Advances in Applied Mathematics 应用数学进展, 2022, 11(12), 8550-8566
Published Online December 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.1112902

Hans汉斯

基于椭圆曲线的高效远程用户认证协议

周　鑫，文　康，翁柏森，吴奕霄，王圣宝*

杭州师范大学信息科学与技术学院，浙江　杭州

收稿日期：2022年11月7日；录用日期：2022年12月1日；发布日期：2022年12月9日

摘　要

大部分现有远程用户认证方案都存在效率不高的缺点，因此不适用于资源受限设备。鉴于此，我们提出一个新的高效的远程用户认证协议。该协议采用椭圆曲线密码技术，并且使用智能卡存储长期秘密数据。我们分别使用形式化验证工具ProVerif、BAN逻辑以及非形式化方法验证和分析协议的安全性。结果表明新协议能抵抗多种常见攻击。通过与现有相关协议进行比较，表明新协议在性能方面也具有优势。

关键词

认证协议，椭圆曲线，BAN逻辑，双因子

An Efficient Remote User Authentication Protocol Based on Elliptic Curve

Xin Zhou, Kang Wen, Bosen Weng, Yixiao Wu, Shengbao Wang*

School of Information Science and Technology, Hangzhou Normal University, Hangzhou Zhejiang

Received: Nov. 7th, 2022; accepted: Dec. 1st, 2022; published: Dec. 9th, 2022

Abstract

Most existing remote user authentication schemes suffer from inefficiencies and are therefore not suitable for resource-constrained devices. In view of this, we propose a new efficient remote user authentication protocol. The protocol uses elliptic curve cryptography and uses smart cards to store long-term secret data. We verify and analyze the security of the protocol using the formal verification tool ProVerif, BAN logic, and non-formal methods, respectively. The results show that the new protocol is resistant to a variety of common attacks. A comparison with existing related protocols shows that the new protocol also has performance advantages.

*通讯作者。

文章引用: 周鑫, 文康, 翁柏森, 吴奕霄, 王圣宝. 基于椭圆曲线的高效远程用户认证协议[J]. 应用数学进展, 2022, 11(12): 8550-8566. DOI: 10.12677/aam.2022.1112902

Keywords

Authentication Protocol, Elliptic Curves, BAN Logic, Two-Factor

1. 引言

如今，第五代(5G)无线网络已经成为新一代的泛在移动通信网络，吸引了全世界学术界和工业界的大量研究。5G 网络具有高速度、低时延、低功耗的特点，开启了万物互联的新时代。目前，在世界一些社区，已经实现了从消费到生产、从人到物的 5G 互联通信。同时，超高可靠性和低时延的通信也已应用于车联网等应用场景。此外，5G 还带动了各种在线服务行业，如远程办公、远程医疗、网络购物、在线直播、云游戏等，越来越多的人开始尝试和享受网络服务带来的便利。可以说，这种趋势在未来会更加明显。

在 5G 改变我们生活的同时，移动设备也因为其便利性而占据了我们的生活。移动设备可以不受地域限制，无论何时何地，都可以享受在线服务。图 1 显示了移动客户端－服务器通信模型。然而，这种便利性也对安全性提出了挑战，也就是说，攻击者可以轻易地窃听合法用户之间共享的信息。例如，如果数据在无线通信中被“嗅出”，或者安装了一些恶意软件，入侵者就可能获得非法访问权。为了在合法参与者之间安全地分享信息，需要一个安全的认证协议。

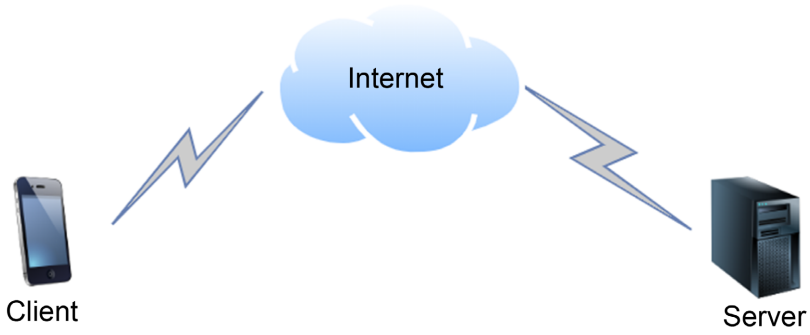


Figure 1. The mobile client-server communication model
图 1. 移动客户端－服务器通信模型

早期的认证协议主要是基于单一因素，也就是口令。1981 年，Lamport 等人[1]首次提出了一个使用单向散列函数将口令存储在服务器数据库中的认证协议。从那时起，许多研究者开始研究单因素认证协议。在此期间，许多基于口令的认证方案被提出[2]-[8]。

尽管基于口令的认证协议具有很高的效率，但研究人员后来意识到，仅仅基于口令的认证协议是不安全的。究其原因是因为口令很容易被破解，存储口令散列值的数据库也很容易被泄露。因此，在 2005 年，Fan 等人[9]提出了一种基于口令和智能卡的双因子认证协议。此后，大量的研究人员关注双因子认证方案，并提出了很多方案[10]-[16]。

之后，为了进一步提高方案的安全性，研究人员提出了大量基于口令、智能卡和生物识别信息的三

因子认证方案[17]-[23]。

然而，我们注意到，虽然大多数研究方案提高了安全性，但它们往往忽略了一个重要的问题，即在一些应用场景中的客户端设备，如资源密集型的物联网设备和移动设备，往往具有计算能力有限、内存空间小和电池寿命短的特点。一些基于双线性配对、三因子的方案往往需要高的计算和内存成本。我们认为，在移动环境中，我们应该在安全性和性能之间做出权衡。

2004 年，Das 等人[24]提出了一个使用智能卡的基于动态 ID 的远程用户认证方案。他们声称，他们的方案允许用户自由选择和改变他们的口令，而不需要维护任何验证表。然而，在 2009 年，Wang 等人[25]指出，文献[24]中方案对于其独立的口令来说是完全不安全的，而且未能实现相互认证，所以无法抵御冒充服务器的攻击。因此，他们在原有方案的基础上提出了一个改进的认证方案。此后不久，2011 年，Khan 等人[26]指出，文献[25]中的方案不能保证用户的匿名性，用户在自由选择口令时没有选择权，而且该方案没有对丢失或被盗的智能卡进行撤销的规定，因此他们提出了一个改进的认证方案，该方案改进了文献[25]的所有已发现的缺点。

2016 年，Xie 等人[27]提出了一个可证明安全的基于动态 ID 的匿名认证密钥交换协议。他们声称，他们的方案具有很高的安全性，可以抵御各种攻击。紧随其后，Li 等人[28]评论说，Xie 等人的方案[27]在智能卡丢失时存在离线字典攻击。此外，Abbasinezhad-Mood 等人[29]指出，文献[27]中的方案容易受到三种攻击，如密钥泄露伪装攻击、已知会话特定临时信息攻击、DOS 攻击。因此，他们提出了一个具有增强安全性的方案，该方案不仅保留了 Xie 等人的方案的优点，而且还提供了更好的执行时间。

不久之后，在 2019 年，Ying 等人[30]提出了一种新的 5G 网络远程用户认证协议，基于椭圆曲线密码学的自认证公钥密码系统。然而，文献[31]表明，Ying 等人的方案不能抵御身份猜测、口令猜测和用户冒充攻击。此外，它缺乏强大的用户匿名性和真正的双因子安全。

最近，有人提出了一些方案。在 2020 年，Kumari 等人[32]提出了一个使用智能卡的基于 ECC 的认证协议。该方案可以抵御各种攻击，并确保安全和隐私。2021 年，Tsobdjou 等人[33]提出了一个客户端–服务器环境下的轻量级相互认证协议。该协议具有匿名性和不可链接性，可以更好地保证用户的隐私。然而，该方案的缺点是有一个口令验证表，这使得它容易受到攻击。

虽然研究人员提出了多种基于不同密码系统的认证协议，但其中一些仍然存在安全漏洞，或者不适合资源受限设备。我们认为，为解决这个问题而提出的认证协议应该满足以下要求。

1) 拟定的协议应该是相对轻量级的，因为移动设备的资源是有限的。

2) 应该保证匿名性和不可链接性，以保护用户的隐私。

3) 为了确保高安全性，防止敌手通过从截获的信息破坏协议的安全性。

本文主要有以下贡献。

1) 我们提出了一种新的基于椭圆曲线的双因子认证密钥交换协议。

2) 相互认证是通过不安全的公共通道在用户和服务器之间实现的。

3) 我们提出的方案具有匿名性、不可链接性、无密码验证表和完美的前向安全性等安全特征，可以抵御离线密码猜测攻击、不同步攻击、密钥泄露冒名攻击、已知会话特定临时信息攻击和重放攻击。

