Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 11, Part 2

Authenticated Encryption

Secrecy and Integrity Combined?

- **Secrecy**: PRF/block cipher in a mode of operation
- **Integrity**: message authentication code (MAC)

Question

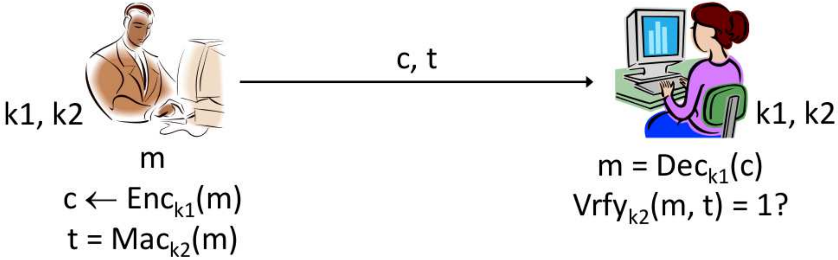
Can we combine both secrecy and integrity in a single private-key scheme?

Constructions

Three natural approaches

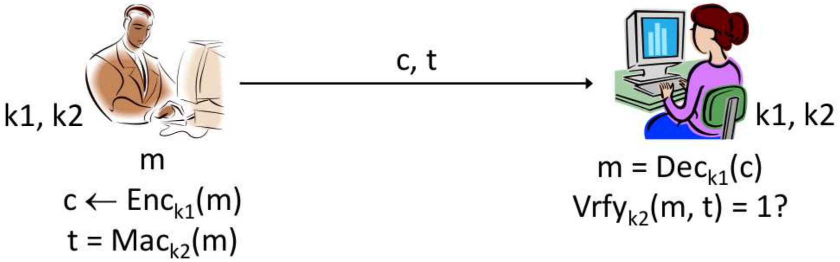
1. Encrypt-and-authenticate (E-and-A)
2. Authenticate-then-encrypt (A-then-E)
3. Encrypt-then-authenticate (E-then-A)

Encrypt-and-authenticate (E-and-A)



Sender and receiver share two keys: $\boldsymbol{k_1}$ for encryption, $\boldsymbol{k_2}$ for authentication

Encrypt-and-authenticate (E-and-A)



- Sender sends $c = \mathsf{Enc}_{k_1}(m),\ t = \mathsf{Mac}_{k_2}(m)$
- Receiver decrypts $m = \mathsf{Dec}_{k_1}(c)$ and verifies $\mathsf{Vrfy}_{k_2}(m, t) = 1$

E-and-A Weaknesses

Not CPA-secure

If the MAC is deterministic (as is CBC-MAC), then the tag leaks whether the same message is encrypted twice

- i.e. E-and-A will not be CPA-secure, even if Enc is CPA-secure

E-and-A Weaknesses

Not EAV-secure

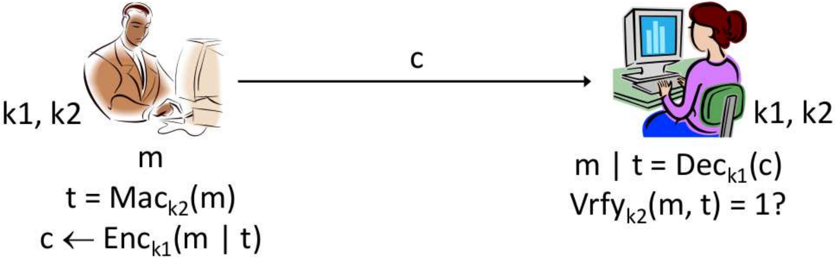
The tag $\boldsymbol{t}$ might leak information about $\boldsymbol{m}$

- Nothing in the definition of security for a MAC implies that it hides information about $\boldsymbol{m}$
- E-and-A may not even be EAV-secure

