面向端到端加密即时通信平台的数据流转可信溯源机制

谢绒娜[1]，王嘉烜[2]，王文鼎[3]，史国振[1,2]，周创[1]，张玲翠[4]
（1.北京电子科技学院密码科学与技术系，北京 100070；2.北京电子科技学院电子与通信工程系，北京 100070；
3.北京电子科技学院网络空间安全系，北京 100070；4.中国科学院信息工程研究所，北京 100085）

摘　要：针对端到端加密在提升即时通信平台隐私性的同时限制了滥用内容审核，致使恶意信息难以准确溯源的问题，提出了一种数据流转可信溯源机制。该机制提出了基于溯源密钥的流转记录可信生成方法，通过溯源密钥绑定收发双方身份，结合基于哈希的消息认证码算法，基于溯源密钥和流转数据生成唯一数据标识符，进而构建防篡改的密态流转记录；提出了基于密态溯源密钥链的可信溯源方法，通过密钥加密机制生成与数据传播路径相映射的密态溯源密钥链，平台可基于最终用户提供的溯源密钥递归解密密钥链，重构流转路径，平衡隐私与溯源需求。安全分析表明，所提机制满足数据机密性、隐私性、不可否认性和溯源可审计性等。仿真结果表明，所提机制较现有方案降低27%计算开销，并减少38%的服务器溯源信息存储。

关键词：数据流转；溯源；溯源密钥；端到端加密
中图分类号：TP302
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025147

Trusted data flow traceability mechanism for end-to-end encrypted instant messaging platforms

XIE Rongna[1], WANG Jiaxuan[2], WANG Wending[3], SHI Guozhen[1,2], ZHOU Chuang[1], ZHANG Lingcui[4]
1. Department of Cryptography and Technology, Beijing Electronic Science and Technology Institute, Beijing 100070, China
2. Department of Electronic and Communication Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China
3. Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China
4. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China

Abstract: While end-to-end encryption (E2EE) improves privacy in instant messaging, it complicates auditing abusive content and tracing malicious information. To address this issue, a trusted data flow traceability mechanism (TDFTM) was proposed. It included a method for generating trusted flow records using traceability keys, which bind sender and receiver identities. A unique data identifier was generated via a Hash-based message authentication code (HMAC) based on the traceability key and the transmitted data, forming tamper-resistant encrypted flow records. Additionally, a trusted trace method using traceability keychains was proposed, in which a ciphertext traceability keychain was dynamically generated and mapped to the data path using a key encryption mechanism. Using the end-user's final traceability key, the platform recursively decrypted this keychain to reconstruct the propagation path, balancing privacy and traceability. Security analysis demonstrates that TDFTM ensures data confidentiality, privacy, non-repudiation, and auditable traceability. Simulations demonstrate a 27% reduction in computational overhead and a 38% reduction in server-side traceability storage compared with existing schemes.

Keywords: data flow, traceability, tracing key, end-to-end encryption

收稿日期：2025–03–10；**修回日期：**2025–08–14
通信作者：史国振，sgz1974@163.com
基金项目：国家重点研发计划基金资助项目（No.2023YFB3106505, No.2017YFB0802705, No.2017YFB0801803）；国家自然科学基金资助项目（No.61932015）
Foundation Items: The National Key Research and Development Program of China (No.2023YFB3106505, No.2017YFB0802705, No.2017YFB0801803), The National Natural Science Foundation of China (No.61932015)

0　引言

受益于互联网技术的快速发展，社交网络迎来了空前的发展高峰，成为信息发布、获取和共享的重要平台[1]。微信、钉钉、Facebook、Twitter等即时通信平台以其便捷性和高效性广泛融入人们的生活与工作，2024年腾讯财报显示，微信月活跃账户数已达13.82亿。社交网络革新了传统沟通方式，重塑了信息传播架构，推动了数据快速、动态流转的网络生态，但与此同时，社交网络数据传播的异构性、高度动态性和快速传播特性[2]，显著增加了敏感信息泄露与恶意内容扩散的风险。

为应对上述安全挑战，数据流转溯源机制成为增强社交网络即时通信平台责任追踪能力的重要技术手段。通过记录数据在各环节的流转信息，溯源机制构建了完整的日志体系，为恶意信息扩散或数据泄露事件的安全审计与取证分析提供支持。现有日志溯源系统主要依赖日志文件记录用户行为与数据传播路径，并通过事件分析[3]、依赖图构建[4-5]和规则匹配[6]等技术重建数据流转过程以定位泄漏源。然而，随着数据流转复杂性提升，传统日志方案存在以下局限：一方面，记录的完整性与安全性保障较薄弱，日志篡改或丢失将引发溯源的可信性问题；另一方面，明态化的流转与存储模式扩大了敏感隐私数据泄露的潜在风险。

随着用户和服务提供商对通信安全性和机密性需求的提升，平台逐步引入加密通信机制以保障数据传输安全。当前，主流即时通信平台普遍采用传输层安全（TLS, transport layer security）协议，通过客户端与服务器间建立加密通信通道，防止传输数据遭窃听、篡改并限制非授权访问。然而，TLS协议的加密范围仅覆盖传输路径，数据在平台服务器端的解密存储致使服务器仍是潜在的安全风险节点。一旦服务器因安全漏洞或管理失误遭受攻击将严重威胁用户数据安全，近年来频发的Facebook数据泄露和iCloud入侵事件凸显了这一问题的严峻性。

为了应对服务器端隐私泄露的风险，端到端加密（E2EE, end-to-end encryption）逐渐成为提升即时通信平台隐私保护能力的重要手段[7]。E2EE通过在用户设备端完成加密，仅允许接收方解密，使得服务器无法解密访问流转数据，从而有效规避了泄露风险。如今E2EE已被广泛应用于WhatsApp和Signal等平台以确保通信机密性与隐私性。然而，E2EE的隐私保护特性也引发了新的问题：平台无法访问密态数据内容，恶意用户可能利用加密机制在平台传播虚假信息、诈骗信息等滥用内容，显著增加了监管难度；同时，数据流转路径的隐匿性削弱了传播透明性，致使平台在审计和恶意行为追踪方面面临技术障碍，在某些场景下对执法活动与国家安全构成了潜在威胁[8]。加强对密态数据的流转监管，并对恶意滥用内容实现可信的溯源取证愈发重要。

针对上述问题，近年来研究者对密态数据流转溯源机制提出了一系列基于密码学的解决方案。文献[9]通过引入关联数据认证加密（AEAD, authenticated encryption with associated data）算法生成加密指纹以验证滥用投诉真实性，文献[10]采用数字签名与传播路径加密机制构建传播链追责恶意行为，文献[11]利用匿名密钥协商协议和链式消息认证码（MAC, message authentication code）验证技术设计隐私保护网络路径验证协议，实现源节点认证。然而，现有方案在溯源范围、抗抵赖能力及计算开销等方面仍存在不足，难以满足高频流转场景下对溯源效率和隐私保护的综合需求。

本文提出了一种针对端到端加密通信系统的数据流转可信溯源机制（TDFTM, trusted data flow traceability mechanism）。该机制从数据流转出发，引入溯源密钥的概念，应用基于哈希的消息认证码（HMAC, Hash-based message authentication code）和对称加密技术，基于溯源密钥动态生成每一跳流转的数据标识符与密态溯源标记，进而构建隐私保护加密链表以映射数据的流转路径。仅在用户举报滥用内容（如垃圾内容、诈骗信息）时，平台才能递归解密溯源标记实现数据流转的完整路径追踪，从而在保障数据安全与用户隐私的同时有效降低计算与存储开销。本文的主要贡献如下。

