计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(1):281−287 [doi: 10.15888/j.cnki.csa.008886]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

云存储中无证书的可净化签名方案①

张维鑫, 严翌瑄, 武忆涵, 胡佳烨

(长安大学 信息工程学院, 西安 710064)
通信作者: 张维鑫, E-mail: 547644628@qq.com

摘　要: 在云存储环境中, 数据所有者不仅能够借助云服务器存储数据, 而且可以通过云服务器与其他用户共享数据. 但是, 当数据所有者通过云服务器存储和共享数据时, 可能存在一些安全问题. 首先, 数据所有者需要保证其数据的可认证性. 其次, 数据所有者的数据中可能包含其敏感信息, 比如姓名、年龄等信息. 因此, 数据所有者在与其他用户共享数据时, 可能会泄露自己的敏感信息. 为了解决上述问题, 本文提出了一个无证书的可净化签名方案, 用于解决云存储环境下共享数据的可认证性与敏感信息隐藏. 具体而言, 所提方案基于无证书密码学, 避免了传统公钥基础设施中昂贵的证书管理开销, 消除了基于身份密码学中复杂的密钥托管缺陷. 此外, 所提方案加入了访问控制, 使得存储在云服务器中的数据只能被授权用户访问. 最后, 安全分析说明了所提方案的安全性; 性能分析体现了所提方案的高效性.

关键词: 云存储; 可净化签名; 敏感信息隐藏; 无证书密码学; 访问控制

引用格式: 张维鑫,严翌瑄,武忆涵,胡佳烨.云存储中无证书的可净化签名方案.计算机系统应用,2023,32(1):281–287. http://www.c-s-a.org.cn/1003-3254/8886.html

Certificateless Sanitizable Signature Scheme in Cloud Storage

ZHANG Wei-Xin, YAN Yi-Xuan, WU Yi-Han, HU Jia-Ye

(School of Information Engineering, Chang'an University, Xi'an 710064, China)

Abstract: In the cloud storage environment, data owners can store and share data through cloud servers, but the following security issues may exist. First, data owners need to guarantee the authentication of their data. Secondly, the data may contain the data owner's sensitive information, such as name, age, and other information. Therefore, data owners may reveal their sensitive information when sharing data with other users. To solve the above problems, this study proposes a certificateless sanitizable signature scheme to ensure the authentication of shared data and the sensitive information hiding in cloud storage environments. Specifically, the proposed scheme is based on certificateless cryptography, which avoids the high certificate management overhead in traditional public key infrastructure and eliminates the key escrow defect in identity-based cryptography. In addition, the scheme adds access control, so that the data stored in the cloud server can only be accessed by authorized users. Finally, the security analysis shows the security of the scheme and the performance analysis reflects the efficiency of the scheme.

Key words: cloud storage; sanitizable signatures; sensitive information hiding; certificateless cryptography; access control

近年来, 由于云计算的广泛应用, 使其逐渐成为各行各业的研究热点[1,2]. 云存储作为云计算的重要组成部分, 以其存储成本低、访问方式简单等优势也逐渐受到了人们的青睐. 借助云存储平台, 数据所有者不仅

① 收稿时间: 2022-05-10; 修改时间: 2022-06-15; 采用时间: 2022-06-27; csa 在线出版时间: 2022-08-26
CNKI 网络首发时间: 2022-11-15

© 中国科学院软件研究所 http://www.c-s-a.org.cn

可以通过云服务器存储数据, 还可以通过云服务器与其他用户共享数据[3,4].

在云存储环境中, 当数据所有者将其数据上传到云服务器进行存储和共享时, 可能会存在一些安全问题. 首先, 当云服务器或者共享用户接收到数据时, 需要验证数据的正确性. 其次, 数据所有者的数据中可能包含其敏感信息. 当数据所有者通过云服务器共享数据时, 不希望将自己的敏感信息泄露给云或共享用户. 因此, 如何确保云服务器中数据的可认证性和敏感信息隐藏成为亟待解决的关键问题.

数字签名是保证数据的可认证性、完整性和不可否认性的重要技术. 然而, 数据所有者为了在数据共享的过程中不泄露自己的敏感信息, 需要对敏感信息对应的数据进行适当的修改. 为了保证修改后的数据能够通过云服务器或者共享用户的验证, 数据所有者一般选择将数据先进行加密, 再进行签名. 然后将加密后的数据以及相应签名上传到云服务器[5]. 但是, 如果共享数据是被加密过的, 那么传统的数字签名方案只能保证加密数据的正确性, 而无法保证原始数据的正确性.

