公开可审计的可修订签名方案

马金花 黄欣沂* 许俊鹏 伍 玮
(福建师范大学数学与信息学院 福州 350007)

摘 要：具有可审计性的可修订签名方案(RSS)允许修订人在不与原始签名人交互的情况下删除已签名数据的部分内容，并为余下的数据生成有效签名，支持对数据发布者身份的追溯，为可修订签名面临的恶意修订问题提供了有效的解决方法。该文首先提出一个新颖的公开可审计的可修订签名方案(PA-RSS)的通用构造，并形式化定义相应的安全模型。利用传统数字签名方案，提出首个PA-RSS的具体设计，可将公开可审计性加入到任意不具有可审计性的可修订签名方案中。并证明该设计具有不可伪造性、隐私性、签名人的公开可审计性以及修订人的公开可审计性。与现有的公开可审计的可修订签名方案相比，该文方案的计算效率更高、通信开销更少，更适用于实现开放共享环境中公开可审计的认证数据修订。

关键词：数字签名；可修订签名；可审计性；认证数据
中图分类号：TN918; TP309 **文献标识码**：A **文章编号**：1009-5896(2020)05-1079-08
DOI: 10.11999/JEIT190836

Public Accountable Redactable Signature Scheme

MA Jinhua HUANG Xinyi XU Junpeng WU Wei
(*Institute of Mathematics and Informatics*, *Fujian Normal University*, *Fuzhou* 350007, *China*)

Abstract: Redactable Signature Scheme (RSS) with accountability allows a redactor to delete some portions of the signed data, and generates a valid signature for the remained data without any interaction with the original signer. It supports to trace the data producer, and is an effective solution to the malicious redaction problem of RSS. A novel design of Public Accountable Redactable Signature Scheme (PA-RSS) is proposed, and its security model is formally defined. The first concrete design of PA-RSS is presented by using the traditional digital signature scheme, which can add public accountability to any RSS without accountability. Its unforgeability, privacy, signer's public accountability, and redactor's public accountability are proved. Compared with the existing public accountable RSS, the presented scheme with less communication cost is more efficient, and much more applicable to realize the public accountability of authenticated data redaction in an open and sharable environment.

Key words: Digital signature; Redactable signature; Accountability; Authenticate data

1 引言

随着云计算、大数据、物联网、区块链等信息技术的快速发展，全球数据量呈指数级增长，大大加速了工业创新和社会变革的进程。同人力资源和物质资本一起，数据已成为国家核心的战略资源。数据大大推动了医疗、教育、金融等领域中各类数据驱动型应用的发展与变革，数据的安全问题已成为影响各行各业数字化、智能化稳健发展的关键性问题。保证数据在收集、处理、使用等过程中的真实性是数据安全保护的重要环节。

数字签名可验证数据内容的完整性和数据源的真实性，是保障数据安全的核心技术之一。其传统安全要求为在自适应选择消息攻击下满足存在不可伪造性(Existential UnForgeability against adaptive Chosen-Message Attacks, EUF-CMA)[1]。具有EUF-CMA的数字签名使得概率多项式时间(Probabilistic Polynomial Time, PPT)的攻击者在没有签名人私钥的情况下不能为新数据生成有效的签名。有效的数字签名使得接收者相信其收到的数据没有被篡改。

为了实现数据的隐私保护，在发布数据之前，通常需要对已签名数据进行修改。比如，当医疗健

收稿日期：2019-10-29；改回日期：2020-02-08；网络出版：2020-03-19
*通信作者： 黄欣沂 xyhuang81@yahoo.com
基金项目：国家自然科学基金(61822202, 61872089, 61872090)
Foundation Items: The National Natural Science Foundation of China (61822202, 61872089, 61872090)

康数据被用于科学研究时，真实姓名和身份证号码等敏感的个人信息应当被删除。因此，数据修订对于实现数据的隐私保护是必要的。然而，数据修订与传统数字签名的EUF-CMA安全要求在本质上是相互冲突的。具有EUF-CMA的数字签名禁止已签名数据被做任何形式的修改，即使是极其微小的变动也会导致原始签名失效，并且修改者无法为微小变动后的数据生成有效的数字签名，为数据的真实性和完整性提供了强有力的保护。虽然具有EUF-CMA的传统数字签名能够满足数据认证的基本需求，但也阻碍了对已签名数据的合理修改操作。

可修订数字签名(Redactable Signatures)是一类支持删除操作的具有同态性质的数字签名，其概念是由Johnson等人[2]于2002年正式提出，实际上Steinfeld等人[3]于2001年就已经提出了具有类似功能的内容可截取数字签名(Content Extraction Signature, CES)。可修订数字签名允许修订人在不与原始签名人交互的情况下删除已签名数据的部分内容，并为修改后的数据生成有效的数字签名。其为解决数据修订与数字签名传统安全要求之间的冲突提供了有效的解决方法，在电子健康记录系统、社会网络、智能电网等场景中具有广泛的应用价值。

自2001年被提出以来，可修订数字签名已取得了一批有意义的研究成果。然而，大多数现有的可修订签名方案(Redactable Signature Schemes, RSSs)没有考虑恶意修订问题，任何人利用公共信息都可以任意地删除已签名数据的部分内容，并为修改后的数据生成有效签名。恶意的修订人可能滥用此功能，恶意地修改数据，导致原始数据信息被篡改。文献[3–9]通过设定修订控制规则限制修订人可修订的内容，修订人只可以为符合修订规则的修订后数据生成有效的签名，目前在该方面已取得了一批有意义的研究成果。还可以通过限制执行修订操作的人员，对修订人员进行追溯审计，通过签名的可审计性实现责任问权，控制滥用修订功能的现象。

虽然文献[10–13]较早讨论了具有可审计性的可修订签名方案的设计，但均只是将其作为一个开放式的问题提出来，并没有进一步地展开研究。直到2015年，Pöhls等人[14]才对这一问题进行了正式的研究，并将该类签名命名为可审计的可修订签名(Accountable Redactable Signature, ARS)。当发生数据责任纠纷时，ARS允许任意第三方利用证据标签识别出一个有效数据/签名对的生成人，且该生成人无法抵赖自己的责任。ARS的可审计性迫使修订人不敢恶意修订数据，为RSS面临的恶意修订问题提供了有效的解决方法。

文献[14]给出了ARS方案的模型以及安全性模型的形式化定义，并给出了一个具体构造。该构造利用净化签名方案(Sanitizable Signature Scheme, SSS)[15]可将任意的RSS转化为ARS，转化后的ARS继承了SSS原有的可审计性。其核心思想是签名人在签名阶段调用SSS的签名算法对原始数据的可修订签名进行再次签名。根据SSS[15]的可审计性定义，文献[14]将ARS的可审计性分为3类：签名人的可审计性(若签名是由原始签名人生成的，则其不能抵赖该事实)、修订人的可审计性(若签名是由修订人生成的，则其不能抵赖该事实)和公开可审计性(通过公开信息任意第三方都可以正确判定签名是谁生成的)。