Example

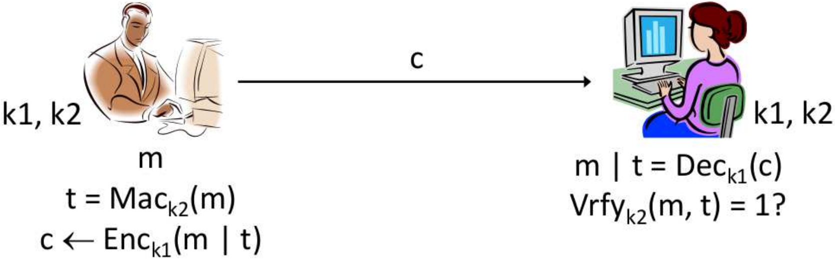
- Let $\boldsymbol{\Pi} = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ be a secure MAC
- Define $\mathsf{Mac}'_{\boldsymbol{k}} = (\boldsymbol{m}, \mathsf{Mac}_{\boldsymbol{k}}(\boldsymbol{m}))$
- $\implies \boldsymbol{\Pi}' = (\mathsf{Gen}, \mathsf{Mac}', \mathsf{Vrfy})$ is a secure MAC
- $\boldsymbol{\Pi}'$ reveals $\boldsymbol{m} \implies$ E-and-A using $\boldsymbol{\Pi}'$ is not CPA-secure

Authenticate-then-encrypt (A-then-E)



Sender and receiver share two keys: $\boldsymbol{k_1}$ for encryption, $\boldsymbol{k_2}$ for authentication

Authenticate-then-encrypt (A-then-E)



- Sender computes tag $t = \mathsf{Mac}_{k_2}(m)$ and sends $c = \mathsf{Enc}_{k_1}(m, t)$
- Receiver decrypts $(m, t) = \mathsf{Dec}_{k_1}(c)$ and verifies $\mathsf{Vrfy}_{k_2}(m, t) = 1$

A-then-E Weaknesses

Problems with A-then-E

- Padding-oracle attack
- Other counter-examples are also possible
  - The combination may not be CCA-secure

A-then-E: Padding Oracle Attack

A-then-E scheme Π

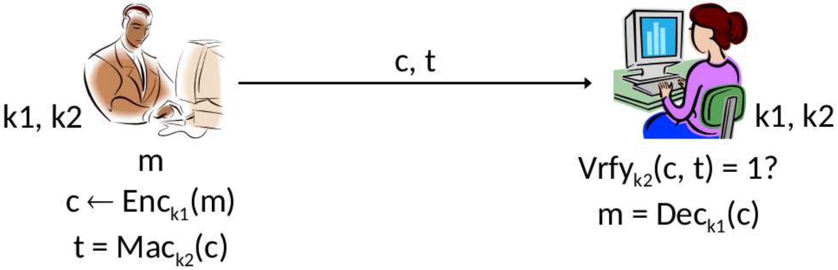
- Encode m applying $T(m)$ as
  - replace $\mathbf{0} \to \mathbf{00}$, replace $\mathbf{1} \to \mathbf{01}$ *or* $\mathbf{10}$
- Decode m from $T(m)$ as:
  - replace $\mathbf{00} \to \mathbf{0}$, replace $\mathbf{01}$ **or** $\mathbf{10} \to \mathbf{1}$
  - if $\mathbf{11}$ return $\perp$ (error)
- Let Enc be a cipher that generates a PR sequence and XORs it with m
  - e.g. PRF/block cipher in CTR mode
- Define $\mathsf{Enc}'_k(m) = \mathsf{Enc}_k(T(m))$
- Let Π be an A-then-E scheme using Enc'

A-then-E: Padding Oracle Attack

Padding-oracle attack on Π

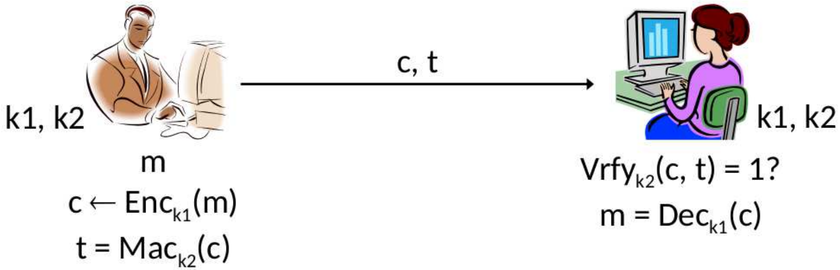
- A attacks Π following the CCA experiment
- A gets challenge $c = \mathsf{Enc}'_{k_1}(T(m, \mathsf{Mac}_{k_2}(m)))$
- A flips first 2 bits of c to get c'
- A submits c' to the decryption oracle $\mathcal{O}$
- If $\mathcal{O}$ returns $\perp \implies A$ infers first bit of c to be 0
- Otherwise A infers the first bit of c to be 1
- $\implies \Pi$ not CCA-secure