1) 提出了一种基于溯源密钥的流转记录可信生成方法。数据发送者基于收发双方身份标识和临时密钥动态生成溯源密钥，通过HMAC算法绑定溯源密钥与数据，生成唯一的数据标识符。平台基于溯源密钥和数据标识符构建防篡改的密态流转记录，实现了数据与收发双方的可信绑定，支持不可否认性和数据完整性验证。

2) 提出了一种基于密态溯源密钥链的可信溯源方法。数据流转过程中，用户利用对称加密算法使用当前节点溯源密钥加密上一节点溯源密钥，生成与数据传播紧密关联的密态溯源密钥链，作为不可篡改的隐私保护流转路径记录。密钥链仅允许最终用户与平台协作解密，通过递归计算存储键值重构数据流转路径，实现隐私性与溯源能力的有效平衡。

3) 安全分析结果表明，同现有文献相比，TDFTM 对数据机密性、完整性、用户隐私性、抗抵赖性、溯源可信性等 E2EE 即时通信平台的安全问题考虑更加全面。仿真结果表明，TDFTM 能有效降低计算与存储开销，并且具有良好的溯源性能优势。

1 相关工作

加密通信场景中，如何在保障数据隐私的前提下实现高效可信的数据流转溯源，已成为信息安全领域亟待解决的问题。该问题不仅要求系统在隐私保护与数据的机密性、完整性方面提供保障，防范记录篡改和节点抵赖，还需支持高频数据流转的实时分析与存储优化。针对上述需求，现有研究主要以 3 种技术路径展开：基于传统日志的方法、基于区块链的方法以及基于密码学的方法。

1) 基于传统日志的方法

Ghoshal 等[12]提出了基于事件时间序列匹配和因果依赖分析的溯源框架，初步解决了大规模日志分析的溯源问题，同时揭示了其存储、计算和分析上的性能瓶颈。此后，研究者围绕日志溯源性能和适用性展开多方面的改进。针对数据库日志溯源，Psallidas 等[3]设计了溯源提取系统 OneProvenance，通过日志分析来识别查询执行的依赖关系，并引入过滤优化以此动态提取数据库日志，实现粗粒度溯源。在此基础上，Gao 等[6]提出了异构日志联合分析方法，通过整合多源日志数据和规则匹配算法提升了近实时处理能力。针对大数据质监，Yang 等[13]提出了基于词嵌入的元组级数据溯源方法，设计了元组向量化编码机制，通过高效计算元组相似度来识别溯源关系，并应用有向无环图直观展示数据流转路径，实现高精确率的来源追踪。针对云环境入侵检测，Lou 等[14]提出基于多源日志关联规则的挖掘检测方案，提升了日志协同分析能力。然而，这些方法对具体场景依赖性强，规则库设计与维护成本较高。为了优化日志存储和检索，Wang 等[15]通过重新定义日志结构支持分布式大数据分析，并引入信任度评估机制检测潜在篡改行为。Siddiqui 等[16]使用布隆过滤器设计日志索引方法，大幅降低了存储检索成本，但假阳性问题影响了其在安全关键场景中的适用性。

尽管上述研究在日志存储优化和多源联合分析方面取得了一定进展，但传统日志溯源在加密通信仍面临挑战：加密通信的隐私保护要求限制了日志记录全面性，降低了溯源信息可获取性；同时，现有方法普遍缺乏完善的机密性保障与防篡改机制，难以满足可信溯源需求。

2) 基于区块链的方法

区块链以其分布式存储、不可篡改性和透明性，逐渐成为数据溯源研究热点。Liang 等[17]提出了云环境下无信任中心的溯源框架 ProvChain，通过钩子程序集中收集溯源信息，并结合默克尔树结构验证数据完整性。然而，该框架依赖于钩子程序的正常运行，存在溯源信息可信性风险。Zeng 等[18]针对无线传感器网络设计了基于区块链的溯源方案，将传感器数据以哈希表形式存储上链，保障数据完整性与可追溯性，但缺乏节点诚实性验证。为了提供区块链敏感数据隐私保护机制，Ramachandran 等[19]开发了溯源数据管理系统，结合零知识证明和公钥匿名化技术保护用户隐私。但该系统仅支持记录文档修改，无法追踪异常使用行为。Ding 等[20]提出的隐私增强溯源框架通过加密与授权共享实现公有链上的细粒度权限控制。

然而，区块链在即时通信中的应用受限于存储与查询性能瓶颈，以及吞吐量和时延无法满足高频数据流转与实时分析需求。

3) 基于密码学的方法

密码学以其在数据安全性与隐私性保障上的优势，为可信溯源提供了新的思路。早期研究主要集中于构建抗篡改的溯源链结构。Hasan 等[21]首次提出溯源链概念，利用链式加密和签名校验构建基于时间排序的溯源记录。然而，该方案难以防范所有权伪造和记录选择性删除问题。Wang 等[22]提出了公钥链溯源结构，通过公钥绑定记录

与用户身份，以签名机制增强记录防伪能力，但嵌套签名设计增加了计算复杂度。为了降低溯源链计算与存储开销，Ahmed 等[23]引入聚合签名优化签名体系，显著降低了存储成本与验证复杂度。Rangwala 等[24]提出了三重签名机制，通过前后记录的交叉验证提高了验证效率。然而，上述方案在隐私保护方面存在不足，难以满足加密通信场景的高隐私性需求。

为了平衡隐私与溯源能力，部分研究引入伪匿名和零知识证明技术[25]，或基于策略化权限的访问控制机制[26]，保护用户身份、操作等敏感信息。然而，上述方案适用场景相对有限。

针对 E2EE 平台的内容审计问题，近年来研究者提出了多种溯源方案。消息公证（MF, message franking）技术[9,27]采用 AEAD 算法生成不可篡改的加密指纹，允许平台验证举报真实性而无须访问原始内容。Issa 等[28]基于 MF 技术提出数据流转令牌预处理模型，提高了追责效率。然而，上述方案仅支持单跳通信验证，难以应对复杂的消息转发场景。为解决多跳溯源问题，Tyagi 等[29]提出了消息追溯机制，通过密码学标记记录消息流转路径，使平台在用户举报后重建完整消息传播链，但该方案缺乏抗抵赖能力。针对此缺陷，Kenney 等[10]设计了基于公钥签名的条件隐私保护溯源方案，实现流转操作可验证性，但该方案存储成本与计算开销较高。Peale 等[30]进一步优化，提出仅追踪原始发送者的源头追踪方案，降低了存储需求和计算复杂度，但仍面临如何嵌入可验证来源信息的挑战。在举报机制研究方面，Lian 等[31]提出的内容审核系统实现了基于阈值的私密举报，仅在举报数量超过阈值时追踪源头，但该系统对封闭恶意群组效果有限。在 E2EE 平台的监管治理方面，Jiang 等[32]提出的访问控制框架，将访问控制加密与消息认证技术结合，实现针对密态消息的管控审核，但该框架依赖于平台中心化审核，存在权力过度集中的治理风险。Namavari 等[33]提出了分层治理系统，通过扩展消息层安全协议，将治理逻辑从平台转移到客户端，使社区能自主实现内容审核和用户投票等功能，但高频治理场景下的共识开销仍待解决。

