

基于服务器随机数异或的短口令方案

沈建人

(南京邮电大学, 江苏 南京 210023)

摘要:为增强口令易用性和安全性,提出一个基于服务器随机数异或的短口令方案。该方案中的口令由用户口令和服务器的口令两部分异或运算而成。用户口令足够短,便于用户使用。服务器的口令被加密保存,由高安全级别的系统存储密钥并提供加解密服务。计算能力不受限制的线下攻击者不能破解该方案中的用户口令,该方案在防御口令猜测攻击上比现有口令方案更优。

关键词:口令;猜测攻击;词典攻击;暴力破解

中图分类号:TP309.2

文献标识码:A

文章编号:1671-0673(2026)03-0304-06

A Short Password Scheme XORed with Server Random Numbers

SHEN Jianren

(Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: In order to strengthen the usability and security of passwords, a short password scheme, XORed with server random numbers, is proposed. In this scheme, a password is formed by XOR operation of a user password and a server password. The user password is short for ease of use. The server password is encrypted, and an additional secure system is introduced to store the secret key and provide encryption and decryption services. Computationally unbounded attackers cannot succeed in offline attacks, this scheme is better than existing password schemes in preventing password guessing attacks.

Key words: password; guessing attacks; dictionary attacks; brute-force attacks

访问控制是系统的基本安全需求。鉴于成本和易用性,口令是网络服务中用于终端用户身份认证的最常用方式^[1]。考虑到方便性,用户通常选择容易记住的口令,这些口令的熵很低^[2]。只要口令是用户容易记住的,即使口令的搜索空间很大,也容易遭受词典攻击的侵害^[3]。同时,硬件计算能力不断提升,价格不断降低,这增强了线下攻击者破解复杂口令的能力。线下口令猜测攻击的方法越来越先进^[4-5]。用户口令安全受到线下攻击的严重威胁。这是一个既基本又严峻的安全问题。口令是互联网网络服务的第一道安全防线。一旦用户口令被破解,随之而来的就是隐私侵犯、数据破坏、经济损失,乃至网络犯罪等一系列恶性事件^[6]。因此,网络服务的系统管理员往往要求用户设置

足够长、足够复杂的口令。复杂口令是一把双刃剑,它增加了线下攻击的难度,但用户不容易记住它^[7]。用户不能方便地记住口令,这违背了口令认证的初衷。

许多工作企图构建好记难猜的口令方案,影响较大的是Honeywords^[8]。Honeywords允许设置好记的用户口令,让服务器伪造与用户真实口令相似的虚假口令。我国南开大学汪定教授发现,Honeywords很难用机器生成的随机虚假口令迷惑攻击者,攻击者仍可高概率区分用户选择的真实口令和机器生成的虚假口令^[9]。Honeywords不能在线下攻击中充分保护真实的用户口令。

为了既要让用户容易记住口令,又要防范线下攻击者破解用户口令,本文提出一个新的口令

认证方案 CSPS(Catchy Short Password Scheme)。在 CSPS 方案中,用户选择容易记住的口令,不需要设置复杂口令。

1 CSPS 方案

1.1 用户口令与服务器口令

本文方案使用口令组合。口令 Password 由用户口令 Puser 和服务器口令 Pserver 异或运算而成。Puser 是用户选择的好记的短口令,包含 m 个键盘字符。Pserver 是由服务器端的认证系统随机生成的,包含 n 位。用户注册网络服务账号时,选择一个 m 字符长的用户口令;同时,服务器为这个用户选择一个 n 位长的服务器口令。两者的组合方式是:用户口令 Puser 与服务器口令 Pserver 执行异或运算,生成口令 Password:

$$\text{Password} = \text{Puser} \oplus \text{Pserver} \quad (1)$$

因为 Puser 的长度远小于 Pserver,所以在 Puser 后面每位填充 0,直到与 Pserver 一样长。Password 的长度由 Pserver 决定。口令组合如图 1 所示。

