CS 6260
Applied Cryptography

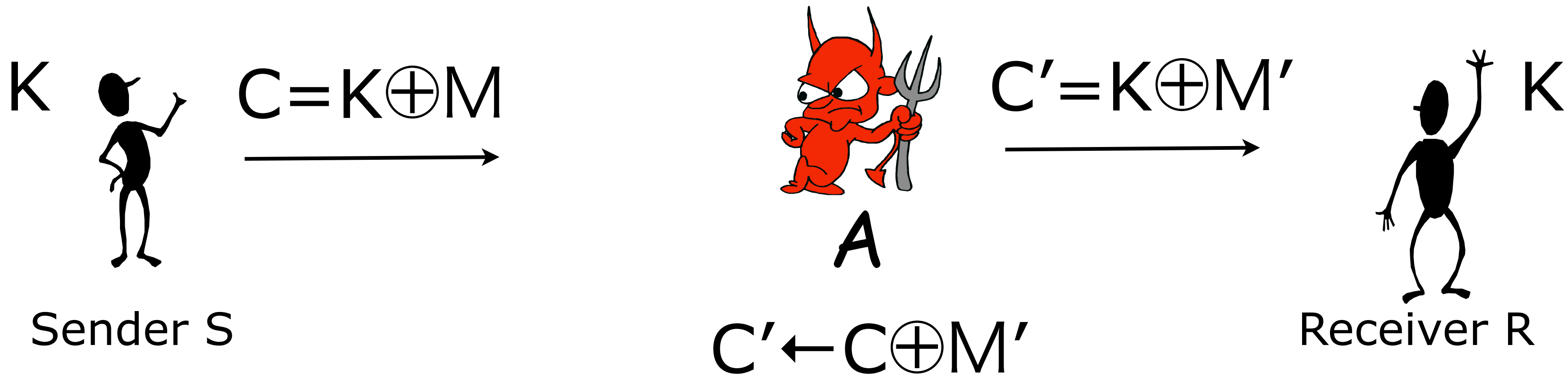
Message Authentication Codes (MACs).

New cryptographic goals

- Data privacy is not the only important cryptographic goal
- It is also important that a receiver is assured that the data it receives has come from the sender and has not been modified on the way (and detect if it is not the case)
- The goals are data authenticity and integrity

Encryption solves data privacy, not authenticity/integrity

- Recall OneTimePad: $E(K,M)=K\oplus M$



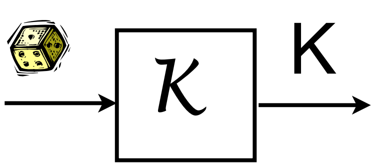
R gets $M\oplus M'$ instead of M

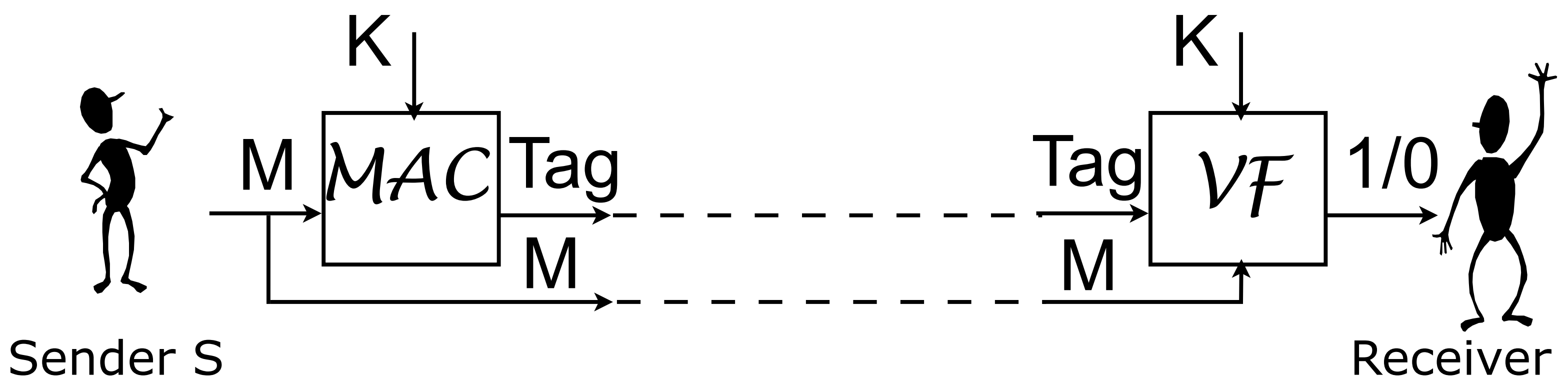
Message Authentication Code (MAC)

- is the primitive for the goal of data authenticity in the symmetric-key setting

$\Pi=(\mathcal{K},\mathcal{MAC},\mathcal{VF})$

MsgSp-message space





It is required that for every M∈MsgSp and every K that can be output by

$\mathcal{K}$, $\mathcal{VF}$(K,M,$\mathcal{MAC}$(K,M))=1

Message Authentication Code (MAC)

- If the key-generation algorithm simply picks a random string from some KeySp, then KeySp describes $\mathcal{K}$
- If the $\mathcal{MAC}$ algorithm is deterministic, then the verification algorithm $\mathcal{VF}$ does not have to be defined as it simply re-computes the MAC by invoking the $\mathcal{MAC}$ algorithm on the given message M and accepts iff the result is equal to its input TAG.

