面向 6G 的雾无线接入网内生安全数据共享机制研究

刘杨[1,2]，李珺[1,2]，陈文韵[1]，彭木根[1,2]
（1. 北京邮电大学信息与通信工程学院，北京 100876；2. 北京邮电大学网络与交换技术国家重点实验室，北京 100876）

摘 要：为解决 6G 移动通信系统中雾无线接入网中数据共享的数据安全问题，提出了一种实现本地差分隐私和动态批量审计的内生安全数据共享机制。首先，用户本地对数据运行 RAPPOR 算法保护数据隐私；其次，雾接入点对数据进行缓存和预处理；再次，大功率节点对雾接入点上的数据进行基于 BLS 签名和 Merkle 哈希树的数据完整性审计；最后，BBU 池通过统计分析推断出共享数据的原始分布。安全性分析和仿真表明，所提机制实现了用户的本地差分隐私，并支持安全的多客户端批量审计和数据动态操作，同时具有较高的时间、空间和通信效率。

关键词：数据共享；雾无线接入网；内生安全；本地化差分隐私；数据完整性审计
中图分类号：TN92
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021005

Research on endogenous security data sharing mechanism of F-RAN for 6G

LIU Yang[1,2], LI Jun[1,2], CHEN Wenyun[1], PENG Mugen[1,2]
1. School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: In order to solve the data security problem of data sharing in the fog radio access network in the 6G mobile communication system, an endogenous secure data sharing mechanism that realized local differential privacy and dynamic batch auditing was proposed. First, the user ran the RAPPOR algorithm locally on the data to protect data privacy. Next, the fog access point cached and preprocessed the data. Then the high power node performed a data integrity audit based on the BLS signature and Merkle hash tree on the data cached on the fog access point. Finally, the BBU pool infered the original distribution of shared data through statistical analysis. Security analysis and simulation show that the proposed mechanism realizes the local differential privacy of users, supports secure multi-client batch audit and data dynamic operation, and has high efficiency in time, space and communication.

Keywords: data sharing, F-RAN, endogenous security, local differential privacy, data integrity auditing

1 引言

随着第五代移动通信系统（5G, the fifth generation mobile communication system）的落地和商用，学术界和产业界共同开启了对第六代移动通信系统（6G, the sixth generation mobile communication system）的研究。5G 的主要目标是实现大连接、高带宽和低时延，并实现“万物互联”，从传统移动通信行业渗透至工业物联网等垂直行业，满足未来 10 年（2020—2030 年）的无线通信需求[1]。对于 6G 而言，随着人工智能（AI, artificial intelligence）不断渗透到各行各业，6G 也将与 AI 深度结合，更

收稿日期：2020−08−07；修回日期：2020−10−24
基金项目：国家自然科学基金资助项目（No.61972049）；国家科技重大专项基金资助项目（No.2018ZX03001023）
Foundation Items: The National Natural Science Foundation of China (No.61972049), The National Science and Technology Major Project of China (No.2018ZX03001023)

多的智能化感知设备、人机接口将接入网络，“智慧连接”“深度连接”将成为新一代移动通信系统的重要特征[2]。

6G 中网络空间和运行的业务将会变得愈发复杂，无线终端数据流量的消耗也将大大增加。由于带宽和频谱的不足，传统的无线接入网络（RAN, radio access network）无法满足移动用户和运营商日益增长的需求。为了支持新的移动通信和服务，产业界先后提出了云无线接入网（CRAN, cloud radio access network）、异构云无线接入网（H-CRAN, heterogeneous cloud radio access network）、基于雾计算的无线接入网（F-RAN, fog radio access network）等架构作为新的无线接入网解决方案[3]。与传统基于集中式云计算的网络架构 C-RAN 和 H-CRAN 相比，F-RAN 充分利用无线远端射频单元（RRH, remote radio head）、雾无线接入点（F-AP, fog access point）和雾用户设备（F-UE, fog user equipment）等边缘设备，将协作无线信号处理（CRSP, collaboration radio signal processing）、协同无线资源管理（CRMM, cooperative radio resource management）和缓存、计算等功能在网络边缘实现，有效减少了前传约束、资源浪费和处理时延，并降低泄露用户隐私数据的风险[4-5]，因此成了 6G 中无线接入网的解决方案。

越来越多的科研机构、社会机构和企业通过收集数据来达成希望完成的任务。数据收集者通过对用户设备上传的数据进行统计分析，能够从真实世界中获得更多知识，从而辅助决策。目前 6G 中热门的应用场景有智能电网、智能家居、智慧医疗、智能汽车等。然而，共享数据中通常包含许多人们不愿意透露给他人的隐私信息，如个人用电习惯、消费习惯、位置信息、医学诊断结果等私密性较强、较能反映个人特征的数据。在数据共享过程中，这些信息不可避免地会被泄露，甚至因此威胁到用户的生命财产安全。此外，数据的完整性也需要得到保证，数据只有正确、完整地存储下来，才能发挥作用。因此，数据完整性审计也是必不可少的一环。雾节点具有数据缓存功能，可以为用户提供共享数据的缓存服务。然而，尽管雾节点距离用户更近、与用户交互时延更低，但由于靠近网络边缘，它们难于管理，易于破坏。

因此，部署在 F-RAN 上的数据共享应用面临 2 个主要的问题：1) 共享数据可能包含不应该暴露给他人的敏感信息，需要一个方案在保护用户隐私的同时保证数据可用性；2) 存储在雾节点的数据必须保证完整性，由于存在软硬件损坏、人为错误等风险，需要对雾节点上的文件定期进行远程数据完整性审计。

对于传统的 C-RAN 架构，文献[6]采用同态可认证环签名，在隐藏用户身份的同时进行数据完整性审计，并支持无块验证。文献[7]使用类似的同态可认证群签名，并在签名前使用基于公钥的编码技术将数据编码成数据块以保护数据隐私。文献[8]在同态可认证环签名的基础上，采用覆盖树算法来确保数据隐私和新鲜度。文献[9]提出的轻量级数据共享方案基于在线/离线签名，并使用 Merkle 哈希树（MHT, Merkle hash tree）支持批量审计和数据动态操作。文献[10]提出的医疗数据共享方案采用基于身份的加密算法。文献[11]通过结合基于密钥同态加密的不经意伪随机函数和基于零知识证明的可验证性，实现数据集隐私保护聚合和共享。

现有网络设计之初缺乏架构级的安全考虑，安全防护依靠外挂式、补丁式的方案，因而无法实现全网的无缝安全通信保障[12]。构建安全可信的 6G 网络迫切需要内生的安全技术。目前，对于面向 6G 的 F-RAN 架构，还没有提出解决数据隐私保护和完整性审计问题的内生安全数据共享机制。

为此，本文基于面向 6G 的 F-RAN 设计了一种具有本地化差分隐私保护和动态数据完整性审计功能的数据共享机制，该机制针对 F-UE 向基带处理单元（BBU, baseband unit）池实现数据共享的过程。F-UE 负责采集或生成数据，以及数据隐私保护处理；F-AP 作为中间节点缓存并预处理共享数据；大功率节点（HPN, high power node）负责对 F-AP 上的缓存数据进行完整性审计，并负责控制信令的分发；BBU 池负责统计推断被保护数据的原始分布。本文的主要贡献介绍如下。

1) 从内生安全的角度出发，根据 F-RAN 的架构特点为机制进行保证数据安全的隐私保护和数据完整性审计技术选型，并对数据完整性审计技术进行了适当的改进，提高了签名和验证的性能。

2) 在 F-RAN 架构的基础上，利用 F-AP 缓存、计算的功能特点，提出了 F-UE 与 BBU 池间内生安全的数据共享机制。相比传统架构，基于 F-RAN 的机制降低了用户交互时延和远距离通信量，并保持了 F-RAN 通信层面的功能优势。在数据共享过

程中，F-UE 对数据运行 RAPPOR（randomized aggregatable privacy-preserving ordinal response）算法，接着 F-AP 对数据进行缓存和预处理，HPN 对各雾节点上的暂存数据进行基于 BLS 签名和 MHT 的动态完整性审计，最终 BBU 池通过统计分析，对收集数据的原始分布进行推断。