4) 我们用 BAN 逻辑正式分析并证明了该增强方案的安全性。

5) 我们将我们的方案与其他相关方案进行了比较，并在安全和性能两方面找到了平衡点。

本文的其余部分组织如下。第 2 节将介绍拟议方案中使用的密码学基础知识。第 3 节介绍我们新的基于椭圆曲线的动态匿名相互认证密钥交换协议，而第 4 节将分析其安全性。第 5 节分析了所提协议的安全功能和性能。最后，我们给出了本文的结论。

2. 基础知识

2.1. Hash 函数

单向哈希函数 $\mathcal{H}:\{0,1\}^* \to \{0,1\}^l$ 的特性是接受任意长度的输入字符串 $k \in \{0,1\}^*$ 并生成确定长度的输出字符串 l。我们认为一个安全的哈希函数应该满足不可逆转性和弱/强抗碰撞性，其定义如下。

1) 给定一个哈希值 Γ，很难找到一个输入 γ，使 $\Gamma = \mathcal{H}(\gamma)$。

2) 给定一个 γ_1，计算 γ_2 满足 $\gamma_1 \neq \gamma_2$，$\mathcal{H}(\gamma_1) = \mathcal{H}(\gamma_2)$ 在计算上是不可行的。

3) 很难找到两个不同的消息 γ_1, γ_2，使得 $\mathcal{H}(\gamma_1) = \mathcal{H}(\gamma_2)$。

$\mathcal{H}(\bullet)$ 的抗碰撞性可以定义为：

定义一：抗碰撞单向哈希函数。敌手 $\mathcal{A}$ 可以找到两个消息 $\gamma_1 \neq \gamma_2$，使 $\mathcal{H}(\gamma_1) = \mathcal{H}(\gamma_2)$ 的优势被描述如下。

$$Adv_{\mathcal{A}}^{HASH}(t) = Prb\left[(\gamma_1, \gamma_2) \leftarrow \mathcal{A} : \gamma_1 \neq \gamma_2, \mathcal{H}(\gamma_1) = \mathcal{H}(\gamma_2)\right]$$

其中，敌手被允许随机选择两个消息 γ_1, γ_2，并以多项式时间 t 计算对随机值的优势概率。如果 $Adv_{\mathcal{A}}^{HASH}(t) \leq \epsilon$，而 $\epsilon \geq 0$ 是一个足够小的量，那么哈希函数 $\mathcal{H}(\cdot)$ 被称为抗碰撞。

2.2. 椭圆曲线加密

与 RSA 等其他传统密码系统相比，椭圆曲线密码系统(Elliptic Curve Cryptography, ECC)可以用较小的密钥提供同样的安全水平，因此，本文将充分利用这一优势。ECC 方程为 $E_p(a,b): y^2 = x^3 + ax + b$，它是在一个有限域 $\mathbb{Z}_p$ 上定义的，其中 p 是一个选定的巨大素数，$a, b \in \mathbb{Z}_p^*$ [34] [35]。此外，非星形椭圆曲线必须满足 $4a^3 + 27b^2 \bmod p \neq 0$。在 ECC 中，标量乘法是通过重复加法获得的。例如，让 G 是椭圆曲线 E_p 上的一个基点，其阶数为 n，则 $n \cdot G = G + G + \cdots + G$ (n 次)。

定义二：椭圆曲线离散对数问题(ECDLP)。给出 $E_p(a,b)$ 中的两个任意点 $G, I \in E_p(a,b)$，计算出满足 $I = n \cdot G$ 的标量 n。敌手 $\mathcal{A}$ 在多项式时间 t 内计算出 n 的概率定义为。

$$Adv_{\mathcal{A}}^{ECDLP}(t) = Prb\left[\mathcal{A}(G, I) = n : n \in \mathbb{Z}_p^*\right]$$

ECDLP 假设的结论是：$Adv_{\mathcal{A}}^{ECDLP}(t) \leq \epsilon$ 而 $\epsilon \geq 0$ 是一个足够微小的量。

定义三：椭圆曲线 Diffie-Hellman 问题(ECDHP)。给定两个点 $a \cdot P$ 和 $b \cdot P$，其中 P 是一个基点，$a, b \in \mathbb{Z}_n^*$，计算点 $ab \cdot P$ 在计算上是不可行的。敌手 $\mathcal{A}$ 在多项式时间 t 内解决 ECDHP 的概率被描述为。

$$Adv_{\mathcal{A}}^{ECDHP}(t) = Prb\left[\mathcal{A}(a \cdot P, b \cdot P) = ab \cdot P : a, b \in \mathbb{Z}_n^*\right]$$

ECDHP 假设的结论是：$Adv_{\mathcal{A}}^{ECDHP}(t) \leq \epsilon$ 而 $\epsilon \geq 0$ 是一个足够微小的量。

2.3. 符号说明

一些常用的符号显示在表 1 中。

2.4. 威胁模型

在本节中，我们定义了敌手的能力。在这里，我们采用 Dolev-Yao 的威胁模型[36]。根据这个模型，协议所使用的加密原语都是安全的。敌手可以在公开信道监听、拦截、修改、删除、存储和重放任何公开信息。

此外，敌手有能力在短时间内窃取合法用户的智能卡，并通过侧信道攻击提取智能卡中的信息[37]。

除此之外，敌手还可以通过从智能卡中获得的信息或从公开渠道截获的信息发起离线身份/密码猜测攻击，但敌手不可能在多项式时间内同时猜出用户的身份和密码。

Table 1. Symbol description
表 1. 符号说明

符号	描述
U_i	第 i 个合法用户
ID_i	用户身份信息
PW_i	用户口令
S	服务器
sk_S	服务器私钥
SC_i	用户智能卡
$SK_{U_i}^S$	会话密钥
$\oplus$	异或操作
$\|\|$	连接操作
$h_i(\cdot)$	不同输出的 Hash 函数

3. 所提方案

在这一节中，我们提出了一个新的动态匿名认证密钥交换方案。我们的方案由四个阶段组成，即初始化阶段、注册阶段、登录和认证阶段以及口令更改阶段。我们 $h_1:\{0,1\}^* \to \mathbb{Z}_p^*$ 的主要思路如下。a) 对用户的身份 ID 进行加密，以确保方案中的匿名性；b) 将用户的加密 ID 隐藏在信息中，从而实现动态性，达到不可链接性；c) 在方案中，智能卡不需要在本地验证存储在智能卡中的信息，从而防止离线字典猜测攻击。

3.1. 初始化阶段

在这个阶段，服务器 S 生成系统的公共参数以及它自己的长期私钥。S 在椭圆曲线 $E_p(a,b)$ 上选择一个由方程 $y^3 = x^2 + ax + b$ 定义的有限域 $\mathbb{Z}_p$ 上的大素数 p。然后，服务器在 $E_p(a,b)$ 上选择一个点 P，在一个有限域 $\mathbb{Z}_p$ 上选择一个单向哈希函数，$h_2:\{0,1\}^* \to \{0,1\}^l$，$h_3:\{0,1\}^* \to \{0,1\}^m$。$S$ 还选择了一个整数 sk_S 作为长期秘钥，$pk_S = sk_S \cdot P$ 是相应的公钥。最后，服务器发布 $\{E_p(a,b), p, P, h(\cdot)\}$ 作为系统的公共参数。

3.2. 注册阶段

在这个阶段，通过一个安全通道，用户 U 通过执行以下步骤与服务器 S 注册。

1) 请求注册。一个用户 U_i 首先选择自己的身份 ID_i，密码 PW_i 和一个随机数 $a_i \in \mathbb{Z}_q^*$。然后，U_i 计算 $RP_i = h_1(ID_i \| PW_i \| a_i)$ 并将他的注册请求 $\{ID_i, RP_i\}$ 发送给服务器。

2) 初始化智能卡。首先，服务器 S 通过查询注册表来检查用户 U_i 的身份 ID_i 是否唯一。如果新注册用户的 ID_i 与已注册用户的 ID 相同，服务器将要求用户选择另一个。第二，服务器将选择两个随机数 $b_i, c_i \in \mathbb{Z}_q^*$，然后通过使用 U_i 的请求信息和他的长期秘密值计算 $\alpha_i = h_1(ID_i \| sk_S \| b_i)$ 和 $\beta_i = RP_i \oplus \alpha_i$。第

