文章编号: 1000-6788(2015)01-0191-14 中图分类号: TP309.08 文献标志码: A

可证明安全的基于 RSA 的远程用户口令认证协议

汪 定[1,2,3], 王 平[1,3], 李增鹏[2], 马春光[2]

(1. 北京大学 信息科学技术学院, 北京 100871; 2. 哈尔滨工程大学 计算机科学与技术学院, 哈尔滨 150001;
3. 软件工程国家工程研究中心 (北京大学), 北京 100871)

摘 要 身份认证是确保信息系统安全的基本手段, 基于 RSA 的认证协议由于实用性较强而成为近期研究热点. 讨论了 Xie 等提出的一个基于 RSA 的双因子远程用户认证协议, 指出该协议不能抵抗重放攻击和密钥泄露仿冒攻击, 无法实现所声称的安全性, 并且存在用户隐私泄露和可修复性差问题, 不适于实际应用. 给出一个改进方案, 在随机预言机模型下, 基于 RSA 假设证明了改进方案的安全性. 与现有的基于 RSA 的同类协议相比, 改进协议在保持较高效率的同时, 首次实现了可证明安全性, 适用于安全需求较高的移动应用环境.

关键词 认证协议; RSA; 随机预言机模型; 重放攻击; 智能卡

Provably secure RSA-based remote user authentication protocol using passwords

WANG Ding[1,2,3], WANG Ping[1,3], LI Zeng-peng[2], MA Chun-guang[2]

(1. School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China; 2. School of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China; 3. National Engineering Research Center for Software Engineering (Peking University), Beijing 100871, China)

Abstract With identity authentication becoming an essential mechanism to ensure robust system security in information systems, RSA-based authentication protocols have been studied intensively for their great practicality. This paper points out that a recent RSA-based remote user two-factor authentication protocol proposed by Xie et al. cannot achieve the claimed security and reports its following flaws: (1) It is vulnerable to replay attack and key compromise impersonation attack; (2) It suffers from the problem of user privacy violation and poor repairability. As our main contribution, an improved scheme is put forward and formally proved secure under the RSA assumption in the random oracle model. As compared with other related schemes, our scheme is the first one that can achieve provable security while keeping the merit of high performance. Consequently, our scheme is more well-suited for mobile application scenarios where resource is severely constrained and security is particularly concerned.

Keywords authentication protocol; RSA; random oracle model; replay attack; smart card

0 引言

随着电子商务、电子医务和电子政务的快速发展, 信息系统安全问题日益突出. 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, 基于口令的认证由于简单易用、费用低廉而成为一种应用最为广泛的身份认证方式, 但这种身份认证系统中用户常为了方便记忆而选择低信息熵的弱口令, 极易遭到离线口令猜测攻击 [1]. 另一方面, 这种身份认证系统中在服务器端需要维护一个口令验证表项 (password-verifier table), 不仅会降低系统并发处理性能, 而且一旦口令验证表项泄露, 系统便毫无安全性可言 [2]. 为提高身份认证的安全性和有效性, 学者们尝试采用智能卡与口令相结合的技术来设

收稿日期: 2013-06-24
资助项目: 国家自然科学基金 (61472016, 61170241); 黑龙江省自然科学基金 (F201229); 哈尔滨市科技创新人才专项资金 (2012RFXXG086)
作者简介: 汪定 (1985–), 男, 湖北十堰人, 博士研究生, 研究方向: 网络信息安全, E-mail: wangdingg@mail.nankai.edu.cn; 王平 (1961–), 男, 北京人, 博士, 教授, 研究方向: 系统软件与信息安全; 李增鹏 (1989–), 男, 山东青岛人, 博士研究生, 研究方向: 应用密码学; 马春光 (1974–), 男, 黑龙江双鸭山人, 博士, 教授, 研究方向: 传感网、密码学与信息安全.

计远程用户身份认证方案[1], 以实现双因子认证, 即用户只有同时拥有智能卡和知道相应的口令才能认证成功[3−5].

上述 [2–5] 中方案的共同特点是, 都假设智能卡是完全抗窜扰的, 智能卡内秘密参数信息是安全的, 攻击者无法获取. 但是, 文献 [6–8] 的研究表明, 普通智能卡内敏感信息可通过能耗分析、逆向工程或旁路信息泄露等边信道攻击技术而被提取出来. 一旦攻击者获取智能卡内全部 (或部分) 秘密参数信息, 这些方案便面临着离线口令猜测攻击、仿冒攻击等安全威胁. 为此, 设计基于非抗窜扰智能卡假设的双因子认证方案不仅具有理论价值, 而且具有重要现实意义, 该方向已成为近年来安全协议领域的一个研究热点. 同时, 为适应智能卡计算资源受限的特点, 学者们尝试在非抗窜扰智能卡假设下, 设计仅基于对称密码技术 (即非公钥密码技术, 如 Hash 函数, 对称加密算法, 消息认证码) 的安全高效的方案 [9−12], 但都随后被发现存在这样或那样的安全缺陷 [13−17].

文献 [18] 深入分析了双因子认证协议的设计原则并首次证明, 在非抗窜扰智能卡假设下, 基于口令的认证协议必须采用公钥技术才能抵抗离线口令猜测攻击. 当前, 主要有两条公钥技术路线可循: 1) 基于离散对数 (或椭圆曲线下的离散对数) 难解性的公钥技术; 2) 基于大整数因子分解难解性的公钥技术 (即 RSA 技术). 其中, 后者由于加密运算量和解密运算量的非对称性 (如文献 [20] 中的实验数据显示, 当 RSA 加密指数 e 取广泛推荐的 $2^{16}+1$ 时, 加密运算量约是解密运算量的 $\frac{1}{33}$), 非常适合在远程用户认证环境中应用 [19]: 在资源受限的用户端执行加密运算, 在资源相对丰富的服务器端执行解密运算.

2009 年 Park 等 [21] 提出了一个基于 RSA 的远程用户口令认证方案, 指出即使攻击者获得智能卡中的秘密信息, 但攻击者猜测的用户口令是无法验证其正确性的, 因此宣称他们的方案是第一个能抵抗离线口令猜测攻击的基于智能卡的口令认证方案, 而且具有不需要存储口令表、没有时间戳、传输和计算量小等优点. 2010 年 Xie 等 [22] 分析发现 Park 等方案无法抵抗伪造攻击和离线口令猜测攻击, 并针对这些安全缺陷提出了改进方案, 声称他们改进后的方案能抵抗重放攻击、离线口令猜测攻击. 本文分析发现, Xie 等方案 [22] 仍然无法实现所声称安全性, 存在严重安全缺陷. 另一方面, 由于基于智能卡的远程用户口令认证协议通常应用于复杂的网络环境, 存在着大量的潜在攻击场景, 传统的启发式安全性分析很难一一枚举, 因此采用形式化方法对协议进行严格归约证明是确保协议安全性的更理想手段. 近两年来虽然有一些基于 RSA 公钥技术的方案 [17,20−25] 被提出, 尽作者所知, 这些方案都如 Park 等方案和 Xie 等方案, 仅给出了启发式安全性分析, 没有进行严格的安全性归约证明.

本文首先分析了 Xie 等方案的安全性, 指出该方案无法抵抗重放攻击和密钥泄露仿冒攻击, 并且存在可修复性问题, 不适于实际应用; 然后在保持 Xie 等方案优势的基础上, 给出了一个改进方案; 最后, 详细分析了改进方案效率, 并基于随机预言机模型 (random oracle model, ROM) 实现了可证明安全性.

1 Xie 等方案回顾

Xie 等方案 [22] 包括基于随机数和时间戳机制的两种版本, 并主要以随机数机制为例进行了分析. 由于两种机制类似, 都是为了来防御重放攻击, 因此本文也只以随机数机制为例进行分析. Xie 等方案包含三个阶段, 即注册阶段、登录阶段和认证阶段, 方案中所使用的符号及其含义如表 1 所示.

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
$\mathcal{A}$	攻击者
$\mathcal{D}$	用户口令空间
ID_i	用户 i 的标识符
PW_i	用户 i 的口令
d	服务器的私钥
e	服务器的公钥
$h(\cdot)$	散列函数
$\oplus$	异或运算
$\|$	比特连接

1.1 注册阶段

1) 用户 U_i 选择 ID_i 和 PW_i, 通过安全信道发给远程服务器 S.

2) 服务器 S 产生大素数 p 和 q, 计算 $n=pq$; 选取 e 计算 d, 满足 $ed=1\bmod(p-1)(q-1)$; 计算 $S_i=h(ID_i\parallel PW_i)\oplus h(d\oplus ID_i)$; 将 $\{n,e,ID_i,S_i,h(\cdot)\}$ 写入智能卡[2], 然后将智能卡通过安全信道发往用户 U_i. 同时, S 为每个用户建立认证请求数据表 $List\_u_i$, 初始为空.

1.2 登陆阶段

用户 U_i 输入口令 PW_i, 智能卡产生随机数 N_1, 计算:

$$B_i=(S_i\oplus h(ID_i\parallel PW_i)\oplus h(N_1)\parallel N_1)^e \bmod n.$$

然后把 $msg1=(ID_i,B_i)$ 发往 S.

1. 本文中 “方案” 和 “协议” 表达同样的概念, 交叉使用.

2. Xie 等原方案中要求将参数 g 也写入智能卡, 而参数 g 既没有在其方案中定义, 在后文也无涉及, 故这里可能出现了笔误.

1.3 认证阶段

1) S 收到来自 U_i 的 ID_i 和 B_i 后, 检查 $List\_u_i$, 如果 $B_i \in List\_u_i$, 则终止协议; 否则将 B_i 写入 $List\_u_i$ 中. 计算 $B_i^d \bmod n$, 读取前 128 位设为 V_i, 后 128 位为 N_1', 计算 $V_i' = h(d \oplus ID_i) \oplus h(N_1')$, 验证 V_i 与 V_i' 是否相等. 若不等, 则认证失败, 为了不让 U_i 觉察出没有通过认证, S 选取随机数 M, 计算 $C_1 = h(M)$. 若相等, 则表明用户合法, S 计算:

$$C_1 = h((N_1 + 1) \parallel (h(d \oplus ID_i) + 2))^d \bmod n.$$

然后 S 将 $msg2 = (C_1)$ 发往 U_i.

2) U_i 收到来自 S 的消息 C_1 后, 计算 $C_1^e \bmod n$, 并比较

$$C_1^e \bmod n = h((N_1 + 1) \parallel (S_i \oplus h(ID_i \parallel PW_i) + 2)),$$

是否相等, 如果不同则终止协议, 否则认证成功.