Towards a security definition for MACs

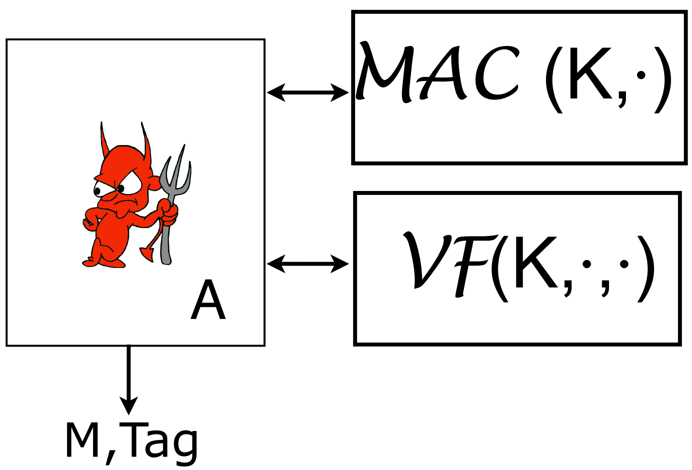
- We imagine that an adversary can see some number of message plus tag pairs
- As usual, it is necessary but not sufficient to require that no adversary can compute the secret key
- Right now we will not be concerned with *replay attacks*
- We don’t want an adversary to be able to compute a new message and a tag such that the receiver accepts (outputs 1).

Security definition for MACs

Fix $\Pi=(\mathcal{K},\text{MAC},\text{VF})$

Run $\mathcal{K}$ to get K

For an adversary A consider an experiment $\mathbf{Exp}_{\Pi}^{\text{uf-cma}}(A)$



Return 1 iff VF(K,M,Tag)=1 and M was not queried to the MAC oracle

The uf-cma advantage of A is defined as

$$\mathbf{Adv}_{\Pi}^{\text{uf-cma}}(A) \quad = \quad \Pr\left[\mathbf{Exp}_{\Pi}^{\text{uf-cma}}(A) = 1\right]$$

UF-CMA security is defined the usual way.

Examples

We fix a PRF F: $\{0,1\}^k \times \{0,1\}^\ell \rightarrow \{0,1\}^L$

$$\Pi_1 = (\mathcal{K}, \mathrm{MAC})$$

algorithm $\mathrm{MAC}_K(M)$
  if $(|M| \bmod \ell \neq 0$ or $|M| = 0)$ **then** **return** $\perp$
  Break M into ℓ bit blocks $M = M[1] \ldots M[n]$
  for $i = 1, \ldots, n$ **do** $y_i \leftarrow F_K(M[i])$
  $Tag \leftarrow y_1 \oplus \cdots \oplus y_n$
  return Tag

It is easy to construct A_1 s.t. $\mathbf{Adv}^{\text{uf-cma}}_{\Pi_1}(A_1) = 1$

Examples

We fix a PRF F: $\{0,1\}^k \times \{0,1\}^\ell \rightarrow \{0,1\}^L$

$\mathbf{\Pi}_2 = (\mathcal{K}, \mathrm{MAC})$

algorithm $\mathrm{MAC}_K(M)$
  $l \leftarrow \ell - m$
  if $(|M| \bmod l \neq 0$ or $|M| = 0$ or $|M|/l \geq 2^m)$ **then** **return** $\perp$
  Break M into l bit blocks $M = M[1] \ldots M[n]$
  for $i = 1, \ldots, n$ **do** $y_i \leftarrow F_K([i]_m \,\|\, M[i])$
  $Tag \leftarrow y_1 \oplus \cdots \oplus y_n$
  return Tag

Adversary $A_2^{\mathrm{MAC}_K(\cdot)}$
  Let a_1, b_1 be distinct, $\ell - m$ bit strings
  Let a_2, b_2 be distinct $\ell - m$ bit strings
  $Tag_1 \leftarrow \mathrm{MAC}_K(a_1 a_2)$; $Tag_2 \leftarrow \mathrm{MAC}_K(a_1 b_2)$; $Tag_3 \leftarrow \mathrm{MAC}_K(b_1 a_2)$
  $Tag \leftarrow Tag_1 \oplus Tag_2 \oplus Tag_3$
  return(b1b2,Tag)

$$\mathbf{Adv}_{\Pi_2}^{\text{uf-cma}}(A_2) = 1$$

Note

- We broke the MAC schemes without breaking the underlying function families (they are secure PRFs).
- The weaknesses were in the schemes, not the tools