三，服务器通过计算 $DID_i = E_{sk_S}\left(ID_i \| c_i\right)$ 加密 ID_i。最后，S 将 $\{ID_i, \alpha_i\}$ 存入其数据库，并将 $\{\beta_i, DID_i\}$ 写入智能卡 SC_i 中。然后服务器通过安全通道发放智能卡。

3) 确认智能卡。在收到智能卡 SC_i 后，用户 U_i 首先计算 $\gamma_i = a_i \oplus h_1\left(ID_i \| PW_i\right)$ 并将其加入 SC_i 中。最后，我们可以看到，诸如 $\{\beta_i, \gamma_i, DID_i\}$ 这样的数据被保存在智能卡中。

详细的注册阶段在图 2 中描述。

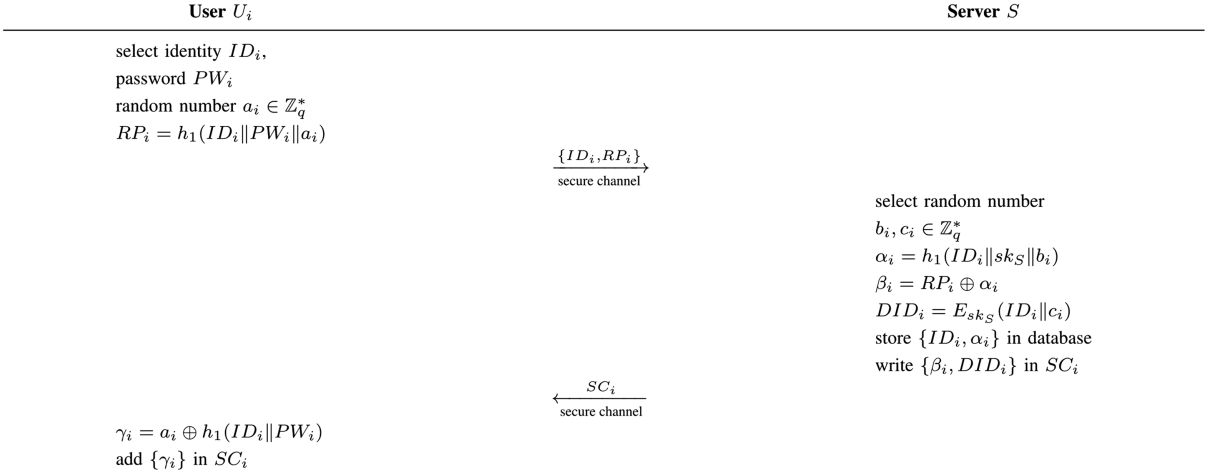


Figure 2. Registration phase
图 2. 认证阶段

3.3. 登录和认证阶段

在本节中，我们将详细描述用户的登录和与服务器的认证。图 3 说明了这个阶段。

1) 发送登录请求。用户 U_i 插入他的智能卡 SC_i 并输入他的身份 ID_i 和密码 PW_i。SC_i 计算 $a_i = \gamma \oplus h_1\left(ID_i \| PW_i\right)$，$RP_i = h_1\left(ID_i \| PW_i \| a_i\right), \alpha_i = \beta_i \oplus RP_i$。注意，为了防止离线密码猜测攻击，我们取消了智能卡的认证，由智能卡计算出的值将直接用于服务器认证。之后，SC 选择一个随机数 $r_i \in \mathbb{Z}_q^*$ 并计算出 Q_i, M_i, τ_i，即 $Q_i = h_1\left(r_i \| a_i\right) \cdot P$。$M_i = Q_i + h_1\left(\alpha_i \| ID_i\right) \cdot P$，$\tau_i = h_2\left(ID_i \| Q_i \| M_i \| \alpha_i\right)$。最后，$SC$ 发送登录请求 $\left\langle DID_r, M_i, \tau_i\right\rangle$ 到服务器 S。

2) 对用户进行认证。当服务器收到用户的登录请求时，服务器首先解密 DID_i，得到用户的 ID_i'，然后通过 ID_i' 搜索数据库，得到 α_i。之后，服务器计算 $Q_i' = M_i - h_1\left(\alpha_i \| ID_i'\right) \cdot P$ 并验证 $\tau_i' = h_2\left(ID_i' \| Q_i' \| M_i \| \alpha_i\right)$ 是否与收到的 τ_i 相等。如果它们不相等，认证阶段中断。否则，服务器继续执行密钥协议阶段。

3) 生成共享密钥。服务器 S 选择一个随机数 $r_S \in \mathbb{Z}_q^*$ 并继续计算 $Q_S = h_1\left(r_S \| sk_S\right) \cdot P$ 和 $M_S = Q_S + Q_i'$。然后，S 将实现共享密钥 $SK_{U_i}^S$ 为 $SK_{U_i}^S = h_2\left(h_1\left(r_S \| sk_S\right) \cdot Q_i' \| ID_i' \| \alpha_i\right)$。

4) 更新用户的身份。服务器 S 选择一个随机的 $c_i^{new} \in \mathbb{Z}_q^*$，像注册阶段一样重新计算 DID_i^{new}，其中 $DID_i^{new} = E_{sk_S}\left(ID_i' \| c_i^{new}\right)$。混淆 DID_i^{new} 为 $ODID_i = h_3\left(SK_{U_i}^S\right) \oplus DID_i^{new}$ 并计算验证人 $\tau_S = h_2\left(ID_i' \| \alpha_i \| Q_S \| M_S \| DID_i^{new}\right)$。最后，$S$ 向 U_i 发送响应信息 $\left\langle ODID_i, M_S, \tau_S\right\rangle$。

5) 确认共享密钥和更新身份。在收到服务器 S 的响应信息后，用户 U_i 首先通过 M_S 和 Q_i 恢复 $Q_S' = M_S - Q_i$，然后确认共享密钥 $SK_{U_i}^S = h_2\left(h_1\left(r_i \| a_i\right) \cdot Q_S' \| ID_i \| \alpha_i\right)$。其次，$U_i$ 通过计算 $DID_i^{new} = ODID_i \oplus h_3\left(SK_{U_i}^S\right)$ 得到新的身份，并用 DID_i^{new} 替代 DID_i。最后，U_i 计算 $\tau_S' = h_2\left(ID_i \| \alpha_i \| Q_S' \| M_S \| DID_i^{new}\right)$ 并验证它是否与收到的 τ_S 相等。如果它们相等，共享密钥协商成功；否则，用户放弃，协商中断。

3.4. 口令修改阶段

在修改口令之前，用户需要输入旧密码并进行上述的相互认证，然后服务器允许用户修改密码。U_i 选择一个新的密码 PW_i^{new} 和一个新的随机 $a_i^{new} \in \mathbb{Z}_q^*$，然后计算 $RP_i^{new} = h_1\left(ID_i \| PW_i^{new} \| a_i^{new}\right)$。$\beta_i^{new} = \beta_i \oplus RP_i \oplus RP_i^{new}$ 和 $\gamma_i^{new} = a_i^{new} \oplus h_1\left(ID_i \| PW_i^{new}\right)$。最后，它用新生成的替换 $\{\beta_i, \gamma_i\}$。

