一个强口令认证方案的攻击与改进

汪 定[1,2] 马春光[1] 张启明[1] 谷德丽[1]
(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)[1]
(解放军汽车管理学院训练部 蚌埠 233011)[2]

摘 要 讨论了于江等新近提出的一个简单高效的基于 USB-Key 的强口令认证方案(USPA),指出该方案无法实现所声称的抵抗 DoS 攻击、重放攻击、Stolen-Verifier 攻击和服务器仿冒攻击。给出一个改进方案,并对其安全性和效率进行了详细的分析。结果表明,改进方案弥补了 USPA 的安全缺陷,并且保持了较高的效率,适用于安全需求较高的移动应用环境。

关键词 强口令,认证,攻击,USB-Key
中图法分类号 TP309 **文献标识码** A

Attacks and Improvements on a Strong-password Authentication Scheme

WANG Ding[1,2] MA Chun-guang[1] ZHANG Qi-ming[1] GU De-li[1]
(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)[1]
(Automobile Management Institute of PLA, Bengbu 233011, China)[2]

Abstract Recently Yu Jiang et al. proposed a USB-Key based strong-password authentication scheme (USPA), and claimed that their scheme was resistant to DoS attack, replay attack, stolen-verifier attack and server impersonation attack. However, we found USPA can't achive these purposes. An improved scheme was advanced and analyzed. The analysis shows that our new scheme precludes the defects of USPA, keeps the merit of high performance, and is suitable for mobile application scenarios where resource is constrained and security is concerned.

Keywords Strong-password, Authentication, Attack, USB-Key

1 引言

随着电子商务和电子政务的快速发展,在远程用户和服务器之间进行身份认证,是确保分布式网络环境中信息系统安全的重要手段。其中,基于口令的认证是最简单、方便且应用最为广泛的一种身份认证方式,但这种身份认证方式易遭到离线口令猜测攻击。为了抵抗这种攻击,基于口令的认证朝着两个方向发展:一个是通过引入公钥密码技术和 Diffie-Hellman 密钥交换技术来支持弱口令的身份认证技术;另一个是以智能卡(如 USB-Key)技术为支撑的强口令身份认证技术。前者由于非对称密码技术的运算开销大,因此其应用范围受到限制。近年来,基于强口令的身份认证已成为安全协议领域研究的热点问题,提出了大量的强口令认证方案,如 SAS[1]、OSPA[2]、SPAS[3]等,但多数被证明存在这样或那样的安全漏洞[4]。关于弱口令、强口令的具体含义详见文献[5],文献[4]是关于强口令认证方案的一个经典综述。

2006 年,虞淑瑶[3]等指出 OSPA 存在 Stolen-Verifier 攻击、重放攻击、预测 T 攻击等安全漏洞,并提出了一种新的基于 Hash 函数的强口令双向认证方案(SPAS)。SPAS 方案虽然具有较高安全性,但整个认证过程需要两轮交互,则其效率较差。2011 年,于江等[6]详细分析了 OSPA 方案存在的无法抵抗 DoS 攻击、重放攻击等安全缺陷,并针对这些不足提出了一种基于 USB-Key 的强口令认证方案(USPA),声称该新方案能够抵抗口令猜测攻击、重放攻击、假冒攻击、DoS 攻击和 Stolen-Verifier 攻击。USPA 整个认证过程只需要一轮交互,十分高效。本文首先分析了 USPA 的安全性,指出该方案无法抵抗 DoS 攻击、重放攻击、Stolen-Verifier 攻击和服务器仿冒攻击;然后结合 USPA 方案的高效性和 SPAS 方案的高安全性,提出了一个改进方案;最后,详细分析了改进方案的安全性和效率。

2 USPA 方案

USPA 方案包含两个阶段,即注册阶段和登录阶段。方案中所使用的符号及其含义如表 1 所列。

到稿日期:2011-08-03 返修日期:2011-10-18 本文受国家自然科学基金(61073042),博士后科研人员落户黑龙江科研启动项目(LBH-Q10141),北京邮电大学网络与交换技术国家重点实验室开放课题(SKLNST-2009-01-10)资助。
汪 定(1985-),男,硕士生,主要研究方向为密码学与无线网络安全,E-mail:wangdingg@mail.nankai.edu.cn;马春光(1974-),男,博士,教授,主要研究方向为密码学与信息安全;张启明(1987-),男,硕士生,主要研究方向为网络信息安全;谷德丽(1983-),女,硕士生,主要研究方向为网络信息安全。

