多云环境下基于智能卡的认证方案

赵森，甘庆晴，王晓明，余芳
（暨南大学信息科学技术学院，广东 广州 510632）

摘 要：针对没有第三方参与认证智能卡存储的访问密钥随注册云的个数增加而线性增长的问题，提出一种多云环境下基于智能卡的认证方案。该方案在无第三方参与认证下，智能卡只需存储 2 个访问密钥就能实现用户与多个云服务器之间的身份认证，有效地减少了智能卡的存储费用。由于没有使用公钥密码技术，而是利用 XOR 同态函数和散列函数生成认证信息，从而有效降低了智能卡和云服务器的计算费用。此外，所提方案也不需要在多云端存储任何用户的信息，降低了云服务器的存储和管理费用。安全性分析和性能分析表明，所提方案能抵御多种攻击，是一个安全、高效的方案。

关键词：多云环境；身份认证；智能卡；XOR 同态函数
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000-436x.2018055

Authentication scheme for multi-cloud environment based on smart card

ZHAO Sen, GAN Qingqing, WANG Xiaoming, YU Fang
College of Information Science and Technology, Jinan University, Guangzhou 510632, China

Abstract: To solve the problem of the access keys stored in a smart card increasing linearly with the number of registered clouds without third party participated in authentication, an authentication scheme was proposed for multi-cloud environment based on smart card. In the proposed scheme, the authentication was realized between user and multiple clouds without third party participation when the smart card only stored two access key. Thus the storage cost of smart card was reduced effectively. Because there was no public key cryptography, the authentication messages was generated by using XOR homomorphic function and Hash function, thus the computational cost of the smart card and the cloud servers was reduced effectively. Moreover, the proposed scheme also didn't need to store any user's information on the cloud servers, thereby reducing the storage and management costs of the cloud servers. The security analysis and the performance analysis show that the proposed scheme is able to resist multiple attacks, which is secure and efficient.

Key words: multi-cloud environment, identity authentication, smart card, XOR homomorphic function

1 引言

随着大数据时代的到来，许多企业都选择运用多云服务对大数据进行存储与处理。由于云计算环境构建于一种开放的架构和接口之上，因此，可以将多个内部或外部的云服务整合在一起提供协同服务，这种分布式的云计算被称为“多云”（multicloud）[1]。在多云环境中，当云服务提供商

收稿日期：2017-10-12；**修回日期**：2018-02-27
通信作者：王晓明，twxm@jnu.edu.cn
基金项目：国家自然科学基金资助项目（No.61070164, No.61272415）；广东省自然科学基金资助项目（No.S012010008767）；广东省科技计划基金资助项目（No.2013B010401015, No.2012B091000136）；珠海优势学科信息安全基金资助项目
Foundation Items: The National Natural Science Foundation of China (No.61070164, No.61272415), The Natural Science Foundation of Guangdong Province (No.S012010008767), The Science and Technology Planning Project of Guangdong Province (No.2013B010401015, No.2012B091000136), Zhuhai Top Discipline-Information Security Project

拥有资源不足时，可以借用其他云的虚拟资源，如网络、服务器、存储、应用程序和服务。因此，多云服务可以为用户提供更好的服务，减少数据丢失和传输阻塞等风险，还能解决供应商锁定问题，从而使用户获得的服务性能得到改善。然而，多云服务在获得快速发展的同时，也带来许多安全问题，如多云环境下的用户认证、访问控制、数据隐私安全等。这些都给多云环境下的安全带来巨大挑战，使用传统的安全机制很难应付复杂多变的环境。因此，多云环境下的安全问题已经是当今的研究热点问题。

认证技术是信息安全的一个重要技术，而身份认证是保护多云安全的一个重要机制，有着举足轻重的作用。身份认证是识别和证明一个主体是否就是它所声称的那个主体的过程，用于鉴别用户的身份，限制非法用户访问系统资源。为了实现用户与云服务器之间的双向认证，已有许多安全认证方案被提出，如文献[1~11]，但它们主要是针对单云服务构建的。在这些方案中，当用户需要云服务时，首先需要在这个云注册，从而获得访问云服务的密钥和权力。在多云环境下，直接应用适合单云的认证方案，那么当用户需要从多个云获得不同的服务时，用户就必须向多个云提交注册信息，以便获得访问密钥和权力。然而，用户记住和管理多个访问密钥是件很困难的事，也是不安全的。更重要的是，随着注册云的个数增多，用户需要保存的访问密钥就越多。例如，用户使用智能卡或移动设备进行认证，那么智能卡或移动设备存储的访问密钥就随着注册云的个数呈线性增长，这对存储能力有限的智能卡或移动设备是不现实的。虽然利用第三方验证用户身份，可以降低用户的存储和计算费用，但要求所有的认证过程都需要第三方的参与，这必然增加了整个系统的通信费用和计算费用。同时，攻击者可能采用拒绝服务攻击等消耗第三方的资源，使第三方压力倍增，容易造成服务崩溃。此时，当合法用户想要获取服务时，第三方不能及时响应用户的请求。因此，基于单云环境下身份认证方案不能直接、高效地应用多云环境下的身份认证。

