

# COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

# Announcements

Reminder: Homework 1 due tomorrow 11:59pm

- Submit through Blackboard

Homework 2 will hopefully be posted tonight

# Last Time

Deterministic and Randomized Encryption

Perfect Secrecy = Perfect Semantic Security

OTP

# Statistical Distance

Given two distributions  $\mathbf{D}_1, \mathbf{D}_2$  over a set  $\mathbf{X}$ , define

$$\Delta(\mathbf{D}_1, \mathbf{D}_2) = \frac{1}{2} \sum_{\mathbf{x}} | \Pr[\mathbf{D}_1 = \mathbf{x}] - \Pr[\mathbf{D}_2 = \mathbf{x}] |$$

Observations:

$$0 \leq \Delta(\mathbf{D}_1, \mathbf{D}_2) \leq 1$$

$$\Delta(\mathbf{D}_1, \mathbf{D}_2) = 0 \iff \mathbf{D}_1 = \mathbf{D}_2$$

$$\Delta(\mathbf{D}_1, \mathbf{D}_2) \leq \Delta(\mathbf{D}_1, \mathbf{D}_3) + \Delta(\mathbf{D}_3, \mathbf{D}_2)$$

( $\Delta$  is a metric)

# Perfect Secrecy [Shannon'49]

**Definition:** A scheme **(Enc, Dec)** has **perfect secrecy** if, for any two messages  $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{M}$

$$\text{Enc}(\mathbf{K}, \mathbf{m}_0) \stackrel{d}{=} \text{Enc}(\mathbf{K}, \mathbf{m}_1)$$

Identical distributions

Random variable corresponding  
to uniform distribution over  $\mathbf{K}$

Random variable corresponding  
to encrypting  $\mathbf{m}_1$  using a  
uniformly random key

# Perfect Secrecy [Shannon'49]

**Definition:** A scheme **(Enc,Dec)** has **perfect secrecy** if, for any two messages  $m_0, m_1 \in M$

$$\Delta( \text{Enc}(K, m_0), \text{Enc}(K, m_1) ) = 0$$

# Another View of Statistical Distance

**Theorem:**  $\Delta(\mathcal{D}_1, \mathcal{D}_2) \geq \varepsilon$  iff  $\exists \mathbf{A}$  s.t.  
 $\left| \Pr[\mathbf{A}(\mathcal{D}_1) = 1] - \Pr[\mathbf{A}(\mathcal{D}_2) = 1] \right| \geq \varepsilon/2$

**Terminology:** for any  $\mathbf{A}$ ,  
 $\left| \Pr[\mathbf{A}(\mathcal{D}_1) = 1] - \Pr[\mathbf{A}(\mathcal{D}_2) = 1] \right|$   
is called the “advantage” of  $\mathbf{A}$  in  
distinguishing  $\mathcal{D}_1$  and  $\mathcal{D}_2$

# Another View of Statistical Distance

**Theorem:**  $\Delta(D_1, D_2) \geq \varepsilon$  iff  $\exists \mathbf{A}$  s.t.  
 $|\Pr[\mathbf{A}(D_1) = 1] - \Pr[\mathbf{A}(D_2) = 1]| \geq \varepsilon/2$

To lower bound  $\Delta$ , just need to show  
adversary  $\mathbf{A}$  with twice that advantage



# Obtaining Perfect Secrecy: The One-Time Pad

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

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

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

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

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

Example:

$k = 0011010110$

$m = 100101$

$c = 101000$

Correctness:

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

# Limitations of OTP

## It is only one-time

- Try to encrypt two messages, security will fail

$$\begin{aligned}\mathbf{Enc(k, m_0)} \oplus \mathbf{Enc(k, m_1)} \\ &= (\mathbf{k} \oplus \mathbf{m_0}) \oplus (\mathbf{k} \oplus \mathbf{m_1}) \\ &= \mathbf{m_0} \oplus \mathbf{m_1}\end{aligned}$$

## Key length $\geq$ message length

- Limited use in practice: if I can securely transmit **n**-bit key, why don't I just use that to transmit **n**-bit message?