3) 对所提机制进行了安全性分析，分析表明提出的数据共享机制能够实现用户本地化差分隐私，并实现安全的数据完整性审计。本文将所提机制与已有机制进行了功能比较，仿真结果表明，所提机制的时间、空间和通信效率较高，同时能够保证隐私保护处理后的数据可用性。

2　预备知识

2.1　本地化差分隐私技术 RAPPOR

定义 1　本地化差分隐私。给定一种隐私算法 L，定义域和值域分别为 $\mathrm{Dom}(L)$ 和 $\mathrm{Ran}(L)$，给定 n 个用户，每个用户对应一条记录。对于任意两条记录 $t\in\mathrm{Dom}(L)$ 和 $t'\in\mathrm{Dom}(L)$，以及任意 $t^*\subseteq\mathrm{Ran}(L)$，若算法 L 满足式(1)，则称算法 L 满足 ε -本地化差分隐私[13-14]。

$$\Pr[L(t)=t^*]\leqslant e^{\varepsilon}\Pr[L(t')=t^*] \tag{1}$$

已知扰动概率 p 和总样本量 n，根据定义 1，隐私预算 ε 为

$$\varepsilon=\ln\frac{p}{1-p} \tag{2}$$

要实现定义 1 描述的 ε -本地化差分隐私，需要数据扰动机制的介入。本文采用 Google 已经投入实际使用的 RAPPOR[15]技术，该技术是一种保护隐私的数据收集技术，可以利用随机性来保证每个用户报告满足本地化差分隐私。

RAPPOR 技术的核心数据结构是 Bloom Filter。Bloom Filter 是一种随机数据结构，使用数组表示集合，数组的每一位只取 0 或 1，它能够确定元素是否属于此集合[16-17]。在没有元素加入时，Bloom Filter 的所有位都置为 0，令其长度为 k。Bloom Filter 使用 h 个相互独立的哈希函数来表示一个集合 $A=\{a_1,a_2,\cdots,a_n\}$，并分别将集合中的每个元素映射到 $\{1,\cdots,k\}$ 的范围中。对任意一个元素 a，第 i 个哈希函数映射的位置 $H_i(a)$（$1\leqslant i\leqslant h$）就会被置为 1。在判断一个元素 a^* 是否属于这个集合时，对 a^* 应用上述 h 个哈希函数，若所有 $H_i(a)$ 的位置都是 1，那么认为 a^* 是集合中的元素，否则认为 a^* 不是集合中的元素。

RAPPOR 基本算法在客户机上本地执行，用来保护数据隐私，具体如算法 1 所示。

算法 1　RAPPOR 基本算法

输入　客户机的真实值 a_0，客户机所属群组编号 cid，系统公共参数 $(f,p_0,p_1,n_{\mathrm{B}},n_{\mathrm{H}},H_{\mathrm{B}})$

1) 初始化 Bloom Filter。拼接真实值 a_0 与群组编号 cid，得到 $a=a_0\,\|\,\mathrm{cid}$，给定一长度为 n_{B} 的 Bloom Filter，记作 B，以哈希函数集合 H_{B} 中的前 n_{H} 个哈希函数作为 B 的哈希函数，并将值 a 加入 B 表示的集合。

2) 生成永久随机响应。对于每个客户端的值 a 和 B 中的位 i（$0\leqslant i\leqslant k$），创建一个二进制报告值 $\boldsymbol{B}_i'$ 为

$$\boldsymbol{B}_i'=\begin{cases}1, & \text{以概率}\dfrac{1}{2}f\\ 0, & \text{以概率}\dfrac{1}{2}f\\ \boldsymbol{B}_i, & \text{以概率}1-f\end{cases} \tag{3}$$

其中，f 是控制纵向隐私保护级别的用户可调参数。随后，这个 $\boldsymbol{B}_i'$ 被记录下来并被重用，作为以后所有关于值 a 的报告的基础。

3) 生成瞬时随机响应。分配大小为 n_{B} 的位数组 S，并将每一位初始化为 0。用概率设置每一位，即

$$P(S_i=1)=\begin{cases}p_1, & \boldsymbol{B}_i'=1\\ p_0, & \boldsymbol{B}_i'=0\end{cases} \tag{4}$$

4) 报告。将瞬时随机响应 S 发送到服务器。

在数据收集之前，设置 n_{C} 个群组，并将每个用户随机分配到 n_{C} 个群组之一。群组内部成员使用相同的 n_{H} 个哈希函数来实现 Bloom Filter，每个群组选择的 n_{H} 个哈希函数各不相同。

采用 RAPPOR 边缘解码算法从收集的 RAPPOR 报告中学习原始数据的边缘分布，如算法 2 所示。

算法 2　RAPPOR 边缘解码算法

1) 创建大小为 $n_{\mathrm{B}}n_{\mathrm{C}}\times M$ 的设计矩阵 $\boldsymbol{X}$，其中，M 为候选字符串数目（如图 1 所示，为 n_{C} 个群组各初始化 M 个大小为 n_{B} 的 Bloom Filter，$\boldsymbol{B}_{i,j}$ 为第 i 个候选字符串加入第 j 个群组的 Bloom Filter 后的位数组）。

2) 令 c_{ij} 为群组 j 中每个位 i 在一组 N_j 个报告中设置为 1 的次数，则群组 j 中每个位 i 在每个群

组中真正设置在 Bloom Filter 中的次数为

$$t_{ij}=\frac{c_{ij}-\left(p_0+\frac{1}{2}fp_1-\frac{1}{2}fp_0\right)N_j}{(1-f)(p_1-p_0)} \tag{5}$$

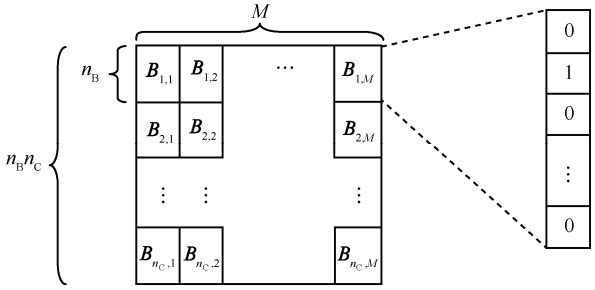


图 1 设计矩阵 $\boldsymbol{X}$ 示意

3) 设 $\boldsymbol{Y}$ 是 t_{ij} 的向量，$i\in[1,n_B], j\in[1,n_C]$。选择候选字符串使用 Lasso 回归[18]拟合模型 $\boldsymbol{Y}\sim\boldsymbol{X}$，并选择对应于非零系数的候选字符串。然后使用所选候选字符串拟合正则最小二乘回归，以估计各字符串的计数、标准误差和 P 值。

4) 确定哪些字符串出现的频率是从 0 开始的有统计学意义，将 P 值与 Bonferroni 校正后的 $\alpha/M=0.05/M$ 进行比较，或者使用 Benjamini-Hochberg 法将伪发现率（FDR, false discovery rate）控制在水平 α。

2.2 BLS 签名

定义 2 双线性映射。双线性映射 $e:G_1\times G_2\to G_T$，其中 G_1,G_2,G_T 是素数 q 阶乘法循环群。e 具有以下属性。1) 可计算：存在一种可有效计算 e 的算法；2) 双线性：对于任意 $h_1\in G_1$，$h_2\in G_2$ 和 $a,b\in\mathbb{Z}_q$，有 $e(h_1^a,h_2^b)=e(h_1,h_2)^{ab}$；3) 非退化性：存在 $g_1\in G_1$，$g_2\in G_2$，满足 $e(g_1,g_2)\neq 1_{G_T}$，其中，g_1 和 g_2 分别是 G_1 和 G_2 的生成元。如果 $G_1=G_2$，则称上述双线性配对是对称的，否则是非对称的。

BLS 签名方案需要如下函数[19]。1) 一个可精确计算的非退化配对 $e:G_1\times G_2\to G_T$，G_1,G_2,G_T 是素数 q 阶乘法循环群；2) 用于签名的哈希函数 $H_0:\mathcal{M}\to G_1$；3) 用于计算安全聚合与验证指数的哈希函数 $H_1:G_2^n\to R^n$，其中，$R:=\{1,2,\cdots,2^{128}\}$，$1\leqslant n\leqslant\tilde{N}$，$\tilde{N}$ 为公钥数目。