表 1　本文所使用的符号及其含义

符号	含义
U_i	用户 i
S	服务器
ID_i	用户 i 的标识符
P_i	用户 i 的口令
x	服务器的私钥
y	服务器的秘密数
$h(\cdot)$	散列函数
⊕	异或运算
||	连接运算
A→B:M	将消息 M 通过普通信道由 A 传送到 B
A⇒B:M	将消息 M 通过安全信道由 A 传送到 B

2.1　注册阶段

1)用户 A 生成随机数 N,并计算 $h(ID_A \| N \| P_A)$和 $h^2(P_A \oplus N)$,将用户身份标识 ID_A,$h(ID_A \| N \| P_A)$和 $h^2(P_A \oplus N)$发送给服务器 S。

2)服务器 S 计算 $SV=h^2(P_A \oplus N) \oplus h(k \| ID_A)$,其中 k 是 S 的安全密钥。服务器 S 将 SV 发送给用户,并存储 ID_A,$h(ID_A \| N \| P_A)$和 $h^2(P_A \oplus N)$。

2.2　登录阶段

用户插入 USB-Key,验证输入口令正确后,开始身份认证过程。

1)用户 A 的 USB-Key 生成随机数 N',利用其内存储的 SV 和 N 计算出 $h(ID_A \| N \| P_A)$和 $h(k \| ID_A)$,然后计算

$v_1=h^2(k \| ID_A) \oplus h(P_A \oplus N)$

$v_2=h^2(k \| ID_A) \oplus h(ID_A \| N' \| P_A)$

$v_3=h^2(k \| ID_A) \oplus h^2(P_A \oplus N')$

$v_4=h(h(ID_A \| N' \| P_A) \| h^2(P_A \oplus N')\backslash \| h(P_A \| N))$

用户 A 将 $h(ID_A \| N \| P_A)$,v_1,v_2,v_3,v_4 发送给服务器 S。

2)服务器 S 在收到用户 A 的认证请求后,用 $h(ID_A \| N \| P_A)$在口令验证库中检索到用户 A 的身份标识 ID_A 和 $h^2(P_A \oplus N)$计算 $Y_1=h^2(k \| ID_A) \oplus v_1$。验证 Y_1 与口令验证库中保存的 $h^2(P_A \oplus N)$是否相等,若不等,则拒绝;若相等,则用户 A 为合法用户。然后计算

$Y_2=h^2(k \| ID_A) \oplus v_2$

$Y_3=h^2(k \| ID_A) \oplus v_3$

$v'_4=h(Y_1 \| Y_2 \| Y_3)$

$v_5=h(h(ID_A \| N' \| P_A) \| h^2(P_A \oplus N'))$

服务器 S 验证 v_4' 与接收到的 v_4 是否相等。如果不等,则拒绝;否则,说明 v_1,v_2,v_3 在传输过程中未被修改,用 Y_2,Y_3 代替口令验证库中的 $h(ID_A \| N \| P_A)$和 $h^2(P_A \oplus N)$,并向用户 A 发送 v_5。

3) 用户 A 接收到 v_5 后,USB-Key 计算 $v_5'=h(h(ID_A \| N' \| P_A) \| h^2(P_A \oplus N'))$,验证 v_5' 是否与接收到的 v_5 相等。如果不等,则拒绝;否则,用 $SV'=h^2(P_A \oplus N') \oplus h(k \| ID_A)$ 代替原有的 SV,用 N' 代替 N,且认证过程结束。

3　USPA 方案的安全缺陷

文献[4]列出了理想的口令认证协议应当满足的 9 项安全需求,包括实现双向认证和匿名性,以及能够抵抗口令猜测攻击、DoS 攻击、仿冒攻击、重放攻击、平行会话攻击、Stolen-Verifier 攻击和智能卡丢失攻击。文献[6]声称 USPA 方案能够抵抗口令猜测攻击、重放攻击、假冒攻击、DoS 攻击和 Stolen-Verifier 攻击,但研究发现该方案无法抵抗 DoS 攻击、重放攻击、Stolen-Verifier 攻击和服务器仿冒攻击。

