secure in scenario
- Only use scheme in that scenario

Takeaway: Importance of Computers



Takeaway: Importance of Computers



Modern Cryptography

Encryption Basics (for now)

Syntax:

- Key space \mathbf{K} (usually $\{0,1\}^\lambda$)
- Message space \mathbf{M} (usually $\{0,1\}^n$)
- Ciphertext space \mathbf{C} (usually $\{0,1\}^m$)
- **Enc:** $\mathbf{K} \times \mathbf{M} \rightarrow \mathbf{C}$
- **Dec:** $\mathbf{K} \times \mathbf{C} \rightarrow \mathbf{M}$

Correctness (aka Completeness):

- For all $\mathbf{k} \in \mathbf{K}$, $\mathbf{m} \in \mathbf{M}$, $\mathbf{Dec}(\mathbf{k}, \mathbf{Enc}(\mathbf{k}, \mathbf{m})) = \mathbf{m}$

The One-Time Pad

Key space $\mathbf{K} = \{0,1\}^n$

Message space $\mathbf{M} = \{0,1\}^n$

Ciphertext space $\mathbf{C} = \{0,1\}^n$

$\mathbf{Enc}(k, m) = k \oplus m$

$\mathbf{Dec}(k, c) = k \oplus c$

Correctness:

$$\begin{aligned}\mathbf{Dec}(k, \mathbf{Enc}(k, m)) &= k \oplus (k \oplus m) \\ &= (k \oplus k) \oplus m \\ &= 0 \oplus m \\ &= m\end{aligned}$$

Encryption Security?

Questions to think about:

What kind of messages?

What does the adversary already know?

What information are we trying to protect?

Examples:

- Messages are always either “attack at dawn” or “attack at dusk”, trying to hide which is the case
- Messages are status updates (“<person> reports <event> at <location>”). Which data is sensitive?

Encryption Security?

Questions to think about:

What kind of messages?

What does the adversary already know?

What information are we trying to protect?

Goal:

Rather than design a separate system for each use case, design a system that works in all possible settings

Semantic Security

Idea:

- Plaintext comes from an arbitrary distribution
- Adversary initially has some information about the plaintext
- Seeing the ciphertext should not reveal any more information
- Model unknown key by assuming it is chosen uniformly at random

(Perfect) Semantic Security

Definition: A scheme **(Enc, Dec)** is **(perfectly) semantically secure** if, for all:

- Distributions **D** on **M** ← Plaintext distribution
- Functions **I: M → {0,1}^{*}** ← Info adv gets
- Functions **f: M → {0,1}^{*}** ← Info adv tries to learn
- Functions **A: C × {0,1}^{*} → {0,1}^{*}** ← Adversary

There exists a function **S: {0,1}^{*} → {0,1}^{*}** ← “Simulator” such that

$$\Pr[A(\text{Enc}(k,m) , I(m)) = f(m)] \\ = \Pr[S(I(m)) = f(m)]$$

where probabilities are taken over $k \leftarrow K, m \leftarrow D$

Semantic Security

Captures what we want out of an encryption scheme

But, complicated, with many moving parts

Want: something simpler... like perfect secrecy

Perfect Secrecy

Perfect secrecy is a great definition

- Simple
- Easy to prove

However, it doesn't obviously capture what we need

- What does adversary learn about plaintext?

Semantic Security = Perfect Secrecy

Theorem: A scheme **(Enc,Dec)** is perfectly semantically secure if and only if it has perfect secrecy

Corollary: the One-Time Pad is perfectly semantically secure

Perfect Secrecy \Rightarrow Semantic Security

Given arbitrary:

- Distribution \mathbf{D} on \mathbf{M}
- Function $\mathbf{I:M} \rightarrow \{0,1\}^*$
- Function $\mathbf{f:M} \rightarrow \{0,1\}^*$
- Function $\mathbf{A:C} \times \{0,1\}^* \rightarrow \{0,1\}^*$

Know: $\mathbf{E(K, m_0)} \stackrel{d}{=} \mathbf{E(K, m_1)}$

Goal: Construct $\mathbf{S:\{0,1\}^* \rightarrow \{0,1\}^*}$ such that

$$\begin{aligned} \Pr[\mathbf{A(Enc(k,m) , I(m)) = f(m) }] \\ = \Pr[\mathbf{S(I(m)) = f(m) }] \end{aligned}$$

