

多服务器环境下基于椭圆曲线密码的认证协议

刘紫瑶¹, 陈建华¹, 韦永霜²

(1. 武汉大学数学与统计学院, 湖北 武汉 430072)

(2. 工业和信息化部电子第五研究所, 广东 广州 510000)

摘要: 本文研究了多服务器环境下的认证密钥协商协议. 通过对 Andola 等人方案的分析, 利用椭圆曲线密码和随机数设计了一种改进的认证密钥协商协议. 使用 BAN 逻辑和非形式化分析方法, 对协议进行安全性分析. 实验结果表明, 所提协议具有更好的安全性和实用性.

关键词: 多服务器; 椭圆曲线密码; 口令; 密钥协商

MR(2010) 主题分类号: 68P25

中图分类号: O29

文献标识码: A

文章编号: 0255-7797(2024)03-0212-13

1 引言

随着互联网和无线通信技术的快速发展, 用户对网络服务的需求也不断增长. 然而, 开放的互联网环境存在着巨大的安全隐患, 敌手通过篡改、窃听、重放等可以获取用户的个人信息. 因此, 身份认证和密钥协商对于安全通信至关重要. 传统的认证协议大多基于单服务器架构, 用户需要注册每个服务器并记住多个身份凭证, 这种方式效率低, 容易造成用户身份信息的泄露. 为解决这个问题, 研究者们提出了基于多服务器架构的认证协议. 在多服务器环境下, 用户仅需在注册中心注册一次, 就可以访问所有应用服务器. 但多服务器系统也会带来新的问题, 用户面临的攻击不仅有来自外部的, 还有来自内部的攻击. 因此, 在多服务器环境下设计一个安全高效的认证协议具有重大意义.

2001 年, Li 等人^[1]首次提出一种基于神经网络的多服务器认证方案, 该方案计算量大, 存储成本高, 且不提供口令更新功能. 接着, Lin 等人^[2], Juang 等人^[3], Chang 和 Lee^[4], Tsai^[5]提出了他们自己的多服务器认证方案. 然而, 这些方案在登录阶段用户使用的 ID 是静态的, 无法实现用户身份的不可追踪性. 2009 年, Liao 和 Wang^[6]提出了一种基于哈希函数和动态 ID 的认证方案, 实现了用户的匿名性, 但容易遭受服务器欺骗攻击, 内部攻击和伪装攻击. 为解决这一问题, Hsiang 和 Shih^[7]提出了一个改进方案. 很快 Sood 等人^[8]和 Lee 等人^[9]指出 Hsiang 和 Shih 的方案无法抵抗口令猜测攻击和重放攻击. 随后, 针对不同的应用环境, 提出了几种轻量级的认证协议^[10-15], 这些协议中仅涉及哈希和对称加密运算, 计算成本低, 但不能提供一些重要的安全属性, 如前向安全性和双因素安全性. 为了提高安全性, 公钥密码被广泛用于多服务器架构下认证协议的设计. 2013 年, Wang 和 Ma^[16]提出一种基于椭圆曲线离散对数问题的认证协议, 但该协议无法抵抗服务器欺骗攻击和离线口令猜测攻击^[17]. Truong 等人^[18]提出一种新方案, 该方案由注册中心选择一个随机数作为用户的口令.

*收稿日期: 2023-03-23

接收日期: 2023-05-22

基金项目: 国家重点研发计划项目 (2021YFB2012205).

作者简介: 刘紫瑶 (1999-), 女, 湖南邵阳, 硕士, 主要研究方向: 密码与信息安全, E-mail: zyliu04@163.com.

然而, 该方案不能抵抗离线口令猜测攻击和假冒攻击. Kumari 和 Om^[19] 利用 RSA 签名设计了一个认证方案. Irshad 等人^[20] 提出了一种基于椭圆曲线密码和对称加密的认证方案, 并使用 GNY 逻辑进行形式化安全分析. 随后, Ying 和 Nayak^[21] 提出了一种基于 5G 网络的椭圆曲线密码认证协议. 但 Haq 等人^[22] 发现该协议不能真正实现用户身份的不可追踪性, 提出了一个改进方案, 改进方案在认证和密钥协商阶段中, 参与者不传输任何静态参数.

最近, Andola 等人^[23] 提出了一种基于智能卡和动态身份的匿名认证方案, 宣称所提出的方案可以抵抗各类已知的攻击. 然而, 经过分析发现该方案容易遭受离线口令猜测攻击, 服务器欺骗攻击, 同时无法实现用户匿名性以及前向安全性. 为解决上述问题, 本文设计了一个基于椭圆曲线密码的认证密钥协商协议.

2 预备工作

2.1 椭圆曲线密码

给定两个大素数 p 和 q , F_p 是一个有限域, E 是定义在 F_p 上的椭圆曲线, G 是 E 上的 q 阶子群. 那么, 我们可以给出椭圆曲线上的两个困难问题^[24]:

(1) 椭圆曲线离散对数问题 (Elliptic curve discrete logarithm problem, ECDLP): 给定两个点 $P \in G, Q \in G$, 求整数 $a \in Z_q^*$, 使得 $Q = aP$ 成立.

(2) 椭圆曲线计算性 Diffie - Hellman 问题 (Elliptic curve computational Diffie - Hellman problem, ECDHP): 给定 $P \in G, aP \in G, bP \in G$ ($a, b \in Z_q^*$ 是随机数), 计算 $abP \in G$.

2.2 模糊验证器

在基于智能卡和口令的认证协议中, 为实现口令本地自由更新, 通常需要在智能卡中存储有关口令的安全参数, 来检测用户输入口令是否正确. 然而, 这种方法带来了新的安全问题, 如果智能卡丢失, 协议容易遭受离线口令猜测攻击. 为解决这一问题, 可以采用模糊验证器 (fuzzy-verifier)^[25].