表 2　口令认证协议的安全需求

符号	含义
$S1$	抗 DoS 攻击
$S2$	抗仿冒攻击
$S3$	前向安全性
$S4$	双向认证
$S5$	抗平行会话攻击
$S6$	抗口令猜测攻击
$S7$	抗重放攻击
$S8$	抗智能卡丢失攻击
$S9$	抗验证表项窃取攻击
$S10$	抗密钥泄露仿冒攻击
$S11$	抗已知密钥攻击
$S12$	抗内部人员攻击

2 Xie 等方案的安全性分析

2006 年 Tsai 等 [26] 给出了一个适于实际应用的口令认证密钥协商方案应满足的 9 项安全需求, 如表 2 中 $S1 \sim S9$ 所示; 根据最近文献 [10, 15] 的研究, 一个理想的双因子认证协议除应实现 $S1 \sim S9$ 外, 抵抗密钥泄露仿冒攻击 ($S10$)、已知密钥攻击 ($S11$) 和内部人员攻击 ($S12$) 也应是重要的安全目标. Xie 等 [22] 声称所提出的方案是安全高效的, 能够抵御各种潜在的攻击, 如重放攻击、口令猜测攻击和假冒攻击等, 但我们的分析发现该方案无法抵抗重放攻击 ($S7$)、密钥泄露仿冒攻击 ($S10$) 和内部人员攻击 ($S12$), 并且存在可修复性和用户隐私泄露等问题. 在对这些缺陷进行修正前, 不适于实际应用.

2.1 重放攻击

分析发现 Park 等方案 [21] 无法抵抗重放攻击, 但 Xie 等没有意识到该安全缺陷, 他们所提出的方案继承了这一缺陷, 下面给出攻击方法.

在用户 U_i 的第 $k(k \geq 1)$ 次登录过程中, 攻击者 $\mathcal{A}$ 把用户 U_i 发给远程服务器 S 的消息 $msg1 = (ID_i, B_i)$ 篡改为 $msg1' = (ID_i, B_i')$, 其中 B_i' 为 $\mathcal{A}$ 产生的一个随机值. 则服务器 S 收到 $msg1'$ 后必将通不过验证, S 向 U_i 发送 $C_1 = h(M)$. 用户 U_i 在收到 C_1 后也将通不过验证, 确定此次认证失败. 实际上, 攻击者将 $msg1 = (ID_i, B_i)$ 篡改为 $msg1' = (ID_i', B_i)$ 亦可达到此目的, 这里只以篡改 B_i 为例.

现在, 攻击者即使不知道用户的口令 PW_i, 由于 B_i 未曾达到远程服务器 S, 因此 $B_i \notin List\_u_i$, 攻击者重放 $msg1 = (ID_i, B_i)$, $msg1$ 将顺利通过 Xie 等方案设计的重放攻击检测机制, 进而通过远程服务器 S 的验证, 攻击成功.

2.2 密钥泄露仿冒攻击

抗密钥泄露仿冒 (key compromise impersonation, KCI) 攻击是认证协议应实现的一个重要安全属性 [27]. 当协议参与实体 A 的长期私钥泄露后, 攻击者 $\mathcal{A}$ 显然能够成功冒充 A 与协议的其它参与者 (如 B) 进行通信, 但该安全属性保证的是, 这一密钥泄露应不能使得 $\mathcal{A}$ 反过来向 A 冒充为其他合法通信实体 (如 B), 即反向身份欺骗应不能得逞. 该安全属性在现实中是有重要意义的, 比如在云计算环境下的文件共享应用中, 其它用户只有通过用户 A 的认证才能获取 A 的云端存储的文件, 现假设 A 的长期私钥因偶然原因泄露. 如果系统所运行的认证协议能够抗 KCI 攻击, 则其它用户 (包括攻击者 $\mathcal{A}$) 仍无法通过 A 的认证, A 云端的文件仍然是安全的; 如果所运行的认证协议不能抵抗 KCI 攻击, 则攻击者 $\mathcal{A}$ 可仿冒任意用户通过 A 的认证, 成功访问 A 云端存储的文件.

在 Xie 等方案中, 如果服务器 S 的长期私钥 d 因偶然泄露或被窃取, 则攻击者 $\mathcal{A}$ 可从公开信道截获用户 U_i 发往远程服务器 S 的 B_i, 用 d 解密 B_i 获得 N_1, 计算 $h(N_1)$, 进而计算出 $h(d \oplus ID_i) = V_i \oplus h(N_1)$. 因此攻击者可计算:

$$C_1 = h((N_1 + 1) \parallel (h(d \oplus ID_i) + 2))^d \bmod n,$$

然后冒充服务器向 U_i 发送 C_1, 显然 C_1 正是用户 U_i 所期望接收的, 冒充服务器成功.

2.3 不可修复性问题

当用户智能卡丢失, 攻击者 $\mathcal{A}$ 通过边信道攻击技术 [6−8] 获取用户 U_i 智能卡内秘密参数信息后, 或如

2.2 节所述当服务器 S 的长期私钥 d 泄露后, $\mathcal{A}$ 可成功计算出 U_i 的核心安全参数 $h(d \oplus ID_i)$, 然后成功仿冒 U_i 登录 S, 或仿冒 S 与 U_i 通信.

现在的问题是, 即使用户 U_i 发现自己的帐户被仿冒后却无能为力. 由于 $h(d \oplus ID_i)$ 的值与用户口令 PW_i 无关, 只与用户 U_i 身份标识 ID_i 和服务器 S 的长期私钥 d 相关, U_i 通过更新口令 PW_i 不能修复或解决该问题. 为修复上述安全威胁, 系统需要更新 $h(d \oplus ID_i)$ 的值, 即更换 ID_i 或 d, 以将发生泄露的帐户冻结或清除. 由于用户通常在许多应用系统中采用相同的身份标识 ID, 且在很多情况下, 应用系统将用户 ID 与用户身份绑定, 用户 ID 应具有长期稳定性, 更改用户身份标识 ID 不可行. 另一方面, 因为服务器长期私钥 d 参与了系统所有用户安全参数的计算, 仅因为某个用户的安全问题而更换 d, 代价十分昂贵, 是不合理的. 因此, 该方案存在可修复性问题.

2.4 其它问题

当前, 个人隐私问题受到广泛关注. 比如, 在电子商务中, 如果用户的 ID 泄露, 个人的相关信息如购物习惯, 生活方式, 基至社交圈子都会容易被分析出来, 进而被滥用于商业活动 [9,23]. 尤其是, 在移动应用环境中, 用户身份信息的泄露会导致用户位置信息的暴露. 因此, 对用户隐私的保护是至关重要的. 相应地, 匿名性[3]是远程用户认证协议应实现的重要属性之一 [28]. 在 Xie 等方案中, 用户 ID 直接明文传输, 攻击者只需要进行在线窃听就可以获取用户的身份信息, 并且轻易分辨出两个会话是否由同一用户参与, 实现对用户活动的跟踪或用户相关敏感信息的收集.

在现实中, 用户为了方便常多个应用系统中使用同样的口令. 如文献 [29] 对 50 万用户的口令习惯研究显示, 平均每个用户拥有 25 个帐户和 6.5 个口令, 这意味着平均每个口令会在 3.9 个不同的帐户中使用. 因此, 如果其中一个口令泄露, 共用此口令的那些帐户将存在潜在威胁. 在 Xie 等方案中, 用户 U_i 直接将口令 PW_i 递交给服务器, 服务器端的内部攻击者 (malicious privileged insider) 可利用该口令访问 U_i 的其它帐户 [9]. 即使服务器端的内部人员都是可信的, U_i 也不在其它帐户中共用口令, 一个好的协议应能消除这种潜在威胁的可能. 此外, 在基于口令的认证系统中, 一个通行做法是, 用户应定期更新自己的口令, 降低口令泄露的风险, 而 Xie 等方案不支持用户口令更新. 还需要指出的是, 通常在用户认证完成之后, 用户和服务器间需要进行保密通信, 这就需要在双方间协商一个会话密钥. 因此, 协商会话密钥是认证协议应实现的重要属性之一 [13,15], 而 Xie 等方案在认证过程中没有协商会话密钥, 未实现这一属性.

3 改进方案

在 Xie 等方案中, 服务器 S 通过在认证请求数据表 $List\_u_i$ 中检查 B_i 是否已经使用过来防御重放攻击, 这种方式不仅增加了服务器的负载, 而且本质上也不能抵抗我们所给出的重放攻击 (任何基于随机数机制的两轮认证协议都无法抵抗重放攻击 [27]), 除非增加一次交互; 系统的安全性如果完全依赖于服务器私钥 d 的保密, 必然存在 KCI 问题, 我们借鉴文献 [9] 和 [23] 中在服务器端保存用户相关注册信息的思想来实现 KCI 鲁棒性、前向安全性和可修复性; 为抵抗内部人员攻击, 借鉴文献 [2, 9, 20] 中的思想, 用户在注册时向服务器提供口令与一个随机数 b 的 Hash 值, 其中 b 由用户产生且在注册后无须记忆; 为防止用户隐私的泄露, 采用动态 ID 技术 [13] 来实现匿名性; 为实现对用户口令更新的支持, 增加口令更新阶段; 当用户智能卡丢失或发现 (因核心安全参数泄露) 被仿冒时, 通过重注册阶段在保持原 ID 的同时完成对认证参数的更新. 改进方案图 1 所示, 分为注册阶段、登录阶段、认证阶段、口令更新阶段和重注册阶段. 除表 1 所示符号外, 我们另外定义 4 个 Hash 函数 $\mathcal{H}_i : \{0,1\}^* \to \{0,1\}^\ell$, 其中 $\ell = 160, i = 0,1,2,3$.

3.1 注册阶段

用户注册开始前, 服务器 S 产生规模大致相同的大素数 p 和 q, 计算 $n = pq$; 选取 e 计算 d, 满足 $ed = 1 \bmod (p-1)(q-1)$; 公开 e, 保密 d, 即令 e 为其公钥, d 为私钥.

1) 用户 U_i 选择用户名 ID_i、口令 PW_i 和一个随机数 $b \in_R \mathbb{Z}_n^*$, 通过安全信道向 S 传送 $\{ID_i, \mathcal{H}_0(b \parallel PW_i)\}$.

2) 服务器 S 读取当前时间戳 T_{reg}, 计算 $k_i = \mathcal{H}_0(ID_i \parallel d \parallel T_{reg})$ 和 $R_i = \mathcal{H}_0(ID_i \parallel \mathcal{H}_0(b \parallel PW_i)) \oplus k_i$, 保存 $\{ID_i, T_{reg}\}$; 将 $n, e, R_i, \mathcal{H}_0(\cdot), \mathcal{H}_1(\cdot), \mathcal{H}_2(\cdot), \mathcal{H}_3(\cdot)$ 写入智能卡, 然后将智能卡通过安全信道发往 U_i.

3.2 登录阶段

1) 用户 U_i 输入用户名 ID_i 和口令 PW_i;

3. 在远程用户认证中, 匿名性针对的是外部攻击者而非服务器 S, S 需知道用户 ID 以进行相应的授权、审计或计费 [30].

