计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200500144

口令 Zipf 分布对相关安全协议的影响分析

董奇颖　单　轩　贾春福
南开大学网络空间安全学院　天津 300350
天津市网络与数据安全技术重点实验室　天津 300350
(dqy@mail.nankai.edu.cn)

摘　要　身份认证是确保网络与信息系统安全的第一道防线,口令则是最普遍的身份认证方式。现有研究通常假设用户构造的口令服从均匀分布,然而,最新的研究表明,口令服从 Zipf 分布,这意味着目前大部分口令相关安全协议都低估了攻击者优势,并不能达到所声称的安全性。针对上述问题,文中以 Gjøsteen 等提出的基于口令的签名(Password-Based Signatures,PBS)协议以及 Jarecki 等提出的口令保护秘密共享(Password-Protected Secret Sharing,PPSS)协议为典型代表,从口令服从 Zipf 分布这一基本假设出发,分析了这两个协议的安全性证明缺陷,并重新定义了其安全性。同时,文中给出了对上述两个协议的改进:对于 PBS 协议,重新计算了攻击者优势,并通过限制攻击者猜测次数和委托可信第三方保管密钥,使得改进后的 PBS 协议可以抵御恶意攻击者仿冒一般用户的攻击,以及恶意服务器猜测用户口令并伪造签名的攻击;对于 PPSS 协议,基于诱饵口令思想,在服务器端设置了 Honey_List 以检测并阻止在线口令猜测攻击。

关键词:Zipf 分布;口令相关安全协议;安全性证明;可信第三方;诱饵口令思想
中图法分类号　TP309

Impact of Zipf's Law on Password-related Security Protocols

DONG Qi-ying,SHAN Xuan and JIA Chun-fu
College of Cyber Science,Nankai University,Tianjin 300350,China
Tianjin Key Laboratory of Network and Data Security Technology,Tianjin 300350,China

Abstract　Identity authentication is the first line of defense for the security of networks and information systems,and password is the most common method of identity authentication. Researches usually assume that user-constructed passwords obey uniform distribution. However,recent studies found that passwords obey Zipf's law,which means that most password-related security protocols underestimate the advantage of an attacker and thus fail to achieve the claimed security. In response to the above problem,first of all,Password-Based Signatures (PBS) protocol proposed by Gjøsteen,et al. and Password-Protected Secret Sharing (PPSS) protocol proposed by Jarecki,et al. are taken as typical representatives. Based on the basic assumption that passwords obey Zipf's law,the security proofs of these two protocols are demonstrated to be flawed,and the security is redefined. Furthermore,the improvements to the two protocols are given respectively. In improved PBS protocol,an attacker's advantage is recalculated. By limiting the guess number of an attacker and entrusting a trusted third party to keep the key,the protocol can prevent a malicious attacker from disguising a legitimate user,and can prevent a malicious server from guessing a user's password and forging the signature. In improved PPSS protocol,a Honey_List is set on the server side based on honeywords to detect and prevent online password guessing attack.

Keywords　Zipf's law,Password-related security protocols,Security proof,Trusted third party,Honeywords

1　引言

口令是信息系统中最流行的身份认证机制,口令安全直接关系到整个应用系统和用户个人信息的安全[1]。目前口令相关的安全协议都对口令分布作出不合实际的假设:口令服从均匀分布[2-3]或其他不合理的分布[4],甚至干脆规避这个问题[5]。这使得目前大部分口令相关安全协议都低估了攻击者的优势,会引起严重的安全性和可用性问题。

Zipf 定律最初用于刻画自然语言中的单词排名与单词出现频率之间的关系[6]。Wang 等[7]通过对 14 个大规模真实

到稿日期:2020-05-28　返修日期:2020-08-07　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61972215)
This work was supported by the National Natural Science Foundation of China (61972215).
通信作者:贾春福(cfjia@nankai.edu.cn)

口令集进行拟合，证明了口令也服从 Zipf 分布，并且将整个口令集的分布的 CDF-Zipf 模型[7]定义为：

$$F_r = C' \cdot r^{s'} \tag{1}$$

其中，F_r 为排名至 r 的口令的累积频率；C' 和 s' 是取决于口令集的常数，可以通过线性回归计算得到。

很多口令相关的安全协议[2-3]假设用户 U 的口令 pw_U 从大小为 $|\mathcal{D}|$ 的字典 $\mathcal{D}$ 中均匀随机选取得到。基于该假设所构建协议的安全模型被定义为[8]：协议 P 是一个安全的基于口令的单因子认证协议，对于所有大小为 $|\mathcal{D}|$ 的口令字典和所有在概率多项式时间内至多进行 $Q(k)$ 次在线猜测的攻击者 A，存在一个大小可忽略的函数 $\varepsilon(k)$，且满足下列条件：