2.3 MHT

MHT 是经过充分研究并用于认证的数据结构，其目标是有效、安全地证明一组元素没有损坏和变化。它被构造为二叉树，其中 MHT 中的叶子是真实数据值的哈希值[20]。

数据元素的 MHT 身份验证如图 2 所示。首先，具有真实根节点 h_r 的验证者请求数据块 $\{x_3,x_6\}$，并要求对接收到的块进行身份验证。证明者除了向验证者提供块 $\{x_3,x_6\}$ 外，还向验证者提供辅助认证信息（AAI, auxiliary authentication information）$\Omega_3=<h(x_4),h_c>$ 和 $\Omega_6=<h(x_5),h_f>$。然后，验证者先后计算 $h(x_3)$，$h(x_6)$，$h_d=h(h(x_3)\|h(x_4))$，$h_e=h(h(x_5)\|h(x_6))$，$h_a=h(h_c\|h_d)$，$h_b=h(h_e\|h_f)$，

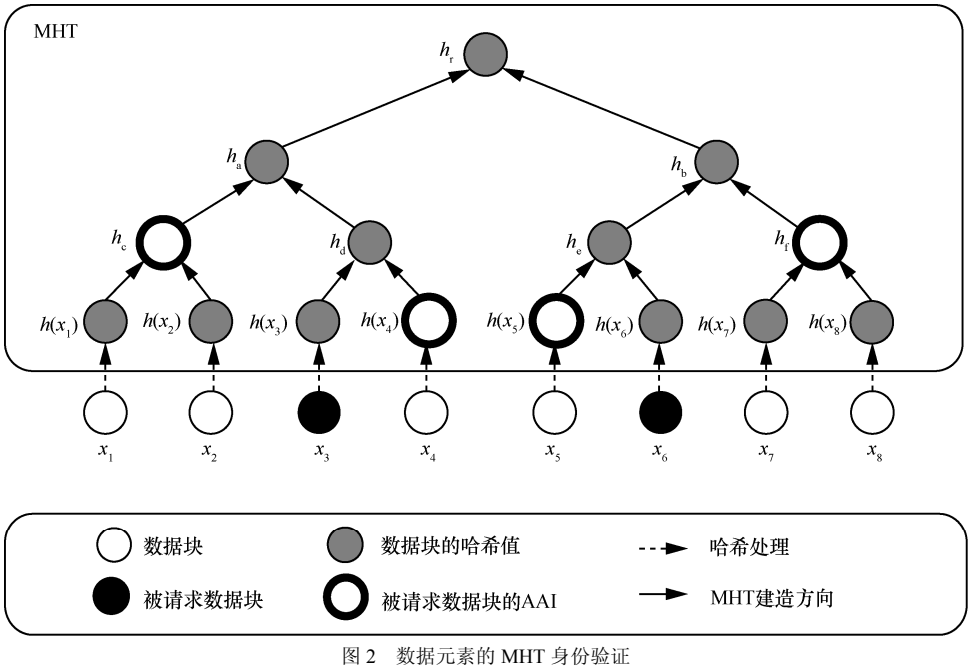


图 2 数据元素的 MHT 身份验证

$h_r = h(h_a \| h_b)$，通过检查计算的 h_r 是否与真实的 h_r 相同，来验证 $\{x_3, x_6\}$。

MHT 通常用于验证数据块的值，而本文进一步采用 MHT 来验证数据块的值和位置。将叶子节点视为从左到右的有序序列，因此可以通过遵循此序列以及 MHT 中计算根的方式来唯一确定任何叶子节点。

3　数据共享机制

3.1　系统模型

本文设计的数据共享机制的系统模型如图 3 所示，它基于面向 6G 的 F-RAN 架构。F-RAN 具有全局 C-RAN 模式、本地分布式协作模式、D2D 模式与 HPN 模式。本文主要关注 F-RAN 的本地分布式协作模式，涉及如下几类网络实体。

1) BBU 池。BBU 池可被视为中心机房，内部集中了大量基带处理单元。BBU 池具有集中式 CRSP 和 CRMM，减少了分散部署 BBU 带来的管理和维护成本，提高了网络频谱效率和能量效率。BBU 池通过多点协调（CoMP, coordinated multiple points）功能来抑制 HPN 和 F-AP 的跨层干扰。

2) F-AP。在 RRH 的射频处理功能基础上，F-AP 还具有 CRSP 和 CRMM 功能，以及额外的存储、计算功能。

3) F-UE。F-UE 是具有 CRSP 和 CRMM 以及存储功能的用户设备，但功能弱于 F-AP。

4) HPN。HPN 负责为所有的 F-UE 提供控制信令和小区特定参考信号，并为移动速率高的用户提供基本比特速率的无缝信号覆盖。

在 F-RAN 中，F-AP 具有一定的计算和缓存功能。通过将 F-UE 共享的数据文件缓存在网络边缘的 F-AP，并由 F-AP 对共享数据进行预处理后交付给 BBU 池，能够有效降低 F-UE 在数据共享过程中的交互时延，并减少 F-AP 与 BBU 池间的通信量。

在本文设定的数据共享过程中，F-RAN 工作在本地分布式协作模式，此时 F-UE 处于低速移动或静止状态。数据收集方委托 BBU 池收集 F-UE 上传的数据。各实体协商公共参数后，F-UE 生成共享

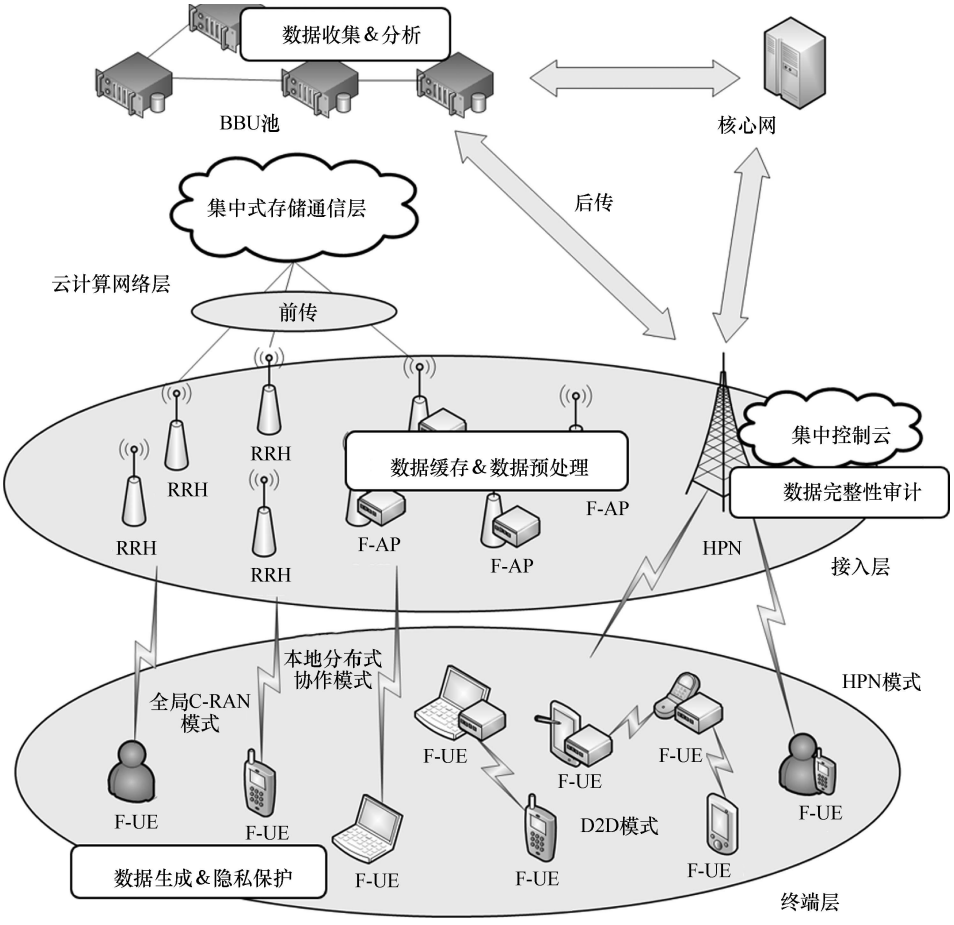


图 3　基于面向 6G 的 F-RAN 网络架构的系统模型

数据，进行隐私保护处理后将数据共享给邻近的 F-AP。F-AP 负责为 F-UE 缓存数据。在此期间，出于应用目的，F-UE 可以与 F-AP 进行交互，以访问或检索其预存储的数据，F-UE 还可以对已上传的数据进行修改、插入和删除。HPN 负责对缓存在 F-AP 上的数据进行数据完整性审计。待 BBU 池通过负责控制信令分发的 HPN 下达上传指令后，F-AP 首先停止收集数据，并停止响应用户数据更新的请求，然后对收到的数据集进行一定的数据预处理后上传到 BBU 池。在 BBU 池获得数据后，统计推断得到数据实际的分布，交付给数据收集方进行科学研究等。