3.1　DoS 攻击

USPA 方案要求用户端和服务器端均存储认证参数,服务器在发送消息 v_5 之前已更新认证参数,而客户端只在收到消息 v_5 并且检验其正确性之后才更新认证参数。某次认证之后,如果攻击者使服务器端和用户端存储的认证参数不一致,此后用户就无法认证成功,只能重新注册,而形成 DoS 攻击。攻击者实施这种攻击的方法非常简单,比如篡改 v_5 或者拦截 v_5;即使没有攻击者的存在,若 v_5 在从服务器 S 到用户 A 的途中丢失或者发生误码,同样会导致用户此后无法认证成功。

3.2　重放攻击

文献[7]指出,OSPA 方案存在着难以抵抗重放攻击的安全缺陷,并给出了一个攻击实例。研究表明,USPA 方案也继承了 OSPA 的这个缺陷,并且文献[7]中的攻击方法对攻击 USPA 具有借鉴意义。在合法用户 A 的某次认证过程中,攻击者把用户 A 发送给服务器 S 的消息 v_1,v_2,v_3,v_4 替换为 v_1,v_2,v_3,v_4'($v_4' \neq v_4$),则服务器收到该组消息后,确定此次认证失败,但认证参数保持不变。实际上,攻击者修改 v_1,v_2,v_3,v_4 中的任意一个即可达到此目的,这里只以篡改 v_4 为例。现在,攻击者即使不知道用户 A 的口令 P_A,由于服务器保存的认证参数没有更新,也仍可以通过重放消息 v_1,v_2,v_3,v_4 达到假冒用户 A 成功登录的目的。

3.3　Stolen-Verifier 攻击

设用户 A 第 i 次成功认证之后,服务器端存储的对应用户 A 的口令凭证为 ID_A,$h(ID_A \| N \| P_A)$和 $h^2(P_A \oplus N_i)$,假设攻击者窃取了服务器端存储的用户 A 的口令凭证 $h^2(P_A \oplus N_i)$。如果攻击者还截获了用户 A 与服务器 S 在此前最后一次(第 $i-1$ 次)成功认证中的消息 v_2^{i-1}、v_3^{i-1},那么此后攻击者可成功仿冒服务器 S,具体攻击流程如下:

1)攻击者可由 $h^2(P_A \oplus N_i)$和 v_3^{i-1} 计算出 $h^2(k \| ID_A)=v_3^{i-1} \oplus h^2(P_A \oplus N_i)$,因为

$$v_3^{i-1}=h^2(k \| ID_A) \oplus h^2(P_A \oplus N'_{i-1})$$
$$=h^2(k \| ID_A) \oplus h^2(P_A \oplus N_i)$$

攻击者在获得 $h^2(k \| ID_A)$之后,可进一步计算 $h(ID_A \| N_i \| P_A)=v_2^{i-1} \oplus h^2(k \| ID_A)$。

2)攻击者监听信道,截获到用户 A 发送给服务器 S 的第 m 次认证请求信息 $h(ID_A \| N_m \| P_A)$,v_1^m,v_2^m,v_3^m,v_4^m,其中 $m \geqslant i$。由于攻击者已知 $h^2(k \| ID_A)$,因此可计算

$h(ID_A \| N_m' \| P_A)=v_2^m \oplus h^2(k \| ID_A)$

$h^2(P_A \oplus N_m')=v_3^m \oplus h^2(k \| ID_A)$

3)攻击者假冒服务器 S 向用户 A 发送 v_5^m,其中 $v_5^m=h(h(ID_A \| N_m' \| P_A) \| h^2(P_A \oplus N_m'))$。

4)用户 A 收到 v_5^m 后,验证 v_5^m 是否正确。因为正确的 $h(ID_A \| N_m' \| P_A)$和 $h^2(P_A \oplus N_m')$已被攻击者获知,攻击者发送的 v_5^m 正是用户 A 所预期接收的,所以其显然能够验证通过。则攻击成功。