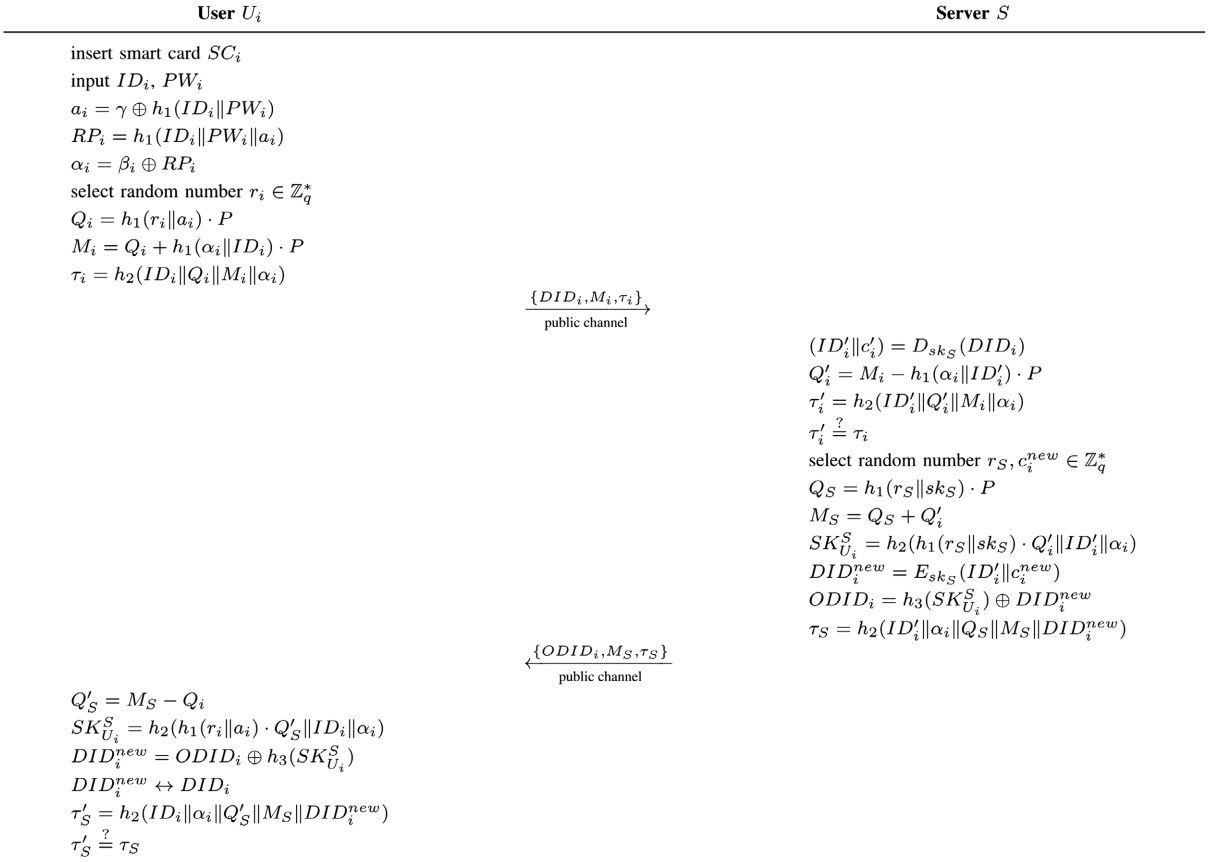


Figure 3. Authentication and key exchange phase
图 3. 认证及密钥交换阶段

4. 安全性分析

在这一节中，我们用非正式的安全分析来证明我们的方案可以抵御各种攻击，如密钥泄露伪装攻击、已知会话特定临时信息攻击和离线口令猜测攻击。同时，我们的方案还具有其他安全属性，如动态匿名性和前向安全性。然后，我们使用 Burrows-Abadi-Needham 逻辑[38]来证明所提方案的相互认证。我们还用强大的工具 ProVerif [39]正式分析了所提协议的安全性。

4.1. 非形式化分析

我们非正式地讨论了我们的方案的能力、安全性以及几个理想的属性，它基于这样的假设：恶意的敌手可以窃听、修改、插入或删除通过公共信道传输的任何信息。

4.1.1. 匿名性和不可链接性

我们的目标是让敌手无法判断多个会话是否由同一用户发送。需要注意的是，用户的 ID_i 受到服务器的私钥 sk_S 的良好保护。在传输过程中，敌手 $\mathcal{A}$ 无法通过开放信道上传输的信息获得用户的 ID。而且，

如果不知道服务器的私钥，他也无法解密 DID_i。另一方面，在两个不同的会话中，所有的信息，由 $\{r_i, r_S, c_i\}$ 随机化，都是新鲜和不同的。因此，我们的协议为用户提供了匿名性和不可链接性。

4.1.2. 离线密码猜测攻击

假设一个敌手 $\mathcal{A}$ 破坏了一张智能卡并获得了存储在其中的参数 β_i, γ_i, DID_i。敌手的目标是离线猜出正确的 (ID_i, PW_i) 对。在我们的方案中，我们取消了智能卡离线验证用户的 (ID_i, PW_i) 对的阶段。用户通过计算 $a_i = \gamma_i \oplus h_1(ID_i \| PW_i)$ 得到 a_i，它将参与后续计算并发送给服务器。然而，敌手不能在线进行多次猜测。简而言之，如果敌手想得到 a_i，他需要先猜出 ID_i 和 PW_i，但如果他想验证 ID_i 和 PW_i，他必须得到 a_i 的知识。这对任何敌手来说都是不可能的。此外，在我们的协议中，ID_i 和 PW_i 首先被连接起来，然后再应用哈希函数，这也增加了猜测的难度。另一方面，敌手不能获得 sk_S，所以不可能通过解密 DID_i 获得 ID_i。

4.1.3. 去同步化攻击

在我们的方案中，不需要增加额外的同步机制来维持用户和服务器之间一次性 DID_i 和 DID_i^{new} 的一致性。请注意，我们的 DID 和相应的更新只存储在用户端，服务器不保存相关的值，只对其进行解密。即使认证过程意外中断，用户仍然可以在随后的会话中通过最新的 DID_i 与服务器互动。

4.1.4. 密钥泄露伪装攻击

假设敌手 $\mathcal{A}$ 得到了服务器 sk_S 的长期秘钥，并且还窃听了公共频道。$\mathcal{A}$ 将使用 sk_S 来获得 ID_i，其中 $ID_i = D_{sk_S}(DID_i)$。然而，$\mathcal{A}$ 不能获得 α_i，所以 $\mathcal{A}$ 不可能通过 M_i 检索到 Q_i。显然，他也不可能生成 $SK_{U_i}^S = h_2(h_1(r_S \| sk_S) \cdot Q_i' \| ID_i' \| \alpha_i)$。

4.1.5. 已知特定会话临时信息攻击

假设敌手在某个会话中得到随机秘密值 r_i 或 r_S，他仍然不能计算出 $Q_i = h_1(r_i \| a_i) \cdot P$ 或 $Q_S = h_1(r_S \| sk_S) \cdot P$，因为他对 a_i 和 sk_S 一无所知。因此，$\mathcal{A}$ 也不能通过认证和恢复会话密钥。因此，我们的协议可以抵御已知会话特定的临时信息攻击。

4.1.6. 已知密钥安全

在我们的方案中，共享密钥 $SK_{U_i}^S = h_2(h_1(r_S \| sk_S) \cdot h_1(r_i \| a_i) \cdot P \| ID_i \| \alpha_i)$ 取决于随机值 $\{r_i, r_S\}$，在任何会话中生成其中的任何一个是不相关的。此外，这些随机值不在公开信道中传输，敌手无法从已知信息中恢复它们。因此，获得一个会话的共享密钥对计算其他会话的共享密钥没有好处。

4.1.7. 完美前向安全性

由于我们协议中的会话密钥是 $SK_{U_i}^S = h_2(h_1(r_S \| sk_S) \cdot h_1(r_i \| a_i) \cdot P \| ID_i \| \alpha_i)$，它包含一个取决于随机值 $\{r_i, r_S\}$ 的 Diffie-Hellman 实例。任何敌手计算 $h_1(r_S \| sk_S) \cdot h_1(r_i \| a_i) \cdot P$ 的概率可以忽略不计，只知道 $Q_i = h_1(r_i \| a_i) \cdot P$ 和 $Q_S = h_1(r_S \| sk_S) \cdot P$，因为 Diffie-Hellman 问题的难解。因此，我们的方案可以提供完美的前向保密性。

4.1.8. 重放攻击

在我们的方案中，我们使用随机值 $\{r_i, r_S\}$ 来抵御重放攻击。即使敌手将信息 $\{DID_i, M_i, \tau_i\}$ 复制到服务器上。他将无法计算会话密钥 $SK_{U_i}^S = h_2(h_1(r_S \| sk_S) \cdot h_1(r_i \| a_i) \cdot P \| ID_i \| \alpha_i)$，因为他对 $\{a_i, sk_S, \alpha_i\}$ 一无所知。

4.2. BAN 逻辑证明

BAN 逻辑在认证协议的形式化分析中发挥了广泛而积极的作用。在本节中，我们用 BAN 逻辑正式证明了用户和服务器之间的认证。BAN 逻辑中使用的一些符号如表 2 所示，常用规则如表 3 所示。