$$Adv_{A,P}(k) \leqslant Q(k)/|\mathcal{D}| + \varepsilon(k) \tag{2}$$

其中，$Adv_{A,P}(k)$ 是 A 攻击 P 时的优势。

通常情况下，用户产生的有效口令的空间大小约为 $2^{20} \sim 2^{21}$[7]。由于攻击者每次随机选取口令集中的口令进行猜测，因此一次在线猜测的成功率 $1/|\mathcal{D}|$ 不高于 $1/2^{21} \sim 1/2^{20}$。然而，Wang 等[7]发现口令实际上服从 Zipf 分布。真实的攻击场景是，一个漫步猜测攻击者 A 会按照口令出现概率递减的顺序尝试猜测，而非随机猜测。在极端情况下，若 A 获得了整个明文口令集，则可直接获得最优的猜测顺序。

不失一般性，本文采用两个典型的安全协议作为示例，以阐明口令服从 Zipf 分布[7]对相关安全协议的影响，即 Gjøsteen 等[2]提出的 PBS 协议以及 Jarecki 等[3]提出的 PPSS 协议。

数字签名的传统实现方法是将签名密钥存储在智能卡等受特殊保护的计算设备内，然后智能卡与计算设备交互，并签署其提交的文件。然而，绝大多数计算设备（尤其是手机和平板电脑）并没有配备相应的读卡器。针对此问题，Gjøsteen 等[2]提出了 PBS 协议，使用口令来替换用户私钥。在口令服从均匀分布的假设下，PBS 协议[2]的安全性证明中给出的攻击者优势 $Adv_{A,P}(k)$ 偏小。本文指出了 PBS 协议[2]的安全性证明的缺陷并给出了正确的证明。

为了保护用户秘密信息免受存储设备故障的影响，一个可行的方案是使用秘密共享方案在一组服务器上分发秘密信息。Bagherzandi 等[9]提出了参数为 (t,n) 的 PPSS 协议，该方案允许用户将存储的秘密信息 s 分为 n 份，分存到 n 个远程服务器上，用户使用其持有的唯一口令 pw，与至少 $t+1$ 个服务器进行交互，即可正确重构 s。值得注意的是，Bagherzandi 等[9]提出的 PPSS 协议认为秘密仅是一条消息，而 Jarecki 等[10]将这一定义扩大化，认为秘密可以是一个随机密钥，这个密钥可以被用来实现加密、签名或密钥交换协议等，从而使得 PPSS 协议的应用更加广泛。在上述理论的基础上，Jarecki 等[3]改善了计算和通信的复杂性，提出了迄今为止最有效的 PPSS 协议，并在通用组合模型下证明了该方案的安全性。基于 Wang 等[11]的工作，本文指出了 PPSS 协议[3]的安全性定义的缺陷。

针对上述协议存在的缺陷，本文重新计算了 PBS 协议[2]中的攻击者优势，并通过限制攻击者猜测次数和委托可信第三方保管密钥，使得改进后的 PBS 协议可以抵御恶意攻击者仿冒一般用户以及恶意服务器猜测用户口令并伪造签名的攻击。进一步地，本文受 Wang 等[12]应用诱饵口令思想（honeywords）改进口令安全协议的启发，在 PPSS 协议[3]的服务器端设置了 Honey_List，以检测并阻止在线口令猜测攻击。

2 PBS 协议的安全性证明缺陷和实现

PBS 协议[2]的实现分为 7 个阶段：参数生成 $Setup$，用户密钥生成 $KeyGen_U$，服务器密钥生成 $KeyGen_S$，发布签名 $Issue$，请求 $Request$，去盲 $Unblind$ 和验证 $Verify$。

PBS 协议[2]假设口令服从均匀分布，即 $pw \leftarrow_R \mathcal{D}$。基于此，Gjøsteen 等[2]定义了 PBS 协议[2]的盲性和不可重建性。本文指出了上述性质的证明过程中存在的缺陷，并重新进行了论证。原 PBS 协议[2]及改进协议（见第 3 节）中使用的符号及其含义如表 1 所列。

表 1 本文中 PBS 协议的符号及其含义

Table 1 Symbols and their meanings in PBS protocol in this paper

符号	含义
U	用户
S	服务器
S_A	恶意服务器
A	攻击者
pw	U 的口令
m	待签名消息
U_{ID}	U 的身份
ρ	U 的签名请求
k	安全参数，RSA 加密算法中的参数 N 满足 $2^{k-1} < N < 2^k$
(e,N)	RSA 加密算法公钥
(d,N)	RSA 加密算法私钥
r	U 选取的随机数
$G(pw)$	映射，$pw \rightarrow \{0,1,2,\cdots,2^{2k}-1\}$
$\tilde{\sigma}$	S 生成的盲签名
sk	S 的私钥
σ	去盲后的签名
$H(\cdot)$	安全哈希函数
$\sigma \sim m$	U 最近一次签名产生的签名～消息对
$flag$	标识 $\sigma \sim m$ 是否被验证过的标志位

