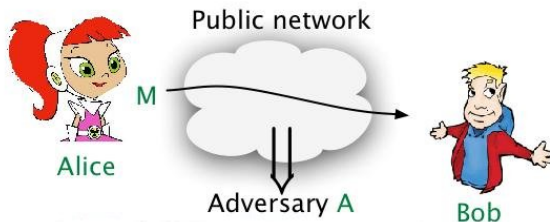


AUTHENTICATED ENCRYPTION

So Far ...



We have looked at methods to provide **privacy** and **authenticity** separately:

Goal	Primitive	Security notion
Data privacy	symmetric encryption	IND-CPA
Data authenticity	MAC	UF-CMA

Authenticated Encryption

In practice we often want **both** privacy and authenticity.

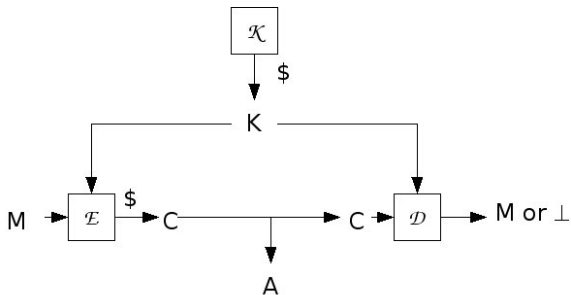
Example: A doctor wishes to send medical information M about Alice to the medical database. Then

- We want **data privacy** to ensure Alice's medical records remain **confidential**.
- We want **authenticity** to ensure the person sending the information is really the doctor and the information was **not modified** in transit.

We refer to this as **authenticated encryption**.

Authenticated Encryption Schemes

Syntactically, an authenticated encryption scheme is just a symmetric encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where



Privacy of Authenticated Encryption Schemes

The notion of **privacy** for symmetric encryption carries over, namely we want IND-CPA.

Integrity of Authenticated Encryption Schemes

Adversary's goal is to get the receiver to accept a “non-authentic” ciphertext C .

Integrity of **ciphertexts**: C is “non-authentic” if it was never transmitted by the sender.

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme and A an adversary.

Game $\text{INTCTXT}_{\mathcal{AE}}$

procedure Initialize

$K \xleftarrow{\$} \mathcal{K} ; S \leftarrow \emptyset$

procedure Enc(M)

$C \xleftarrow{\$} \mathcal{E}_K(M)$

$S \leftarrow S \cup \{C\}$

Return C

procedure Finalize(C)

$M \leftarrow \mathcal{D}_K(C)$

if ($C \notin S \wedge M \neq \perp$) then

 return true

Else return false

The int-ctxt advantage of A is

$$\text{Adv}_{\mathcal{AE}}^{\text{int-ctxt}}(A) = \Pr[\text{INTCTXT}_{\mathcal{AE}}^A \Rightarrow \text{true}]$$

The goal of authenticated encryption is to provide both integrity and privacy. We will be interested in IND-CPA + INT-CTXT.

Plain Encryption Does Not Provide Integrity

Alg $\mathcal{E}_K(M)$

$C[0] \xleftarrow{\$} \{0, 1\}^n$

For $i = 1, \dots, m$ do

$C[i] \leftarrow E_K(C[i-1] \oplus M[i])$

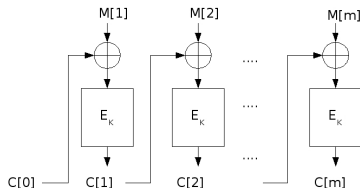
Return C

Alg $\mathcal{D}_K(C)$

For $i = 1, \dots, m$ do

$M[i] \leftarrow E_K^{-1}(C[i]) \oplus C[i-1]$

Return M



Question: Is CBC\$ encryption INT-CTXT secure?