# Today

## Multiple message security

### Using the OTP more than once

- Stateful encryption
- Limitations

### Multiple messages with stateless encryption

- Impossibility of perfect secrecy
- Security parameter
- Statistical secrecy

# Reusing the OTP

For today, assume both parties have extremely long shared secret key

When encrypting  $\mathbf{m}$  s.t.  $|\mathbf{m}| \ll |\mathbf{k}|$ , don't have to throw away all of  $\mathbf{k}$

- Only  $\mathbf{k}_{[1, |\mathbf{m}|]}$  has been used
- Use rest of  $\mathbf{k}$  to encrypt next message

# Syntax for Stateful Encryption

## Syntax:

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

**State<sub>0</sub> ← Init()**

**(c<sub>0</sub>, state<sub>1</sub>) ← Enc(k, m<sub>0</sub>, state<sub>0</sub>)**

**(c<sub>1</sub>, state<sub>2</sub>) ← Enc(k, m<sub>1</sub>, state<sub>1</sub>)**

**...**

# Reusing the OTP

k

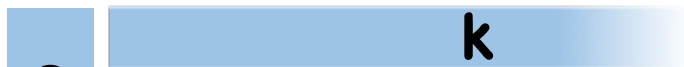
m



k



# Reusing the OTP



# Reusing the OTP

k

k

c





# Reusing the OTP

k



c



k



# Reusing the OTP

k



k

c



# Reusing the OTP

k



k



# Reusing the OTP

k



k



# Reusing the OTP

k



k

m'



# Reusing the OTP

$k$



$\oplus$   $k$

$m'$



$c'$



# Reusing the OTP

$k$

$k$



$c'$



# Reusing the OTP

k

c'

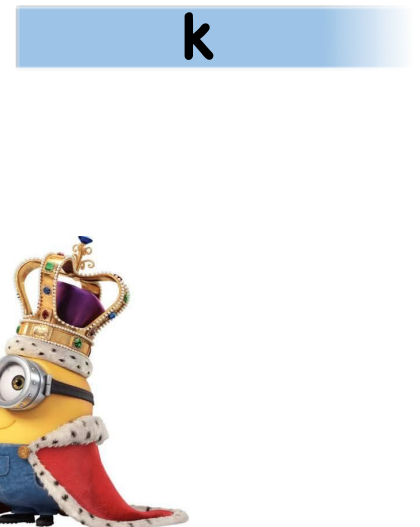
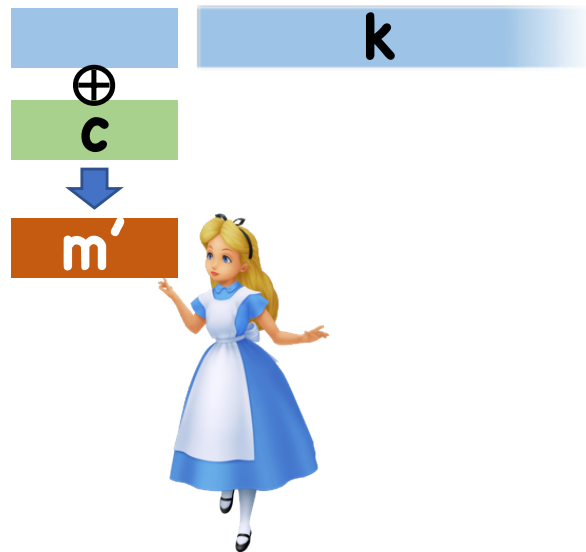