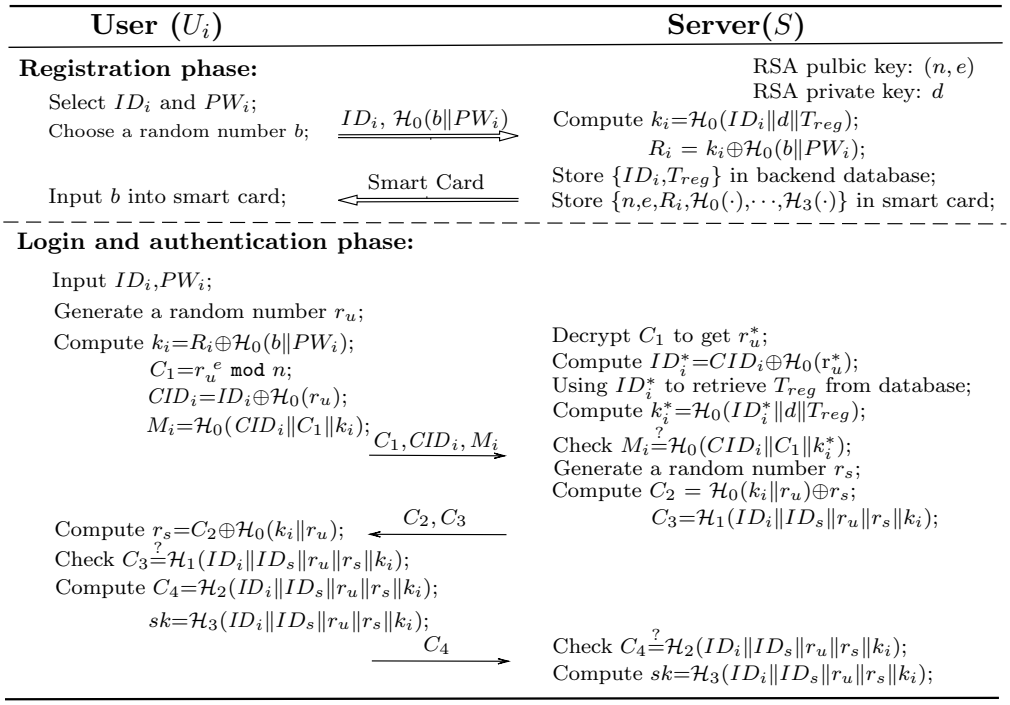

图 1　本文改进方案

2) 智能卡产生随机数 $r_u \in_R \mathbb{Z}_n^*$, 计算 $k_i = R_i \oplus \mathcal{H}_0(b \parallel PW_i)$, $C_1 = (r_u)^e \bmod n$, $CID_i = ID_i \oplus \mathcal{H}_0(r_u)$, $M_i = \mathcal{H}_0(CID_i \parallel C_1 \parallel k_i)$, 然后把 $msg1 = (CID_i, C_1, M_i)$ 发往 S[4].

3.3　认证阶段

1) 服务器 S 收到来自 U_i 的 $msg1$ 后, 计算 $r_u^* = (C_1)^d \bmod n$ 和 $ID_i^* = CID_i \oplus h(r_u)$. 由 ID_i^* 检索后台数据库, 如果在后台数据库中 ID_i^* 没有对应项, 则终止协议; 否则, 由 ID_i^* 检索到后台数据库对应表项中的 T_{reg}, 计算 $k_i^* = \mathcal{H}_0(ID_i \parallel d \parallel T_{reg})$, 验证

$$M_i = \mathcal{H}_0(CID_i \parallel C_1 \parallel k_i^*),$$

是否成立, 如果不成立则终止协议.

2) 服务器 S 产生随机数 $r_s \in_R \mathbb{Z}_n^*$, 计算 $C_2 = r_s \oplus \mathcal{H}_0(k_i \parallel r_u)$ 和 $C_3 = \mathcal{H}_1(ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i)$, 然后将 $msg2 = (C_2, C_3)$ 发往 U_i.

3) 用户 U_i 的智能卡收到来自 S 的 $msg2 = (C_2, C_3)$ 后, 计算 $r_s^* = C_2 \oplus \mathcal{H}_0(k_i \parallel r_u)$, 并验证

$$C_3 = \mathcal{H}_1(ID_i \parallel ID_s \parallel r_u \parallel r_s^* \parallel k_i),$$

是否成立, 如果成立则意味着服务器 s 是合法的; 否则, 终止协议.

4) 智能卡计算 $C_4 = \mathcal{H}_2(ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i)$, 并生成会话密钥 $sk = \mathcal{H}_3(ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i)$, 将 $msg3 = (C_4)$ 发往 S.

5) 服务器 S 收到来自 U_i 的 $msg3 = (C_4)$ 后, 验证

$$C_4 = \mathcal{H}_2(ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i),$$

是否成立, 如果不成立则终止协议; 否则, 意味着用户 U_i 是合法的, 接受 U_i 的认证请求, 并计算会话密钥 $sk = \mathcal{H}_3(ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i)$, 认证成功.

3.4　口令更新阶段

当用户需要更新口令时, 运行此阶段, 具体如下:

1) 运行登录阶段和认证阶段的步骤 $1 \sim 3$;

2) 如果上述步骤成功运行, 智能卡提示用户输入新口令 PW_i^{new} 两次, 如果两次输入的口令不同, 则拒绝用户的口令更新请求. 如果两次输入的口令相同, 则智能卡计算 $R_i^{\text{new}} = R_i \oplus \mathcal{H}_0(b \parallel PW_i) \oplus \mathcal{H}_0(b \parallel PW_i^{\text{new}})$, 然后用 R_i^{new} 替换 R_i, 口令更新完成.

4. 在登录消息中增加验证因子 M_i 是为了防御 Scott 教授提出的一种针对动态 ID 方案的离线口令猜测攻击 [34].

3.5 重注册阶段

当智能卡丢失, 或者发现参数 R_i 泄露自己被仿冒时, 用户使用原用户名 ID_i 和新口令 PW_i^{new} 到服务器重新注册, 具体过程与注册阶段相同.

4 改进方案的分析

安全性和效率是衡量密码协议优劣的两个核心指标, 下面分别从这两个方面进行分析.

4.1 安全性分析

本节首先为基于智能卡的远程用户口令认证协议定义一个形式化安全模型, 然后证明节 3 中提出的新方案在该模型和 RSA 困难性假设下是安全的. 如 Menezes[31] 所言, 虽然形式化方法已逐渐成为证明密码协议安全性的基本手段, 但传统的启发式分析具有直观、易于为更广范围读者理解的优势, 且在建立对密码协议安全性的信任方面仍具有重要作用, 故节 4.1.3 也对一些潜在攻击给出相应的启发式分析.

4.1.1 安全模型

本节简要回顾 Bellare 等在文献 [32] 中为基于口令密钥协商协议定义的形式化安全模型 —— ROM 模型, 该模型通过查询预言机来模拟攻击者的能力. 需要注意的是, 由于原始 ROM 模型并不针对智能卡环境, 不能模拟像智能卡丢失攻击、恶意读卡器等智能卡环境下专有的安全威胁, 因此本文基于 Xu 等 [33] 和 Wang 等 [15] 的工作, 在原始 ROM 模型的基础上进行一些相应的调整, 使其能够反映基于智能卡的远程用户口令认证环境中的特殊安全威胁和需求.

通信模型 远程用户认证协议是一个由用户和服务器参与的两方协议, 用户和服务器间事先共享一些私密参数, 协议的目标是用户和服务器通过交互的方式完成对彼此身份合法性的确认, 并协商一个会话密钥来保护后续通信. 用户和服务器之间的通信信道由攻击者完全控制, 攻击者可以窃听、阻断、删除和窜改流经公开网络中的任何消息, 也可以插入伪造的消息.

参与者 协议 $\mathcal{P}$ 中的参与者为服务器 $S \in \mathcal{S}$ 和用户 $U \in \mathcal{U}$. 每个参与者被模拟成一组预言机, 每个预言机 (也称实例) 可独立, 甚至并发执行协议 $\mathcal{P}$. 将用户和服务器的实例分别设为 U^i 和 S^j, 用 I^k 表示 $\mathcal{U} \cup \mathcal{S}$ 中的一个任意参与者.

初始化 根据节 3 中描述, 服务器 S 拥有一个公私钥对 (e, d), 用户 U 拥有口令 $pw_U \in \mathcal{D}$, 其中口令空间 $\mathcal{D}$ 服从均匀分布[5]. 此外, 当用户 U 向服务器 S 注册时, S 向 U 颁发一个内置有安全参数 $\{R_i\}$ 的智能卡.

查询 攻击者 $\mathcal{A}$ 与协议参与方 $I \in \mathcal{U} \cup \mathcal{S}$ 间的交互通过预言查询来实现, 以此来模拟现实中攻击者的能力. 在协议的运行过程中, $\mathcal{A}$ 可针对某个参与者 I 产生多个并行的会话实例. 赋予 $\mathcal{A}$ 的查询类型有:

Execute(U^i, S^j): 这个查询模拟攻击者的被动攻击 (侦听) 的能力, 输出为协议正常运行情况下的完整的交互信息;

Send(I^k, m): 这个查询模拟攻击者的主动攻击, 向实例 I^k 发送消息 m, I^k 根据协议向 $\mathcal{A}$ 返回处理 m 所产生的相应的信息. 其中, Send$(U^k$, start) 发起一个会话;

Reveal(I^k): 这个查询模拟会话密钥的滥用 (已知密钥攻击), 如果实例 I^k 确实拥有会话密钥 sk, 且 I^k 和其伙伴没有被执行 Test 查询, 则返回 sk; 否则, 返回 $\perp$. 执行 reveal 查询后, I^k 的状态称为打开 (opened).

Corrupt(I, a): 这个查询模拟攻击者 $\mathcal{A}$ 对实例 $I = U($ 或 $S)$ 的腐化能力. 通过恶意读卡器, $\mathcal{A}$ 可获得用户 U 的口令; 通过边信道攻击技术, $\mathcal{A}$ 可获得用户 U 的智能卡中秘密参数; 通过欺骗或偶然因素, $\mathcal{A}$ 可获得服务器 S 的私钥 d; 通过攻破服务器或偶然因素, $\mathcal{A}$ 可获得 S 保存的验证表项 $\{ID_i, T_{reg}\}$. 通过赋予攻击者 $\mathcal{A}$ 对服务器的腐化能力, 可使模型能够模拟 KCI 攻击、验证表项窃取攻击和前向安全性等终极安全威胁. 响应过 Corrupt 查询的实例状态称为 “已腐化”(corrupted).

- Corrupt(U, a) 当 $a = 1$ 时, 输出用户 U 的口令; 当 $a = 2$ 时, 输出用户 U 的智能卡中秘密参数;
- Corrupt(S, a) 当 $a = 1$ 时, 输出服务器 S 的私钥 d; 当 $a = 2$ 时, 输出 S 的验证表项 $\{ID_i, T_{reg}\}$;

我们禁止 $\mathcal{A}$ 同时获得用户口令和智能卡内私密参数信息的能力, 因为这种情况下意味着用户参与认证的双因子都泄露, $\mathcal{A}$ 可轻易攻破任何此类协议, 即平凡攻击. 这个限制在现实中是合理的. 如文献 [35] 所指出的, 在 $\mathcal{A}$ 通过恶意读卡器窃取用户口令的同时 (此时, 用户在现场且输入了正确口令), $\mathcal{A}$ 应不能实施边信道攻击, 因为一方面用户在现场, 另一方面该攻击需要专业的环境和较长的时间. 如文献 [32], 我们也限制 $\mathcal{A}$

5. 即使用户口令空间不是均匀分布, 本文中的相关安全定义也可容易进行相应扩展 [32].