针对以上这些问题，提出了一种多云环境下基于智能卡的认证方案。所提方案不需要第三方参与认证，智能卡只需存储 2 个访问密钥就可以实现与多个云之间的认证和访问。所提方案解决了多云环境下智能卡存储的访问密钥随注册云的个数增加而呈线性增长的问题，有效地减少了智能卡的存储费用。利用 XOR 同态函数和散列函数生成认证信息，有效降低了智能卡和云服务器的计算费用。此外，所提方案也不需要在多云端存储任何用户的信息，从而降低了云服务器的存储和管理费用。安全性分析和性能分析表明，该方案能抵御多种攻击，是一种安全且高效的方案。

2 相关工作

随着云计算的不断发展，基于云计算的认证方案也逐渐成为研究热点之一，学者们在网络用户认证安全的基础上，分析了云计算在信息安全领域中的特点，提出了一系列云环境下的安全认证方案[3~17]。其中，文献[3]提出了一个在云环境下的身份认证方案，该方案是基于 IBE（基于身份的加密）和 IBS（基于身份的签名）提出来的。文献[4]采用盲签名和双线性对的方法实现了在云环境下的一个联合身份认证方案。该方案与传统方案相比，实现了用户匿名性、防止追踪以及隐私保护。通过分析，该方案在计算效率和安全性上有很大优势，但在一些特定的云应用中，该方案也没有很好地保护用户隐私。其后，许多改进方案被提出，如文献[5~7]。Choksi[8]从云环境下的安全和隐私保护技术的角度出发，对比分析了其他方案的特点，进而指出未来发展趋势和后续研究方向。为了加强认证的安全性，文献[9]提出了一个多层次认证方案，该方案利用多层结构认证实现访问云服务控制，且能抵抗虚拟化的攻击和内部攻击。文献[10]提出了一种基于云存储的认证方案，该方案利用证书和身份的加密（IBC）对访问云存储的用户进行身份验证。随后，文献[11]提出了一个云环境下的多层次图形密钥认证方案，利用图形密钥实现用户身份的认证。

针对多云环境下的身份认证模式，学术界已提出了不同的解决办法和策略。文献[12]提出了一个在多云环境下具有服务透明的用户认证方案。在该方案中，定义一个云代理的多云模型，并提出了一个适用于该模型的认证协议。文献[13]提出了一种保证匿名性的零知识认证协议。该协议能抵抗各种攻击，可以作为一种安全协议在多云环境中使用。文献[14]提出了一种将中央机构的秘密值秘密共享给参与主体的思想，并构建了一种混合云统一认证机制。在该方案中，认证中心的工作改由参与主体

合作完成，并给出了跨域认证方案和会话密钥协商方案。文献[15]提出了一种基于 Kerberos 的混合云服务中跨云际认证的机制。在这种机制中，云终端采取基于身份认证的方式直接和私有云进行认证，凭借企业私有云发放的票据访问企业存放在公有云中的数据。Bong 等[16]提出了多云环境下的一种快速的用户认证方法。该方案是基于票证的用户认证，即用户首先向云认证中心注册，当申请云服务时，需要云认证中心进行身份认证，然后发放访问票据。在这种机制中，云终端采取基于身份认证的方式直接和私有云进行认证，凭借企业私有云发放的票据访问企业存放在公有云中的数据。然而，这些方案都需要第三方参与认证，整个系统的通信费用和计算费用较高，还存在单点瓶颈等问题。文献[17]提出了一种分布式移动云计算服务的隐私认证方案，实现移动用户访问多个移动云的身份认证。该方案不需要第三方参与，且用户端存储的密钥也是常量。但该方案是基于双线性对的密码系统和动态随机数构造的，计算费用较高。

3　预备知识

3.1　散列函数

散列函数的输入长度是变长的，但其输出长度是固定的，该输出值被称为散列值。符号采用 $H(\cdot)$ 表示。散列函数具备以下几点性质[18]。

1) 输出长度固定性。任意长度的输入，得到的输出结果都是固定长度的。

2) 单向性。如果输入信息为 m，能计算出 $H(m)$，但如果已知 $H(m)$，则无法逆向计算出信息 m，散列函数是单向不可逆的。

3) 抗碰撞性。对于任意一个输入信息 m，找出另一个任意信息 m'使之满足 $m \neq m'$，则不可能出现 $H(m)=H(m')$的情况。

3.2　XOR 同态函数

XOR 同态函数指具有异或同态性质的伪随机函数，它能增强对数据泄露的保护，保证数据的隐私性。它的属性如下所示[19]。

命题 1　对于一个 XOR 同态函数 f，有

$$f\left(x_1 \oplus x_2\right) = f\left(x_1\right) \oplus f\left(x_2\right)$$

命题 2　如 k_1 和 k_2 是置换密钥，则

$$f_{k_1}\left(x_1\right) \oplus f_{k_2}\left(x_2\right) = f_{k_1 \oplus k_2}\left(x_1 \oplus x_2\right)$$