Perfect Secrecy \Rightarrow Semantic Security

S(i):

- Choose random $\mathbf{k} \leftarrow \mathbf{K}$
- Set $\mathbf{c} \leftarrow \mathbf{Enc}(\mathbf{k}, \mathbf{0})$
- Run and output $\mathbf{A}(\mathbf{c}, \mathbf{i})$

$\Pr[\mathbf{S}(\mathbf{I}(m)) = \mathbf{f}(m)]$

$$= \Pr[\mathbf{A}(\mathbf{Enc}(\mathbf{k}, \mathbf{0}) , \mathbf{I}(m)) = \mathbf{f}(m) : m \leftarrow \mathbf{D}]$$

$$= \sum_{m,c} \Pr[\mathbf{D}=m] \Pr[\mathbf{Enc}(\mathbf{K}, \mathbf{0})=\mathbf{c}] \Pr[\mathbf{A}(\mathbf{c}, \mathbf{I}(m)) = \mathbf{f}(m)]$$

$$= \sum_{m,c} \Pr[\mathbf{D}=m] \Pr[\mathbf{Enc}(\mathbf{K}, m)=\mathbf{c}] \Pr[\mathbf{A}(\mathbf{c}, \mathbf{I}(m)) = \mathbf{f}(m)]$$

$$= \Pr[\mathbf{A}(\mathbf{Enc}(\mathbf{k}, m) , \mathbf{I}(m)) = \mathbf{f}(m)]$$

Semantic Security \Rightarrow Perfect Secrecy

Proof by contrapositive:

- Assume $\exists m_0, m_1$ s.t. $\text{Enc}(K, m_0) \stackrel{d}{\neq} \text{Enc}(K, m_1)$
- Devise $\mathbf{D, I, f, A}$ such that no \mathbf{S} exists

\mathbf{D} : pick $\mathbf{b} \leftarrow \{0, 1\}$ at random, output \mathbf{m}_b

\mathbf{I} : empty

$\mathbf{f}(m_b) = b$

$\mathbf{A}(c) = 1$ iff $\Pr[\text{Enc}(K, m_1) = c] > \Pr[\text{Enc}(K, m_0) = c]$

Semantic Security \Rightarrow Perfect Secrecy

Let $T = \{c: \Pr[\text{Enc}(K,m_1) = c] > \Pr[\text{Enc}(K,m_0) = c]\}$

$\Pr[A(\text{Enc}(K,m)) = f(m) : m \leftarrow D]$

$$= \frac{1}{2} \Pr[A(\text{Enc}(K,m_0)) = 0] \\ + \frac{1}{2} \Pr[A(\text{Enc}(K,m_1)) = 1]$$

$$= \frac{1}{2} \Pr[\text{Enc}(K,m_0) \notin T] \\ + \frac{1}{2} \Pr[\text{Enc}(K,m_1) \in T]$$

$$= \frac{1}{2} + \frac{1}{2} (\Pr[\text{Enc}(K,m_1) \in T] \\ - \Pr[\text{Enc}(K,m_0) \in T])$$

Semantic Security \Rightarrow Perfect Secrecy

$$\begin{aligned}\Pr[\text{Enc}(K, m_b) \in T] \\ &= \sum_{c \in T} \Pr[\text{Enc}(K, m_b) = c] \\ &= 1 - \sum_{c \notin T} \Pr[\text{Enc}(K, m_b) = c]\end{aligned}$$

$$\begin{aligned}\Pr[\text{Enc}(K, m_1) \in T] - \Pr[\text{Enc}(K, m_0) \in T] \\ &= \sum_{c \in T} \Pr[\text{Enc}(K, m_1) = c] - \Pr[\text{Enc}(K, m_0) = c] \\ &= \sum_{c \notin T} \Pr[\text{Enc}(K, m_0) = c] - \Pr[\text{Enc}(K, m_1) = c] \\ &= \frac{1}{2} \sum_c | \Pr[\text{Enc}(K, m_1) = c] - \Pr[\text{Enc}(K, m_0) = c] |\end{aligned}$$

Perfect Secrecy vs Semantic Security

Perfect secrecy is much easier to reason about, so we will usually analyze schemes for perfect secrecy

However, semantic security is really the definition we care about, so always keep in mind