A PRF as a MAC

Fix a function family $F\colon \text{Keys} \times D \rightarrow \{0,1\}^{\tau}$

Consider a MAC $\Pi = (\mathcal{K}, \mathrm{MAC})$

algorithm $\mathcal{K}$	**algorithm** $\mathrm{MAC}_K(M)$
$K \xleftarrow{\$} \text{Keys}$	**if** $(M \notin D)$ **then return** $\perp$
return K	$Tag \leftarrow F_K(M)$
	Return Tag

<u>Theorem</u>. Let A be an adversary attacking Π making qma MAC oracle queries of total length mma and qva verification oracle queries of total length mva and running time *ta*. Then there exists an adversary B attacking F as a PRF such that

$$\mathbf{Adv}_{\Pi}^{\text{uf-cma}}(A) \quad \leq \quad \mathbf{Adv}_{F}^{\text{prf}}(B) + \frac{1}{2^{\tau}}$$

and B makes qma+qva+1 queries and runs the time *ta+qva tc,* where tc is the time to compare strings of the tag length. The total length of the queries is at most mma+mva+the largest length of strings in D.

- **Proof.**

Adversary B^f

$d \leftarrow 0$; $S \leftarrow \emptyset$

Run A

When A asks its signing oracle some query M:

Answer $f(M)$ to A ; $S \leftarrow S \cup \{M\}$

When A asks its verification oracle some query (M, Tag):

if $f(M) = Tag$ **then**

answer 1 to A

else answer 0 to A

When A outputs forgery (M',t')

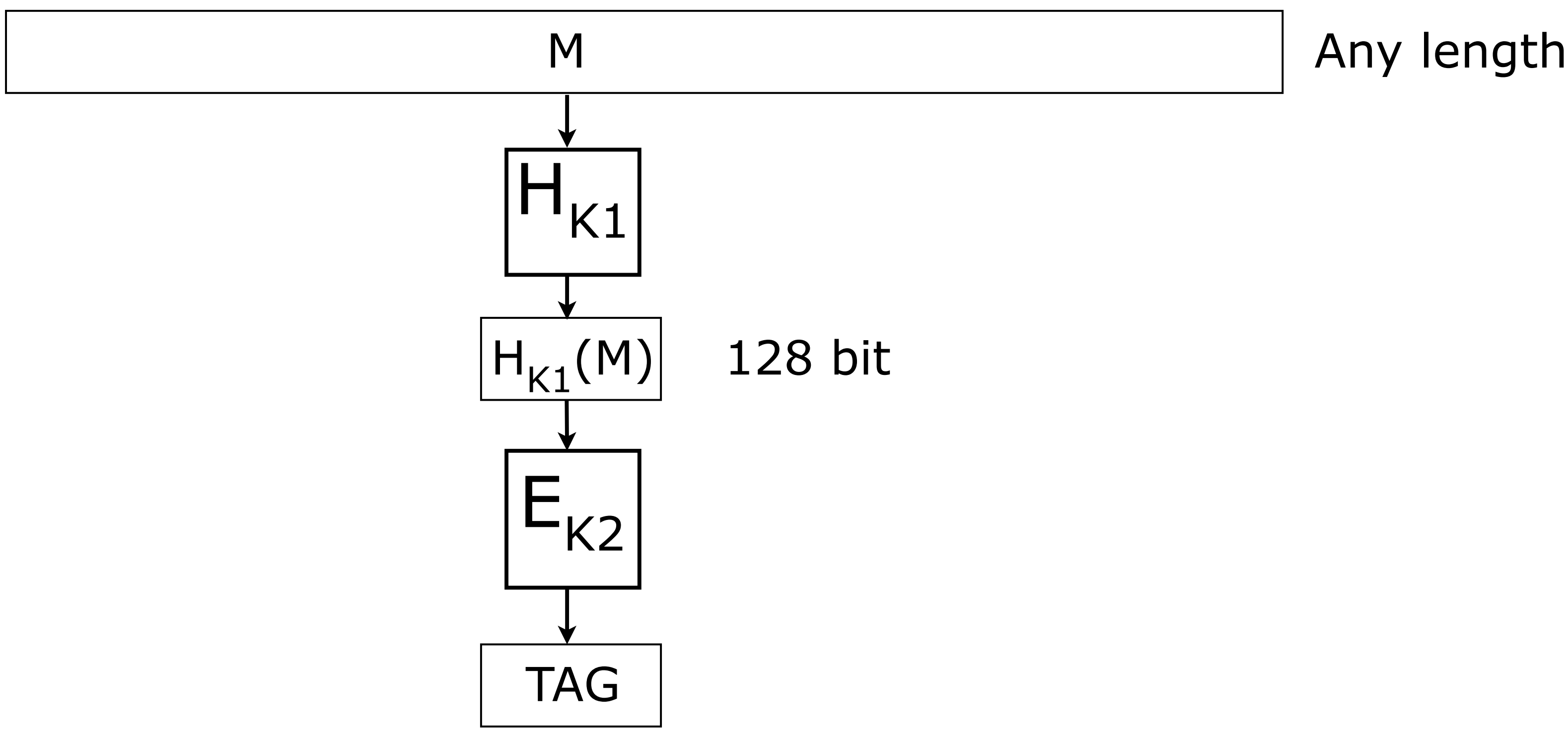
If f(m')=t' then return 1

otherwise return 0

$$\Pr\left[\mathbf{Exp}_F^{\text{prf-1}}(B) = 1\right] = \mathbf{Adv}_{\Pi}^{\text{uf-cma}}(A)$$