假设 A_i 是口令验证参数, 将 A_i 修改为 $A_i \bmod n$, 其中 $2^4 \leq n \leq 2^8$. 如果敌手想要从 A_i 中猜测口令, 将会得到 $\frac{|\mathcal{D}_{ID}| * |\mathcal{D}_{PW}|}{n}$ 个可能的 (ID_i^*, PW_i^*) , 他们都满足 $A_i^* = A_i$, 其中 $|\mathcal{D}_{ID}|, |\mathcal{D}_{PW}|$ 分别表示用户身份和口令的空间规模. 因此, 敌手无法确定真正的 (ID_i, PW_i) , 可以有效抵抗离线口令猜测攻击.

3 Andola 等人协议的回顾

Andola 等人^[23] 的协议涉及三个参与方: 注册中心 RC , 服务器 S_j 和用户 U_i . RC 选取系统主密钥 x 和秘密参数 y , 计算 $h(y), h(x||y)$, 然后将这两个参数通过安全信道发送给 S_j , S_j 将其秘密保存.

协议由 4 个阶段组成: 注册阶段, 登录阶段, 认证阶段和口令更新阶段. 具体步骤如下:

3.1 注册阶段

用户 U_i 通过安全信道向注册中心 RC 申请注册, 过程如下:

步骤 1. 用户 U_i 选择身份 ID_i , 口令 PW_i , 以及随机数 b . 计算 $RPW_i = h(b \oplus PW_i)$,

然后将消息 $\{ID_i, RPW_i\}$ 通过安全信道发送给 RC .

步骤 2. 注册中心 RC 收到消息后, 选择一个随机数 Z_i , 计算 $A_i = h(x||ID_i)$, $B_i = Z_i \oplus ID_i \oplus RPW_i$, $C_i = h(RPW_i||ID_i||Z_i) \oplus B_i$, $E_i = h(h(x||y)||A_i) \oplus Z_i$. 将 $\{A_i, B_i, C_i, E_i, h(y), h(\cdot)\}$ 写入智能卡 SC_i 中, 并将 SC_i 通过安全信道发送给 U_i .

步骤 3. U_i 将 b 存储到智能卡中.

3.2 登录阶段

登录阶段分为如下 3 个步骤:

步骤 1. 当用户 U_i 登录服务器 S_j 时, 将智能卡 SC_i 插入读卡机并输入 ID_i, PW_i , 以及 S_j 的身份 SID_j .

步骤 2. SC_i 计算 $Z_i = ID_i \oplus h(b \oplus PW_i) \oplus B_i$, $C'_i = h(h(b \oplus PW_i)||ID_i||Z_i) \oplus B_i$, 然后判断 $C'_i \stackrel{?}{=} C_i$. 如果不相等, 会话终止.

步骤 3. SC_i 选择随机数 N_i , 计算 $P_{ij} = h(N_i||h(y)||SID_j) \oplus A_i$, $CID_i = E_i \oplus h(N_i||SID_j||A_i)$, $M_1 = h(Z_i||N_i||E_i)$, $M_2 = N_i \oplus h(SID_j||h(y))$. 然后通过公共信道将消息 $\{P_{ij}, CID_i, M_1, M_2\}$ 发送给服务器 S_j .

3.3 认证阶段

认证阶段分为如下 4 个步骤:

步骤 1. 当收到登录请求 $\{P_{ij}, CID_i, M_1, M_2\}$ 后, S_j 计算 $N_i = M_2 \oplus h(SID_j||h(y))$, $A_i = h(N_i||h(y)||SID_j) \oplus P_{ij}$, $E_i = CID_i \oplus h(N_i||SID_j||A_i)$, $Z'_i = h(h(x||y)||A_i) \oplus E_i$, $M'_1 = h(Z'_i||N_i||E_i)$, 然后判断 $M'_1 \stackrel{?}{=} M_1$. 如果不相等, 会话终止.

步骤 2. S_j 选择随机数 N_j , 计算 $M_3 = h(Z'_i||N_i||SID_j||CID_i)$, $M_4 = N_i \oplus N_j \oplus A_i$. 然后通过公共信道将消息 $\{M_3, M_4\}$ 发送给用户 U_i .

步骤 3. U_i 收到消息 $\{M_3, M_4\}$ 后, 计算 $N_j = M_4 \oplus A_i \oplus N_i$, $M'_3 = h(Z_i||N_i||SID_j||CID_i)$, 然后判断 $M'_3 \stackrel{?}{=} M_3$. 如果不相等, 会话终止. 如果相等, 计算 $M_5 = h(Z_i||N_j||SID_j||CID_i)$, 会话密钥 $SK = h(Z_i||SID_j||N_i||N_j||CID_i)$, 然后通过公共信道将消息 $\{M_5\}$ 发送给 S_j .

步骤 4. S_j 收到消息 $\{M_5\}$ 后, 计算 $M'_5 = h(Z'_i||N_j||SID_j||CID_i)$, 然后判断 $M'_5 \stackrel{?}{=} M_5$. 如果不相等, 会话终止. 如果相等, S_j 和 U_i 实现了相互认证并协商出共同的会话密钥 $SK = h(Z'_i||SID_j||N_i||N_j||CID_i)$.

3.4 口令更新阶段

如果用户 U_i 想要将口令 PW_i 修改为 PW_i^{new} 时, 执行如下过程:

步骤 1. 用户将智能卡 SC_i 插入读卡机, 输入 ID_i, PW_i .

步骤 2. SC_i 计算 $Z_i = ID_i \oplus h(b \oplus PW_i) \oplus B_i$, $C_i^* = h(h(b \oplus PW_i)||ID_i||Z_i) \oplus B_i$, 然后判断 $C_i^* \stackrel{?}{=} C_i$. 如果相等, 提示用户输入新口令 PW_i^{new} .

步骤 3. SC_i 选择一个新的随机数 b^{new} , 计算 $RPW_i^{new} = h(b^{new} \oplus PW_i^{new})$, $B_i^{new} = Z_i \oplus ID_i \oplus RPW_i^{new}$, $C_i^{new} = h(RPW_i^{new}||ID_i||Z_i) \oplus B_i^{new}$. 然后, SC_i 用 $B_i^{new}, C_i^{new}, b^{new}$ 替换卡内的 B_i, C_i, b .

