多服务器环境下基于椭圆曲线密码的改进的身份认证协议

殷秋实 陈建华
(武汉大学数学与统计学院 武汉 430072)

摘 要 传统的身份认证协议大部分都是采用用户名和口令的模式在基于数学问题难解的情况下衍生出来的。这类协议往往依赖于口令的复杂性、随机数发生器的性能以及较大的计算开销来确保通信的安全性,因而效率较低且实用性不强。为了成功规避上述问题,在引入生物因子及模糊提取器的基础上提出了一个基于椭圆曲线密码改进的身份认证协议,并用 Burrows-Abadi-Needham (BAN 逻辑)形式化地完成了双方密钥认证性的验证,随后又对其进行了安全性分析并与其他相关协议进行了性能比较。实验结果表明,此协议具备更高的安全性和更强的实用性。

关键词 多服务器环境,椭圆曲线密码,身份认证,模糊提取器,BAN 逻辑
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.019

Improved Identity Authentication Protocol Based on Elliptic Curve Cryptography in Multi-server Environment

YIN Qiu-shi CHEN Jian-hua
(School of Mathematics & Statistics, Wuhan University, Wuhan 430072, China)

Abstract Based on the model of user's name and password, most of the traditional identity authentication protocols are derived from the mathematical difficult problems. They often rely on the complexity of password, the performance of random generator and large computational cost to ensure the security of the communication, so they are lack of high efficiency and practicality. In order to avoid above problems successfully, based on the introduction of biological factors and fuzzy extractor, this paper proposed an improved identity authentication protocol based on elliptic curve cryptography and verified key authentication formally in both sides through Burrows-Abadi-Needham (short for BAN), and then carried out security analysis. Compared with other related protocols in performance, the proposed scheme is more secure and practical.

Keywords Multi-server environment, Elliptic curve cryptography, Identity authentication, Fuzzy extractor, BAN logic

随着网络技术的快速发展,尤其是智能手机的普及,人们的生活正式步入了高速信息化时代。通过智能手机,人们可以随时随地、方便快捷地获取所需的不同应用程序服务器的各种资源,但这些行为也使得人们的大量私密信息在公共网络中传输而容易遭到攻击者的窃听及非法攻击,因此亟需足够安全且实用的身份认证协议来确保通信安全。传统的身份认证协议大多采取"用户名+口令"的模式,利用 RSA 算法、ElGamal 算法、LUC 算法等公开密钥密码算法,通过较大的计算开销来保证通信的安全性。但是,由于口令具有低熵值性,且真正的随机数不可能由随机数发生器产生,其安全性实际上并未达到理论上的高度;而且对于服务器而言,随着分布式系统的广泛运用,同一在线服务提供商可同时提供多种不同的服务运用(以下简称为多服务器环境),如果用户在每个所属同一在线服务提供商的服务器上都需进行注册、登录以进行身份认证,显然会大大增加整个通信系统的开销。这些问题都极大地阻挠了安全通信的发展,因此寻求新的身份认证协议势在必行。

为了克服上述的缺点,Chang 等人[1]提出了多服务器场景下实现单点登录的身份认证思想,但该协议无法抵抗内部攻击。Fan 等人[2]于 2005 年提出了一个改进的可以拦截、删除、修改和重放等攻击的认证方案,但是由于口令自身具有低熵值性,其无法抵抗离线字典的攻击。Li 等人[3]于 2010 年提出了一种基于生物特征值的身份认证方案。由于生物特征值自身具有高熵值、唯一性及难伪造性的特点,其彻底突破了离线字典攻击的瓶颈。但遗憾的是,此协议并非多服务器环境下的身份认证协议,并且也无法抵抗服务器拒绝服务等攻击;此外,生物特征值也并非绝对安全,由于其具有唯一性,信息一旦泄露,将面临终身泄露的风险。文献[4-5]提出的多服务器环境下利用椭圆曲线的身份认证协议虽然较为成功,但计算量较大,且无法规避离线字典攻击。Chaudhry[6]虽在多服务器环境下将椭圆曲线与生物特征值融合在一起,并在计算效率上做出了长足的改进,但 Xia[7]却指出该协议无法抵抗拒绝服务、用户模仿等常见攻击。为此,Xia[7]提出了自己的优化方案,并声称所提协议无论是在 BAN 检测环境下亦

到稿日期:2017-04-12 返修日期:2017-08-09
殷秋实(1993-),男,硕士生,主要研究方向为密码与信息安全,E-mail:qiusy_2017@163.com;**陈建华**(1963-),男,教授,博士生导师,主要研究方向为数论与密码,E-mail:chenjh_ecc@163.com(通信作者)。