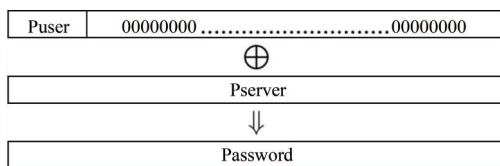


图1 口令组合: $\text{Password} = \text{Puser} \oplus \text{Pserver}$

在常见的口令认证中,对用户口令 Puser 进行哈希处理后,将它的哈希值存放到口令文件中。而在 CSPS 中,不是计算 Puser 的哈希值,而是计算 Password 的哈希值。这样,每个用户的用户名和 Password 的哈希值组成了口令文件。

根据式(1),由于 Pserver 的存在,即使在口令文件中两个 Password 的哈希值相同,这两个用户相应的用户口令 Puser 一般也不同。Pserver 比盐更前进了一步。在目标猜测攻击^[10]中,攻击者获得口令文件之后,大的盐、甚至多个盐都不能阻止攻击者破解目标用户的口令。而在 CSPS 方案中, Pserver 是由认证系统随机生成的服务器口令,是服务器的秘密,攻击者无法获取这个秘密,因此攻击者无法通过目标猜测攻击破解目标用户的口令。Pserver 可看作是为每个用户设置的一个秘密的 pepper^[11]。

为了抗击线下攻击, CSPS 采用 balloon 这样的贵哈希函数^[12]用于口令哈希算法。

1.2 方案部署

Puser 是用户的秘密, Pserver 是服务器的秘密。

用密码学的方法来保持 Pserver 的秘密状态是恰当的。当用户注册账号时,认证系统先为用户生成 Pserver,接着用密钥加密 Pserver,再将 Pserver 的密文和用户名存入文件,然后丢弃 Pserver。等用户下次登录时,系统通过用户名搜索文件,并解密 Pserver。

在部署时,哈希的 Password 文件、加密的 Pserver 文件和密钥非常重要。在图 2 中,它们分别用 A、B、C 标示。密钥必须存放在安全的地方,因为它是服务器的秘密。在图 2 中,因为去除了表示 Puser 中每个字符的 ASCII 码的最高位,所以 Puser 的长度是 $7m$ 位,而不是 $8m$ 位。

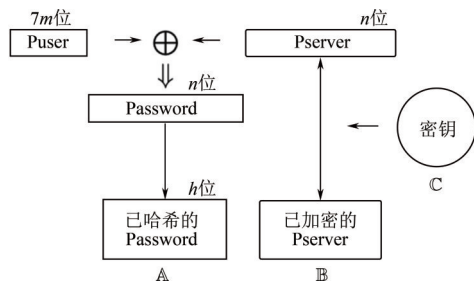


图2 CSPS 略图

一个可行的方法是以分布式的方式存储它们。可以将它们存放在 3 个安全级别不同的计算机系统里。当用户注册或登录账号时,这 3 个计算机系统可以相互通信。在图 3 和图 4 中,这 3 个计算机系统分别以系统 A、系统 B、系统 C 标示。

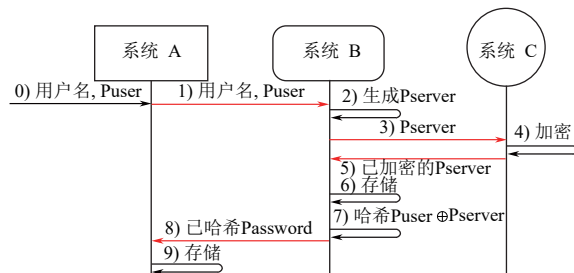


图3 用户注册过程

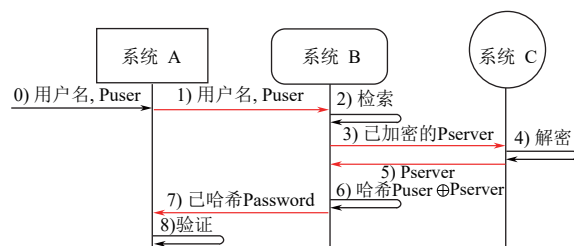


图4 用户登录过程

网络信道安全假设如下。这 3 个系统中,系统 A 只与系统 B 通信,系统 B 分别与系统 A、系统 C 通

信,系统 A 和系统 C 相互之间没有任何直接通信。系统 A 是连接互联网的,用户的网络服务通常也部署在系统 A 上。系统 A 最易受到各种网络攻击。系统 B 和系统 C 不直接与互联网相连,系统 C 在物理上只与系统 B 相连,系统 B 在物理上只与系统 A、系统 C 相连。通过互联网,用户可以访问系统 A,但不能访问系统 B 和系统 C。在系统 A 和系统 B、系统 B 和系统 C 之间可以同时并发多个请求和响应,但每个请求和响应的类型和长度都受到严格限制。