$$\Pr\left[\mathbf{Exp}_F^{\text{prf-0}}(B) = 1\right] \leq \frac{1}{2^{\tau}}$$

- Any PRF makes a good MAC
- Are we done?
- Efficient PRFs (e.g. block ciphers) has short fixed input length
- We want it to work for arbitrary-length messages
- What if we hash a message first before applying the block cipher:



What H will be good?

- Definition. [universal function family]
  Let H: KeySp(H)$\times$Dom(H)$\rightarrow$Ran(H) be a function family. It is called universal if
  $\forall$ X,Y$\in$Dom(H) s.t. X$\neq$Y: $\Pr_K[H_K(X)=H_K(Y)]=1/|\mathrm{Ran}(H)|$

- "Matrix" Construction. Let KeySp(H) be a set of all n$\times$m matrices, where each element can be either 0 or 1. Let Dom(H)=$\{0,1\}^m$, Ran(H)=$\{0,1\}^n$.
  Define $H_K(X)=K\cdot X$ (where addition is mod 2)

- Claim. The above "matrix" function family is universal.

- The problem with the matrix construction is that the key is big.
- There are other efficient constructions of universal hash functions
- But will combining a universal hash and a PRF will really give us a secure MAC?
- Yes. And let’s prove it.

"Hash-and-PRF" MAC

- Construction. Let H: KeySp(H)×Dom(H)→Ran(H) and F: KeySp(F)×Ran(H)→Ran(F) be function families. Define a MAC HPRF=($\mathcal{K}$,$\mathcal{MAC}$,$\mathcal{VF}$) with MsgSp=Dom(H) as follows:
  - $\mathcal{K}$: K1 $\xleftarrow{\$}$ KeySp(H), K2 $\xleftarrow{\$}$ KeySp(F), Return K1||K2
  - $\mathcal{MAC}$(K1||K2,M): Tag←$F_{K2}(H_{K1}(M))$, Return Tag
  - $\mathcal{VF}$(K1||K2,M,Tag): If Tag=$F_{K2}(H_{K1}(M))$ then return 1, otherwise return 0

- Theorem. If F is PRF and H is universal, then HPRF is a secure MAC.
- Lemma. If F is PRF and H is universal then HPRF is PRF.
- Proof of the Theorem. Follows from the Lemma and the fact that any PRF is a secure MAC.
- Proof of the Lemma. We will prove that for any A there exists B with $t_B=O(t_A)$, $q_B=q_B$ s.t.

$$\mathbf{Adv}_{HPRF}^{prf}(A) \leq \mathbf{Adv}_{F}^{prf}(B) + \frac{q_A(q_A-1)}{2 \cdot |Ran(H)|}$$

Adversary B^f

$K1 \xleftarrow{\$} KeySp(H)$

Answer B's queries M with $f(H_{K1}(M))$

Output the same bit B outputs

Let g be a random function with domain Ran(H) and range Ran(F)

Let g' be a random function with domain Dom(H) and range Ran(F)

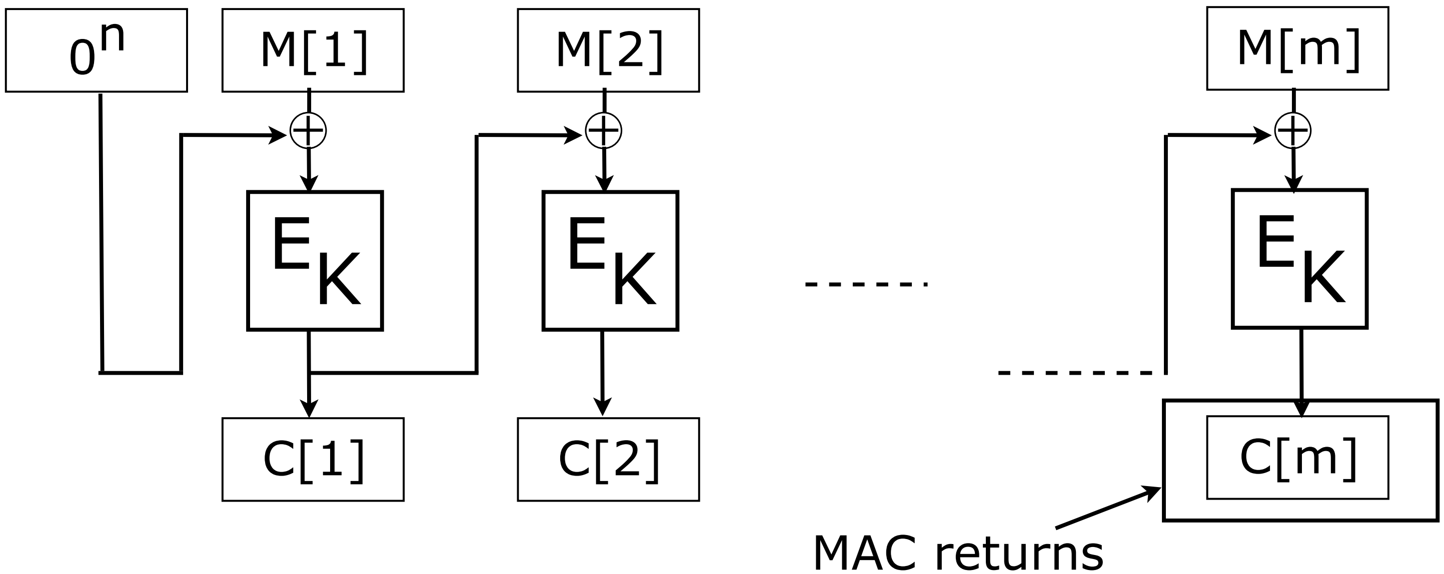
Let coll be an event when HK1(M)=HK1(M') for any two queries M,M' made by A