或是启发式安全分析下都足以抵抗各类常见攻击。然而，文献[8-9]却指出包含 Xia 提出的方法在内的绝大部分方法对基于智能卡的双因子或三因子的身份认证协议并没有建立明确的“攻击者模型”，而仅依据其在某些方面具有的优势便笼统地断言所提协议在各方面都优于先前的协议。这种评价显然是没有说服力的，因而必须建立一个衡量安全标准的“尺度”来作出客观的评价。基于此，我们对文献[7]进行深入研究，发现其除了缺乏文献[8-9]所论述的“攻击者模型”外，安全性也是建立在相当大的计算开销上，如果将其应用到有限带宽的通信线路上，显然是不切实际的。此外，虽然上述文献都声称自己的协议能够应用到多服务器环境，但对于所属同一在线服务提供商的服务器是如何避免重复注册以达到直接认证的过程并没有予以说明。针对上述缺陷，同时为了凸显本文的优势，在文献[7]的基础上，对文献[8-9]的“攻击者模型”稍加改进，在强化协议安全性的同时大大简化了繁琐的计算过程，并系统地阐述了如何将协议应用到多服务器环境下；然后利用 BAN 逻辑对会话密钥的认证性进行验证；最后通过将所提协议与文献[6-7]协议的安全性和计算效率进行对比，证实了本文的协议更加实用及安全。

1 相关概念

1.1 单向散列函数

单向散列函数是现代密码学的中心，也是许多协议的另一个结构模块。由于函数自身具有单向性，它无法反向求逆，且能够把可变长度的输入串映射成固定长度（通常比输入长度更短）的输出串。好的单向散列函数同时也是无冲突的，即很难产生两个不同的输入串而使它们有相同的输出值。抗强碰撞的哈希函数 $h(\cdot)$即为较好的单向散列函数，因此本文使用 $h(\cdot)$作为单向散列函数。

1.2 模糊提取器

传统的密码和随机串相结合的口令形式由于自身具有低熵值的特点而无法抵抗离线字典攻击，因此被生物特征值所取代。但是，密码机制中的秘密值通常是均匀分布的随机串，而且在需要时还必须精确再生（譬如作为 $h(\cdot)$的输入值，如果预映射值有一位偏差，散列值便会产生巨大的差异）。这显然与现实世界中的秘密值相冲突，现实中即便是复杂得无法伪造的生物特征值，也无法实现精确再生。为了从根本上解决此类问题，真正将生物特征值用于密码技术，Dodis 等人[10]引入了模糊提取器，其利用纠错码技术，在误差允许的范围内从变化的生物特征中稳定地提取出分布一致的密钥。为了简化起见，本文中的模糊提取器沿用文献[7]的构造，即用一个函数对(Gen,Rep)表示模糊提取器。其中，$Gen(\cdot)$为随机生成函数，形式为 $Gen(BIO_i)=(\delta_i,v_i)$，$BIO_i$ 是用户 U_i 的生物特征值，δ_i 是与 BIO_i 相对应的随机字符串，v_i 是辅助的随机字符串；$Rep(\cdot)$是确定性恢复函数，形式为 $Rep(BIO_i',v_i)=\delta_i$，BIO_i'是 U_i 误差允许范围内的生物特征值。经过上述过程，最终得到与 BIO_i 相同的随机串 δ_i。

1.3 椭圆曲线密码

椭圆曲线已被研究了很多年。早在 20 世纪 80 年代中叶，Koblitz 和 Miller 就分别提出了将它用于公开密钥密码体制[11-12]，曲线由方程式 $y^2=x^3+ax+b(\mathrm{mod}\ p)$定义。其中，$p$ 是一个素数或素数的幂（有时也称为椭圆曲线的秩），且判别式 $\Delta=4a^3+27b^2\neq 0$。公钥加密算法必须建立在一个难解的数学问题之上。例如，RSA 算法是基于大整数素因子分解的困难性；ElGamal 算法是基于离散对数求解的困难性；ECC 亦是如此，但是其实现同等级别安全性所需密钥的大小却远非 RSA 所能比拟的。从表 1 可以看出，ECC 比 RSA 更适合应用在身份认证协议中。

表 1 同等安全级别所需密钥的大小

Table 1 Secret key required for same security level

安全性	RSA	ECC
低	512	112
中	1024	161
高	3072	256
很高	15360	512

本文协议的安全性就建立在有限域（为大素数）上的椭圆曲线离散对数问题（Elliptic Curve Discrete Logarithm Problem，ECDLP）及 Diffie-Hellman 问题（ECDHP）上。对于这两个问题，到目前为止都没有行之有效的算法能够在较短的时间内得到相应的解，而这也恰恰满足了下面行的公开密钥密码体制的要求。下面对这两个问题进行简要介绍。

1.3.1 ECDLP

设 F_p 为有限域，$E_p(a,b)$是 F_p 上的椭圆曲线（$a,b\in F_p$）。给定 $E_p(a,b)$上的两点 P 和 Q，ECDLP 是指找一个整数 $s\in F_p^*$，使得满足 $Q=sP$。

1.3.2 ECDHP

已知 $E_p(a,b)$上的 3 个点 $P,sP,tP(s,t\in F_p^*)$，ECDHP 是指寻求 $E_p(a,b)$上的点 Q，使得 $Q=stP$。

1.4 攻击者模型

攻击者模型即对攻击者能力事先做出明确的假定，通过此假定使得协议的安全性变得“可测”。为了突出协议的实用性，这个假定必须基于现实中的合理性。基于本文提出的“智能卡＋生物特征值”的双因子认证方案，本文对文献[8]的攻击者模型稍加改进。将其主要特征归纳如下：