由于比特置换具有异或同态的性质，可以采用安全置换算法来保证方案的安全性。例如，著名洗牌算法 [20]能产生均匀分布的序列。

4　方案设计

所提方案包括可信中心（T）、多云服务器（S_j）和多个用户（U_i）。可信中心只负责生成系统参数和用户及云服务器的注册，不参与用户和多云服务器的认证。所提方案包括 5 个阶段：注册阶段、登录阶段、认证阶段、密码更改阶段和密码撤销阶段。表 1 列出本文方案涉及的符号及其含义。

表 1　　本文方案涉及的符号及其含义

符号	含义
T	可信中心
U_i	第 i 个用户
ID_i	用户 U_i 的身份
PWD_i	用户 U_i 的密码
S_j	第 j 个云服务器
ID_{S_j}	云服务器 S_j 的身份
p	一个大素数
k	T 的密钥
k_i	用户 U_i 的密钥
$\hat{k}_{S_j}$	云服务器 S_j 的密钥
$f_k(\cdot)$	带密钥的 XOR 同态函数
$H(\cdot)$	单向散列函数
T_i	读卡器的当前时间
T_S	服务器的当前时间
$\oplus$	异或操作
$\|\|$	连接操作

4.1　注册阶段

1) 用户注册

如果一个用户想要访问云端的数据，用户需要先向可信中心注册，注册的步骤如下所示。

Step1　用户 U_i 选择他的身份 ID_i、密码 PW_i 和一个随机数 $\alpha \in Z_p^*$（p 是一个大素数），并通过安全信道提交 $H(\alpha \| PW_i)$ 和 ID_i 信息给可信中心 T。

Step2　收到用户注册请求后，T 选择随机数 $(k, k_i) \in Z_p^*$，计算用户的访问密钥为($f_k(ID_i)$, k_i)和认证信息，并将($\sigma_i, \hat{\sigma}_i, \bar{\sigma}_i$)压入智能卡。

$$\sigma_i = f_k(ID_i) \oplus H(\alpha \| PW_i) \tag{1}$$

$$\hat{\sigma}_i = H(f_k(ID_i) \| ID_i) \tag{2}$$

$$\bar{\sigma}_i = k_i \oplus H(\alpha \| PW_i) \tag{3}$$

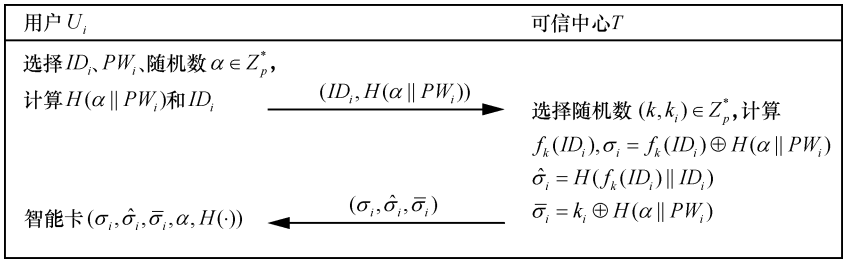

图 1 用户注册

Step3 T 通过安全信道发送智能卡给用户 U_i，智能卡中包含参数$(\sigma_i, \hat{\sigma}_i, \bar{\sigma}_i, H(\cdot))$。

Step4 收到智能卡后，用户 U_i 输入随机数 α 进入智能卡，最终智能卡中包含的参数为 $(\sigma_i, \hat{\sigma}_i, \bar{\sigma}_i, \alpha, H(\cdot))$。

用户注册过程如图 1 所示。

2) 云服务器注册

云服务器 S_j 按以下步骤完成注册。

Step1 云服务器 S_j 发送身份 ID_{s_j} 给可信中心 T。

Step2 收到 S_j 注册请求后，T 选择随机数 $\hat{k}_j \in Z_p^*$，则服务密钥为 $(f_k(ID_{s_j}), \hat{k}_j)$，并通过安全通道发送$(\hat{k}_j, f_k(ID_{s_j}))$给 S_j，完成 S_j 的注册。

云服务器注册过程如图 2 所示。

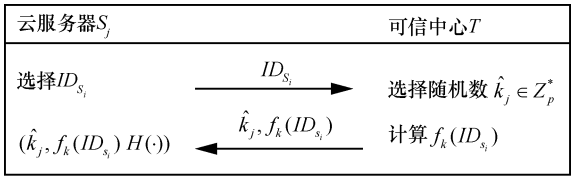

图 2 云服务器注册过程

4.2 登录阶段

用户登录智能卡阶段的步骤如下所示。

Step1 用户 U_i 插入智能卡进入读卡器，然后输入他的身份 ID_i 和密码 PW_i。

Step2 智能卡计算 $f_k(ID_i) = \sigma_i \oplus H(\alpha \| PW_i)$，$\hat{\sigma}_i' = H(f_k(ID_i) \| ID_i)$，并验证等式 $\hat{\sigma}_i \stackrel{?}{=} \hat{\sigma}_i'$ 是否成立。如果等式不成立，则智能卡拒绝用户的登录请求。否则，用户 U_i 可成功登录智能卡。