同时获得服务器 S 的私钥 d 和验证表项 $\{ID_i, T_{reg}\}$ 的能力. 现实中, 偶尔会出现服务器端验证表项泄露事件 (如近期几起社交网站大规模口令泄露事件), 但服务器的长期私钥一般经特殊保护 (如秘密分割或多级加密), 被 $\mathcal{A}$ 获取的可能性较小, 出现两者同时被 $\mathcal{A}$ 获取的概率是可以忽略的, 因此限制 $\mathcal{A}$ 同时获取两者也是符合实际的. Corrupt(S, a) 对应 $\mathcal{A}$ 对整个系统的终极破坏, 用以模拟前向安全性、KCI 攻击和窃取验证表项攻击等安全威胁.

Test(I): 这个查询不是模拟攻击者的攻击能力, 而是用来定义会话密钥的语义安全性, 只运行一次, 且只对 "新鲜" 的会话有效. 如果实例 I 尚无会话密钥, 返回 $\perp$. 否则, 根据协议预先选择的一个随机比特 b 返回输出: 如果 $b = 1$, 向 $\mathcal{A}$ 返回会话密钥 sk; 如果 $b = 0$, 向 $\mathcal{A}$ 返回一个等长的随机串.

会话标识　定义实例 I^k 的会话标识 sid 为 I^k 在执行协议过程中所有发送和接收消息的有序级联.

接受/拒绝　如果一个实例完成协议运行并且生成会话密钥, 则称该实例接受, 否则称为拒绝.

伙伴关系　我们基于会话标识符 sid 来定义伙伴关系. 如果下述条件满足, 则称实例 U^i 和实例 S^j 为一对伙伴: ① U^i 和 S^j 均已接受; ② U^i 和 S^j 拥有相同的会话标识符 sid; ③ 实例 U^i 的伙伴标识符 $pid_U^i = S^j$, 实例 S^j 的伙伴标识符 $pid_S^j = U^i$; ④ 不存在其它的实例 $I^k (k \neq i, j)$, 有 $pid_I^k = U^i$ 或 S^j.

新鲜性　新鲜性的概念用来描述这样一个直观事实: 实例 I^k(Test 查询的对象) 的会话密钥 sk_I^k 不能轻易被 $\mathcal{A}$ 所获知, $\mathcal{A}$ 只能对新鲜的实例进行 Test 询问. 如果实例 I^k 满足下述条件, 则认为是新鲜的: ① 已接受且计算出会话密钥 sk_I^k; ② I^k 和它的伙伴这两个实例都未被执行 Reveal 查询; ③ 自协议运行开始, 若 $I = U$, 对 I^k 至多执行一种 Corrupt 查询, 且未对 I^k 的伙伴执行 Corrupt 查询; 若 $I = S$, 未对 I^k 执行 Corrupt 查询, 且对 I^k 的伙伴至多执行一种 Corrupt 查询. 由于 Corrupt(S, a) 查询用以模拟 $\mathcal{A}$ 对整个系统的终极破坏, 故在讨论单个会话的安全性时 $\mathcal{A}$ 应未曾使用该查询.

认证　认证协议的一个基本目标就是防止攻击者 $\mathcal{A}$ 仿冒用户 U 或服务器 S. 用 $\mathrm{Adv}_{\mathcal{P}}^{\mathrm{auth}}(\mathcal{A})$ 表示攻击者 $\mathcal{A}$ 成功仿冒用户 U(或服务器 S) 的优势, 即 $pid_S \neq U$(或 $pid_U \neq S$) 的概率.

语义安全性　带会话密钥协商的认证协议另一个关注目标就是语义安全性. 在协议 $\mathcal{P}$ 的一次运行中, $\mathcal{A}$ 可以执行多项式次 Execute、Send、Reveal、Corrupt 查询和对新鲜实例的一次 Test 查询. 在游戏结束时, $\mathcal{A}$ 输出对 Test 查询中涉及的比特 b 的猜测 b'. 如果 $b = b'$, 我们说 $\mathcal{A}$ 取得游戏的胜利, 并将此事件定义为 **Succ**. 相应地, $\mathcal{A}$ 破坏协议 $\mathcal{P}$ 的语义安全性的优势为

$$\mathrm{Adv}_{\mathcal{P}}^{\mathrm{ake}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} 2\Pr[\mathrm{Succ}(\mathcal{A})] - 1 = 2\Pr[b' = b] - 1,$$

其中, 概率空间定义在攻击者所有的掷币和所有的实例之上. 如果在安全参数 ℓ 下, 任一多项式攻击者 $\mathcal{A}$ 成功的优势都只比 $\mathcal{O}(q_{send}/|\mathcal{D}|)$ 大一个可忽略的量, 则说协议 $\mathcal{P}$ 是语义安全的.

4.1.2 安全性证明

本小节给出本文改进协议 $\mathcal{P}$ 的安全性证明. 协议的安全性基于 RSA 假设, 下面给出 RSA 假设的定义.

RSA 假设　设 ℓ 是系统安全参数, $\mathcal{IG}$ 为 RSA 生成器, 即运行 $\mathcal{IG}$ 有 $(n, e, d) \leftarrow \mathcal{IG}(1^\ell)$, 其中 $n = pq$, 大素数 p, q 的规模相同, $\gcd(e, \phi(n)) = 1$ 并且有 $ed = 1 \bmod \phi(n)$. 当 ℓ 充分大时 (如令 $\ell = 2^{160}$), 对于任意运行时间至多为 t 的概率多项式时间 (probabilistic polynomial time, PPT) 算法 $\mathcal{C}$, 概率

$$\mathrm{Adv}_{\mathcal{C}}^{\mathrm{rsa}}(\mathcal{A}) = \Pr[x^e = c \bmod n : (n, e, d) \leftarrow \mathcal{IG}(1^\ell), c \in \{0,1\}^\ell, x \leftarrow (c, e, n)]$$

是可忽略的. 如果对于任意 PPT 攻击者 $\mathcal{A}$, 其优势 $\mathrm{Adv}^{\mathrm{rsa}}(t) = \max_{\mathcal{A}}\{\mathrm{Adv}_{\mathcal{C}}^{\mathrm{rsa}}(\mathcal{A})\} \leq \epsilon(\ell)$, $\epsilon(\ell)$ 是一个可忽略的量, 则称 $\mathbb{Z}_n$ 中 RSA 假设成立.

定理 1 (语义安全)　设 $\mathcal{A}$ 是一个运行时间为 t, 并且进行了 q_{send} 次 Send 询问、q_{exe} 次 Execute 询问和 q_h 次随机预言机询问的概率多项式攻击者. 如果 RSA 假设在 $\mathbb{Z}_n$ 中成立, 那么敌手 $\mathcal{A}$ 破坏协议 $\mathcal{P}$ 的语义安全性的优势至多为

$$\mathrm{Adv}_{\mathcal{P},\mathcal{D}}^{\mathrm{ake}}(\mathcal{A}) \leq \frac{2q_{\mathrm{send}}}{|\mathcal{D}|} + 6q_{\mathrm{send}}\mathrm{Adv}^{\mathrm{rsa}}(\mathcal{O}(t)) + \frac{q_h^2 + 6q_{\mathrm{send}}}{2^\ell} + \frac{(q_{\mathrm{send}} + q_{\mathrm{exe}})^2}{\phi(n)}.$$

证明　令 $\mathcal{A}$ 为试图破坏协议语义安全性的攻击者. 证明的主要思想是, 利用 $\mathcal{A}$ 来构建破坏 RSA 假设的攻击者, 如果 $\mathcal{A}$ 成功破坏协议的语义安全性, 则 RSA 假设被攻破. 我们使用与文献 [15, 33] 类似的技巧, 定义一系列混合仿真游戏 (Game_0, Game_1, Game_2, $\cdots$), 其中初始游戏 Game_0 为真实攻击, 逐渐对游戏规则进行改变, 直到在某个游戏中 $\mathcal{A}$ 的优势为可忽略的函数为止. 然后, 通过依次界定各仿真游戏间 $\mathcal{A}$ 的优势的差值来确定 $\mathcal{A}$ 的总的优势, 具体证明过程详见附录 1.

定理 2 (认证性) 设 $\mathcal{A}$ 是一个运行时间为 t, 并且进行了 q_{send} 次 Send 询问、q_{exe} 次 Execute 询问和 q_{h} 次随机预言机询问的概率多项式攻击者. 如果 RSA 假设在 $\mathbb{Z}_n$ 中成立, 那么敌手 $\mathcal{A}$ 破坏协议 $\mathcal{P}$ 的认证性的优势至多为

$$\text{Adv}_{\mathcal{P},\mathcal{D}}^{\text{auth}}(\mathcal{A}) \leq \frac{q_{\text{send}}}{|\mathcal{D}|} + 2q_{\text{send}}\text{Adv}^{\text{rsa}}(\mathcal{O}(t)) + \frac{q_{\text{h}}^2 + 6q_{\text{send}}}{2^{\ell+1}} + \frac{(q_{\text{send}} + q_{\text{exe}})^2}{2\phi(n)}.$$

证明 令 $\mathcal{A}$ 为试图破坏协议认证性的攻击者. 证明的主要思想和过程与定理 1 类似, 详见附录 2.

4.1.3 潜在攻击的讨论

由于改进方案基于 Xie 等 [22] 方案, 本节仅对增强的安全特性给出简要的分析.

1) 重放攻击

改进方案的整个认证过程需要三次交互, M_i 保证了 $msg1$ 的完整性, C_3 保证了 $msg2$ 的完整性, C_4 保证了 $msg3$ 的完整性. 因此, 重放攻击若能成功, 必须是对整个 $msg1$、$msg2$ 或 $msg3$ 的重放. 另一方面, $msg2$ 和 $msg3$ 中包含了用户 U_i 和服务器 S 各自新生成的随机数 r_u、r_s, 确保了 $msg2$ 和 $msg3$ 的新鲜性. 因此, 只有 $msg1$ 被重放后不会被 S 检测到, 但只重放 $msg1$ 攻击者将通不过 S 对 C_4 的验证. 因此, 改进方案能够抵抗重放攻击, 满足安全需求 $S7$.