Proper Use Case for Perfect Security

- Message can come from any distribution ✓
- Adversary can know anything about message ✓
- Encryption hides anything ✓
- But, definition only says something about an adversary that sees a single message ✗
 - ⇒ If two messages, no security guarantee
- Assumes no side-channels ✗
- Assumes key is uniformly random ✗

One-time Pad

We know OTP is perfectly semantically secure 

But, we know it is insecure if:

- Used to encrypt multiple messages 
- Key length shorter than message 

Variable-Length Messages

OTP has message-length $\{0,1\}^n$ where n is the key length

In practice, fixing the message size is unreasonable

So instead, will allow for smaller messages to be encrypted

Variable-Length OTP

Key space $\mathbf{K} = \{0,1\}^n$

Message space $\mathbf{M} = \{0,1\}^{\leq n}$

Ciphertext space $\mathbf{C} = \{0,1\}^{\leq n}$

$$\mathbf{Enc}(k, m) = k_{[1, |m|]} \oplus m$$

$$\mathbf{Dec}(k, c) = k_{[1, |c|]} \oplus c$$

Correctness:

$$\begin{aligned} \mathbf{Dec}(k, \mathbf{Enc}(k, m)) &= k \oplus (k \oplus m) \\ &= (k \oplus k) \oplus m \\ &= 0 \oplus m \\ &= m \end{aligned}$$

Does the variable length OTP
have perfect secrecy according
to our definition?

Ciphertext Size

Theorem: For scheme with perfect secrecy, the expected ciphertext size for any message, $\mathbb{E}[|\text{Enc}(K,m)|]$, is at least $(\log_2 |M|) - 3$

Proof

Fix a key \mathbf{k} .

Let $\mathbf{C}_{\mathbf{k},m}$ be set of ciphertexts \mathbf{c} s.t. $\Pr[\text{Enc}(\mathbf{k},m)=\mathbf{c}]>0$

By correctness, each $\mathbf{C}_{\mathbf{k},m}$ as m varies are disjoint and non-empty

• If $\mathbf{c} \in \mathbf{C}_{\mathbf{k},m}$ and $\mathbf{c} \in \mathbf{C}_{\mathbf{k},m'}$, then $m' = \text{Dec}(\mathbf{k},\mathbf{c}) = m$

Therefore, therefore $|\cup_m \mathbf{C}_{\mathbf{k},m}| \geq |M|$

Proof

$$|\cup_m C_{k,m}| \geq |M|$$

Therefore, if we encrypt a random message, the expected size of a ciphertext is at least

$$\sum_m \min(|c| : c \in C_{k,m}) / |M|$$

$\min(|c| : c \in C_{k,m}) = t$ for at most 2^t different m

Proof

Let $r = \text{floor}(\log_2 |M|)$

$$\begin{aligned} & \sum_m \min(|c| : c \in C_{k,m}) / |M| \\ &= (1 \times 0 + 2 \times 1 + 4 \times 2 + \dots + 2^{r-1} \times (r-1) \\ & \quad + (|M| - (2^r - 1)) \times r) / |M| \\ &= (2^r(r-2) + 2 + (|M| - (2^r - 1)) \times r) / |M| \\ &= (r - 2(2^r - 1) + |M| \times r) / |M| \\ &\geq (0 - 2|M| + |M| \times r) / |M| = r - 2 \end{aligned}$$

Proof

Therefore, for a random message, the expected ciphertext length for any key is at least $\log_2 |M| - 3$

Must also be true for a random key \mathbf{k}

By perfect secrecy, for any messages $\mathbf{m}_0, \mathbf{m}_1$

$$\mathbb{E}_{\mathbf{K}}[|\text{Enc}(\mathbf{K}, \mathbf{m}_0)|] = \mathbb{E}_{\mathbf{K}}[|\text{Enc}(\mathbf{K}, \mathbf{m}_1)|]$$

Therefore,

$$\begin{aligned} \mathbb{E}_{\mathbf{K}}[|\text{Enc}(\mathbf{K}, \mathbf{m}_0)|] \\ = \mathbb{E}_{\mathbf{K}, \mathbf{M}}[|\text{Enc}(\mathbf{K}, \mathbf{M})|] \geq \log_2 |M| - 3 \end{aligned}$$

Variable-Length Messages

For perfect secrecy of variable length messages, must have expected ciphertext length for short messages almost as long as longest messages

In practice, very undesirable

- What if I want to either send a **100mb** attachment, or just a message “How are you?”