3.4　服务器仿冒攻击

上面 Stolen-Verifier 攻击成功的关键在于攻击者获取了 $h^2(k \| ID_A)$。实际上,攻击者还有另外一种更简单的获取 h^2

$(k \| ID_A)$的方法，在获取 $h^2(k \| ID_A)$之后可随时进行服务器仿冒攻击。具体攻击流程如下：

1)攻击者从信道中获取用户 A 和服务器 S 第 i 次的认证信息 $v_2^i = h^2(k \| ID_A) \oplus h(ID_A \| N_i' \| P_A)$。

2)攻击者从信道中获取用户 A 和服务器 S 第 $i+1$ 次的认证信息 $h(ID_A \| N_{i+1} \| P_A)$。

3)攻击者由 $h(ID_A \| N_{i+1} | P_A)$和 v_2^i 计算 $h^2(k \| ID_A) = v_2^i \oplus h^2(ID_A \| N_{i+1} \| P_A)$，因为 $N_{i+1} = N_i'$。攻击者监听信道，截获到用户 A 发送给服务器 S 的第 m 次认证请求信息 $h(ID_A \| N_m \| P_A)$，v_1^m，v_2^m，v_3^m，v_4^m，其中 $m > i$。攻击者已知 $h^2(k \| ID_A)$，因此可计算

$$h(ID_A \| N_m' \| P_A) = v_2^m \oplus h^2(k \| ID_A)$$

$$h^2(P_A \oplus N_m') = v_3^m \oplus h^2(k \| ID_A)$$

$$v_5^m = h(h(ID_A \| N_m' \| P_A) \| h^2(P_A \oplus N_m'))$$

4)攻击者假冒服务器 S 向用户 A 发送 v_5^m。

5)用户 A 收到 v_5^m 后，验证 v_5^m 是否正确。因为正确的 $h(ID_A \| N_m' \| P_A)$和 $h^2(P_A \oplus N_m')$已被攻击者获知，攻击者发送的 v_5^m 正是用户 A 所预期接收的，所以其显然能够通过验证。攻击成功。

4 改进方案

本节介绍了一种结合 USPA 方案和 SPAS 方案优点的改进方案。USPA 方案虽然比较高效，整个认证过程只需一轮交互，且实现了用户匿名认证，但存在如前文所述的众多安全缺陷；而 SPAS 方案虽然安全性较高，但整个认证过程需要两轮交互，效率较差。另外，SPAS 方案虽然存在如文献[4]所指出的易遭离线口令猜测攻击的缺陷，但如果借鉴 USPA 方案，将认证参数中的服务器名 S 替换为服务器的安全私钥 k，则可抵抗离线口令猜测攻击。因此，USPA 方案和 SPAS 方案在安全性和效率方面具有很强的互补性。此外，改进方案借鉴了文献[8,9]所提方案中采用双安全因子用以减少认证交互次数和增强匿名性的思想。

改进方案也分为注册阶段和登录阶段，其符号含义如表 1 所列。

4.1 注册阶段

1)用户 A 将用户名 ID_A 和口令 P_A 通过安全信道传送给服务器 S。

2)服务器 S 生成随机数 N_1，计算 $SV = h(x_A \| ID_A)$和 $H_A = h(ID_A \oplus P_A)$，将 SV，H_A，$h(\cdot)$和 $h(\cdot)$写入 USB-Key，然后将 USB-Key 通过安全信道传送给用户 A，并存储 ID_A，N_1，$h^3(P_A \oplus N_1)$，$y \oplus x_A$。其中，x_A 是服务器 S 针对 ID_A 选取的秘密数，用来确保每个用户对应的 SV 不同；y 是 S 的安全密钥，x_A 和 y 作为服务器端的双安全因子，其安全强度足够。

4.2 登录阶段

用户插入 USB-Key，输入用户名 ID_A 和口令 P_A，验证 $h(ID_A \oplus P_A)$与 H_A 是否相等。验证通过后开始第 i 次身份认证过程，如图 1 所示。

