计算机和网络安全课程笔记 Week 7

Nite

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

秘密分割 (Secret Splitting) & 秘密共享 (Secret Sharing)

- **秘密分割 (Secret Splitting)**
  - 问题：如何将一个秘密分发给多人，使得任何单一份都没有用?
  - 场景：CEO 保护秘密配方免受工业间谍窃取。
  - **使用 XOR 进行秘密分割 (Secret Splitting with XOR)**
    - 流程：
      1. Trent 生成与消息 m 等长的随机字符串 r。
      2. 计算 $s = m \oplus r$。
      3. 将 r 交给 Alice，将 s 交给 Bob。
    - 秘密碎片 (shadows)：r 和 s。
    - 恢复秘密：Alice 和 Bob 计算 $s \oplus r = m$。
    - 安全性：如果 r 是真随机的，则系统是完美安全的 (类似 One Time Pad)。
    - 扩展到 n 人：生成 $n-1$ 个随机字符串 $r_1, \ldots, r_{n-1}$。前 $n-1$ 人各持一个 r_i，第 n 人持 $r_1 \oplus \cdots \oplus r_{n-1} \oplus m$。
    - 目的：通过分发信任增强可靠性，不增加风险。
  - **XOR 秘密分割的问题 (Issues with secret splitting with XOR)**
    1. 系统由 Trent 裁决 (adjudicated)。
       - Trent 可能分发错误信息。
       - Trent 可能欺骗参与者数量。
    2. 所有参与者知道消息长度。
    3. 消息是可塑的 (malleable)。
    4. 需要所有参与者才能恢复秘密。
- **秘密共享 (Secret Sharing)**
  - 问题：管理公司资金，需要多于某个门限的参与者同意才能执行操作。例如，(3, 5) 门限方案需要 5 人中的至少 3 人同意。
  - 目标：确保没有单个人或一对参与者可以执行操作。
  - **Shamir 秘密共享 (Shamir's Secret Sharing - SSS)**
    - 算法：将秘密分割成 m 份，任意 n 份即可重建秘密。
    - 基础：一个 n 次多项式可以由 $n+1$ 个点唯一确定。
      -
    - 流程 (Trent 将秘密 m 分发给 n 用户，任意 t 个用户可恢复 m):
      1. 选择素数 $p > \max\{m, n\}$。
      2. 创建 $t-1$ 次多项式 $f(x) = m + a_1 x + \cdots + a_{t-1} x^{t-1}$，其中 a_i 是 $0 \leq a_i < p$ 的随机独立系数。($f(0) = m$)

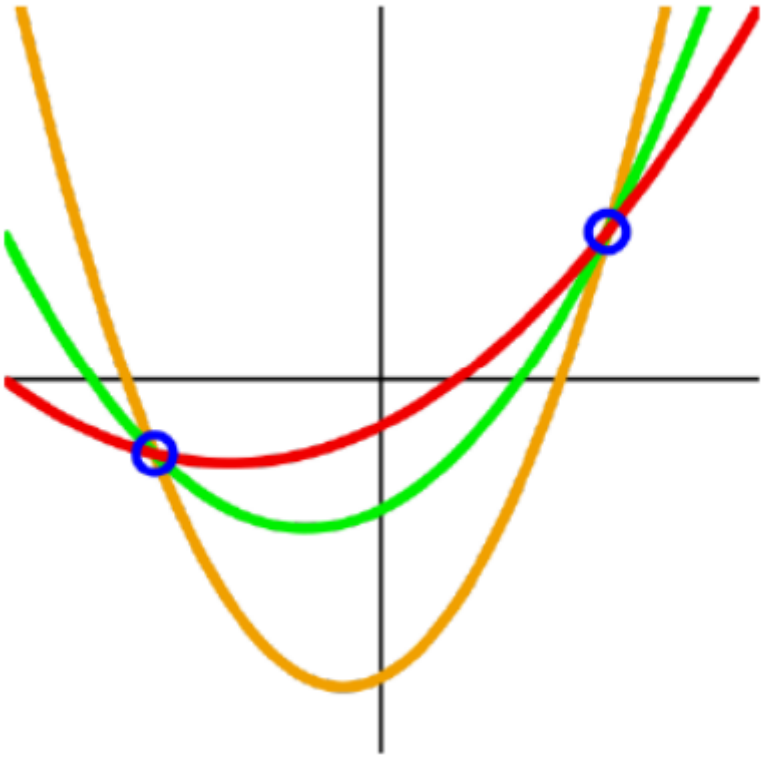

图 1: 通过点确定多项式的示意图

3. Trent 选择 n 个不同的点 x_i $(1 \leq x_i < p)$。
4. Trent 将点 $(x_i, f(x_i))$ 交给第 i 个人。

⋆ 恢复：任意 t 个参与者汇集他们的点 $(x_i, f(x_i))$，通过插值重建 $f(x)$，然后计算 $f(0)$ 恢复 m。

承诺协议 (Commitment Protocols)