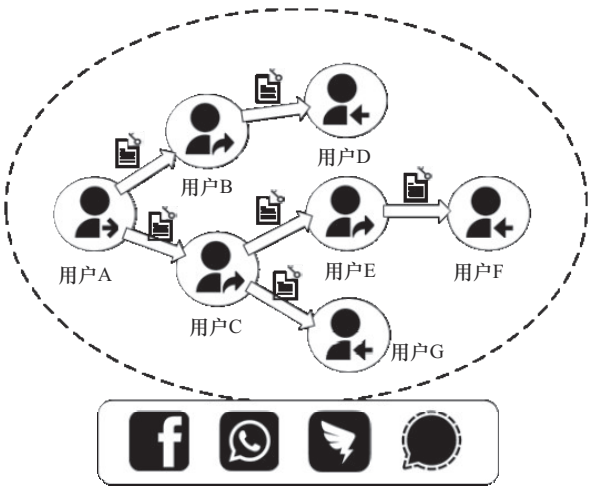
综上，现有密码学溯源方案在可信性和隐私保护方面具有一定优势，但普遍依赖复杂链式结构或生成额外签名与加密数据，致使计算与存储开销较高。针对 E2EE 即时通信平台的高隐私性需求，亟须设计轻量化且高效的可信溯源机制，实现隐私保护与溯源性能的平衡。

2　问题描述

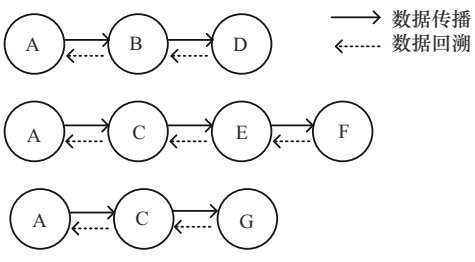
本节将对 E2EE 即时通信平台的数据流转应用场景进行分析，根据密态数据流转溯源的安全需求，提出 TDFTM 框架。

2.1　通信平台数据流转场景分析

在 E2EE 即时通信平台中，消息数据通过端到端加密通道在不同用户间流转，形成数据传播网络。传播过程可用如图 1 所示的传播模型描述。以图 1(a)为例，其中节点代表通信平台中的用户，边代表数据在用户间的传播路径，针对某一条数据的传播网络可以转化为图 1(b)所示的 3 条数据传播链。



(a) 数据传播网络



(b) 数据传播链

图 1　E2EE 即时通信平台数据流转模型

传播链上的用户被分为 3 种角色：起始节点表示数据创作者，生成并最初发布消息；中间节点表

示数据转发者，接收数据并将其转发给下一用户；终点节点表示数据静默的接收者，接收数据且不再执行传播操作。链状结构的长度表示数据在用户间转发的次数。从数据传播链中的节点出发向起始节点溯源，可以形成一条唯一的溯源链。溯源链的构建对于追踪数据来源、确定数据创作者以及分析取证具有重要意义。

2.2 数据流转可信溯源机制框架

2.2.1 数据流转溯源安全需求

E2EE使消息数据明文仅对发送方和接收方可见，平台无法直接访问和审查具体内容或流转路径。该机制有效保障了数据内容本身的机密性，但在数据的流转与溯源过程中存在以下安全问题。

1) 恶意用户试图通过伪造身份或实施中间人攻击，对流转数据进行篡改、伪造，破坏数据完整性和真实性。

2) 流转溯源机制未绑定用户的身份和操作行为，用户可能对自身发送、接收行为抵赖，致使用户行为责任界定模糊，无法保证溯源结果可信度。

3) 平台或第三方用户可能试图获取并分析数据流转中的元数据（如用户身份、节点关系），进而揭示链路关系，引发隐私风险。

4) 恶意用户试图通过篡改或替换流转记录破坏溯源链条的完整性，伪造错误的溯源链条以此混淆恶意内容或虚假信息的真实来源。

针对上述问题，数据流转溯源机制需满足以下安全需求。

1) 数据机密性与完整性：数据在流转过程中应始终以密文形式存在并受到完整性保护，所有用户在接收、转发数据前必须验证数据的完整性，防止恶意篡改或伪造行为。

2) 不可否认性：流转溯源机制应确保用户身份与其执行的操作行为绑定，使用户不能否认任何操作的真实性，包括发送、接收、修改。

3) 隐私性：数据流转过程中，当前节点用户应仅能了解数据的实时收发情况，即前后节点用户信息，而无法推断流转中其他节点的信息，平台和第三方用户无法获取数据流转链路任何相关信息。溯源过程中，仅受信任机构可在特定条件下基于溯源链追踪用户身份，其他人均无法推断相关信息。

4) 溯源可审计性：溯源机制应真实准确记录数据流转过程，确保溯源信息的不可伪造性，且能够防御恶意用户试图篡改溯源链或嫁祸诚实用户的行为，保障溯源结果可信性。

2.2.2 TDFTM框架

基于应用场景和安全需求分析，本文提出了TDFTM，实现对E2EE即时通信平台中数据流转路径的可信追踪。

如何可信生成数据流转记录是可信溯源的前提和基础。TDFTM引入溯源密钥的概念，通过随机的溯源密钥绑定发送者和接收者之间的一次数据流转，并将溯源密钥与数据收发双方的身份紧密关联，有效杜绝了用户事后抵赖。生成溯源密钥时，通过引入临时密钥替代用户永久身份密钥，杜绝了用户隐私泄露。基于溯源密钥和流转数据生成代表本次转发数据的唯一数据标识，并将其作为平台服务器存储溯源信息的键值，映射由溯源密钥与用户标识生成的密态溯源元数据，以此记录数据在用户间的流转轨迹。上述方法实现了数据、发送者、接收者的三者绑定，支持收发双方不可否认性及数据完整性验证。每条流转记录由唯一的数据标识和密态溯源密钥组成，有效防止平台对溯源元数据的篡改或泄露。

如何基于可信溯源信息对数据流转过程进行可信溯源，是需要解决的另一个关键问题。TDFTM引入溯源密钥链的概念，用户在每次转发数据前，使用本次生成的溯源密钥对接收的溯源密钥进行对称加密，随着数据在不同用户间的不断转发，密态溯源密钥之间将构成溯源密钥链。当数据接收用户发现数据存在不实或滥用内容时，可将接收的明文数据与对应的溯源密钥报告给平台申请溯源。平台接收溯源材料后，从报告节点出发，基于用户提交的溯源密钥递归解密溯源密钥链，向起始节点（数据创作者）方向回溯，最终生成一条完整的数据传播路径溯源证据链，实现流转过程可信溯源。

TDFTM框架如图2所示，涉及实体包括通信平台和平台用户（数据发送者U_s、数据接收者U_r）。本文机制基于E2EE消息传递系统，将底层E2EE系统视为黑盒，消息m本身通过用户间的端到端加密通道传输。系统参数及其含义如表1所示。