在上述数据共享过程的通信层面，各实体仍保持其通信功能，包括 F-UE、F-AP 的 CRSP 和 CRMM 功能，BBU 池的基带处理、集中式 CRSP、CRMM 和 CoMP 功能，HPN 的控制信令分发功能等。

3.2 数据共享机制设计

本节介绍数据共享机制的详细设计，记 F-UE 数量为 n_{U}，F-AP 数量为 n_{A}，且 $n_{\mathrm{U}} >> n_{\mathrm{A}}$，BBU 池与 HPN 数量均为 1。

阶段 1 系统设定

KeyGen(1^k)。F-UE 生成一个随机签名密钥对(spk,ssk)。选择一个随机的 $\alpha \xleftarrow{R} \mathbb{Z}_q$ 并计算 $v \leftarrow g_2^{\alpha}$。密钥为 $\mathrm{sk}=(\alpha,\mathrm{ssk})$，公钥为 $\mathrm{pk}=(v,\mathrm{spk})$。将公钥 pk 与自己的身份标识符 uid 发送给 F-AP 与 HPN。

SetCID(uid,n_{C})。F-AP 收到 F-UE 的身份标识符 uid 后，根据群组数目 n_{C}，输出 uid 所属的群组编号 cid。F-AP 将 cid 发送给 F-UE。

阶段 2 数据发布

RAPPOR(m_i^0,pub_i,cid)。给定需要隐私保护的数据块 m_i^0，F-UE 根据 BBU 池给定第 i 个数据块的公共参数 $\mathrm{pub}_i=(f,p_0,p_1,n_{\mathrm{B}},n_{\mathrm{H}},n_{\mathrm{C}},H_{\mathrm{B}})$ 以及所属群组编号 cid，确定 Bloom Filter 的长度和使用的哈希函数。接着，对 m_i^0 执行基本 RAPPOR 算法（如 3.1 节所述），生成随机化数据块 m_i。通过对每一个需要保护的数据块 m_i^0 进行上述处理，不需要隐私保护的数据块保持不变，F-UE 生成隐私处理后的文件 $F=(m_1,m_2,\cdots,m_n)$。

SigGen(sk,F)。给定 $F=(m_1,m_2,\cdots,m_n)$，F-UE 随机选择 $u \leftarrow G_1$。令 F 的文件标签为 $\mathrm{tag}=\mathrm{name}\|n\|u\|\mathrm{SSig}_{\mathrm{ssk}}(\mathrm{name}\|n\|u)$。接着，F-UE 为每个块 $m_i\ (i=1,2,\cdots,n)$ 计算签名 $\sigma_i \leftarrow (H(m_i))^{\alpha}$。用 $\Phi=\{\sigma_i\}_{i=1}^{n}$ 表示签名集。然后，F-UE 根据 MHT 的构造生成根 R，其中树的叶子节点是 $H(m_i)\ (i=1,2,\cdots,n)$ 的有序散列集。然后，F-UE 在私钥 α 下签署根 R：$\mathrm{sig}_{\mathrm{sk}}(H(R)) \leftarrow (H(R))^{\alpha}$。最后，F-UE 发送 $(F,\mathrm{tag},\Phi,\mathrm{sig}_{\mathrm{sk}}(H(R)))$ 到 F-AP。

阶段 3 数据完整性审计

GenProof(F,tag,Φ,chal)。收到文件标签 tag 和挑战 chal=$\{i\}_{s_1\leqslant i\leqslant s_c}$ 后，F-AP 检索对应文件，计算聚合签名。如果 chal 指定的所有数据块 $m_{s_1},\cdots,m_{s_c}$ 都是不同的，计算简单聚合签名 $\sigma=\sigma_{s_1}\cdots\sigma_{s_c}\in G_1$。若存在相同数据块，记发生碰撞的挑战数据块 $\{i\}_{s_1'\leqslant i\leqslant s_{c'}'}$，对发生碰撞的数据块计算指数 $(t_1,\cdots,t_{c'}) \leftarrow H_1(v_{s_1'},\cdots,v_{s_{c'}'})\in R^n$，并输出安全聚合签名 $\sigma' \leftarrow \sigma_{s_1'}^{t_1}\cdots\sigma_{s_{c'}'}^{t_c}\in G_1$。最后将简单聚合签名和安全聚合签名进行简单聚合。另外，F-AP 还将向验证者提供少量辅助信息 $\{\Omega_i\}_{s_1\leqslant i\leqslant s_c}$，它们是从叶子 $\{H(m_i)\}_{s_1\leqslant i\leqslant s_c}$ 到 MHT 的根 R 的路径上的节点同级物。最后，F-AP 证明 $P=(\sigma,\{H(m_i),\Omega_i\}_{s_1\leqslant i\leqslant s_c},\mathrm{sig}_{\mathrm{sk}}(H(R)))$ 以响应 HPN。

VerifyProof(pk,chal,P,tag)。收到 F-AP 的响应后，HPN 将使用 $\{H(m_i),\Omega_i\}_{s_1\leqslant i\leqslant s_c}$ 生成根 R，并通过检查式(5)是否成立来进行身份验证。

$$e(g_2,\mathrm{sig}_{\mathrm{sk}}(H(R)))=e(g_2^{\alpha},H(R)) \tag{5}$$

如果认证失败，则 HPN 通过发出 FALSE 拒绝。否则，HPN 检查聚合签名。如果所有数据块 $m_{s_1},\cdots,m_{s_c}$ 都是不同的，通过检查式(6)来验证聚合签名。

$$e(g_2,\sigma)=e(v_{s_1},H_0(m_{s_1}))\cdots e(v_{s_c},H_0(m_{s_c})) \tag{6}$$

若存在相同数据块，记发生碰撞的数据块数量为 c' 个，索引为 chal=$\{i\}_{s_1'\leqslant i\leqslant s_{c'}'}$。对于发生碰撞的数据块 m 计算指数 $(t_{s_1'},\cdots,t_{s_{c'}'}) \leftarrow H_1(v_{s_1'},\cdots,v_{s_{c'}'})\in R^n$，计算聚合公钥 $\mathrm{apk} \leftarrow v_{s_1'}^{t_1}\cdots v_{s_{c'}'}^{t_{c'}}\in G_2$。若满足式(7)，则输出 TRUE，否则为 FALSE。

$$e(g_2,\sigma)=e(\mathrm{apk},H_0(m)) \tag{7}$$

BatchVerifyProof(pks,chals,Ps)。为了提高效率，HPN 可以对不同 F-AP 给出的关于不同 F-UE 数据文件的证明 P 进行批量验证，pks、chals 和 Ps 分别为参与批量验证的公钥集合、挑战集合和数据

存在证明集合。具体步骤与 VerifyProof()相似，只是在验证聚合签名时，VerifyProof()中公钥$(v_{s_1},\cdots,v_{s_c})$均来自同一用户，因此有$v_{s_1}=v_{s_2}\cdots=v_{s_c}$；而在本算法中，公钥$(v_{s_1},\cdots,v_{s_c})$可以各不相同。

ExecUpdate($F,\Phi,$update)。假设 F-UE 要在第i个块m_i处进行更新操作$\mathcal{X}$，包括修改（$\mathcal{U}$）、插入（$\mathcal{I}$）和删除（$\mathcal{D}$），分别对应于将m_i修改为m_i'、在m_i处插入新的块m_i'、删除m_i。开始时，F-UE 基于新的块m_i'生成相应的签名$\sigma_i'=(H(m_i'))^{\alpha}$（删除操作不需要该步骤）。然后，F-UE 构造一个更新请求消息“update$=(\mathcal{X},i,m_i',\sigma_i')$”并发送到 F-AP。收到请求后，F-AP 将运行 ExecUpdate($F,\Phi,$update)。