2.1 PBS 协议的安全性证明缺陷

本节以基于 RSA 的 PBS 协议[2]为例，在口令服从 Zipf 分布[7]这一前提下，讨论其安全性证明中存在的缺陷。

2.1.1 盲性证明

PBS 协议[2]的盲性的含义为，攻击者 A 只能看到用户 U 的签名请求。

$$\rho \leftarrow H(m) r^e \bmod N \tag{3}$$

由于 r 是用户随机选取的值，ρ 在 $\mathbb{Z}_N^*$ 中分布广泛，所以此过程不会泄露任何关于 m 的信息。

虽然 PBS 协议[2]的盲性证明假设口令服从均匀分布，然而，A 获取到 ρ 后，无论是直接通过 ρ 还是计算盲签名 $\tilde{\sigma} \leftarrow \rho^{sk} \bmod N$，都无法得知 $\tilde{\sigma}$ 和 m 之间的对应关系。因此，对于盲性，尽管证明过程中存在缺陷，但结论是正确的。

2.1.2 不可重建性证明

PBS 协议[2]的不可重建性的含义为：如果攻击者 A 的优势 $Adv_{PBS,A}^{nonframe}(k)$ 超出获取口令明文 pw 的概率的数值可以忽略不计，即满足：

$$Adv_{PBS,A}^{nonframe}(k) = \varepsilon + \theta \tag{4}$$

那么该方案是不可重建的，其中 θ 是一个可以忽略不计的小量，ε 是猜测 k 次能够成功获得 pw 的概率。

不可重建性体现了服务器无法代替用户签名。然而在口令服从 Zipf 分布[7]的假设下，攻击者 A 或恶意服务器 S_A 猜出口令的概率大幅提升，相应的 $Adv_{PBS,A}^{nonframe}(k)$ 也会变大，证明如下：

A 猜得 pw 后，对于 m，A 可以自行选取 r，根据式(3)代替 U 发出 ρ，无须通过请求阶段查询 ρ。同时，A 可根据其拥有的服务器密钥 sk 计算盲签名 $\tilde{\sigma} \leftarrow \rho^{sk} \bmod N$。在去盲阶段，$A$ 利用记录的 r 和猜得的 pw，可以计算去盲后的签名：

$$\sigma \leftarrow \tilde{\sigma} \rho^{G(pw)} r^{-1} \bmod N \tag{5}$$

该签名可以通过式(6)验证：

$$H(m) \equiv \sigma^e \bmod N \tag{6}$$

因此，A 只要猜出 U 的 pw 就可攻击成功。在口令服从 Zipf 分布[7]这一前提下，A 猜出 pw 的概率远大于假设口令均匀分布的概率，所以不可重建性证明中 $Adv_{PBS,A}^{nonframe}(k)$ 的定义并不合理，即不能忽略 $Adv_{PBS,A}^{nonframe}(k)$ 超出 A 获取到 pw 概率的数值。

2.2 基于 RSA 的 PBS 协议的实现

基于 RSA 的 PBS 协议[2]的实现过程存在一个缺陷：使用式(5)计算去盲后签名 σ 时，需要对用户选取的 r 求逆，实际上该步骤并非必要。若将去盲过程修改为：

$$\sigma_1 \leftarrow \tilde{\sigma} \rho^{G(pw)} \bmod N \tag{7}$$

即可避免求逆运算。

原验证阶段计算式(6)时需要传入参数 (m, σ, e)，由于式(7)中有：

$$\sigma_1 \equiv H(m)^d r \bmod N \tag{8}$$

故验证过程只需计算：

$$\rho' \equiv (\sigma_1)^e \bmod N \tag{9}$$

并验证 $\rho' = \rho$ 是否成立即可（ρ 在请求阶段已经计算得到）。经上述修改，无须增加传入参数即可完成验签过程。

3 PBS 协议的改进

PBS 协议[2]的缺陷源于口令本身的脆弱性，因此可以从两个角度进行改进：1)阻止用户的脆弱口令行为，改变口令分布；2)限制攻击者的尝试次数。阻止用户的脆弱口令行为可能给用户增加额外的构造和记忆口令的负担，所以本文通过限制攻击者尝试次数来改进 PBS 协议[2]。改进后的 PBS 协议[2]委托可信第三方保管签名所需的重要参数，服务器和用户均须向可信第三方证明其非恶意，才能生成正确的签名，这样可以避免服务器掌握过多信息从而代替用户产生签名。

