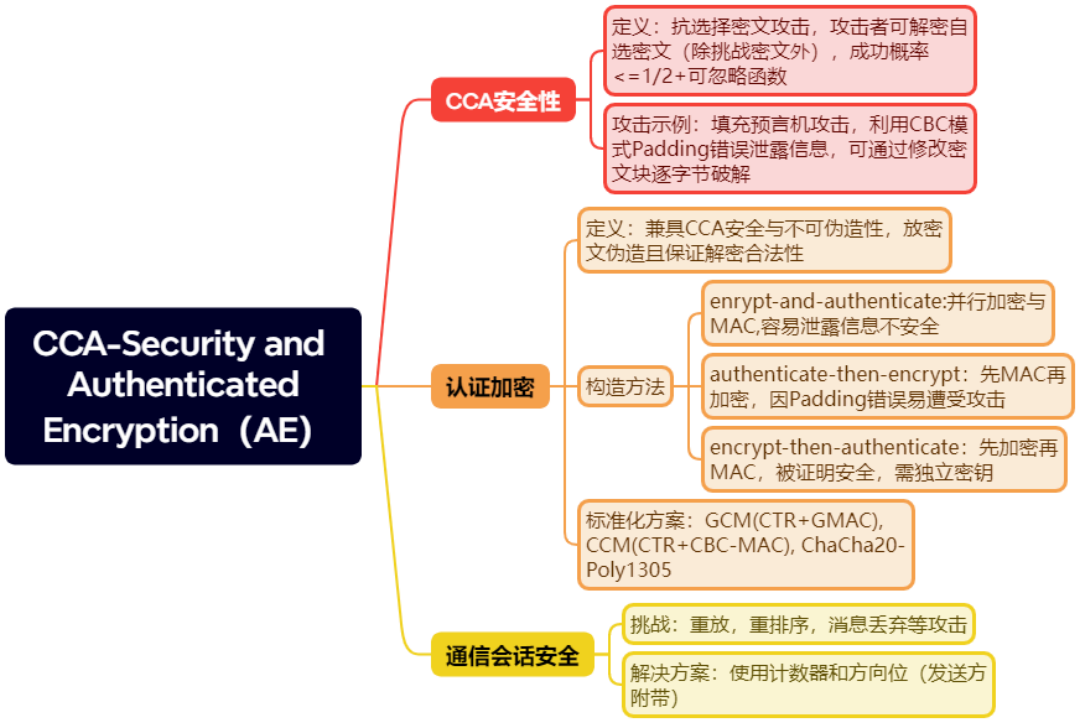
# 一段话总结

文档聚焦于**CCA安全性**与**认证加密**，先通过填充预言机攻击实例阐释选择密文攻击的危害，定义CCA安全需抵御攻击者对密文的选择与解密操作；进而提出认证加密需同时实现保密性与完整性，分析了encrypt-and-authenticate、authenticate-then-encrypt、encrypt-then-authenticate三种通用构造，其中仅后者被证明安全，最后介绍了GCM、CCM、ChaCha20–Poly1305等标准化认证加密方案及其在通信会话中的应用挑战。

# 思维导图




# 详细总结

## 一、CCA安全（抗选择密文攻击）

**1. 定义：攻击者可访问加密/解密预言机，除挑战密文外可解密任意密文，成功区分两消息加密的概率≤1/2+可忽略函数。**

**2. 攻击实例：填充预言机攻击　- 场景：CBC模式加密使用PKCS#7 padding，攻击者可伪造密文触发服务器Padding验证错误。　- 原理：通过修改前一密文块，使解密后明文块的Padding符合规则，逐字节破解明文。如已知密文块(c_1,c_2)，修改(c_1)的最后一个字节，若解密后Padding正确（如最后b字节为0xb），则可推算明文。　- 危害：可完全破解加密消息，如CAPTCHA系统中破解加密的验证词。**

## 二、认证加密（AE）

**1. 定义：需满足CCA安全与不可伪造性　- 不可伪造性：攻击者无法生成解密后非⊥且未加密过的消息。 - AE实验：攻击者无法区分真实加密/解密预言机与加密0串/始终错误的预言机。**

2.**通用构造对比**

**| 构造方法 | 步骤 | 安全性 | 问题 |**

**| encrypt-and-authenticate | 并行加密(c=Enc(m))与MAC(t=Mac(m)) | 不安全 | MAC可能泄露明文信息，如 deterministic MAC暴露消息重复 |**

**| authenticate-then-encrypt | 先(t=Mac(m))，再加密(m|t) | 不安全 | 若加密使用CBC+Padding，Padding错误可被利用破解 |**

**| encrypt-then-authenticate | 先(c=Enc(m))，再(t=Mac(c)) | 安全 | 需独立密钥，MAC确保密文合法性，加密保证保密性 |**

3. **关键定理**：若(\Pi_E)是CPA安全加密，(\Pi_M)是强安全MAC，则encrypt-then-authenticate构造为认证加密。

## 三、标准化认证加密方案

| 方案 | 核心组件 | 特点 | 注意事项 |
| --- | --- | --- | --- |
| GCM | CTR 模式 + GMAC | 硬件优化快，并行性好 | IV 不可重复，否则完整性可能破坏 |
| CCM | CTR 模式 + CBC-MAC | 单密钥易实现，速度慢 | 需消息长度已知，无法在线加密 |
| ChaCha20–Poly1305 | ChaCha20 流密码 + Poly1305 MAC | 软件效率高，无硬件依赖 | 适用于无 GCM 硬件加速场景 |

## 四、通信会话安全挑战与解决方案

1. **攻击类型**：重放、重排序、消息丢弃、反射攻击等。
2. **解决方案**：使用计数器与方向位。如发送方在消息中附加方向位（如(b*{A,B}）和计数器*((ctr*{A,B}))，每发送一次计数器递增，接收方验证方向位和计数器合法性，防重放与顺序错误。

# 关键问题

1. **CCA安全与CPA安全的核心区别是什么?**

   **答案**：CPA安全允许攻击者选择明文加密并观察密文，而CCA安全在此基础上允许攻击者选择密文解密（除挑战密文），更严格地模拟主动攻击场景。CCA安全要求即使攻击者能让接收方解密任意密文，也无法获取真实加密消息的信息，而CPA安全未考虑密文解密攻击。

2. **为何encrypt-then-authenticate是安全的认证加密构造?**

   **答案**：先加密再MAC的构造中，加密确保消息保密性，MAC确保密文合法性。强安全MAC保证攻击者无法伪造合法密文，即使攻击者提交密文到解密预言机，若未从加密预言机获取过该密文，解密将因MAC验证失败返回错误，从而阻断攻击。同时，独立密钥避免密钥重用导致的安全漏洞，如密钥共享时可能泄露明文。

3. **填充预言机攻击的基本原理是什么？如何防范？** -

   答案**：原理是利用CBC模式加密中Padding验证机制，攻击者通过修改密文块，使解密后的明文块满足Padding规则（如最后b字节为0xb），从而逐字节推断明文。例如，修改前一密文块的某字节，若服务器返回Padding正确，则可确定该字节修改后与原密文异或得到正确Padding值。防范措施包括：使用不依赖Padding的加密模式（如CTR模式）、统一错误消息避免泄露Padding错误类型、或在解密时先验证MAC再处理Padding（如encrypt-then-authenticate构造）。