k





# 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

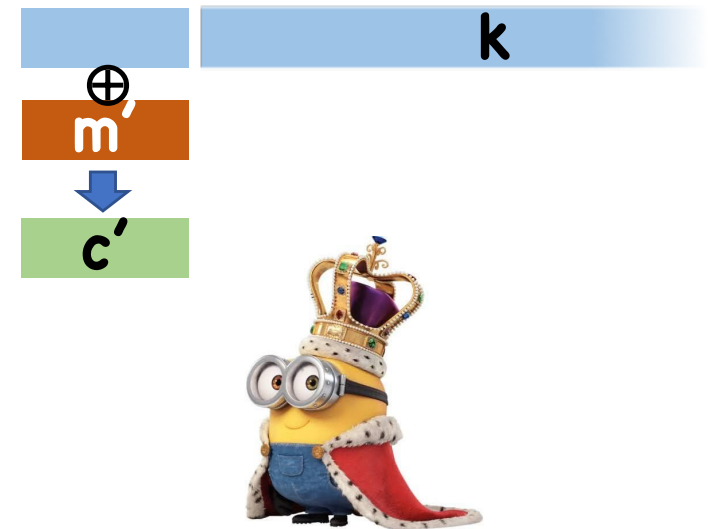
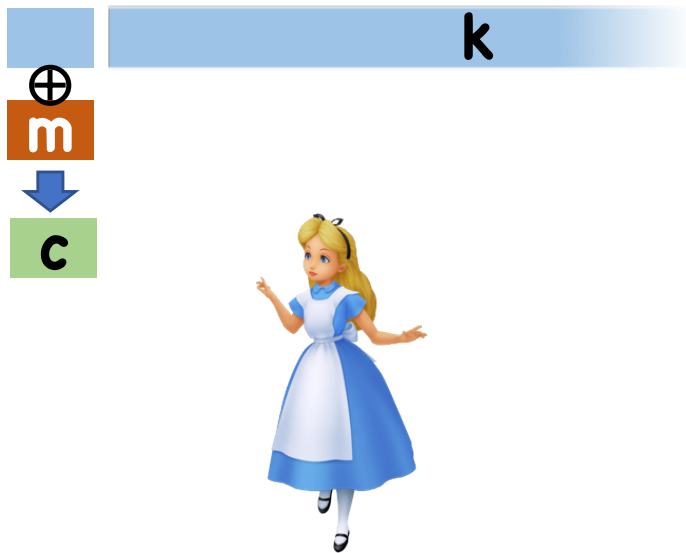


k

m'



# Problem



# Problem

k

c

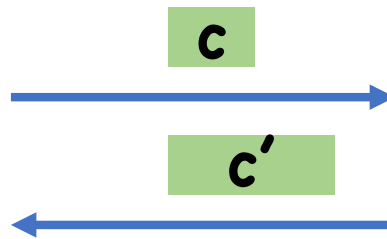
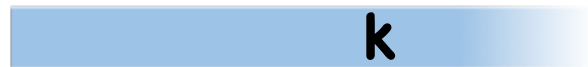


k

c'

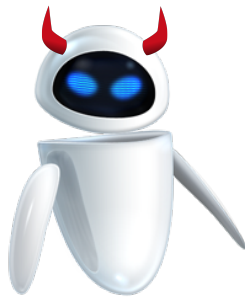


# Problem



# Problem

k



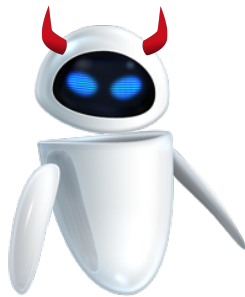
c

c'

k



# Problem





# 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

# Back to Stateless Encryption

## Syntax:

- Key space  $\mathbf{K}$
- Message space  $\mathbf{M}$
- Ciphertext space  $\mathbf{C}$
- **Enc:**  $\mathbf{K} \times \mathbf{M} \rightarrow \mathbf{C}$
- **Dec:**  $\mathbf{K} \times \mathbf{C} \rightarrow \mathbf{M}$

# Perfect Security for Multiple Messages

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

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

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

# Stateless Encryption with Multiple Messages

Ex:

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

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

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

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

Q: Is this perfectly secure for two messages?

