

# Cryptography: HW4

Trevor Bramwell

Professor Mike Rosulek

February 5, 2015

**1a** Consider a variant of the CPA definition for CBC encryption in which an adversary is able to see the next IV before choosing a plaintext.

$k \leftarrow \Sigma.\text{KeyGen}$  <u>CHALLENGE</u> ( $m_L, m_R$ ): return null if $ m_L  \neq  m_R $ $c_0 c_1 \dots c_\ell := \Sigma.\text{ENC}(k, m_L)$ return $c_0 c_1 \dots c_\ell$  <u>VEC</u> : return $iv_{next}$	$k \leftarrow \Sigma.\text{KeyGen}$  <u>CHALLENGE</u> ( $m_L, m_R$ ): return null if $ m_L  \neq  m_R $ $c_0 c_1 \dots c_\ell := \Sigma.\text{ENC}(k, m_R)$ return $c_0 c_1 \dots c_\ell$  <u>VEC</u> : return $iv_{next}$
--	--

**1b** This modification breaks CPA security for CBC encryption. This can be seen by the following distinguisher  $A$ .

<u>A</u> : $iv_1 \leftarrow \text{VEC}$ $iv_0 c a_1 \dots c a_\ell := \text{CHALLENGE}(m_1 m_2 \dots m_\ell)$ $iv_1 c b_1 \dots c b_\ell := \text{CHALLENGE}((iv_0 \oplus iv_1 \oplus m_1) m_2 \dots m_\ell)$ return $ca_{1-\ell} \stackrel{?}{=} cb_{1-\ell}$
--

Assume  $k$  does not change for each call to CHALLENGE. In the first call to CHALLENGE,  $F_k$  is called with  $(iv_0 \oplus m_1)$ . In the second call to

CHALLENGE, we know the  $iv$  ahead of time. This allows us to replace it with the previously used  $iv_0$ .

$$iv_0 \oplus m_1 = iv_1 \oplus (iv_0 \oplus iv_1 \oplus m_1)$$

This results in the same ciphertext being output twice. The bias for A is then:

$$\begin{aligned} \text{bias}(A, \mathcal{L}_{\text{cpa-real}}^{CBC}, \mathcal{L}_{\text{cpa-rand}}^{CBC}) &= |\Pr[A \diamond \mathcal{L}_{\text{cpa-real}}^{CBC} \text{ outputs } 1] - \Pr[A \diamond \mathcal{L}_{\text{cpa-rand}}^{CBC} \text{ outputs } 1]| \\ &= |1 - \frac{1}{2^n}| \\ &= \text{Non-negligible} \end{aligned}$$

**2** Consider a variant of CPA\$ security in which the adversary is allowed to *choose* but not *re-use* IVs. We know "standard" CBC encryption (with a random IV) is CPA\$-secure.

(a)

$k \leftarrow \{0, 1\}^{2n}$   
 $iv \leftarrow \text{Adversary Choice}$

---

**CHALLENGE**( $m_1 \cdots m_\ell$ ):  
 $iv' \leftarrow F(k_{\text{left}}, iv)$   
 $c := \text{CBC encryption of } m_1 \cdots m_\ell$   
     under key  $k_{\text{right}}$   
     using initialization vector  $iv'$   
 return  $c_1 \cdots c_\ell$

(b)

$k_{left} \leftarrow \{0, 1\}^n$   
 $k_{right} \leftarrow \{0, 1\}^n$   
 $iv \leftarrow \text{Adversary Choice}$

---

CHALLENGE( $m_1 \cdots m_\ell$ ):  
 $iv' \leftarrow F(k_{left}, iv)$   
 $c := \text{CBC encryption of } m_1 \cdots m_\ell$   
under key  $k_{right}$   
using initialization vector  $iv'$   
return  $c_1 \cdots c_\ell$

(c)

$k_{right} \leftarrow \{0, 1\}^n$   
 $iv \leftarrow \text{Adversary Choice}$

---

CHALLENGE( $m_1 \cdots m_\ell$ ):  
 $iv' \leftarrow \text{QUERY}(iv)$   
 $c := \text{CBC encryption of } m_1 \cdots m_\ell$   
under key  $k_{right}$   
using initialization vector  $iv'$   
return  $c_1 \cdots c_\ell$

$k_{left} \leftarrow \{0, 1\}^n$

---

QUERY( $iv$ ):  
return  $F(k_{left}, iv)$

◇

(d)

$k_{right} \leftarrow \{0, 1\}^n$   
 $iv \leftarrow \text{Adversary Choice}$

---

CHALLENGE( $m_1 \cdots m_\ell$ ):  
 $iv' \leftarrow \text{QUERY}(iv)$   
 $c := \text{CBC encryption of } m_1 \cdots m_\ell$   
under key  $k_{right}$   
using initialization vector  $iv'$   
return  $c_1 \cdots c_\ell$