$$
\begin{aligned}
& \mathsf{Adv}_F^{\mathrm{prf}}(B) \\
= \; & \Pr\left[\mathbf{Exp}_F^{prf-1}(B)\right] - \Pr\left[\mathbf{Exp}_F^{prf-0}(B)\right] \\
= \; & \Pr\left[\mathbf{Exp}_{HPRF(H\circ F)}^{prf-1}(A)\right] - \Pr\left[\mathbf{Exp}_{H\circ g}^{prf-1}(A)\right] \\
= \; & \Pr\left[\mathbf{Exp}_{HPRF(H\circ F)}^{prf-1}(A)\right] - \Pr\left[\mathbf{Exp}_{g'}^{prf-1}(A)\right] + \Pr\left[\mathbf{Exp}_{g'}^{prf-1}(A)\right] - \Pr\left[\mathbf{Exp}_{H\circ g}^{prf-1}(A)\right] \\
= \; & \Pr\left[\mathbf{Exp}_{HPRF(H\circ F)}^{prf-1}(A)\right] - \Pr\left[\mathbf{Exp}_{HPRF(H\circ F)}^{prf-0}(A)\right] + \Pr\left[\mathbf{Exp}_{g'}^{prf-1}(A)\right] - \Pr\left[\mathbf{Exp}_{H\circ g}^{prf-1}(A)\right] \\
= \; & \mathsf{Adv}_{HPRF}^{\mathrm{prf}}(A) + \Pr\left[\mathbf{Exp}_{g'}^{prf-1}(A)\right] - \Pr\left[\mathbf{Exp}_{H\circ g}^{prf-1}(A)\right] \\
= \; & \mathsf{Adv}_{HPRF}^{\mathrm{prf}}(A) + \Pr\left[\mathbf{Exp}_{g'}^{prf-1}(A)\right] \\
- \; & \Pr\left[\mathbf{Exp}_{H\circ g}^{prf-1}(A) \mid coll\right] \cdot \Pr[\, coll \,] - \Pr\left[\mathbf{Exp}_{H\circ g}^{prf-1}(A) \mid \overline{coll}\right] \cdot \Pr[\, \overline{coll} \,] \\
\leq \; & \mathsf{Adv}_{HPRF}^{\mathrm{prf}}(A) + \Pr\left[\mathbf{Exp}_{g'}^{prf-1}(A)\right] - \Pr[\, coll \,] - \Pr\left[\mathbf{Exp}_{H\circ g}^{prf-1}(A) \mid \overline{coll}\right] \\
= \; & \mathsf{Adv}_{HPRF}^{\mathrm{prf}}(A) - \Pr[\, coll \,] = \mathsf{Adv}_{HPRF}^{\mathrm{prf}}(A) - \frac{q_A \cdot (q_A - 1)}{2Ran(H)}
\end{aligned}
$$

CBC-MAC

Let E:$\{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher. CBC-MAC=($\{0,1\}^k$,$\mathcal{MAC}$):

MsgSp=$\{0,1\}^{nm}$ for some m≥1.



<u>Theorem.</u> For any adversary A there exists an adversary B such that

$$\mathbf{Adv}^{uf-cma}_{CBC-MAC} \leq \mathbf{Adv}^{prp-cpa}_{E}(B) + \frac{m^2 q_A^2}{2^{n-1}}$$

where $q_B = q_A + 1, t_B = t_A$

Can we use a hash function as a building block?

- SHA1: $\{0,1\}^{< 2^{64}} \rightarrow \{0,1\}^{160}$
- Collision-resistant: hard to fund M,M' s.t. SHA1(M)=SHA1(M')
- Is it a good idea to use SHA1 as a MAC?
- What about:
  - $MAC_K(M)=SHA1(M||K)$?
  - $MAC_K(M)=SHA1(K||M)$?
  - $MAC_K(M)=SHA1(K||M||K)$?
- Cannot prove security for these constructions.
- Secure construction: HMAC
  - $HMAC_K(M)=SHA1(K \oplus c||SHA1(K \oplus d||M))$, where c,d are some constants

Can we get it all?