一般用户和攻击者的行为特征存在明显差异。理想情况下，一般用户使用加密口令来对某条消息产生签名并通过验证，只需向服务器提交一次请求。在不知晓用户口令的情况下，攻击者若要对某条消息产生可通过验证的签名，则需要伪装成特定用户，并进行大量尝试。本文利用这一差异区分一般用户和攻击者。理想情况下，一般用户每次提交的请求都能够产生对应的签名，并且签名和消息的内容向可信第三方公开。在保证协议安全性不受影响的情况下，本文利用原 PBS 协议[2]向验证者公开的签名和消息来识别用户身份。

由于篇幅限制，本文以基于 RSA 的 PBS 协议[2]的实现为例，说明改进方案。

3.1 抵御恶意攻击者仿冒一般用户的攻击

用户 U 要想向服务器 S 证明其不是恶意攻击者 A，就需要证明其向 S 发送的签名请求 ρ 均得到了可验证的签名 σ。S 存储一个表单，记录 U 最近一次签名产生的签名～消息对 $\sigma \sim m$ 以及签名是否已被验证过的标志位 $flag$(1 表示已被验证，0 表示未被验证)。为了区分用户个体，初始化表单，令每一个 U_{ID} 对应的 $\sigma = \rho = flag = 0$。

在原 PBS 协议[2]中，U 在请求签名阶段只发送 ρ，而在本文修改后的协议中，U 还需发送 U_{ID} 以及最近一次签名产生的 $\sigma \sim m$(若为初次请求，则令 $\sigma = m = 0$)，以证明其不是 A。S 接收到请求后，首先验证 $flag(U_{ID} \sim \sigma \sim m)$。若 $flag(U_{ID} \sim \sigma \sim m) = 0$，则说明 U 并非 A，并令 $flag(U_{ID} \sim \sigma \sim m) = 1$，避免 $\sigma \sim m$ 被重复使用，然后 S 计算此次请求对应的盲签名 $\tilde{\sigma}$；若 $flag(U_{ID} \sim \sigma \sim m) \neq 0$，则视 U 为 A，不计算也不反馈盲签名 $\tilde{\sigma}$。U 得到反馈的 $\tilde{\sigma}$ 后，计算 σ，并将 $U_{ID} \sim \sigma \sim m$ 返送给 S，S 得到该值后确认 $\sigma \sim m$ 的对应关系，验证无误后再更新表单。图 1 展示了可抵御恶意攻击者仿冒一般用户攻击的改进的 PBS 协议[2]的交互过程，红色字体为改进内容。本文利用 2.2 节的方法修改了 PBS 协议[2]的去盲和验证过程，故图 1 中记录的签名～消息对为 $\sigma_1 \sim m$。

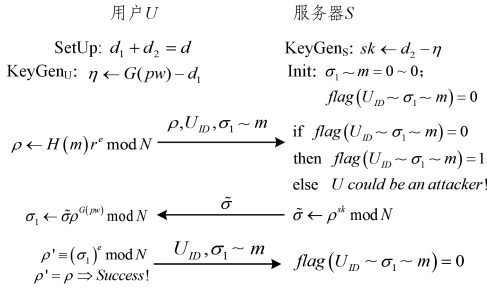

图 1 改进的 PBS 协议：可抵御恶意攻击者仿冒一般用户的攻击
(电子版为彩色)

Fig. 1 Improved PBS protocol: prevent a malicious attacker from disguising legitimate users

在初始状态下，U 在表单中的记录为 $\sigma \sim m = 0 \sim 0$，签名请求会消耗一次这个特殊的值对，并得到一个新的可用 $\sigma \sim m$。之后每产生一个新的签名请求，都要消耗 U 持有的一对未使用过的 $\sigma \sim m$。故在任何时刻 U 有且仅有一对可用的 $\sigma \sim m$，由此可以限制 A 进行口令猜测攻击时访问 S 的次数。即使 A 知道 U_{ID} 以及可用的 $\sigma \sim m$，当其猜测用户口令 pw 失败时，得到的 $\sigma \sim m$ 无法通过验证，即唯一可用的 $\sigma \sim m$ 被消耗，因此其再次访问 S 也不会得到相应反馈。