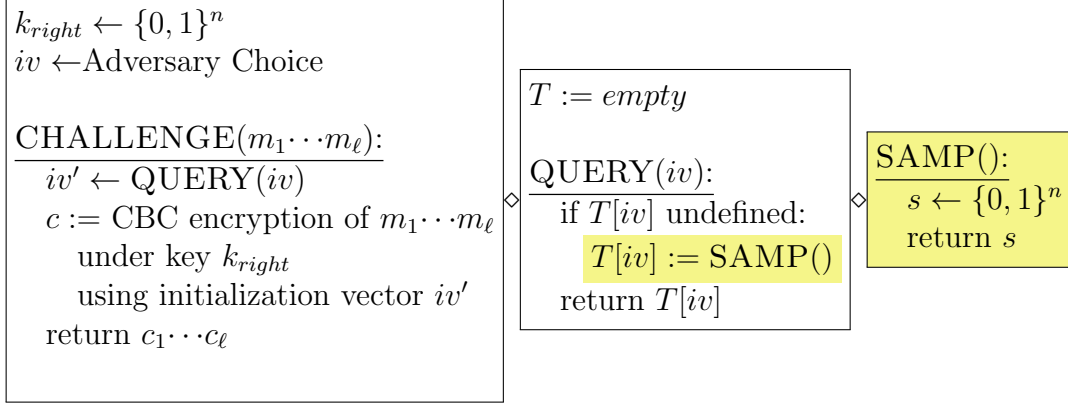
$T := \text{empty}$

---

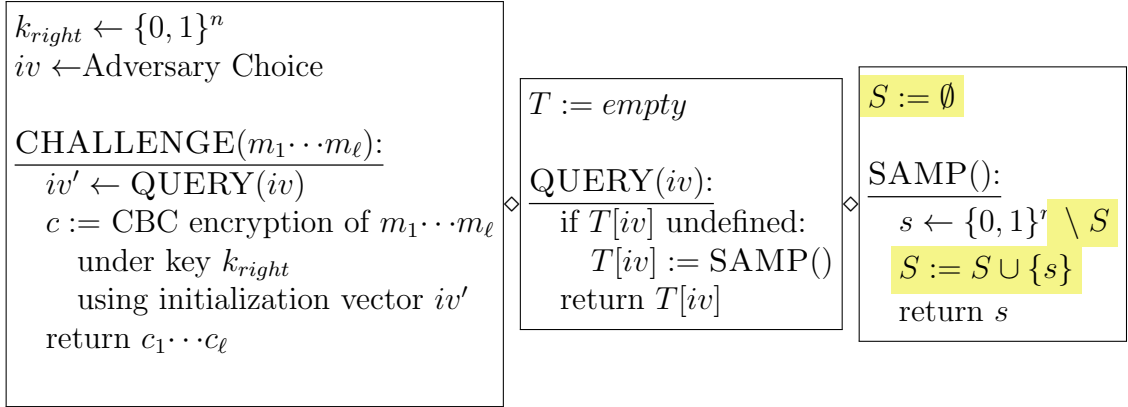
QUERY( $iv$ ):  
if  $T[iv]$  undefined:  
 $T[iv] \leftarrow \{0, 1\}^n$   
return  $T[iv]$

◇

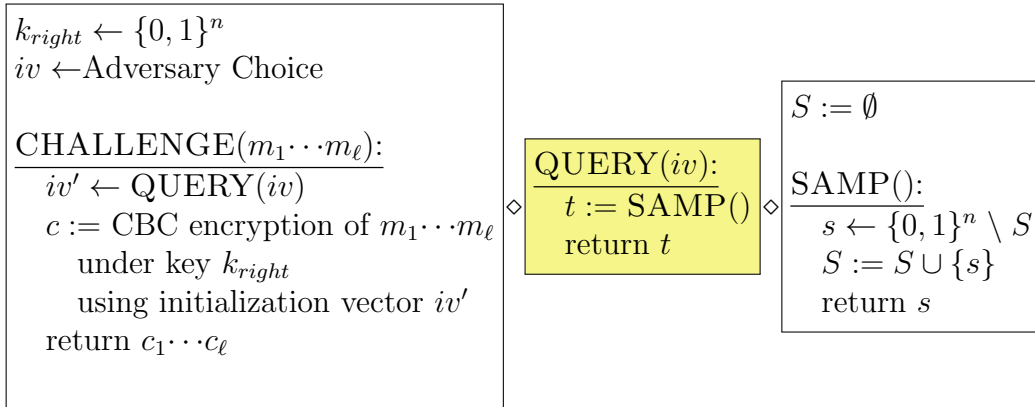
(e)