用户注册的过程如下。0)当 Alice 注册网络服务时,她通过互联网将用户名和口令 Puser 输入系统 A。1)系统 A 将她的用户名和 Puser 传给系统 B。2)系统 B 为 Alice 随机生成 Pserver。3)系统 B 将 Pserver 传给系统 C。4)系统 C 用密钥加密 Pserver。5)系统 C 将加密后的 Pserver 传回系统 B。6)系统 B 存储加密后的 Pserver。7)系统 B 根据式(1)计算 Password 并将 Password 哈希处理。8)系统 B 将哈希的 Password 传给系统 A。9)系统 A 将哈希的 Password 存储在口令文件中。

用户更改口令的过程与图 3 中用户注册的过程类似,此处不再详细叙述。

用户登录的过程如下。0)当 Alice 登录网络服务时,她通过互联网将用户名和 Puser 输入系统 A。1)系统 A 验证 Alice 的账号,并将她的用户名和 Puser 传给系统 B。2)系统 B 检索磁盘文件,找到加密的 Pserver。3)系统 B 将加密的 Pserver 传给系统 C。4)系统 C 解密 Pserver。5)系统 C 将解密的明文 Pserver 传回系统 B。6)系统 B 接收明文 Pserver,并根据式(1)计算 Password,然后哈希 Password。7)系统 B 将哈希的 Password 传给系统 A。8)系统 A 接收这个哈希的 Password,并将它与口令文件中相应的哈希值进行比较。如果两者相同,Alice 成功登录;否则,登录失败。

在系统 A、系统 B、系统 C 中,有两类通信信息,即数据流和控制信号。

1)数据流:只有哈希的 Password 从系统 B 传回系统 A。除此之外,其他数据都不允许从系统 B 传到系统 A。同时,只有用户的用户名和 Puser 从系统 A 传到系统 B。在系统 B 和系统 C 之间的数据流只有 Pserver 的密文或明文。系统间的数据流在图 3 和图 4 中用带箭头的红线表示。

2)控制流:为了协同认证用户身份,3 个系统需要相互发送控制指令。从系统 A 到系统 B 有 3

个控制信号:0 表示用户注册;1 表示用户登录;2 表示用户更改口令。从系统 B 到系统 C 有两个控制信号:0 表示加密;1 表示解密。从系统 B 到系统 A、从系统 C 到系统 B 只有响应没有控制信号。控制信号与数据流一起传输,如“0||用户名,Puser”。

系统 A 的主要工作是向用户提供网络服务。系统 B 主要负责用户身份认证过程中的一些重要工作。系统 C 存放密钥并提供加密、解密服务。系统 A 只做了少量的用户认证工作。

安全起见,建议系统 A、系统 B、系统 C 使用不同的操作系统,运行不同的应用程序。一般地,系统 C 或系统 B 应当具有高安全级别,系统 A 暴露在普通的互联网环境中。

将哈希的口令文件、加密的 Pserver 文件和密钥分别存储在系统 A、系统 B 和系统 C 中可以避免许多安全威胁。首先,盗取哈希口令文件或者入侵系统 A 的攻击者对系统 B 和系统 C 没有威胁。由于系统 A 和系统 B 之间的通信被严格限制,攻击者无法通过系统 A 获取加密的 Pserver 文件。其次,怀有恶意的内部工作人员无法获取密钥,因为密钥存放在具有最高安全级别的系统 C 上。只有少数人员具有访问系统 C 的权限。虽然让每个服务器口令 Pserver 都拥有属于自己的密钥会让系统 C 存储许多密钥而不是一个或一对密钥,但这是可能的,从安全角度讲,也是值得推荐的。