1）对于任何在公共信道上传输的消息，攻击者拥有完全的访问控制权，包括但不限于窃听、截获、删除、修改、重放等操作[13]；

2）攻击者在离线的情况下能够枚举出所有可能的由用户名和口令组成的笛卡尔积空间；

3）在已窃取 SC 的基础上，攻击者可以利用能量分析攻击[14-16]在多项式时间复杂度内得到存储在 SC 里的全部私密信息；

4）攻击者能够获取通信双方以前的会话密钥；

5）攻击者能够以某种非法手段得到受害者的生物特征值；

6）特征 3）和 5）不可能同时成立。

上述假定基本与文献[8]的攻击者模型相同，但用生物特征值取代了传统的口令模式，而且附加的特征 6）也是较为合理的。考虑到生物特征值自身泄露的低可能性以及利用能量分析攻击获取 SC 存储的私密信息不可能在短时间内（其时间复杂度是多项式的）完成的特点，U_i 有足够多的时间令攻

击者手上的 SC 失效(可通过挂失等操作)；否则，此类攻击者模型为平凡的模型[8]，即对任何双因子认证协议，攻击者都能够予以破解，这显然是不合实际的。

2　改进方案

2.1　协议的概括及相关符号说明

改进的方案建立在文献[7]的基础上。为了更加方便地实现生物特征值的输入，仍采用文献[7]的生物特征值及模糊提取器的概念。但对于文献[7]而言，在多服务器环境下，以较大的计算开销来换取协议的安全性显然是不切实际的。本文虽然用到了服务器的公钥与私钥，但采用的却是对称密码体制，这在优化计算性能的同时把 ECC 上的公钥加密及私钥解密等复杂运算留给了攻击者。攻击者试图破解此类协议时，必然面临 ECDLP 及 ECDHP 这类不可解问题。同时，本文系统地探讨了如何对同一在线服务提供商的服务器实现免注册的认证过程。由此可见，此协议在不失安全性的基础上，较文献[7]在实用性及计算效率上都得到了较大程度的优化。为了更形象地描述协议的具体实施过程，本文另附图 1 来形象地展示改进协议的工作流程。

为了简化起见，改进方案的通信主体仍沿用在多服务器环境上的 3 个参与方：注册中心 RC、服务器 S_j 与用户 U_i。其中，注册中心是一个可信的第三方，它负责系统的建立、参数的选取以及用户和服务器的注册。RC 首先选定基于 F_p 的椭圆曲线 $E_p(a,b)$、其解点集上的生成元 P 和一个抗强碰撞的哈希函数 $h(\cdot)$，并将 $E_p(a,b)$，P 及 $h(\cdot)$公示出来。同时，为了体现本文方案能够适应多服务器环境下的身份认证，本文对 S_j 引入了参数 σ_j。与 S_j 属于同一在线服务商的服务器(S_{j1})均具有相同的 σ_j，且与 RC 持有相同的安全密钥 PSK_j。相较于传统的身份认证协议，本文采用模糊提取器并使用生物特征值这一高熵值生成随机串的方式取代了传统模式下的“口令＋随机数”的方式，使得新的协议在避免离线字典攻击的前提下成功地省去了传统协议中的口令修改阶段。因此，新协议分为了 4 个阶段，即注册阶段、登录阶段、认证阶段以及多服务器认证阶段，从而在节省系统计算开销的同时，真正意义上地实现了多服务器环境下的应用。在本文中，U_i 通过注册来获取 RC 颁发的智能卡 SC，并利用 SSC 来完成与服务器之间的双向认证。协议中所用到的符号及说明如表 2 所列。