2) 抵抗密钥泄露仿冒 (KCI) 攻击

假设服务器 S 的长期私钥 d 泄露 (但验证表项 $\{ID_i, T_{reg}\}$ 仍然是保密的). 因为攻击者 $\mathcal{A}$ 不知道 U_i 的注册时间戳 T_{reg} 的值, $\mathcal{A}$ 将无法计算出用户 U_i 的安全参数 k_i. 因此, $\mathcal{A}$ 无法计算出合法的 C_4, 进而也就不能仿冒用户 U_i 来欺骗服务器 S. 另一方面, 假设用户 U_i 的口令泄露, 由于攻击者 $\mathcal{A}$ 不知道服务器 s 的私钥 d, 无法解密出用户 U_i 的随机数 r_u, 进而无法计算出合法的 C_3, 进而也就不能仿冒服务器 S 来欺骗用户 U_i. 综上可得, 改进方案可抗密钥泄露仿冒攻击, 满足安全需求 $S10$.

3) 前向安全性

假若服务器 S 的私钥 d 泄露 (但验证表项 $\{ID_i, T_{reg}\}$ 仍然是保密的), 并且攻击者 $\mathcal{A}$ 记录了先前在公开信道中窃听的通信消息 $\{(CID_i, C_1, M_i), (C_2, C_3), C_4\}$. 现在, $\mathcal{A}$ 可解密 C_1 获得用户 U_i 产生的随机数 r_u, 但 U_i 的注册时间戳 T_{reg} 仍是保密的, $\mathcal{A}$ 无法计算出用户 U_i 的核心安全参数 k_i, 进而也就无法计算出合法用户 U_i 先前的会话密钥 $sk = \mathcal{H}_3(ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i)$. 另一方面, 假若服务器 S 的验证表项 $\{ID_i, T_{reg}\}$ 泄露 (但私钥 d 仍然是保密的), 并且攻击者 $\mathcal{A}$ 记录了先前在公开信道中窃听的通信消息 $\{(CID_i, C_1, M_i), (C_2, C_3), C_4\}$. 现在, $\mathcal{A}$ 无法解密 C_1 来获得用户 U_i 产生的随机数 r_u, $\mathcal{A}$ 同样也无法计算出用户 U_i 的核心安全参数 k_i, 进而也就无法计算出会话密钥 sk. 因此, 改进方案实现了前向安全性, 满足安全需求 $S3$.

4) 抗已知密钥攻击

对于实现会话密钥协商的认证协议, 能抵抗已知密钥攻击是一个重要安全属性 [22,27]. 该安全属性确保的是, 某次会话的会话密钥的泄露会不会影响到其它会话的安全性, 这就要求每次会话中产生的会话密钥具有会话相关性和会话间独立性. 在改进方案中, 会话密钥 $sk = \mathcal{H}_3(ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i)$, 即 sk 与本次会话的随机数 r_u, r_s 直接相关; 另一方面, 由于不同会话中产生的随机数 r_u 和 r_s 相同的概率是可忽略的, sk 满足会话间独立性要求. 因此, 改进方案可抗已知密钥攻击, 满足安全需求 $S11$.

5) 抗窃取验证表项攻击

窃取验证表项攻击 [10] 是指攻击者 $\mathcal{A}$ 获取 (如通过服务器的偶然泄露或 $\mathcal{A}$ 的入侵) 用户 U_i 的验证表项后, 可猜测用户口令 PW_i 或发起仿冒攻击. 改进方案中, 即使 $\mathcal{A}$ 获取了用户 U_i 的验证表项为 $\{ID_i, T_{reg}\}$, 但由于服务器 S 的私钥 d 是保密的, $\mathcal{A}$ 无法计算出 U_i 的核心安全参数 k_i, 进而也就无法计算出合法的认证元 C_3 或 C_4. 因此, 改进方案可抗窃取验证表项攻击, 满足安全需求 $S9$.

6) 匿名性

改进方案中采用动态身份标识 CID_i 代替静态的身份标识 ID_i, 假设 $\mathcal{A}$ 通过监听获取到了 CID_i, $\mathcal{A}$ 只有通过解密 C_1 得到 r_u, 才能恢复出 U_i 的身份标识 $ID_i = CID_i \oplus \mathcal{H}_0(r_u)$. 另一方面, 通信消息 $\{CID_i, C_1, M_i, C_2, C_3, C_4\}$ 都是基于随机数 r_u 和 (或) r_s 的 Hash 值, 随着 r_u 和 (或) r_s 的变化而变化, $\mathcal{A}$ 无法分辨出两个会话是否由同一个用户发起, 实现会话的不可跟踪性 (un-traceablity). 因此, 改进方案实现了匿名认证, 保护了用户的隐私, 避免用户被跟踪和相关信息泄露.

表 3 总结了本文改进方案和近期几个基于 RSA 的典型方案的安全性质. 由表 3 可知, 改进方案弥补了 Park 等方案 [21]、Xie 等方案 [22]、Awasthi 等方案 [23]、Zhu 方案 [25] 和 Lee-Liu 方案 [17] 的众多安全缺陷,

表 3　基于 RSA 的口令认证协议的安全性比较

方案	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$	$S9$	$S10$	$S11$	$S12$	可证明安全性
Park 等方案 [21]	是	否	否	是	是	否	否	是	是	否	否	否	否
Xie 等方案 [22]	是	是	否	是	是	是	否	是	是	否	否	否	否
Awasthi 等方案 [23]	是	否	否	是	是	否	是	是	是	是	否	是	否
Zhu 方案 [25]	否	是	是	是	否 *	是	是	是	是	否	是	是	否
Wang 等方案 [20]	是	是	是	是	是	是	是	是	是	是	是	是	否
Lee-Liu 方案 [17]	否 **	是	否	是	是	是	是	是	是	否	是	是	否
本文方案	是	是	是	是	是	是	是	是	是	是	是	是	是

* 文献 [17] 指出 Zhu 方案 [25] 无法抵抗平行会话攻击. ** Lee-Liu 方案 [17] 在用户更新口令时不验证旧口令的合法性, 无法抵抗如文献 [10, 15] 中所描述的 DoS 攻击.

满足口令认证协议如表 2 中所列出的 $S1 \sim S12$ 全部 12 项安全需求. 相较 Wang 等方案, 改进方案实现了可证明安全性, 具有更强的安全保障. 尽作者所知, 改进方案是第一个可证明安全的基于 RSA 的远程用户口令认证协议, 首次实现了此类协议 (基于 RSA) 的可证明安全性. 我们注意到, 与任何 ROM 模型下可证明安全的协议一样, 节 4.1.2 的结果基于随机预言机存在性假设, 相较而言, 标准模型下可证明安全的协议更令人满意. 遗憾的是, 当前最常用的实现抗适应性选择密文攻击的公钥加密方案都也仅给出了 ROM 模型下的证明结果 [22]. 总言之, 改进方案实现 ROM 模型下的可证明安全性, 仍是具有吸引力的.

4.2　效率分析

根据计算复杂性理论, 远程用户认证协议的效率主要取决于交互的轮数、计算量、通信量、和存储量. 计算量主要取决于小指数模幂乘运算 T_{se}、大 (普通) 指数模幂乘运算 T_e 和 Hash 运算 T_h, 据文献 [20] 实验结果有 $T_e \approx 33T_{se} \approx 348T_h$, 其余的轻量级运算如 "$\oplus$"、"$\|$" 忽略不计; 通信量主要取决于传送的信息量; 存储量主要指用户智能卡需要的静态存储空间, 可能随着用户认证次数的变化而动态变化. 考虑到注册往往是事先完成的, 并且用户 (重) 注册的频率要远远低于用户登录认证的频率, 因此方案的效率主要取决于登录认证阶段. 不失一般性, 假设 Hash 函数散列值为 160 bit, 系统生成的随机数、时间戳长度为 128 bit, 服务器产生的模数 n 及其私钥长度均为 1024 bit, 公钥 e 取广泛推荐的 $2^{16}+1$, 即 $|e| = 17$ bit, 用户 ID 和口令 PW 的长度均为 128 bit. 六个相关方案的效率比较如表 4 所示.

表 4　基于 RSA 的口令认证协议的效率比较

方案	交互次数	用户端			服务器		密钥协商	可修复性	匿名性
		计算量	通信量/bit	存储量/bit	计算量	通信量/bit			
Park 等方案 [21]	2*	$2T_{se}+2T_h$	1024	1329+**	$3T_e+1T_h$	1184	否	否	否
Xie 等方案 [22]	2*	$2T_{se}+3T_h$	1152	1329	$2T_e+3T_h$	1024	否	否	否
Awasthi 方案 [23]	2***	$2T_e+1T_{se}$	3345	3217+**	$2T_e+3T_h$	1152	否	否	否
Zhu 方案 [25]	3	$1T_{se}+5T_h$	1312	1191	$1T_e+4T_h$	288	否	否	否
Wang 等方案 [20]	2***	$1T_{se}+4T_h$	1440	1201	$1T_e+4T_h$	288	是	是	否
Lee-Liu 方案 [17]	3	$1T_{se}+6T_h$	1184	1329	$1T_e+5T_h$	288	是	否	是
本文方案	3	$1T_{se}+7T_h$	1504	1184	$1T_e+6T_h$	320	是	是	是

* 基于随机数机制来提供消息新鲜性的两轮认证协议都无法抵抗重放攻击. **Park 等 [21] 和 Awasthi 等 [23] 方案的智能卡中除了存储长度可计算的数据外, 还存储了一个长度未定的原根 g. *** 基于时间戳机制来提供消息新鲜性的两轮认证协议存在时间同步问题 [15].

由表 4 对比可知: 改进方案为克服 Park 等方案和 Xie 等方案存在的重放攻击缺陷、Awasthi 等方案和 Wang 等方案存在的时间同步问题, 增加了一次交互过程; 在计算量方面, 改进方案相较 Park 等方案、Xie 等方案和 Awasthi 等方案, 在客户端和服务器端计算代价均降低 50% 以上, 与其它三个方案代价大致相同; 在通信量方面, 改进方案为了抵御文献 [34] 提出的专门针对动态 ID 方案的离线口令猜测攻击, 客户端数据传送量相较其它五个方案略有增加, 但仍优于 Awasthi 等方案; 在储量方面, 改进方案的用户端智能卡需要 1184 bit 存储空间, 空间复杂度优于其它六种方案. 需要特别指出的是, 在远程用户认证方案中, 用户端 (智能卡) 一般资源受限, 而服务器端资源相对丰富, 因此用户端的效率的高低很大程度上影响着整个方案的优劣. 综上可知, 改进方案在提高安全性和增加匿名性、可修复性等理想特性的同时, 保持了较高的效率, 更适于计算资源受限的移动应用场合.

5 结束语

设计基于非抗窜扰智能卡假设的安全高效的口令认证协议是安全协议研究领域的一个难题. 本文首先分析了 Xie 等近期提出的一个基于 RSA 的口令认证方案, 指出该方案无法抵抗重放攻击和密钥泄露仿冒攻击, 未实现所声称的安全性, 并且存在可修复性差、隐私泄露等问题; 然后在保持 Xie 等方案优势的基础上, 给出了一个改进方案. 安全性分析表明, 改进方案弥补了 Xie 等方案的安全缺陷, 实现了匿名认证, 保护了用户的隐私, 并且首次在 ROM 模型下实现了可证明安全性; 效率分析表明, 改进方案在增强安全性的同时, 在计算量、存储量和通信量方面仍保持了较高的效率. 综合来看, 改进方案的实用性大大提高, 更适合于安全需求较高的移动应用环境. 但是, 改进方案与文中其它相关 RSA 方案一样, 未能实现 "用户口令本地自由安全更新" 这一理想特性. 因此, 设计用户口令可本地安全更新的、可证明安全的高效口令认证协议, 是值得进一步研究的方向.