A 实施在线口令猜测攻击的成功率随着尝试次数的增加而提高。根据口令服从 Zipf 分布[7]，A 的最佳尝试顺序为概率递减顺序。Wang 等[11]将 A 的优势拟合为：

$$Adv_A(k) = C' \cdot Q(k)^{s'} + \varepsilon(k) \tag{10}$$

其中，$Adv_A(k)$ 随着猜测口令次数 $Q(k)$ 的增加而增大。然而，在本文对 PBS 协议[2]进行改进后，攻击者优势近似为一定值：

$$Adv_A(k)=C'\cdot 1^{s'}+\varepsilon(k)=C'+\varepsilon(k) \tag{11}$$

尽管以上方法有效降低了 $Adv_A(k)$，但会导致服务器 S 的权限过大。由于方案对 S 的依赖性较强，当 S 是恶意服务器 S_A 时，该方案则会完全失效。针对此种情况，本文提出进一步的改进方案。

3.2 抵御恶意服务器猜测用户口令并伪造签名的攻击

S_A 拥有服务器私钥 sk 且可以产生大量不能通过验证的签名，因此 S_A 只需猜测得到 pw 即可独立猜测和伪造签名，这种攻击难以被直接抵御。本文通过监测 S 对同一 U_{ID} 和 m 产生的签名次数来监测 S 是否恶意。可信第三方的引入，使得 S 在没有得到其允许的情况下很难进行有效签名。改进后的方案在保证 A 无法进行多次在线口令猜测的同时，抵御恶意服务器 S_A 猜测用户口令并伪造签名的攻击。本文将 3.1 节中的表单独立为一个限制访问权限的用户-签 名-消 息数据库（User-Signature-Message database，USM），存储于可信第三方。

改进后的 PBS 协议[2]主要添加了对 USM 的查询与更新操作以实现认证。USM 记录 U 最近产生的 $\sigma\sim m$ 和 $flag$，以及待分配给 U 和 S 的重要参数 e 与 sk。与 3.1 节类似，首先令每一个 U_{ID} 对应的 $\sigma=\rho=flag=0$。

U 向 S 请求签名时，首先需要向 USM 请求参数 e。U 向 USM 发送 U_{ID} 以及前一次交互产生的 $\sigma\sim m$ 和 pw（若是第一次请求，则令 $\sigma=m=0$），然后查询得到 e。当 U 计算出 ρ 之后，将其与 $U_{ID}\sim\sigma\sim m$ 发送给 S。

S 获得 ρ 后，首先需要向 USM 验证 $flag(U_{ID}\sim\sigma\sim m)$ 是否存在。如果 $flag$ 存在且 $flag=0$，说明 U 并非 A，并令 $flag(U_{ID}\sim\sigma\sim m)=1$；如果 $flag$ 存在且 $flag=1$，说明 $U_{ID}\sim\sigma\sim m$ 已经被使用过，U 可能为 A。此外，通过 USM，S 也可以得到 sk，并依据其计算 $\tilde{\sigma}$。U 得到反馈的 $\tilde{\sigma}$，计算出对应的 σ 并进行验证。若验证成功，则 USM 采用新的 $U_{ID}\sim\sigma\sim m$ 进行更新。

图 2 展示了可抵御恶意服务器猜测用户口令并伪造签名攻击的改进的 PBS 协议[2]的交互过程，红色字体为改进内容。本文利用 2.2 节的方法修改了 PBS 协议[2]的去盲和验证过程，故 USM 中记录的签名～消息对为 $\sigma_1\sim m$。

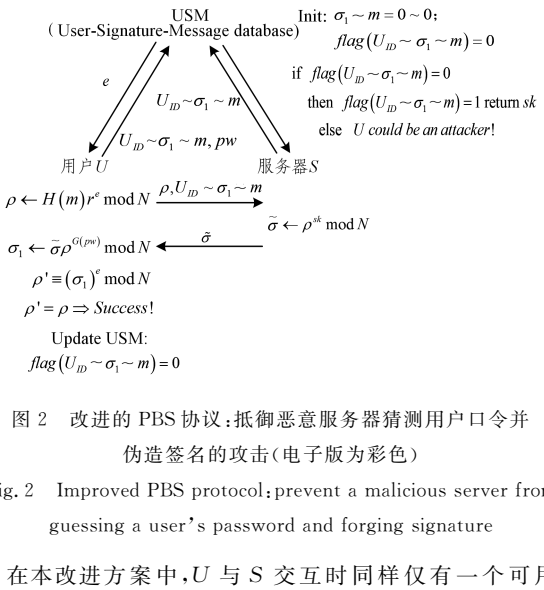

图 2 改进的 PBS 协议：抵御恶意服务器猜测用户口令并伪造签名的攻击（电子版为彩色）

Fig. 2 Improved PBS protocol: prevent a malicious server from guessing a user's password and forging signature