Encrypt-then-authenticate (E-then-A)



Sender and receiver share two keys: $\boldsymbol{k_1}$ for encryption, $\boldsymbol{k_2}$ for authentication

Encrypt-then-authenticate (E-then-A)



- Sender sends $c = \mathsf{Enc}_{k_1}(m),\ t = \mathsf{Mac}_{k_2}(c)$
- Receiver verifies $\mathsf{Vrfy}_{k_2}(c, t) = 1$ and (if t is valid) decrypts $m = \mathsf{Dec}_{k_1}(c)$

Security of E-then-A

Theorem

If the underlying encryption scheme is CPA-secure and the MAC is secure (i.e. existentially unforgeable) then the E-then-A combination is a CCA-secure encryption scheme

Proof

Omitted

Note

The encryption and authentication keys $\boldsymbol{k_1}$ and $\boldsymbol{k_2}$ must be independent

A CCA-secure Scheme

Encrypt-then-authenticate

E-and-A is the right way to combine secrecy with integrity:

- Use a CPA-secure encryption scheme to encrypt the message
- Use a MAC to prevent the ciphertext from being modified

A stronger notion than CCA

Observation

The E-then-A approach results in a stronger notion than CCA-security:

- The MAC is applied on the **ciphertext** produced by the sender
- $\implies$ The adversary is **not able to obtain any valid ciphertext** that was not generated by the legitimate parties
  - thus rendering the decryption oracle useless
- This property is **not implied by CCA-security**
  - where the attacker is allowed to query the decryption oracle on any chosen ciphertexts and receive the corresponding plaintexts

Authenticated Encryption

A stronger property than CCA

Given ciphertexts $(c_1, t_1), (c_2, t_2), \ldots$ corresponding to (chosen) plaintexts $m_1, m_2, \ldots$, it is infeasible for an attacker to generate any new valid ciphertext (c, t).

- i.e. if an attacker injects his own ciphertext, the decryption oracle will output an error (rather than the corresponding plaintext)

Authenticated encryption (AE) scheme

Schemes with the above property are called **authenticated encryption** schemes

Authenticated Encryption

Theorem

If the underlying encryption scheme is CPA-secure and the MAC is secure then the E-then-A combination is an AE scheme

E-then-A is the recommended generic approach to constructing an AE scheme

- "Generic" = using any CPA-secure scheme and any secure MAC

Direct AE Constructions

Other, more-efficient AE constructions exist:

- OCB, CCM, GCM
- Finalists from the CAESAR competition
    - https://competitions.cr.yp.to/caesar-submissions.html

End of Symmetric-key Part

Summary of Symmetric-key Topics

- Historical ciphers: Shift cipher, Vigenère
- Perfect secrecy
- One-time pad (OTP)
- Computational secrecy
- Pseudorandom generators (PRG)
- Pseudo-OTP
- Security against chosen-plaintext attacks (CPA)
- Pseudorandom functions / permutations (PRF / PRP)

Summary of Symmetric-key Topics

- CPA-secure encryption using PRF/PRP
- Modes of operation: block ciphers
- Malleability
- Security against chosen-ciphertext attacks (CCA)
- Non-CCA secure schemes: padding-oracle attacks
- Secrecy vs. integrity: message authentication codes (MAC)
- Hash functions
- Secrecy and integrity; authenticated encryption

What next?

Observe

The security of symmetric-key schemes ultimately depends on the **secrecy of the key**

Problem

How do we distribute the keys in the first place?

Solution

Public-key cryptography.

End

References: Sec 4.5.1, 4.5.2 (not Theorem 4.19)