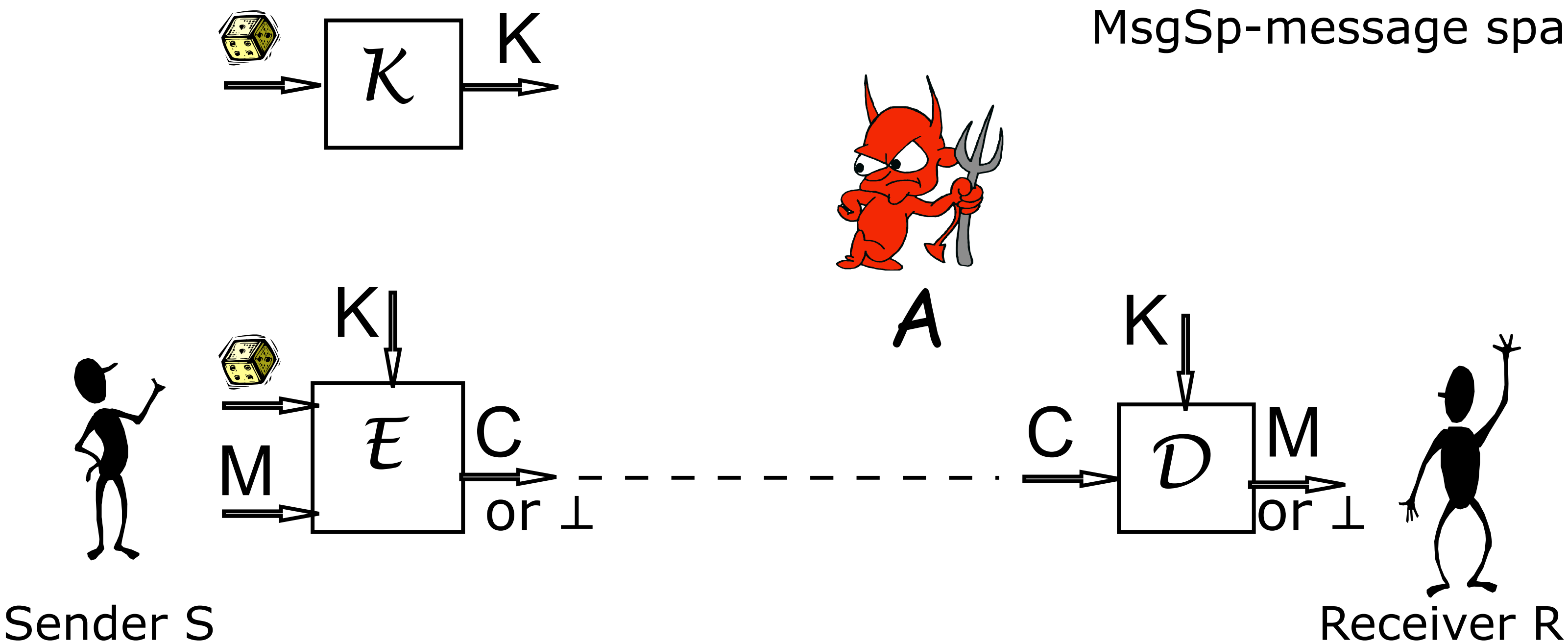
- We know how to achieve data privacy (IND-CPA security) and data authenticity/integrity (UF-CMA security) separately.
- Can we achieve the both goals at the same time (can we send messages securely s.t. a sender is assured in their authenticity/integrity)?
- Can we use the existing primitives: encryption schemes and MACs?

Recall: symmetric encryption scheme

A scheme SE is specified a key generation algorithm $\mathcal{K}$, an encryption algorithm $\mathcal{E}$, and a decryption algorithm $\mathcal{D}$.

$\text{SE}=(\mathcal{K},\mathcal{E},\mathcal{D})$

MsgSp-message space



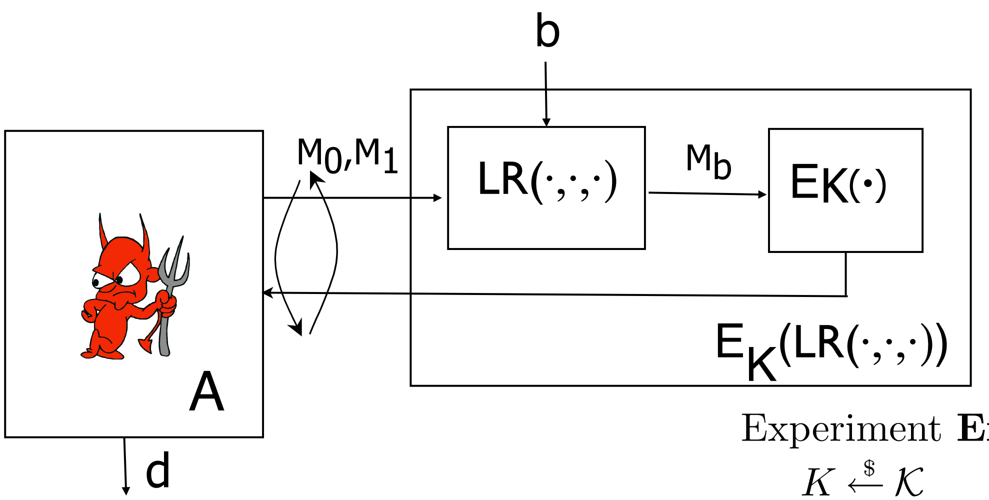
It is required that for every M∈MsgSp and every K that can be output by $\mathcal{K}$, $\mathcal{D}(K,\mathcal{E}(K,M))=M$