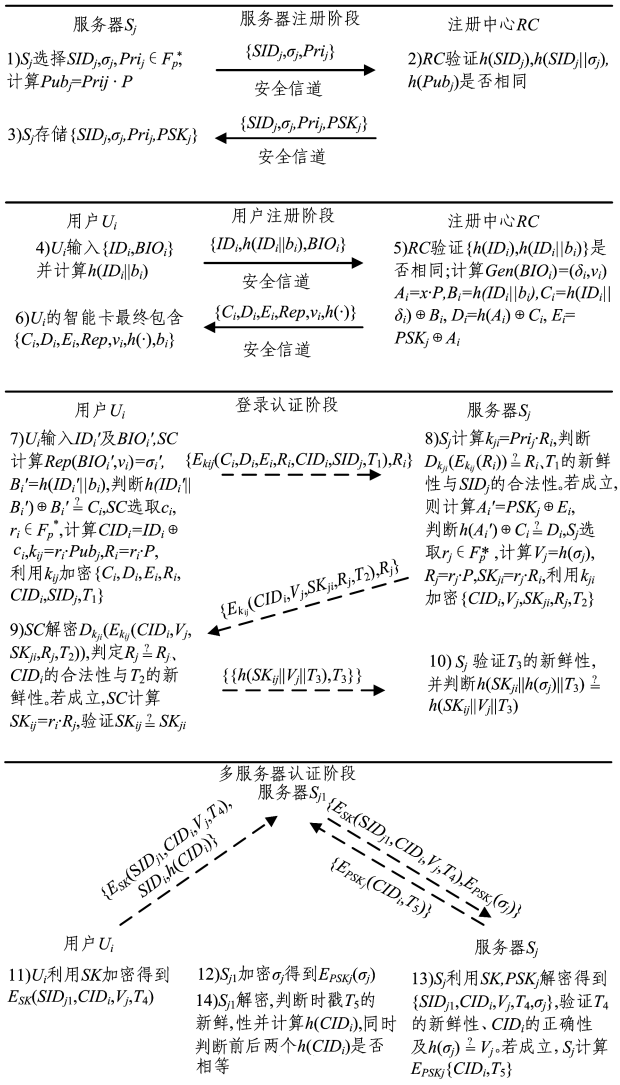

图 1　协议工作流程图

Fig. 1　Flowchart of protocol

表 2　本文用到的符号及说明

Table 2　Symbols and instructions used in this paper

符号	说明	符号	说明
U_i	用户 i	P	$E_p(a,b)$上的生成元
RC	注册中心	x	RC 的主密钥($x\in F_p^*$)
ID_i	i 的用户名	PSK_j	RC 与 S_j，S_{j1}共享的安全密钥
BIO_i	i 的生物特征值	Pub_j，Pub_{j1}	服务器 j 与 j_1 的公钥
S_j，S_{j1}	服务器 j 与服务器 j_1	Pri_j，Pri_{j1}	服务器 j 与 j_1 的私钥
SID_j，SID_{j1}	服务器 j 与 j_1 名称	‖	字符串连接运算符
σ_j	S_j，S_{j1}所属同一服务商的属性	$\oplus$	异或运算

2.2　注册阶段

注册阶段分为服务器注册和用户注册两部分。

2.2.1　服务器注册

S_j 首先选定它的身份 SID_j、所属服务商的属性 σ_j、私钥 $Pri_j\in F_p^*$，并计算它的公钥 $Pub_j=Pri_j\cdot P$；随后将$\{SID_j,\sigma_j,Pub_j\}$通过安全信道[17]发送给 RC。RC 收到消息后，先利用 $h(\cdot)$分别计算 $h(SID_j)$，$h(SID_j\|\sigma_j)$，$h(Pub_j)$，并将其与 RC 中存储的哈希服务器身份表与哈希服务器公钥身份表进行比对。若有一个与之相同，则返回$\{SID_j,\sigma_j,Pub_j\}$让其重新注册；否则，RC 通过安全信道向其颁发 PSK_j(PSK_j 的值由 S_j 的 σ_j 所确定)，并把 $h(SID_j)$，$h(SID_j\|\sigma_j)$，$h(Pub_j)$存储在相应的哈希表中，随后公示 S_j 的 Pub_j。

2.2.2 用户注册

U_i 选取自己的 ID_i 并键入自己的生物特征值 BIO_i，随后选取一个随机数 b_i，并计算 $h(ID_i \| b_i)$，最后将$\{ID_i, h(ID_i \| b_i), BIO_i\}$通过安全信道发送给 RC。

RC 收到消息后，先判断 $h(ID_i)$与 $h(ID_i \| b_i)$是否与自己所存储的哈希用户身份表的名单相同。若相同，则返回 U_i 的注册信息令其重新注册；否则，RC 利用自身的模糊提取器中的随机生成函数计算 $Gen(BIO_i)=(\delta_i, v_i)$，$A_i=x \cdot P$，$B_i=h(ID_i \| b_i)$，$C_i=h(ID_i \| \delta_i) \oplus B_i$，$D_i=h(A_i) \oplus C_i$，$E_i=PSK_j \oplus A_i$，并把$\{C_i, D_i, E_i, Rep, v_i, h(\cdot)\}$输入到智能卡 SC，最后通过安全信道将智能卡 SC 颁发给 U_i。

U_i 收到 SC 后，将自己选择的随机数键入到 SC 中。这样，SC 最终包含$\{C_i, D_i, E_i, Rep, v_i, h(\cdot), b_i\}$。

2.3 登录阶段

用户 U_i 需要 S_j 提供服务时，将智能卡插入到读卡器终端，并输入自己的 ID_i'及 BIO_i'，智能卡随后执行如下操作。

1)SC 利用确定性恢复函数 $Rep(\cdot)$及辅助随机字符串 v_i 计算 $Rep(BIO_i', v_i)=\sigma_i'$，$B_i'=h(ID_i' \| b_i)$，并判断 $h(ID_i' \| \sigma_i') \oplus B_i' \stackrel{?}{=} C_i$。若不等，则返回并令其重新登录，如果连续 3 次输入出错，则 SC 当天自动处于锁定状态，不接受任何登录请求；否则，进入下一步。

2)SC 选取随机数 $c_i, r_i \in F_p^*$，首先计算 U_i 的临时身份信息 $CID_i=ID_i \oplus c_i$；然后利用需要接入服务器 S_j 的公钥 Pub_j 计算对称密钥 $k_{ij}=r_i \cdot Pub_j$ 以及密钥生成因子 $R_i=r_i \cdot P$，并利用 k_{ij} 对$\{C_i, D_i, E_i, R_i, CID_i, SID_j, T_1\}$进行加密；最后将加密结果 $E_{k_{ij}}(C_i, D_i, E_i, R_i, CID_i, SID_j, T_1)$连同 R_i 一并发送给 S_j，其中 T_1 为当前时间戳。