为了解决该问题, 2005 年, Ateniese 等人[6] 首先提出了可净化签名的概念. 可净化签名允许数据所有者在对数据进行签名的同时指定一个净化者, 净化者能够修改数据所有者指定的部分数据. 并且净化者能够在不与数据所有者交互的前提下, 为修改后的数据生成有效签名, 使得修改后的数据也能通过云服务器或者共享用户的验证. 因此, 可净化签名能够被用来解决云存储环境下共享数据的可认证性与敏感信息隐藏. 之后, Miyazaki 等人[7] 提出了基于双线性对的可净化签名方案. Agrawal 等人[8] 提出了一个标准模型中强透明度的可净化签名方案. 但是上述可净化签名方案效率低下. 考虑到这个问题, 2013 年, 文献 [9] 利用 Waters 签名技术, 提出了一个标准模型中高效的基于身份可净化签名方案. 2014 年, 文献 [10] 提出了一个基于属性的可净化签名方案. 2016 年, 文献 [11] 提出了一个没有随机谕言机的高效的可净化签名. 然而, 以上可净化签名方案功能单一, 并不适用于实际应用场景. 为了解决该问题, 2017 年, 文献 [12] 提出了一种支持树形访问结构的属性基可净化签名方案. 2019 年, 文献 [13] 提出了一个高效的不可链接的可净化签名. 2020 年, 文献 [14] 提出了一种基于环签名和短签名的可净化签名方案. 文献 [15] 将可净化签名引入到了智能医疗场景中, 提出了一个基于可净化签名的智能移动医疗场景隐私保护方案. 文献 [16] 使用 Affine 消息鉴别码提出了一个基于身份的陷门可净化签名. 此外, 为了保证共享数据的可认证性、完整性以及敏感信息隐藏, 文献 [17] 和文献 [18] 分别提出了一种基于可净化签名的云存储数据完整性审计方案.

但是, 现有的大部分可净化签名方案有以下两个问题: 很多方案依赖于传统公钥基础设施或者基于身份密码学, 这使得大多方案存在昂贵的证书管理开销或者复杂的密钥托管缺陷; 此外, 现有方案都没有考虑数据访问控制的问题, 如果数据所有者的数据存储在云服务器后, 能被任何用户访问, 那么可能会造成数据滥用的问题.

针对上述问题, 本文提出了一个云存储中无证书的可净化签名方案. 具体而言, 本文的贡献如下.

(1) 由于无证书密码学[19] 固有的结构优势, 所提方案摆脱了传统公钥基础设施中昂贵的证书管理开销、消除了基于身份密码学中复杂的密钥托管缺陷.

(2) 所提方案实现了云存储环境下数据的可认证性和敏感信息隐藏. 此外, 所提方案加入了访问控制, 使得存储在云服务器中的数据只能被授权用户访问.

(3) 安全分析表明了所提方案的安全性; 性能分析表明, 与其他相关方案相比, 所提方案的计算代价和通信代价是可接受的.

1 背景知识

1.1 双线性映射

设q是一个大素数, $\mathbb{G}_1$是阶数为q的加法循环群, $\mathbb{G}_2$是阶数为q的乘法循环群. $\mathbb{G}_1$到$\mathbb{G}_2$的双线性映射$e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$满足如下性质.

(1) 双线性: 如果对任意的$X,Y\in\mathbb{G}_1$和$a,b\in\mathbb{Z}_q^*$, 等式$e(a\cdot X,b\cdot Y)=e(X,Y)^{a\cdot b}$成立, 那么就称该映射为双线性映射.

(2) 非退化性: 映射不把$\mathbb{G}_1\times\mathbb{G}_1$中的所有元素映射到$\mathbb{G}_2$中的单位元. 即对于元素$X,Y\in\mathbb{G}_1$, 满足$e(X,Y)\neq 1_{\mathbb{G}_2}$.

(3) 可计算性: 对于任意的$X,Y\in\mathbb{G}_1$, 存在一个有效算法计算$e(X,Y)$.

1.2 困难问题和安全假设

(1) 离散对数问题 (DL 问题): 假设$\mathbb{G}_1$是一个加法循环群, P是$\mathbb{G}_1$的生成元. 对于未知的$x\in\mathbb{Z}_q^*$, 给定元组$(P,\ x\cdot P\in\mathbb{G}_1)$, DL 问题是计算$x\in\mathbb{Z}_q^*$.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(2) 离散对数假设 (DL 假设): 对任意的概率多项式时间敌手$\mathcal{A}$, $\mathcal{A}$能够解决 DL 问题的概率是可忽略的.

(3) 计算性 Diffie-Hellman 问题 (CDH 问题): 假设$\mathbb{G}_1$是一个加法循环群, P是$\mathbb{G}_1$的生成元. 对于未知的$a,b\in\mathbb{Z}_q^*$, 给定元组$(P, a\cdot P, b\cdot P\in\mathbb{G}_1)$, CDH 问题是计算$a\cdot b\cdot P$.