参考文献

[1] 冯登国, 陈伟东. 基于口令的安全协议的模块化设计与分析 [J]. 中国科学 (E 辑), 2007, 37(2): 223–237.
Feng Dengguo, Chen Weidong. Modular approach to the design and analysis of password-based security protocols[J]. Science in China Series E, 2007, 37(2): 223–237.

[2] Chen T H, Hsiang H C, Shih W K. Security enhancement on an improvement on two remote user authentication schemes using smart cards[J]. Future Generation Computer Systems, 2011, 27(4): 377–380.

[3] Das M L. Two-factor user authentication in wireless sensor networks[J]. IEEE Transactions on Wireless Communications, 2009, 8(3): 1086–1090.

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

[5] Yoon E J, Yoo K Y, Ha K S. A user friendly authentication scheme with anonymity for wireless communications[J]. Computers & Electrical Engineering, 2011, 37(3): 356–364.

[6] Messerges T S, Dabbish E A, Sloan R H. Examining smart card security under the threat of power analysis attacks[J]. IEEE Transactions on Computers, 2002, 51(5): 541–552.

[7] Markantonakis K, Tunstall M, Hancke F, et al. Attacking smart card systems: Theory and practice[J]. Information Security Technical Report, 2009, 14(2): 46–56.

[8] Kim T H, Kim C, Park I. Side channel analysis attacks using am demodulation on commercial smart cards with seed[J]. Journal of Systems and Software, 2012, 85(12): 2899–2908.

[9] Khan M K, Kim S K, Alghathbar K. Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme[J]. Computer Communications, 2011, 34(3): 305–309.

[10] Sood S K. Secure dynamic identity-based authentication scheme using smart cards[J]. Information Security Journal: A Global Perspective, 2011, 20(2): 67–77.

[11] Wen F, Li X. An improved dynamic ID-based remote user authentication with key agreement scheme[J]. Computers & Electrical Engineering, 2012, 38(2): 381–387.

[12] Hsieh W, Leu J. Exploiting hash functions to intensify the remote user authentication scheme[J]. Computers & Security, 2012, 31(6): 791–798.

[13] He D B, Chen J H, Zhang R. Weakness of a dynamic ID-based remote user authentication scheme[J]. International Journal of Electronic Security and Digital Forensics, 2010, 3(4): 355–362.

[14] Shim K. Security flaws in three password-based remote user authentication schemes with smart cards[J]. Cryptologia, 2012, 36(1): 62–69.

[15] Wang D, Ma C G. Robust smart card based password authentication scheme against smart card security breach[R]. Cryptology ePrint Archive, 2012, Report 2012/439: 1–32. http://eprint.iacr.org/2012/439.pdf.

[16] He D B, Hu J. Cryptanalysis of a dynamic ID-based remote user authentication scheme with access control for multi-server environments[J]. IEICE Transactions on Information and Systems, 2013, 96(1): 138–140.

[17] Lee T F, Liu C M. A secure smart-card based authentication and key agreement scheme for telecare medicine information systems[J]. Journal of Medical Systems, 2013, Doi: 10.1007/s10916-013-9933-8.

[18] Ma C G, Wang D, Zhao S D. Security flaws in two improved remote user authentication schemes using smart cards[J]. International Journal of Communication Systems, 2014, 27(10): 2215–2227.

[19] Zhu F. RSA-based password authenticated key exchange for imbalanced wireless networks[C]// Proceedings of the 5th Information Security Conference (ISC 2002), Berlin: Springer-Verlag, LNCS 2433, 2002: 150–161.

[20] 汪定, 马春光, 翁臣, 等. 一种适于受限资源环境的远程用户认证方案的分析与改进 [J]. 电子与信息学报, 2012, 34(10): 2520–2526.
Wang Ding, Ma Chunguang, Weng Chen, et al. Cryptanalysis and improvement of a remote user authentication

scheme for resource-limited environment[J]. Journal of Electronics & Information Technology, 2012, 34(10): 2520–2526.

[21] Park J K, Lee J S, Chang J H. An efficient remote user authentication scheme secure against the offline password guessing attack by power analysis[C]// Proceedings of the 11th IEEE International Conference on Advanced Communication Technology, Feb 2–5, Phonenix, U.S.A. Phonenix: IEEE Press, 2009: 1289–1292.

[22] 谢琪, 陈德人, 于秀源. Park 等远程用户认证协议的分析与改进 [J]. 系统工程理论与实践, 2010, 30(10): 1877–1882.
Xie Qi, Chen Deren, Yu Xiuyuan. Cryptanalysis and improvement of Park et al.'s remote user authentication protocol[J]. Systems Engineering — Theory & Practice, 2010, 30(10): 1877–1882.

[23] Awasthi A K, Srivastava K, Mittal R C. An improved timestamp-based remote user authentication scheme[J]. Computers & Electrical Engineering, 2011, 37(6): 869–874.

[24] Ramasamy R, Muniyandi A P. An efficient password authentication scheme for smart card[J]. International Journal of Network Security, 2012, 14(3): 180–186.

[25] Zhu Z A. An efficient authentication scheme for telecare medicine information systems[J]. Journal of Medical Systems, 2012, 36(6): 3833–3838.

[26] 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.

[27] Krawczyk H. HMQV: A high-performance secure Diffie-Hellman protocol[C]// Proceedings of the 25th Annual International Cryptology Conference (CRYPTO 2005), Aug 14–18, 2005, Santa Barbara, CA, USA, Berlin: Springer-Verlag, LNCS 3621, 2005: 546–566.

[28] Li X, Qiu W, Zheng D, et al. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards[J]. IEEE Transactions on Industrial Electronics, 2010, 57(2): 793–800.

[29] Florencio D, Herley C. A large-scale study of web password habits[C]// Proceedings of the 16th International World Wide Web Conference (WWW 2007), New York: ACM, 2007: 657–666.

[30] Mangipudi K, Katti R. A secure identification and key agreement protocol with user anonymity[J]. Computers & Security, 2006, 25(6): 420–425.

[31] Menezes A. Another look at provable security[C]// Proceedings of the 31th International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2012), Berlin: Springer-Verlag, LNCS 7237, 2012: 8. Keynote, http://www.cs.bris.ac.uk/eurocrypt2012/Program/Weds/Menezes.pdf.

[32] Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks[C]// Proceedings of the 19th International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2000), May 14–18, 2000, Bruges, Belgium, Berlin: Springer-Verlag, LNCS 1807, 2000: 139–155.

[33] Xu J, Zhu W, Feng D. An improved smart card based password authentication scheme with provable security[J]. Computer Standards & Interfaces, 2009, 31(4): 723–728.

[34] Scott M. Cryptanalysis of a recent two factor authentication scheme[R]. Cryptology ePrint Archive, 2012, Report 2012/527: 1–3. http://eprint.iacr.org/2012/527.pdf.

[35] Wang Y G. Password protected smart card and memory stick authentication against off-line dictionary attacks[C]// Proceedings of the 27th IFIP International Information Security and Privacy Conference (SEC 2012), June 4–6, 2012, Heraklion, Greece, Berlin: Springer-Verlag, IFIP AICT 376, 2012: 489–500.

[36] Zhang M X. New approaches to password authenticated key exchange based on RSA[C]// Proceedings of 10th International Conference on the Theory and Application of Cryptology and Information Security (AsiaCrypt 2004), Dec 5–9, 2004, Jeju Island, Korea, Berlin: Springer-Verlag, LNCS 3329, 2004: 230–244.

附录 1

为证明定理 1, 我们使用与文献 [15, 33] 类似的技巧. 令 $\mathcal{A}$ 为试图破坏协议语义安全性的攻击者. 通过一系列逐步修改规则的仿真游戏将 $\mathcal{A}$ 对协议的攻击归约到 $\mathcal{A}$ 对计算性数学难题 (密码原语) 的攻击, 如果 $\mathcal{A}$ 成功破坏协议的语义安全性, 则至少一个计算性数学难题 (如 RSA 假设) 被攻破, 而后者发生的概率是可忽略的. 我们定义一系列仿真游戏 (Game_0, Game_1, Game_2, $\cdots$, Game_7), 在所有的游戏中, 预言机按照协议的描述处理查询. 其中, Game_0 模拟的是 $\mathcal{A}$ 攻击真实协议 $\mathcal{P}$, 在后续的游戏中逐步修改预言机的回答方式, 以致于 Game_8 中 $\mathcal{A}$ 的优势为 0, 且在两个相邻实验中 $\mathcal{A}$ 成功概率的差值可以忽略. 在每个游戏 Game_n $(n = 0, 1, 2, \cdots, 7)$ 中, 我们定义下述事件:

Succ$_n$: $\mathcal{A}$ 成功猜测 Test 查询中的测试比特 b;

AskPara$_n$: $\mathcal{A}$ 通过对 $b \parallel PW_i$ 或 $ID_i \parallel d \parallel T_{reg}$ 查询 $\mathcal{H}_0$, 从而成功计算出 U_i 的核心安全参数 k_i;

AskAuth$_n$: $\mathcal{A}$ 成功计算出核心安全参数 k_i, 并且对 $ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i$ 查询 $\mathcal{H}_1$ 或 $\mathcal{H}_2$;

AskH$_n$: $\mathcal{A}$ 成功计算出核心安全参数 k_i, 并且对 $ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i$ 查询 $\mathcal{H}_i (i = 1, 2, 3)$;

Game$_0$: 该游戏对应 ROM 下的真实攻击, 我们根据协议描述模拟所有查询: Execute、Send、Reveal、Corrupt 和 Test, 攻击者 $\mathcal{A}$ 与协议参与方 $I \in \mathcal{U} \cup \mathcal{S}$ 间的交互通过预言查询来模拟. 根据定义有

$$\mathrm{Adv}^{\mathrm{ake}}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) = 2\Pr[\mathrm{Succ}_0] - 1 \tag{1}$$

Game$_1$: 在这个游戏中, 我们通过维护 Hash 查询结果列表 $\Lambda_{\mathcal{H}}$ 和 $\Lambda_{\mathcal{H}'}$ 来模拟所有的 Hash 查询 $\mathcal{H}_i(i = 0,1,2,3)$ 以及将在 Game$_7$ 中出现的私有 Hash 查询 $\mathcal{H}'_i(i = 1,2,3)$. 为实现对随机预言查询的跟踪, 我们增加一个 Hash 查询结果列表 $\Lambda_{\mathcal{A}}$, 该列表记录由 $\mathcal{A}$ 直接实施的 Hash 查询. 随机预言查询的模拟规则如下:

1) 查询列表 $\Lambda_{\mathcal{H}}$、$\Lambda_{\mathcal{H}'}$ 和 $\Lambda_{\mathcal{A}}$ 初始为空.