在本改进方案中，U 与 S 交互时同样仅有一个可用的 $\sigma\sim m$。S_A 每次猜测 pw 都会消耗其持有的 $\sigma\sim m$，这抵御了在线猜测用户口令并伪造签名的攻击。

S_A 在实施攻击时，需要同时掌握 pw 和 sk 才能够生成对消息 m 可用的签名。然而，S_A 至多拥有一个可用的 $U_{ID}\sim\sigma\sim m$，即至多有一次访问 USM 的机会。与 3.1 节中的方案相似，本改进方案的攻击者优势近似为常数 $Adv_A(k)=C'+\varepsilon(k)$。

4 PPSS 协议的安全性证明缺陷

类似于 PBS 协议[2]，PPSS 协议[3]也假设口令服从均匀分布，因此需要重新讨论安全性证明。原 PPSS 协议[3]和改进协议（见第 5 节）中使用的符号及其含义如表 2 所列。

表 2 本文中 PPSS 协议的符号及其含义

Table 2 Symbols and their meanings in PPSS protocol in this paper

符号	含义
(t,n)	用户至少与 $t+1$ 个服务器交互才能重构被分为 n 份的秘密信息
U	用户
U_{ID}	U 的身份
$S_i(i\in[1,n])$	服务器
A	攻击者
pw	U 的正确口令
pw'	U 在重构阶段使用的口令
$r_i(i\in[1,n])$	U 与 S_i 交互时使用的私钥
$k_i(i\in[1,n])$	S_i 的私钥
$s_i(i\in[1,n])$	U 通过 Shamir 密钥共享方案生成的共享信息
s	待共享的秘密信息，由 s_i 组成
ρ_i	U 与 S_i 交互时，对 pw 进行哈希运算后产生的中间结果
$e_i(i\in[1,n])$	U 生成的分存密钥
e	密钥，由 e_i 组成
$H_1(\cdot),H_2(\cdot),H_3(\cdot)$	安全哈希函数
K	由 s 解析得到；U 通过计算 $[r\Vert K]=H_3(0,s)$ 判断重构是否成功
C	U 通过计算 $C=H_3(1,pw,e,s,r)$ 得到的承诺
$\omega=(e,C)$	存储在每个服务器上，包含密钥 e 和承诺 C
$\oplus$	异或运算
$\Vert$	连接运算
Honey_List	存储 sweetwords 对应的承诺 C 的列表

基于口令服从均匀分布的假设，Jarecki 等[3]给出 PPSS 协议的安全性定义为：

$$Adv_A^{PPSS}(k)\leqslant(q_S+q_U)/|\mathcal{D}|+\varepsilon \tag{12}$$

其中，q_U 是攻击者 A 与用户 U 交互的次数上限，q_S 是攻击者与服务器 S 交互的次数上限。

基于口令服从 Zipf 分布的前提，Wang 等[11]给出在 PDF-Zipf 模型下，PPSS 协议[3]的安全性表达函数为：

$$\begin{aligned}Adv_{A,P}(k)&=\frac{C/1^s}{\sum_{i=1}^{|\mathcal{D}|}C/i^s}+\frac{C/2^s}{\sum_{i=1}^{|\mathcal{D}|}C/i^s}+\cdots+\frac{C/Q(k)^s}{\sum_{i=1}^{|\mathcal{D}|}C/i^s}\\&=\frac{\sum_{j=1}^{Q(k)}1/j^s}{\sum_{i=1}^{|\mathcal{D}|}C/i^s}+\varepsilon(k)\end{aligned} \tag{13}$$

在 CDF-Zipf 模型[11]下的攻击者优势应表示为式(10)。

Wang 等[11]通过对真实攻击者、基于口令均匀分布模型的攻击者、基于最小熵模型的攻击者、基于 CDF-Zipf 模型的攻击者以及基于 PDF-Zipf 模型的攻击者进行对比，发现式(10)更符合真实的攻击者优势。式(12)的错误直接导致了 PPSS 协议[3]中的 πPPSS 方案的博弈安全性证明错误。

5 PPSS 协议的改进

受 honeywords[13]启发，本文从提高抵抗在线口令猜测攻击的能力切入，增强 PPSS 协议[3]的安全性。honeywords[13]能够有效监测口令文件泄露并延缓攻击者 A 的在线猜测速率。该机制的核心是对每个用户 U 的账户都关联一些诱饵口令(honeywords)。即使 A 从服务器 S_i 窃取到口令文件，也必须要从一组由算法生成的 sweetwords(由若干 honeywords 和真实口令 pw 组成)中尝试找出 pw。理论上 A 不可区分 honeywords 和 pw，只能进行多次登录尝试[13]，此时 S_i 便会察觉到口令文件已经泄露。