Plain Encryption Does Not Provide Integrity

Alg $\mathcal{E}_K(M)$

$C[0] \xleftarrow{\$} \{0, 1\}^n$

For $i = 1, \dots, m$ do

$C[i] \leftarrow E_K(C[i-1] \oplus M[i])$

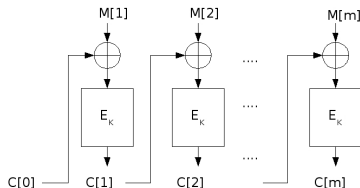
Return C

Alg $\mathcal{D}_K(C)$

For $i = 1, \dots, m$ do

$M[i] \leftarrow E_K^{-1}(C[i]) \oplus C[i-1]$

Return M



Question: Is CBC\$ encryption INT-CTXT secure?

Answer: No, because any string $C[0]C[1]\dots C[m]$ has a valid decryption.

Plain Encryption Does Not Provide Integrity

Alg $\mathcal{E}_K(M)$

$C[0] \xleftarrow{\$} \{0, 1\}^n$

For $i = 1, \dots, m$ do

$C[i] \leftarrow E_K(C[i-1] \oplus M[i])$

Return C

Alg $\mathcal{D}_K(C)$

For $i = 1, \dots, m$ do

$M[i] \leftarrow E_K^{-1}(C[i]) \oplus C[i-1]$

Return M

adversary A

$C[0]C[1]C[2] \xleftarrow{\$} \{0, 1\}^{3n}$

Return $C[0]C[1]C[2]$

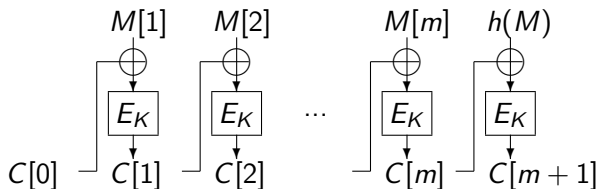
Then

$$\text{Adv}_{\mathcal{SE}}^{\text{int-ctxt}}(A) = 1$$

This violates INT-CTXT.

A scheme whose decryption algorithm **never** outputs \perp **cannot** provide **integrity!**

Encryption with Redundancy

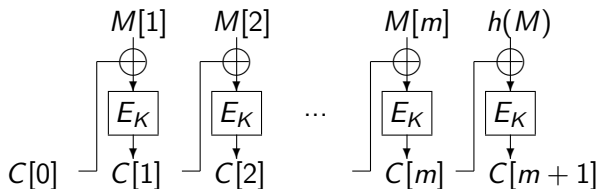


Here $E: \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is our block cipher and $h: \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a “redundancy” function, for example

- $h(M[1] \dots M[m]) = 0^n$
- $h(M[1] \dots M[m]) = M[1] \oplus \dots \oplus M[m]$
- A CRC
- $h(M[1] \dots M[m])$ is the first n bits of $\text{SHA1}(M[1] \dots M[m])$.

The redundancy is verified upon decryption.

Encryption with Redundancy



Let $E: \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be our block cipher and $h: \{0, 1\}^* \rightarrow \{0, 1\}^n$ a redundancy function. Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ be CBC\$ encryption and define the encryption with redundancy scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ via

Alg $\mathcal{E}_K(M)$

$M[1] \dots M[m] \leftarrow M$

$M[m+1] \leftarrow h(M)$

$C \xleftarrow{\$} \mathcal{E}'_K(M[1] \dots M[m]M[m+1])$

return C

Alg $\mathcal{D}_K(C)$

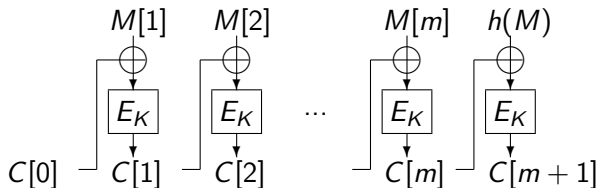
$M[1] \dots M[m]T \leftarrow \mathcal{D}'_K(C)$

if $(T = h(M))$ then

return M

else return \perp

Arguments in Favor of Encryption with Redundancy



The adversary will have a hard time producing the last enciphered block of a new message.