Table 2. BAN logic symbol description
表 2. BAN 逻辑符号说明

符号	含义
$P \mid\equiv X$	P 相信 X
$P \lhd X$	P 曾经收到包含 X 的消息
$P \mid\sim X$	P 曾经发送包含 X 的消息
$\#X$	X 是新鲜的
$P \mid\Rightarrow X$	P 对 X 有管辖权
$P \overset{K}{\longleftrightarrow} Q$	P 和 Q 之间共享密钥 K
$P \overset{X}{\rightleftharpoons} Q$	P 和 Q 之间共享秘密 X
$\langle X \rangle_K$	使用 K 加密 X

Table 3. BAN logic rule description
表 3. BAN 逻辑规则说明

标记	含义	公式
R_1	消息含义规则	$\dfrac{P \mid\equiv Q \overset{K}{\longleftrightarrow} P, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$
R_2	管辖权规则	$\dfrac{P \mid\equiv Q \mid\Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$
R_3	Nonce 验证规则	$\dfrac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$
R_4	新鲜度规则	$\dfrac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)}$
R_5	信念规则	$\dfrac{P \mid\equiv (X), P \mid\equiv (Y)}{P \mid\equiv (X,Y)}$, $\dfrac{P \mid\equiv Q \mid\equiv (X,Y)}{P \mid\equiv Q \mid\equiv X}$

BAN 逻辑分析的理想前提为：

消息 1： $U_i \to S : \left\langle \{ID_i\}_{sk_S}, r_i, Q_i \right\rangle_{U_i \overset{\alpha_i}{\leftrightarrow} S}$

消息 2： $S \to U_i : \left\langle \{ID_i\}_{sk_S}, q_S, Q_S \right\rangle_{U_i \overset{\alpha_i}{\leftrightarrow} S}$

需要证明的安全目标为：

GOAL1： $U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S$

GOAL2： $U_i \mid\equiv S \mid\equiv U_i \overset{SK}{\leftrightarrow} S$

GOAL3： $S \mid\equiv U_i \overset{SK}{\leftrightarrow} S$

GOAL4： $S \mid\equiv U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S$

新协议的初始化假定为：

A1： $S \mid\equiv \#(r_i)$

A2： $U_i \mid\equiv \#(r_S)$

A3： $U_i \mid\equiv U_i \overset{\alpha_i}{\rightleftharpoons} S$

A4： $S \mid\equiv U_i \overset{\alpha_i}{\rightleftharpoons} S$

A5： $U_i \mid\equiv S \mid\Rightarrow U_i \overset{SK}{\rightleftharpoons} S$

证明过程如下：

S1：由消息 1 可得到 $S \lhd \left\{\{ID_i\}_{sk_S}, r_i, Q_i, M_1, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}_{U_i \overset{\alpha_i}{\rightleftharpoons} S}$

S2：根据 A4，S1 和 R1，可以得到 $S \mid\equiv U_i \mid\sim \left\{\{ID_i\}_{sk_S}, r_i, Q_i, M_1, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}$

S3：根据 S2，A1 和 R4，得到 $S \mid\equiv U_i \mid\equiv \left\{\{ID_i\}_{sk_S}, r_i, Q_i, M_1, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}$

S4：根据 $SK_{U_i}^S = h\left(h\left(r_S \parallel sk_S\right) \cdot Q_i \parallel ID_i \parallel \alpha_i\right)$ 可以得到 $S \mid\equiv U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S$ **(GOAL4)**

S5：根据 S3，A1 和 R3，我们可以得到 $S \mid\equiv \left\{r_i, Q_i, \{ID_i\}_{sk_S}, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}$

S6：进而可以得到 $S \mid\equiv U_i \overset{SK}{\leftrightarrow} S$ **(GOAL3)**

S7：根据消息 2，可以得到 $U_i \lhd \left\{\{ID_i\}_{SK}, q_S, Q_S, M_2, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}_{U_i \overset{\alpha_i}{\rightleftharpoons} S}$

S8：根据 S7，A4 和 R1，可以得到 $U_i \mid\equiv S \mid\sim \left\{\{ID_i\}_{SK}, q_S, Q_S, M_2, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}$

S9：根据 S8，A1 和 R4，得到 $U_i \mid\equiv S \mid\equiv \left\{\{ID_i\}_{SK}, q_S, Q_S, M_2, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}$

S10：由会话密钥的计算公式可得 $U_i \mid\equiv S \mid\equiv U_i \overset{SK}{\leftrightarrow} S$ **(GOAL2)**

S11：根据 S9，A1 和 R3，可以得到 $U_i \mid\equiv \left\{r_S, Q_S, \{ID_i\}_{sk_S}, U_i \overset{\alpha_i}{\rightleftharpoons} S\right\}$

S12：因此，可以推出 $U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S$ **(GOAL1)**

