CPA Security, Randomized Encryption

Why must encryption be randomized?

How can encryption even be randomized?

Ex: Is a PRP also a CPA-secure Enc scheme? NO

Let's distinguish between: (assume F is good PRP)

$k \leftarrow \{0,1\}^\lambda$	$k \leftarrow \{0,1\}^\lambda$
CHALLENGE(m_L, m_R):	CHALLENGE(m_L, m_R):
ret $F(k, m_L)$	ret $F(k, m_R)$

Idea: Can tell whether same ptxt encrypted twice ✳

A:

$c_1 =$ CHALLENGE$(0^\lambda, 0^\lambda)$

$c_2 =$ CHALLENGE$(0^\lambda, 1^\lambda)$

return $c_1 \stackrel{?}{=} c_2$

in presence of L library, c_1 & c_2 encrypt same ptxt $\Rightarrow c_1 = c_2$

in presence of R library, c_1 & c_2 encrypt different ptxts $\Rightarrow c_1 \neq c_2$

$\Rightarrow$ advantage $= 1$

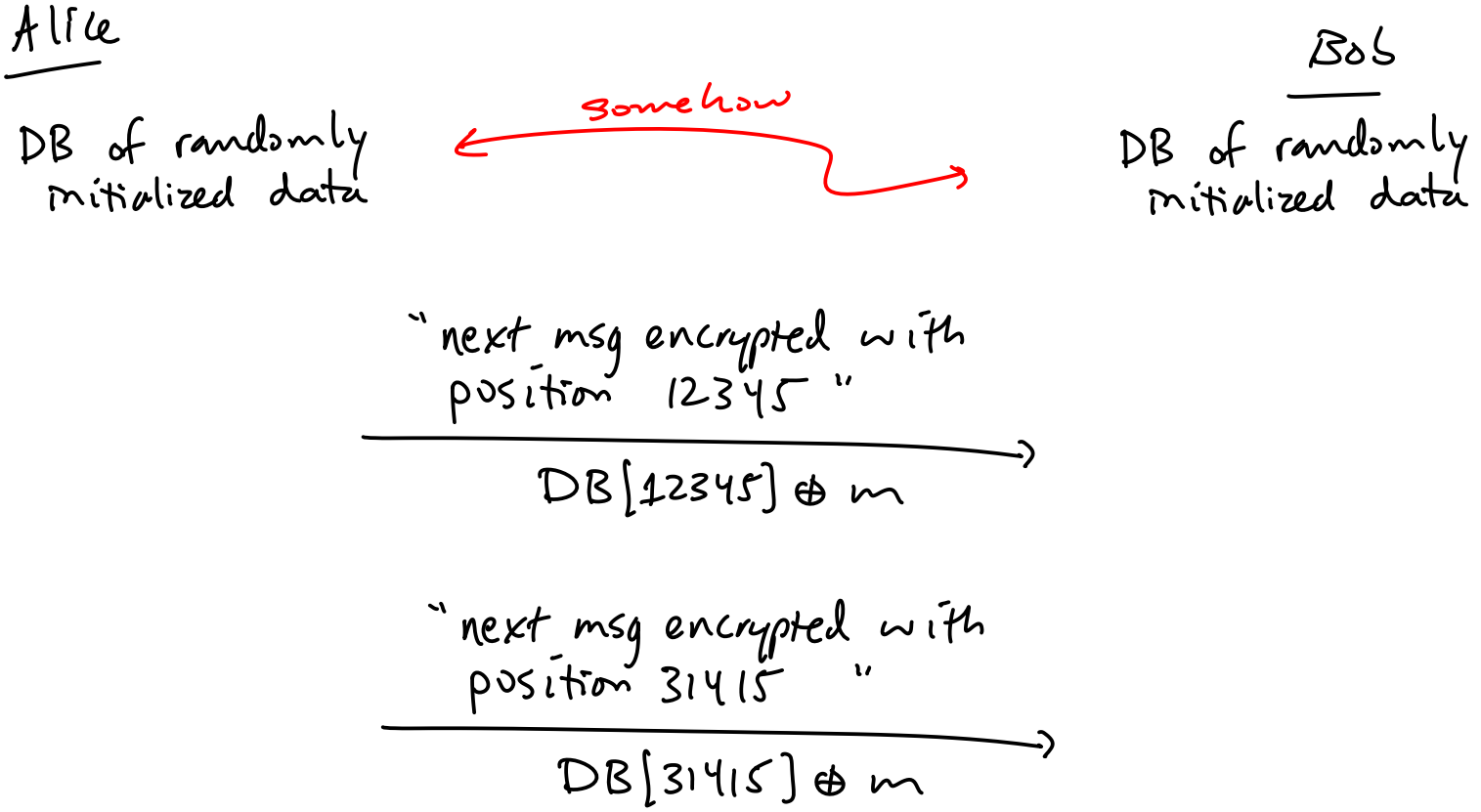
PRP vs CPA-encryption:

- both are invertible if you have the key (a bit string)
- without the key, they're not invertible, outputs "look random"

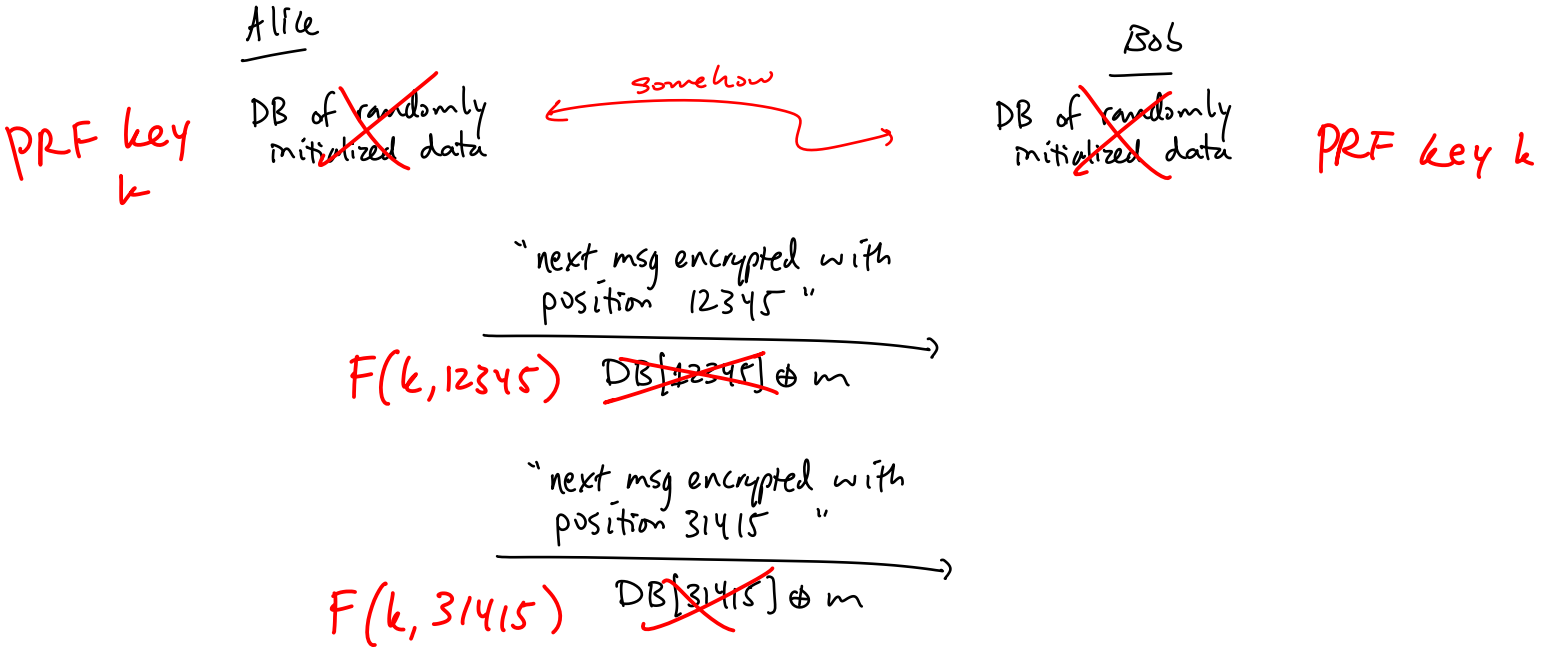
✳ But: PRF/PRP outputs on distinct inputs look random
Enc outputs on all calls (even same input) look #

How can it be randomized?

many possible outputs of Enc(k,m)
ALL of them can be decrypted to m



secure, correct, many ways to Enc same ptxt



Security Proof:

for $Enc(k,m) = (r,\ F(k,r) \oplus m)$

outputs of Enc look random

Why?

F(k,r) looks random, so it's like OTP ✡

why does F(k,r) look random?

PRF outputs look random if its inputs are distinct ✡

why are PRF inputs distinct?

r-values (inputs to PRF) are long strings chosen randomly, so Pr[repeat] is negligible ✡

ok.

Next time: Attacks:

Secure

Enc(k,m):
  $r \leftarrow \{0,1\}^\lambda$
  return $(r,\ F(k,r) \oplus m)$

Insecure

Enc(k,m):
  $r \leftarrow \{0,1\}^\lambda$
  return $(r,\ F(k,m) \oplus r)$