(4) 计算性 Diffie-Hellman 假设 (CDH 假设): 对任意的概率多项式时间敌手$\mathcal{A}$, $\mathcal{A}$能够解决 CDH 问题的概率是可忽略的.

1.3 系统模型

本文的系统模型如图 1 所示, 其中包括 5 个实体: 密钥生成中心 (KGC)、数据所有者 (DO)、净化者 (Sanitizer)、云服务器 (CS) 以及共享用户 (User).

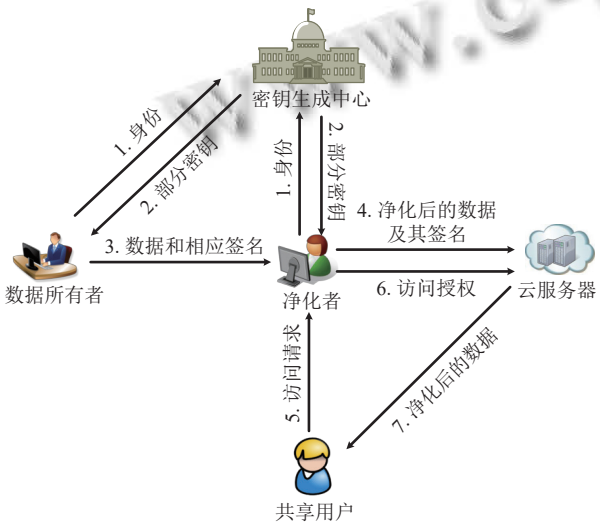

图 1 系统模型

(1) 密钥生成中心 (KGC): 一个可信实体, 具有强大的计算能力, 负责整个系统的初始化, 为系统生成主密钥和公开参数. 此外, 在接收到 DO 和 Sanitizer 的身份后, 为 DO 和 Sanitizer 生成相应的部分密钥.

(2) 数据所有者 (DO): 一个可信实体, 有大量的数据需要存储在云服务器. 此外, DO 会将其数据和相应签名发送给 Sanitizer, 并指定数据中需要净化的敏感信息, 让净化者执行净化.

(3) 净化者 (Sanitizer): 一个半可信实体, 负责对 DO 数据中的敏感信息进行净化, 并为净化后的数据生成有效签名. 然后, 将净化后的数据和相应签名发送给 CS 进行存储和共享.

(4) 云服务器 (CS): 一个半可信实体, 具有丰富的存储空间和强大的计算能力, 负责对净化后的数据进行存储和维护. 此外, 在接收到 Sanitizer 的访问授权时, 验证授权的正确性, 并将净化后的数据发送给 User.

(5) 共享用户 (User): 它主要是指需要访问数据所有者数据的用户.

在该模型中, DO 和 Sanitizer 首先分别将其身份发送给 KGC 进行注册, KGC 为 DO 和 Sanitizer 分发部分密钥. 然后, DO 为原始数据生成相应的签名, 并指定原始数据中需要净化的敏感信息数据块. 最后, DO 将原始数据、敏感信息对应的数据块索引以及相应签名发送给 Sanitizer. Sanitizer 接收到数据后, 对敏感信息对应的数据块执行净化, 并将净化后的数据和相应签名发送到 CS 进行存储和共享. 然后, CS 验证数据和签名的正确性. 当 User 需要使用数据时, 会向发送访问请求. Sanitizer 收到请求后为用户生成访问授权, 并将其发送给 CS. 访问授权通过 CS 的验证后, CS 会将净化后的数据发送给 User.

1.4 安全需求

(1) 正确性: 当接收到 Sanitizer 发送的数据和相应签名时, CS 应确保数据和相应签名的正确性. 此外, 当接收到 Sanitizer 生成的访问授权时, CS 需确保访问授权的正确性.

(2) 指定 Sanitizer: 只有被 DO 指定的 Sanitizer 才能够执行净化.

(3) 不变性: 除了 DO 指定的敏感信息数据块外, Sanitizer 不能对其他数据生成有效签名.

(4) 隐私性: DO 数据中的敏感信息不应暴露给 CS 和 User.

(5) 访问控制: 只有 User 才能够访问云服务器中的数据.

(6) 消除密钥托管: DO 和 Sanitizer 的私钥不需要由 KGC 进行托管.

2 本文方案

方案由 6 个算法组成, 包括系统初始化算法, 密钥生成算法, 签名生成算法, 净化算法, 验证算法和访问算法.

2.1 系统初始化算法

该算法由 KGC 执行, 输入安全参数k, KGC 执行如下过程生成系统参数$params$和主密钥msk.