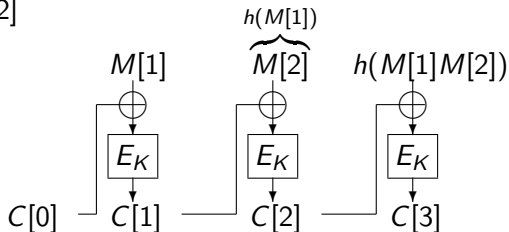
Encryption with Redundancy Fails

adversary A

$M[1] \xleftarrow{\$} \{0, 1\}^n$; $M[2] \leftarrow h(M[1])$

$C[0]C[1]C[2]C[3] \xleftarrow{\$} \mathbf{Enc}(M[1]M[2])$

Return $C[0]C[1]C[2]$



This attack succeeds for any (not secret-key dependent) redundancy function h .

A “real-life” rendition of this attack broke the 802.11 WEP protocol, which instantiated h as CRC and used a stream cipher for encryption [BGW].

What makes the attack easy to see is having a clear, strong and formal security model.

Generic Composition

A *generic composition method* **Comp** builds an authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D}) = \mathbf{Comp}[\mathcal{SE}, F]$ by combining

- a given IND-CPA symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$
- a given PRF $F : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n$

	CBC\$-AES	CTR\$-AES	...
HMAC-SHA1			
CMAC			
ECBC			
⋮			

Generic Composition

A *generic composition method* **Comp** builds an authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D}) = \mathbf{Comp}[\mathcal{SE}, F]$ by combining

- a given IND-CPA symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$
- a given PRF $F : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n$

A key $K = K_e || K_m$ for \mathcal{AE} always consists of a key K_e for \mathcal{SE} and a key K_m for F :

Alg \mathcal{K}

$K_e \xleftarrow{\$} \mathcal{K}'; K_m \xleftarrow{\$} \{0, 1\}^k$

Return $K_e || K_m$

Generic Composition Methods

The **order** in which the primitives are applied is important. Can consider

Method	Usage
Encrypt-and-MAC (E&M)	SSH
MAC-then-encrypt (MtE)	TLS 1.2
Encrypt-then-MAC (EtM)	IPSec

We study these following [BN].

Assessing **Comp**

Given a generic composition method **Comp**, and a security goal $X \in \{\text{IND-CPA}, \text{INT-CTXT}\}$, we ask, does **Comp** provide X-security? There are two possible answers:

- **YES:** This means that FOR ALL secure choices of \mathcal{SE}, F , the authenticated encryption scheme $\mathcal{AE} = \mathbf{Comp}[\mathcal{SE}, F]$ is X-secure.
- **NO:** This means that THERE EXIST secure choices of \mathcal{SE}, F for which the authenticated encryption scheme $\mathcal{AE} = \mathbf{Comp}[\mathcal{SE}, F]$ is NOT X-secure.

Above, secure choices of \mathcal{SE}, F means these are IND-CPA-secure and PRF-secure, respectively.

So a NO does not mean **Comp** always fails to be X-secure, just that there are counter-example choices of IND-CPA \mathcal{SE} and PRF F for which \mathcal{AE} fails to be X-secure.

Encrypt-and-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$

$T \leftarrow F_{K_m}(M)$

Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$

If $(T = F_{K_m}(M))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	
INT-CTXT	

Encrypt-and-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$

$T \leftarrow F_{K_m}(M)$

Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$

If $(T = F_{K_m}(M))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	NO
INT-CTXT	

Why? $T = F_{K_m}(M)$ is a deterministic function of M and allows detection of repeats.