若$\mathcal{X}=\mathcal{U}$，F-AP 将块m_i替换为m_i'并输出F'，将σ_i替换为σ_i'并输出Φ'。接着在 MHT 构造中用$H(m_i')$替换$H(m_i)$，并生成新的根R'。最后，F-AP 使用证明$P_{\text{update}}=(\Omega_i,H(m_i),\text{sig}_{\text{sk}}(H(R)),R')$响应 F-UE，证明 F-AP 完成了指定的修改操作，其中Ω_i是用于m_i身份验证的 AAI。

若$\mathcal{X}=\mathcal{I}$，F-AP 存储m_i'并在 MHT 中$h(H(m_i))$后添加叶子$h(H(m_i'))$，重构 MHT，并输出F'，将σ_i'加入签名集Φ'并输出。接着根据更新的 MHT 生成新的根R'。最后，F-AP 使用证明$P_{\text{update}}=(\Omega_i,H(m_i),\Omega_i',\text{sig}_{\text{sk}}(H(R)),R')$响应 F-UE。其中$\Omega_i$、$\Omega_i'$分别是用于$m_i$、$m_i'$身份验证的 AAI。

若$\mathcal{X}=\mathcal{D}$，F-AP 删除m_i并在 MHT 中删除叶子$h(H(m_i))$重构 MHT，并输出F'。将σ_i移出签名集Φ并输出Φ'。根据更新的 MHT 生成新的根R'。最后，F-AP 使用$P_{\text{update}}=(\Omega_i,H(m_i),\Omega_i',H(m_i'),\text{sig}_{\text{sk}}(H(R)),R')$响应 F-UE。记$m_i'$为删除$m_i$后取代其位置的数据块，$P_{\text{update}}$中$\Omega_i$、$\Omega_i'$分别是用于$m_i$、$m_i'$身份验证的 AAI。需要注意的是，若$m_i$是最后一个数据块，则$m_i'$取其前一数据块，若删除$m_i$后没有其他数据块，则$m_i'$、$\Omega_i'$、$R'$都取空值。

VerifyUpdate(pk,$\text{sig}_{\text{sk}}(H(R))$,update,$P_{\text{update}}$)。从 F-AP 接收到修改操作的证明后，F-UE 运行 VerifyUpdate(pk,$\text{sig}_{\text{sk}}(H(R))$,update,$P_{\text{update}}$)。

1) F-UE 使用$\{\Omega_i,H(m_i)\}$生成根R，并通过检查$e(g_2,\text{sig}_{\text{sk}}(H(R)))=e(g_2^{\alpha},H(R))$是否成立来验证 AAI 或$R$。如果不正确，则输出 FALSE，否则进行步骤 2)。

2) 若$\mathcal{X}=\mathcal{U}$，F-UE 使用$\{\Omega_i,H(m_i')\}$进一步计算新的根值，并将其与R'进行比较，来检查 F-AP 是否已按照要求执行了修改。如果不是，则输出 FALSE，否则输出 TRUE，并执行步骤 3)。

若$\mathcal{X}=\mathcal{I}$或$\mathcal{X}=\mathcal{D}$，F-UE 使用$\{\Omega_i',H(m_i')\}$进一步计算新的根值，并将其与R'进行比较，来检查 F-AP 是否按照要求执行了插入或删除操作。如果不是，则输出 FALSE，否则输出 TRUE，并执行步骤 3)。

3) F-UE 通过$\text{sig}_{\text{sk}}(H(R'))$对新的根元数据$R'$进行签名，并将其发送到 F-AP 以进行更新。最后，F-UE 请求 HPN 执行默认的完整性验证协议。如果输出为 TRUE，则从本地存储删除$\text{sig}_{\text{sk}}(H(R'))$和$P_{\text{update}}$，若$\mathcal{X}=\mathcal{U}$或$\mathcal{X}=\mathcal{I}$，删除$m_i'$，若$\mathcal{X}=\mathcal{D}$，则删除$m_i$。

阶段 4　数据收集

$\text{SumBits}_{\text{F-AP}}(M_i,\text{Cid}_i)$。F-AP 根据对应数据块的群组编号列表$\text{Cid}_i=\{\text{cid}_{i,u}\}_{u=1}^{n_{\text{U}}}$，对收集的所有$n_{\text{U}}$个 F-UE 的第$i$个随机化数据块列表$M_i=\{m_{i,u}\}_{u=1}^{n_{\text{U}}}$按群组编号进行比特求和，得到比特数组列表$\text{counts}_i$，并发送给 BBU 池。

$\text{SumBits}_{\text{BBU}}(\text{Counts}_i,\text{CID}_i)$。BBU 池根据对应数据块的群组编号列表$\text{CID}_i=\{\text{Cid}_{i,w}\}_{w=1}^{n_{\text{A}}}$，对收集的来自$n_{\text{A}}$个 F-AP 的比特数组列表$\text{Counts}_i=\{\text{counts}_{i,w}\}_{w=1}^{n_{\text{A}}}$按群组编号进行求和，得到最终进行数据分析的比特数组列表finalcounts_i。

HashCandidates($n_{\text{B}},n_{\text{H}},n_{\text{C}},H_{\text{B}},\text{candidates}_i$)。BBU 池根据第$i$个数据块对应的参数$(n_{\text{B}},n_{\text{H}},n_{\text{C}},H_{\text{B}})$和候选字符串列表$\text{candidates}_i$生成设计矩阵$\text{map}_i$（详见 2.1 节算法 2 的步骤 1)）。

Decode($\text{map}_i,\text{finalcounts}_i,\text{pub}_i$)。BBU 池根据设计矩阵$\text{map}_i$，比特数组按群组编号求和的列表$\text{finalcounts}_i$，公共参数$\text{pub}_i$，构造模型$\boldsymbol{Y}\sim\boldsymbol{X}$并对其进行 Lasso 回归，以选择对应于非零系数的候选字符串，对选择的候选字符串进一步进行正则最小二乘回归，统计出数据块m_i的真实分布Distr_i（详见 2.1 节算法 2 步骤 2)~步骤 5)）。

4　安全性

本文提出的数据共享机制的安全性基于 RAPPOR 基本算法和数据完整性审计机制。

RAPPOR 算法的安全性分析在文献[15]已经有完整的阐述，本节只关注数据完整性审计部分。

4.1 安全模型

对于数据完整性审计方案来说，数据安全的关键在于审计者 HPN 是否能切实判断数据在 F-AP 上存储的实际情况。

根据文献[20]中的安全模型定义，本文给出形式化的安全模型，主要关注 4 种算法，KeyGen()、SigGen()、GenProof()和 VerifyProof()，它们的行为如 3.2 节所述。将执行 GenProof()与 VerifyProof()两台机器的运行表示为 $\{\text{TRUE,FALSE}\}\leftarrow(\text{VerifyProof}(\text{pk,chal},P,\text{tag})\rightleftharpoons\text{GenProof}(F,\text{tag},\Phi,\text{chal}))$。

本文希望数据完整性审计协议是正确且可靠的。正确性要求对于 KeyGen()输出的所有(pk,sk)，以及 SigGen()输出的所有$(F,\text{tag},\Phi,\text{sig}_{\text{sk}}(H(R)))$，VerifyProof()在与有效 GenProof()交互时接受式(8)。

$$(\text{VerifyProof}(\text{pk,chal},P,\text{tag})\rightleftharpoons \text{GenProof}(F,\text{tag},\Phi,\text{chal}))=\text{TRUE} \tag{8}$$