# Stateless Encryption with Multiple Messages

Ex:

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

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

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

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

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

Random in  $\mathbb{Z}_p$



Q: Is this perfectly secure for two messages?

# Stateless Encryption with Multiple Messages

**Theorem:** No stateless encryption scheme\*  
can be perfectly secure for two messages

\* with finite ciphertext size



# Stateless Encryption with Multiple Messages

Easier case:

**Theorem:** No stateless *deterministic* encryption scheme can be 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:

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

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

•  $(\mathbf{c}^{(0)}, \mathbf{c}^{(1)}) = (\text{Enc}(\mathbf{K}, \mathbf{m}_1^{(0)}), \text{Enc}(\mathbf{K}, \mathbf{m}_1^{(1)}))$   
 $\Rightarrow \mathbf{c}^{(0)} \neq \mathbf{c}^{(1)}$  (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  **$\mathbf{D}$**  over a finite set  **$\mathbf{X}$** ,  **$\Pr[Y=Y': Y \leftarrow \mathbf{D}, Y' \leftarrow \mathbf{D}] \geq 1/|\mathbf{X}|$**

- 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:

- $(\mathbf{c}^{(0)}, \mathbf{c}^{(1)}) = (\text{Enc}(\mathbf{K}, \mathbf{m}_0^{(0)}), \text{Enc}(\mathbf{K}, \mathbf{m}_0^{(1)}))$   
 $\Rightarrow \Pr[\mathbf{c}^{(0)} = \mathbf{c}^{(1)}] > 0$
- $(\mathbf{c}^{(0)}, \mathbf{c}^{(1)}) = (\text{Enc}(\mathbf{K}, \mathbf{m}_1^{(0)}), \text{Enc}(\mathbf{K}, \mathbf{m}_1^{(1)}))$   
 $\Rightarrow \Pr[\mathbf{c}^{(0)} = \mathbf{c}^{(1)}] = 0$

# 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?

- Practitioner:  $2^{-80}$ ,  $2^{-128}$ , or maybe  $2^{-258}$
- Theorist: ????

# Big O Notation Recap

$$O( f(\lambda) ) = \{g(\lambda): \exists c, \lambda_0, \forall \lambda > \lambda_0 \ g(\lambda) \leq cf(\lambda) \}$$

$$\Omega( f(\lambda) ) = \{g(\lambda): \exists c, \lambda_0, \forall \lambda > \lambda_0 \ g(\lambda) \geq cf(\lambda) \}$$

$$\Theta( f(\lambda) ) = O( f(\lambda) ) \cap \Omega( f(\lambda) )$$

$$o( f(\lambda) ) = O( f(\lambda) ) \setminus \Omega( f(\lambda) )$$

$$\omega( f(\lambda) ) = \Omega( f(\lambda) ) \setminus O( f(\lambda) )$$

Notation abuse:

$$g(\lambda) = O( f(\lambda) ) \text{ means } g(\lambda) \in O( f(\lambda) )$$

# Polynomial

**Def:  $f(\lambda)$  is polynomially bounded** if  $f(\lambda) \in O(\lambda^c)$  for some constant  $c$

- Sometimes will just say “ **$f(\lambda)$  is polynomial**”
- Equivalent def:  **$\log(f(\lambda)) \in O(\log \lambda)$**
- Set of polynomial functions:  **$n^{O(1)}$**

**Def:  $f(\lambda)$  is inverse polynomial** if  $f(\lambda) \in \Omega(\lambda^{-c})$  for some constant  $c$

**Def:  $f(\lambda)$  is super polynomial** if  $f(\lambda) \notin O(\lambda^c)$  for any constant  $c$



# Negligible

**Def:  $f(\lambda)$  is negligible** if  $f(\lambda) \in O(\lambda^{-c})$  for any constant  $c$

- Equivalent def:  $\log(1/f(\lambda)) \in \omega(\log \lambda)$
- Set of negligible functions:  $2^{-\omega(\log \lambda)}$