2) 对于每一次随机预言查询 $\mathcal{H}_i(x)$, 如果列表 $\Lambda_{\mathcal{H}}$ 中存在记录 (i,x,y), 则返回 y; 否则, 选择 $r \in_R \{0,1\}^{\ell}$, 将 r 作为查询输出返回给询问者, 并且将记录 (i,x,r) 添加到列表 $\Lambda_{\mathcal{H}}$ 中 (如果询问者为 $\mathcal{A}$, 还需要将记录 (i,x,r) 添加到列表 $\Lambda_{\mathcal{A}}$ 中).

3) 对于每一次随机预言查询 $\mathcal{H}'_i(x)$, 如果列表 $\Lambda_{\mathcal{H}'}$ 中存在记录 $(i,x,y,\triangle)$, 则返回 y; 否则, 选择 $r \in_R \{0,1\}^{\ell}$, 将 r 作为查询输出返回给询问者, 并且将记录 $(i,x,r,\triangle)$ 添加到列表 $\Lambda_{\mathcal{H}'}$ 中. 需要指出的是, 由于 $\mathcal{H}'_i(i = 1,2,3)$ 是私有查询不对 $\mathcal{A}$ 开放, 询问者不可能为 $\mathcal{A}$, 故列表 $\Lambda_{\mathcal{A}}$ 中不会存在记录 $(*,*,*,\triangle)$.

除了模拟随机预言查询外, 我们还根据协议描述模拟所有的 Execute、Send、Reveal、Corrupt 和 Test 查询. 不难看出, Game$_1$ 与真实攻击是不可区分的, 可得

$$\left|\Pr[\mathrm{Succ}_1] - \Pr[\mathrm{Succ}_0]\right| = 0 \tag{2}$$

Game$_2$: 为便于分析, 本游戏中我们去掉一些不太可能发生的碰撞:

1) 随机预言查询 (Hash 查询) 输出的碰撞;

2) 通信消息 $((C_1, CID_i, M_i),(C_2,C_3),C_4)$ 的碰撞, 需要指出的是, 这些产生这些通信消息的参与方中至少有一个是诚实实体, 故 C_1 和 C_2 中至少有一个随机分布的.

根据生日悖论 (birthday paradox) 原理, 可得

$$\left|\Pr[\mathrm{Succ}_2] - \Pr[\mathrm{Succ}_1]\right| \le \frac{(q_{\mathrm{send}} + q_{\mathrm{exe}})^2}{2\phi(n)} + \frac{q_{\mathrm{h}}^2}{2^{\ell+1}} \tag{3}$$

Game$_3$: 本游戏中, 如果 $\mathcal{A}$ 幸运地猜测出认证元 C_3, 则终止协议的执行. 我们通过检查列表 $\Lambda_{\mathcal{A}}$ 来实现这一策略: 如果记录 $(1, ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i, C_3) \notin \Lambda_{\mathcal{A}}$, 但用户 U_i 接受, 则终止协议. 类似地, 如果 $(2, ID_i \parallel ID_s \parallel r_u \parallel r_s \parallel k_i, C_3) \notin \Lambda_{\mathcal{A}}$, 则意味着 $\mathcal{A}$ 幸运地猜测出认证元 C_4, 则终止协议的执行. 由于 Hash 输出碰撞的可能性在 Game$_2$ 中已排除, 因此除非合法认证元 (由 $\mathcal{A}$ 猜测而来) 被拒绝, Game$_3$ 和 Game$_2$ 是不可区分的, 可得

$$\left|\Pr[\mathrm{Succ}_3] - \Pr[\mathrm{Succ}_2]\right| \le \frac{q_{\mathrm{send}}}{2^{\ell}} \tag{4}$$

Game$_4$: 本游戏中, 如果 $\mathcal{A}$ 幸运地猜测出用户 U_i 的核心安全参数 k_i, 且成功地仿冒用户或服务器, 则终止协议的执行. 我们通过检查列表 $\Lambda_{\mathcal{A}}$ 来实现这一策略: 如果记录 $(0, ID_i \parallel d \parallel T_{reg}, k_i) \notin \Lambda_{\mathcal{A}}$, 且记录 $(0, b\|PW_i, \mathcal{H}_0(b\|PW_i)) \notin \Lambda_{\mathcal{A}}$, 但 $\mathcal{A}$ 成功地计算出认证元 C_3 或 C_4(如 Game$_3$, 通过检查 $\Lambda_{\mathcal{A}}$ 列表来判别), 则终止协议. 由于 r_s 和 r_u 中至少有一个是诚实实体产生的, r_s 和 r_u 中至少有一个是随机的, 并且我们在 Game$_2$ 中已排除 Hash 输出碰撞的可能性, 故除非 k_i 由 $\mathcal{A}$ 猜测而来, 否则 Game$_4$ 和 Game$_3$ 是不可区分的, 可得

$$\left|\Pr[\mathrm{Succ}_4] - \Pr[\mathrm{Succ}_3]\right| \le \frac{q_{\mathrm{send}}}{2^{\ell}} \tag{5}$$

Game$_5$: 如果 $\mathcal{A}$ 成功地计算出安全参数 k_i, 即 $\mathcal{A}$ 通过查询 $\mathcal{H}_0$ 来得到参数 k_i(如 Game$_3$, 通过检查 $\Lambda_{\mathcal{A}}$ 列表来判别该行为), 则终止协议的运行. 在事件 **AskPara$_5$** 不发生的情况下, Game$_5$ 和 Game$_4$ 是不可区分的, 可得

$$\left|\Pr[\mathrm{Succ}_5] - \Pr[\mathrm{Succ}_5]\right| \le \Pr[\mathrm{AskPara}_5] \tag{6}$$

引理 1 在游戏 Game$_5$ 中, 事件 **AskPara$_5$** 发生的概率约束如下:

$$\left|\Pr[\mathrm{AskPara}_5]\right| \le \frac{q_{\mathrm{send}}}{|\mathcal{D}|} + \frac{q_{\mathrm{send}}}{2^{\ell}} \tag{7}$$

证明 游戏 Game$_5$ 中, 事件 **AskPara$_5$** 发生意味着 $\mathcal{A}$ 对 $b\|PW_i$ 或 $ID_i\|d\|T_{reg}$ 查询 $\mathcal{H}_0$. 根据节 4.1.1 中安全模型定义, 查询 Corrupt$(S,1)$ 和 Corrupt$(S,2)$ 在讨论语义安全性和认证性时是禁止的, 在未知 d 和

T_{reg} 的情况下, $\mathcal{A}$ 对 $ID_i\|d\|T_{reg}$ 查询 $\mathcal{H}_0$ 的概率小于 $\mathcal{A}$ 对 k_i 的直接猜测的概率, 而我们在 Game_4 中已去除了 $\mathcal{A}$ 直接猜测 k_i 情形.

$\mathcal{A}$ 只可通过 $\text{Corrupt}(U,1)$ 或 $\text{Corrupt}(U,2)$ 这两种查询的帮助来计算安全参数 k_i, 我们将这两种不同情况下的概率分别记为 $\Pr[\text{AskPara}_5\text{WithCorr}_1]$ 和 $\Pr[\text{AskPara}_5\text{WithCorr}_2]$. 设 $P_i=\mathcal{H}_0(b\|PW_i)$, 我们用 $R(U)$ 表示用户实例接收到的通信消息 (C_2,C_3) 的集合, 用 $R(S)$ 表示服务器实例接收到的通信消息 (C_4) 的集合, 由于在 Game_2 中已去除了通信消息碰撞的可能性, 从信息论的角度, 则有

$$\begin{aligned}\left|\Pr[\text{AskPara}_5\text{WithCorr}_1]\right| =& \Pr_{pw}[\exists C_3\in R(U),((1,ID_i\|ID_s\|r_u\|r_s\|k_i,C_3)\in\Lambda_{\mathcal{A}})\wedge((0,*\|PW_i,P_i)\in\Lambda_{\mathcal{A}})]\\ &+\Pr_{pw}[\exists C_4\in R(S),((2,ID_i\|ID_s\|r_u\|r_s\|k_i,C_4)\in\Lambda_{\mathcal{A}})\wedge((0,*\|PW_i,P_i)\in\Lambda_{\mathcal{A}})]\\ &\le\frac{\#R(S)+\#R(U)}{2^{\ell}}=\frac{q_{\text{send}}}{2^{\ell}}\end{aligned}\tag{8}$$

$$\begin{aligned}\left|\Pr[\text{AskPara}_5\text{WithCorr}_2]\right| =& \Pr_{pw}[\exists C_3\in R(U),((1,ID_i\|ID_s\|r_u\|r_s\|k_i,C_3)\in\Lambda_{\mathcal{A}})\wedge((0,b\|*,P_i)\in\Lambda_{\mathcal{A}})]\\ &+\Pr_{pw}[\exists C_4\in R(S),((2,ID_i\|ID_s\|r_u\|r_s\|k_i,C_4)\in\Lambda_{\mathcal{A}})\wedge((0,b\|*,P_i)\in\Lambda_{\mathcal{A}})]\\ &\le\frac{\#R(S)+\#R(U)}{|\mathcal{D}|}=\frac{q_{\text{send}}}{|\mathcal{D}|}\end{aligned}\tag{9}$$

Game_6: 本游戏中, 如果 $\mathcal{A}$ 成功地计算出认证元 C_3 或 C_4, 即 $\mathcal{A}$ 通过查询 $\mathcal{H}_1$ 或 $\mathcal{H}_2$(如 Game_3, 通过检查 $\Lambda_{\mathcal{A}}$ 列表来判别) 得到认证元, 终止协议的运行. 在事件 **AskAuth_6** 不发生的情况下, Game_6 和 Game_5 是不可区分的, 可得

$$\left|\Pr[\text{Succ}_6]-\Pr[\text{Succ}_5]\right|\le\Pr[\text{AskAuth}_6],\qquad\left|\Pr[\text{AskPara}_6]-\Pr[\text{AskPara}_5]\right|\le\Pr[\text{AskAuth}_6]\tag{10}$$

Game_7: 本游戏中, 我们用私有的 Hash 函数 $\mathcal{H}_i'(i=1,2,3)$ 来代替 $\mathcal{H}_i$, 即令 $C_3=\mathcal{H}_1'(ID_i\parallel ID_s\parallel C_1\parallel C_2)$, $C_4=\mathcal{H}_2'(ID_i\parallel ID_s\parallel C_1\parallel C_2)$, $sk=\mathcal{H}_3'(ID_i\parallel ID_s\parallel C_1\parallel C_2)$. 这样, 认证元 C_3、C_4 和会话密钥 sk 完全独立于用户 U_i 的核心安全参数 k_i 和协议运行中产生的随机数 r_s、r_u. 在事件 **AskH_7** 不发生的情况下, Game_7 和 Game_6 是不可区分的, 可得

$$\left|\Pr[\text{Succ}_7]-\Pr[\text{Succ}_6]\right|\le\Pr[\text{AskH}_7],\qquad\left|\Pr[\text{AskAuth}_7]-\Pr[\text{AskAuth}_6]\right|\le\Pr[\text{AskH}_7]\tag{11}$$

引理 2 在游戏 Game_7 中, 事件 **Succ_7** 和 **AskH_7** 发生的概率约束如下:

$$\left|\Pr[\text{Succ}_7]\right|=\frac{1}{2},\qquad\left|\Pr[\text{AskH}_7]\right|\le q_{\text{send}}\text{Adv}^{\text{rsa}}(\mathcal{O}(t))\tag{12}$$

证明 游戏 Game_7 中, 会话密钥 sk 由私有 Hash 查询 $\mathcal{H}_3'$ 计算而来, 完全独立于 $\mathcal{A}$ 的攻击能力, 不难得出 $\Pr[\text{Succ}_7]=\frac{1}{2}$. 由于我们在 Game_5 已去掉了 $\mathcal{A}$ 计算出 k_i 的可能, 可知 $\Pr[\text{AskPara}_7]=0$.