将具有审计能力的 HPN 和 F-UE 看作共同的挑战者 $\mathcal{C}$，将不受信任的 F-AP 看作对手 $\mathcal{A}$。考虑对手 $\mathcal{A}$ 和挑战者 $\mathcal{C}$ 之间的博弈：挑战者 $\mathcal{C}$ 通过运行 KeyGen()生成密钥对(pk,sk)，并向 $\mathcal{A}$ 提供 pk。现在对手 $\mathcal{A}$ 可以与挑战者 $\mathcal{C}$ 交互，也可以向 SigGen()进行查询，为每个查询提供一些文件 F。挑战者 $\mathcal{C}$ 计算 SigGen(sk,F)，并返回所有输出$(F,\text{tag},\Phi,\text{sig}_{\text{sk}}(H(R)))$给对手 $\mathcal{A}$。对于之前对手 $\mathcal{A}$ 向挑战者 $\mathcal{C}$ 进行查询的任何文件 F，对手 $\mathcal{A}$ 可以扮演验证者，通过指定相应的标记 tag 来执行 $\text{VerifyProof}(\text{pk,chal},P,\text{tag})\rightleftharpoons\mathcal{A}$，即执行数据完整性审计协议。当协议执行完成时，对手 $\mathcal{A}$ 向对方提供 VerifyProof()的输出。这些协议执行可以相互任意交错，并与文件 F 的查询同时进行。最后，对手 $\mathcal{A}$ 输出从某个查询返回的标签 tag，以及证明算法 $\text{GenProof}^*()$。

如果作弊证明者能令人信服地回答挑战 chal 的 δ 部分，即如果满足式(9)，则它是 δ-可接受的。

$$P[(\text{VerifyProof}(\text{pk,chal},P,\text{tag})\rightleftharpoons \text{GenProof}^*())=\text{TRUE}]\geqslant\delta \tag{9}$$

定义 3 和定义 4 描述数据完整性审计协议的可靠性要求。令 F 为查询中输入的文件，查询返回$(F,\text{tag},\Phi,\text{sig}_{\text{sk}}(H(R)))$。

定义 3 如果存在一个提取算法 Extr()，对于每个对手 $\mathcal{A}$，每当 $\mathcal{A}$ 运行上述博弈，为文件 F 输出一个 δ-可接受的作弊证明算法 $\text{GenProof}^*()$，满足 $\text{Extr}(\text{pk},\Phi,\text{chal},\text{tag},\text{GenProof}^*())=F$，即 Extr() 从 $\text{GenProof}^*()$ 中恢复 F，那么就说一个可检索性证明方案是 δ-完备的，除非式(9)可以忽略不计。

定义 4 如果不存在能够以不可忽略的概率欺骗验证者的多项式时间算法，则数据完整性审计协议是安全的。

4.2 安全性分析

根据上述安全模型来评估本文提出的数据完整性审计方案的安全性，由于该方案基于文献[20]，因此证明过程除了审计细节外，大体上与其相似。BLS 签名的安全性证明在文献[19]中已经给出。

定理 1 如果签名方案在本质上是不可伪造的，并且在双线性群中 CDH 问题很难解决，那么任何反对本文公开审计方案合理性的对手都不能导致验证者以不可忽略的概率接受可检索性协议实例，除非对手使用正确计算的值做出响应。

证明 在文献[19]中的 BLS 签名方案是安全的前提下，容易证明本文的数据完整性审计方案所要求的数据完整性证明是不可伪造的。

首先假设 BLS 签名方案是安全的。设 $P=(\sigma,\{H(m_i),\Omega_i\}_{s_1\leqslant i\leqslant s_c},\text{sig}_{\text{sk}}(H(R)))$ 为从诚实的验证者获得的预期响应，其中 $H_0(m_i)$ 的正确性可以通过 $\{H(m_i),\Omega_i\}_{s_1\leqslant i\leqslant s_c}$ 和 $\text{sig}_{\text{sk}}(H(R))$ 来验证，签名 σ 的正确性根据式(10)和式(11)验证，m 为碰撞数据块。

$$e(g_2,\sigma)=e(v_{s_1},H_0(m_{s_1}))\cdots e(v_{s_c},H_0(m_{s_c})) \tag{10}$$

$$e(g_2,\sigma)=e(\text{apk},H_0(m)) \tag{11}$$

设 $P'=(\sigma',\{H_0(m_i'),\Omega_i'\}_{s_1\leqslant i\leqslant s_c},\text{sig}_{\text{sk}}(H(R)))$ 为对手的响应，P' 在 MHT 中的认证结果应该与 $\{H_0(m_i),\Omega_i\}_{s_1\leqslant i\leqslant s_c}$ 与 $\text{sig}_{\text{sk}}(H(R))$ 的认证结果相同。用式(12)和式(13)验证 σ'。

$$e(g_2,\sigma')=e(v_{s_1},H_0(m_{s_1}'))\cdots e(v_{s_c},H_0(m_{s_c}')) \tag{12}$$

$$e(g_2,\sigma')=e(\text{apk},H_0(m')) \tag{13}$$

显然，若式(12)和式(13)成立，那么 $H_0(m_i')\neq H_0(m_i)$，否则 $\sigma'=\sigma$，这与本文的假设相矛盾。若 $H_0(m_i')\neq H_0(m_i)$，那么由 $H_0(m_i')$ 与 Ω_i' 对 MHT 的验证将失败。

表 1　　相关方案功能比较

功能	数据共享	远程数据完整性审计	公共审计	动态审计	批量审计	隐私保护	隐私保护数据可用性	本地化隐私保护
文献[6]	√	√	√	√	√	√	√	×
文献[10]	√	√	√	×	×	√	×	×
文献[11]	√	√	×	×	×	√	√	×
本文方案	√	√	√	√	√	√	√	√

5　仿真与评价

5.1　功能比较

由于目前对 F-RAN 的数据共享机制研究还很少，表 1 给出本文提出的数据共享方案（以下简称本文方案）和已有的传统云架构上的几个数据共享方案进行对比。如表 1 所示，本文方案是唯一支持数据共享、远程数据完整性审计、公共审计、动态审计、批量审计、隐私保护、隐私保护数据可用性和本地化隐私保护的方案。注意，文献[6]、文献[10]、文献[11]均不支持本地化隐私保护，且均需要由用户之外的第三方完成隐私保护处理。表 1 中√表示支持，×表示不支持。

5.2　性能分析与比较

本文方案主要是基于 BLS 签名和 MHT 的动态审计方案[20]，由于其在聚合签名和批量验证时有一定的缺陷，因此本文结合 Dan Boneh 在 2018 年提出的公钥聚合 BLS 多签名方案对其进行了改进，并优化了 MHT 在动态操作时的响应方式。表 2 展示了文献[20]方案的细节。

表 2　文献[20]方案的数据完整性审计方案细节

步骤	实现细节
生成签名	$\sigma_i \leftarrow (H(m_i)u^{m_i})^{\alpha}$，$u \in G_1$ 为用户随机值
发送挑战	$\text{chal} = \{(i, w_i)\}_{s_1 \le i \le s_c}$，其中 $w_i \xleftarrow{R} \mathbb{Z}_q$
签名聚合	$\sigma \leftarrow \sigma_{s_1}^{w_{s_1}} \cdots \sigma_{s_c}^{w_{s_c}} \in G_1$
辅助签名验证	$\mu = \sum_{i=s_1}^{s_c} w_i m_i \in \mathbb{Z}_q$
签名验证	$e(g_2, \sigma) = e(v, \prod_{i=1}^{n} H_0(m_i)^{w_i} u^{\mu})$
MHT 实现	结构体

文献[20]方案中签名和验证的计算量比本文方案要大，分别具有额外项 u^{m_i} 和 u^{μ}，且在聚合签名时，对于所有数据块均需要计算指数，本文方案只有发生碰撞的数据块需要计算指数。此外，文献[20]方案的通信量也比本文方案大，其传输每个数据块时均需要额外传输随机值 w_i。

对于 MHT，文献[20]方案采用结构体实现，而本文方案采用不定长二维数组实现。一方面，使用二维数组能够减少树占用的存储空间。结构体表示的节点需要存储本节点的值以及父节点、孩子节点的地址，而二维数组中每个节点仅需存储本节点的值。另一方面，使用二维数组可以直接使用索引访问 MHT 的底层节点，时间复杂度为 $O(1)$，而结构体实现的 MHT 所需的时间复杂度最坏取决于 MHT 的深度和底层节点的个数。