(1) KGC 选择阶数同为q的一个加法循环群$\mathbb{G}_1$以

© 中国科学院软件研究所 http://www.c-s-a.org.cn

及一个乘法循环群$\mathbb{G}_2$, 其中q为一个大素数. 令P为$\mathbb{G}_1$的生成元. 然后选择双线性映射$e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$.

(2) KGC 选择$\lambda\in\mathbb{Z}_q^*$作为主密钥, 即$msk=\lambda$. 并计算系统公钥$P_{\text{pub}}=\lambda\cdot P$.

(3) KGC 选择 3 个安全的哈希函数$H_1:\{0,1\}^*\to\mathbb{Z}_q^*$、$H_2:\{0,1\}^*\to\mathbb{Z}_q^*$、$H_3:\{0,1\}^*\to\mathbb{G}_1$.

KGC 保存主密钥$msk=\lambda$, 公开系统参数$params=\{q,\mathbb{G}_1,\mathbb{G}_2,e,P,H_1,H_2,H_3,P_{\text{pub}}\}$.

2.2 密钥生成算法

该算法由 KGC, DO 和 Sanitizer 执行. 输入主密钥$msk=\lambda$, 系统参数$params$, KGC、DO 和 Sanitizer 执行如下过程.

(1) 当接收到 DO 的身份ID_o后, KGC 随机选择$y_o\in\mathbb{Z}_q^*$, 并计算$Y_o=y_o\cdot P$, $h_o=H_1(ID_o\|Y_o)$, $d_o=y_o+h_o\cdot\lambda$. KGC 将(Y_o,d_o)作为 DO 的部分密钥, 并通过安全信道发送给 DO.

(2) 当接收到 Sanitizer 的身份ID_s后, KGC 随机选择$y_s\in\mathbb{Z}_q^*$, 并计算$Y_s=y_s\cdot P$, $h_s=H_1(ID_s\|Y_s)$, $d_s=y_s+h_s\cdot\lambda$. KGC 将(Y_s,d_s)作为 Sanitizer 的部分密钥, 并通过安全信道发送给 Sanitizer.

(3) DO 随机选择$x_o\in\mathbb{Z}_q^*$作为自己的秘密值, 并计算$X_o=x_o\cdot P$. 然后 DO 设置私钥为$sk_o=(d_o,x_o)$, 相应的公钥为$pk_o=(Y_o,X_o)$.

(4) Sanitizer 随机选择$x_s\in\mathbb{Z}_q^*$作为自己的秘密值, 并计算$X_s=x_s\cdot P$. 然后 Sanitizer 设置私钥为$sk_s=(d_s,x_s)$, 相应的公钥为$pk_s=(Y_s,X_s)$.

2.3 签名生成算法

该算法由 DO 执行, 输入 DO 的原始数据M、私钥$sk_o=(d_o,x_o)$, DO 执行如下过程为原始数据生成相应签名.

(1) DO 将原始数据M分为n个数据块, 即$M=\{m_1,m_2,\cdots,m_n\}$, 其中$m_i\in\mathbb{Z}_q^*$, $i\in[1,n]$. 并选择$name\in\mathbb{Z}_q^*$作为M的标识符. 假设数据M中需要净化的敏感信息数据块的集合为K, 即$K\subseteq[1,n]$.

(2) 对于原始数据M中每个数据块$m_i\in\mathbb{Z}_q^*$ $(i\in[1,n])$, DO 随机选择$r_i\in\mathbb{Z}_q^*$, 并计算$R_i=r_i\cdot P$, $\alpha=H_2(ID_o\|X_o\|Y_o)$, $v_i=H_3(name\|R_i)$.

(3) DO 为不需要净化的数据块计算签名$\sigma_i=d_o\cdot v_i+x_o\cdot m_i\cdot\alpha\cdot R_i$ $(i\in[1,n]$且$i\notin K)$.

(4) DO 为需要净化的敏感信息数据块计算签名$\sigma_i=d_o\cdot v_i+x_o\cdot m_i\cdot\alpha\cdot r_i\cdot X_s$ $(i\in K)$, 并计算转换值$\beta_i=x_o\cdot R_i$ $(i\in K)$. 一般而言, 一份数据中敏感信息的数量并不多, 因此, DO 只需要为某几个需要净化的数据块单独计算签名即可.

(5) DO 将$(M,K,\{\sigma_i\}_{i\in[1,n]},\{R_i\}_{i\in[1,n]},\{\beta_i\}_{i\in K})$发送给 Sanitizer, 并在本地删除这些信息.

