Solutions to Selected Problems

Guide to Internet Cryptography

Companion Material

February 24, 2026

Preface This document provides solutions to selected problems from the book *Guide to Internet Cryptography: Security Protocols and Real-World Attack Implications*. The material is intended for educational use in courses and self-study.

Book website: https://link.springer.com/book/10.1007/978-3-031-19439-9

1 Chapter 18: Attacks on S/MIME and PGP

Problem 18.1 Crypto gadgets

(a) Sketch a crypto gadget for CFB mode, similar to Figure 18.2.

(b) Which other encryption modes besides CBC and CFB could be used for crypto gadget attacks? Are there any restrictions?

(c) Suppose your e-mail client allows to load external CSS files via the `<style>` element. Sketch, in the context of AES-CBC encryption, the sequence of 16-byte chunks to exfiltrate the unknown plaintext. Make sure to exclude the intermediate pseudorandom 16-byte-chunks from HTML parsing.

Solution

(a) See Figure 1.

(b) All crypto modes that somehow XOR plaintext and ciphertext. In section 2.2.2, these

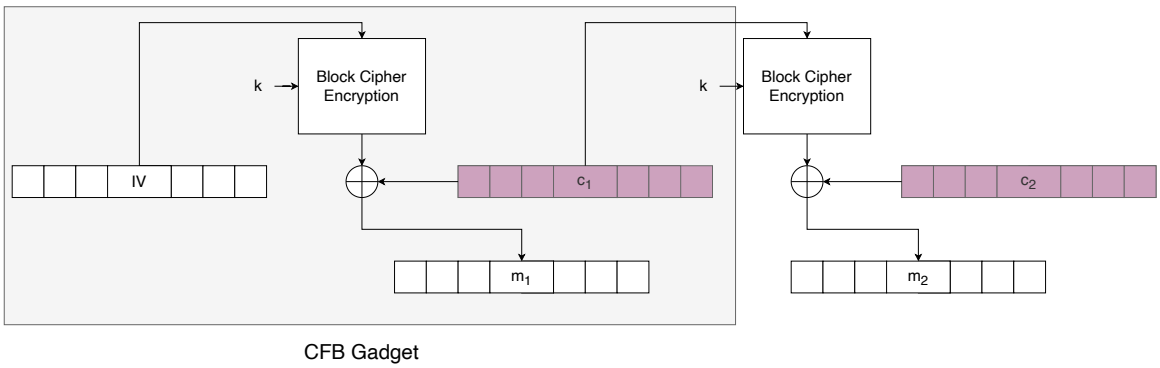

Figure 1: CFB Gadget. A known plaintext in m_1 can be changed into a chosen plaintext m_1' by XORing c_1 with $m_1 \oplus m_1'$.

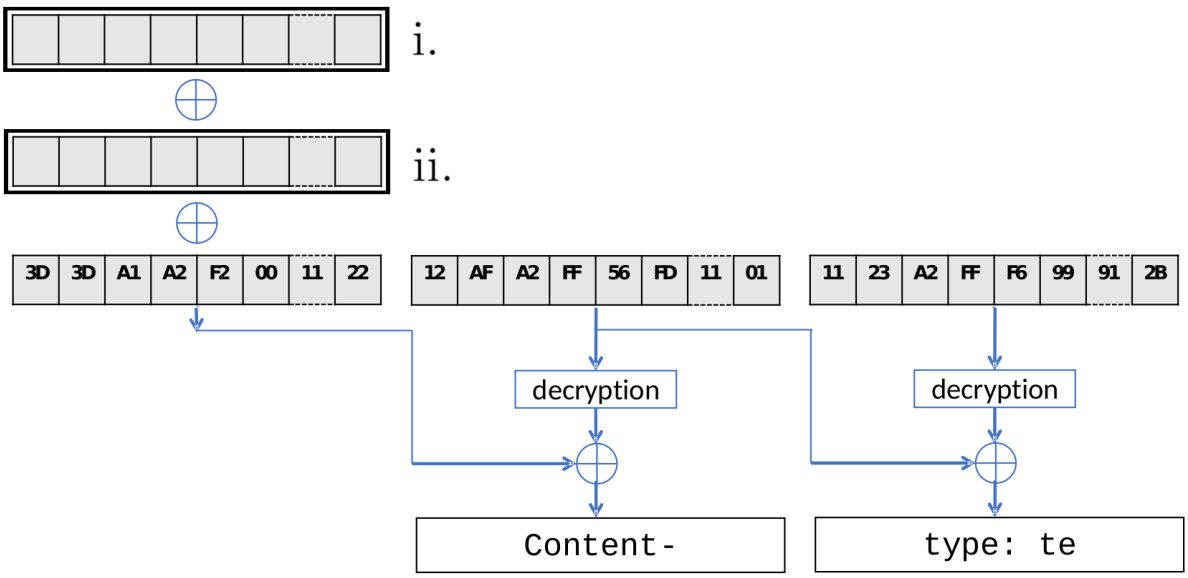

Figure 2: 3DES crypto gadget pattern.