在更新操作时，需要先查找到 MHT 对应的节点，再更新底层节点，并计算出顶层根节点。例如，在底层节点后插入一个新节点，令 n 为 MHT 节点总数，k 为底层节点个数。文献[20]方案访问目标节点的时间复杂度最坏为 $O(\log n)$ 或 $O(k)$，插入操作的时间复杂度为 $O(1)$，自底向上计算根节点的时间复杂度为 $O(\log n)$。本文方案访问底层节点的时间复杂度为 $O(1)$，插入操作的时间复杂度为 $O(k)$，自底向上重建 MHT 并计算根节点的时间复杂度为 $O(\log n)$。因此，在进行树的更新时，2 种方案的时间复杂度相差不大，均取 $O(\log n)$ 与 $O(k)$ 中的较大值，但在访问特定节点的值时本文方案更具有优势。

尽管本文方案与文献[20]方案的时间复杂度持平，但考虑到 F-UE 对已上传的数据进行动态操作并不会很频繁，节省本就有限的存储空间显得更加重要。因此，在本文方案的系统设定下，选择不定长二维数组更佳。

5.3　仿真分析

通过仿真来评估本文方案的时间性能。本文采用的仿真机器为 Intel Core i5-8250U 1.60 GHz 处理器，16 GB 内存，Windows 10 操作系统。仿真采用 Python 语言和 R 语言编程，IDE 分别采用 Spyder 和 RStudio，Python 版本为 3.6.10，R 版本为 3.6.3。需要使用额外的 Python 依赖库 pandas、pysha3 1.0b1；

额外的 R 依赖库 glmnet、limSolve。在仿真中，设置基本数据块 $m_i \in \mathcal{M}$（$i \in [1,n]$，n 为文件数据块数量）所占内存大小为 28 B，MHT 的各个节点大小为 49 B，$\mathbb{Z}_q$ 中一个元素的大小为 1 056 B。

1) 签名方案的性能

首先对密钥生成算法进行仿真。重复运行 100 次 KeyGen()，得到 100 次运行的平均值为 0.05 s。运行 100 次 KeyGen()每次耗时情况如图 4 所示。

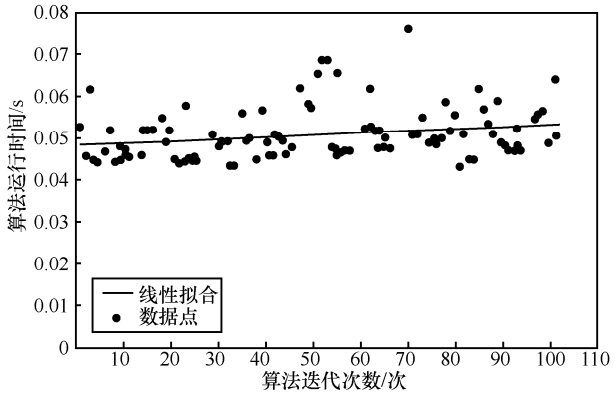


图 4　运行 100 次 KeyGen()每次耗时情况

接着对签名生成算法进行仿真，生成从 0 到 1 000 个不同数据块数的签名，每次运行数据块增加 100 个。仿真结果如图 5 所示，数据块数量 x 和 SigGen()的运行时间 y 基本呈线性关系，拟合的线性关系式为 $y = 0.438\,9x + 0.727\,3$，$R^2 = 0.999\,3$，这表明拟合接近实际情况。从拟合的线性关系式中可以看出，每增加一个数据块，SigGen()的运行时间将增加约 0.44 s。

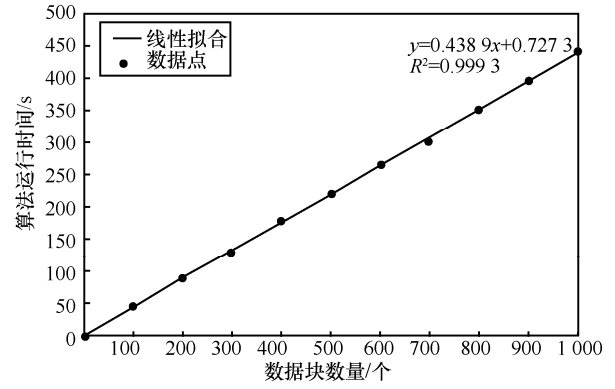


图 5　数据块数量对 SigGen()算法运行时间的影响

2) 批量审计性能

分别运行若干次 VerifyProof()和 BatchVerifyProof()，探究随着签名数目的增加，逐个验证签名和批量验证签名的耗时变化以及二者之间的对比。仿真生成从 0 到 100 个来自不同 F-UE 的单个数据块签名，每次运行 F-UE 数目增加 10 个。仿真结果如图 6 所示。

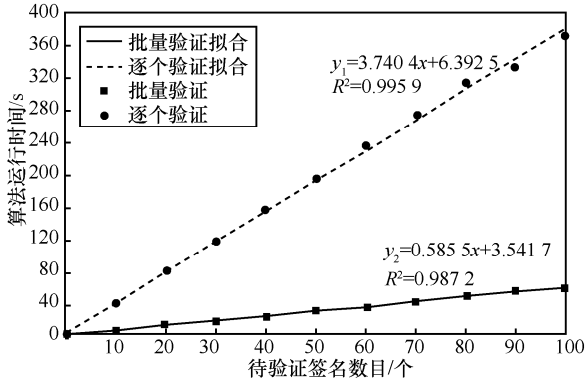


图 6　验证签名数目对 VerifyProof()与 BatchVerifyProof()算法运行时间的影响

图 6 中，两组散点分别表示对 x 个签名进行一次批量验证和逐个验证花费时间 y s，连线是对散点图的线性拟合。可以看到，无论是批量验证还是逐个验证，两条直线拟合的 R^2 均非常接近 1，可以认为随着签名数目的增加，算法运行时间大致上呈线性增长。且二者线性拟合表达式的斜率相差较大，随着 x 增加，$\Delta y = y_1 - y_2 = 3.154\,9x + 2.850\,8$。这表明逐个验证与批量验证的耗时差距将随着签名数目的增加而增大，批量验证效率更高。

3) 动态审计性能

对修改、插入、删除这 3 种操作分别模拟 10 次动态操作执行 ExecUpdate()和 10 次动态操作验证 VerifyUpdate()，原文件数据块数量为 110，进行动态操作的数据块数量从 0 到 100，每一次仿真数据块数量增加 10 个，仿真结果分别如图 7 和图 8 所示。

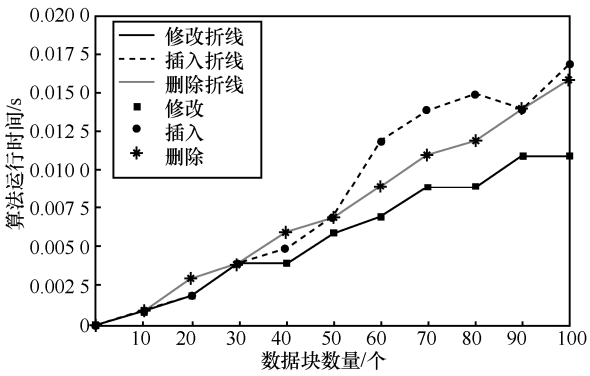


图 7　数据块数量对算法 ExecUpdate()的修改、插入、删除操作耗时的影响

从图 7 可以看到，对于修改、插入和删除这 3

种操作，ExecUpdate()的耗时均随着动态操作的数据块的增加而增大，二者大致呈线性关系，且当动态操作的数据块数量相同时，3 种操作的耗时比较接近。从图 8 可以看到，对于修改、插入和删除这 3 种操作，除了数据块数量为 0 时，其余 VerifyUpdate()的耗时均与动态操作的数据块数量没有明显关系，随着动态操作数据块数量的增加，耗时几乎保持不变。除去数据块数量为 0 的点外，修改、插入和删除 3 种操作的平均耗时分别为 4.18 s、4.19 s 和 4.20 s，非常接近。