在游戏 Game_6 中, C_3、C_4 和 sk 是随机预言 $\mathcal{H}_i(i=1,2,3)$ 在输入为 $(ID_i\|ID_s\|r_u\|r_s\|k_i)$ 时的输出. 下面仅以 sk 为例进行说明. 如果 $\mathcal{A}$ 不知道 r_u、r_s 和 k_i, 那么不可能区分真正的会话密钥和从 $\{0,1\}^{\ell}$ 中选择的随机数. 因此, 只有 $\mathcal{A}$ 获知 r_u、r_s 和 k_i, 才能进行 $\mathcal{H}_3$ 查询, 即事件 **AskH_7** 发生, 进而区分 Game_6 和 Game_7. 我们采用与文献 [36] 类似技巧, 认为攻击者只要恢复 r_u 和 r_s 就算成功. 依据查询 $\text{Send}(S^j,<C_1,CID_i,M_i>)$ 发起方的不同, 由下面分两种情况讨论:

1) 查询 $\text{Send}(S^j,<C_1,CID_i,M_i>)$ 由 $\mathcal{A}$ 发起, 设此时 $\mathcal{A}$ 成功的概率为 $\Pr[\text{AskH}_7\text{WithSend}_1]$. 依据登录消息 $<C_1,CID_i,M_i>$ 是否由 $\mathcal{A}$ 自己产生, 又可再细分为两种情况:

$C_1=(r_u)^e \bmod n$ 由 $\mathcal{A}$ 自己产生. 此时, 易见整个登录消息 $<C_1,CID_i,M_i>$ 必须由 $\mathcal{A}$ 自己产生, 否则 M_i 必将通不过服务器 S 的验证. 为计算 M_i, $\mathcal{A}$ 需要对 $C_1\|CID_i\|k_i$ 查询 $\mathcal{H}_0$, 为正确执行这一查询, $\mathcal{A}$ 需要知道 k_i 的值. 若 $\mathcal{A}$ 知道 k_i 的值, 收到来自服务器 S 的响应 C_2,C_3 后, 可容易计算 $r_s=C_2\oplus\mathcal{H}_0(k_i\parallel r_u)$, 攻击成功. 现在的关键是, 确定 $\mathcal{A}$ 获知 k_i 的概率 $\Pr[\text{AskPara}_7]$. 可得

$$\Pr[\text{AskH}_7\text{WithSend}_1]\le\Pr[\text{AskPara}_7]\tag{13}$$

$\mathcal{A}$ 只是重放 U_i 此前的登录消息 $<C_1,CID_i,M_i>$. 给定公钥 n',e' 和整数 $c\in_R\mathbb{Z}_n$, 若 $\mathcal{A}$ 能计算出 r_u, 我们可以通过 $\mathcal{A}$ 来构造一个求解 RSA 难题的多项式时间算法 $\mathcal{C}$: 算法 $\mathcal{C}$ 在游戏 Game_7 中运行攻击者 $\mathcal{A}$, 令 $n=n',e=e'$, 并且将用户 U_i 生成的 RSA 密文 C_1 就设定为 c. 如果 $\mathcal{A}$ 在游戏 Game_7 中输出正确的 r_u,

那么 r_u 就是整数 c 对应的 RSA 明文. 此外, 为进一步计算 $r_s = C_2 \oplus \mathcal{H}_0(k_i \parallel r_u)$, $\mathcal{A}$ 需要知道 k_i 的值. 因此有

$$\Pr[\text{AskH}_7\text{WithSend}_1] \leq \Pr[\text{AskPara}_7] * \text{Adv}^{\text{rsa}}(\mathcal{O}(t)) \tag{14}$$

2) 查询 Send(S^j,<C_1, CID_i, M_i)> 由用户 U_i 发起, 设此时 $\mathcal{A}$ 成功的概率为 $\Pr[\text{AskH}_7\text{WithoutSend}_1]$. 在这种情况下, r_s 可由 $\mathcal{A}$ 自己产生. 如果 $\mathcal{A}$ 能进一步计算出 r_u, 与上面的讨论类似, 我们可以通过 $\mathcal{A}$ 来构造一个求解 RSA 难题的多项式时间 t 算法 $\mathcal{C}$. 因此有

$$\Pr[\text{AskH}_7\text{WithoutSend}_1] \leq q_{send}\text{Adv}^{\text{rsa}}(\mathcal{O}(t)) \tag{15}$$

综上所述, 通过关系 (1) ∼ (5) 可得

$$\left|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_0]\right| \leq \frac{(q_{\text{send}} + q_{\text{exe}})^2}{2\phi(n)} + \frac{q_{\text{h}}^2}{2^{\ell+1}} + \frac{2q_{\text{send}}}{2^{\ell}} \tag{16}$$

通过关系 (6) ∼ (12) 可得

$$\left|\Pr[\text{Succ}_7] - \Pr[\text{Succ}_4]\right| \leq \Pr[\text{AskPara}_5] + \Pr[\text{AskAuth}_6] + \Pr[\text{AskH}_7] \tag{17}$$

根据定义有

$$\Pr[\text{AskAuth}_7] \leq \Pr[\text{AskH}_7] \tag{18}$$

综合得

$$\begin{aligned}
\text{Adv}^{\text{ake}}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) &= 2\Pr[Succ_0] - 1 = 2\Pr[\text{Succ}_7] - 1 + 2(\Pr[\text{Succ}_0] - \Pr[\text{Succ}_7]) \\
&= 2 \cdot \{(\Pr[\text{Succ}_0] - \Pr[\text{Succ}_1]) + (\Pr[\text{Succ}_1] - \Pr[\text{Succ}_2]) + \cdots + (\Pr[\text{Succ}_6] - \Pr[\text{Succ}_7])\} \\
&\leq 2 \cdot \{|\Pr[\text{Succ}_0] - \Pr[\text{Succ}_1]| + |\Pr[\text{Succ}_1] - \Pr[\text{Succ}_2]| + \cdots + |\Pr[\text{Succ}_6] - \Pr[\text{Succ}_7]|\} \\
&\leq \frac{2q_{\text{send}}}{|\mathcal{D}|} + 6q_{\text{send}}\text{Adv}^{\text{rsa}}(\mathcal{O}(t)) + \frac{q_{\text{h}}^2 + 6q_{\text{send}}}{2^l} + \frac{(q_{\text{send}} + q_{\text{exe}})^2}{\phi(n)}.
\end{aligned}$$

定理 1 得证.

附录 2

定理 2 的证明思想和过程与定理 1 类似. 除了附录 1 中定义的四个事件 (即 $\mathbf{Succ}_{\boldsymbol{n}}$、$\mathbf{AskPara}_{\boldsymbol{n}}$、$\mathbf{AskAuth}_{\boldsymbol{n}}$ 和 $\mathbf{AskH}_{\boldsymbol{n}}$, 其中 $n = 0, 1, 2, \cdots, 6$), 我们再定义一个事件:

$\mathbf{Auth}_{\boldsymbol{n}}$: $\mathcal{A}$ 在游戏 Game_n 中成功计算出认证元 C_3 或 C_4, 并且被对应的接收方接受.

首先, 根据节 4.1 中认证性的定义, 有

$$\text{Adv}^{\text{auth}}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) = \Pr[\text{Auth}_0] \tag{19}$$

接下来, 我们采用与附录 1 中的一系列游戏 $\text{Game}_0 \sim \text{Game}_7$ 来确定 $\text{Adv}^{\text{auth}}_{\mathcal{P},\mathcal{D}}(\mathcal{A})$ 的上限. 根据关系 (1) ∼ (12) 可得

$$\left|\Pr[\text{Auth}_1] - \Pr[\text{Auth}_0]\right| = 0 \tag{20}$$

$$\left|\Pr[\text{Auth}_2] - \Pr[\text{Auth}_1]\right| \leq \frac{(q_{\text{send}} + q_{\text{exe}})^2}{2\phi(n)} + \frac{q_{\text{h}}^2}{2^{\ell+1}} \tag{21}$$

$$\left|\Pr[\text{Auth}_3] - \Pr[\text{Auth}_2]\right| \leq \frac{q_{\text{send}}}{2^{\ell}} \tag{22}$$

$$\left|\Pr[\text{Auth}_4] - \Pr[\text{Auth}_3]\right| \leq \frac{q_{\text{send}}}{2^{\ell}} \tag{23}$$

$$\left|\Pr[\text{Auth}_5] - \Pr[\text{Auth}_4]\right| \leq \Pr[\text{AskPara}_5] \tag{24}$$

$$\left|\Pr[\text{Auth}_6] - \Pr[\text{Auth}_5]\right| \leq \Pr[\text{AskAuth}_6] \leq 2\Pr[\text{AskH}_7] \tag{25}$$

$$\Pr[\text{Auth}_7] = \Pr[\text{Auth}_6] = 0 \tag{26}$$

综合关系 (19) ∼ (26) 可得

$$\begin{aligned}
\text{Adv}^{\text{auth}}_{\mathcal{P}}(\mathcal{A}) &= (\Pr[\text{Auth}_0] - \Pr[\text{Auth}_1]) + (\Pr[\text{Auth}_1] - \Pr[\text{Auth}_3]) + \cdots + (\Pr[\text{Auth}_5] - \Pr[\text{Auth}_6]) + \Pr[\text{Auth}_6] \\
&\leq |\Pr[\text{Auth}_1] - \Pr[\text{Auth}_0]| + |\Pr[\text{Auth}_2] - \Pr[\text{Auth}_1]| + \cdots + |\Pr[\text{Auth}_6] - \Pr[\text{Auth}_5]| + |\Pr[\text{Auth}_6]| \\
&\leq \frac{q_{\text{send}}}{|\mathcal{D}|} + 2q_{\text{send}}\text{Adv}^{\text{rsa}}(\mathcal{O}(t)) + \frac{q_{\text{h}}^2 + 6q_{\text{send}}}{2^{\ell+1}} + \frac{(q_{\text{send}} + q_{\text{exe}})^2}{2\phi(n)}.
\end{aligned}$$

定理 2 得证.