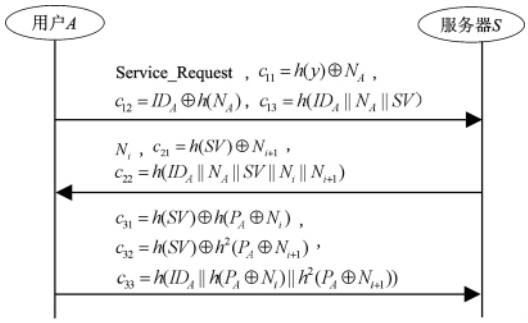

图 1 用户第 i 次身份认证过程

1)用户 A 的 USB-Key 生成随机数 N_A，利用其内存储的认证参数，计算

$$c_{11} = h(y) \oplus N_A$$

$$c_{12} = ID_A \oplus h(N_A)$$

$$c_{13} = h(ID_A \| N_A \| SV)$$

然后将 $msg1 = (c_{11}, c_{12}, c_{13})$发往服务器 S。

2)服务器 S 收到用户 A 的认证请求 $msg1$ 后，计算 $N_A' = c_{11} \oplus h(y)$，$ID_A' = h(N_A') \oplus c_{12}$。由 ID_A'检索口令验证库得到对应的 $y \oplus x_A'$后，计算

$$SV' = h(((x_A' \oplus y) \oplus y) \| ID_A')$$

$$c_{13}' = h(ID_A' \| N_A' \| SV')$$

再比较 c_{13}'与接收到的 c_{13} 是否相等。若不等，则拒绝；否则，有 $SV' = SV$，$ID_A' = ID_A$，$N_A' = N_A$。由 ID_A 检索口令验证库可得到对应的 N_i，生成随机数 N_{i+1}，计算

$$c_{21} = h(SV) \oplus N_{i+1}$$

$$c_{22} = h(ID_A \| N_A \| SV \| N_i \| N_{i+1})$$

服务器 S 将 $msg2 = (N_i, c_{21}, c_{22})$发往用户 A。

3)用户 A 接收到 $msg2$ 后，计算

$$N_{i+1}' = c_{21} \oplus h(SV)$$

$$c_{22}' = h(ID_A \| N_A \| SV \| N_i \| N_{i+1}')$$

验证 c_{22}'与接收到的 c_{22} 是否相等。若不等，则拒绝；如果相等，则可知 $N_{i+1}' = N_{i+1}$，计算

$$c_{31} = h(SV) \oplus h(P_A \oplus N_i)$$

$$c_{32} = h(SV) \oplus h^2(P_A \oplus N_{i+1})$$

$$c_{33} = h(ID_A \| h(P_A \oplus N_i) \| h^2(P_A \oplus N_{i+1}))$$

将 $msg3 = (c_{31}, c_{32}, c_{33})$发往服务器 S。

4)服务器 S 在收到来自用户 A 的 $msg3$ 后，计算 $Y_1 = c_{31} \oplus h(SV)$。验证 $h^2(Y_1)$与口令验证库中保存的 $h^3(P_A \oplus N_i)$是否相等，若不等，则拒绝；若相等，则计算

$$Y_2 = c_{32} \oplus h(SV)$$

$$Y_3 = h(ID_A \| Y_1 \| Y_2)$$

验证 Y_3 与 c_{33} 是否相等。若不等，则拒绝；若相等，则更新服务器端认证参数为 ID_A、N_{i+1}、$h^3(P_A \oplus N_{i+1})$、$y \oplus x_A$，用户 A 认证通过，认证过程结束。

5 改进方案的分析

安全性和效率是衡量基于口令认证方案优劣的两个最重要指标[4]，下面分别从这两个方面进行分析。

5.1 安全性分析

1)匿名性

用户 A 在认证请求中不直接传递标识符 ID_A，而是传递 $c_{12}=ID_A \oplus h(N_A)$。将 ID_A 用一个随机数的散列值隐藏，认证过程中 c_{12} 每次均随机变化，使攻击者无法跟踪，实现匿名认证。

2)双向认证

首先是服务器 S 到用户 A 的认证。在认证过程的步骤 2 中，服务器 S 发往用户 A 的消息 $c_{22}=h(ID_A \| N_A \| SV \| N_i \| N_{i+1})$，$S$ 只有知道 SV 才能计算出 c_{22}，而 $SV=h(((x_A \oplus y) \oplus y) \| ID_A)$，即 S 只有知道服务器的安全密钥 y 才能计算 SV。因此，如果 S 能正确计算出 c_{22}，则可证明 S 知道服务器的安全密钥 y，即 S 是意定的服务器。