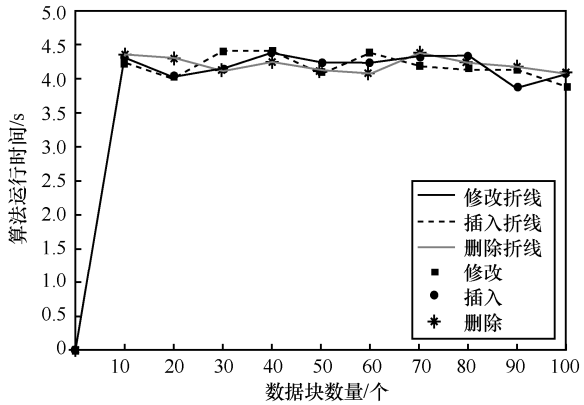


图 8　数据块数量对算法 VerifyUpdate()的修改、插入、删除操作耗时的影响

4) 隐私保护性能

对 RAPPOR()进行仿真，数据块数量从 0 到 50 000 变化，每次迭代数据块数量增加 1 000，仿真结果如图 9 所示。从图 9 可以看到，RAPPOR 算法的耗时与数据块数量呈线性关系，每增加一个数据块，时间大约增加 7×10^{-4} s，因此对于 F-UE 来说，进行隐私保护所花费的时间成本较低。

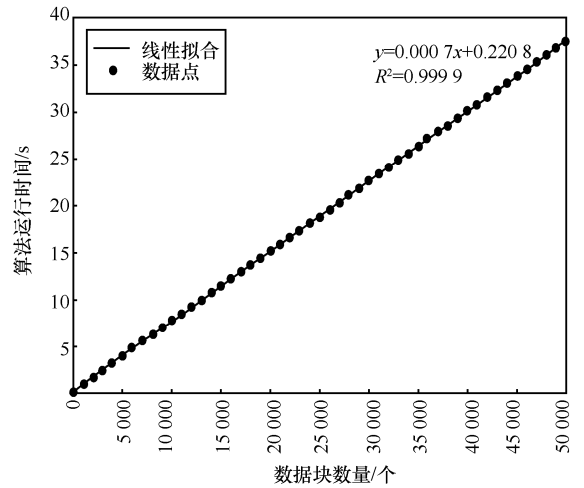


图 9　数据块数量对 RAPPOR()隐私保护处理耗时的影响

对于边缘解码算法 Decode()的统计推断效果，文献[15]中已经给出了详细的仿真结果和分析，本文不再赘述。

6　结束语

本文针对 6G 时代发展前景广阔的 F-RAN 架构，提出一种具有本地化差分隐私保护和动态数据完整性审计功能的内生安全数据共享机制。基于 F-RAN 的本地分布式协作模式，建立机制运行的系统模型。数据共享时，F-UE 本地对数据运行 RAPPOR 隐私保护算法；F-AP 对数据暂时存储和预处理；HPN 对各 F-AP 上暂存数据进行基于 BLS 和 MHT 的完整性审计，F-UE 可对 F-AP 上数据进行动态操作与相应审计；最终 BBU 池通过统计分析，对收集数据的原始分布进行推断。理论分析与仿真结果表明，本文提出的内生安全数据共享机制能够在保持较高时间、空间和通信效率的同时，实现内生安全的动态审计和多客户端批量审计，并在实现用户本地化差分隐私的同时保证数据可用性。

参考文献：

[1] 赵亚军, 郁光辉, 徐汉青. 6G 移动通信网络: 愿景、挑战与关键技术[J]. 中国科学: 信息科学, 2019, 49(8): 963-987.
ZHAO Y J, YU G H, XU H Q. 6G mobile communication networks: vision, challenges, and key technologies[J]. Scientia Sinica(Informationis), 2019, 49(8): 963-987.

[2] 张平, 牛凯, 田辉, 等. 6G 移动通信技术展望[J]. 通信学报, 2019, 40(1): 141-148.
ZHANG P, NIU K, TIAN H, et al. Technology prospect of 6G mobile communications[J]. Journal on Communications, 2019, 40(1): 141-148.

[3] 尹博南, 艾元, 彭木根. 雾无线接入网: 架构、原理和挑战[J]. 电信科学, 2016, 32(6): 20-27.
YIN B N, AI Y, PENG M G. Fog computing based radio access networks:architecture, principles and challenges[J]. Telecommunications Science, 2016, 32(6): 20-27.

[4] PENG M, ZHANG K. Recent advances in fog radio access networks: performance analysis and radio resource allocation[J]. IEEE Access, 2016, 4: 5003-5009.

[5] 刘铎, 杨涓, 谭玉娟. 边缘存储的发展现状与挑战[J]. 中兴通讯技术, 2019, 25(3): 15-22.
LIU D, YANG J, TAN Y J. A survey on the storage issues in edge computing[J]. ZTE Technology Journal, 2019, 25(3): 15-22.

[6] WANG B, LI B, LI H. Oruta: privacy-preserving public auditing for shared data in the cloud[J]. IEEE Transactions on Cloud Computing, 2014, 2(1): 43-56.

[7] 唐春明, 郑晓龙. 云计算中一种对大群组用户的隐私保护公共审计方案[J]. 信息网络安全, 2015(2): 19-25.
TANG C M, ZHENG X L. A privacy-preserving public auditing me-

chanism for the date with large groups users in the cloud computing[J]. Netinfo Security, 2015(2): 19-25.

[8] TRUEMAN T E, NARAYANASAMY P. Ensuring privacy and data freshness for public auditing of shared data in cloud[C]//IEEE International Conference on Cloud Computing in Emerging Markets. Piscataway: IEEE Press, 2016: 22-27.

[9] LI J, ZHANG L, LIU J K, et al. Privacy-preserving public auditing protocol for low-performance end devices in cloud[J]. IEEE Transactions on Information Forensics & Security, 2016, 11(11): 2572-2583.

[10] SHEN W, QIN J, YU J, et al. Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2018, 14(2): 331-346.

[11] LIM H W, POH G S, XU J, et al. PrivateLink: privacy-preserving integration and sharing of datasets[J]. IEEE Transactions on Information Forensics and Security, 2019, PP(99): 564-577.

[12] 刘杨, 彭木根. 6G 内生安全: 体系结构与关键技术[J]. 电信科学, 2020, 36(1): 11-20.
LIU Y, PENG M G. 6G endogenous security: architecture and key technologies[J]. Telecommunications Science, 2020, 36(1): 11-20.

[13] 叶青青, 孟小峰, 朱敏杰, 等. 本地化差分隐私研究综述[J]. 软件学报, 2018, 29(7): 1981-2005.
YE Q Q, MENG X F, ZHU M J, et al. Survey on local differential privacy[J]. Journal of Software, 2018, 29(7): 1981-2005.

[14] 方俊斌, 蒋千越, 李爱平. 本地化差分隐私在数据众包中的应用[J]. 信息技术与网络安全, 2018, 37(6): 32-35, 51.
FANG J B, JIANG Q Y, LI A P. Local differential privacy applications in data crowdsourcing[J]. Information Technology and Network Security, 2018, 37(6): 32-35, 51.

[15] ERLINGSSON L, PIHUR V, KOROLOVA A. RAPPOR: randomized aggregatable privacy-preserving ordinal response[C]//Proceedings of the ACM Conference on Computer and Communications Security. New York: ACM Press, 2014: 1054-1067.

[16] BRODER A, MITZENMACHER M. Network applications of bloom filters: a survey[J]. Internet Mathematics, 2004, 1(4): 485-509.

[17] 吕云翔. 云计算与大数据技术[M]. 北京: 清华大学出版社, 2018.
LYU Y X. Cloud computing and big data technology[M]. Beijing: Tsinghua University Press, 2018.

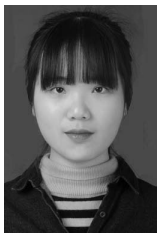
[18] ROBERT T. Regression shrinkage and selection via the lasso: a retrospective[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2011, 73.

[19] BONEH D, DRIJVERS M, NEVEN G. Compact multi-signatures for smaller blockchains[R]. Palo Alto, CA: Stanford University, 2018.

[20] WANG Q, WANG C, REN K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(5): 847-859.

[作者简介]

刘杨（1984− ），男，黑龙江哈尔滨人，北京邮电大学副教授、硕士生导师，主要研究方向为 6G 内生安全等。



李珺（1999− ），女，湖南株洲人，北京邮电大学硕士生，主要研究方向为 6G 内生安全等。

陈文韵（1996− ），女，上海人，北京邮电大学博士生，主要研究方向为 6G 内生安全等。

彭木根（1978− ），男，江西吉安人，博士，北京邮电大学教授、博士生导师，主要研究方向为移动通信组网等。