2.4 双向认证阶段

S_j 收到消息后，利用自己的私钥 Pri_j 计算 $k_{ji}=Pri_j \cdot R_i$，并利用 k_{ji} 对 $E_{k_{ij}}(C_i, D_i, E_i, R_i, CID_i, SID_j, T_1)$进行解密，确定加密消息里的 R_i 与未加密的 R_i 是否相同、当前时戳 T_1 的新鲜性以及 SID_j 是否为自己的身份信息。若与接收时间点相比超过了允许的误差、两个 R_i 不同以及 SID_j 不是自己的身份信息时，终止认证；否则 S_j 利用 PSK_j 计算 $A_i'=PSK_j \oplus E_i$，判断 $h(A_i') \oplus C_i \stackrel{?}{=} D_i$。如果相等，协议继续进行；否则，终止认证。

S_j 选取随机数 $r_j \in F_p^*$，计算 $V_j=h(\sigma_j)$，$R_j=r_j \cdot P$ 及会话密钥 $SK_{ji}=r_j \cdot R_i$，并用密钥 k_{ji} 对$\{CID_i, V_j, SK_{ji}, R_j, T_2\}$进行加密；最后将加密结果 $E_{k_{ji}}(CID_i, V_j, SK_{ji}, R_j, T_2)$连同 R_j 一并发送给 SC，其中 T_2 为当前时间戳。

SC 收到消息后，先利用对称密钥 k_{ij} 对 $E_{k_{ji}}(CID_i, V_j, SK_{ji}, R_j, T_2)$进行解密，确定 CID_i 是否为此次通话的临时身份信息且加密消息里的 R_j 与未加密的 R_j 是否相同，并判断当前时戳 T_2 的新鲜性。若有一个不满足要求，则终止认证；否则，SC 计算 $SK_{ij}=r_i \cdot R_j$，并验证 $SK_{ij} \stackrel{?}{=} SK_{ji}$。如果相等，$U_i$ 完成对 S_j 的认证，同时 SC 选取新的时戳 T_3，计算并发送$\{h(SK_{ij} \| V_j \| T_3), T_3\}$给 S_j。

S_j 收到消息后，先判断 T_3 的新鲜性。如不新鲜，则终止认证；否则 S_j 验证 $h(SK_{ji} \| h(\sigma_j) \| T_3) \stackrel{?}{=} h(SK_{ij} \| V_j \| T_3)$。若两者相等，则 U_i 与 S_j 完成双向认证，两者共享会话密钥 $SK=SK_{ij}=SK_{ji}$。

2.5 多服务器认证阶段

U_i 一旦完成了与 S_j 的双向认证，便可直接得到与 S_j 属于同一在线服务商的服务器 S_{j1} 的认证，从而达到免注册而直接登录的状态。

U_i 用此次验证成功的会话密钥 SK 加密$\{SID_{j1}, CID_i, V_j, T_4\}$得到 $E_{SK}(SID_{j1}, CID_i, V_j, T_4)$，随后将$\{E_{SK}(SID_{j1}, CID_i, V_j, T_4), SID_j, h(CID_i)\}$发送给服务器 S_{j1}。S_{j1} 首先利用自己与注册中心共享的安全密钥 PSK_j 对自己的属性 σ_j 进行加密，从而得到 $E_{PSK_j}(\sigma_j)$，然后根据 SID_j 将加密消息$\{E_{SK}(SID_{j1}, CID_i, V_j, T_4), E_{PSK_j}(\sigma_j)\}$发送给服务器 S_j。S_j 利用自己的 SK 与 PSK_j 分别对其进行解密运算，得到$\{SID_{j1}, CID_i, V_j, T_4, \sigma_j\}$，验证 T_4 的新鲜性并判断 CID_i 的正确性以及 $h(\sigma_j) \stackrel{?}{=} V_j$。若不同，则终止后续操作；否则，$S_j$ 认为 U_i 为已处于登录状态的用户，并且其申请访问的服务器 S_{j1} 与自己属于同一在线服务提供商。于是，S_j 接受请求并用 PSK_j 对 CID_i 和 T_5 进行加密，得到 $E_{PSK_j}(CID_i, T_5)$，然后根据请求登录的服务器名 SID_{j1} 将 $E_{PSK_j}(CID_i, T_5)$发送给 S_{j1}。S_{j1} 进行解密，判断时间戳 T_5 的新鲜性，并计算 $h(CID_i)$，最后判断前后两个 $h(CID_i)$是否相等。如果不等，则终止访问；否则，S_{j1} 视 U_i 为已注册且通过 S_j 认证的合法用户，接受其访问请求。

3 会话密钥认证性验证及性能分析

3.1 BAN 逻辑验证

本节将对改进协议的会话密钥进行认证性验证。本文采用著名的 Burrows-Abadi-Needham（BAN 逻辑）[18-19]进行形式化分析。BAN 逻辑的基本符号及其意义如表 3 所列。

表 3 BAN 逻辑中的符号及意义

Table 3 Symbolic description in BAN'S logic

符号	意义
P, Q	参加通信的主体
X, Y	消息语句
K	加密密钥
(X, Y)	X 和 Y 的连接
$\langle X \rangle_Y$	X 和 Y 的组合
$\{X\}_K$	用 K 加密 X 的结果
$P \mid \equiv X$	P 认为 X 为真
$P \lhd X$	P 曾收到包含 X 的消息
$P \mid \sim X$	P 发送过包含 X 的消息
$P \mid \Rightarrow X$	P 对 X 有控制权
$\sharp(X)$	X 是新鲜的
$P \overset{K}{\leftrightarrow} Q$	P 和 Q 共享密钥 K
$P \overset{X}{\rightleftharpoons} Q$	P 和 Q 共享秘密 X
$\overset{K}{\rightarrow} P$	K 是 P 的公开密钥