然后是用户 A 到服务器 S 的认证。在认证过程的步骤 3 中，用户发往服务器 S 的消息 $c_{31}=h(SV) \oplus h(P_A \oplus N_i)$，用户 A 只有知道 P_A 才能计算 c_{31}。因此，在步骤 4 中，如果服务器 S 验证通过了用户 A 的认证，则证明用户 A 即为意定的通信方。

3)口令猜测攻击

用户在网络中传输的消息不包含口令 P 的明文形式，攻击者只能截获用户与服务器交互过程中的信息，发动离线猜测攻击。在注册阶段，服务器发给用户的签发信息是经过安全信道传输的，可以阻止攻击者对信息的非法截获。在认证阶段，只有 $msg3=(c_{31},c_{32},c_{33})$ 包含 P 的信息，假设攻击者截获了 $msg3$。但是由于 c_{31}、c_{32}、c_{33} 中都至少包含一个随机数或秘密值 $h(SV)$，因此攻击者无法通过所截获的消息来猜测出用户的口令信息，从而可以抵御攻击者对用户口令的离线猜测。

4)DoS 攻击

USPA 方案易遭 DoS 攻击的根本原因在于用户端和服务器端均需要更新认证参数，很容易造成二者之间的认证参数不一致。一方面，改进方案不需要在用户端更新认证参数；另一方面，服务器端认证参数更新通知消息 c_{32} 的完整性由 c_{33} 来确保，攻击者篡改将会被服务器发现。因此，改进方案从上述两方面消除了 DoS 攻击隐患。

5)重放攻击

改进方案的整个认证过程需要 3 次交互，其中 c_{13} 保证了 $msg1$ 的完整性，c_{22} 保证了 $msg2$ 的完整性，c_{33} 保证了 $msg3$ 的完整性。因此，重放攻击若能成功，则必须是对整个 $msg1$、$msg2$ 或 $msg3$ 的重放。另一方面，$msg1$ 中包含用户 A 新生成的随机数 N_A，$msg2$ 和 $msg3$ 中包含服务器 S 新生成的随机数 N_{i+1}，确保了整个消息的新鲜性。因此，改进方案能够抵抗重放攻击。

6)仿冒攻击

改进方案中用户对服务器的认证发生在第二次交互，即服务器 S 向用户 A 发送 $msg2$ 来证明自己的身份。服务器只有知道安全密钥 y 才能计算出正确的 SV，而 SV 被包含在 $msg2$ 的 c_{22} 中。因此，服务器仿冒攻击不会成功。服务器 S 对用户 A 的认证发生在第 3 次交互。只有知道正确的用户口令 P_A，才能计算出新鲜的 $h(P_A \oplus N_i)$；只有输入正确的用户名 ID_A 和口令 P_A，智能卡才计算秘密值 $h(SV)$。因此，用户仿冒攻击也不会成功。

7)平行会话攻击

由于 c_{13} 保证了 $msg1$ 的完整性，c_{22} 保证了 $msg2$ 的完整性，c_{33} 保证了 $msg3$ 的完整性，平行会话攻击若能成功，则必须是对整个 $msg1$、$msg2$ 或 $msg3$ 的替换。另一方面，c_{13}、c_{22} 和 c_{33} 中均嵌入了用户的身份标识符 ID。因此，将 $msg1$、$msg2$ 或 $msg3$ 替换为其它用户认证流中的消息，都将通不过验证。

8)智能卡丢失攻击

用户 A 使用智能卡时，需要提交正确的用户名 ID_A 和口令 P_A，智能卡验证 $h(ID_A \oplus P_A)$ 是否等于 H_A，只有验证通过后才能取出认证参数，仿冒合法用户。智能卡(USB-Key)对用户口令的登录验证具有记录功能，口令错误累计次数到达设定的上限值后将出现锁死，从而防止了此类攻击。

9)Stolen-Verifier 攻击

USPA 方案不能抵抗 Stolen-Verifier 攻击的原因在于，服务器端存储的是 $h^2(P_A \oplus N_i)$，而 $h^2(P_A \oplus N_i)$ 与 $h^2(k \| ID_A)$ 经简单的异或运算后便出现在认证消息流中，攻击者通过监听信道很容易计算出 $h^2(k \| ID_A)$。改进方案在服务器端存储的是 $h^3(P_A \oplus N_i)$，散列函数的单向性保证了攻击者由 $h^3(P_A \oplus N_i)$ 无法得到 $h^2(P_A \oplus N_i)$ 或 $h(P_A \oplus N_i)$。另一方面，认证消息流中只含有 $h^2(P_A \oplus N_i)$ 和 $h(P_A \oplus N_i)$，由 $h^3(P_A \oplus N_i)$ 无法获取其它有价值信息。因此，改进方案可抗 Stolen-Verifier 攻击。