4 Andola 等人协议的安全缺陷

4.1 离线口令猜测攻击

假设敌手 \mathcal{A} 拿到了用户 U_i 的智能卡, 通过边信道技术可以提取卡内的秘密信息 $\{B_i, C_i, h(\cdot), b\}$. 那么 \mathcal{A} 可以发起离线口令猜测攻击, 得到用户的口令 PW_i , 具体过程如下:

步骤 1. \mathcal{A} 从用户身份空间 \mathcal{D}_{ID} 和口令空间 \mathcal{D}_{PW} 中猜测 (ID_i^*, PW_i^*) .

步骤 2. \mathcal{A} 计算 $RPW_i^* = h(b \oplus PW_i^*)$, $Z_i^* = B_i \oplus ID_i^* \oplus RPW_i^*$, $C_i^* = h(RPW_i^* || ID_i^* || Z_i^*) \oplus B_i$.

步骤 3. \mathcal{A} 检查 $C_i^* \stackrel{?}{=} C_i$. 若相等, 则 \mathcal{A} 猜测的 (ID_i^*, PW_i^*) 正确. 否则, 跳转步骤 1.

4.2 服务器欺骗攻击

假设敌手 \mathcal{A} 拿到了用户 U_i 的智能卡并提取卡内的秘密信息 $\{A_i, B_i, C_i, h(y), h(\cdot), b\}$, 同时, \mathcal{A} 拦截 U_i 发送给服务器 S_j 的登录信息 $\{CID_i, M_2\}$. 那么 \mathcal{A} 可以发起服务器欺骗攻击, 假冒一个合法的服务器. 具体过程如下:

步骤 1. 由 4.1 节可知, \mathcal{A} 可以通过离线口令猜测攻击得到用户的身份和口令 (ID_i, PW_i) , 从而可以计算出 $RPW_i = h(b \oplus PW_i)$, $Z_i = B_i \oplus ID_i \oplus RPW_i$.

步骤 2. \mathcal{A} 选取随机数 N_j^* , 计算 $N_i = M_2 \oplus h(SID_j || h(y))$, $M_3^* = h(Z_i || N_i || SID_j || CID_i)$, $M_4^* = N_i \oplus N_j^* \oplus A_i$. 然后 \mathcal{A} 将消息 $\{M_3^*, M_4^*\}$ 发送给用户 U_i .

步骤 3. U_i 计算 M_3' , 将会得到 $M_3' = M_3^*$. 因此, U_i 和 \mathcal{A} 协商出一个共同的会话密钥 $SK = h(Z_i || SID_j || N_i || N_j^* || CID_i)$.

4.3 匿名性失效

假设敌手 \mathcal{A} 是一个恶意用户, 可以提取自己智能卡中的参数 $h(y)$. 同时 \mathcal{A} 可以窃听到用户 U_i 发送给服务器 S_j 的登录信息 $\{P_{ij}, M_2\}$. 所以, 可以计算出 $N_i = M_2 \oplus h(SID_j || h(y))$, $A_i = h(N_i || h(y) || SID_j) \oplus P_{ij}$. 因为 $A_i = h(x || ID_i)$, 是一个与用户身份 ID_i 相关的固定参数, 可以追踪 U_i 的访问行为. 因此, 用户匿名性失效.

4.4 前向安全性问题

假设敌手 \mathcal{A} 可以从恶意服务器中获得长期私钥 $h(y)$ 和 $h(x || y)$, 同时 \mathcal{A} 可以窃听到用户 U_i 发送给服务器 S_j 的登录信息 $\{P_{ij}, CID_i, M_2\}$ 以及 S_j 发送给 U_i 的响应消息 $\{M_4\}$. 那么, \mathcal{A} 可以计算出 $N_i = M_2 \oplus h(SID_j || h(y))$, $A_i = h(N_i || h(y) || SID_j) \oplus P_{ij}$, $E_i = CID_i \oplus h(N_i || SID_j || A_i)$, $Z_i = h(h(x || y) || A_i) \oplus E_i$, $N_j = M_4 \oplus A_i \oplus N_i$. 因此, \mathcal{A} 可以获得会话密钥 $SK = h(Z_i || SID_j || N_i || N_j || CID_i)$. 因此, 无法实现前向安全性.

5 改进方案

针对文献 [23] 的安全缺陷, 本文设计了一种基于椭圆曲线密码的认证密钥协商协议, 包括注册中心 RC , 服务器 S_j , 用户 U_i 三个参与方. 以下是详细步骤:

5.1 系统初始化阶段

RC 在素域 F_p 上选择一个椭圆曲线 E , P 是 E 上阶为 q 的生成元. RC 选取私钥 x , 单向哈希函数 $h: \{a, b\}^* \rightarrow Z_q^*$, 和整数 n , 其中 $2^4 \leq n \leq 2^8$. RC 将参数 $\{E, P, h(\cdot), n\}$ 公开.

5.2 服务器注册阶段

服务器 S_j 通过安全信道向注册中心 RC 申请注册, 其过程如图 1 所示.

步骤 1. 服务器 S_j 选择身份 SID_j , 然后将消息 $\{SID_j\}$ 通过安全信道发送给 RC .

步骤 2. 注册中心 RC 收到消息后, 检查 $h(SID_j||x)$ 是否存在于身份验证表 Γ_1 中. 如果存在, RC 拒绝 S_j 的注册请求, 要求 S_j 提供新的身份标识. 否则, RC 选择一个随机数 r_j , 计算 S_j 的私钥 $x_j = h(SID_j||x||r_j)$, 公钥 $Q_j = x_jP$. 然后, 把 $\{h(SID_j||x)\}$ 存储到服务器身份验证表 Γ_1 中, 公开参数 $\{SID_j, Q_j\}$. 最后, 将 $\{x_j\}$ 通过安全信道发送给 S_j .

步骤 3. S_j 秘密保存 x_j .