下面给出后文证明中可能用到的 BAN 逻辑的几条基本规则。

1)消息含义规则

$$\frac{P \mid \equiv P \overset{K}{\leftrightarrow} Q, P \lhd \{X\}_K}{P \mid \equiv Q \mid \sim X}, \frac{P \mid \equiv P \overset{Y}{\rightleftharpoons} Q, P \lhd \{X\}_Y}{P \mid \equiv Q \mid \sim X}$$

2）管辖权规则

$$\frac{P|\equiv Q|\Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X)}$$

3）临时值校验规则

$$\frac{P|\equiv \#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$$

4）接受消息规则

$$\frac{P\lhd (X,Y)}{P\lhd X}, \frac{P\lhd \{X\}_K}{P\lhd X}, \frac{P|\equiv P\overset{K}{\leftrightarrow}Q, P\lhd \{X\}_K}{P\lhd X}$$

5）新鲜性规则

$$\frac{P|\equiv \#(X)}{P|\equiv \#(X,Y)}$$

6）信念规则

$$\frac{P|\equiv X, P|\equiv Y}{P|\equiv (X,Y)}, \quad \frac{P|\equiv (X,Y)}{P|\equiv X, P|\equiv Y}, \quad \frac{P|\equiv Q|\equiv (X,Y)}{P|\equiv Q|\equiv X}, \quad \frac{P|\equiv Q|\sim (X,Y)}{P|\equiv Q|\sim X}$$

7）会话密钥规则

$$\frac{P|\equiv \#(K), P|\equiv Q|\equiv R}{P|\equiv P\overset{K}{\leftrightarrow}Q}$$

其中，R 是 K 的重要组成部分。

对于本文而言，需证明两个目标：1）$U_i|\equiv U_i\overset{SK}{\leftrightarrow}S_j$；2）$S_j|\equiv U_i\overset{SK}{\leftrightarrow}S_j$。

现将此协议简化为如下模型。

消息 1：$U_i\rightarrow S_j$：$\{\{C_i, D_i, E_i, R_i, CID_i, SID_j, T_1\}_{k_{ij}}, R_i\}$

消息 2：$S_j\rightarrow U_i$：$\{\{CID_i, V_j, SK_{ji}, R_j, T_2\}_{k_{js}}, R_j\}$

消息 3：$U_i\rightarrow S_j$：$\{\{h(SK_{ij}, V_j, T_3), T_3\}\}$

同时，对该协议做如下初始化假设。

假设 1：$U_i|\equiv (U_i\overset{k_{ij}}{\leftrightarrow}S_j)$

假设 2：$U_i|\equiv \{D_i, E_i, r_i\}$

假设 3：$U_i|\equiv \#(r_j), U_i|\equiv \#(T_2)$

假设 4：$U_i|\equiv S_j|\Rightarrow (r_j), U_i|\equiv S_j|\Rightarrow (T_2)$

假设 5：$S_j|\equiv (U_i\overset{k_{ij}}{\leftrightarrow}S_j)$

假设 6：$S_j|\equiv \{SID_j, r_j\}$

假设 7：$S_j|\equiv \#(T_1), S_j|\equiv \#(T_3), S_j|\equiv \#(r_i)$

假设 8：$S_j|\equiv U_i|\Rightarrow (T_1), S_j|\equiv U_i|\Rightarrow (T_3), S_j|\equiv U_i|\Rightarrow (r_i)$

因此，用 BAN 逻辑进行形式化证明的步骤如下。

1）根据消息 1 有：$S_j\lhd (\{C_i, D_i, E_i, R_i, CID_i, SID_j, T_1\}_{k_{ij}}, R_i)$。

2）根据消息接收规则有：$S_j\lhd \{C_i, D_i, E_i, R_i, CID_i, SID_j, T_1\}_{k_{js}}$。

3）根据假设 5 及消息含义规则有：$S_j|\equiv U_i|\sim \{C_i, D_i, E_i, R_i, CID_i, SID_j, T_1\}$。

4）根据假设 7 和新鲜性规则有：$S_j|\equiv \#(\{C_i, D_i, E_i, R_i, CID_i, SID_j, T_1\})$。

5）根据临时校验值规则，由步骤 3）和步骤 4）可得：$S_j|\equiv U_i|\equiv \{C_i, D_i, E_i, R_i, CID_i, SID_j, T_1\}$。

6）再通过信念规则有：$S_j|\equiv U_i|\equiv \{R_i, T_1\}$。

7）通过假设 8，根据管辖权规则有：$S_j|\equiv \{R_i, T_1\}$。

8）根据步骤 7）和假设 6，由信念规则可知：$S_j|\equiv r_i, S_j|\equiv r_j$。

9）根据假设 7，由新鲜性规则可得：$S_j|\equiv \#(SK)$。

10）根据步骤 6）与步骤 9），由会话密钥规则可知：$S_j|\equiv S_j\overset{SK}{\leftrightarrow}U_i$，从而目标 2）得证。