- **承诺方案 (Commitment Schemes)**
  - 问题：Alice 向 Bob 发送消息，但只允许 Bob 稍后凭 Alice 提供的信息揭示，且 Bob 能验证消息未被篡改。
  - **承诺阶段 (Commitment)**
    1. Bob 生成随机字符串 r，发送给 Alice。
    2. Alice 生成随机密钥 k，发送 Bob 加密消息 $E_k(r \parallel m)$。
  - **揭示阶段 (Revelation)**
    1. Alice 发送密钥 k 给 Bob。
    2. Bob 用 k 解密，验证 r。
  - 讨论：
    ⋆ r 用于保证新鲜度 (freshness)，防止 Alice 寻找碰撞。
    ⋆ Alice 发送 $E_k(r \parallel m)$ 时必须承诺 m。
    ⋆ Bob 在揭示前不知 k。
- **非交互式比特承诺 (Non-interactive Bit Commitment)**
  - 特点：Bob 无需发送消息。
  - **承诺阶段 (Commitment)**

1. Alice 生成随机 r, s，计算 $a = h(r \parallel s \parallel m)$ (blob)，h 是密码学哈希函数。
2. Alice 发送 (r, a) 给 Bob。

- **揭示阶段 (Revelation)**
  1. Alice 发送 (s, m) 给 Bob。
  2. Bob 验证 $h(r \parallel s \parallel m)$ 是否等于 a。
- 讨论:
  * Alice 发送两次消息（承诺和揭示）。
  * 哈希函数的抗碰撞性阻止 Alice 更改消息。
  * s 保密防止 Bob 暴力破解 m。

基于承诺协议/公钥的协议 (Protocols using Commitment/Public Keys)

- **使用承诺方案进行公平掷币 (Fair Coin Flipping using Commitment Schemes)**
  - 问题：互不信任的 Alice 和 Bob 如何公平掷币决定先后。
  - 流程:
    1. Alice 使用承诺方案承诺一个比特 b (结果)。
    2. Bob 猜测 b。
    3. Alice 揭示 b。猜对者获胜。
  - 安全性：依赖承诺方案的安全性 (隐藏性和绑定性)。
- **使用公钥进行公平掷币 (Fair Coin Flipping using Public Keys)**
  - 要求：可交换的公钥密码系统 (commutative public key cryptosystem) $E_A(E_B(m)) = E_B(E_A(m))$。
  - 流程:
    1. Alice 和 Bob 生成可交换公钥对 A, B。
    2. Alice 生成秘密随机数 r_H, r_T。
    3. Alice 以随机顺序发送 $E_A(\text{heads}, r_H)$ 和 $E_A(\text{tails}, r_T)$ 给 Bob。
    4. Bob 选择一个加密消息 $E_A(a)$，发送 $E_B(E_A(a))$ 给 Alice。
    5. Alice 解密 ($E_B(a)$)，发回 Bob。
    6. Bob 解密 (a)，得知结果，发回 Alice。
    7. Alice 验证结果是否匹配其 r_H/r_T。
  - 讨论:
    * 自强制 (self-enforcing)：双方都可检测作弊，无需第三方。
    * Bob 先知结果。他不能改结果，但可延迟 (“扔进井里”)。
    * 可用于会话密钥生成，双方不能影响结果。
- **心理扑克 (Mental Poker)**
  - 问题：如何用可交换密钥公平发牌。
  - 流程:
    1. Bob 用秘密密钥加密 52 张牌，随机顺序发给 Alice。
    2. Alice 随机选 5 张加密牌发 Bob，Bob 解密得手牌。
    3. Alice 另选 5 张加密牌，用自己密钥加密发 Bob，Bob 解密再发回 Alice，Alice 解密得手牌 (发更多牌重复 2, 3)。
    4. 游戏结束，双方揭示秘密密钥验算是否作弊。
  - 进一步阅读。

隐蔽信道 (Covert Channels) 与隐写术 (Steganography)