2.4 净化算法

该算法由 Sanitizer 执行. 当接收到 DO 发送信息后, Sanitizer 执行如下过程对数据M中的敏感信息数据块执行净化.

(1) Sanitizer 首先对集合K中的数据块执行净化, 通常, Sanitizer 使用通配符替换敏感信息所对应的数据块, 生成净化后的数据$M'=\{m_1',m_2',\cdots,m_n'\}$. 其中当且仅当$i\in[1,n]$且$i\notin K$时$m_i=m_i'$, 否则$m_i\neq m_i'$.

(2) Sanitizer 通过如下方式将集合K中的数据块所对应的签名转换为经过净化后的数据块的有效签名:

$$\begin{aligned}\sigma_i'&=\begin{cases}\sigma_i-m_i\cdot\alpha\cdot x_s\cdot\beta_i+m_i'\cdot\alpha\cdot\beta_i, & i\in K\\ \sigma_i, & i\in[1,n]\text{且}i\notin K\end{cases}\\&=d_o\cdot v_i+x_o\cdot m_i'\cdot\alpha\cdot R_i\end{aligned}\tag{1}$$

(3) Sanitizer 将$(M',\{\sigma_i'\}_{i\in[1,n]},\{R_i\}_{i\in[1,n]})$发送到 CS 进行存储和维护, 并在本地删除这些信息.

2.5 验证算法

当接收到 Sanitizer 发送的数据和相应签名后, CS 执行如下过程进行验证.

(1) CS 计算$h_o=H_1(ID_o\|Y_o)$, $\alpha=H_2(ID_o\|X_o\|Y_o)$, $v_i=H_3(name\|R_i)$, $\sigma=\sum_{i=1}^{n}\sigma_i'$, $\mu=\sum_{i=1}^{n}m_i'\cdot R_i$.

(2) CS 通过式 (2) 是否成立来判断接收到的数据和签名是否正确:

$$e(\sigma,P)=e\left(\sum_{i=1}^{n}v_i,Y_o+h_o\cdot P_{\text{pub}}\right)\cdot e(\alpha\cdot\mu,X_o)\tag{2}$$

如果式 (2) 成立, CS 存储数据和相应签名; 否则, CS 拒绝存储数据和相应签名.

2.6 访问算法

该算法由 User、Sanitizer 和 CS 执行, 具体过程如下.

(1) User 利用其身份ID_u和 DO 数据的标识符$name$生成访问请求$\omega=H_3(ID_u\|name)$, 并将ω发送给 Sanitizer.

(2) Sanitizer 接收到访问请求ω后, 计算$\delta_\omega=d_s\cdot\omega$, 为 User 生成访问授权$(\omega,\delta_\omega)$, 并将$(\omega,\delta_\omega)$发送给 CS.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(3) CS 接收到访问授权(ω,δ_ω)后, 计算$h_s = H_1(ID_s \| Y_s)$, 然后 CS 通过式 (3) 验证授权的正确性:

$$e(\delta_\omega, P) = e(\omega, Y_s + h_s \cdot P_{\text{pub}}) \tag{3}$$

如果式 (3) 成立, CS 将净化后的数据M'发送给 User.

3 安全分析与性能评价

3.1 安全需求分析

本节对第 1.4 节中的安全需求进行分析.

(1) 正确性: CS 根据式 (2) 是否成立来判断接收到的数据和签名是否正确; 此外, CS 根据式 (3) 是否成立来判断接收到的访问授权是否正确.

式 (2) 的正确性:

$$\begin{aligned} e(\sigma, P) &= e\left(\sum_{i=1}^{n} \sigma'_i, P\right) \\ &= e\left(\sum_{i=1}^{n} \left(d_o \cdot v_i + x_o \cdot m'_i \cdot \alpha \cdot R_i\right), P\right) \\ &= e\left(\sum_{i=1}^{n} d_o \cdot v_i, P\right) \cdot e\left(\sum_{i=1}^{n} x_o \cdot m'_i \cdot \alpha \cdot R_i, P\right) \\ &= e\left(\sum_{i=1}^{n} v_i, d_o \cdot P\right) \cdot e\left(\alpha \cdot \sum_{i=1}^{n} m'_i \cdot R_i, x_o \cdot P\right) \\ &= e\left(\sum_{i=1}^{n} v_i, (y_o + h_o \cdot \lambda) \cdot P\right) \cdot e(\alpha \cdot \mu, X_o) \\ &= e\left(\sum_{i=1}^{n} v_i, Y_o + h_o \cdot P_{\text{pub}}\right) \cdot e(\alpha \cdot \mu, X_o) \end{aligned}$$