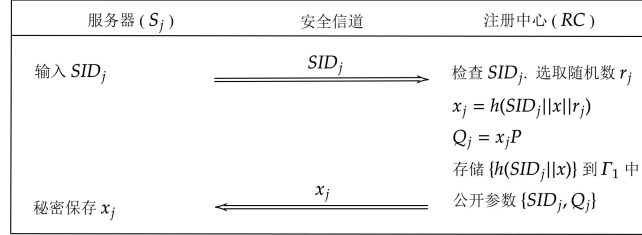


图 1: 所提出方案的服务器注册阶段

5.3 用户注册阶段

用户 U_i 通过安全信道向注册中心 RC 申请注册, 其过程如图 2 所示.

步骤 1. 用户 U_i 选择身份 ID_i , 口令 PW_i , 以及随机数 b . 计算 $RPW_i = h(PW_i||b)$, 然后将消息 $\{ID_i, RPW_i\}$ 通过安全信道发送给 RC .

步骤 2. 注册中心 RC 收到消息后, 检查 $h(ID_i||x)$ 是否存在于身份验证表 Γ_2 中. 如果存在, RC 拒绝 U_i 的注册请求, 要求 U_i 提供新的身份标识. 否则, RC 选择一个随机数 r_i , 计算 U_i 的私钥 $x_i = h(ID_i||x||r_i)$, 公钥 $Q_i = x_iP$, $A_i = h((h(ID_i) \oplus RPW_i) \bmod n)$, $B_i = x_i \oplus RPW_i$. RC 把 $\{h(ID_i||x), T_i = 1\}$ 存储到用户身份验证表 Γ_2 中, 其中 $T_i = 1$ 意味着用户 U_i 只注册过一次并且智能卡处于激活状态. 然后, RC 公开参数 $\{ID_i, Q_i\}$, 将 $\{A_i, B_i, n, h(\cdot)\}$ 写入智能卡 SC_i 中, 并将 SC_i 通过安全信道发送给 U_i .

步骤 3. U_i 收到智能卡 SC_i 后, 计算 $\tilde{b} = b \oplus (h(ID_i||PW_i) \bmod n)$, 然后将 \tilde{b} 存储到智能卡 SC_i 中.

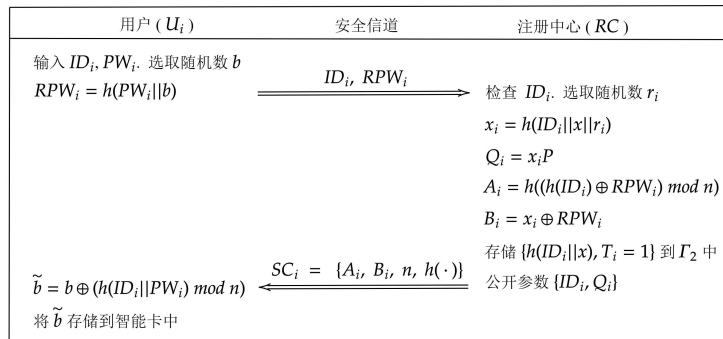


图 2: 所提出方案的用户注册阶段

5.4 登录阶段

图 3 描述了方案的登录与认证阶段. 登录阶段分为如下 3 个步骤:

步骤 1. 当用户 U_i 登录服务器 S_j 时, 将智能卡 SC_i 插入读卡机并输入 ID_i , PW_i 和 SID_j .

步骤 2. SC_i 计算 $b = \tilde{b} \oplus (h(ID_i || PW_i) \bmod n)$, $RPW_i^* = h(PW_i || b)$, $A_i^* = h((h(ID_i) \oplus RPW_i^*) \bmod n)$, 然后判断 $A_i^* \stackrel{?}{=} A_i$. 如果不相等, 会话终止.

步骤 3. SC_i 选择两个随机数 a_i, b_i , 计算 $x_i = B_i \oplus RPW_i$, $C_i = a_i P = (C_x, C_y)$, $D_{ij} = a_i Q_j$, $E_i = x_i h(ID_i) + a_i C_x \bmod q$, $F_i = b_i P$, $CID_i = (ID_i || E_i) \oplus h(D_{ij})$. 然后通过公共信道将消息 $\{C_i, F_i, CID_i\}$ 发送给服务器 S_j .

5.5 认证阶段

认证阶段分为如下 4 个步骤:

步骤 1. 当收到登录请求 $\{C_i, F_i, CID_i\}$ 后, S_j 计算 $D_{ji} = x_j C_i$, $(C_x, C_y) = C_i$, $ID_i || E_i = CID_i \oplus h(D_{ji})$, 然后判断 $E_i P \stackrel{?}{=} Q_j h(ID_i) + C_x C_i$. 如果不相等, 会话终止.

步骤 2. S_j 选择随机数 d_j , 计算 $H_j = d_j P$, $K_{ji} = d_j F_i$, 会话密钥 $SK = h(K_{ji} || E_i)$, $L_j = h(SK || D_{ji} || F_i)$. 然后通过公共信道将消息 $\{H_j, L_j\}$ 发送给用户 U_i .