在文献[14]所构造的ARS方案中，只有原始签名人指定的修订人才可以合法地修订数据。然而，原始签名人很难预测原始数据在处理和使用过程中的数据持有人，在签名时很难指定所有相关的修订人。此外，该ARS方案的总开销是一个具体的RSS和一个具体的SSS的开销之和，在公钥长度、通信开销、签名生成与验证、证据生成与判定方面都存在不小的代价。并且，目前针对ARS的研究工作只有文献[14]，相关的研究仍处于起步阶段。

鉴于此，本文提出一类新颖的ARS方案的通用构造——公开可审计的可修订签名方案(Public Accountable Redactable Signature Scheme, PA-RSS)。原始签名人在签名时无需指定修订人，任何人在不与原始签名人交互的情况下都可以合法地修订数据以及确定一个有效数据/签名对的生成人，具有公开可修订和公开可审计功能。形式化定义PA-RSS方案的通用模型以及安全性需求，基于传统数字签名方案给出PA-RSS的一个具体构造，并证明该构造的安全性。与文献[14]中具有公开可审计功能的ARS方案相比，本文构造的PA-RSS方案具有更高的效率。

2 原语介绍

2.1 传统数字签名方案

传统数字签名方案(Digital Signature Scheme, DSS)由如下3个有效算法构成：

(1) 密钥生成算法$\mathrm{DSS.KeyGen}(\lambda)$：该算法的输入为系统安全参数$\lambda$，输出签名人的公/私钥对$(\mathrm{pk_S}, \mathrm{sk_S})$，书写为：$(\mathrm{pk_S}, \mathrm{sk_S}) \leftarrow \mathrm{DSS.KeyGen}(\lambda)$；

(2) 签名算法$\mathrm{DSS.Sign}(\mathrm{sk_S}, M)$：该算法的输入为签名人的私钥$\mathrm{sk_S}$和数据$M$，输出数据$M$的签名$\sigma_{\mathrm{DSS}}$，书写为：$\sigma_{\mathrm{DSS}} \leftarrow \mathrm{DSS.Sign}(\mathrm{sk_S}, M)$；

(3) 验证算法$\mathrm{DSS.Verify}(\mathrm{pk_S}, (M, \sigma_{\mathrm{DSS}}))$：该算

法的输入为签名人的公钥$\mathrm{pk_S}$和数据/签名对$(M,\sigma_{\mathrm{DSS}})$，若$(M,\sigma_{\mathrm{DSS}})$是有效的，输出Valid，否则输出Invalid。书写为：$\{\mathrm{Valid},\mathrm{Invalid}\}\leftarrow \mathrm{DSS.Verify}(\mathrm{pk_S},(M,\sigma_{\mathrm{DSS}}))$。

DSS的正确性要求：如果$(\mathrm{pk_S},\mathrm{sk_S})$是由算法$\mathrm{DSS.KeyGen}(\lambda)$正确生成，并且数据$M$的签名$\sigma_{\mathrm{DSS}}$是由算法$\mathrm{DSS.Sign}(\mathrm{sk_S},M)$正确生成，那么算法$\mathrm{DSS.Verify}(\mathrm{pk_S},(M,\sigma_{\mathrm{DSS}}))$的输出结果一定为Valid。DSS的传统安全要求为EUF-CMA，具体的形式化定义可参见文献[1]。

2.2 可修订签名方案

可修订签名方案RSS由如下4个有效算法构成：

(1) 密钥生成算法$\mathrm{RSS.KeyGen}(\lambda)$：该算法的输入为系统安全参数$\lambda$，输出签名人的公/私钥对$(\mathrm{pk_S},\mathrm{sk_S})$，书写为：$(\mathrm{pk_S},\mathrm{sk_S})\leftarrow \mathrm{RSS.KeyGen}(\lambda)$；

(2) 签名算法$\mathrm{RSS.Sign}(\mathrm{sk_S},M)$：该算法的输入为签名人的私钥$\mathrm{sk_S}$和数据M，输出数据M的签名$\sigma_{\mathrm{RSS}}$，书写为：$\sigma_{\mathrm{RSS}}\leftarrow \mathrm{RSS.Sign}(\mathrm{sk_S},M)$；

(3) 修订算法$\mathrm{RSS.Redact}(\mathrm{pk_S},(M,\sigma_{\mathrm{RSS}}),X)$：该算法的输入为有效的数据/签名对$(M,\sigma_{\mathrm{RSS}})$，修订子集$X\subseteq M$以及签名人的公钥$\mathrm{pk_S}$，输出新数据$M'=M\backslash X$以及$M'$的签名$\sigma'_{\mathrm{RSS}}$，书写为：$(M',\sigma'_{\mathrm{RSS}})\leftarrow \mathrm{RSS.Redact}(\mathrm{pk_S},(M,\sigma_{\mathrm{RSS}}),X)$；

(4) 验证算法$\mathrm{RSS.Verify}(\mathrm{pk_S},(M,\sigma_{\mathrm{RSS}}))$：该算法的输入为签名人的公钥$\mathrm{pk_S}$和数据/签名对$(M,\sigma_{\mathrm{RSS}})$，若$(M,\sigma_{\mathrm{RSS}})$是有效的，输出Valid，否则输出Invalid。书写为：$\{\mathrm{Valid},\mathrm{Invalid}\}\leftarrow \mathrm{RSS.Verify}(\mathrm{pk_S},(M,\sigma_{\mathrm{RSS}}))$。

RSS的正确性要求：如果$(\mathrm{pk_S},\mathrm{sk_S})$是由算法$\mathrm{RSS.KeyGen}(\lambda)$正确生成，数据M的签名$\sigma_{\mathrm{RSS}}$是由算法$\mathrm{RSS.Sign}(\mathrm{sk_S},M)$正确生成，那么：(1)算法$\mathrm{RSS.Verify}(\mathrm{pk_S},(M,\sigma_{\mathrm{RSS}}))$的输出结果一定为Valid; (2)若$(M',\sigma'_{\mathrm{RSS}})$是由算法$\mathrm{RSS.Redact}(\mathrm{pk_S},(M,\sigma_{\mathrm{RSS}}),X)$正确生成，那么算法$\mathrm{RSS.Verify}(\mathrm{pk_S},(M',\sigma'_{\mathrm{RSS}}))$的输出结果一定为Valid。RSS的两个基本安全需求为不可伪造性(unforgeability)和隐私性(privacy)。不可伪造性要求修订人只可以删除已签名原始数据的部分数据，不能为其他新数据生成有效的签名。隐私性要求修订后的数据/签名对不泄露关于被删除数据的任何额外信息。具体的形式化安全定义可参见文献[3,16]。