Step3 智能卡产生一个随机数 $r_i \in Z_p^*$，计算

$$k_i = \bar{\sigma}_i \oplus H(\alpha \| PW_i) \tag{4}$$

$$V_{i1} = f_k(ID_i) \oplus f_{k_i}(ID_{s_j} \oplus r_i) \tag{5}$$

$$V_{i2} = H(T_i \| V_{i1} \| f_{k_i}(ID_i \oplus r_i)) \tag{6}$$

然后，用户发送一个登录请求信息(V_{i1}, V_{i2}, T_i)到云服务器 S_j。其中，T_i 是时间戳。

4.3 认证阶段

用户与云服务器相互认证彼此的合法身份，并协商通信所需的会话密钥，其步骤如下所示。

Step1 云服务器 S_j 在接收到信息(V_{i1}, V_{i2}, T_i)后，首先检查时间戳 T_i。如果时间戳 T_i 有效，则云服务器计算 $\lambda_i = f_{k_i}(ID_i \oplus r_i) = V_{i1} \oplus f_k(ID_{s_j})$。然后云服务器 S_j 验证等式 $V_{i2} \stackrel{?}{=} H(T_i \| V_{i1} \| \sigma_i)$ 是否成立。如果等式不成立，则云服务器 S_j 拒绝认证请求。否则，云服务器 S_j 接收认证请求，并产生一个随机数 w_j，计算

$$S_{i1} = f_{\hat{k}_j}(ID_i \oplus w_j) \oplus f_k(ID_{s_j}) \tag{7}$$

$$S_{i2} = f_{\hat{k}_j}(ID_{s_j} \oplus w_j) \oplus \lambda_i \tag{8}$$

然后发送消息(S_{i1}, S_{i2}, T_s)给用户。同时，云服务器 S_j 可以计算出会话密钥

$$\begin{aligned} \zeta_{is_j} &= H(\lambda_i \| f_{\hat{k}_j}(ID_{s_j} \oplus w_j) \\ &= H(f_{k_i}(ID_i \oplus r_i) \| f_{\hat{k}_j}(ID_{s_j} \oplus w_j) \end{aligned} \tag{9}$$

Step2 接收到信息(S_{i1}, S_{i2}, T_s)后，用户 U_i 首先检查时间戳 T_s。如果时间戳 T_s 有效，则用户 U_i 计算 $\mu_j = f_{\hat{k}_j}(ID_{s_j} \oplus w_j) = S_{i1} \oplus f_k(ID_i)$，并验证式(10)。

$$H(f_{k_i}(ID_i \oplus r_i) \| T_s) \stackrel{?}{=} H((\mu_j \oplus S_{i2}) \| T_s) \tag{10}$$

如果等式成立，则用户认定云服务器是合法的。否则，用户 U_i 认为消息(S_{i1}, S_{i2}, T_s)不是服务器发送的合法信息。用户可以计算出会话密钥

$$\zeta_{is_j} = H(\lambda_i \| \mu_j) = H(f_{k_i}(ID_i \oplus r_i) \| f_{\hat{k}_j}(ID_{s_j} \oplus w_j)$$

与此同时，用户 U_i 计算确认信息给云服务器。如式(11)所示。

$$\rho_i = H(\mu_j \| \zeta_{is_j} \| T_i') \tag{11}$$

Step3　收到(ρ_i, T_i')，云服务器验证

$$\rho_i \overset{?}{=} H(f_{\hat{k}_j}(ID_{s_j} \oplus w_j) \| \zeta_{is_j} \| T_i') \tag{12}$$

如果等式成立，则确认对方是具有身份标识ID_i的合法用户。

登录和相互认证过程如图 3 所示。

4.4　密码更改阶段

用户U_i能在任何时间改变其密码，更改密码的步骤如下。

Step1　用户U_i插入他的智能卡进入智能卡读卡器，然后输入用户的身份ID_i和密码PW_i。

Step2　智能卡计算

$$f_k(ID_i) = \sigma_i \oplus H(\alpha \| PW_i) \tag{13}$$

$$\hat{\sigma}_i' = H(f_k(ID_i) \| ID_i) \tag{14}$$

并验证等式$\hat{\sigma}_i \overset{?}{=} \hat{\sigma}_i'$是否成立。如果等式成立，则用户$U_i$允许输入新的密码。否则，更改密码阶段终止。

Step3　输入新的密码PW_i'后，智能卡计算

$$\sigma_i'' = f_k(ID_i) \oplus H(\alpha \| PW_i') \tag{15}$$

$$\bar{\sigma}_i'' = k_i \oplus H(\alpha \| PW_i') \tag{16}$$

并用$(\sigma_i'', \bar{\sigma}_i'')$代替$(\sigma_i, \hat{\sigma}_i)$，智能卡最终包含公共参数$(\sigma_i'', \hat{\sigma}_i, \bar{\sigma}_i'', \alpha, H(\cdot))$。