Encrypt-and-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$

$T \leftarrow F_{K_m}(M)$

Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$

If $(T = F_{K_m}(M))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	NO
INT-CTXT	

Encrypt-and-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$
 $T \leftarrow F_{K_m}(M)$
Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$
If $(T = F_{K_m}(M))$ then return M
Else return \perp

Security	Achieved?
IND-CPA	NO
INT-CTXT	NO

Why? May be able to modify C' in such a way that its decryption is unchanged.

MAC-then-Encrypt

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$T \leftarrow F_{K_m}(M)$

$C \xleftarrow{s} \mathcal{E}'_{K_e}(M||T)$

Return C

Alg $\mathcal{D}_{K_e||K_m}(C)$

$M||T \leftarrow \mathcal{D}'_{K_e}(C)$

If $(T = F_{K_m}(M))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	
INT-CTXT	

MAC-then-Encrypt

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$T \leftarrow F_{K_m}(M)$

$C \xleftarrow{\$} \mathcal{E}'_{K_e}(M||T)$

Return C

Alg $\mathcal{D}_{K_e||K_m}(C)$

$M||T \leftarrow \mathcal{D}'_{K_e}(C)$

If $(T = F_{K_m}(M))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	YES
INT-CTXT	

Why? $\mathcal{SE}' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ is IND-CPA secure.

MAC-then-Encrypt

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$T \leftarrow F_{K_m}(M)$

$C \xleftarrow{s} \mathcal{E}'_{K_e}(M||T)$

Return C

Alg $\mathcal{D}_{K_e||K_m}(C)$

$M||T \leftarrow \mathcal{D}'_{K_e}(C)$

If $(T = F_{K_m}(M))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	YES
INT-CTXT	

MAC-then-Encrypt

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$T \leftarrow F_{K_m}(M)$

$C \xleftarrow{\$} \mathcal{E}'_{K_e}(M||T)$

Return C

Alg $\mathcal{D}_{K_e||K_m}(C)$

$M||T \leftarrow \mathcal{D}'_{K_e}(C)$

If $(T = F_{K_m}(M))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	YES
INT-CTXT	NO

Why? May be able to modify C in such a way that its decryption is unchanged.

Encrypt-then-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$

$T \leftarrow F_{K_m}(C')$

Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$

If $(T = F_{K_m}(C'))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	
INT-CTXT	

Encrypt-then-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$
 $T \leftarrow F_{K_m}(C')$
Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$
If $(T = F_{K_m}(C'))$ then return M
Else return \perp

Security	Achieved?
IND-CPA	YES
INT-CTXT	

Why? $\mathcal{SE}' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ is IND-CPA secure.

Encrypt-then-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$

$T \leftarrow F_{K_m}(C')$

Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$

If $(T = F_{K_m}(C'))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	YES
INT-CTXT	

Encrypt-then-MAC

$\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{E}_{K_e||K_m}(M)$

$C' \xleftarrow{\$} \mathcal{E}'_{K_e}(M)$

$T \leftarrow F_{K_m}(C')$

Return $C' || T$

Alg $\mathcal{D}_{K_e||K_m}(C' || T)$

$M \leftarrow \mathcal{D}'_{K_e}(C')$

If $(T = F_{K_m}(C'))$ then return M

Else return \perp

Security	Achieved?
IND-CPA	YES
INT-CTXT	YES

Why? If $C || T$ is new then T will be wrong.

Two keys or one?

We have used separate keys K_e, K_m for the encryption and message authentication. However, these can be derived from a single key K via $K_e = F_K(0)$ and $K_m = F_K(1)$, where F is a PRF such as a block cipher, the CBC-MAC or HMAC.

Trying to directly use the same key for the encryption and message authentication is error-prone, but works if done correctly.

Authenticated encryption today

- Dedicated schemes: OCB, OCB_x (x=1,2,3), GCM, CCM, EAX
- TLS uses GCM
- CAESAR competition to standardize new schemes:
<http://competitions.cr.yp.to/caesar.html>