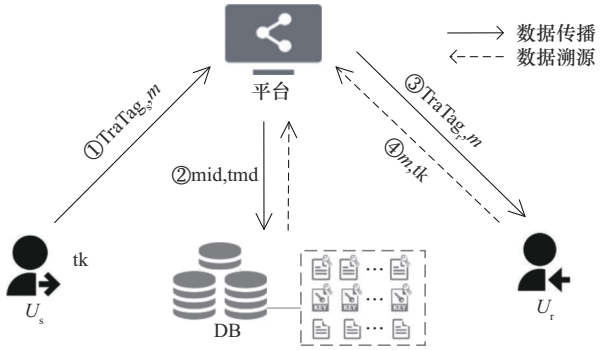


图2　TDFTM 框架

表1　　**系统参数及其含义**

参数	含义
U_i	用户 i
U_s、U_r	数据发送者、数据接收者
ID_i	用户 i 的身份标识
ID_s、ID_r	数据发送者身份标识、数据接收者身份标识
mid	数据标识符
tk	溯源密钥
C_{tk}	密态溯源密钥
tsk、tPk	双方通信使用的临时私钥、临时公钥
$TraTag_s$、$TraTag_r$	发送者溯源标识、接收者溯源标识
tmd	平台存储的溯源元数据
Enc (·)、Dec (·)	对称加密运算、对称解密运算
H (·)	安全的单向哈希函数
HMAC (·)	基于哈希的消息认证码生成函数
PRF (·)	伪随机数生成函数
EC	溯源证据链

本文机制构造的是一个五元组算法：TDFTM = (NewMsg, FwdMsg, Plat-Process, RecMsg, ECGen)，其中，NewMsg 表示数据生成算法；FwdMsg 表示数据转发算法，RecMsg 表示数据接收算法，它们在用户端进行数据的发送和接收时调用；Plat-Process 表示平台处理算法；ECGen 表示溯源证据链生成算法在平台服务器端进行流转记录处理存储和溯源追踪时调用。

TDFTM 框架中各实体之间的交互流程与数据传递方式按照如下步骤进行。

步骤 1　新数据生成或者数据转发。

当数据创作者 U_{auth} 创建新数据时，输入数据发送者身份标识 ID_s、数据接收者身份标识 ID_r 和数据明文 m，调用 NewMsg 生成发送者溯源标识 $TraTag_{s1}$，$TraTag_{s1}$= NewMsg(ID_s, ID_r, m)，并将 $TraTag_{s1}$ 和数据明文 m 发送给平台。$TraTag_{s1}$ 包括能够反映本次新数据流转操作的溯源密钥 tk、基于 tk 生成的数据标识符 mid 及代表数据流转路径的密态溯源密钥 C_{tk}。

当转发数据时，数据转发者 U_{fwd} 输入发送者身份标识 ID_s、接收者身份标识 ID_r、明文 m 和代表数据前一跳流转操作的溯源密钥 tk_{i-1} 调用 FwdMsg 生成发送者溯源标识 $TraTag_{si}$，$TraTag_{si}$=FwdMsg (ID_s,ID_r,m,tk_{i-1})，并将 $TraTag_{si}$ 和数据明文 m 发送给平台。FwdMsg 逻辑与 NewMsg 基本一致，$TraTag_{si}$ 同样包括能够反映本次新数据流转操作的溯源密钥 tk、基于 tk 生成的数据标识符 mid 及代表数据流转路径的密态溯源密钥 C_{tk}。

步骤 2　平台处理。

平台接收发送者发送的发送者溯源标识 $TraTag_{si}$、明文消息 m 以及发送者和接收者的身份标识，调用 Plat-Process 对标识进行验证，验证通过后生成本次流转操作的溯源元数据 tmd_i、接收者溯源标识 $TraTag_{ri}$，将〈mid_i,tmd_i〉以键值对形式更新平台数据库（DB），并将接收者溯源标识 $TraTag_{ri}$ 和明文消息 m 发送给接收者。((mid_i,tmd_i), $TraTag_{ri}$)=Plat-Process(ID_s,ID_r,$TraTag_{si}$,DB)。

步骤 3　接收者接收数据。

数据接收者接收发送者身份标识、接收者身份标识、接收者溯源标识 $TraTag_{ri}$ 及解密后的消息明文 m，调用 RecMsg，校验接收数据的完整性，生成新的溯源密钥 tk，用于数据的进一步转发或报告溯源。tk_i=RecMsg(ID_s,ID_r,$TraTag_{ri}$,m)。

步骤 4　数据溯源。

当数据接收者发现数据有误或存在滥用时，向平台报告最新的溯源密钥 tk 和明文 m 申请溯源。平台输入报告者身份标识、密钥和明文数据，调用 ECGen，EC =ECGen(ID_r,tk_i, m, DB)，通过递归计算并查询存储键值，依次解密溯源元数据，输出证据链 (EC, evidence chain)，得到数据源头与流转路径。

2.2.3　安全模型

针对安全需求，本文提出了形式化安全模型。具体定义如下。

定义 1　数据机密性。在 TDFTM 中，未经授

权的实体无法获取数据内容或流转路径等隐私信息。数据机密性具体分为平台存储机密性和用户追踪机密性

平台存储机密性。平台在用户未举报的前提下，不应能区分真实生成的溯源标识与伪造的随机值。敌手$\mathcal{A}$的优势$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{p-conf}}(\mathcal{A})$定义为敌手在如下游戏中猜测概率差的绝对值

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{p-conf}}(\mathcal{A})=\left|\Pr\left[\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},1}\Rightarrow 1\right]-\Pr\left[\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},0}\Rightarrow 1\right]\right|$$

其中，游戏$\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},0}$中的挑战比特$b$决定了敌手收到的挑战值类型。游戏具体步骤如下。

1) 初始化。挑战者$\mathcal{C}$初始化系统参数、生成密钥，设置挑战$b\in\{0,1\}$。

2) 查询阶段。$\mathcal{A}$向$\mathcal{C}$进行多项式时间内询问，模拟数据流转行为，包括以下查询。

①NewMsg查询。$\mathcal{A}$选择$\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},m\right)$，调用NewMsg生成溯源密钥tk和发送者溯源标识$\mathrm{TraTag}_{\mathrm{s}}$并返回。

②FwdMsg查询。$\mathcal{A}$选择$\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},m,\mathrm{tk}_{\mathrm{prev}}\right)$，其中$\mathrm{tk}_{\mathrm{prev}}$代表前一跳溯源密钥，调用FwdMsg生成新溯源标识$\mathrm{TraTag}_{\mathrm{s}}$并返回。

3) 挑战阶段。敌手$\mathcal{A}$提交挑战请求Chal$\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},m,\mathrm{tk}_{\mathrm{prev}}\right)$，挑战者$\mathcal{C}$根据挑战位$b$执行以下操作。

若b=1，$\mathcal{C}$调用$\mathrm{NewMsg}\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},m\right)$或$\mathrm{FwdMsg}\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},m,\mathrm{tk}_{\mathrm{prev}}\right)$生成真实溯源标识$\mathrm{TraTag}_{\mathrm{s}}^{1}=\left(\mathrm{mid},C_{\mathrm{tk}},\mathrm{tPk}\right)$并返回给$\mathcal{A}$。

若b=0，$\mathcal{C}$生成与$\mathrm{TraTag}_{\mathrm{s}}$长度相同的随机串$\mathrm{TraTag}_{\mathrm{s}}^{0}=\{0,1\}^{\mathrm{len}\left(\mathrm{TraTag}_{\mathrm{s}}\right)}$并返回给$\mathcal{A}$。