3 公开可审计的可修订签名方案(PA-RSS)的通用模型和形式化安全定义

3.1 PA-RSS方案的通用模型

PA-RSS由如下5个有效算法构成：

(1) 密钥生成算法$\mathrm{KeyGen}(\lambda)$：该算法的输入为系统安全参数λ，输出签名人的公/私钥对$(\mathrm{pk_S},\mathrm{sk_S})$和修订人的公/私钥对$(\mathrm{pk_R},\mathrm{sk_R})$，书写为：$\{(\mathrm{pk_S},\mathrm{sk_S}),(\mathrm{pk_R},\mathrm{sk_R})\}\leftarrow \mathrm{KeyGen}(\lambda)$；

(2) 签名算法$\mathrm{Sign}(\mathrm{sk_S},M)$：该算法的输入为签名人的私钥$\mathrm{sk_S}$和数据$M$，输出数据$M$的签名$\sigma$，书写为：$\sigma\leftarrow \mathrm{Sign}(\mathrm{sk_S},M)$；

(3) 修订算法$\mathrm{Redact}(\mathrm{sk_R},\mathrm{pk_S},(M,\sigma),X)$：该算法的输入为修订人的私钥$\mathrm{sk_R}$、签名人的公钥$\mathrm{pk_S}$、有效的数据/签名对$(M,\sigma)$以及修订子集$X\subseteq M$，输出新数据$M'=M\backslash X$以及$M'$的签名$\sigma'$，书写为：$(M',\sigma')\leftarrow \mathrm{Redact}(\mathrm{sk_R},\mathrm{pk_S},(M,\sigma),X)$；

(4) 验证算法$\mathrm{Verify}(\mathrm{pk_S},\mathrm{pk_R},(M,\sigma))$：该算法的输入为签名人的公钥$\mathrm{pk_S}$、修订人的公钥$\mathrm{pk_R}$以及数据/签名对$(M,\sigma)$，若$(M,\sigma)$是有效的，输出Valid，否则输出Invalid。书写为：$\{\mathrm{Valid},\mathrm{Invalid}\}\leftarrow \mathrm{Verify}(\mathrm{pk_S},\mathrm{pk_R},(M,\sigma))$；

(5) 判断算法$\mathrm{Judge}(\mathrm{pk_S},\mathrm{pk_R},(M,\sigma))$：该算法的输入为签名人的公钥$\mathrm{pk_S}$、修订人的公钥$\mathrm{pk_R}$以及有效的数据/签名对$(M,\sigma)$，输出$(M,\sigma)$的生成人。若$(M,\sigma)$是由签名人生成，则输出Signer的身份标识$\mathrm{ID_S}$；若是修订人生成的，输出Redactor的身份标识$\mathrm{ID_R}$；否则，输出$\perp$。书写为：$\{\mathrm{ID_S},\mathrm{ID_R},\perp\}\leftarrow \mathrm{Judge}(\mathrm{pk_S},\mathrm{pk_R},(M,\sigma))$。

PA-RSS的正确性要求：如果密钥对$\{(\mathrm{pk_S},\mathrm{sk_S}),(\mathrm{pk_R},\mathrm{sk_R})\}$是由算法$\mathrm{KeyGen}(\lambda)$正确生成，数据$M$的签名$\sigma$是由算法$\mathrm{Sign}(\mathrm{sk_S},M)$正确生成，那么：(1)算法$\mathrm{Verify}(\mathrm{pk_S},\mathrm{pk_R},(M,\sigma))$的输出结果一定是Valid，并且算法$\mathrm{Judge}(\mathrm{pk_S},\mathrm{pk_R},(M,\sigma))$的输出结果一定是$\mathrm{ID_S}$；(2)如果$(M',\sigma')$是由算法$\mathrm{Redact}(\mathrm{sk_R},\mathrm{pk_S},(M,\sigma),\mathrm{X})$正确生成，那么验证算法$\mathrm{Verify}(\mathrm{pk_S},\mathrm{pk_R},(M',\sigma'))$的输出结果一定是Valid，并且算法$\mathrm{Judge}(\mathrm{pk_S},\mathrm{pk_R},(M',\sigma'))$的输出结果一定是$\mathrm{ID_R}$。

3.2 PA-RSS的形式化安全定义

PA-RSS的安全性包括不可伪造性、隐私性和公开可审计性。公开可审计性又分为签名人的公开可审计性和修订人的公开可审计性。安全的PA-RSS应满足上述所有安全性要求。

(1) PA-RSS的不可伪造性要求：在没有签名人私钥$\mathrm{sk_S}$的情况下，即使是不诚实的PPT修订人可以自适应地询问签名预言机，也不能为新数据生成有效的签名。PA-RSS的不可伪造性可用下面的挑战者C和攻击者A之间的游戏Game 1刻画：

(a) 挑战者C运行密钥生成算法$\mathrm{KeyGen}(\lambda)$，得到$(\mathrm{pk_S},\mathrm{sk_S})$，并将$(\lambda,\mathrm{pk_S})$发送给攻击者A。A运行算法$\mathrm{KeyGen}(\lambda)$，得到$(\mathrm{pk_R},\mathrm{sk_R})$，并将$\mathrm{pk_R}$发送给C；

(b) A可自适应地选择q个数据M_i询问签名预言机Sign，并得到有效的签名回复σ_i，其中$1 \le i \le q$；

(c) A输出数据/签名对(M^*, σ^*)；

(d) 若M^*未被询问过签名预言机，并且M^*不是任一被询问数据的子数据，挑战者C运行验证算法$\text{Verify}(\text{pk}_\text{S}, \text{pk}_\text{R}, (M^*, \sigma^*))$，若验证结果为Valid，则说明攻击者A在游戏Game 1中获胜；否则A失败。

定义1 不可伪造性：在自适应选择消息攻击下，如果任意的PPT攻击者在游戏Game 1中获胜的概率是可忽略的，那么就称该PA-RSS在适应性选择消息攻击下满足不可伪造性要求。

(2) PA-RSS的隐私性要求：在已知签名人公钥pk_S和修订人公钥pk_R的情况下，任意的PPT攻击者可以自适应地询问签名预言机和修订预言机，其在概率多项式时间内推导出有关被删除数据内容是困难的。PA-RSS的隐私性可用下面的挑战者C和攻击者A之间的游戏Game 2刻画：

(a) 挑战者C运行密钥生成算法$\text{KeyGen}(\lambda)$，得到$(\text{pk}_\text{S}, \text{sk}_\text{S})$和$(\text{pk}_\text{R}, \text{sk}_\text{R})$，并将$(\lambda, \text{pk}_\text{S}, \text{pk}_\text{R})$发送给攻击者A；