4.5　密码撤销阶段

当用户注册成功后，就具有更改密码的权限，然而，如果当该用户不想访问某云服务器时，用户需要进行密码撤销，并且该操作不影响用户在其他云的访问权限。为了实现该功能，对于每个云服务器需要进行初始化操作，具体步骤如下所示。

初始化。假设用户U_i和云服务器S_j已成功注册，并完成了登录和相互认证过程，那么云服务器S_j将以合法用户的身份ID构造函数$F_j(x) = (x - ID_1)(x - ID_2)\cdots(x - ID_{i-1})$。

当有新用户身份为ID_i通过认证时，S_j更新函数$F_j(x)$为$F_j'(x) = F_j(x)(x - ID_i)$，即

$$F_j'(x) = (x - ID_1)(x - ID_2)\cdots(x - ID_{i-1})(x - ID_i)$$

当用户想要撤销某个云服务器的密码时，他能在任何时间内实现该操作，具体步骤如下所示。

Step1　用户U_i插入他的智能卡进入智能卡读卡器，然后输入用户的身份ID_i和密码PW_i。

Step2　智能卡计算$f_k(ID_i) = \sigma_i \oplus H(\alpha \| PW_i)$，$\hat{\sigma}_i' = H(f_k(ID_i) \| ID_i)$，并验证等式$\hat{\sigma}_i \overset{?}{=} \hat{\sigma}_i'$是否成立。如果等式成立，则用户$U_i$允许密码撤销。否则，撤销密码阶段终止。

Step3　云服务器S_j计算$F_j(ID_i)$，并验证

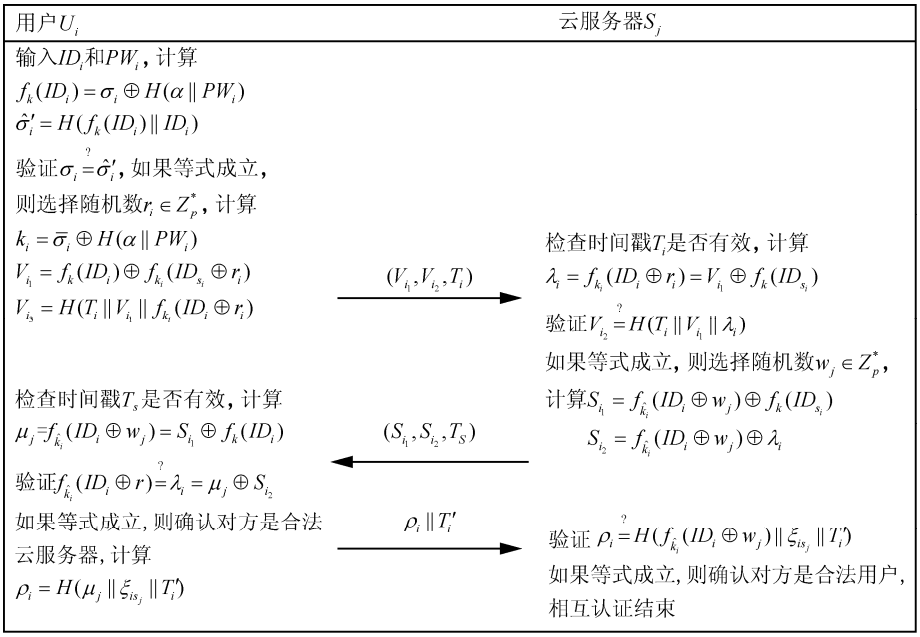

图 3　登录和相互认证过程

$F_j(ID_i)\overset{?}{=}0$ 是否成立。如果等式成立，则用户 U_i 允许密码撤销。否则，撤销密码阶段终止。云服务器 S_j 计算

$$F_j'(x)=\frac{F_j(x)}{(x-ID_i)} \tag{17}$$

并用 $F_j'(x)$ 代替 $F_j(x)$，云服务器 S_j 最终保存 $F_j'(x)$，从而完成了用户在该云服务器的密码撤销功能。

5 安全性分析

1) 双向认证

所提方案能实现云服务器与用户的双向认证。在提出的方案中，每个用户都拥有自己的访问密钥 $(f_k(ID_i),k_i)$，同样每个云服务器也都拥有自己的服务密钥 $(f_k(ID_{s_j}),\hat{k}_j)$。申请服务的用户只需用自己的访问密钥 $(f_k(ID_i),k_i)$ 从云服务器返回的信息 (S_{i1},S_{i2}) 获得 $\lambda_i=f_{k_i}(ID_i\oplus r_i)$，并验证 λ_i 是否是自己发出的身份验证信息 $f_{k_i}(ID_i\oplus r_i)$，就可以确认对方是否是用户申请的合法云服务器 S_j。因为只有合法云服务器才能使用自己的服务密钥 $(f_k(ID_{s_j}),\hat{k}_j)$ 解出秘密验证信息 $f_{k_i}(ID_i\oplus r_i)$，并返回确认信息 (S_{i1},S_{i2})。攻击者或任何人都无法获得服务密钥 $(f_k(ID_{s_j}),\hat{k}_j)$，从而也就无法获得秘密验证信息 $f_{k_i}(ID_i\oplus r_i)$，并生成合法的确认信息 (S_{i1},S_{i2})。同理，云服务器 S_j 只需验证用户返回的信息 ρ_i 和自己生成的确认信息 $f_{\hat{k}_j}(ID_{s_j}\oplus w_j)$ 及会话密钥 ζ_{is_j} 的散列值是否相等就可以确认对方是否为合法的用户。