Therefore, we usually allow message length to be revealed

(Perfect) Semantic Security for Variable Length Messages

Definition: A scheme **(Enc, Dec)** is **(perfectly) semantically secure** if, for all:

- Distributions **D** on **M**
- (Probabilistic) Functions **I: M → {0,1}***
- (Probabilistic) Functions **f: M → {0,1}***
- (Probabilistic) Functions **A: C × {0,1}* → {0,1}***

There exists (probabilistic) func **S: {0,1}* → {0,1}*** s.t.

$$\begin{aligned} & \Pr[A(\text{Enc}(k,m) , I(m)) = f(m)] \\ & = \Pr[S(I(m), |m|) = f(m)] \end{aligned}$$

where probabilities are taken over $k \leftarrow K, m \leftarrow D$

Perfect Secrecy For Variable Length Messages

Definition: A scheme **(Enc,Dec)** has **perfect secrecy** if, for any two messages m_0, m_1 where $|m_0| = |m_1|$,

$$\text{Enc}(K, m_0) \stackrel{d}{=} \text{Enc}(K, m_1)$$

Easy to adapt earlier proof to show:

Theorem: A scheme **(Enc,Dec)** is semantically secure if and only if it has perfect secrecy

Variable-Length OTP

Key space $\mathbf{K} = \{0,1\}^n$

Message space $\mathbf{M} = \{0,1\}^{\leq n}$

Ciphertext space $\mathbf{C} = \{0,1\}^{\leq n}$

$$\mathbf{Enc}(k, m) = k_{[1, |m|]} \oplus m$$

$$\mathbf{Dec}(k, c) = k_{[1, |m|]} \oplus c$$

Theorem: Variable-Length OTP has perfect secrecy

Re-using the OTP

What if we have a **100mb** long key **k**, but encrypt only **1mb**?

Can't use first **1mb** of **k** again, but remaining **99mb** is still usable

However, basic OTP definition does not allow us to re-use the key ever

Syntax for Stateful Encryption

Syntax:

- Key space \mathbf{K} , Message space \mathbf{M} , Ciphertext space \mathbf{C}
- State Space \mathbf{S}
- **Init:** $\{\} \rightarrow \mathbf{S}$
- **Enc:** $\mathbf{K} \times \mathbf{M} \times \mathbf{S} \rightarrow \mathbf{C} \times \mathbf{S}$
- **Dec:** $\mathbf{K} \times \mathbf{C} \times \mathbf{S} \rightarrow \mathbf{M} \times \mathbf{S}$

$\text{State}_0 \leftarrow \text{Init}()$

$(c_0, \text{state}_1) \leftarrow \text{Enc}(k, m_0, \text{state}_0)$

$(c_1, \text{state}_2) \leftarrow \text{Enc}(k, m_1, \text{state}_1)$

...

Reusing the OTP

k

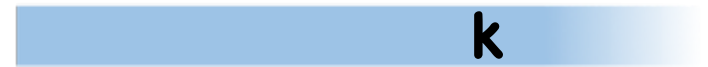
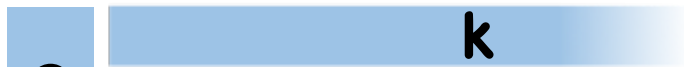
m



k



Reusing the OTP



Reusing the OTP

k

k

c



Reusing the OTP

k

k



Reusing the OTP

k



k

c



Reusing the OTP

k



k

\oplus
c
↓
m



Reusing the OTP

k



k

m



Reusing the OTP

k



k

m'



Reusing the OTP

k



⊕ k

\oplus
m'



c'



Reusing the OTP

k

k



c'