(b) A可自适应地选择数据询问签名预言机Sign和修订预言机Redact，并得到相应的签名回复和修订回复。最后，A输出2组挑战的数据/修订子集对$\{(M_0, X_0), (M_1, X_1)\}$，记$M'_0 = M_0 \backslash X_0$, $M'_1 = M_1 \backslash X_1$，其满足$M_0 \ne M_1$, $M'_0 = M'_1$，并且M_0和M_1均没有被询问过签名预言机Sign, (M_0, X_0)和(M_1, X_1)均没有被询问过修订预言机Redact；

(c) C随机均匀地选择一个比特值$\text{b} \in \{0, 1\}$，顺序地执行：$\sigma_\text{b} \leftarrow \text{Sign}(\text{sk}_\text{S}, M_\text{b})$和$(M'_\text{b}, \sigma'_\text{b}) \leftarrow \text{Redact}(\text{sk}_\text{R}, \text{pk}_\text{S}, (M_\text{b}, \sigma_\text{b}), X_\text{b})$，并将$(M'_\text{b}, \sigma'_\text{b})$发送给A；

(d) A在获得$(M'_\text{b}, \sigma'_\text{b})$后，依然可以自适应地选择数据询问签名预言机Sign和修订预言机Redact，但不可以询问挑战的数据/修订子集对。最后，A输出其猜测b^*。如果$b^* = b$，则说明攻击者A在游戏Game 2中获胜；否则A失败。

假设攻击者A在Game 2中获胜的概率为$\Pr[\text{A}]$，那么其获胜的优势为$\text{Adv}[\text{A}] = |\Pr[\text{A}] - 1/2|$。

定义 2 隐私性：若任意的PPT攻击者在游戏Game 2中获胜的优势$\text{Adv}[\text{A}]$是可忽略的，那么就称该PA-RSS满足隐私性要求。

(3) PA-RSS的签名人的公开可审计性要求：在已知修订人公钥pk_R的情况下，即使是不诚实的PPT签名人可以自适应地询问修订预言机，也不能生成一个有效的数据/签名对使得第三方相信该数据/签名对是由修订人生成的。PA-RSS的签名人的公开可审计性可用下面的挑战者C和攻击者A之间的游戏Game 3刻画：

(a) 挑战者C运行密钥生成算法$\text{KeyGen}(\lambda)$，得到$(\text{pk}_\text{R}, \text{sk}_\text{R})$，并将$(\lambda, \text{pk}_\text{R})$发送给攻击者A。A运行算法$\text{KeyGen}(\lambda)$，得到$(\text{pk}_\text{S}, \text{sk}_\text{S})$，并将$\text{pk}_\text{S}$发送给C；

(b) A可自适应地询问修订预言机Redact，并得到相应的修订回复。最后，A输出数据/签名对(M^*, σ^*)；

(c) 若(M^*, σ^*)不是修订预言机Redact的修订回复，$\text{Valid} \leftarrow \text{Verify}(\text{pk}_\text{S}, \text{pk}_\text{R}, (M^*, \sigma^*))$，并且$\text{ID}_\text{R} \leftarrow \text{Judge}(\text{pk}_\text{S}, \text{pk}_\text{R}, (M^*, \sigma^*))$，则说明攻击者A在游戏Game 3中获胜；否则A失败。

定义3 签名人的公开可审计性：如果任意的PPT攻击者在游戏Game 3中获胜的概率是可忽略的，就称该PA-RSS满足签名人的公开可审计性要求。

(4) PA-RSS的修订人的公开可审计性要求：在已知签名人公钥pk_S的情况下，即使是不诚实的PPT修订人可以自适应地询问签名预言机，也不能生成一个有效的数据/签名对使得第三方相信该数据/签名对是由签名人生成的。PA-RSS的修订人的公开可审计性可用下面的挑战者C和攻击者A之间的游戏Game 4刻画：

(a) 挑战者C运行密钥生成算法$\text{KeyGen}(\lambda)$，得到$(\text{pk}_\text{S}, \text{sk}_\text{S})$，并将$(\lambda, \text{pk}_\text{S})$发送给攻击者A。A运行算法$\text{KeyGen}(\lambda)$，得到$(\text{pk}_\text{R}, \text{sk}_\text{R})$，并将$\text{pk}_\text{R}$发送给C；

(b) A可自适应地选择数据询问签名预言机Sign，并得到相应的签名回复。最后，A输出数据/签名对(M^*, σ^*)；

(c) 若(M^*, σ^*)不是签名预言机Sign的签名回复，$\text{Valid} \leftarrow \text{Verify}(\text{pk}_\text{S}, \text{pk}_\text{R}, (M^*, \sigma^*))$，并且$\text{ID}_\text{S} \leftarrow \text{Judge}(\text{pk}_\text{S}, \text{pk}_\text{R}, (M^*, \sigma^*))$，则说明攻击者A在游戏Game 4中获胜；否则A失败。

定义 4 修订人的公开可审计性：如果任意的PPT攻击者在游戏Game 4中获胜的概率是可忽略的，那么就称该PA-RSS满足修订人的公开可审计性要求。

4 公开可审计的可修订签名方案(PA-RSS)的具体构造

本文提出的PA-RSS具体构造是一个通用转化，利用任意的DSS，可以将任意的RSS转化为PA-RSS。具体构造过程如下：

(1) 密钥生成算法$\text{KeyGen}(\lambda)$：该算法为签名人生成两个密钥对$\left(\text{pk}_\text{S}^{\text{DSS}}, \text{sk}_\text{S}^{\text{DSS}}\right)$和$\left(\text{pk}_\text{S}^{\text{RSS}}, \text{sk}_\text{S}^{\text{RSS}}\right)$，并为修订人生成密钥对$\left(\text{pk}_\text{R}^{\text{DSS}}, \text{sk}_\text{R}^{\text{DSS}}\right)$。具体地，调

用算法DSS.KeyGen(λ)两次，输出分别为$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}, \mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$和$\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}, \mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$；调用算法RSS.KeyGen($\lambda$)，输出为$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, \mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$。

(2) 签名算法Sign $\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}, \mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}, M\right)$：数据$M$的签名$\sigma$包含两个部分：$\sigma_1$和$\sigma_2$。具体地，签名人首先调用算法RSS.Sign，该算法的输入为签名人的私钥$\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}$和原始数据$M$，输出原始数据$M$的可修订签名$\sigma_{\mathrm{RSS}}$，即$\sigma_{\mathrm{RSS}} \leftarrow \mathrm{RSS.Sign}\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}, M\right)$。设置$\sigma_1 = \left(\sigma_{\mathrm{RSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$。然后调用算法DSS.Sign，该算法的输入为签名人的私钥$\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}$和原始可修订数据/签名对$(M, \sigma_1)$，输出为$(M, \sigma_1)$的数字签名$\sigma_{\mathrm{DSS}}$，即$\sigma_{\mathrm{DSS}} \leftarrow \mathrm{DSS.Sign}\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}, (M, \sigma_1)\right)$。设置$\sigma_2 = \left(\sigma_{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$。最后，输出原始签名$\sigma = (\sigma_1, \sigma_2)$。