Recall: IND-CPA security

Fix *SE*=(KeySp,E,D)

$K \xleftarrow{\$} \text{KeySp}$

For an adversary A consider an experiment $\mathbf{Exp}^{\text{ind-cpa-b}}_{SE}(A)$



The experiment returns d

Experiment $\mathbf{Exp}^{\text{ind-cpa-1}}_{\mathcal{SE}}(A)$	Experiment $\mathbf{Exp}^{\text{ind-cpa-0}}_{\mathcal{SE}}(A)$
$K \xleftarrow{\$} \mathcal{K}$	$K \xleftarrow{\$} \mathcal{K}$
$d \xleftarrow{\$} A^{\mathcal{E}_K(\text{LR}(\cdot,\cdot,1))}$	$d \xleftarrow{\$} A^{\mathcal{E}_K(\text{LR}(\cdot,\cdot,0))}$
Return d	Return d

The IND-CPA advantage of A is:

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(A) \quad = \quad \Pr\left[\mathbf{Exp}^{\text{ind-cpa-1}}_{\mathcal{SE}}(A) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-cpa-0}}_{\mathcal{SE}}(A) = 1\right]$$

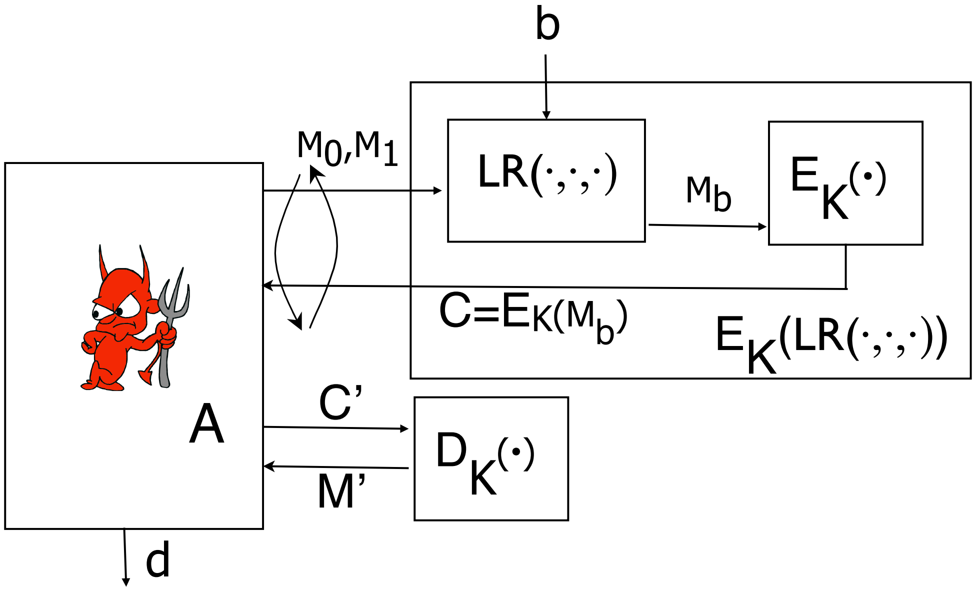
A symmetric encryption scheme *SE* is indistinguishable under chosen-plaintext attacks if for any adversary A with “reasonable” resources $\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(A)$ is “small” (close to 0).

Recall: IND-CCA security

Fix *SE*=(KeySp,E,D)

$K \xleftarrow{\$} \text{KeySp}$

For an adversary A and a bit b consider an experiment $\mathbf{Exp}_{\mathcal{SE}}^{ind-cca-b}(A)$



A is not allowed to query its decryption oracle on ciphertexts returned by its LR encryption oracle

The experiment returns d

The IND-CCA advantage of A is:

$$\mathbf{Adv}_{\mathcal{SE}}^{ind-cca}(A) = \Pr\left[\mathbf{Exp}_{\mathcal{SE}}^{ind-cca-1}(A) = 1\right] - \Pr\left[\mathbf{Exp}_{\mathcal{SE}}^{ind-cca-0}(A) = 1\right]$$