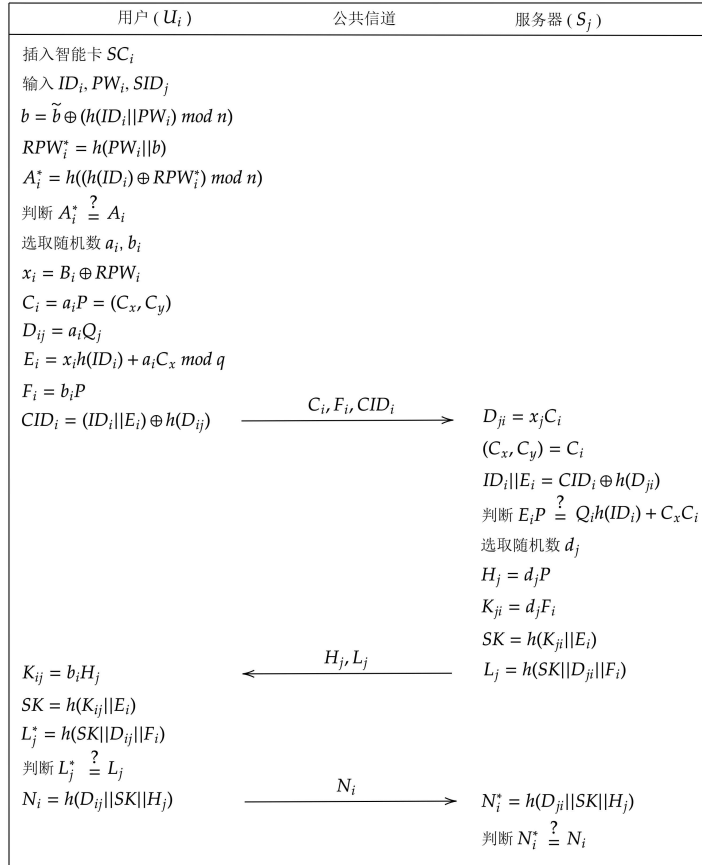


图 3: 所提出方案的登录与认证阶段

步骤 3. U_i 收到消息 $\{H_j, L_j\}$ 后, 计算 $K_{ij} = b_i H_j$, $SK = h(K_{ij} || E_i)$, $L_j^* = h(SK || D_{ij} || F_i)$, 然后判断 $L_j^* \stackrel{?}{=} L_j$. 如果不相等, 会话终止. 如果相等, 计算 $N_i = h(D_{ij} || SK || H_j)$, 然后

通过公共信道将消息 $\{N_i\}$ 发送给服务器 S_j .

步骤 4. S_j 收到消息 $\{N_i\}$ 后, 计算 $N_i^* = h(D_{ji}||SK||H_j)$, 然后判断 $N_i^* \stackrel{?}{=} N_i$. 如果不相等, 会话终止. 如果相等, S_j 和 U_i 实现了相互认证并协商出共同的会话密钥 SK .

5.6 口令更新阶段

如果用户 U_i 想要将口令 PW_i 修改为 PW_i^{new} 时, 执行如下过程:

步骤 1. 用户将智能卡 SC_i 插入读卡机, 输入 ID_i, PW_i .

步骤 2. SC_i 计算 $b^* = \tilde{b} \oplus (h(ID_i||PW_i) \bmod n)$, $RPW_i^* = h(PW_i||b^*)$, $A_i^* = h((h(ID_i) \oplus RPW_i^*) \bmod n)$, 然后判断 $A_i^* \stackrel{?}{=} A_i$. 如果相等, 提示用户输入新口令 PW_i^{new} .

步骤 3. SC_i 选择一个新的随机数 b^{new} , 计算 $RPW_i^{new} = h(PW_i^{new}||b^{new})$, $A_i^{new} = h((h(ID_i) \oplus RPW_i^{new}) \bmod n)$, $B_i^{new} = B_i \oplus RPW_i^* \oplus RPW_i^{new}$, $\tilde{b}^{new} = b^{new} \oplus (h(ID_i||RPW_i^{new}) \bmod n)$. 然后, SC_i 用 $A_i^{new}, B_i^{new}, \tilde{b}^{new}$ 替换卡内的 A_i, B_i, \tilde{b} .

5.7 智能卡撤销和重新注册阶段

当用户 U_i 的智能卡丢失或被盗时, U_i 可以撤销智能卡并使用同一身份 ID_i 重新进行注册.

在智能卡撤销阶段, 用户需要向注册中心 RC 提交注销申请和身份凭证. RC 收到申请后, 更新验证表 Γ_2 中的数据 $\{h(ID_i||x), T_i = 0\}$, 其中 $T_i = 0$ 意味着用户 U_i 被撤销, 智能卡处于未激活状态.

如果 U_i 想要使用身份 ID_i 重新注册, 需要向注册中心 RC 提交重新注册申请和身份凭证. RC 收到申请后, 按照用户注册阶段的步骤进行注册, 更新验证表 Γ_2 中的数据 $\{h(ID_i||x), T_i = T_i + 1\}$, 公开用户新的公钥信息 $\{ID_i, Q_i^{new}\}$, 并且给 U_i 颁发一个新的智能卡.

6 协议的安全性分析及性能分析

6.1 BAN 逻辑验证

本文利用 BAN 逻辑 [26] 对协议进行形式化安全分析. 表 1 为 BAN 逻辑的基本符号与含义, 表 2 为 BAN 逻辑的推理规则.

本文方案的具体分析过程如下所示:

(1) 协议模型的理想化

消息 1. $U_i \rightarrow S_j: \{ID_i, \{ID_i, a_i, C_i\}_{x_i}\}_{U_i \xleftrightarrow{D_{ij}} S_j}, C_i, F_i$

消息 2. $S_j \rightarrow U_i: \{U_i \xleftrightarrow{SK} S_j, F_i\}_{U_i \xleftrightarrow{D_{ij}} S_j}, H_j$

消息 3. $U_i \rightarrow S_j: \{U_i \xleftrightarrow{SK} S_j, H_j\}_{U_i \xleftrightarrow{D_{ij}} S_j}$

(2) 安全目标

目标 1. $U_i | \equiv S_j | \equiv (U_i \xleftrightarrow{SK} S_j)$

目标 2. $U_i | \equiv (U_i \xleftrightarrow{SK} S_j)$

目标 3. $S_j | \equiv U_i | \equiv (U_i \xleftrightarrow{SK} S_j)$

目标 4. $S_j | \equiv (U_i \xleftrightarrow{SK} S_j)$

表 1 BAN 逻辑的基本符号与含义

符号	含义
P, Q	参与通信的主体
X, Y	消息语句
K	密钥
(X, Y)	X 与 Y 的连接
$P \equiv X$	P 相信 X 为真
$P \triangleleft X$	P 收到消息 X
$P \sim X$	P 发送过消息 X
$P \Rightarrow X$	P 对 X 有控制权
$\sharp(X)$	X 是新鲜的
$P \xleftrightarrow{K} Q$	P 与 Q 共享密钥 K
$\{X\}_K$	用密钥 K 加密 X