1.3 方案优点

现有的加盐哈希口令文件通常为每个用户口令 P 生成一个随机的盐 S ,然后将盐 S 和用户口令 P 串接级联成“ $S||P$ ”,最后将“ $S||P$ ”的哈希值和对应的用户名都存放在口令文件中。单向哈希函数仅保证完整性,而不保证保密性(私密性),因此,在理论上,“ $S||P$ ”的哈希值可能会泄漏用户口令 P 中的几个比特位^[13];在实践中,已经公开了诸多破解单向哈希函数的有效算法^[14]。对于现有将盐 S 与用户口令 P 简单串接级联的口令方案,即便对盐 S 进行额外保护(譬如盐加密),也仍然存在泄漏用户口令 P 中几个比特位的风险。词典攻击中,即使哈希值仅仅泄露用户口令 P 中 3 个比特位,口令猜测攻击者也有可能直接排除攻击词典中 7/8 的口令,提高口令猜测攻击的攻击效率。

与现有加盐哈希口令方案将盐与用户口令串接级联操作不同,CSPS 方案将加密保存的随机服务器口令 Pserver 与用户口令 Puser 进行异或运算,运算结果为 Password,然后对 Password 执行哈希算

法生成口令文件。不论 Puser 怎么取值, Password 总是一个均匀分布的随机数, 定理 1(见第 2 节)对此作了严格证明。由于线下攻击者不能获取 Pserver, CSPS 方案只可能泄露随机数 Password 中的几个比特位, 而不会泄露用户口令 Puser 中的比特位。可见 CSPS 方案中的异或运算优于现有口令方案中简单的串接级联操作。图 5 对此作了形象化表示。

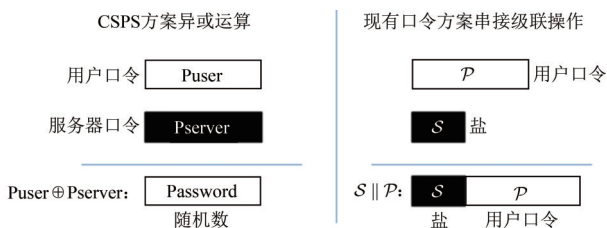


图5 CSPS方案与现有口令方案的比照

异或由于运算高效,并且具有自反、可逆等优良性质,符合加密和解密的需求,在对称加密算法中被广泛使用。在 CSPS 方案中也采用异或运算来操作 Puser 和 Pserver。CSPS 方案没有采用同或、按位与等方式操作 Puser 和 Pserver 的具体原因如下。

1) 同或和异或互为非运算。同或和异或具有相似的运算性质。但是在数学上,异或运算实际上就是伽罗瓦域 GF(2) 中的加法运算,比同或具有先天优势。由于历史原因,在硬件上实现异或运算比实现同或运算更便宜、更快、更省电。大多数高级编程语言都具有异或算子,但没有同或算子;类似地, x86 编程语言具有异或操作指令,但缺少同或运算指令。综上, CSPS 方案没有采用同或。

2) 按位与运算“ $a \& b = c$ ”,可由结果 c 的取值推断参与运算的 a 、 b 不同取值的不同概率。按位与不具有异或运算自反、可逆等性质。CSPS 方案若使用按位与操作 Puser 和 Pserver,将会带来安全隐患。

2 安全分析

对线下口令猜测攻击者的行为和目标建模。

攻击行为 假定线下口令猜测攻击者成功盗取哈希口令文件,但是不能盗取存储在高安全设备上的有关服务器口令的秘密数据(密钥或密文),拥有足量的计算设备和充分的计算时间,计算能力不受限制,既可以对盗取的哈希口令文件发起词典攻击,也允许对盗取的哈希口令文件实施暴力破解。接下来的定理和推论建立在上述攻击行为的基础上。

攻击目标 通过线下攻击猜中用户口令。

定理 1 若随机生成的服务器口令 Pserver 均匀分布,那么任取用户口令 Puser, $\text{Password} = \text{Puser} \oplus \text{Pserver}$ 也是一个均匀分布的随机数。

证明:记 w 为用户口令 Puser, K 为服务器口令 Pserver, c 为 Password。

先计算对任意 c 和 w , 概率 $\Pr[\text{Password} = c \mid \text{Puser} = w]$ 的值。 $\Pr[\text{Password} = c \mid \text{Puser} = w] = \Pr[c = w \oplus K] = \Pr[K = c \oplus w] = 2^{-n}$, 由于 K 是随机选择并均匀分布的, 因此上述最后一个等式成立, 其中 n 为服务器口令 Pserver 的位长。