# Negligible Function Examples

Negligible:

- $c^{-\lambda}$  for any constant  $c$
- $\lambda^d c^{-\lambda}$  for any constants  $c, d$
- $\lambda^{-\log \lambda}$  (for any logarithm base)

Not negligible:

- $\lambda^{-2}$
- $1/\log \lambda$
- $\lambda^{-2}/\log \lambda$

# Negligible?

Is the following function negligible?

$$f(\lambda) = 2^{-\lambda^{1/2}}$$

Yes:  $\log(1/f(\lambda)) = \lambda^{1/2} \in \omega(\log \lambda)$

# Negligible?

Is the following function negligible?

$$f(\lambda) = 2^{-\lambda} \text{ if } \lambda \text{ odd, } 1/\lambda \text{ if } \lambda \text{ even}$$

No:  $f(\lambda) \notin O(\lambda^{-2})$

# Security Parameter

System parameterized by security parameter  $\lambda$

- Represents security level of system
- System parameters allowed to grow poly in  $\lambda$ 
  - i.e.  $|ctx|, |key| = O(\lambda^c)$  for some constant  $c$
- Adversary distinguishing advantage **negligible** in  $\lambda$
- Idea: poly = tractable, negl/superpoly = intractable

# Encryption with Security Parameter

## Syntax:

- Key space  $\mathbf{K}_\lambda$
- Message space  $\mathbf{M}$  (sometimes depends on  $\lambda$ )
- Ciphertext space  $\mathbf{C}_\lambda$
- **Enc:**  $\mathbf{K}_\lambda \times \mathbf{M} \rightarrow \mathbf{C}_\lambda$
- **Dec:**  $\mathbf{K}_\lambda \times \mathbf{C}_\lambda \rightarrow \mathbf{M}$

# Statistical Secrecy

**Definition:** A scheme **(Enc, Dec)** has **statistical secrecy for n messages** if  $\exists$  negligible function  $\epsilon$  s.t.  $\forall$  two sequences of messages  $(m_0^{(i)})_{i \in [n]}$ ,  $(m_1^{(i)})_{i \in [n]} \in M^n$

$$\Delta \left[ (Enc(K_\lambda, m_0^{(i)}))_{i \in [n]}, (Enc(K_\lambda, m_1^{(i)}))_{i \in [n]} \right] < \epsilon(\lambda)$$

# Stateless Encryption with Multiple Messages

Ex:

$$M_\lambda = C_\lambda = \mathbb{Z}_p \text{ (} p \text{ a prime of size } 2^\lambda \text{)}$$

$$K_\lambda = \mathbb{Z}_p^* \times \mathbb{Z}_p$$

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

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

Q: Is this statistically secure for two messages?



# Stateless Encryption with Multiple Messages

Ex:

$$\mathbf{M}_\lambda = \mathbf{C}_\lambda = \mathbb{Z}_p \text{ (} p \text{ a prime of size } 2^\lambda \text{)}$$

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

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

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

Attack:

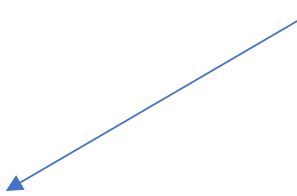
- $m_0^{(0)}=m_0^{(1)}=0, \quad m_1^{(0)}=0, m_1^{(1)}=1$
- $A(c^{(0)}, c^{(1)}) = 1$  iff  $c^{(0)}=c^{(1)}$
- Advantage: 1 (non-negligible)

# Example

Ex:

$$\begin{aligned} M_\lambda &= \mathbb{Z}_p \text{ (} p \text{ a prime of size } 2^\lambda \text{)} \\ C_\lambda &= \mathbb{Z}_p^2 \\ K_\lambda &= \mathbb{Z}_p^2 \end{aligned}$$

Random in  $\mathbb{Z}_p$


$$\begin{aligned} \text{Enc}( (a,b), m ) &= (r, (ar+b) + m ) \\ \text{Dec}( (a,b), (r,c) ) &= c - (ar+b) \end{aligned}$$

Q: Is this statistically secure for two messages?