$$\rho_i=H(f_{\hat{k}_j}(ID_{s_j}\oplus w_j)\|\zeta_{is_j})$$

2) 不需存验证表

在所提方案中，云服务器不需要存储任何的用户信息的验证表，只需存储自己的服务密钥，降低了云服务器的存储费用和管理费用。此外，即使攻击者攻破云服务器，他仍然无法获得任何用户的信息。

3) 抵抗伪造攻击

攻击者或其他用户要伪造一个合法用户 U_i 申请云服务，他必须伪造一个合法的请求信息 (V_{i1},V_{i2},T_i)，否则无法通过云服务器的验证。然而，生成合法的请求信息 (V_{i1},V_{i2},T_i) 需要合法用户 U_i 的访问密钥 $(f_k(ID_i),k_i)$，而攻击者或其他用户无法获得 U_i 的访问密钥 $(f_k(ID_i),k_i)$。因此，他们也就无法伪造出合法的请求信息 (V_{i1},V_{i2},T_i)。同理，攻击者或其他服务器无法获得云服务器 S_j 的服务密钥 $(f_k(ID_{s_j}),\hat{k}_j)$，因此，他们也无法伪造出合法的确认信息 (S_{i1},S_{i2})。此外，从信息 (V_{i1},V_{i2},T_i) 或 (S_{i1},S_{i2}) 也无法计算出访问密钥 $(f_k(ID_i),k_i)$ 和服务密钥 $(f_k(ID_{s_j}),\hat{k}_j)$。因此，所提方案能抵抗伪造攻击。

4) 抵抗重放攻击

在所提方案中，加入时间戳 (T_i,T_s,T_i') 检查，从而能判断是否是重放信息。如果一个攻击者截获用户发给云服务器的请求信息 (V_{i1},V_{i2},T_i)，并修改时间戳，则云服务器通过验证 $V_{i2}=H(T_i\|V_{i1}\|\sigma_i)$ 就可以发现伪造的时间戳。同理，用户也可以通过验证 $H(f_{k_i}(ID_i\oplus r_i)\|T_s)=H((\mu_j\oplus S_{i2})\|T_s)$ 发现伪造的时间戳。因此，该方案可以抵抗重放攻击。

5) 协商会话密钥

在所提方案中，用户和云服务器各自独立计算一个共同的会话密钥

$$\zeta_{is_j}=H(f_{k_i}(ID_i\oplus r_i)\|f_{\hat{k}_j}(ID_{s_j}\oplus w_j)$$

然后，使用该会话密钥来加密之后通信的数据分组，以确保通信是保密的。此外，会话密钥是由随机数和一个单向散列函数生成的。因此，会话密钥在每次通信中都是不同的。且由单向散列函数的性质可知，攻击者很难从被截取的消息中计算出会话密钥。

6) 抵抗智能卡丢失攻击

假设用户的智能卡丢失了或被盗了，攻击者可以从智能卡存储的信息中提取出 $(\sigma_i,\hat{\sigma}_i,\bar{\sigma}_i,\alpha,H(\cdot))$。然后，如果攻击者试图利用这些信息更改用户的密码或登录系统，那么他必须同时知道用户的真实身份 ID_i 和正确的密码 PW_i。如果试图密码猜测攻击，但用户登录智能卡的验证次数会被预先设定（一般设置为 3 次）。如输入密码超过 3 次，就自动锁死。因此，所提方案可以抵抗智能卡攻击，是安全的。

7) 前向保密

前向保密意味着即使云服务器的私有密钥被泄露，也不应该影响先前建立的会话密钥的保密性。在本文所提方案中，用户和云服务器独立地计

算会话密钥

$$\zeta_{is_j} = H(f_{k_i}(ID_i \oplus r_i) \| f_{\hat{k}_j}(ID_{s_j} \oplus w_j)$$

该会话密钥包含 2 个随机数(r_i, w_j)，每次会话过程中的随机数都是不同的。因此，即使云服务器的私钥被泄露也不可能破解用户与云服务器先前的通信信息。

8) 抵抗内部攻击

在注册阶段，用户发送注册信息$(ID_i, H(\alpha \| PW_i))$给可信中心 T。其中，用户的密码并未将明文发送给可信中心，而是通过单向散列函数 $H(.)$进行了保护，使 T不可能知道用户的密码 PW_i和随机数 α，从而使用户不受内部攻击。因此，该方案可以抵抗内部攻击。