式 (3) 的正确性:

$$\begin{aligned} e(\delta_\omega, P) &= e(d_s \cdot \omega, P) = e(\omega, d_s \cdot P) \\ &= e(\omega, (y_s + h_s \cdot \lambda) \cdot P) \\ &= e(\omega, Y_s + h_s \cdot P_{\text{pub}}) \end{aligned}$$

(2) 指定 Sanitizer: 在本方案中, 需要净化的数据块的签名为$\sigma_i = d_o \cdot v_i + x_o \cdot m_i \cdot \alpha \cdot r_i \cdot X_s\ (i \in K)$, 其中$X_s$表示 Sanitizer 的公钥. 根据 Diffie-Hellman 协议, 只有使用 Sanitizer 的私钥x_s才能执行净化. 因此, 在本方案只有 Sanitizer 才能执行净化, 其他任何敌手无法执行净化, 本文方案满足指定 Sanitizer 的安全需求.

(3) 不变性: 在本方案中, 不允许净化的数据块的签名为$\sigma_i = d_o \cdot v_i + x_o \cdot m_i \cdot \alpha \cdot R_i\ (i \in [1,n]且 i \notin K)$. 如果 Sanitizer 想要净化这一部分数据块, 就需要 DO 的私钥x_o, 而x_o存在于$X_o = x_o \cdot P$, 这可以被看作是 DL 问题. 根据 DL 问题, Sanitizer 无法获得x_o. 因此, Sanitizer 无法对这部分数据块执行净化, 本文方案满足不变性.

(4) 隐私性: Sanitizer 会使用通配符替换敏感信息对应的数据块, 因此, CS 和 User 不能获得 DO 的敏感信息, 本文方案满足隐私性.

(5) 访问控制: Sanitizer 使用自己的私钥为 User 生成访问授权, 因此, 只有 User 才能访问云服务器中的数据.

(6) 消除密钥托管: 由于无证书密码学固有的结构优势. 在本文方案中 DO 和 Sanitizer 的私钥不需要由 KGC 进行托管.

3.2 功能比较

本节将所提方案和文献 [17,18] 进行功能对比. 在比较过程中只考虑文献 [17,18] 中和可净化签名相关的部分. 功能比较如表 1 所示, 其中F1表示正确性; F2表示指定 Sanitizer; F3表示不变性; F4表示隐私性; F5表示访问控制; F6表示消除密钥托管. 从表 1 中可以看出, 文献 [17] 不满足指定 Sanitizer 和访问控制, 并且文献 [17,18] 都不满足不变性. 此外, 文献 [17,18] 是基于身份下的可净化签名, 故都存在密钥托管的缺陷. 因此, 只有本文方案满足第 1.4 节中的所有安全需求.

表 1 功能对比

功能	文献[17]	文献[18]	所提方案
F1	√	√	√
F2	×	√	√
F3	×	×	√
F4	√	√	√
F5	×	√	√
F6	×	×	√

注: “√”表示满足, “×”表示不满足.

3.3 计算代价比较

为了确保公平比较, 应采用相同的 80 bit 安全性. 由于文献 [17,18] 和本文方案都使用了双线性映射, 故选择双线性映射$e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, 群$\mathbb{G}_1$的阶数为$q$, q是 512 bit 的素数. 基于 MIRACL Crypto SDK[20] 得到密码学操作的执行时间并列举在表 2 中, 其硬件设备为 2.90 GHz i5 CPU 和 16 GB 内存的台式电脑, 操作系统是64位Windows 10. 表 3 给出了文献 [17,18] 和所提方案的计算代价比较, 其中n代表数据块的数量, K代表需要净化的数据块的数量.

图 2–图 4 分别表示签名生成算法, 净化算法和验证算法的计算代价比较. 由图 2 和图 3 可以看出, 所提

© 中国科学院软件研究所 http://www.c-s-a.org.cn

方案在签名生成算法和净化算法中的计算代价优于文献 [18], 而与文献 [17] 基本相同. 虽然所提方案的计算代价与文献 [17] 基本相同, 但是文献 [17] 的功能性较差, 它不满足指定 Sanitizer、不变性、访问控制以及消除密钥托管, 而所提方案满足所有的安全需求. 因此, 整体而言, 所提方案优于文献 [17]. 由图 4 可知, 在验证算法中, 随着数据块数量的增加, 所提方案的计算代价明显优于文献 [17,18]. 更进一步, 根据总体的计算代价而言, 所提方案优于文献 [17,18]. 综上所述, 所提方案更加适用于解决云存储环境中共享数据的可认证性与敏感信息隐藏.