11）根据消息 2 和假设 1，由消息含义规则可得：$U_i|\equiv S_j|\sim \{CID_i, V_j, SK_{ji}, R_j, T_2\}$。

12）由假设 3 和新鲜性规则有：$U_i|\equiv \#(\{CID_i, V_j, SK_{ji}, R_j, T_2\})$。

13）根据临时校验值规则可得：$U_i|\equiv S_j|\equiv \{CID_i, V_j, SK_{ji}, R_j, T_2\}$。

14）由信念规则得：$U_i|\equiv S_j|\equiv \{R_j, T_2\}$。

15）根据假设 4，由管辖权规则可知：$U_i|\equiv \{R_j, T_2\}$。

16）根据信念规则：$U_i|\equiv R_j$。

17）由假设 3 和新鲜性规则可得 $U_i|\equiv \#(SK)$。

18）根据步骤 14）及会话密钥规则有：$U_i|\equiv U_i\overset{SK}{\leftrightarrow}S_j$，从而目标 1）得证。

至此，利用 BAN 逻辑成功地实现了对上述协议的证明。

3.2　安全性分析

3.2.1　抵抗拒绝服务攻击

由于生物特征值的不稳定性，已注册成为合法用户的 U_i 再次键入 BIO_i' 申请访问服务器时，会因为前后生物特征值不一致，导致服务器误将 U_i 视为非法用户而拒绝其访问请求。本文并没有将高熵值且不稳定的生物特征值 BIO_i 直接作为散列函数的预映射值，而是通过引入文献[7]的模糊提取器模型$\langle Gen, Ren\rangle$，采用纠错码的技术对同一 U_i 存在变化的 BIO_i 进行了预处理操作，解决了生物特征值的不稳定性与密码学要求密钥分布均匀一致的不兼容性问题，从而成功地应对了上述攻击。

3.2.2　提供用户匿名性

由于在注册阶段采用了安全信道来实现信息的传输，因此攻击者如果试图实现非法攻击，就只能根据 U_i 的 SC 及非安全信道截获的信息进行密码分析。假定攻击者窃取了 U_i 的智能卡 SC，利用能量分析攻击获取了卡内的参数$\{C_i, D_i, E_i, Rep, v_i, h(\cdot), b_i\}$，此时由于 SC 中仅有参数 C_i 含有 ID_i，因此攻击者利用 C_i 求解 ID_i 时必然面临着哈希函数逆向不可解的困难；即便攻击者同时截获了通信双方在非安全信道传输的信息 $E_{k_{ij}}(C_i, D_i, E_i, R_i, CID_i, SID_j, T_1)$ 和 $E_{k_{js}}(CID_i, V_j, SK_{ji}, R_j, T_2)$，也因为密钥 k_{ji} 的解密运算而同时面临 ECDLP，ECDHP 以及单项函数求逆等诸多不可解问题。由此可知，本方案很好地实现了用户的匿名性。

3.2.3　双因子安全性

本文的一大亮点在于虽然协议本身是基于“智能卡＋生物特征值”的双因子认证协议，但是由于 ID_i 具有匿名性，即便 U_i 的 SC 丢失且 BIO_i 遭到泄露（前文已论述两者不能同时成立，此处仅是为了凸显协议的安全性所做的假设），攻击者要想成功模仿 U_i 进入系统仍必须从笛卡尔积中得到 U_i 的 ID_i。由于 SC 自身提供了用户名猜测次数的限制性，因此从某种意义上可将协议视为基于“智能卡＋用户名＋生物特征值”的“三因子认证协议”。在“三因子”的条件下，双因子的安全性自然得到了保证。

3.2.4 抗离线字典攻击

由于本文是基于“智能卡+生物特征值”的双因子认证协议，因此此处针对 ID_i 的离线字典攻击进行分析。基于事先给定攻击者模型，攻击者成功猜测出 ID_i 是小概率事件，故此类攻击为不可能攻击。

3.2.5 抗重放攻击

重放攻击是指攻击者截获之前已经经 U_i 或 S_j 认证过的消息并将其再次重新发送给 U_i 或 S_j 以达到欺骗的目的。由于在 U_i 与 S_j 通信过程中始终有新鲜时间戳随机数作保证，并且对于已截获的加密消息 $E_{k_{ij}}(C_i, D_i, E_i, R_i, CID_i, SID_j, T_1)$，由于有 ECDLP 困难性的保证，攻击者无法改变 $E_{k_{ij}}$ 所加密的消息。因此，协议能有效地抵制重放攻击。

3.2.6 抗用户模仿攻击

无论是未经注册的非法用户，还是恶意的合法用户，要想模仿合法用户 U_i 完成后续认证过程，就必须成功拥有 U_i 的 ID_i 与 BIO_i。而攻击者即便在获取了 U_i 的智能卡条件下利用能量分析攻击得到了卡内的相关参数，并在之前截获了通信信息，要想得到正确的 ID_i 与 BIO_i，也面临着目前仍然不可解的数学难题。