Reusing the OTP

k

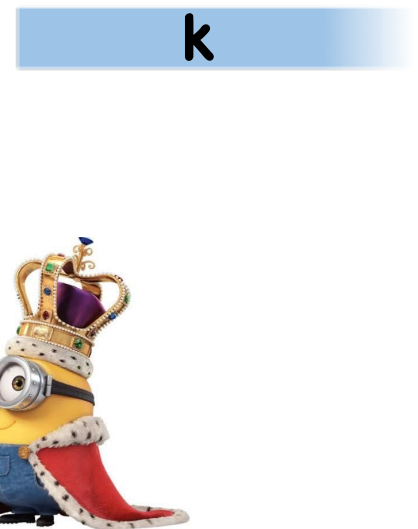
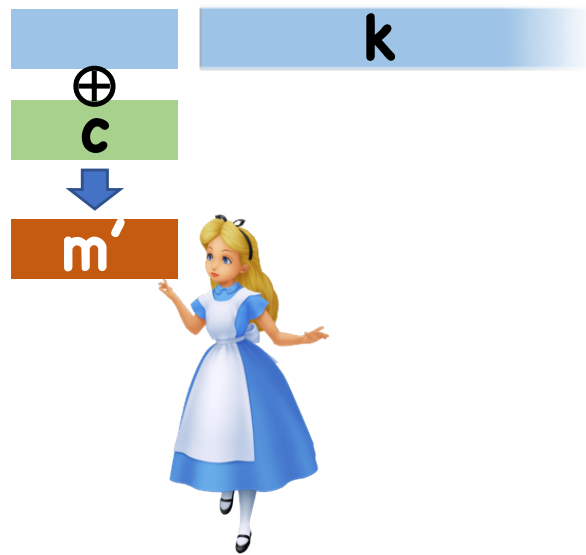
c'



k



Reusing the OTP



Problem

In real world, messages aren't always synchronous

What happens if Alice and Bob try to send message at the same time?

They will both use the same part of the key!

Problem

k

m

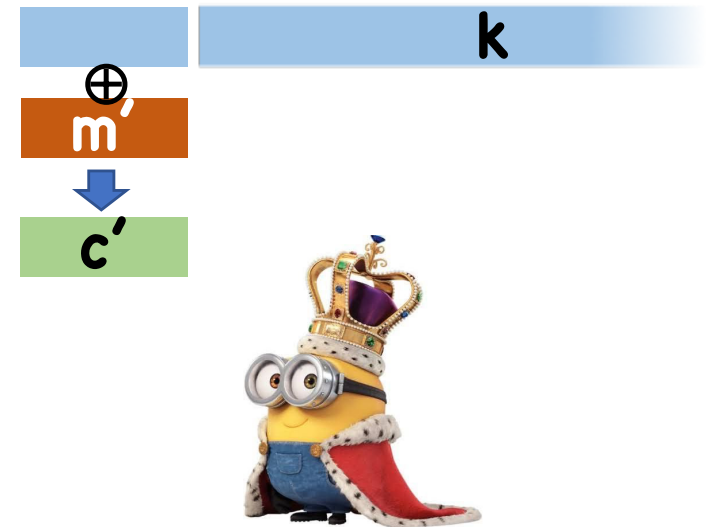
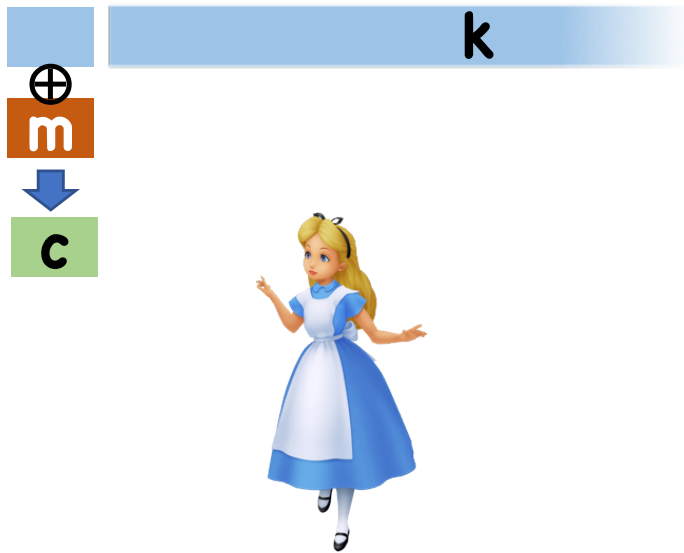


k

m'



Problem



Problem

k

k

c

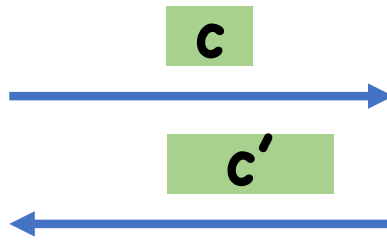


c'