- **隐蔽信道 (Covert Channels)**
  - 问题：Alice 和 Bob 在 Walter 监听下秘密通信。Walter 可能插入虚假消息。
  - 隐写术 (Steganography)：信息隐藏技术。
    - 不是密码学，是“通过模糊达到安全”(security through obscurity)。
    - 通常与密码学结合使用。
- **文本隐写术 (Text Steganography)**
  - 方法：
    1. 计算句子词数奇偶性编码比特。
    2. 同义词替换。
    3. 句法替换。
    4. 上下文同义词替换和顶点编码。
- **图像隐写术 (Image Steganography)**
  - 原理：修改像素值的最低有效位 (least-significant bit - LSB)。
    - 对图像质量影响微乎其微，肉眼难以察觉。
    - 例如，32x32 灰度图每个像素改 1 比特，可隐藏 1024 比特 (128 字符)。
    - 彩色 (RGB) 图像每像素可用 3 比特 (每个颜色分量 1 比特)。
    - 分辨率加倍可隐藏数据量翻四倍。
    - 可接受更多质量损失，如每像素用 6 比特。
    - 存在抵抗有损压缩和调整大小的技术。
- **网络隐写术 (Network Steganography)**
  - 方法：
    1. Loki: 利用 ICMP Echo 请求/回复数据字段。
    2. ICMP Backdoor: 伪装成 ping 包分段。
    3. Rwwwshell: 伪装成 HTTP Response 和 HTTP GET。
    4. AckCmd: 利用 TCP ACK 包。
    - 可用于穿透防火墙。

网络安全概述 (Network Security Overview)

- **网络安全的挑战 (Challenges of Network Security)**
  1. 互联网设计时缺乏安全性，存在许多漏洞。
     - 操作系统、协议栈 (IP, 802.11/WEP)、用户协议 (Telnet, FTP, HTTP)、网络协议 (BGP, DNS)、管理协议 (SNMP)、编程语言。
  2. 缺乏合适的基础设施和高质量开发者。
  3. 糟糕的设计和编程实践。
  4. 信息泄露风险。
  5. 系统 24x7x365 暴露于外部。
  6. 妥协风险增加导致风险提高。
  7. 攻击者易于发起攻击并逃脱，缺乏国际司法管辖。
  8. 安全总是滞后的，响应慢。
  9. 安全常被视为成本中心。
  10. 互联网的同质性与异质性并存。
  11. 政治、出口限制、专利问题。

12. 使用互联网的人类因素。

- **互联网的同质性 (The Internet is a Monoculture)**
  - 许多软件 (OpenSSL, Windows, Bind, Apache/Nginx) 市场主导。
  - 软件副本相同，一个版本漏洞影响所有副本。
  - 攻击流行软件导致大规模机器被控。
  - IoT 设备快速普及，安全性不足。
- **关于安全的常见信念 (Common Beliefs about Security)**
  - “保护边界即可保证内部安全”是错误假设。
  - 现代网络**没有清晰边界**。
  - **不能信任任何人**。
  - 进入网络途径众多：互联网连接、未更新的老旧机器、无线网络、第三方连接。
  - **用户是 90% 的问题源**，且已在内部。
- **安全漏洞 (Security Flaws)**
  - 现代安全问题与过去相似，只是技术普及导致漏洞增多。
  - 许多源于缺乏强认证 (lack of strong authentication)、糟糕编程 (bad programming)、差劲管理 (poor security administration)。
- **特定攻击 (Specific Attacks)**
  - **分布式拒绝服务攻击 (Distributed Denial of Service - DDOS)**
    - 目的：使目标服务不可用。
    - 原理：利用大量机器发送大量请求淹没目标。
    - 历史案例：2000 年 2 月多网站遭遇 DDOS 中断 (Yahoo, Ebay, Amazon, CNN, ZDnet, E*trade)。攻击方式包括放大 Ping 和 SYN floods。
  - **僵尸网络攻击 (Botnet Attacks)**
    - Conficker 蠕虫 (2008):
      - 目标：Microsoft Windows。
      - 后果：创建了计算能力超强的僵尸网络。
      - 特点：能通过变体抵抗，修复自身漏洞，更难阻止。
      - 利用漏洞 (MS08-067)，即使补丁已发布，大量机器 (>=30%) 未及时更新仍受影响。

可能的考法与示例考题

- 理解**秘密分割 (Secret Splitting)** 和**秘密共享 (Secret Sharing)** 的**区别**及其**应用场景**。Shamir 秘密共享如何解决 XOR 分割的**局限性**?
- **承诺协议 (Commitment Schemes)** 的**基本流程**和**重要属性**（如隐藏性 hiding 和绑定性 binding）。它们在**公平掷币**中的作用。
- **基于公钥的公平掷币**协议流程，理解**可交换性**的作用以及为何它是**自强制**的。
- **隐写术 (Steganography)** 的**定义**、与**密码学**的**区别**，以及**不同类型**（文本、图像、网络）的隐写术原理。
- 互联网安全面临的**挑战**，如**固有漏洞**、**缺乏基础设施/人才**、**人类因素**等。
- 互联网作为**同质化单点故障**环境的含义及其对安全的影响。
- 理解**常见安全信念**的错误性，如“保护边界”，以及现代网络**模糊的边界**和**多种攻击途径**。
- 掌握 **DDOS** 和**僵尸网络**等典型网络攻击的**原理**和**危害**，能够结合 Conficker 等案例进行说明。