(3) 修订算法Redact$\left(\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, M, \sigma, X\right)$：$(M, \sigma)$是有效的数据/签名对，其中$\sigma = (\sigma_1, \sigma_2)$，$\sigma_1 = \left(\sigma_{\mathrm{RSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，$\sigma_2 = \left(\sigma_{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$。数据$M$的修订签名$\sigma'$包含两个部分：$\sigma'_1$和$\sigma'_2$。具体地，首先调用算法RSS.Redact，该算法的输入为有效的数据/签名对$(M, \sigma_{\mathrm{RSS}})$，修订子集$X \subseteq M$以及签名人的公钥$\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}$，输出新数据$M' = M \backslash X$以及$M'$的签名$\sigma'_{\mathrm{RSS}}$，即$(M', \sigma'_{\mathrm{RSS}}) \leftarrow \mathrm{RSS.Redact}\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, (M, \sigma_{\mathrm{RSS}}), X\right)$。设置$\sigma'_1 = \left(\sigma'_{\mathrm{RSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$。然后调用算法DSS.Sign，该算法的输入为修订人的私钥$\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}$和数据/签名对$(M', \sigma'_1)$，输出$(M', \sigma'_1)$的数字签名$\sigma'_{\mathrm{DSS}}$，即$\sigma'_{\mathrm{DSS}} \leftarrow \mathrm{DSS.Sign}\left(s\mathrm{k}_{\mathrm{R}}^{\mathrm{DSS}}, (M', \sigma'_1)\right)$。设置$\sigma'_2 = \left(\sigma'_{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$。最后，输出修订后的数据/签名对$(M', \sigma')$，其中$\sigma' = (\sigma'_1, \sigma'_2)$。

(4) 验证算法Verify $\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}, (M, \sigma)\right)$：该算法调用算法RSS.Verify和DSS.Verify验证数据/签名对(M, σ)的有效性，其中$\sigma = (\sigma_1, \sigma_2)$，$\sigma_1 = \left(\sigma_{\mathrm{RSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，$\sigma_2 = \left(\sigma_{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$或$\sigma_2 = \left(\sigma_{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$；若$(M, \sigma)$是有效的，输出Valid，否则输出Invalid。具体地，首先调用算法RSS.Verify，该算法的输入为签名人的公钥$\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}$和数据/签名对$(M, \sigma_{\mathrm{RSS}})$，即$\{\mathrm{Valid}, \mathrm{Invalid}\} \leftarrow \mathrm{RSS.Verify}\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, (M, \sigma_{\mathrm{RSS}})\right)$。若$(M, \sigma_{\mathrm{RSS}})$是无效的签名，输出Invalid；否则，继续调用算法DSS.Verify，该算法的输入为签名人的公钥$\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}$、修订人的公钥$\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}$、数据/签名对$(M, \sigma_1)$以及签名$\sigma_{\mathrm{DSS}}$，运行$\{\mathrm{Valid}, \mathrm{Invalid}\} \leftarrow \mathrm{DSS.Verify}(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}, (M, \sigma_1), \sigma_{\mathrm{DSS}})$或$\{\mathrm{Valid}, \mathrm{Invalid}\} \leftarrow \mathrm{DSS.Verify}(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}, (M, \sigma_1), \sigma_{\mathrm{DSS}})$，若DSS.Verify的验证结果均为Invalid，输出Invalid；否则输出Valid。

(5) 判断算法Judge $\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}, (M, \sigma)\right)$：$(M, \sigma)$是有效的数据/签名对，其中$\sigma = (\sigma_1, \sigma_2)$，$\sigma_1 = \left(\sigma_{\mathrm{RSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，$\sigma_2 = \left(\sigma_{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$或$\sigma_2 = \left(\sigma_{\mathrm{DSS}}, \mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$。该算法调用算法RSS.Verify和DSS.Verify判定有效数据/签名对(M, σ)的生成人。若算法DSS.Verify $\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, (M, \sigma_{\mathrm{RSS}}), \sigma_{\mathrm{DSS}}\right)$的验证结果为Valid，那么确定$(M, \sigma)$的生成人为签名人，输出的签名人的身份标识$\mathrm{ID}_{\mathrm{S}}$；若算法DSS.Verify $\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}, (M, \sigma_{\mathrm{RSS}}), \sigma_{\mathrm{DSS}}\right)$的验证结果为Valid，那么确定$(M, \sigma)$的生成人为修订人，输出修订人的身份标识$\mathrm{ID}_{\mathrm{R}}$。

本文构造的PA-RSS方案的正确性基于所采用的DSS和RSS方案的正确性，可以被直接验证。限于篇幅，这里略去该验证过程。

5　PA-RSS具体构造的安全性分析

本文构造的PA-RSS方案满足不可伪造性、隐私性、签名人的公开可审计性和修订人的公开可审计性。

5.1　不可伪造性证明

定理 1　如果采用的DSS方案满足EUF-CMA要求，并且采用的RSS方案满足不可伪造性要求，那么本文的PA-RSS方案也满足不可伪造性要求。

证明　假设存在一个PPT攻击者$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$可以在PA-RSS的不可伪造性游戏Game 1中获胜，其获胜的概率记为$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$。那么利用$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$，可以构造：(1) 能够为新数据伪造有效数字签名的攻击者$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$，其成功的概率记为$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}$; (2) 能够为新数据伪造有效可修订签名的攻击者$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$，其成功的概率记为$\varepsilon_{\mathrm{RSS}}^{\mathrm{Unf}}$。下面将对两种情况分别展开证明。

(1) 挑战者$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法DSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}, \mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$，并将$\left(\lambda, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$发送给$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。假设采用的RSS方案满足不可伪造性要求。为了实现其目标，$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$模拟$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$的挑战者，执行以下操作：

初始化阶段：$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法RSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, \mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，并将$\left(\lambda, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$发送给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$运行算法DSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}, \mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$，并将$\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}$发送给$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。

询问阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$可以自适应地选择数据询问签名预言机Sign，该预言机由$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$模拟，模拟过程如下：记数据$\mathrm{M}_i$是$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$询问Sign的第$i$个数据，$i \in [1, q]$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$首先运行算法RSS.Sign $\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}, M_i\right)$生成$M_i$的可修订签名$\sigma_{\mathrm{RSS},i}$，设置$\sigma_{1,i} = \left(\sigma_{\mathrm{RSS},i}, \mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$询问挑战者$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$关于$(M_i, \sigma_{1,i})$的数字签名，

$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法DSS.Sign$(\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}},(M_i,\sigma_{1,i}))$生成签名$\sigma_{\mathrm{DSS},i}$，并将$\sigma_{\mathrm{DSS},i}$返回给$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$设置$\sigma_{2,i}=\left(\sigma_{\mathrm{DSS,i}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$以及$\sigma_i=(\sigma_{1,i},\sigma_{2,i})$。最后，$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$将$\sigma_i$返回给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$作为数据$M_i$的签名回复。