4) 判断阶段。$\mathcal{A}$输出b'，若$b'=b$，则$\mathcal{A}$获胜。

用户追踪机密性。除收发双方外，任何用户不应能推断其所接收数据的历史流转路径。即接收者无法判断收到的数据是由创作者新创建，还是由上游用户转发。敌手$\mathcal{A}$的优势$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{u-conf}}(\mathcal{A})$定义为敌手在如下游戏中猜测概率差的绝对值

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{u-conf}}(\mathcal{A})=\left|\Pr\left[\mathrm{UTrConf}_{\mathrm{TDFTM}}^{\mathcal{A},1}\Rightarrow 1\right]-\Pr\left[\mathrm{UTrConf}_{\mathrm{TDFTM}}^{\mathcal{A},0}\Rightarrow 1\right]\right|$$

其中，游戏$\mathrm{UTrConf}_{\mathrm{TDFTM}}^{\mathcal{A},b}$中的挑战比特$b$决定了敌手所接收的数据来源。游戏具体步骤如下。

1) 初始化。挑战者$\mathcal{C}$初始化系统参数，设置挑战位$b\in\{0,1\}$。

2) 查询阶段。$\mathcal{A}$向$\mathcal{C}$进行多项式时间内询问，包括以下查询

①NewMsg查询。$\mathcal{A}$选择$\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},m\right)$，调用NewMsg生成相应的tk和$\mathrm{TraTag}_{\mathrm{s}}$并返回。

②FwdMsg查询。$\mathcal{A}$选择$\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},m,\mathrm{tk}_{\mathrm{prev}}\right)$，调用FwdMsg生成相应的$\mathrm{TraTag}_{\mathrm{s}}$并返回。

③Plat-Process查询。$\mathcal{A}$选择$\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},\mathrm{TraTag}_{\mathrm{s}}\right)$，调用Plat-Process生成相应的$\mathrm{TraTag}_{\mathrm{r}}$，存储溯源元数据并返回。

④RecMsg查询。$\mathcal{A}$选择$\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},\mathrm{TraTag}_{\mathrm{r}},m\right)$，调用RecMsg生成tk并返回。

3) 挑战阶段。敌手$\mathcal{A}$提交挑战请求$\mathrm{Chal}\left(U_0,U_1,U_2,m,\mathrm{TraTag}_{\mathrm{r}},\mathrm{DB}\right)$，$\mathcal{C}$执行以下操作。

若b=1，$\mathcal{C}$调用$\mathrm{NewMsg}\left(U_1,U_2,m\right)$生成创作者新创建场景下的$\mathrm{tk}_1$。

若b=0，$\mathcal{C}$调用$\mathrm{RecMsg}\left(U_0,U_1,m,\mathrm{TraTag}_{\mathrm{r}}\right)$生成从上游转发场景下的$\mathrm{tk}_0$。

挑战者$\mathcal{C}$根据挑战位b选择tk_b调用$\mathrm{FwdMsg}\left(U_1,U_2,m,\mathrm{tk}_b\right)$生成$\left(\mathrm{TraTag}_{\mathrm{s}},\mathrm{tk}'\right)$，随后调用$\mathrm{Plat\text{-}Process}\left(\mathrm{DB},U_1,U_2,\mathrm{TraTag}_{\mathrm{s}}\right)$生成$\mathrm{TraTag}_{\mathrm{r}}'$，最后调用$\mathrm{RecMsg}\left(U_1,U_2,m,\mathrm{TraTag}_{\mathrm{r}}'\right)$并返回接收者视图给敌手$\mathcal{A}$。

4) 判断阶段。$\mathcal{A}$输出b'，若$b'=b$，则$\mathcal{A}$获胜。

定义2 不可否认性。TDFTM中用户不能否认其已执行的数据收发操作，敌手$\mathcal{A}$的优势$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{non-rep}}(\mathcal{A})$定义为敌手在如下游戏中成功实现抵赖的概率

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{non-rep}}(\mathcal{A})=\Pr\left[\mathrm{NonRep}_{\mathrm{TDFTM}}^{\mathcal{A}}\Rightarrow\mathrm{true}\right]$$

其中，游戏$\mathrm{NonRep}_{\mathrm{TDFTM}}^{\mathcal{A}}$模拟了敌手尝试否认其收发操作的过程。游戏具体步骤如下。

1) 初始化。挑战者$\mathcal{C}$初始化参数与DB，为n个诚实用户生成密钥对$\left(\mathrm{Pk}_i,\mathrm{sk}_i\right),\forall i\in[1,n]$。初始化状态表$\mathrm{WasSent}\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},\mathrm{mid},m\right)\leftarrow\mathrm{false}$和$\mathrm{WasRec}\left(\mathrm{ID}_{\mathrm{s}},\mathrm{ID}_{\mathrm{r}},\mathrm{mid},m\right)\leftarrow\mathrm{false}$，用于记录用户数据发送和接收操作的发生与否。

2) 查询阶段。敌手$\mathcal{A}$向挑战者$\mathcal{C}$进行多项式时间内询问，包括以下查询。

①Send 查询。敌手$\mathcal{A}$提交$(\mathrm{ID_s},\mathrm{ID_r},m,\mathrm{tk_{prev}})$模拟诚实用户的发送操作，若$\mathrm{ID_s}\notin[1,n]$则返回⊥。若$\mathrm{tk_{prev}}$不存在，调用 NewMsg 生成 tk 与$\mathrm{TraTag_s}$，否则调用 FwdMsg 生成相应标识。随后调用 Plat-Process 更新 DB 并生成$\mathrm{TraTag_r}$，同时设置$\mathrm{WasSent}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)\leftarrow\mathrm{true}$。若$\mathrm{ID_r}\in[1,n]$，进一步调用 RecMsg 进行数据完整性验证，验证成功则设置$\mathrm{WasRec}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)\leftarrow\mathrm{true}$并返回$\mathrm{TraTag_r}$。否则返回$(\mathrm{TraTag_r},\mathrm{tk})$。

②SendMal 查询。敌手$\mathcal{A}$提交$(\mathrm{ID_s},\mathrm{ID_r},m,\mathrm{tk},\mathrm{TraTag_s})$模拟恶意用户向平台注入伪造数据，若$\mathrm{ID_s}\in[1,n]$则返回⊥。计算数据标识符$\mathrm{mid}\leftarrow\mathrm{HMAC}(\mathrm{tk},m)$，调用 Plat-Process 更新存储并生成$\mathrm{TraTag_r}$。若$\mathrm{ID_r}\in[1,n]$且调用 RecMsg 数据完整性成功，设置$\mathrm{WasRec}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)\leftarrow\mathrm{true}$，返回$\mathrm{TraTag_r}$。

3) 挑战阶段。敌手$\mathcal{A}$输出四元组$(\mathrm{ID_U}^*,m^*,\mathrm{tk}^*,\mathrm{rep_type})$，其中$\mathrm{ID_U}^*$表示目标用户身份标识，$m^*$表示数据，$\mathrm{tk}^*$表示敌手伪造的溯源密钥，$\mathrm{rep_type}\in\{\text{"send","receive"}\}$表示攻击类型。