任给一个 Puser 的概率分布 $\mathcal{M}$, 对于任意一个 c , 有: $\Pr[\text{Password} = c] = \sum_{w \in \mathcal{M}} (\Pr[\text{Password} = c \mid \text{Puser} = w] \cdot \Pr[\text{Puser} = w]) = 2^{-n} \cdot \sum_{w \in \mathcal{M}} \Pr[\text{Puser} = w] = 2^{-n}$ 。

综上,任给 Puser 一分布,任一 Password 概率都是 2^{-n} , 因此, Password 是均匀分布的随机数。证毕。

线下口令猜测攻击者盗取哈希口令文件, 式(1)中的 Password 的哈希值存放在这个口令文件中。线下攻击者拥有关于 Puser 的攻击词典。注意到 Pserver 和 Puser 相互独立, 又任一 Pserver 出现的概率都是 2^{-n} , 根据定理 1, 不论 Puser 取何值, 任一 Password 出现的概率均为 2^{-n} 。从而, 线下攻击者不能从现有的关于用户口令 Puser 的攻击词典中得出关于 Password 的攻击词典。同时, 因为任一 Password 出现的概率都相同, 所以线下攻击者也不能建立关于 Password 的新词典。综上, 可得推论 1。

推论 1 线下口令猜测攻击者无法对 CSPS 方案中的哈希口令文件实施词典攻击。

线下攻击者的计算能力不受限制。根据式(1), 线下攻击者可以为每一对可能的 Puser 和 Pserver 计算 Password, 然后计算每个 Password 的哈希值, 并将计算得到的哈希值跟哈希口令文件中的哈希值进行比较。然而, 攻击者无法盗取服务器口令的秘密数据, 因此不妨假设攻击者暴力破解时选取的 Pserver 是随机且均匀分布的。根据定理 1, 不管攻击者对用户口令 Puser 作什么样先验的猜测, Password 总是均匀分布的。从而, 攻击者从盗取的 Password 哈希表中不但无法确定真实 Puser, 而且无法高概率猜中, 或得到一个足以登录的等价凭证, 或缩小搜索空间显著提高在线攻击成功率。综上, 可得推论 2。

推论 2 线下口令猜测攻击者无法对 CSPS 方案中的哈希口令文件实施暴力破解。

由于 Pserver 的存在, Password 哈希泄露不增加跨站口令 Puser 被线下攻击的风险, 用户可跨站复用 Puser。但仍不建议用户跨站复用口令。

这里仅考虑线下口令猜测攻击, 不考虑侧信道攻击、信道监听、篡改、重放等利用泄露信息行为的攻击情形^[15], 也不考虑可证明安全的局限性^[16]。

3 对抗在线攻击

CSPS 方案中, 对在线词典攻击提高警惕尤为重要。好记的短口令在在线词典攻击面前极为脆弱。网络服务提供商有必要不间断检测可疑登录请求以便防范在线攻击, 应记录所有可疑的登录请求, 设置登录规则以便让用户自己警惕并防范在线攻击。

规则 1: 如果在 24 h 内用户口令被输错 4 次以上, 那么这个账号将被锁住 1 d, 1 d 后自动解锁。即便攻击者专门攻击一个特定的用户账号, 他们在 1 a 之内最多也只能猜测 $5 \times 365 = 1\,825$ 次。如果攻击者的词典不少于 10^5 条, 攻击者需要超过 50 a 的时间才能完成攻击。一般地, 这是安全的。

规则 2: 用户的用户名可以分成两部分, 登录名和账号名。用户登录时, 输入登录名。在口令文件中不直接存储登录名, 而存储登录名的哈希值。账号名不用于登录账号。账号名用以标识用户, 显示在网络服务的相关网页上。账号名直接存储在哈希口令文件中。这样, 口令文件的结构为: |账号名|哈希的登录名|哈希的 Password|。

规则 3: 一旦用户账号因输错口令而被锁住, 在解锁后, 网络服务提供商强制要求用户必须更换登录名和口令。从用户注册账号开始, 或者从用户上次修改口令开始算起, 当口令累计输错一定次数 (如 10 次) 后, 网络服务提供商提醒用户修改口令。