伪造阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$输出其伪造的数据/签名对$(M^*,\sigma^*)$，其中$\sigma^*=(\sigma_1{}^*,\sigma_2{}^*)$，$\sigma_1^*=(\sigma_{\mathrm{RSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}})$，$\sigma_2{}^*=(\sigma_{\mathrm{DSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}})$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$输出数据/签名对$((M^*,\sigma_{\mathrm{RSS}}^*),\sigma_2^*)$作为自己的伪造结果。

根据PA-RSS的不可伪造性定义可知，如果$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$在Game 1中获胜，那么$(M^*,\sigma^*)$满足：$M^*$没有被询问过签名预言机Sign，并且$M^*$不是任一被询问数据$M_i$的子数据，同时验证算法Verify$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},(M^*,\sigma^*)\right)$的输出结果为Valid。由此可见，$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}=\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$。如果$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$是可忽略的，那么$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}$也是可忽略的。

(2) 挑战者$\mathrm{C}_{\mathrm{RSS}}^{\mathrm{Unf}}$运行算法RSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，并将$(\lambda,\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}})$发送给$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$。假设采用的DSS方案满足EUF-CMA要求。为了实现其目标，$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$模拟$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$的挑战者，执行以下操作：

初始化阶段：$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$运行算法DSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$，并将$\left(\lambda,\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$发送给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$运行算法DSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$，并将$\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}$发送给$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$。

询问阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$可以自适应地选择数据询问签名预言机Sign，该预言机由$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$模拟，模拟过程如下：记数据$M_i$是$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$询问Sign的第$i$个数据，$i\in[1,\mathrm{q}]$。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$首先询问挑战者$\mathrm{C}_{\mathrm{RSS}}^{\mathrm{Unf}}$关于数据$M_i$的可修订签名。$\mathrm{C}_{\mathrm{RSS}}^{\mathrm{Unf}}$运行算法RSS.Sign$\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}},M_i\right)$生成$\sigma_{\mathrm{RSS},i}$，并将$\sigma_{\mathrm{RSS},i}$返回给$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$首先设置$\sigma_{1,i}=\left(\sigma_{\mathrm{RSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，接着运行算法DSS.Sign$\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}},(M_i,\sigma_{1,i})\right)$生成签名$\sigma_{\mathrm{DSS},i}$，然后设置$\sigma_{2,i}=\left(\sigma_{\mathrm{DSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$以及$\sigma_i=(\sigma_{1,i},\sigma_{2,i})$。最后，$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$将$\sigma_i$返回给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$作为数据$M_i$的签名回复。

伪造阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$输出其伪造的数据/签名对$(M^*,\sigma^*)$，其中$\sigma^*=(\sigma_1^*,\sigma_2^*)$，$\sigma_1^*=\left(\sigma_{\mathrm{RSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，$\sigma_2^*=\left(\sigma_{\mathrm{DSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Unf}}$输出数据/签名对$(M*,\sigma_{\mathrm{RSS}}*)$作为自己的伪造结果。

根据PA-RSS的不可伪造性定义可知，如果$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$在Game 1中获胜，那么$(M^*,\sigma^*)$满足：$M^*$没有被询问过签名预言机Sign，并且$M^*$不是任一被询问数据$M_i$的子数据，同时验证算法Verify$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},(M^*,\sigma^*)\right)$的输出结果为Valid。由此可见，$\varepsilon_{\mathrm{RSS}}^{\mathrm{Unf}}=\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$。如果$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Unf}}$是可忽略的，那么$\varepsilon_{\mathrm{RSS}}^{\mathrm{Unf}}$也是可忽略的。 证毕

5.2 隐私性证明

定理 2 如果采用的RSS方案满足隐私性要求，那么本文PA-RSS方案也满足隐私性要求。

证明 假设存在一个PPT攻击者$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$可以在PA-RSS的隐私性游戏Game 2中获胜，其获胜的概率记为$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$。那么利用$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$，可以构造能够攻破RSS方案隐私性的攻击者$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$，其成功的概率记为$\varepsilon_{\mathrm{RSS}}^{\mathrm{Pri}}$。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$模拟$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$的挑战者，执行以下操作：

初始化阶段：挑战者$\mathrm{C}_{\mathrm{RSS}}^{\mathrm{Pri}}$运行算法RSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，并将$\left(\lambda,\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$发送给$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$运行算法DSS.KeyGen($\lambda$)两次，得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$和$\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$，并将$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$发送给攻击者$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$。

询问阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$可以自适应地选择数据询问签名预言机Sign和修订预言机Redact，两个预言机均由$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$模拟。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$模拟签名预言机Sign的响应过程同5.1节第2种情况证明中询问阶段的响应过程一样。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$利用密钥对$\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$运行修订算法响应$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$的修订签名询问。

挑战阶段：询问阶段结束之后，$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$选择两个数据$M_0$和$M_1$以及两个修订子集$X_0$和$X_1$作为自己的挑战，满足$M_0\neq M_1$，并且$M_0\backslash X_0=M_1\backslash X_1$。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$将$(M_0,X_0,M_1,X_1)$发送给$\mathrm{C}_{\mathrm{RSS}}^{\mathrm{Pri}}$作为自己的挑战。$\mathrm{C}_{\mathrm{RSS}}^{\mathrm{Pri}}$随机均匀地选择一个比特值$b\in\{0,1\}$，顺序地运行算法RSS.Sign$(\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}},M_b)$生成$\sigma_{\mathrm{RSS},b}$以及算法$(M'_b,\sigma'_{\mathrm{RSS},b})\leftarrow\mathrm{RSS.Redact}\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},(M_b,\sigma_{\mathrm{RSS},b}),X_b\right)$，并将$(M'_b,\sigma'_{\mathrm{RSS},b})$返回给$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$。$\mathrm{A}_{\mathrm{RSS}}^{\mathrm{Pri}}$设置$\sigma'_{1,b}=\left(\sigma'_{\mathrm{RSS},b},\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，运行算法$\sigma'_{\mathrm{DSS},b}\leftarrow\mathrm{DSS.Sign}\left(\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}},(M'_b,\sigma'_{1,b})\right)$，设置$\sigma'_{2,b}=\left(\sigma'_{\mathrm{DSS},b},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$，然后将$(M'_b,\sigma'_b)$返回给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$，其中$\sigma'_b=(\sigma'_{1,b},\sigma'_{2,b})$。

猜测阶段：在收到$(M'_{\mathrm{b}},\sigma'_{\mathrm{b}})$后，$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$依然可以自适应地选择数据询问签名预言机Sign和修订预言机Redact，但不可以询问关于挑战的两个数据M_0和M_1的签名以及(M_0,X_0)和(M_1,X_1)的修订签名。最后，$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$输出其猜测$b^*$。$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$输出$b^*$作为自己的猜测。