表 2 密码学操作执行时间

符号	描述	执行时间 (ms)
T_P	双线性映射操作	1.6678
T_H	映射到点的哈希	1.5586
T_A	$\mathbb{G}_1$下的加法运算	0.0051
T_M	$\mathbb{G}_1$下的乘法运算	0.6126
$T_{M'}$	$\mathbb{G}_2$下的乘法运算	0.0016

注: 一般哈希的执行时间和$\mathbb{Z}_q^*$中的运算已经被忽略.

表 3 计算代价比较

方案	签名生成算法	净化算法	验证算法
文献[17]	$n(T_H+2T_A+2T_M)=2.794n$	$K(T_A+T_M)=0.6177K$	$(4T_p+2T_{M'}+3T_M+2T_A+T_H)n=10.6936n$
文献[18]	$n(2T_H+2T_A+3T_M)=4.9652n$	$K(T_A+2T_M)=1.2303K$	$(n+2)T_p+nT_{M'}+2T_M+(2n-1)T_A+nT_H=3.2382n+4.5557$
本文方案	$n(T_H+T_A+2T_M)=2.7889n$	$K(T_A+T_M)=0.6177K$	$3T_p+T_{M'}+nT_H+nT_A+2T_M=1.5637n+6.2302$

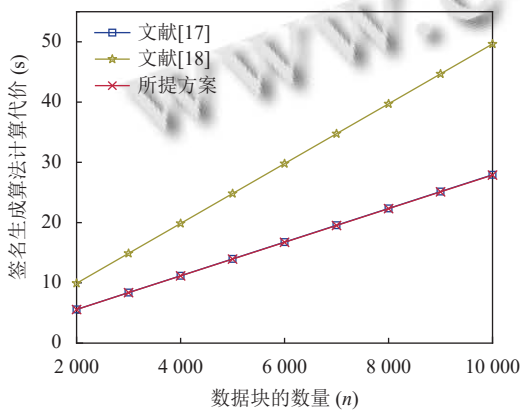

图 2 签名生成算法计算代价比较

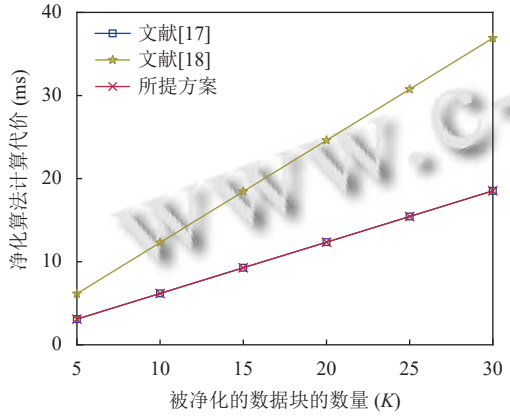

图 3 净化算法计算代价比较

3.4 通信代价比较

为了更实际的比较通信代价, 本文设定如下: $|\mathbb{G}_1|$, $|\mathbb{Z}_q^*|$ 分别为 512 bit 和 160 bit. 表 4 比较了文献 [17,18] 与所提方案的通信代价. 在通信代价的比较中, 我们主要对每个方案的主密钥长度, 签名长度, 净化签名长度以及访问授权长度进行比较.

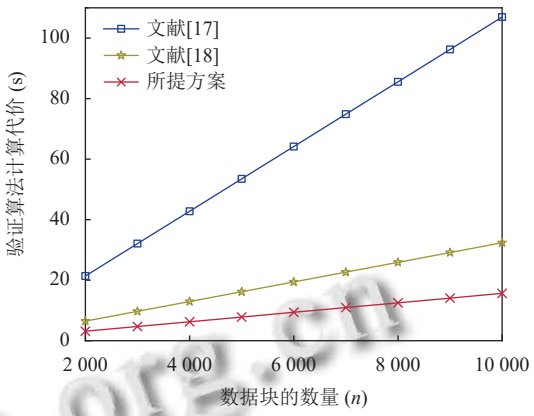

图 4 验证算法计算代价比较

表 4 通信代价比较 (bit)

长度	文献[17]	文献[18]	本文方案
主密钥	512	160	160
签名	512	512	512
净化签名	512	512	512
访问授权	—	672	1 024

由表 4 可以看出, 首先所提方案的主密钥长度少于文献 [17], 而与文献 [18] 相同. 其次, 所提方案与文献 [17,18] 具有相同的签名长度和净化签名长度. 最后, 文献 [18] 的访问授权长度少于所提方案. 据此可以看出所提方案的整体通信代价优于文献 [17], 但是略高于文献 [18]. 由于所提方案在功能和计算代价方面优于文献 [18], 故所提方案多出的通信代价是可接受的.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

4 结论

为了解决现有可净化签名方案的问题, 本文提出了一个云存储中无证书的可净化签名方案, 能够使得数据所有者在隐藏其敏感信息的情况下, 通过云服务器与其他用户共享数据. 本文方案解决了基于身份可净化签名方案中的密钥托管问题; 并且所提方案加入了访问控制, 使得只有授权用户才能访问云服务器中的数据. 此外, 与现有方案相比, 本文方案能够满足所提的所有安全需求; 并且本文所提方案的计算代价优于现有方案. 因此, 本文所提方案更加适用于解决云存储中共享数据的可认证性与敏感信息隐藏问题.