表 2 总结了 USPA、SPAS 和所提改进方案的安全性质。由表 2 可知，改进方案弥补了 USPA 方案的众多安全缺陷，在匿名性和抗口令猜测攻击方面也优于 SPAS，实现了文献[4]所列出的认证协议相关的 9 项主要安全目标。

表 2　3 种强口令认证方案的安全性比较

方案	匿名性	双向认证	抗 DoS 攻击	抗口令猜测攻击	抗重放攻击	抗仿冒* 攻击	抗平行会话攻击	抗智能卡丢失攻击	抗 Stolen-Verifier 攻击
USPA	是	是	否	是	否	否	是	是	否
SPAS	否	是	是	否	是	是	是	是	是
改进方案	是	是	是	是	是	是	是	是	是

注：* 抗仿冒攻击包括抗服务器仿冒攻击和抗用户仿冒攻击两方面

5.2　效率分析

根据计算复杂性理论，远程用户双向认证方案的效率主要取决于交互的轮数、交互信息量、计算量和存储量。其中，交互轮数是影响认证方案时间性能的最重要因素，尤其是在远程用户登录应用中；方案的计算量主要取决于 Hash 函数的运算 $h(\cdot)$，其余的计算量如“⊕”可忽略。考虑到注册往往是事先完成的，并且用户注册的频率要远远低于用户认证的频率，因此关键的计算量和通信量主要取决于登录认证阶段。不失一般性，假设 Hash 函数散列值、服务器安全密钥和随机数的长度均为 128 bit，用户标识符 ID 和口令 P 的长度均为 64bit。3 个方案的效率比较如表 3 所列。

表 3　3 种强口令认证方案的效率比较

方案	服务器				用户端			
	交互轮数	h(·)运算次数	存储量(bit)	通信量(bit)	交互轮数	h(·)运算次数	存储量(bit)	通信量(bit)
USPA	1	5	320	128	1	9	256	640
SPAS	2	8	320	384	2	11	0	576
改进方案	1.5	11	448	256	1.5	10	384	768

由表 3 对比可知：3 个方案中 USPA 的时间性能最优，计算量最小，综合效率最高；SPAS 的时间性能最差，存储需求最低，综合效率最差；改进方案的效率较 USPA 稍差，在交互轮数、计算量、通信量方面均优于 SPAS。因此，改进方案保持了 USPA 的高效性，相比 SPAS 更适用于对认证时延比较敏感的移动应用场合。

结束语　本文首先分析了 USPA 方案的安全性，指出该方案无法抵抗 DoS 攻击、重放攻击、Stolen-Verifier 攻击和服务器仿冒攻击；然后结合 USPA 方案的高效性和 SPAS 方案的高安全性，给出了一个改进方案。安全性分析表明，新的改进方案弥补了 USPA 的安全缺陷，能够达到文献[4]所列出的全部 9 个安全目标，并且较好地解决了 USPA 方案中用户端和服务器端认证参数不一致问题；效率分析表明，改进方案在交互轮数、计算量、通信量方面均优于 SPAS。综合来看，改进方案的实用性比 USPA 和 SPAS 有较大提高，非常适合于对安全需求较高的移动应用环境。

参 考 文 献

[1] Sandirigama M, Shimizu A, Noda M T. Simple and secure password authentication protocol[J]. IEICE Transactions on Communications, 2000, E83(B): 1363-1365

[2] Lin C L, Sun H M, Hwang T. Attacks and solutions on strong-password authentication[J]. IEICE Transactions on Communications, 2001, E84(B): 2622-2627

[3] 虞淑瑶，叶润国，张友坤，等. 一种安全高效的强口令认证协议[J]. 计算机工程，2006，32(6)：146-147

[4] Tsai C S, Lee C C, Hwang M S. Password Authentication Schemes: Current Status and Key Issues[J]. International Journal of Network Security, 2006, 3(2): 101-115