上述证明表明我们的方案能确保 U_i 和 S 之间的相互认证。

4.3. ProVerif 形式化验证

在这一节中，我们使用 ProVerif，一个基于 pi-calculus 的自动验证工具[40]，来模拟拟议协议的注册阶段、登录和认证阶段。我们参考了 Abbasinezhad-Mood 等人的[29]分析模型，源代码和验证结果如图 4 所示。结果表明，我们提出的协议可以提供生成的会话密钥的安全性，用户的匿名性，并能抵抗离线口令猜测攻击、重放攻击、伪装攻击和篡改攻击。

```
(* Dynamic ID-Based Authenticated Key Agreement Scheme *)
(* Channel Definition *)
free SCH:channel[private]. (* Secure channel between server and user *)
free CH:channel. (* Public channel between server and user *)

(*Private Terms*)
free IDi:bitstring [private]. (* User's identifier *)
free PWi:bitstring [private]. (* User's password *)
free SK: bitstring [private]. (* Shared session key *)
free sks:bitstring [private]. (* Secret key of server *)

(* Public Terms *)
const P:bitstring. (* Base point *)

(* Functions *)
fun CON(bitstring, bitstring):bitstring. (* Concatenation *)
fun SymEnc(bitstring, bitstring):bitstring. (* Symmetric encryption *)
fun HOne(bitstring):bitstring. (* One-input Hash *)
fun HTwo(bitstring, bitstring):bitstring. (* Two-input Hash *)
fun HThree(bitstring, bitstring, bitstring):bitstring. (* Three-input Hash *)
fun HFour(bitstring, bitstring, bitstring, bitstring):bitstring. (* Four-input Hash *)
fun HFive(bitstring, bitstring, bitstring, bitstring, bitstring):bitstring. (* Five-input Hash *)
fun XOR(bitstring, bitstring):bitstring. (* Exclusive OR *)
fun Mul(bitstring, bitstring):bitstring. (* EC point multiplication *)
fun SEncrypt(bitstring, bitstring): bitstring. (* Auxiliary *)
fun ADD(bitstring, bitstring):bitstring. (* Addition *)


(* Destructors *)
reduc forall m1:bitstring, key:bitstring; SymDec(SymEnc(m1, key), key) = m1.
reduc forall m1:bitstring, m2:bitstring; SeparateFirst(CON(m1, m2)) = m1.
reduc forall m1:bitstring, m2:bitstring; SeparateSecond(CON(m1, m2)) = m2.
reduc forall m1:bitstring, m2:bitstring; DXOR(XOR(m1, m2), m1) = m2.
reduc forall m3:bitstring, m4:bitstring; SubFun(ADD(m3,m4),m4) = m3.
reduc forall m5:bitstring, m6:bitstring; SDecrypt(SEncrypt(m5, m6), m6) = m5.

(* Equations *)
equation forall m1:bitstring, m2:bitstring, m3:bitstring, m4:bitstring; Mul(ADD(m1, m2),
ADD(Mul(m3, P), Mul(m4, P))) = Mul(ADD(m3, m4), ADD(Mul(m1, P), Mul(m2, P))).

(* queries *)
(* query attacker (new bs). *) (* Uncomment when checking KSSTIA resistance *)
(* query attacker (new ascr). *) (* Uncomment when checking KSSTIA resistance *)
query attacker (SK). (* A query to check the secrecy of generated session key *)
query attacker (IDi).(* A query to check the strong anonymity of user *)
weaksecret PWi. (* A query to check the resistance against the offline password guessing
attack *)
query m1 : bitstring, m2 : bitstring, m3: bitstring; inj-event(Server_Accepts_Key (m1, m2, m3))
==> inj-event (User_Initiates_Session (m1, m2, m3)).
query m1 : bitstring, m2 : bitstring, m3: bitstring; inj-event (User_Accepts_Key (m1, m2,
m3))==> inj-event (Server_Responds_User (m1, m2, m3)).


(* Events *)
event User_Initiates_Session (bitstring, bitstring, bitstring).
event Server_Accepts_Key (bitstring, bitstring, bitstring).
event Server_Responds_User (bitstring, bitstring, bitstring).
event User_Accepts_Key (bitstring, bitstring, bitstring).
```

```
(*Server Process*)
let pS=

(*Registration Phase*)
new bi:bitstring;
new ci:bitstring;
new rs:bitstring;
new ci_new:bitstring;
in(SCH,(IDi:bitstring,RPi:bitstring));
let Alpha=HThree(IDi,sks,bi) in
let Beta=XOR(RPi,Alpha) in
let temp0=CON(IDi,ci) in
let DIDi=SymEnc(temp0,sks) in
out(SCH,(Beta,DIDi));

(* Login and Authentication Phase *)
in(CH,(xDIDi:bitstring,xMi:bitstring,xTaui:bitstring));
let temp1=SymDec(xDIDi,sks) in
let xIDi=SeparateFirst(temp1) in
let xci=SeparateSecond(temp1) in
let xQi=SubFun(xMi,Mul(HTwo(Alpha,xIDi),P)) in
let xxTaui=HFour(xIDi,xQi,xMi,Alpha) in
if(xTaui=xxTaui) then
let Qs=Mul(HTwo(rs,sks),P) in
let Ms=ADD(Qs,xQi) in
let SKu_s=HThree(Mul(HTwo(rs,sks),xQi),xIDi,Alpha) in
out(CH, SEncrypt(SK, SKu_s));
let temp2=CON(xIDi,ci_new) in
let DIDi_new=SymEnc(temp2,sks) in
let ODIDi=XOR(HOne(SKu_s),DIDi_new) in
let Taus=HFive(xIDi,Alpha,Qs,Ms,DIDi_new) in
event Server_Responds_User(DIDi_new,Ms,Taus);
out(CH,(ODIDi,Ms,Taus));
event Server_Accepts_Key(DIDi,xMi,xTaui);
0.

(* User Process *)
let pU=

(* Registration Phase *)
new ai:bitstring;
new ri:bitstring;
let RPi=HThree(IDi,PWi,ai) in
out(SCH,(IDi,RPi));
in(SCH,(Beta:bitstring,DIDi:bitstring));
let Gamma=XOR(ai,HTwo(IDi,PWi)) in

(*   Login and Authentication Phase *)
let xai=XOR(Gamma,HTwo(IDi,PWi)) in
let xRPi=HThree(IDi,PWi,ai) in
let xAlpha=XOR(Beta,xRPi) in
let Qi=Mul(HTwo(ri,ai),P) in
let Mi=ADD(Qi,Mul(HTwo(xAlpha,IDi),P)) in
let Taui=HFour(IDi,Qi,Mi,xAlpha) in
event User_Initiates_Session(DIDi,Mi,Taui);
out(CH,(DIDi,Mi,Taui));
in(CH,(xODIDi:bitstring,xMs:bitstring,xTaus:bitstring));
let xQs=SubFun(xMs,Qi) in
let SKu_s=HThree(Mul(HTwo(ri,ai),xQs),IDi,xAlpha) in
out(CH,SEncrypt(SK,SKu_s));
let xDIDi_new=XOR(xODIDi,HOne(SKu_s)) in
let xxTaus=HFive(IDi,xAlpha,xQs,xMs,xDIDi_new) in
if(xxTaus=xTaus) then
event User_Accepts_Key(xDIDi_new,xMs,xTaus);
0.

process
((!pS) | (!pU)) (* Comment in case of the perfect forward secrecy check*)
```

```
--------------------------------------------------------------
Verification summary:

Query not attacker(SK[]) is true.

Query not attacker(IDi[]) is true.

Weak secret PWi is true.

Query inj-event(Server_Accepts_Key(m1,m2,m3)) ==> inj-event(User_Initiates_Session(m1,m2,m3)) is true.

Query inj-event(User_Accepts_Key(m1,m2,m3)) ==> inj-event(Server_Responds_User(m1,m2,m3)) is true.

--------------------------------------------------------------
```

Figure 4. Source code of Proverif and verification results
图 4. Proverif 工具源代码及验证结果

5. 功能和性能对比

在本节中，我们将从计算成本和通信成本两个方面评估所提出的方案的性能。请注意，初始化阶段、用户注册阶段和密码修改阶段是一次性的执行阶段，因此不在性能分析的范围之内。此外，我们考虑服务器使用具有高计算性能的服务器，而客户端使用具有一般计算性能的移动设备。

5.1. 计算开销

计算成本包括客户端和服务器执行各种操作所需的时间成本。在表 5 中，我们定义了一些需要使用的符号。需要注意的是，我们认为一些操作(如异或、连接操作等)的执行时间可以忽略不计，因此没有考虑在计算开销中。

我们提出的协议的计算成本在客户端为 $9T_h + 3T_{sm\_ecc} + 2T_{pa\_ecc}$，在服务器端为 $7T_h + T_{E/D} + 3T_{sm\_ecc} + 2T_{pa\_ecc}$。

5.2. 通信开销

为了分析通信成本，我们首先计算每个参数的大小(比特)。而计算出的身份和密码的大小为 64 比特；哈希(SHA-256)输出为 256 比特。随机生成的随机数为 64 比特，时间戳大小为 32 比特。而椭圆曲线的点的大小为 320 比特。因此，在我们提出的方案中，客户和服务器各发送一个消息，其中消息 1 由 $\langle DID_i, M_1, \tau_i \rangle$ 组成，消息 2 由 $\langle ODID_i, M_S, \tau_S \rangle$ 组成，所以总的通信开销是 $\left[(128+320+256)+(128+320+256)\right] = 1408$ bits。

5.3. 安全功能对比

我们将我们提出的方案与 Xie 等人[27]、Ying 等人[30]、Haq 等人[31]、Kumari 等人[32]以及 Tsobdjou 等人[33]的相关方案进行比较。这些安全性功能包括无密码验证表、口令友好、无时钟同步、相互认证、用户匿名、不可追踪性、前向保密性、抗重放攻击、抗伪装攻击。表 4 显示了比较结果，证实我们的方案同时实现了上述所有特性。

Table 4. Security functionalities comparision
表 4. 安全功能对比

	不可链接性	用户匿名性	前向安全性	口令友好	双向认证	抗重放攻击	无时钟同步	无口令验证表	抗伪装攻击
[27]	Y	Y	Y	N	Y	Y	N	Y	Y
[30]	N	Y	Y	Y	N	Y	Y	Y	Y
[31]	Y	Y	Y	Y	Y	Y	Y	Y	Y
[32]	N	N	Y	Y	Y	Y	N	Y	N
[33]	Y	Y	Y	N	Y	Y	Y	N	Y
新协议	Y	Y	Y	Y	Y	Y	Y	Y	Y

注：表中 Y 代表方案满足对应功能；N 代表方案不满足对应功能；N/A 表示该属性不适用于该方案。

5.4. 性能对比

为了证明我们提出的方案的效率，我们将我们的方案与 Xie 等人[27]、Ying 等人[30]、Haq 等人[31]、Kumari 等人[32]以及 Tsobdjou 等人[33]的相关方案在计算开销和通信开销进行了对比。

我们考虑采用 128 位密钥的对称加密算法的高级加密标准。我们使用方案[33]中定义的服务器和移动设备执行时间，其中服务器使用英特尔酷睿 i7-6700 3.40 GHz 处理器，16 GB 内存和 Windows 10 Enterprise 64 位操作系统，移动设备使用 Asus Z00D，搭配 Intel Atom Z2560 1.60 GHz 处理器，2 GB 内存和 Android 5.0.1 操作系统，并使用 Java 编程语言来编写各种操作。

表 5 显示了在服务器端和客户端不同操作的执行时间。表 6 和图 5(a)显示了计算成本的比较。表 7 和图 5(b)显示了通信成本的比较。

通过比较，我们可以得出结论，我们提出的方案在计算成本和通信成本方面优于文献[30] [31] [32]中方案。在计算成本方面，我们与文献[27] [33]中方案相当，但通过功能比较，我们可以得出结论，我们的方案在相同的计算成本下更安全。总的来说，我们在安全性和性能之间有一个权衡。