are CBC, CFB, OFB and CTR. For CBC and CFB, every second block of plaintext will be destroyed, but we can re-use malleability gadgets. In OFB and CTR, we can replace any sequence of known plaintext with chosen pliantext, but we cannot re-use gedagets since we do not control the input to AES.

(c) You can craft a solution as follows:

```
IV' (mod)     |
c1 (original) | <style>@import '
IV'' (mod)    | pseudorandom
c1 (original) | 'https://ev.il/'
IV (orig)     | pseudorandom
c1 (original) | secret text
c2 (original) | text
```

Problem 18.2 Crypto gadgets 2

In the following [Figure 2] 3DES-CBC crypto gadget pattern, insert the hexadecimal values to change the known plaintext `Content-` into the chosen plaintext `<img src`. Use (i) to create the all-zero plaintext block and (ii) to insert the chosen plaintext.

Solution

You can replace the lookup table with an online ascii-to-hex converter like https://www.utilities-online.info/ascii-to-hex.

(i) `Content- --> 43 6f 6e 74 65 6e 74 2d`

(ii) `<img src --> 3c 69 6d 67 20 73 72 63`

Problem 18.3 Crypto gadgets 3

(a) How many blocks of known plaintext are needed for a crypto gadget attack?
(b) Suppose you only know 14 of the 16 plaintext bytes in an AES-CBC block. Can you still construct a crypto gadget?

Solution

(a) One block only.

(b) Yes. Since the malleability through XOR applies bit-to-bit, we can modify the 14 byte and just comment out the two remaining, inknown bytes.

Problem 18.4 Direct exfiltration

A new startup company announced they had implemented mitigation for EFAIL direct exfiltration attacks. Their e-mail client only decrypts an e-mail if all leaves of the MIME tree are encrypted. They argue that this way, an attacker cannot insert his malicious content. Would you invest in this startup?

Solution

No. There are many reasons for this idea to be stupid.

- This would be a recipient-enforced security policy. The sender cannot enforce this behaviour, so she/he cannot be sure if this policy ois applied by all recipients.
- The startup didn't guarantee that all leaves must be symmetrically encrypted with the *same symmetric key*. So an attacker could simply use hybrid encryption to encryt the remaining leaves with the public key of the chosen victim, and another randomly chosen symmetric key.

Problem 18.5 Direct exfiltration 2

Suppose the public key needed to verify a `multipart/signed` e-mail is used as associated data in the AEAD encryption of the first MIME entity. Can this reliably prevent direct exfiltration attacks?

Solution

No. It would prevent Malleability Gadget attacks, because the signature protects the integrity of the ciphertext, and cannot be removed because without this signature, the associated data would be missing.

It does not prevent direct exfiltration attacks, because a ne multipart/mixed root can be added to the MIME tree, and the multipart/signed would just become the middle subtree between two text/html leaves.

Problem 18.6 Digital signatures

Please sketch the MIME source code of an e-mail, where unsigned HTML content is inserted before a `multipart/signed` MIME entity.

Solution

```
Content-Type: multipart/mixed; boundary="ZZZ"

--ZZZ
Content-Type: text/html

<html>...
--ZZZ
Content-Type: multipart/signed; boundary="YYY"

--YYY
Content-Type: text/plain

Hello world!
--YYY
Content-Type: application/pkcs7-signature
Content-Transfer-Encoding: base64

AB3DeFGH....
--YYY--

--ZZZ--
```

Problem 18.7 Digital signatures 2

Write a procedure in pseudocode to prevent CMS wrapping attacks.

Solution

```
IF (MIMEObject.Content-Type=multipart/signed) SET eci=FALSE;
CMS.Verify(MIMEObject.Child[2], Hash(MIMEObject.Child[1]), eci)

CMS.Verify(A,B,C):
IF B=FALSE CHECK treat EncapsulatedContentInfo as empty;
CHECK that the signed Attribute ContentHash equals B;
CHECK that A contains a valid signature for B;
RETURN Result
```

Problem 18.8 Reply attacks

Consult RFC 5322 about which RFC 822 headers determine the recipient of a reply e-mail.

Solution

See section 3.6.2; 3.6.3; 3.6.4; 3.6.6; 5; A.2; A.3.