[5] 程英，高庆德. 一个强口令认证协议的漏洞研究[J]. 计算机科学，2009，36(10)：106-116

[6] 于江，苏锦海，张永福. 基于 USB-Key 的强口令认证方案设计与分析[J]. 计算机应用，2011，31(2)：511-513

[7] 秦小龙，杨义先. 强口令认证协议的组合攻击[J]. 电子学报，2003，31(7)：1043-1045

[8] Wang Y Y, Liu J, Xiao F, et al. A more efficient and secure dynamic ID-based remote user authentication scheme[J]. Computer Communications, 2009, 32(4): 583-585

[9] Sandeep K S. Secure Dynamic Identity-Based Authentication Scheme Using Smart Cards[J]. Information Security Journal: A Global Perspective, 2011, 20(2): 67-77

（上接第 57 页）

[6] Srinivasan N, Paolucci M, Sycara K. Semantic Web Service Discovery in the OWL-S IDE[C]// Proceeding of the 39th Hawaii International Conference on System Sciences. Volume 06, 2006: 109

[7] 孙萍，蒋昌俊. 利用服务聚类优化面向过程模型的语义 Web 服务发现[J]. 计算机学报，2008，31(8)

[8] 李小林，张力娜，李卫斌. 一种基于 QOS 的扩展语义 Web 服务发现方法[J]. 重庆师范大学学报：自然科学版，2010，27(6)

[9] 喻坚，韩燕波. 面向服务的计算——原理与应用[M]. 北京：清华大学出版社，14

[10] Gruber T R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing[R]. Padova: Stanford University, 2001

[11] Giuliano C, Lavelli A, Romano L. Relation extraction and the influence of automatic named-entity recognition[J]. ACM Transa-ctions on Speech and Language Processing(TSLP), 2007, 5(1): 11-23

[12] Noor N M M, Ali N H, Ibrahim N S. A new framework to extract WordNet lexicographer files for semi-formal notation: A preliminary study[C]// International Symposium on Information Technology. 2010: 1027-1031

[13] Shibata N, Kajikawa Y, Sakata I. How to measure the semantic rsimilarities between scientific papers and patents in order to discover uncommercialized research fonts: A case study of solar cells [C]// Technology Management for Global Economic Growth(PICMET), 2010 Proceedings of PICMET'10. 2010: 1-6

[14] Shi Lei, Fan Lei, Meng Zhen-zhen. The Research of Using Jena in the Semantic-based Online Learning Intelligent Behavior Analysis System[C]// Proceedings of the 2009 Fifth International Joint Conference on INC. 2009: 926-929

[15] Morohoshi H, Huang Run-he. A user-friendly platform for developing grid services over Globus Toolkit 3[C]// International Conference on Parallel and Distributed Systems (ICPADS). 2005: 668-674

（上接第 63 页）

网络安全态势评估方法，建立了评估与预测模型，并通过实验仿真和分析，验证了该方法的有效性。

参 考 文 献

[1] Bass T. Multisensor data fusion for next generation distributed intrusion detection systems[C]// 1999 IRIS National Symp. on Sensor and Data Fusion. Laurel, 1999: 24-27

[2] 陈秀真，郑庆华，管晓宏，等. 层次化网络安全威胁态势量化评估方法[J]. 软件学报，2006，17(4)

[3] 舒南飞，牛少彰. 网络安全态势评估和预测的新进展[C]// 信息技术与应用学术会议. 2009: 267-273

[4] 韦勇，连一峰. 基于日志审计与性能修正算法的网络安全态势评估模型[J]. 计算机学报，2009，32(4)

[5] 周开利，康耀红. 神经网络模型及其 MATLAB 仿真程序设计[M]. 北京：清华大学出版社，2005

[6] Eleman J L. Finding Structure in Time[J]. Cognitive Science, 1990, 14: 179-211

[7] 党小超，郝占军. 季节周期性 Elman 网络的网络流量分析与应用[J]. 计算机工程与应用，2010，46(28)

[8] Ham F M, Kostanic I. 神经计算原理[M]. 北京：机械工业出版社，2007

[9] http://old.honeynet.org

[10] 王宏伟，杨先一，金文标. 基于 Elman 网络的时延预测与改进[J]. 计算机工程与应用，2008，44(6)：136-138