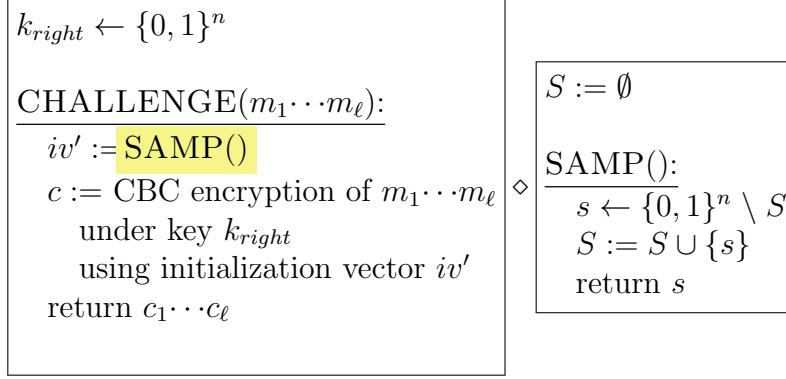
(f)



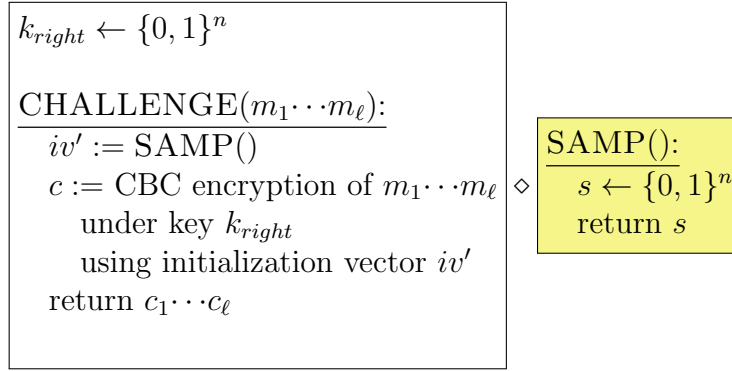
(g)



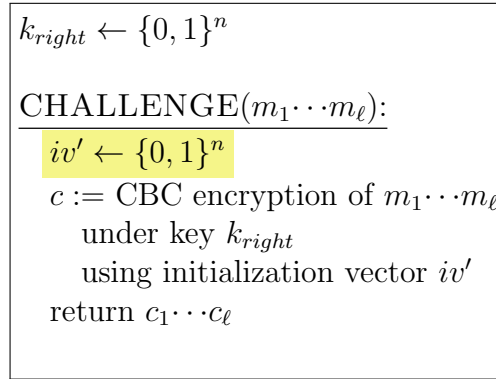
(h)



(i)



(j)



The CPA\$ security relies on  $iv'$ 's being pseudorandom. Since each  $iv'$  is a result from the call to the PRF, the outputs will only be pseudorandom if

the inputs to the PRF are distinct. Because this variant does not allow for  $ivs$  to be repeated, they are guaranteed to be distinct.

Justification of steps:

- (a) This is the inlining of the encryption scheme to the CPA\$ security variant.
- (a) $\Rightarrow$ (b)  $k$  is split into it's two halves. No affect on the program output distribution.
- (b) $\Rightarrow$ (c) The PRF call is factored out into it's own function along with  $k_{left}$ . No affect on the program.
- (c) $\Rightarrow$ (d)  $\mathcal{L}_{\text{prf-real}}^F$  replaced with  $\mathcal{L}_{\text{prf-rand}}^F$ . Program output distribution changes by a negligible amount.
- (d) $\Rightarrow$ (e) Factoring out the sampling of random blocks. No effect on program.
- (e) $\Rightarrow$ (f) Sampling without replacement instead of with replacement. Negligible effect to output distribution.
- (g) Because  $iv$  are not allowed to be reused, QUERY is only called with a non-defined  $T[iv]$ . Removing the if-statement does not affect the output distribution.
- (h)  $iv$  is now never used, so it is removed from the definition. QUERY merely calls SAMP, so this redundancy is removed.
- (h) $\Rightarrow$ (i) Sampling with replacement instead of sample without replacement. Negligible effect on program's output distribution.
- (i) $\Rightarrow$ (j) Call to SAMP inlined. No effect on program.

Program (j) now has pseudorandom ciphertexts under chosen plaintext and *chosen-but-never-repeated-IV* attacks.

**3** A distinguisher showing  $\Sigma^2$  does not have CCA security is as follows:

A:  
 $c_1, c_2 = \text{CHALLENGE}(m_L, m_R)$   
 $m = \text{DEC}(k, (c_2, c_1))$   
 return  $m_{right} || m_{left} \stackrel{?}{=} m_L$

By switching  $c_1$  and  $c_2$  when calling DEC, the output will be reversed.  
 The bias of  $A$  is as follows:

$$\begin{aligned}
 \text{bias}(A, \mathcal{L}_{\text{cca-real}}^{\Sigma^2}, \mathcal{L}_{\text{cca-rand}}^{\Sigma^2}) &= |\Pr[A \diamond \mathcal{L}_{\text{cca-real}}^{\Sigma^2} \text{ outputs } 1] - \Pr[A \diamond \mathcal{L}_{\text{cca-rand}}^{\Sigma^2} \text{ outputs } 1]| \\
 &= |1 - \frac{1}{2^{2n}}| \\
 &= \text{Non-negligible}
 \end{aligned}$$