Wang 等[12]将 honeywords 应用于口令相关的安全协议中，提出设置 Honey_List 列表来存储 sweetwords 的方法。本文对于 PPSS 协议[3]的改进参考了该方法，从而可以有效地预防和检测在线口令猜测攻击。

5.1 改进方案

在改进后的 PPSS 协议[3]中，S_i 为每个 U 都绑定了一个 Honey_List，用于存储 sweetwords。本文还设计了 Honey_List 填充策略，从而使 PPSS 协议[3]能够更好地检测并阻止在线口令猜测攻击，改进后的协议描述如下。

5.1.1 初始化阶段

(1)U 随机选取秘密信息 s，并通过 Shamir 密钥共享方案[14]生成 n 份共享信息$\{s_i\}$，$i=1,2,\cdots,n$，任取其中 $t+1$ 份即可恢复 s。同时，U 随机选取私钥 r_i 并计算 $a_i=(H_1(pw))^{r_i}$，将$((U_{ID},i),a_i)$发送给 S_i。

(2)S_i 收到用户的注册请求后，首先为每个 U 创建一个新的表项，设置 Honey_List=NULL。为避免服务器掌握过多信息，Honey_List 存储 sweetwords 对应的承诺 C，其条目数 num 的上限为 m。然后 S_i 计算 $b_i=(a_i)^{k_i}$ 并将其反馈给 U。

(3)U 收到来自 S_i 的 b_i 后，计算 $\rho_i=H_2(pw,b_i^{1/r_i})$，并使用 ρ_i 对 s_i 加密得到分存密钥 $e_i=s_i\oplus\rho_i$。令 $e=\{e_1,e_2\cdots,e_n\}$，$[r\parallel K]=H_3(0,s)$，进一步通过哈希运算得到承诺 $C=H_3(1,pw,e,s,r)$，将承诺连同分存密钥 $\omega_i=(e,C)$发送给 S_i。

5.1.2 重构阶段

(1)S_i 在收到 U 发送的$((U_{ID},i),H_1(pw')^{r_i})$后，首先检查 U 的身份 U_{ID} 是否合法；若合法则向 U 发送$((U_{ID},i,1),(H_1(pw')^{r_ik_i},i,\omega))$。

(2)若 U 与不同的 S_i 交互得到的 ω_i 不尽相同，则意味着后续重构 s 必然失败。反之，U 可以通过 $\omega_i=(e,C)$解析得到 e 和 C。与初始化阶段相似，U 计算得到 $\rho_i=H_2(pw',(H_1(pw')^{r_ik_i})^{1/r_i})$，然后 e 与$\{\rho_1,\rho_2,\cdots,\rho_{t+1}\}$进行解密运算即可得到 $t+1$ 个插入点$\{(1,s_1),(2,s_2),\cdots,(t+1,s_{t+1})\}$，进而重构秘密信息 s'。

(3)U 计算得到$[r\parallel K]=H_3(0,s')$与 $C'=H_3(1,pw',e,s',r)$。若 $C'=C$ 则说明重构的秘密信息 $s'=s$，反之则将 C' 存储于 Honey_List。

图 3 展示了改进后的 PPSS 协议[3]的重构阶段的交互过程，红色字体为改进内容。U 尝试使用口令 pw' 登录。若 $pw'=pw$，则通过认证并重构 s；若 $pw'\neq pw$，则根据 num 的大小执行不同的操作。当 $num<m$ 时，若 pw'对应的承诺 C' 在 Honey_List 中，则中止程序并警告“可能存在在线口令猜测攻击”，否则将 pw'对应的 C'写入 Honey_List 中；当 $num=m$ 时，中止程序并警告“尝试次数过多”。

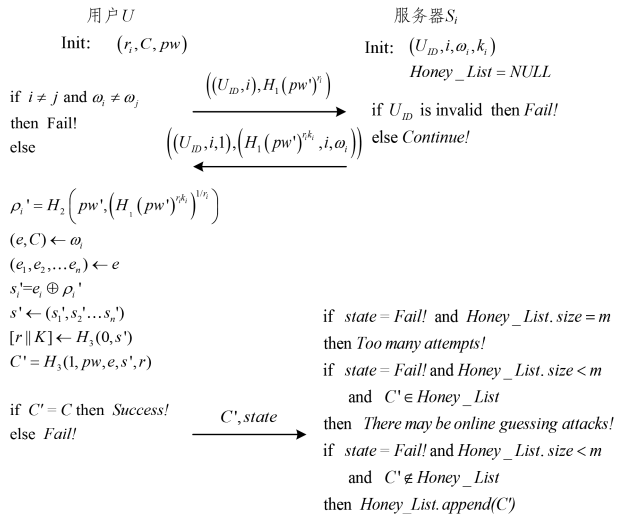

图 3 使用 honeywords 改进之后的 PPSS 协议的重构阶段

(电子版为彩色)