表 2 BAN 逻辑的推理规则

规则	含义
消息含义规则	$\frac{P \equiv P \xleftrightarrow{K} Q, P \triangleleft \{X\}_K}{P \equiv Q \mid \sim X}$
临时值校验规则	$\frac{P \mid \equiv \sharp(X), P \mid \equiv Q \mid \sim X}{P \mid \equiv Q \mid \equiv X}$
管辖权规则	$\frac{P \mid \equiv Q \mid \Rightarrow (X), P \mid \equiv Q \mid \equiv X}{P \mid \equiv X}$
新鲜性规则	$\frac{P \mid \equiv \sharp(X)}{P \mid \equiv \sharp(X, Y)}$
信念规则	$\frac{P \mid \equiv Q \mid \equiv (X, Y)}{P \mid \equiv Q \mid \equiv X}$

(3) 初始化假设

假设 1. $U_i \mid \equiv \sharp(b_i)$ 假设 2. $S_j \mid \equiv \sharp(d_j)$ 假设 3. $U_i \mid \equiv (U_i \xleftrightarrow{D_{ij}} S_j)$ 假设 4. $S_j \mid \equiv (U_i \xleftrightarrow{D_{ij}} S_j)$ 假设 5. $U_i \mid \equiv S_j \mid \Rightarrow (U_i \xleftrightarrow{SK} S_j)$ 假设 6. $S_j \mid \equiv U_i \mid \Rightarrow (U_i \xleftrightarrow{SK} S_j)$

(4) 证明过程

步骤 1. 根据消息 2 有: $U_i \triangleleft \{U_i \xleftrightarrow{SK} S_j, F_i\}_{U_i \xleftrightarrow{D_{ij}} S_j}$.步骤 2. 根据步骤 1, 假设 3 和消息含义规则有: $U_i \mid \equiv S_j \mid \sim (U_i \xleftrightarrow{SK} S_j, F_i)$.步骤 3. 根据假设 1 和新鲜性规则有: $U_i \mid \equiv \sharp(U_i \xleftrightarrow{SK} S_j, F_i)$.步骤 4. 根据步骤 2, 步骤 3 和临时值校验规则有: $U_i \mid \equiv S_j \mid \equiv (U_i \xleftrightarrow{SK} S_j, F_i)$.步骤 5. 根据步骤 4 和信念规则有: $U_i \mid \equiv S_j \mid \equiv (U_i \xleftrightarrow{SK} S_j)$, 从而目标 1 得证.步骤 6. 根据步骤 5, 假设 5 和管辖权规则有: $U_i \mid \equiv (U_i \xleftrightarrow{SK} S_j)$, 从而目标 2 得证.步骤 7. 根据消息 3 有: $S_j \triangleleft \{U_i \xleftrightarrow{SK} S_j, H_j\}_{U_i \xleftrightarrow{D_{ij}} S_j}$.步骤 8. 根据步骤 7, 假设 4 和消息含义规则有: $S_j \mid \equiv U_i \mid \sim (U_i \xleftrightarrow{SK} S_j, H_j)$.步骤 9. 根据假设 2 和新鲜性规则有: $S_j \mid \equiv \sharp(U_i \xleftrightarrow{SK} S_j, H_j)$.步骤 10. 根据步骤 8, 步骤 9 和临时值校验规则有: $S_j \mid \equiv U_i \mid \equiv (U_i \xleftrightarrow{SK} S_j, H_j)$.

步骤 11. 根据步骤 10 和信念规则有: $S_j | \equiv U_i | \equiv (U_i \xrightarrow{SK} S_j)$, 从而目标 3 得证.

步骤 12. 根据步骤 11, 假设 6 和管辖权规则有: $S_j | \equiv (U_i \xrightarrow{SK} S_j)$, 从而目标 4 得证.

6.2 非形式化安全分析

(1) 相互认证与密钥协商

在认证过程中, 服务器 S_j 通过检查 $E_i P \stackrel{?}{=} Q_i h(ID_i) + C_x C_i$, 来验证用户 U_i 身份的合法性. 由于 E_i 是由 U_i 的私钥 x_i 生成的签名, 基于 ECDLP 问题是不可解的, 只有合法用户才能生成正确签名通过服务器的认证. 用户 U_i 通过检查 $L_j^* \stackrel{?}{=} L_j$, 来验证 S_j 身份的合法性. 事实上, $L_j = h(SK || a_i x_j P || F_i)$, 只有用户 U_i 和拥有私钥 x_j 的服务器才能计算出合法的验证消息 L_j . 因此, 本文方案实现了用户和服务器的相互认证. 另一方面, U_i 和 S_j 共同协商出一个会话密钥 $SK = h(K_{ij} || E_i)$. 由于 $K_{ij} = b_i H_j = d_j F_i$, 基于 ECDHP 问题是不可解的, 保证会话密钥的正确性.

(2) 用户匿名性

在登录和认证阶段, 敌手可以窃听会话信息, 其中, 与用户身份相关的消息有: CID_i, L_j, N_i . L_j 和 N_i 受到单向哈希函数的保护, 无法从中得到 ID_i . 而 $CID_i = (ID_i || E_i) \oplus h(D_{ij})$ 且 $D_{ij} = a_i x_j P$, 由于 ECDHP 问题是不可解的, 敌手无法通过 $a_i P$ 和 $x_j P$ 计算出 D_{ij} . 因此, 不能解密 ID_i . 另一方面, CID_i, L_j 和 N_i 中有随机数 a_i, b_i, d_j 参与运算, 具有动态性, 无法判断两次会话是否由同一用户发起, 可以实现用户身份的不可追踪.

(3) 前向安全性

用户 U_i 和服务器 S_j 通过相互认证协商出会话密钥 $SK = h(b_i d_j P || E_i)$, 而敌手只能通过窃听公共信道中的消息得到 $F_i = b_i P, H_j = d_j P$, 由于 ECDHP 问题是不可解的, 敌手无法计算出 $b_i d_j P$, 可以实现前向安全性.

(4) 抗离线口令猜测攻击

假设敌手可以获取用户智能卡, 并通过侧信道技术提取卡内存储的秘密信息 $\{A_i, B_i, n, h(\cdot), \tilde{b}\}$. 同时, 敌手可以窃听公共信道中传输的消息 $\{C_i, F_i, CID_i\}$. 我们从以下两个方面阐述本文方案可以抵抗离线口令猜测攻击.