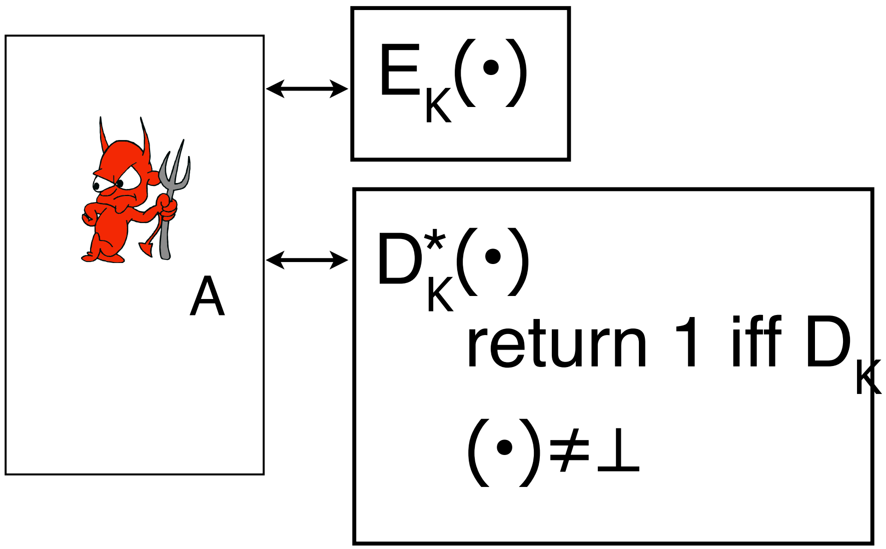
A symmetric encryption scheme *SE* is indistinguishable under chosen-ciphertext attacks (IND-CCA secure) if for any adversary A with “reasonable” resources $\mathbf{Adv}_{\mathcal{SE}}^{ind-cca}(A)$ is “small” (close to 0).

Integrity (INT-CTXT) of symmetric encryption schemes

Fix *SE*=(KeySp,E,D)

$K \xleftarrow{\$} \text{KeySp}$

For an adversary A consider an experiment $\mathbf{Exp}_{SE}^{int-ctxt}(A)$



Return 1 if A made a query C to $D^*_K(\cdot)$ s.t. $D^*_K(C)$ returns 1 and C was never a response of $E_K(\cdot)$.

$$\mathbf{Adv}_{SE}^{int-ctxt}(A) = \Pr\left[\mathbf{Exp}_{SE}^{int-ctxt}(A) = 1\right]$$

- Theorem. [IND-CPA $\wedge$ INT-CTXT $\Rightarrow$ IND-CCA] For any SE and an adversary A there exist adversaries A_c, A_p s.t.

$$\mathbf{Adv}_{SE}^{ind-cca}(A) \leq 2 \cdot \mathbf{Adv}_{SE}^{int-ctxt}(A_c) + \mathbf{Adv}_{SE}^{ind-cpa}(A_p)$$

s.t. the adversaries' resources are about the same

- Proof. Let E denote the event that A makes at least one valid decryption oracle query C, i.e. $D_K(C) \neq \perp$

Adversary $A_c^{\mathcal{E}_K(\cdot),\mathcal{D}^*_K(\cdot)}$

$b' \stackrel{\$}{\leftarrow} \{0,1\}$

When A makes a query $M_{i,0}, M_{i,1}$
to its left-or-right encryption oracle do
$A \Leftarrow \mathcal{E}_K(M_{i,b'})$.

When A makes a query C_i
to its decryption oracle do
$v \leftarrow \mathcal{D}^*_K(C_i)$
`If` $v = 0$,
`then` $A \Leftarrow \perp$,
`else` stop.

$$
\begin{aligned}
\Pr[\, b' = b \ \wedge \ E \,] &\leq \Pr[\, E \,] \\
&= \Pr_c[\, A_c \text{ succeeds} \,] \\
&= \mathbf{Adv}_{SE}^{int-ctxt}(A_c)
\end{aligned}
$$

Adversary $A_p^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}$

When A makes a query $M_{i,0}, M_{i,1}$
to its left-or-right encryption oracle do
$A \Leftarrow \mathcal{E}_K(\mathcal{LR}(M_{i,0}, M_{i,1}, b))$
When A makes a query C_i
to its decryption oracle do
$A \Leftarrow \perp$

$A \Rightarrow b'$
`Return` b'

$$\Pr\left[\, b' = b \;\wedge\; \neg E \,\right] \;\leq\; \Pr_p\left[\, b' = b \,\right]$$

$$= \frac{1}{2} \cdot \mathbf{Adv}_{SE}^{int-cpa}(A_p) + \frac{1}{2}$$

$$\frac{1}{2} \cdot \mathbf{Adv}_{SE}^{int-cca}(A) + \frac{1}{2}$$

$$= \Pr[\, b' = b \,]$$

$$= \Pr[\, b' = b \ \wedge \ E \,] + \Pr[\, b' = b \ \wedge \ \neg E \,]$$

$$\leq \frac{1}{2} \cdot \mathbf{Adv}_{SE}^{int-cpa}(A_p) + \mathbf{Adv}_{SE}^{int-ctxt}(A_c) + \frac{1}{2}$$