# **(d+1)**-time Secure Encryption

Ex:

$$M_\lambda = \mathbb{Z}_p \text{ (} p \text{ a prime of size } 2^\lambda \text{)}$$

$$C_\lambda = \mathbb{Z}_p^2$$

$$K_\lambda = \{\text{degree } d \text{ polynomials over } \mathbb{Z}_p\}$$

$$\text{Enc}(P, m) = (r, P(r) + m)$$

$$\text{Dec}(P, (r, c)) = c - P(r)$$

**Theorem:** This scheme is secure for up to **d+1** messages

# Proof

What is the distribution  $(\text{Enc}(K_\lambda, m^{(i)}))_{i \in [d+1]}$  ?

- First, fix  $(r^{(i)})_{i \in [d+1]}$

**Claim:** If the  $r^{(i)}$  are distinct ( $r^{(i)} \neq r^{(j)}$  for any  $i \neq j$ ), then

$$(P(r^{(i)}))_{i \in [d+1]} \stackrel{d}{=} \mathbb{Z}_p^{d+1}$$

**Corollary:** If the  $r^{(i)}$  are distinct ( $r^{(i)} \neq r^{(j)}$  for any  $i \neq j$ ), then

$$(P(r^{(i)}) + m^{(i)})_{i \in [d+1]} \stackrel{d}{=} \mathbb{Z}_p^{d+1}$$

# Proof

Proof of Claim:

- Fix distinct  $(\mathbf{r}^{(i)})_{i \in [d+1]}$
- Fix tuple  $(\mathbf{y}^{(i)})_{i \in [d+1]}$
- Exactly one  $\mathbf{P}$  such that  $\mathbf{P}(\mathbf{r}^{(i)}) = \mathbf{y}^{(i)}$  for all  $i$
- Total number of polynomials:  $p^{d+1}$
- $\Pr[ (\mathbf{P}(\mathbf{r}^{(i)}))_{i \in [d+1]} = (\mathbf{y}^{(i)})_{i \in [d+1]} ] = 1/p^{d+1}$
- Therefore  $(\mathbf{P}(\mathbf{r}^{(i)}))_{i \in [d+1]} \stackrel{d}{=} \mathbb{Z}_p^{d+1}$

# Proof

What is the distribution  $(\text{Enc}(K_\lambda, m^{(i)}))_{i \in [d+1]}$  ?

- First, fix  $(r^{(i)})_{i \in [d+1]}$

**Claim:** If the  $r^{(i)}$  are distinct ( $r^{(i)} \neq r^{(j)}$  for any  $i \neq j$ ), then

$$(P(r^{(i)}))_{i \in [d+1]} \stackrel{d}{=} \mathbb{Z}_p^{d+1}$$

**Corollary:** If the  $r^{(i)}$  are distinct ( $r^{(i)} \neq r^{(j)}$  for any  $i \neq j$ ), then

$$(P(r^{(i)}) + m^{(i)})_{i \in [d+1]} \stackrel{d}{=} \mathbb{Z}_p^{d+1}$$

# Proof

**Lemma:  $\Delta(\mathcal{D}_1, \mathcal{D}_2) \leq \Pr[\text{bad}|\mathcal{D}_1] + \Pr[\text{bad}|\mathcal{D}_2] + \Delta(\mathcal{D}_{1,\text{good}}, \mathcal{D}_{2,\text{good}})$**

Where:

- **“bad”** is some event
- **$\Pr[\text{bad}|\mathcal{D}_b]$**  is probability **“bad”** when sampling from  **$\mathcal{D}_b$**
- **$\mathcal{D}_{b,\text{good}}$**  is the distribution  **$\mathcal{D}_b$**  conditioned on **not “bad”**