4) 验证阶段。挑战者$\mathcal{C}$根据 rep_type 验证敌手是否成功否认。

rep_type = "send"：计算数据标识符$\mathrm{mid}^*\leftarrow\mathrm{HMAC}(\mathrm{tk}^*,m^*)$，查询数据库对应条目$\mathrm{DB}[\mathrm{mid}^*]\rightarrow(\mathrm{ID_s}^*,\mathrm{ID_r}^*,C_{\mathrm{tk}}^*)$。若以下条件同时成立则攻击成功：$\mathrm{WasSent}(\mathrm{ID_U}^*,-,\mathrm{mid}^*,m^*)=\mathrm{false}$，$\mathrm{ID_s}^*=\mathrm{ID_U}^*$。

rep_type = "receive"：计算数据标识符$\mathrm{mid}^*\leftarrow\mathrm{HMAC}(\mathrm{tk}^*,m^*)$，查询数据库对应条目$\mathrm{DB}[\mathrm{mid}^*]\rightarrow(\mathrm{ID_s}^*,\mathrm{ID_r}^*,C_{tk}^*)$。若以下条件同时成立则攻击成功：$\mathrm{WasRec}(-,\mathrm{ID_U}^*,\mathrm{mid}^*,m^*)=\mathrm{false}$，$\mathrm{ID_r}^*=\mathrm{ID_U}^*$。

若上述任何一种情况成立，则$\mathcal{C}$返回 true，敌手$\mathcal{A}$获胜，否则失败。

定义 3　溯源可审计性。TDFTM 确保系统能够真实准确地记录数据流转过程，使得溯源信息具有不可伪造性，并能够防御恶意用户试图篡改溯源链或嫁祸诚实用户的行为。具体而言，恶意用户不应能够：替换数据内容；替换用户身份；伪造传播路径；篡改流转记录。敌手$\mathcal{A}$的优势$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{tr-audit}}(\mathcal{A})$定义为敌手在如下游戏中成功实现伪造的概率

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{tr-audit}}(\mathcal{A})=\Pr\left[\mathrm{TrAudit}_{\mathrm{TDFTM}}^{\mathcal{A}}\Rightarrow\mathrm{true}\right]$$

其中，游戏$\mathrm{TrAudit}_{\mathrm{TDFTM}}^{\mathcal{A}}$模拟了敌手尝试突破上述任一安全约束的过程。游戏具体步骤如下。

1) 初始化。$\mathcal{C}$初始化系统参数与 DB，为n个诚实用户生成密钥对$(\mathrm{Pk}_i,\mathrm{sk}_i),\forall i\in[1,n]$。初始化状态表$\mathrm{Audit}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)\leftarrow\mathrm{false}$，用于记录数据流转行为是否真实发生。

2) 查询阶段。敌手$\mathcal{A}$向挑战者$\mathcal{C}$进行多项式时间内询问，包括以下查询。

①Send 查询。$\mathcal{A}$提交$(\mathrm{ID_s},\mathrm{ID_r},m,\mathrm{tk_{prev}})$模拟诚实用户的流转操作，若$\mathrm{ID_s}\notin[1,n]$则返回⊥。根据$\mathrm{tk_{prev}}$是否存在，调用 NewMsg 或 FwdMsg 生成相应溯源标识$\mathrm{TraTag_s}$。调用 Plat-Process 更新数据库$\mathrm{DB}[\mathrm{mid}]\leftarrow\mathrm{tmd}=(\mathrm{ID_s},\mathrm{ID_r},C_{\mathrm{tk}})$并生成$\mathrm{TraTag_r}$。若$\mathrm{ID_r}\in[1,n]$，调用 RecMsg 验证数据完整性，成功则设置$\mathrm{Audit}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)\leftarrow\mathrm{true}$并返回$\mathrm{TraTag_r}$。否则返回$(\mathrm{TraTag_r},\mathrm{tk})$。

②SendMal 查询。$\mathcal{A}$提交$(\mathrm{ID_s},\mathrm{ID_r},m,\mathrm{tk},\mathrm{TraTag_s})$模拟恶意用户向平台注入伪造数据，若$\mathrm{ID_s}\in[1,n]$则返回⊥。计算数据标识符$\mathrm{mid}\leftarrow\mathrm{HMAC}(\mathrm{tk},m)$，调用 Plat-Process 更新存储并生成$\mathrm{TraTag_r}$。若$\mathrm{ID_r}\in[1,n]$，调用 RecMsg 验证数据完整性，成功则设置$\mathrm{Audit}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)\leftarrow\mathrm{true}$，返回$\mathrm{TraTag_r}$。

③ECGen 查询。执行 ECGen，生成并返回证据链 EC。

3) 挑战阶段。敌手$\mathcal{A}$输出四元组$(\mathrm{ID_U}^*,m^*,\mathrm{tk}^*,\mathrm{attack_type})$，其中$\mathrm{ID_U}^*$表示目标用户身份标识，$m^*$表示数据，$\mathrm{tk}^*$表示伪造的溯源密钥，$\mathrm{attack_type}\in\{\text{"msg_replace","id_replace","path_forge","record_ tamper"}\}$表示溯源伪造攻击类型。

4) 验证阶段。挑战者$\mathcal{C}$根据 attack_type 验证敌手$\mathcal{A}$是否成功伪造溯源路径。

$\mathrm{attack\_type}\in\{\text{"msg\_replace"}\}$：计算标识符$\mathrm{mid}^*\leftarrow\mathrm{HMAC}(\mathrm{tk}^*,m^*)$，查询数据库对应条目$\mathrm{DB}[\mathrm{mid}^*]\rightarrow(\mathrm{ID_s}^*,\mathrm{ID_r}^*,C_{tk}^*)$。若以下条件同时成立则攻击成功：$\mathrm{ID_s}^*=\mathrm{ID_U}^*$，$\exists m'\neq m^*$，$\mathrm{Audit}(\mathrm{ID_U}^*,-,\mathrm{mid}^*,m')=\mathrm{true}$。

attack_type ∈ {"id_replace"}：计算标识符 mid* ← HMAC(tk*,m*)，查询数据库对应条目 DB[mid*] → ($\mathrm{ID_s}$*,$\mathrm{ID_r}$*,C_{tk}*)。若以下条件同时成立则攻击成功：$\mathrm{ID_s}^* = \mathrm{ID_U}^*$，$\exists \mathrm{ID_r}' \neq \mathrm{ID_r}^*$，$\mathrm{Audit}(\mathrm{ID_U}^*,\mathrm{ID_r}',\mathrm{mid}^*,m^*) = \mathrm{true}$。

attack_type ∈ {"path_forge"}：生成证据链 EC ← ECGen($\mathrm{ID_U}$*,tk*,m*,DB)，检查 EC 路径中是否存在未在 Audit 表中记录的流转对。若存在则攻击成功。

attack_type ∈ {"record_tamper"}：计算标识符 mid* ← HMAC(tk*,m*)，查询数据库对应条目 DB[mid*] → ($\mathrm{ID_s}$*,$\mathrm{ID_r}$*,C_{tk}*)，查询 Audit 表中是否存在对应记录 $\mathrm{Audit}(\mathrm{ID_s}^*,\mathrm{ID_r}^*,\mathrm{mid}^*,m^*)$。解密验证 C_{tk}*，检查溯源链完整性。若解密失败或发现溯源链遭篡改，则攻击成功。