Problem

k

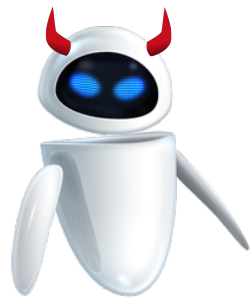


k



Problem

k



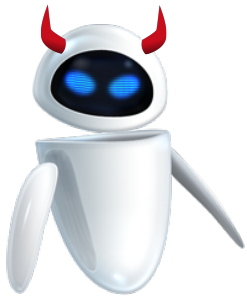
c
c'

k



Problem

k



k



Solution

Alice and Bob have two keys

- One for communication from Alice to Bob
- One for communication from Bob to Alice

Can obtain two logical keys from one by splitting key in half

- Ex: odd bits form $\mathbf{k}_{A \rightarrow B}$, even bits form $\mathbf{k}_{B \rightarrow A}$

Reusing the OTP

$k_{A \rightarrow B}$

$k_{B \rightarrow A}$



$k_{A \rightarrow B}$

$k_{B \rightarrow A}$



Still A Problem

In real world, messages aren't always synchronous

Also, sometimes messages arrive out of order or get dropped

- Need to be very careful to make sure decryption succeeds

These difficulties exist in any stateful encryption

- For this course, we will generally consider only **stateless** encryption

Perfect Security for Multiple Messages

Definition: A stateless scheme **(Enc, Dec)** has **perfect secrecy for n messages** if, for any two sequences of messages $(m_0^{(i)})_{i \in [n]}$, $(m_1^{(i)})_{i \in [n]} \in M^n$

$$\left(\text{Enc}(K, m_0^{(i)}) \right)_{i \in [n]} \stackrel{d}{=} \left(\text{Enc}(K, m_1^{(i)}) \right)_{i \in [n]}$$

Notation: $(f(i))_{i \in [n]} = (f(1), f(2), \dots, f(n))$

Stateless Encryption with Multiple Messages

Ex:

$$\mathbf{M} = \mathbf{C} = \mathbb{Z}_p \text{ (p a prime)}$$

$$\mathbf{K} = \mathbb{Z}_p^* \times \mathbb{Z}_p$$

$$\mathbf{Enc}(\mathbf{(a,b)}, \mathbf{m}) = \mathbf{(am + b) \bmod p}$$

$$\mathbf{Dec}(\mathbf{(a,b)}, \mathbf{c}) = \mathbf{(c-b)/a \bmod p}$$

Q: Is this perfectly secure for two messages?

Theorem: No stateless deterministic encryption scheme can have perfect security for multiple messages

Randomized Encryption

Syntax:

- Key space \mathbf{K} (usually $\{0,1\}^\lambda$)
- Message space \mathbf{M} (usually $\{0,1\}^n$)
- Ciphertext space \mathbf{C} (usually $\{0,1\}^m$)
- **Enc:** $\mathbf{K} \times \mathbf{M} \rightarrow \mathbf{C}$ (potentially probabilistic)
- **Dec:** $\mathbf{K} \times \mathbf{C} \rightarrow \mathbf{M}$ (usually deterministic)

Correctness:

- ~~• For all $k \in \mathbf{K}$, $m \in \mathbf{M}$, $\text{Dec}(k, \text{Enc}(k,m)) = m$~~

Randomized Encryption

Syntax:

- Key space \mathbf{K} (usually $\{0,1\}^\lambda$)
- Message space \mathbf{M} (usually $\{0,1\}^n$)
- Ciphertext space \mathbf{C} (usually $\{0,1\}^m$)
- **Enc:** $\mathbf{K} \times \mathbf{M} \rightarrow \mathbf{C}$ (potentially probabilistic)
- **Dec:** $\mathbf{K} \times \mathbf{C} \rightarrow \mathbf{M}$ (usually deterministic)

Correctness:

- For all $k \in \mathbf{K}$, $m \in \mathbf{M}$,
$$\Pr[\text{Dec}(k, \text{Enc}(k,m)) = m] = 1$$

Stateless Encryption with Multiple Messages

Ex:

$$\mathbf{M} = \mathbb{Z}_p \text{ (p a prime)}$$

$$\mathbf{C} = \mathbb{Z}_p^2$$

$$\mathbf{K} = \mathbb{Z}_p^2$$

$$\mathbf{Enc}((a,b), m) = (r, (ar+b) + m)$$

$$\mathbf{Dec}((a,b), (r,c)) = c - (ar+b)$$

Random in \mathbb{Z}_p



Q: Is this perfectly secure for two messages?