根据PA-RSS的隐私性定义可知，如果$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$在Game 2中获胜，那么$b^*=b$。由此可见，$\varepsilon_{\mathrm{RSS}}^{\mathrm{Pri}}=\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$。如果$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{Pri}}$是可忽略的，那么$\varepsilon_{\mathrm{RSS}}^{\mathrm{Pri}}$也是可忽略的。 证毕

5.3 签名人的公开可审计性证明

定理 3　如果采用的DSS方案满足EUF-CMA要求，那么本文PA-RSS方案满足签名人的公开可审计性要求。

证明　假设存在一个PPT攻击者$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$可以在PA-RSS的签名人的公开可审计游戏Game 3中获胜，其获胜的概率记为$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$。那么利用$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$，可以构造能够为新数据伪造有效数字签名的攻击者$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$，其成功的概率记为$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$模拟$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$的挑战者，执行以下操作：

初始化阶段：挑战者$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法DSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$，并将$\left(\lambda,\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$发送给$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$再将$\left(\lambda,\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$发送给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$。$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$运行算法RSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，运行算法DSS.KeyGen($\lambda$) 得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$，并将$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$发送给攻击者$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。

询问阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$可以自适应地选择数据询问修订预言机Redact，该预言机由$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$模拟，模拟过程如下：记$(M_i,\sigma_i,X_i)$是$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$的第$i$个修订签名询问，其中，$\sigma_i=(\sigma_{1,i},\sigma_{2,i})$, $\sigma_{1,i}=\left(\sigma_{\mathrm{RSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$, $\sigma_{2,i}=\left(\sigma_{\mathrm{DSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$, $i\in[1,q]$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法RSS.Redact$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},(M_i,\sigma_{\mathrm{RSS},i}),X_i\right)$得到$(M'_i,\sigma'_{\mathrm{RSS},i})$，设置$\sigma'_{1,i}=(\sigma'_{\mathrm{RSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}})$。然后$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$询问挑战者$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$关于$(M'_i,\sigma'_{1,i})$的数字签名，$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法DSS.Sign$\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}},(M'_i,\sigma'_{1,i})\right)$生成签名$\sigma'_{\mathrm{DSS},i}$，并将$\sigma'_{\mathrm{DSS},i}$返回给$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$设置$\sigma'_{2,i}=\left(\sigma'_{\mathrm{DSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$以及$\sigma'_i=(\sigma'_{1,i},\sigma'_{2,i})$。最后，$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$将$\sigma'_i$返回给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$作为$(M_i,\sigma_i,X_i)$的修订签名回复。

伪造阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$输出其伪造的数据/签名对$(M^*,\sigma^*)$，其中$\sigma^*=(\sigma_1^*,\sigma_2^*)$，$\sigma_1^*=\left(\sigma_{\mathrm{RSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$, $\sigma_2^*=\left(\sigma_{\mathrm{DSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$输出数据/签名对$((M^*,\sigma_{\mathrm{RSS}}^*),\sigma_2^*)$作为自己的伪造结果。

根据PA-RSS的签名人的公开可审计性定义可知，如果$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$在Game 3中获胜，那么$(M^*,\sigma^*)$不是修订预言机Redact的修订回复，Valid $\leftarrow$ Verify$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},(M^*,\sigma^*)\right)$，并且$\mathrm{ID}_{\mathrm{R}}\leftarrow$ Judge$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},(M^*,\sigma^*)\right)$。由此可见，$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}=\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$。如果$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{S-Accout}}$是可忽略的，那么$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}$也是可忽略的。证毕

5.4 修订人的公开可审计性证明

定理 4　如果采用的DSS方案满足EUF-CMA要求，那么本文PA-RSS方案满足修订人的公开可审计性要求。

证明　假设存在一个PPT攻击者$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$可以在PA-RSS的修订人的公开可审计游戏Game 4中获胜，其获胜的概率记为$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$。那么利用$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$，可以构造能够为新数据伪造有效数字签名的攻击者$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$，其成功的概率记为$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$模拟$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$的挑战者，执行以下操作：

初始化阶段：挑战者$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法DSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$，并将$\left(\lambda,\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$发送给$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法RSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$，并将$\left(\lambda,\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$发送给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$。$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$运行算法DSS.KeyGen($\lambda$)得到$\left(\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},\mathrm{sk}_{\mathrm{R}}^{\mathrm{DSS}}\right)$，并将$\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}}$发送给攻击者$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。

询问阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$可以自适应地选择数据询问签名预言机Sign，该预言机由$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$模拟，模拟过程如下：记数据$M_i$是$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$询问Sign的第$i$个数据，$i\in[1,q]$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$首先运行算法RSS.Sign$\left(\mathrm{sk}_{\mathrm{S}}^{\mathrm{RSS}},M_i\right)$生成$M_i$的可修订签名$\sigma_{\mathrm{RSS},i}$，设置$\sigma_{1,i}=\left(\sigma_{\mathrm{RSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$询问挑战者$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$关于$(M_i,\sigma_{1,i})$的数字签名，$\mathrm{C}_{\mathrm{DSS}}^{\mathrm{Unf}}$运行算法DSS.Sign$(\mathrm{sk}_{\mathrm{S}}^{\mathrm{DSS}},(M_i,\sigma_{1,i}))$生成签名$\sigma_{\mathrm{DSS},i}$，并将$\sigma_{\mathrm{DSS},i}$返回给$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$设置$\sigma_{2,i}=\left(\sigma_{\mathrm{DSS},i},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$以及$\sigma_i=(\sigma_{1,i},\sigma_{2,i})$。最后，$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$将$\sigma_i$返回给$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$作为数据$M_i$的签名回复。

伪造阶段：$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$输出其伪造的数据/签名对$(M^*,\sigma^*)$，其中$\sigma^*=(\sigma_1^*,\sigma_2^*)$，$\sigma_1^*=\left(\sigma_{\mathrm{RSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}}\right)$, $\sigma_2^*=\left(\sigma_{\mathrm{DSS}}^*,\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}}\right)$。$\mathrm{A}_{\mathrm{DSS}}^{\mathrm{Unf}}$输出数据/签名对$((M^*,\sigma_{\mathrm{RSS}}^*),\sigma_2^*)$作为自己的伪造结果。

根据PA-RSS的修订人的公开可审计性定义可知，如果$\mathrm{A}_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$在Game 4中获胜，那么$(M^*,\sigma^*)$不是签名预言机Sign的签名回复，Valid $\leftarrow$ Verify$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{RSS}},\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},(M^*,\sigma^*)\right)$，并且$\mathrm{ID}_{\mathrm{S}}\leftarrow$ Judge$\left(\mathrm{pk}_{\mathrm{S}}^{\mathrm{DSS}},\mathrm{pk}_{\mathrm{R}}^{\mathrm{DSS}},(M^*,\sigma^*)\right)$。由此可见，$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}=\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$。如果$\varepsilon_{\mathrm{PA-RSS}}^{\mathrm{R-Accout}}$是可忽略的，那么$\varepsilon_{\mathrm{DSS}}^{\mathrm{Unf}}$也是可忽略的。证毕