Table 5. Multiple cryptographic operations and execution times
表 5. 多种密码学操作及执行时间

操作	说明	执行时间	
		用户端	服务器端
T_h	执行 Hash 函数的时间	0.053	0.173
$T_{E/D}$	执行对称加解密算法时间	0.114	0.147
T_{sm_ecc}	执行椭圆曲线加法时间	350.493	32.478
T_{pa_ecc}	执行椭圆曲线乘法时间	1.039	0.731

Table 6. Computation cost comparison
表 6. 计算开销对比

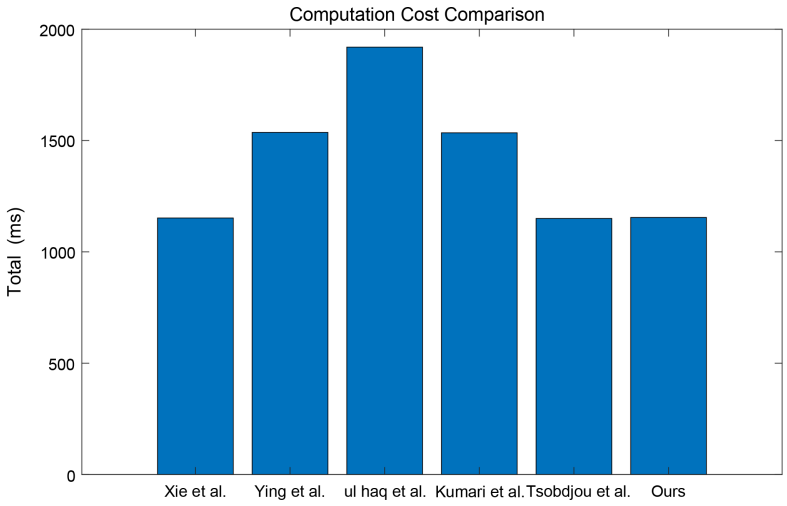
方案	计算开销		总计(ms)
	用户端	服务器端	
[27]	$6T_h + 3T_{sm\_ecc} + 1T_{pa\_ecc}$	$5T_h + 3T_{sm\_ecc} + 1T_{pa\_ecc} + 2T_{E/D}$	1152.16
[30]	$7T_h + 4T_{sm\_ecc} + 2T_{pa\_ecc}$	$3T_h + 4T_{sm\_ecc} + 2T_{pa\_ecc}$	1536.314
[31]	$7T_h + 5T_{sm\_ecc} + 2T_{pa\_ecc}$	$4T_h + 5T_{sm\_ecc} + 2T_{pa\_ecc}$	1919.458
[32]	$12T_h + 4T_{sm\_ecc} + 2T_{E/D}$	$9T_h + 4T_{sm\_ecc} + 2T_{E/D}$	1534.599
[33]	$5T_h + 3T_{sm\_ecc}$	$3T_h + 3T_{sm\_ecc} + 1T_{E/D}$	1149.844
新协议	$9T_h + 3T_{sm\_ecc} + 2T_{pa\_ecc}$	$7T_h + 3T_{sm\_ecc} + 2T_{pa\_ecc} + 2T_{E/D}$	1154.435

Table 7. Communication cost comparison
表 7. 通信开销对比

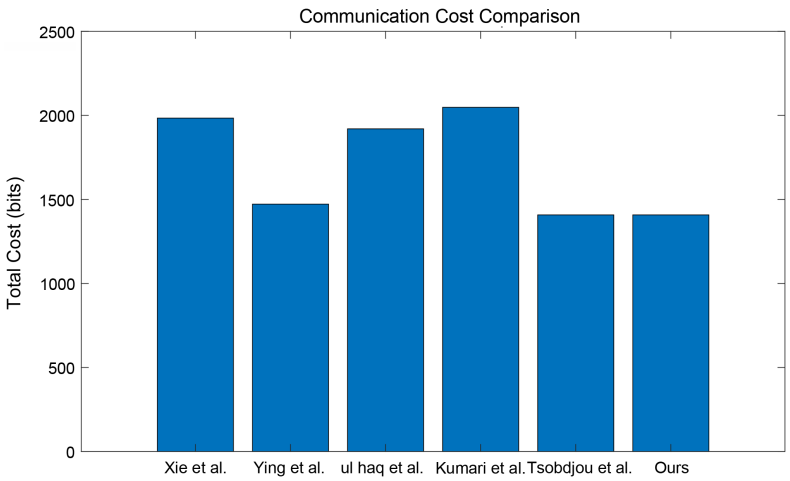
方案	消息数	总计(bits)
[27]	2	1984
[30]	3	1472
[31]	2	1920

Continued

[32]	2	2048
[33]	3	1408
新协议	2	1408



(a) 计算开销



(b) 通信开销

Figure 5. Performance comparision
图 5. 性能对比

6. 结论

在本文中，我们提出了一个新的远程用户认证协议。它解决了现有协议容易受到冒名攻击、重放攻击以及缺乏匿名性和不可追踪性的安全问题。非正式安全分析表明，新协议具有匿名性、不可链接性、已知密钥安全性和完美的前向保密性。并且，它还可以抵御离线密码攻击、去同步攻击、密钥泄露冒名攻击、已知会话特定临时信息攻击和回复攻击。此外，我们还使用 BAN 逻辑证明了该协议的相互认证性。最后，分析和比较结果表明，新协议具有较好的性能，它可以完全应用于计算能力有限的设备和服务器之间的相互认证。

基金项目

本研究得到了国家自然科学基金(基金号：U21A20466)的支持。

参考文献

[1] Lamport, L. (1981) Password Authentication with Insecure Communication. *Communications of the ACM*, **24**, 770-772. https://doi.org/10.1145/358790.358797

[2] Seo, D.H. and Sweeney, P. (1999) Simple Authenticated Key Agreement Algorithm. *Electronics Letters*, **35**, 1073-1074. https://doi.org/10.1049/el:19990724

[3] Hwang, M.S. and Li, L.H. (2000) A New Remote User Authentication Scheme Using Smart Cards. *IEEE Transactions on Consumer Electronics*, **46**, 28-30. https://doi.org/10.1109/30.826377

[4] Sun, H.M. (2000) An Efficient Remote Use Authentication Scheme Using Smart Cards. *IEEE Transactions on Consumer Electronics*, **46**, 958-961. https://doi.org/10.1109/30.920446

[5] Chien, H.Y., Jan, J.K. and Tseng, Y.M. (2002) An Efficient and Practical Solution to Remote Authentication: Smart Card. *Computers & Security*, **21**, 372-375. https://doi.org/10.1016/S0167-4048(02)00415-7

[6] Lee, S.W., Kim, W.H., Kim, H.S. and Yoo, K.Y. (2004) Efficient Password-Based Authenticated Key Agreement Protocol. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K. and Gervasi, O., Eds., *Computational Science and Its Applications—ICCSA* 2004, Lecture Notes in Computer Science, Vol. 3046, Springer, Berlin, 617-626. https://doi.org/10.1007/978-3-540-24768-5_66

[7] Pointcheval, D. (2012) Password-Based Authenticated Key Exchange. In: *International Workshop on Public Key Cryptography*, Springer, Berlin, 390-397. https://doi.org/10.1007/978-3-642-30057-8_23

[8] Farash, M.S. and Attari, M.A. (2014) An Efficient and Provably Secure Three-Party Password-Based Authenticated Key Exchange Protocol Based on Chebyshev Chaotic Maps. *Nonlinear Dynamics*, **77**, 399-411. https://doi.org/10.1007/s11071-014-1304-6

[9] Fan, C.I., Chan, Y.C. and Zhang, Z.K. (2005) Robust Remote Authentication Scheme with Smart Cards. *Computers & Security*, **24**, 619-628. https://doi.org/10.1016/j.cose.2005.03.006

[10] He, D., Kumar, N., Khan, M.K. and Lee, J.H. (2013) Anonymous Two-Factor Authentication for Consumer Roaming Service in Global Mobility Networks. *IEEE Transactions on Consumer Electronics*, **59**, 811-817. https://doi.org/10.1109/TCE.2013.6689693

[11] Huang, X., Chen, X., Li, J., Xiang, Y. and Xu, L. (2013) Further Observations on Smart-Card-Based Password-Authenticated Key Agreement in Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*, **25**, 1767-1775. https://doi.org/10.1109/TPDS.2013.230

[12] Chang, I.P., Lee, T.F., Lin, T.H. and Liu, C.M. (2015) Enhanced Two-Factor Authentication and Key Agreement Using Dynamic Identities in Wireless Sensor Networks. *Sensors*, **15**, 29841-29854. https://doi.org/10.3390/s151229767