Proof of Easy Case

Let **(Enc, Dec)** be stateless, deterministic

Let $\mathbf{m}_0^{(0)} = \mathbf{m}_0^{(1)}$

Let $\mathbf{m}_1^{(0)} \neq \mathbf{m}_1^{(1)}$

Consider distributions of encryptions:

- $(\mathbf{c}^{(0)} , \mathbf{c}^{(1)}) = (\mathbf{Enc}(K, \mathbf{m}_0^{(0)}) , \mathbf{Enc}(K, \mathbf{m}_0^{(1)}))$
 $\Rightarrow \mathbf{c}^{(0)} = \mathbf{c}^{(1)}$ (by determinism)
- $(\mathbf{c}^{(0)} , \mathbf{c}^{(1)}) = (\mathbf{Enc}(K, \mathbf{m}_1^{(0)}) , \mathbf{Enc}(K, \mathbf{m}_1^{(1)}))$
 $\Rightarrow \mathbf{c}^{(0)} \neq \mathbf{c}^{(1)}$ (by correctness)

Generalize to Randomized Encryption

Let **(Enc, Dec)** be stateless, ~~deterministic~~

Let $\mathbf{m}_0^{(0)} = \mathbf{m}_0^{(1)}$

Let $\mathbf{m}_1^{(0)} \neq \mathbf{m}_1^{(1)}$

Consider distributions of encryptions:

• $(\mathbf{c}^{(0)} , \mathbf{c}^{(1)}) = (\mathbf{Enc}(K, \mathbf{m}_0^{(0)}) , \mathbf{Enc}(K, \mathbf{m}_0^{(1)}))$

\Rightarrow ????

• $(\mathbf{c}^{(0)} , \mathbf{c}^{(1)}) = (\mathbf{Enc}(K, \mathbf{m}_1^{(0)}) , \mathbf{Enc}(K, \mathbf{m}_1^{(1)}))$

$\Rightarrow \mathbf{c}^{(0)} \neq \mathbf{c}^{(1)}$ (by correctness)

Generalize to Randomized Encryption

$$(c^{(0)}, c^{(1)}) = (\text{Enc}(K, m), \text{Enc}(K, m))$$

$\Pr[c^{(0)} = c^{(1)}]$?

- Fix **k**
- Conditioned on **k** , **$c^{(0)}$** , **$c^{(1)}$** are two independent samples from same distribution **$\text{Enc}(k, m)$**

Lemma: Given any distribution **D** over a finite set **X** , **$\Pr[Y=Y': Y \leftarrow D, Y' \leftarrow D] \geq 1/|X|$**

- Therefore, **$\Pr[c^{(0)} = c^{(1)}]$** is non-zero

Generalize to Randomized Encryption

Let **(Enc, Dec)** be stateless, deterministic

Let $\mathbf{m}_0^{(0)} = \mathbf{m}_0^{(1)}$

Let $\mathbf{m}_1^{(0)} \neq \mathbf{m}_1^{(1)}$

Consider distributions of encryptions:

$$\bullet (\mathbf{c}^{(0)} , \mathbf{c}^{(1)}) = (\text{Enc}(\mathbf{K}, \mathbf{m}_0^{(0)}) , \text{Enc}(\mathbf{K}, \mathbf{m}_0^{(1)}))$$

$$\Rightarrow \Pr[\mathbf{c}^{(0)} = \mathbf{c}^{(1)}] > 0$$

$$\bullet (\mathbf{c}^{(0)} , \mathbf{c}^{(1)}) = (\text{Enc}(\mathbf{K}, \mathbf{m}_1^{(0)}) , \text{Enc}(\mathbf{K}, \mathbf{m}_1^{(1)}))$$

$$\Rightarrow \Pr[\mathbf{c}^{(0)} = \mathbf{c}^{(1)}] = 0$$

What do we do now?

Tolerate tiny probability of distinguishing

- If $\Pr[\mathbf{c}^{(0)} = \mathbf{c}^{(1)}] = 2^{-128}$, in reality never going to happen

How small is ok?

- Usually 2^{-80} , 2^{-128} , or maybe 2^{-258}

Next time: formalize weaker notion of secrecy to allow for small probability of detection

Reminders

HW1 Due Tomorrow (Feb 13th)

Start working on PR1

- Part 1 is due next week (Feb 20th)