9) 抵抗 stolen-verifier 攻击和 modification 攻击

在所提方案中，云服务器只需存储自己的私钥服务密钥($f_k(ID_{s_j})$, $\hat{k}_j$)，不需要存储任何用户信息。因此，攻击者不可能窃取和修改用户的密码，该方案可以抵抗 stolen-verifier 攻击和 modification 攻击。

10) 密码更改的友好性

本文方案允许用户在任何时间更改其密码 PW_i。如果用户想要改变他的密码，首先他需要登录智能卡。当登录成功后，他可以输入新的密码，智能卡将为其重新计算 $\sigma_i'' = f_k(ID_i) \oplus H(\alpha \| PW_i')$，$\bar{\sigma}_i'' = k_i \oplus H(\alpha \| PW_i')$，并用$(\sigma_i'', \bar{\sigma}_i'')$代替$(\sigma_i, \hat{\sigma}_i)$。最后智能卡包含公共参数$(\sigma_i'', \hat{\sigma}_i, \bar{\sigma}_i'', \alpha, H(.))$。由于密码更改阶段是在智能卡上执行，不存在安全漏洞。

11) 安全的密码撤销

本文方案允许用户在任何时间撤销其在某云服务器中的密码 PW_i。如果用户想要撤销他的密码，首先他需要登录智能卡。当登录成功后，云服务器 S_j 需要先验证 $F_j(ID_i) \stackrel{?}{=} 0$ 是否成立。如果成立，说明用户和云服务器完成了相互认证，具备密码撤销的权限，那么云服务器 S_j 执行相应的操作完成密码撤销。如果不成立，则说明该用户没有和云服务器相互认证，不具备密码撤销的权限，那么就无法执行撤销密码的操作。并且用户撤销了某一个云服务器的密码，不影响该用户在其他云服务器中的访问权限。

6　性能分析

本节将从存储费用、是否需要第三方参与认证、通信费用、计算费用等方面分析所提认证方案的性能，并将本文方案和已有方案进行对比，如表 2 所示。其中计算费用主要比较了认证过程中用户和服务器的计算时间。

从存储费用来说，本文方案中用户只存储访问密钥($f_k(ID_i)$, k_i)，不随着注册云服务器增多而增多，且服务器也不需要存储用户的认证信息，只需存储服务密钥($f_k(ID_{s_j})$, $\hat{k}_j$)以及用于实现密码撤销的函数 $F_j(x)$。文献[13~15, 17]只需用户存储常量的秘密信息，所以文献[13~15, 17]和本文方案的存储费用均为 $O(1)$。但文献[13~15]都需要第三方参与认证，而文献[17]和本文方案不需要第三方参与认证。

从通信费用来说，由于第三方参与认证，云服务器在收到用户的请求并经过相关处理后发送给第三方，第三方认证问题再将认证结果和会话秘钥嵌入消息发回给云服务器，云服务器处理结果后提取出共享的密钥发回用户。因此，文献[13~15]都需要产生 4 次通信。而文献[17]和本文方案认证过程

表 2　　**本文方案与已有方案对比**

方案	存储费用	第三方参与认证	通信费用	计算费用	
				用户	云服务器
文献[13]	$O(1)$	是	4 次	$T_{PD} + T_H + 3T_M$	$2T_{PE} + 2T_{PD} + T_H + 4T_M$
文献[14]	$O(1)$	是	4 次	$2T_{PD} + T_H$	$2T_{PE} + T_H$
文献[15]	$O(1)$	是	4 次	$2T_{SE} + 2T_{SD}$	$4T_{SE} + 2T_{SD}$
文献[17]	$O(1)$	否	3 次	$3T_m + 4T_H$	$2T_b + 4T_m + 4T_H$
本文方案	$O(1)$	否	3 次	$2T_f + T_H$	$3T_f + 2T_H$

注：T_H 表示散列函数运算时间，T_{PE} 表示公钥加密运算时间，T_{PD} 表示公钥解密运算时间，T_M 表示乘法运算时间，T_{SE} 表示对称加密运算时间，T_{SD} 表示对称解密运算时间，T_m 表示点乘运算时间，T_b 表示双线性对运算时间，T_f 表示 XOR 同态函数运算时间。认证过程中的异或操作 ⊕ 和连接操作 ‖ 运算开销比较少，已忽略不计。

只需 3 次通信。因此，文献[17]和本文方案具有较低的通信费用，优于文献[13~15]。

从计算费用来说，所提方案利用 XOR 同态函数和散列函数实现了用户和多云之间的认证。没有使用双线性对等公钥密码技术，因此，计算费用非常低。而文献[17]通过使用双线性对进行认证，所以，本文方案比文献[17]具有更低的计算费用。

7　结束语

身份认证作为多云服务的第一道安全保障，显得尤为重要。本文通过智能卡和密码双因素保证用户身份信息，并利用 XOR 同态函数和散列函数生成认证信息，从而有效降低了智能卡和云服务器的存储费用和计算费用。通过安全分析和性能分析，证明了本文方案不仅是安全的，而且计算和存储费用较小，更符合智能卡或移动设备使用的应用场景。

参考文献：

[1] PATI M, RAO G R. Integrity verification in multi-cloud storage using cooperative provable data possession [J]. International Journal of Computer Science and Information Technologies(IJCSIT), 2014, 5 (2):982-985.