一方面, 如果敌手利用 A_i 来进行离线口令猜测攻击, 需要计算 $b = \tilde{b} \oplus (h(ID_i^* || PW_i^*) \text{ mod } n)$, $RPW_i^* = h(PW_i^* || b)$, $A_i^* = h((h(ID_i^*) \oplus RPW_i^*) \text{ mod } n)$, 通过 A_i^* 是否等于 A_i 来判断 (ID_i^*, PW_i^*) 的正确性. 根据 2.2 节, 由于模糊验证因子的存在, 满足等式成立的 (ID_i^*, PW_i^*) 有很多, 无法通过猜测得到正确的口令.

另一方面, 如果敌手利用 $CID_i = (ID_i || E_i) \oplus h(D_{ij})$ 来进行离线口令猜测攻击, 需要计算出 $D_{ij} = a_i x_j P$, 这对敌手来说相当于解决 ECDHP 问题. 因此, 这种方式也是不可行的.

因此, 本文方案可以抵抗离线口令猜测攻击.

(5) 抗重放攻击

敌手截获合法用户 U_i 发给服务器 S_j 的登录消息 $\{C_i, F_i, CID_i\}$, 将同样的消息发送给 S_j , 可以通过 S_j 的认证. 但只有知道该会话中的随机数 a_i, d_j , 才能计算出会话密钥 SK 以及验证值 M_i . 因而, 敌手无法完成后续流程, 会话终止. 另一方面, 敌手重放 S_j 的响应消息 $\{H_j, L_j\}$ 也是不可行的, 因为该消息依赖于之前用户选择的随机数 a_i 和 b_i , 而随机数在每个会话中都不一样, 此消息无法通过当前用户的认证. 因此, 本文方案可以抵抗重放攻击.

(6) 抗服务器欺骗攻击

敌手可以截获用户发送给合法服务器 S_j 的登录消息, 由于敌手不知道 S_j 的私钥 x_j 或者随机数 a_i , 无法计算出 $D_{ij} = a_i x_j P$, 也就不能伪造出合法的响应消息 $\{H_j, L_j\}$. 因此, 可以抵抗服务器欺骗攻击.

6.3 性能分析

在本节中, 从安全性能, 计算成本两个方面将本文方案与近年来提出的 5 个方案进行比较, 即 Jangirala 等人的方案 [14], Kumari 等人的方案 [19], Irshad 等人的方案 [20], Haq 等人的方案 [22], Andola 等人的方案 [23].

从表 3 安全性能对比可以看出, 本文方案可以满足列出的所有安全属性, 而其他方案或多或少存在着一些安全缺陷. 比如, 方案 [14, 19, 20, 22, 23] 不能提供相互认证, 也不能抵抗服务器欺骗攻击. 方案 [14, 19, 22, 23] 不能实现用户匿名性. 方案 [19, 20, 22, 23] 不能抵抗离线口令猜测攻击. 方案 [14, 22] 不能抵抗重放攻击. 方案 [23] 不能实现前向安全性. 因此, 本文方案拥有更高的安全性.

表 3 安全性能对比

安全性能	方案 [14]	方案 [19]	方案 [20]	方案 [22]	方案 [23]	本文方案
相互认证	否	否	否	否	否	是
密钥协商	是	是	是	是	是	是
用户匿名性	否	否	是	否	否	是
前向安全性	是	是	是	是	否	是
抗重放攻击	否	是	是	否	是	是
抗服务器欺骗攻击	否	否	否	否	否	是
抗离线口令猜测攻击	是	否	否	否	否	是

表 4, 表 5 分别展示了本文与相关方案在计算量和运行时间上的对比结果. 由于用户和服务器仅需注册一次, 且口令更新操作不会频繁发生, 我们只考虑了登录与认证阶段的计算成本. 下面给出所用到的符号 — T_e : 模幂运算的时间复杂度; T_m : 椭圆曲线点乘运算的时间复杂度; T_a : 椭圆曲线点加运算的时间复杂度; T_s : 对称加密/解密的时间复杂度; T_h : 单向哈希函数的时间复杂度. 为了便于评估计算成本, 本文采用了文献 [27] 中的实验数据: $T_e \approx 3.85 \text{ ms}$, $T_m \approx 2.226 \text{ ms}$, $T_a \approx 0.0288 \text{ ms}$, $T_s \approx 0.0046 \text{ ms}$, $T_h \approx 0.0023 \text{ ms}$. 相比于这些运算, 异或运算和连接运算可以忽略不计 [25].

表 4 计算量对比

	用户	服务器	总计
方案 [14]	$11T_h$	$9T_h$	$20T_h$
方案 [19]	$3T_e + 6T_h$	$4T_e + 3T_h$	$7T_e + 9T_h$
方案 [20]	$5T_m + T_a + T_s + 9T_h$	$5T_m + T_a + T_s + 6T_h$	$10T_m + 2T_a + 2T_s + 15T_h$
方案 [22]	$5T_m + T_a + 7T_h$	$5T_m + 2T_a + 4T_h$	$10T_m + 3T_a + 11T_h$
方案 [23]	$9T_h$	$8T_h$	$17T_h$
本文方案	$4T_m + 8T_h$	$6T_m + T_a + 5T_h$	$10T_m + T_a + 13T_h$

表 5 运行时间对比

	方案 [14]	方案 [19]	方案 [20]	方案 [22]	方案 [23]	本文方案
运行时间 (ms)	0.046	26.9707	22.3613	22.3717	0.0391	22.3187

从表 4 可以看出, 本文方案和方案 [20, 22] 使用椭圆曲线点乘运算, 方案 [19] 使用模幂运算, 方案 [14, 23] 使用哈希运算. 表 5 显示, 基于哈希的方案 (即 [14, 23]) 拥有更低的计算成本. 在其他基于公钥密码的方案中 (即 [19, 20, 22]), 本文方案的计算成本最低.