若上述任何一种攻击成立，则 $\mathcal{C}$ 返回 true，敌手 $\mathcal{A}$ 获胜，否则失败。

3 数据流转与溯源取证方案

本节将详细介绍 TDFTM，共包括以下 2 个部分：基于溯源密钥的流转记录可信生成方法和基于密态溯源密钥链的可信溯源方法。

3.1 基于溯源密钥的流转记录可信生成方法

3.1.1 系统初始化

系统初始化阶段包括系统参数设置和用户密钥对生成 2 个环节。

1) 系统参数设置：给定有限域 F_p 上的椭圆曲线 $E_p(a,b)$，选择基点 $G \in E_p(a,b)$，G 的阶记为 $n = \mathrm{ord}(G)$，n 是一个大的素数，基于 G 生成椭圆曲线加法群 $\langle G \rangle$。选取单向哈希函数 H: $\langle G \rangle \rightarrow Z_n^*$。公开系统参数 $\{p,a,b,G,n,\mathrm{H}\}$。

2) 用户密钥对生成：系统参数设置完成，系统为每个注册用户 U_i 生成唯一身份标识 ID_i 以及用户的公私钥对 $(\mathrm{Pk_U},\mathrm{sk_U})$ 表示用户永久身份，其中 $\mathrm{sk_U} \in [1,n-1]$，$\mathrm{Pk_U} = \mathrm{sk_U} \cdot G$。

3.1.2 数据发送

数据发送阶段包括新数据生成与发送、数据转发 2 种情况。

1)新数据生成与发送

数据创作者 U_{auth} 生成新数据 m 后，调用 NewMsg 生成发送者溯源标识 $\mathrm{TraTag_{s1}}$。

溯源密钥生成：U_{auth} 首先随机生成初始溯源密钥 $\mathrm{tk_0} \leftarrow \{0,1\}^*$，作为数据传播链的源头标识。

计算溯源密钥 $\mathrm{tk_1} = \mathrm{H}\left(\left(\mathrm{tsk} + \mathrm{sk_s}\right) \cdot \mathrm{Pk_r}\right)$，以此表示 U_{auth} 和第一个接收者 U_{r} 之间的数据流转。其中，tsk 为创作者调用伪随机数生成函数（PRF）生成的临时私钥，$\mathrm{sk_s}$ 表示创作者的私钥，$\mathrm{Pk_r}$ 表示接收者的公钥。发送者的私钥和接收者的公钥作为 $\mathrm{tk_1}$ 的输入参数，实现了溯源密钥与收发双方身份的绑定；引入临时生成的随机数 tsk 作为临时私钥，实现了对发送者身份的盲化，达到隐私保护的目的。

数据标识符生成：创作者基于溯源密钥 $\mathrm{tk_1}$ 和流转数据调用 HMAC 函数，生成数据标识符 $\mathrm{mid_1}$，$\mathrm{mid_1} = \mathrm{HMAC}(\mathrm{tk_1},m)$。HMAC 函数的抗碰撞性确保 $\mathrm{mid_1}$ 与 U_{s}、U_{r} 和 m 建立唯一对应的关系，同时基于 $\mathrm{tk_1}$ 可有效验证传输数据的完整性与真实性。

密态溯源密钥生成：使用 $\mathrm{tk_1}$ 加密 $\mathrm{tk_0}$ 生成密态初始溯源密钥 $C_{\mathrm{tk_0}} = \mathrm{Enc}(\mathrm{tk_1},\mathrm{tk_0})$，为数据传播提供安全的源头链接。

溯源标识符生成：基于数据标识符 $\mathrm{mid_1}$、密态初始溯源密钥 $C_{\mathrm{tk_0}}$、创作者的临时公钥 tPk = tsk · G 生成溯源标识 $\mathrm{TraTag_{s1}} = (\mathrm{mid_1},C_{\mathrm{tk_0}},\mathrm{tPk})$。

明文 m 生成端到端加密密文，该密文与发送者溯源标识 $\mathrm{TraTag_{s1}}$ 一起通过平台发送给接收者。

2) 数据转发

数据转发者 U_{fwd} 调用 FwdMsg，生成与接收数据 m 相关的溯源信息，并执行转发操作。RecMsg 的细节介绍参见 3.1.4 节。为保证数据传输的隐私性，第三方平台不应能区分创作和转发的数据，因此 NewMsg 与 FwdMsg 在算法逻辑上保持一致。不同之处是 FwdMsg 额外引入了前一次数据流转中的溯源密钥 tk_{i-1}（由 RecMsg 验证输出）。

采用 U_{fwd} 新生成的溯源密钥 tk_i 加密 tk_{i-1}，形成密态链接 $C_{\mathrm{tk}_{i-1}} = \mathrm{Enc}(\mathrm{tk}_i,\mathrm{tk}_{i-1})$，作为指向上一条数据标识符的加密指针。基于数据标识符 mid_i、密态溯源密钥 $C_{\mathrm{tk}_{i-1}}$、转发者临时公钥 tPk 生成发送者溯源标识 $\mathrm{TraTag}_{si} = (\mathrm{mid}_i,C_{\mathrm{tk}_{i-1}},\mathrm{tPk})$，将 TraTag_{si} 以及端到端加密密文发送给平台。

随着数据的不断转发，用户依次采用后一个溯源密钥加密前一个溯源密钥，由此形成的溯源密钥链与数据的传播路径紧密关联，如图 3 所示。

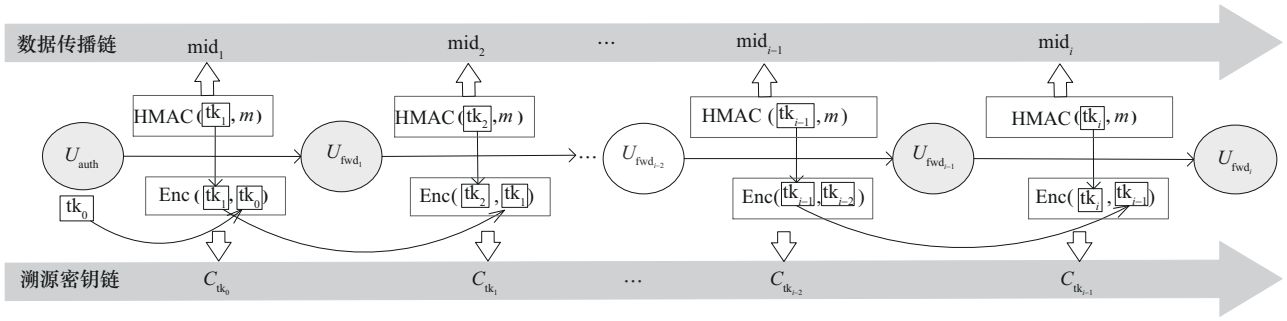


图3　溯源密钥链生成示意

3.1.3　溯源元数据生成

平台接收到发送者溯源标识后，调用Plat-Process生成溯源元数据，并存储在DB中，为溯源追踪提供数据基础。

平台接收溯源标识TraTag_{si}后，首先检查DB中是否存在相同的mid_i，以防止恶意用户的重放攻击。检验通过，生成溯源元数据$\mathrm{tmd}_i=(\mathrm{ID_s},\mathrm{ID_r},C_{\mathrm{tk}_{i-1}})$，将数据标识符$\mathrm{mid}_i$和溯源元数据$\mathrm{tmd}_i$映射成键值对〈$\mathrm{mid}_i$,$\mathrm{tmd}_i$〉的形式存储在DB中。更新完成后平台创建接收者溯源标识$\mathrm{TraTag}_{ri}=(\mathrm{mid}_i,\mathrm{tPk})$，并将$\mathrm{TraTag}_{ri}$及端到端加密密文传递给接收者。