# Proof of Lemma

$$\begin{aligned}\Delta(D_1, D_2) &= \sum_x | \Pr[D_1=x] - \Pr[D_2=x] | \\ &= \sum_{x:\text{bad}} | \Pr[D_1=x] - \Pr[D_2=x] | \\ &\quad + \sum_{x:\text{good}} | \Pr[D_1=x] - \Pr[D_2=x] | \\ &\leq \sum_{x:\text{bad}} | \Pr[D_1=x] | + \sum_{x:\text{bad}} | \Pr[D_2=x] | \\ &\quad + \sum_{x:\text{good}} | \Pr[D_1=x] - \Pr[D_2=x] | \\ &\leq \Pr[\text{bad}|D_1] + \Pr[\text{bad}|D_2] + \Delta(D_{1,\text{good}}, D_{2,\text{good}})\end{aligned}$$



# Back to Security Proof

Goal: bound  $\Delta( (P(r^{(i)})+m^{(i)})_{i \in [d+1]}, \mathbb{Z}_p^{d+1} )$

Define “**bad**” to be that the  $r^{(i)}$  are not distinct

- Conditioned on “**good**”,  $\Delta=0$
- So using previous lemma  $\Delta \leq 2\Pr[\mathbf{bad}]$

Lemma:  $\Pr[\mathbf{bad}] \leq (d+1)^2/2p$

# Proof

**Lemma:  $\Pr[\text{bad}] \leq (d+1)^2/p$**

$$\begin{aligned}\Pr[\text{bad}] &= \Pr[r^{(1)}=r^{(2)} \text{ or } r^{(1)}=r^{(3)} \text{ or } \dots \text{ or } r^{(1)}=r^{(d+1)} \\ &\quad \text{or } r^{(2)}=r^{(3)} \text{ or } \dots ] \\ &\leq \Pr[r^{(1)}=r^{(2)}] + \Pr[r^{(1)}=r^{(3)}] + \dots + \Pr[r^{(1)}=r^{(d+1)}] \\ &\quad + \Pr[r^{(2)}=r^{(3)}] + \dots \quad (\text{Union Bound}) \\ &= (1/p) \binom{d+1}{2} \\ &\leq (d+1)^2/2p\end{aligned}$$

# Back to Security Proof

Goal: bound  $\Delta( (P(r^{(i)})+m^{(i)})_{i \in [d+1]}, \mathbb{Z}_p^{d+1} )$

Define “**bad**” to be that the  $r^{(i)}$  are not distinct

- Conditioned on “**good**”,  $\Delta=0$
- So using previous lemma  $\Delta \leq \Pr[\mathbf{bad}]$

Lemma:  $\Pr[\mathbf{bad}] \leq (d+1)^2/2p$

- So  $\Delta \leq (d+1)^2/p$

Finishing up the proof

$$\begin{aligned} & \Delta[ (\text{Enc}(K_\lambda, m_0^{(i)}))_{i \in [n]}, (\text{Enc}(K_\lambda, m_1^{(i)}))_{i \in [n]} ] \\ & \leq \Delta[ (\text{Enc}(K_\lambda, m_0^{(i)}))_{i \in [n]}, \mathbb{Z}_p^{d+1} ] \\ & \quad + \Delta[ (\text{Enc}(K_\lambda, m_1^{(i)}))_{i \in [n]}, \mathbb{Z}_p^{d+1} ] \\ & \leq 2(d+1)^2/p \leq 2(d+1)^2/2^\lambda \end{aligned}$$



negligible

# Summary

Stateful encryption is hard to manage

Stateless encryption cannot be perfectly secure for multiple messages

Therefore, use statistical security

Unfortunately, for our example, total number of messages bounded by key length

- Really want unbounded number of messages

# Next Time

Bound on message length/number of messages necessary for our security definitions

Computational security: security against computationally bounded adversaries

- Allows for keys that are very small (e.g. 128 bits)
- Can encrypt arbitrary number of messages of arbitrary length
- However, cannot prove security unconditionally