

Security of OTP (Secret Sharing)

HW1 will be released today

BIG PICTURE OF SECURITY DEFS

- ▶ Define 2 libraries with a common interface but different internals
- ▶ Adv is a calling program (trying to behave differently in presence of 2 libraries)
- ▶ If no Adv behaves differently then "interface leaks no information about internal differences"

Ex:

$Z_{otp-real}$
$ctxt(m):$
$k \leftarrow \{0,1\}^\lambda$
ret $k \oplus m$

$Z_{otp-rand}$
$ctxt(m):$
$c \leftarrow \{0,1\}^\lambda$
ret c

Claim: (last time)

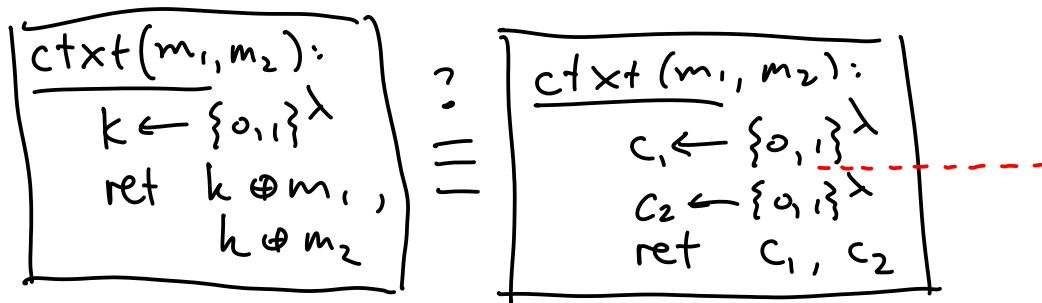
$$\text{for all } A, \quad \Pr[A \diamond \overbrace{Z_{otp-real}}^{A \text{ linked to the library}} \Rightarrow 1] \\ = \Pr[A \diamond \underbrace{Z_{otp-rand}}_{\text{event that linked program outputs 1}} \Rightarrow 1]$$

(shorthand: $Z_{otp-real} \equiv Z_{otp-rand}$)
they are interchangeable

Notes:

- Adv can choose m
- Adv can "know everything" about 2 libraries
- only thing Adv doesn't "know" is
 - which L he is currently linked to
 - values of privately scoped variables
- Kerckhoff's principle (1883) = Assume Adv knows everything about your system except the key

Q: What happens if Adv sees 2 ctxts under same k ?



Claim: \neq , need to show A that behaves differently when linked to left/right L

A:

$(c_1, c_2) \leftarrow \text{ctxt}(0^\lambda, 1^\lambda)$
// if in "left world", $c_1 = k$, $c_2 = k \oplus 1^\lambda$
return $c_1 \oplus c_2 \stackrel{?}{=} 1^\lambda$

When $A \diamond L_{\text{left}}$: $c_1 = k \oplus 0^\lambda = k$
 $c_2 = k \oplus 1^\lambda$

$\Rightarrow c_1 \oplus c_2 = 1^\lambda \Rightarrow A$ returns true, always

When $A \diamond \mathcal{I}_{\text{right}}$: c_1, c_2 uniformly chosen

$$\begin{aligned} & \Pr[c_1 \oplus c_2 = 1^\lambda] \\ &= \Pr[\underbrace{c_2}_{\text{fixed at time of breakpoint}} = \underbrace{1^\lambda \oplus c_1}_{\text{about to be chosen uniformly}}] \\ &= 1/2^\lambda \neq 1 \end{aligned}$$

BIG PICTURE of breaking security:

- ▶ write down code of Adv (calling program)
- ▶ show that $\Pr[A \diamond \mathcal{I}_{\text{left}} \Rightarrow 1] \neq \Pr[A \diamond \mathcal{I}_{\text{right}} \Rightarrow 1]$

Better definition:

(prev def for OTP is very specific to OTP)

Def: enc. scheme Σ has one-time secrecy if

$$\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$$

$$\boxed{\begin{array}{l} \text{Query}(m_L, m_R): \\ k \leftarrow \Sigma.\text{KeyGen} \\ \text{ret } \Sigma.\text{Enc}(k, m_L) \end{array}} \equiv \boxed{\begin{array}{l} \text{Query}(m_L, m_R): \\ k \leftarrow \Sigma.\text{KeyGen} \\ \text{ret } \Sigma.\text{Enc}(k, m_R) \end{array}}$$

$\mathcal{L}_{\text{ots-L}} \quad \mathcal{L}_{\text{ots-R}}$

only diff between \mathcal{L} 's is choice of ptxt

\Rightarrow "seeing 1 ctxt leaks no info about choice of ptxt"

Claim: OTP has one-time secrecy [see PDF]

BIG PICTURE of proving security:

- ▶ proving security means showing $\mathcal{L}_1 \equiv \mathcal{L}_2$
- ▶ start w/ \mathcal{L}_1 and make sequence of small modifications
- ▶ justify why each modification has no effect to calling program
- ▶ end at \mathcal{L}_2