参考文献

1 Jung HS, Yoon CS, Lee YW, *et al*. Processing IoT data with cloud computing for smart cities. International Journal of Web Applications, 2017, 9(3): 88–95.

2 张立强, 吕建荣, 严飞, 等. 可信云计算研究综述. 郑州大学学报 (理学版), 2022, 54(4): 1–11. [doi: 10.13705/j.issn.1671-6841.2021487]

3 Quick D, Martini B, Choo R. Cloud Storage Forensics. Boston: Syngress Publishing, 2013.

4 Ge CP, Susilo W, Liu Z, *et al*. Secure keyword search and data sharing mechanism for cloud computing. IEEE Transactions on Dependable and Secure Computing, 2021, 18(6): 2787–2800. [doi: 10.1109/TDSC.2020.2963978]

5 牛淑芬, 宋蜜, 方丽芝, 等. 智慧医疗中基于属性加密的云存储数据共享. 电子与信息学报, 2022, 44(1): 107–117. [doi: 10.11999/JEIT210858]

6 Ateniese G, Chou DH, De Medeiros B, *et al*. Sanitizable signatures. Proceedings of the 10th European Symposium on Research in Computer Security. Milan: Springer, 2005. 159–177.

7 Miyazaki K, Hanaoka G, Imai H. Digitally signed document sanitizing scheme based on bilinear maps. Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security. Taipei: ACM, 2005. 343–354.

8 Agrawal S, Kumar S, Shareef A, *et al*. Sanitizable signatures with strong transparency in the standard model. Proceedings of the 5th International Conference on Information Security and Cryptology. Beijing: Springer, 2010. 93–107.

9 明洋, 李瑞. 标准模型下高效的基于身份可净化签名方案. 计算机科学, 2013, 40(5): 158–163. [doi: 10.3969/j.issn.1002-137X.2013.05.039]

10 刘西蒙, 马建峰, 熊金波, 等. 云计算环境下基于属性的可净化签名方案. 电子与信息学报, 2014, 36(7): 1749–1754.

11 Lai RWF, Zhang T, Chow SSM, *et al*. Efficient sanitizable signatures without random oracles. Proceedings of the 21st European Symposium on Research in Computer Security. Heraklion: Springer, 2016. 363–380.

12 莫若, 马建峰, 刘西蒙, 等. 一种支持树形访问结构的属性基可净化签名方案. 电子学报, 2017, 45(11): 2715–2720. [doi: 10.3969/j.issn.0372-2112.2017.11.019]

13 Bultel X, Lafourcade P, Lai RWF, *et al*. Efficient invisible and unlinkable sanitizable signatures. Proceedings of the 22nd IACR International Workshop on Public Key Cryptography. Beijing: Springer, 2019. 159–189.

14 张君何, 周清雷, 韩英杰. 一种基于环签名和短签名的可净化签名方案. 计算机科学, 2020, 47(S1): 386–390, 399.

15 Xu ZY, Luo M, Kumar N, *et al*. Privacy-protection scheme based on sanitizable signature for smart mobile medical scenarios. Wireless Communications and Mobile Computing, 2020, 2020: 8877405.

16 Ishizaka M, Kiyomoto S. Downgradable identity-based signatures and trapdoor sanitizable signatures from downgradable affine MACs. IACR Cryptology ePrint Archive, 2021: 1170.

17 Shen WT, Qin J, Yu J, *et al*. Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. IEEE Transactions on Information Forensics and Security, 2019, 14(2): 331–346. [doi: 10.1109/TIFS.2018.2850312]

18 Xu Y, Ding L, Cui J, *et al*. PP-CSA: A privacy-preserving cloud storage auditing scheme for data Sharing. IEEE Systems Journal, 2021, 15(3): 3730–3739. [doi: 10.1109/JSYST.2020.3018692]

19 Al-Riyami SS, Paterson KG. Certificateless public key cryptography. International Conference on the Theory and Application of Cryptology and Information Security. Taipei: Springer, 2003. 452–473.

20 Shamus S. Multi precision integer and rational arithmetic cryptographic library (MIRACL). https://miracl.com/blog/. (2021-01-10).

(校对责编: 牛欣悦)

© 中国科学院软件研究所 http://www.c-s-a.org.cn