7 结论

本文分析了 Andola 等人^[23]提出的认证协议, 并指出该协议的安全漏洞: 无法抵抗离线口令猜测攻击, 服务器欺骗攻击, 缺乏用户匿名性以及前向安全性. 本文利用椭圆曲线离散对数问题和计算性 Diffie - Hellman 问题, 提出一个改进方案, 克服了上述缺陷. 通过对比分析, 本文方案满足各类安全属性, 具有更好的安全性能, 其计算效率也比同类型的协议有所提升, 更加适用于多服务器环境.

参 考 文 献

- [1] Li L H, Lin L C, Hwang M S. A remote password authentication scheme for multiserver architecture using neural networks[J]. *IEEE Transactions on Neural Networks*, 2001, 12(6): 1498–1504.
- [2] Lin I C, Hwang M S, Li L H. A new remote user authentication scheme for multi-server architecture[J]. *Future Generation Computer Systems*, 2003, 19(1): 13–22.
- [3] Juang W S. An efficient and secure multi-server password authentication scheme using smart cards [J]. *IEEE Transactions on Consumer Electronics*, 2004, 50(1): 251–255.
- [4] Chang C C, Lee J S. An efficient and secure multi-server password authentication scheme using smart cards [C]. Tokyo: IEEE, 2004.
- [5] Tsai J L. Efficient multi-server authentication scheme based on one-way hash function without verification table [J]. *Computers & Security*, 2008, 27(3-4): 115–121.
- [6] Liao Y P, Wang S S. A secure dynamic ID based remote user authentication scheme for multi-server environment[J]. *Computer Standards & Interfaces*, 2009, 31(1): 24–29.
- [7] Hsiang H, Shih W. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment[J]. *Computer Standards & Interfaces*, 2009, 31(6): 1118–1123.
- [8] Sood S K, Sarje A K, Singh K. A secure dynamic identity based authentication protocol for multi-server architecture [J]. *Journal of Network and Computer Applications*, 2011, 34(2): 609–618.
- [9] Lee C C, Lin T H, Chang R X. A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards [J]. *Expert Systems with Applications*, 2011, 38(11): 13863–13870.
- [10] Tsaur W J, Li J H, Lee W B. An efficient and secure multi-server authentication scheme with key agreement [J]. *Journal of Systems and Software*, 2012, 85(4): 876–882.
- [11] Li Xiong, Ma Jian, Wang Wendong, et al. A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments[J]. *Mathematical and Computer Modelling*, 2013, 58(1-2): 85–95.
- [12] Chen C T, Lee C C. A two-factor authentication scheme with anonymity for multi-server environments[J]. *Security and Communication Networks*, 2015, 8(8): 1608–1625.
- [13] Shunmuganathan S, Saravanan R D, Palanichamy Y. Secure and efficient smartcard-based remote user authentication scheme for multiserver environment[J]. *Canadian Journal of Electrical and Computer Engineering*, 2015, 38(1): 20–30.

- [14] Janglrala S, Mukhopadhyay S, Das A K. A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards [J]. *Wireless Personal Communications*, 2017, 95(3): 2735–2767.
- [15] Sahoo S S, Mohanty S, Majhi B. An improved and secure two-factor dynamic ID based authenticated key agreement scheme for multiserver environment[J]. *Wireless Personal Communications*, 2018, 101(3): 1307–1333.
- [16] Wang Bin, Ma Maode. A smart card based efficient and secured multi-server authentication scheme[J]. *Wireless Personal Communications*, 2013, 68(2): 361–378.
- [17] Pippal R S, Jaidhar C, Tapaswi S. Robust smart card authentication scheme for multi-server architecture [J]. *Wireless Personal Communications*, 2013, 72(1): 729–745.
- [18] Truong T T, Tran M T, Duong A D, et al. Provable identity based user authentication scheme on ECC in multi-server environment [J]. *Wireless Personal Communications*, 2017, 95(3): 2785–2801.
- [19] Kumari S, Om H. Cryptanalysis and improvement of an anonymous multi-server authenticated key agreement scheme [J]. *Wireless Personal Communications*, 2017, 96(2): 2513–2537.
- [20] Irshad A, Sher M, Chaudhry S A, et al. A secure mutual authenticated key agreement of user with multiple servers for critical systems [J]. *Multimedia Tools and Applications*, 2018, 77(9): 11067–11099.
- [21] Ying B, Nayak A. Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography [J]. *Journal of Network and Computer Applications*, 2019, 131: 66–74.
- [22] Ul Haq I, Wang J, Zhu Y. Secure two-factor lightweight authentication protocol using selfcertified public key cryptography for multi-server 5G networks[J]. *Journal of Network and Computer Applications*, 2020, 161: 102660.
- [23] Andola N, Prakash S, Gahlot R, et al. An enhanced smart card and dynamic ID based remote multi-server user authentication scheme [J]. *Cluster Computing*, 2022, 25(5): 3699–3717.
- [24] Koblitz N. Elliptic curve cryptosystems[J]. *Mathematics of Computation*, 1987, 48(177): 203–209.
- [25] Wang Ding, Wang Ping. 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.
- [26] Burrows M, Abadi M, Needham R. A logic of authentication[J]. *ACM Transactions on Computer Systems (TOCS)*, 1990, 8(1): 18–36.
- [27] Kilinc H H, Yanik T. A survey of sip authentication and key agreement schemes[J]. *IEEE Communications Surveys & Tutorials*, 2013, 16(2): 1005–1023.

AN ELLIPTIC CURVE CRYPTOGRAPHY BASED AUTHENTICATION SCHEME FOR MULTI-SERVER ENVIRONMENT

LIU Zi-yao¹, CHEN Jian-hua¹, WEI Yong-shuang²

(1. School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China)

*(2. The Fifth Electronic Research Institute of Ministry of Industry and Information Technology,
Guangzhou 510000, China)*

Abstract: In this paper, we study the authentication and key agreement protocol for multi-server environment. Through the analysis of the scheme of Andola et al, an improved authentication and key agreement protocol is proposed by using the elliptic curve cryptography and random number. Besides, the security of the improved scheme is proved by BAN logic and the informal security analysis. Experimental results show that the proposed protocol has better security and practicability.

Keywords: multi-server; elliptic curve cryptography; password; key agreement

2010 MR Subject Classification: 68P25