Fig. 3 Reconstruction phase of PPSS protocol improved by honeywords

5.2 改进方案的分析

理论上，A 不可区分明文口令中的 honeywords 和 pw[13]，因此在进行在线登录尝试时，A 很大概率会采用 honeywords。改进后的 PPSS 协议[3]在两种情况下会发出警告：1)$pw'\neq pw$ 且 $num=m$ 时。发生这种情况是因为 A 或 U 尝试登录 m 次均未正确输入口令。2)$pw'\neq pw$ 且 pw'对应的承诺C'存在于 Honey_List 中。发生这种情况的可能性较大，例如 A 尝试使用流行口令进行猜测攻击。

本文对 PPSS 协议[3]的改进保留了其本身在密钥交换、秘密信息存储等方面的安全性。同时，改进后的 PPSS 协议[3]根据不同的尝试登录次数和输入口令的出现情况(或流行程度)，采取不同的操作并发出警告，有效区分了在线口令猜测攻击的不同威胁程度和潜在可能。

结束语 基于口令服从 Zipf 分布[7]这一最新发现，本文对 PBS 协议[2]和 PPSS 协议[3]的安全性重新进行了分析，并给出了新的安全定义。本文还作出以下改进：对于 PBS 协议[2]，通过限制攻击者猜测次数和委托可信第三方保管密钥，使得改进后的协议可以抵御恶意攻击者仿冒一般用户的攻击以及恶意服务器猜测用户口令并伪造签名的攻击；对于 PPSS

协议[3],受 Wang 等[12]应用 honeywords 改进口令安全协议的启发,本文在服务器端设置了 Honey_List,以检测并阻止在线猜测攻击。

参 考 文 献

[1] BONNEAU J,HERLEY C,VAN O P C,et al. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes [C]// 2012 IEEE Symposium on Security and Privacy. IEEE,2012:553-567.

[2] GJØSTEEN K,THUEN Ø. Password-based signatures [C]// European Public Key Infrastructure Workshop. Springer,2011:17-33.

[3] JARECKI S,KIAYIAS A,KRAWCZYK H,et al. Highly-efficient and composable password-protected secret sharing (or: how to protect your bitcoin wallet online) [C]// 2016 IEEE European Symposium on Security and Privacy (EuroS&P). 2016:276-291.

[4] CASTELLUCCIA C,DÜRMUTH M,PERITO D. Adaptive Password-Strength Meters from Markov Models [C]// NDSS. 2012.

[5] SCHECHTER S,HERLEY C,MITZENMACH-ER M. Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks [C]// Proceedings of the 5th USENIX Conference on Hot Topics in Security. USENIX Association,2010:1-8.

[6] NEWMAN M E J. Power laws,Pareto distributions and Zipf's law [J]. Contemporary Physics,2005,46(5):323-351.

[7] WANG D,CHENG H,WANG P,et al. Zipf's law in passwords [J]. IEEE Transactions on Information Forensics and Security,2017,12(11):2776-2791.

[8] KATZ J,OSTROVSKY R,YUNG M. Effi cient and secure authenticated key exchange using weak passwords [J]. Journal of the ACM(JACM),2009,57(1):1-39.

[9] BAGHERZANDI A,JARECKI S,SAXENA N,et al. Password-protected secret sharing [C]// Proceedings of the 18th ACM conference on Computer and Communications Security,2011:433-444.

[10] JARECKI S,KIAYIAS A,KRAWCYZK H. Round-optimal password-protected secret sharing and T-PAKE in the password-only model [C]// International Conference on the Theory and Application of Cryptology and Information Security. Springer,2014:233-253.

[11] WANG D,WANG P. On the implications of Zipf's law in passwords[C]// European Symposium On Research in Computer Security. Springer,2016:111-131.

[12] WANG D,WANG P. Two birds with one stone:Two-factor authentication with security beyond conventional bound [J]. IEEE Transactions on Dependable and Secure Computing,2016,15(4):708-722.

[13] JUELS A,RIVEST R L. Honeywords:Making password-cracking detectable [C]// Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. 2013:145-160.

[14] SHAMIR A. How to share a secret [J]. Communications of the ACM,1979,22(11):612-613.

DONG Qi-ying, born in 1996, Ph.D, is a member of China Computer Federation. Her main research interests include password security, identity authentication and deep learning.

JIA Chun-fu, born in 1967, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include network and information security, trusted computing and software security, malicious code analysis and cryptography applications.