[2] CHATURVEDI A, DAS A K, MISHRA D, et al. Design of a secure smart card-based multi-server authentication scheme[J]. Journal of Information Security and Applications, 2016, 30: 64-80.

[3] LI H, DAI Y, TIAN L, et al. Identity-based authentication for cloud computing[M]. Cloud Computing. Springer Berlin Heidelberg, 2009: 157-166.

[4] ZHANG Q, LI Y Z, SONG D J. et al. Alliance-authentication protocol in cloud computing environment[J]. China Communications, 2012, 7: 42-54.

[5] CHEN T H, YEH H L. SHIH W K. An advanced ECC dynamic ID-based remote mutual authentication scheme for cloud computing[C]//5th FTRA International Conference on Multimedia and Ubiquitous Engineering (MUE). 2011: 155-159.

[6] YASSIN A, JIN H, IBRAHIM A, et al. A practical privacy-preserving password authentication scheme for cloud computing[C]//2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum. 2012:21-25.

[7] JAIDHAR D. Enhanced mutual authentication scheme for cloud architecture[C]//2013 IEEE 3rd International Advance Computing Conference (IACC). 2013:70-77.

[8] CHOKSI S. Comparative study on authentication schemes for cloud computing[J]. International Journal of Engineering Development and Research(JIEDR), 2014,2(2): 1-7.

[9] SINGH A, CHATTERJEE K. A secure multi-tier authentication scheme in cloud computing environment[C]//2015 International Conference on Circuit, Power and Computing Technologies (ICCPCT). 2015: 2-8.

[10] MEDHIOUB M, HAMDI M, KIM T. A new authentication scheme for cloud-based storage applications[C]//The 9th International Conference on Security of Information and Networks. 2016: 57-60.

[11] KAUR R, KAUR A. Enhancing authentication schemes for multi-level graphical password in cloud environment, communications on applied electronics (CAE)[C]// Foundation of Computer Science FCS. 2016:12-18.

[12] LEE J, SON J, KIM H, et al. An authentication scheme for providing to user service transparency in multicloud environment[J]. Journal of the Korea Institute of Information Security and Cryptology, 2013, 23(6):1131-1141.

[13] KIM H, CHUNG H, KANG J. Zero-knowledge authentication for secure multi-cloud computing environments[C]//Advances in Computer Science and Ubiquitous Computing. Lecture Notes in Electrical Engineering, 2015: 255-261.

[14] 田俊峰, 孙可辉. 基于 HIBC 的云信任分散统一认证机制[J]. 计算机研究与发展, 2015, 52(7):1660-1671.
TIAN J F, SUN K H. Trusted-distributed-based authentication mechanism using hierarchical identity-based cryptography[J]. Journal of Computer Research and Development, 2015, 52(7):1660-1671.

[15] 周艺华, 蒿金志, 赵航. 混合云服务中的跨云际认证机制[J]. 计算机系统应用, 2015, 24(4):118-122.
ZHOU Y H, HAO J Z, ZHAO H. Authentication mechanism of crossing clouds in hybrid cloud services[J]. Computer Systems and Applications, 2015, 24(4):118-122.

[16] BONG J, SUH Y, SHIN Y. Fast user authentication method considering mobility in multi-clouds[C]//2016 International Conference on Information Networking (ICOIN). 2016:445-448.

[17] TSAD J L, LO N W. A privacy-aware authentication scheme for distributed mobile cloud computing services[J]. IEEE Systems Journal, 2015, 9(3):1-11.

[18] CORON J S, DODIS Y, MALINAUD C, et al. Merkle-Damgård revisited: How to construct a hash function[C]//Advances in Cryptology-CRYPTO 2005. 2005: 430-448.

[19] REN S Q, TAN B H, SUNDARAM S, et al. Secure searching on cloud storage enhanced by homomorphic indexing[J]. Future Generation Computer Systems, 2016, 65: 102-110.

[20] ADE-IBIJOLA O. A simulated enhancement of Fisher-Yates algorithm for shuffling in virtual card games using domain-specific data structures[J]. International Journal of Computer Applications, 2012, 54(11):24-28.

[作者简介]

赵森（1972-），女，黑龙江哈尔滨人，博士，暨南大学讲师，主要研究方向为算法分析与设计、信息安全等。

甘庆晴（1992-），女，江西新余人，暨南大学博士生，主要研究方向为密码学与信息安全。

王晓明（1960-），女，重庆人，博士，暨南大学教授、博士生导师，主要研究方向为网络安全、大数据安全及隐私保护、物联网中的数据安全及隐私保护。

余芳（1976-），女，江西分宜人，博士，暨南大学讲师，主要研究方向为量子计算与量子信息、基于知识的系统、信息安全等。