规则 1 受到银行 ATM 机的启发。ATM 取款机一般限制用户只能输错 4 次银行卡口令。因此一个用户有 5 次机会输入正确口令。如果连着输错 5 次, 银行帐户会被冻结。在网络服务中, 不能永久性地冻结用户账号, 否则会招致拒绝服务攻击。

规则 2 增加了在线攻击的难度。攻击者在线下攻击之前几乎不能发起在线攻击, 因为只知道用户账号名, 不知道登录名, 很难发起有效的在线攻击。为了从哈希值回溯用户确切的登录名, 线下攻击者可能会被哈希运算所需要付出的代价挫败, 毕竟贵哈希函数计算代价很高。

规则 3 假定在线攻击者同时猜测用户的登录名和口令。由于用户选择好记的短口令, 攻击者的口令词典可以很小。假如攻击者的口令词典条目小于 2 000, 那么攻击者在 1 a 不到的时间内有可能在线词典攻击中获得成功。为了避免发生这样的情况, 一旦攻击者发起在线攻击, 用户应当及时更换登录名和口令。显而易见, 由于目前没有用户登录名的攻击词典, 对于在线攻击者来说, 猜测用户的登录名比猜测用户的口令要难得多。遵循规则 3, 在线攻击者无法持续地针对特定用户发起目标猜测攻击。

用户短口令的最短长度可取 4 位: 银行卡口令为 6 位, 口令空间规模为 10^6 。一个 4 位长的用户短口令的空间规模为 95^4 。由于 $95^4 > 10^6$, 一个 4 位长的用户短口令可以达到银行卡口令的安全强度。

此外, CAPTCHA^[17]是防范在线机器人攻击的常用技术, 还可使用金融^[18]、用户登录 IP 地址、地理位置等手段对抗在线攻击。

4 讨论

在用户登录时的响应延迟方面, CSPS 方案除了要将 Puser 和 Pserver 异或后进行哈希运算外, 还要对加密的 Pserver 解密; 而一般的口令认证方案只需要哈希运算。CSPS 方案除了需要在口令文件中验证已哈希的 Password 之外, 还需要在加密的 Pserver 文件中查找已加密的 Pserver; 而一般的口令认证方案只需要在口令文件中验证一下哈希值。考虑到系统 B 和系统 C 在服务器端仅用于身份认证, 都由网络服务提供商部署, 通常与系统 A 距离不远, 系统 B 与系统 A、系统 C 与系统 B 之间的信道通信信号质量可以得到保证, 因此忽略 3 个系统相互之间在信道通信的延迟。这样, 较之一般的现有口令认证方案, CSPS 方案在用户登录时的响应延迟主要源于 Pserver 的查找和解密。在已加密的 Pserver 文件中查找 Pserver 所需时间与一般口令认证方案在口令文件中验证哈希值的时间相同, 在 CSPS 方案中由查找 Pserver 所带来的响应延迟完全可以接受。解密 Pserver 所需时间依赖于加密算法, 但由于 Pserver 是一个长度有限的字符串, 一般认为解密所需时间小于贵哈希函数所需的哈希操作时间, 因此解密 Pserver 所需时间也完全可以接受。类似地, 用户注册时的响应延迟也可接受。

在存储空间代价方面, CSPS 方案除哈希的

Password文件需要存储外,还需存储加密的Pserver文件和密钥。今天存储设备价格相对低廉,加密的Pserver文件和密钥带来的存储空间方面的代价可以不予考虑。不过,因为不能让攻击者获取加密的Pserver文件或密钥,所以需要网络服务提供商安全存储加密的Pserver文件或密钥,这需要网络服务提供商付出额外的代价。

安全性方面,系统B和系统C不直接与互联网相连,攻击者无法对系统B或系统C发起在线攻击。系统B与系统A之间的通信严格受限,攻击者通过SQL注入只可能获取Password文件上的数据,不可能获取加密的Pserver文件上的数据,更不可能获取密钥。系统B只向系统A发送已哈希的Password,因此系统B不会向系统A泄露其他信息。