6 性能分析

文献[14]构造了目前唯一的可审计的可修订签名方案ARS，该方案的总开销是一个具体的RSS方案和一个具体的SSS方案的开销之和。本文构造的PA-RSS方案的总开销是一个具体的RSS方案和一个具体的DSS方案的开销之和。SSS方案将可净化功能加入到具体的DSS方案中需要额外的计算/存储开销。因此，一个具体的SSS方案的总开销通常是大于其采用的具体的DSS方案的总开销。由此可见，与文献[14]中具有公开可审计性的ARS方案相

比，本文PA-RSS方案的效率更高。此外，PA-RSS方案中的原始签名人在签名阶段无需预设潜在的修订人，任何人都可以合法地修订数据，并且修订人无法抵赖其修订行为，因此本文方案更适用于实现开放共享环境中认证数据删除的公开可审计性。

7 结束语

具有可审计功能的可修订签名方案为遏制可修订签名方案面临的恶意修订问题提供了较好的解决方法。本文提出了一类新颖的公开可审计的可修订签名方案的通用构造AP-RSS，其通用构造可以将公开可审计性功能加入到任意的RSS方案。本文首先定义了AP-RSS方案的通用模型，并形式化定义了相应的安全性需求。接着，构造了一个具体的AP-RSS方案，并证明了所提方案满足不可伪造性、隐私性、签名人的公开可审计性以及修订人的公开可审计性需求。与现有的具有可审计功能的可修订签名方案相比，本文方案效率更高，更适用于实现开放共享环境中公开可审计的认证数据删除。

参 考 文 献

[1] GOLDWASSER S, MICALI S, and RIVEST R L. A digital signature scheme secure against adaptive chosen-message attacks[J]. *SIAM Journal on Computing*, 1988, 17(2): 281–308. doi: 10.1137/0217017.

[2] JOHNSON R, MOLNAR D, SONG D, *et al*. Homomorphic signature schemes[C]. Cryptographers' Track at the RSA Conference, San Jose, USA, 2002: 244–262. doi: 10.1007/3-540-45760-7_17.

[3] STEINFELD R, BULL L, and ZHENG Yuliang. Content extraction signatures[C]. The 4th International Conference on Information Security and Cryptology, Seoul, Korea, 2001: 285–304. doi: 10.1007/3-540-45861-1_22.

[4] MIYAZAKI K, IWAMURA M, MATSUMOTO T, *et al*. Digitally signed document sanitizing scheme with disclosure condition control[J]. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2005, E88-A(1): 239–246. doi: 10.1093/ietfec/e88-a.1.239.

[5] MA Jinhua, LIU Jianghua, WANG Min, *et al*. An efficient and secure design of redactable signature scheme with redaction condition control[C]. The 12th International Conference on Green, Pervasive, and Cloud Computing, Cetara, Italy, 2017: 38–52. doi: 10.1007/978-3-319-57186-7_4.

[6] MIYAZAKI K, HANAOKA G, and IMAI H. Digitally signed document sanitizing scheme based on bilinear maps[C]. The 2006 ACM Symposium on Information, Computer and Communications Security, Taipei, China, 2006: 343–354. doi: 10.1145/1128817.1128868.

[7] BULL L, SQUIRE D M G, and ZHENG Yuliang. A hierarchical extraction policy for content extraction signatures: Selectively handling verifiable digital content[J]. *International Journal on Digital Libraries*, 2004, 4(3): 208–222. doi: 10.1007/s00799-004-0082-z.

[8] MA Jinhua, LIU Jianghua, HUANG Xinyi, *et al*. Authenticated data redaction with fine-grained control[J]. *IEEE Transactions on Emerging Topics in Computing*, To be published. doi: 10.1109/TETC.2017.2754646.

[9] LIU Jianghua, MA Jinhua, XIANG Yang, *et al*. Authenticated medical documents releasing with privacy protection and release control[J]. *IEEE Transactions on Dependable and Secure Computing*, To be published. doi: 10.1109/TDSC.2019.2892446.

[10] SAMELIN K, PÖHLS H C, BILZHAUSE A, *et al*. Redactable signatures for independent removal of structure and content[C]. The 8th International Conference on Information Security Practice and Experience, Hangzhou, China, 2012: 17–33. doi: 10.1007/978-3-642-29101-2_2.

[11] DE MEER H, PÖHLS H C, POSEGGA J, *et al*. Redactable signature schemes for trees with signer-controlled non-leaf-redactions[C]. International Conference on E-Business and Telecommunications. Berlin, Germany, 2012: 155–171. doi: 10.1007/978-3-662-44791-8_10.

[12] SAMELIN K, PÖHLS H C, BILZHAUSE A, *et al*. On structural signatures for tree data structures[C]. The 10th International Conference on Applied Cryptography and Network Security, Singapore, 2012: 171–187. doi: 10.1007/978-3-642-31284-7_11.

[13] DE MEER H, PÖHLS H C, POSEGGA J, *et al*. On the relation between redactable and sanitizable signature schemes[C]. The 6th International Symposium on Engineering Secure Software and Systems, Munich, Germany, 2014: 113–130. doi: 10.1007/978-3-319-04897-0_8.

[14] PÖHLS H C and SAMELIN K. Accountable redactable signatures[C]. The 2015 10th International Conference on Availability, Reliability and Security, Toulouse, France, 2015: 60–69. doi: 10.1109/ARES.2015.10.

[15] BRZUSKA C, FISCHLIN M, FREUDENREICH T, *et al*. Security of sanitizable signatures revisited[C]. The 12th International Workshop on Public Key Cryptography, Irvine, USA, 2009: 317–336. doi: 10.1007/978-3-642-00468-1_18.

[16] 马金花, 刘江华, 伍玮, 等. 可修订数字签名研究综述[J]. 计算机研究与发展, 2017, 54(10): 2144–2152. doi: 10.7544/issn1000-1239.2017.20170646.

MA Jinhua, LIU Jianghua, WU Wei, *et al*. Survey on redactable signatures[J]. *Journal of Computer Research and Development*, 2017, 54(10): 2144–2152. doi: 10.7544/issn1000-1239.2017.20170646.

马金花：女，1990年生，博士生，研究方向为网络安全与密码学.

黄欣沂：男，1981年生，博士，研究方向为网络安全与密码学.

许俊鹏：男，1995年生，硕士生，研究方向为网络安全与密码学.

伍 玮：女，1981年生，博士，研究方向为网络安全与密码学.