[13] Xie, Q., Dong, N., Wong, D.S. and Hu, B. (2016) Cryptanalysis and Security Enhancement of a Robust Two-Factor Authentication and Key Agreement Protocol. *International Journal of Communication Systems*, **29**, 478-487. https://doi.org/10.1002/dac.2858

[14] Yang, Z., He, J., Tian, Y. and Zhou, J. (2019) Faster Authenticated Key Agreement with Perfect Forward Secrecy for Industrial Internet-of-Things. *IEEE Transactions on Industrial Informatics*, **16**, 6584-6596. https://doi.org/10.1109/TII.2019.2963328

[15] Li, W., Li, X., Gao, J. and Wang, H. (2019) Design of Secure Authenticated Key Management Protocol for Cloud Computing Environments. *IEEE Transactions on Dependable and Secure Computing*, **18**, 1276-1290. https://doi.org/10.1109/TDSC.2019.2909890

[16] Mo, J., Hu, Z. and Lin, Y. (2018) Remote User Authentication and Key Agreement for Mobile Client-Server Environments on Elliptic Curve Cryptography. *The Journal of Supercomputing*, **74**, 5927-5943. https://doi.org/10.1007/s11227-018-2507-2

[17] Srinivas, J., Das, A.K., Wazid, M. and Kumar, N. (2018) Anonymous Lightweight Chaotic Map-Based Authenticated Key Agreement Protocol for Industrial Internet of Things. *IEEE Transactions on Dependable and Secure Computing*, **17**, 1133-1146. https://doi.org/10.1109/TDSC.2018.2857811

[18] Zhang, L., Zhang, Y., Tang, S. and Luo, H. (2017) Privacy Protection for e-Health Systems by Means of Dynamic Authentication and Three-Factor Key Agreement. *IEEE Transactions on Industrial Electronics*, **65**, 2795-2805. https://doi.org/10.1109/TIE.2017.2739683

[19] Jiang, Q., Zhang, N., Ni, J., Ma, J., Ma, X. and Choo, K.K.R. (2020) Unified Biometric Privacy Preserving Three-

Factor Authentication and Key Agreement for Cloud-Assisted Autonomous Vehicles. *IEEE Transactions on Vehicular Technology*, **69**, 9390-9401. https://doi.org/10.1109/TVT.2020.2971254

[20] Sutrala, A.K., Obaidat, M.S., Saha, S., Das, A.K., Alazab, M. and Park, Y. (2021) Authenticated Key Agreement Scheme with User Anonymity and Untraceability for 5G-Enabled Softwarized Industrial Cyber-Physical Systems. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 2316-2330. https://doi.org/10.1109/TITS.2021.3056704

[21] Qiu, S., Wang, D., Xu, G. and Kumari, S. (2022) Practical and Provably Secure Three-Factor Authentication Protocol Based on Extended Chaotic-Maps for Mobile Lightweight Devices. *IEEE Transactions on Dependable and Secure Computing*, **19**, 1338-1351.

[22] Reddy, A.G., Das, A.K., Odelu, V., Ahmad, A. and Shin, J.S. (2019) A Privacy Preserving Three-Factor Authenticated Key Agreement Protocol for Client-Server Environment. *Journal of Ambient Intelligence and Humanized Computing*, **10**, 661-680. https://doi.org/10.1007/s12652-018-0716-4

[23] Mohit, P. (2021) An Efficient Mutual Authentication and Privacy Prevention Scheme for e-Healthcare Monitoring. *Journal of Information Security and Applications*, **63**, Article ID: 102992. https://doi.org/10.1016/j.jisa.2021.102992

[24] Das, M.L., Saxena, A. and Gulati, V.P. (2004) A Dynamic ID-Based Remote User Authentication Scheme. *IEEE Transactions on Consumer Electronics*, **50**, 629-631. https://doi.org/10.1109/TCE.2004.1309441

[25] Wang, Y.Y., Liu, J.Y., Xiao, F.X. and Dan, J. (2009) A More Efficient and Secure Dynamic ID-Based Remote User Authentication Scheme. *Computer Communications*, **32**, 583-585. https://doi.org/10.1016/j.comcom.2008.11.008

[26] Khan, M.K., Kim, S.K. andAlghathbar, K. (2011) Cryptanalysis and Security Enhancement of a "More Efficient & Secure Dynamic ID-Based Remote User Authentication Scheme". *Computer Communications*, **34**, 305-309. https://doi.org/10.1016/j.comcom.2010.02.011

[27] Xie, Q., Wong, D.S., Wang, G., Tan, X., Chen, K.F. and Fang, L. (2017) Provably Secure Dynamic ID-Based Anonymous Two-Factor Authenticated Key Exchange Protocol with Extended Security Model. *IEEE Transactions on Information Forensics and Security*, **12**, 1382-1392. https://doi.org/10.1109/TIFS.2017.2659640

[28] Li, X., Yang, D., Zeng, X., Chen, B. and Zhang, Y. (2018) Comments on "Provably Secure Dynamic ID-Based Anonymous Two-Factor Authenticated Key Exchange Protocol with Extended Security Model". *IEEE Transactions on Information Forensics and Security*, **14**, 3344-3345. https://doi.org/10.1109/TIFS.2018.2866304

[29] Abbasinezhad-Mood, D., Mazinani, S.M., Nikooghadam, M. and Sharif, A.O. (2020) Efficient Provably-Secure Dynamic ID-Based Authenticated Key Agreement Scheme with Enhanced Security Provision. *IEEE Transactions on Dependable and Secure Computing*, **19**, 1227-1238. https://doi.org/10.1109/TDSC.2020.3024654

[30] Ying, B. and Nayak, A. (2019) Lightweight Remote User Authentication Protocol for Multi-Server 5G Networks Using Self-Certified Public Key Cryptography. *Journal of Network and Computer Applications*, **131**, 66-74. https://doi.org/10.1016/j.jnca.2019.01.017

[31] Wang, J. and Zhu, Y. (2020) Secure Two-Factor Lightweight Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-Server 5G Networks. *Journal of Network and Computer Applications*, **161**, Article ID: 102660. https://doi.org/10.1016/j.jnca.2020.102660

[32] Kumari, A., Jangirala, S., Abbasi, M.Y., Kumar, V. and Alam, M. (2020) ESEAP: ECC Based Secure and Efficient Mutual Authentication Protocol Using Smart Card. *Journal of Information Security and Applications*, **51**, Article ID: 102443. https://doi.org/10.1016/j.jisa.2019.102443

[33] Tsobdjou, L.D., Pierre, S. and Quintero, A. (2021) A New Mutual Authentication and Key Agreement Protocol for Mobile Client—Server Environment. *IEEE Transactions on Network and Service Management*, **18**, 1275-1286. https://doi.org/10.1109/TNSM.2021.3071087

[34] Miller, V.S. (1986) Use of Elliptic Curves in Cryptography. In: Williams, H.C., Ed., *Advances in Cryptology—CRYPTO'*85 *Proceedings*, *CRYPTO* 1985, Lecture Notes in Computer Science, Vol. 218, Springer, Berlin, 417-426. https://doi.org/10.1007/3-540-39799-X_31

[35] Koblitz, N. (1987) Elliptic Curve Cryptosystems. *Mathematics of Computation*, **48**, 203-209. https://doi.org/10.1090/S0025-5718-1987-0866109-5

[36] Dolev, D. and Yao, A. (1983) On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, **29**, 198-208. https://doi.org/10.1109/TIT.1983.1056650

[37] Kocher, P., Jaffe, J. and Jun, B. (1999) Differential Power Analysis. In: Wiener, M., Ed., *Advances in Cryptology—CRYPTO'*99, Lecture Notes in Computer Science, Vol. 1666, Springer, Berlin, 388-397. https://doi.org/10.1007/3-540-48405-1_25

[38] Burrows, M., Abadi, M. and Needham, R. (1990) A Logic of Authentication. *ACM Transactions on Computer Systems* (*TOCS*), **8**, 18-36. https://doi.org/10.1145/77648.77649

[39] Blanchet, B. (2016) Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif. *Foundations and Trends® in Privacy and Security*, **1**, 1-135. https://doi.org/10.1561/3300000004

[40] Abadi, M. and Fournet, C. (2001) Mobile Values, New Names, and Secure Communication. *ACM SIGPLAN Notices*, **36**, 104-115. https://doi.org/10.1145/373243.360213