5 结束语

用户可在CSPS方案中选择好记的短口令。若网络服务提供商额外提供高安全级别的计算机系统,CSPS方案作为一种分解口令的方法,可以充分对抗线下口令猜测攻击。CSPS方案中用户口令与服务器口令的异或运算优于现有口令方案中用户口令与盐简单的串接级联操作。CSPS方案中,线下攻击者不能成功实施词典攻击或暴力破解。

参考文献:

- [1] 汪定,朱留富.基于国密SM2支持动态口令恢复的多因素认证方案[J].计算机学报,2025,48(9):2241-2262.
- [2] XU M, HAN W, YU J T, et al. On the account security risks posed by password strength meters[C]//Proceedings of the 20th ACM Asia Conference on Computer and Communications Security. New York, USA: ACM, 2025: 1582-1599.
- [3] ZOU Y K, AN M X, WANG D. Password guessing using large language models [C] //Proceedings of the 34th USENIX Security Symposium. Berkeley, USA: USENIX Association, 2025:7799-7818.
- [4] 邹云开,汪定.口令猜测研究进展[J].密码学报(中英文),2024,11(1):67-100.
- [5] 程鹏文,颜学雄,章明.7-Zip加密文件快速口令恢复算法研究[J].信息工程大学学报,2024,25(1):45-51.
- [6] MARJANOV T, HUTCHINGS A. SoK: digging into the digital underworld of stolen data markets [C]//Proceedings of the 2025 IEEE Symposium on Security and Privacy. Piscataway, USA: IEEE, 2025:1-18.
- [7] HOU Z D, FAN T W, DUAN F, et al. How to design secure honey vault schemes [C]//Proceedings of the 2025 ACM Conference on Computer and Communications Security. New York, USA: ACM, 2025:4319-4333.
- [8] JUELS A, RIVEST R L. Honeywords: making password-cracking detectable [C]//Proceedings of the 2013 ACM Conference on Computer and Communications Security. New York, USA: ACM, 2013:145-160.
- [9] WANG D, CHENG H B, WANG P, et al. A security analysis of honeywords [C]//Proceedings of the 2018 Network and Distributed System Security Symposium. San Diego, USA: Internet Society, 2018:1-16.
- [10] XIE Z J, ZHANG M, YIN A Q, et al. A new targeted password guessing model [C]//Proceedings of the 25th Australasian Conference on Information Security and Privacy. Cham, Switzerland: Springer, 2020:350-368.
- [11] PAL B, DANIEL T, CHATTERJEE R, et al. Beyond credential stuffing: password similarity models using neural networks [C]//Proceedings of the 2019 IEEE Symposium on Security and Privacy. Piscataway, USA: IEEE, 2019: 417-434.
- [12] BEAME P, KORNERUP N. Cumulative memory lower bounds for randomized and quantum computation[J]. ACM Transactions on Computation Theory, 2025,17(3):1-33.
- [13] DAMGÅRD I B. A design principle for hash functions [C]//Proceedings of the 9th Annual International Cryptology Conference. Cham, Switzerland: Springer, 1989: 416-427.
- [14] AHN J, HUSSAIN R, KANG K, et al. Exploring encryption algorithms and network protocols: a comprehensive survey of threats and vulnerabilities [J]. IEEE Communications Surveys and Tutorials, 2025,27(6):3587-3614.
- [15] 黄海,唐新琳,吴金明,等.基于CLRM模型的侧信道攻击[J].密码学报(中英文),2025,12(2):414-428.
- [16] BELLARE M, ROGAWAY P. Random oracles are practical: a paradigm for designing efficient protocols [C] //Proceedings of the 1st ACM Conference on Computer and Communications Security. New York, USA: ACM, 1993: 62-73.
- [17] TEOH X, LIN Y, LI S Q, et al. Are CAPTCHAs still bot-hard? generalized visual CAPTCHA solving with agentic vision language model [C] //Proceedings of the 34th USENIX Security Symposium. Berkeley, USA: USENIX Association, 2025:3747-3766.
- [18] GHOSH T, HASAN M M, GUDLA C, et al. Robust password security: a method in preventing guessing attacks and authenticating users [C]//Proceedings of the 8th International Conference on Information and Computer Technologies. Piscataway, USA: IEEE, 2025:49-54.

(编辑:邵 华)