3.2.7 抗服务器模仿攻击

根据服务器模仿的对象，可以将攻击分为两类：1)模仿合法用户来非法访问其他服务器资源；2)模仿合法服务器对用户进行欺骗。一个恶意的合法服务器 S_j 要想模仿 U_i 登录到其他服务器，就必须得到 U_i 的 ID_i 与 BIO_i；而协议中 U_i 在与 S_j 认证的过程中始终匿名地采用自己的临时身份信息 CID_i，S_j 更无法获取 U_i 的 BIO_i。此外，攻击者要想模仿合法服务器 S_j 来达到欺骗 U_i 的目的，则必须知道 S_j 的私钥 Pri_j 及 PSK_j；而 Pri_j 的求解必然会遇到 ECDLP 困难。因此，攻击者必然无法制造此类攻击。

3.2.8 提供前向安全性

前向安全性是指即使系统的主密钥丢失，也不会给依据该主密钥所建立的会话密钥 SK 带来影响。对于本文而言，RC 的主密钥 x 即使丢失，也不影响 S_j 的私钥 Pri_j 和 R_i 的选取，即它们之间是相互独立的。由于每次通信所建立的 SK 都不相同，且 SK 的求解面临 ECDHP 困难，因此该方案很好地保证了前向安全性。

3.2.9 抗内部攻击

由于生物特征值 BIO_i 具有不可伪造性，内部人员即便在注册阶段知晓了 U_i 的 ID_i 与 BIO_i，也无法伪造 BIO_i 进行攻击。

3.2.10 抗会话密钥攻击

在上述模型的假设下，虽然攻击者有能力窃取通信双方以前的会话密钥，但是由于每次通信的 SK 具有随机性的特点，攻击者对所拥有的 SK 并不足以产生有威胁的攻击。

3.2.11 抗修改攻击

由于在公共信道上通信的整个过程采用的是基于椭圆曲线密码的对称加密技术，因此，对加密信息本身进行篡改在 ECDLP 及 ECDHP 的条件下为不可能事件，攻击者能篡改的仅是未加密的密钥生成因子 GR 与时间戳 T。无论何者发生变化，U_i 与 S_j 都都够轻易利用自身密钥解密并发现两者的不同，从而终止认证过程。

3.2.12 抗中间人攻击

由于此协议能较好地实现了 U_i 与 S_j 之间的双向认证，因此改进方案能够成功地规避中间人攻击。

3.3 计算效率的对比

前文已对本文方案的安全性给出了证明及分析，现将其与文献[6-7]的方案进行效率对比，以突出本文方案的优势。

为了一目了然地刻画协议的计算消耗，采用以下符号表示各种不同运算所需的时间及对比关系[20]：T_h 表示一次哈希运算所需要的时间；T_s 表示一次对称加密所需要的时间；T_{ECADD} 表示椭圆曲线上一次加法运算所需要的时间；T_{ECMUL} 表示椭圆曲线上的一次点乘运算所需要的时间，其中 $T_{ECMUL}=6T_{ECADD}$。此外，由于在本协议中省去了传统方法的口令变更阶段，并诠释了单点登录的思想[21-22]，因此与其他协议相比，本文在保证安全性的基础上进一步优化了协议的通信效率，使得该协议更具通用性。本文协议与其他协议的比较结果如表 4 所列。

表 4 本文协议与其他协议的效率比较

Table 4 Efficiency comparison of proposed protocol and other protocols

	文献[6]协议	文献[7]协议	本文协议
注册	$T_{ECMUL}+5T_h$	$T_{ECMUL}+10T_h$	$T_{ECMUL}+8T_h$
登录	$2T_{ECMUL}+5T_h$	$2T_{ECMUL}+7T_h$	$2T_{ECMUL}+2T_h+T_S$
认证	$T_{ECMUL}+6T_h$	$4T_{ECMUL}+2T_{ECADD}+9T_h$	$4T_{ECMUL}+3T_h+3T_S$
口令修改	$7T_h$	$7T_h$	—
合计	$4T_{ECMUL}+23T_h$	$7T_{ECMUL}+2T_{ECADD}+33T_h$	$7T_{ECMUL}+4T_S+23T_h$

由于文献[6-7]并未说明如何将协议应用到多服务器环境下，因此在计算方面的比较并没有引入多服务器认证阶段。显然，对称加密算法要优于椭圆曲线上的运算，因此本文方案较文献[7]的方案具有更高的计算效率。虽然所提方案在计算上稍逊于文献[6]的方案，但在安全性及实用性上远优于文献[6-7]的方案。

结束语 本文在引入文献[7]的生物特征值、模糊提取器及文献[8]的攻击者模型的基础上，结合椭圆曲线上的公钥密码体制与传统的对称密码算法，对文献[7]的模型在计算效率上进行了较大程度的优化。同时，针对包含文献[7]在内的其他文献没有明确的攻击者模型以及如何将方案应用于多服务器环境下的问题，本文系统地阐述了所提协议在多服务器环境下的应用过程。最后，通过 BAN 逻辑进行了形式化分析，并将其与其他相关协议进行对比，突出了此方案更安全和实用。因此，所提方案非常适用于分布式网络上的多服务器环境。

参 考 文 献

[1] CHANG C, LEE J. An efficient and secure multi-server password authentication scheme using smart card[C]// International Conference on Innovative Computing Information and Control. 2012:725-728.

(下转第 150 页)