3.1.4　数据接收

根据传递的溯源标识TraTag_{ri}，接收者调用RecMsg验证流转数据。首先基于临时公钥tPk、发送者公钥$\mathrm{Pk_s}$和接收者私钥$\mathrm{sk_r}$计算新的溯源密钥$\mathrm{tk}_i'=\mathrm{H}((\mathrm{tPk}+\mathrm{Pk_s})\cdot\mathrm{sk_r})$，进而生成数据标识符$\mathrm{mid}_i'=\mathrm{HMAC}(\mathrm{tk}_i',m)$。比对$\mathrm{mid}_i'$和接收到的$\mathrm{mid}_i$是否相等，其中

$$\begin{aligned}&\mathrm{mid}_i'=\mathrm{HMAC}(\mathrm{tk}_i',m)=\\&\mathrm{HMAC}(\mathrm{H}((\mathrm{tPk}+\mathrm{Pk_s})\cdot\mathrm{sk_r}),m)=\\&\mathrm{HMAC}(\mathrm{H}((\mathrm{tsk}\cdot G+\mathrm{sk_s}\cdot G)\cdot\mathrm{sk_r}),m)=\\&\mathrm{HMAC}(\mathrm{H}((\mathrm{tsk}+\mathrm{sk_s})\cdot\mathrm{sk_r}\cdot G),m)=\\&\mathrm{HMAC}(\mathrm{H}((\mathrm{tsk}+\mathrm{sk_s})\cdot\mathrm{Pk_r}),m)=\\&\mathrm{HMAC}(\mathrm{tk}_i,m)=\mathrm{mid}_i\end{aligned}$$

两者相等从而验证接收数据完整性，并确保收发双方身份匹配，方可接收数据，输出溯源密钥tk_i用于后续转发或报告溯源。否则说明数据遭篡改破坏或不是指定的发送者所传输的数据，此时接收者须向平台报告数据有误情况，停止转发操作。通过上述实现数据完整性及收发双方的身份验证。一旦接收者验证通过并接收了数据，后续将不可抵赖。

3.2　基于密态溯源密钥链的可信溯源方法

在数据流转过程中，当系统用户识别到虚假信息或恶意内容时，可向平台报告申请溯源。由于解密溯源元数据的能力仅在平台和U_{r}之间秘密共享，平台需在U_{r}提交溯源材料后，调用ECGen对数据库中的溯源元数据进行回溯。通过解密并追踪各数据标识符间的密态溯源密钥，平台生成证据链揭示数据的源头及传播路径。证据链EC由用户标识和用户间数据传输的关联标识符mid交替构成，结构为

$$\mathrm{EC}=(U_1,\mathrm{mid}_1,U_2,\mathrm{mid}_2,\cdots,\mathrm{mid}_{\tau-1},U_\tau)$$

其中，τ表示追踪路径长度。报告用户U_{rpt}向平台发送明文m和溯源密钥tk_τ，平台核实报告后调用ECGen执行追踪取证操作，EC生成流程如图4所示。

平台初始化EC，并从U_{rpt}开始追踪。基于U_{rpt}提供的m和tk_τ，生成数据标识符$\mathrm{mid}_\tau=\mathrm{HMAC}\left(\mathrm{tk}_\tau,m\right)$。随后，平台在数据库中检索$\mathrm{mid}_\tau$，查找对应的溯源密钥指针$C_{\mathrm{tk}_{\tau-1}}$，并将$\mathrm{mid}_\tau$和识别出的发送者身份$\mathrm{ID_s}$添加到EC中。以$\mathrm{tk}_\tau$作为解密密钥，解密$C_{\mathrm{tk}_{\tau-1}}$以得到前一个溯源密钥$\mathrm{tk}_{\tau-1}=\mathrm{Dec}\left(\mathrm{tk}_\tau,C_{\mathrm{tk}_{\tau-1}}\right)$，基于$\mathrm{tk}_{\tau-1}$生成关联着前一个节点的数据标识符$\mathrm{mid}_{\tau-1}=\mathrm{HMAC}\left(\mathrm{tk}_{\tau-1},m\right)$。重复上述步骤，直到$\mathrm{mid}=\mathrm{HMAC}\left(\mathrm{tk}_0,m\right)$在数据库中检索失败，此时表明已回溯至$m$的原始创作者$U_{\mathrm{auth}}$，传播路径追踪结束。本文提出的基于密态溯源密钥链的可信溯源方法的伪代码如算法1所示。

算法1　ECGen( )

输入　接收者标识$\mathrm{ID_r}$，溯源密钥tk，明文m，数据库DB

输出　溯源证据链EC

1) init list EC
2) $i \leftarrow 0$
3) $\mathrm{EC}[i] \leftarrow \mathrm{ID_r}$
4) $\mathrm{mid} = \mathrm{HMAC}(\mathrm{tk},m)$
5) while mid ∈ DB:
6) $(C_{\mathrm{tk}},\mathrm{ID_s},\mathrm{ID_r}) \leftarrow \mathrm{DB}[\mathrm{mid}]$
7) if $\mathrm{ID_r} \neq \mathrm{EC}[i]$
8) break
9) $\mathrm{EC}[i+1] \leftarrow \mathrm{mid}$
10) $\mathrm{EC}[i+2] \leftarrow \mathrm{ID_s}$
11) $\mathrm{mid} \leftarrow \mathrm{HMAC}(\mathrm{tk},m)$
12) $\mathrm{tk} \leftarrow \mathrm{Dec}(\mathrm{tk},C_{\mathrm{tk}})$
13) $i \leftarrow i+2$
14) return EC^{-1}

4 安全性分析

定理 1 数据机密性。假设哈希函数 H 可被建模为随机预言机，HMAC 函数是安全的伪随机函数（PRF），对称加密方案 $E=(\mathrm{Enc},\mathrm{Dec})$ 满足选择明文攻击下的不可区分性（IND-CPA），则 TDFTM 满足数据机密性，具体包括平台存储机密性和用户追踪机密性，即

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\mathrm{conf}}(\mathcal{A})=\mathrm{Adv}_{\mathrm{TDFTM}}^{\mathrm{p\text{-}conf}}(\mathcal{A})+\mathrm{Adv}_{\mathrm{TDFTM}}^{\mathrm{u\text{-}conf}}(\mathcal{A})$$

引理 1 平台存储机密性。在定理 1 的假设下，TDFTM 对于任意概率多项式时间（PPT, probabilistic polynomial time）的平台视图敌手 $\mathcal{A}$，存在 PPT 敌手 $\mathcal{B}$ 和 $\mathcal{C}$ 使

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\mathrm{p\text{-}conf}}(\mathcal{A})\leqslant\mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{prf}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{E},q}^{\mathrm{ind\text{-}cpa}}(\mathcal{C})$$

其中，q 表示发送查询次数。

证明 以下通过游戏跳转进行证明。定义初始游戏 G_0 为平台机密性安全游戏 $\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},1}$，即挑战者返回真实的溯源标识 $\mathrm{TraTag_s}=\left(\mathrm{mid}_i,C_{\mathrm{tk}_{i-1}},\mathrm{tPk}\right)$。

游戏跳转 1 ($G_0 \rightarrow G_1$)：将 $\mathrm{mid}_i=\mathrm{HMAC}(\mathrm{tk},m)$ 替换为随机函数输出 $F(\mathrm{tk},m)$。基于 PRF 安全性。存在 PPT 敌手 $\mathcal{B}$ 使

$$\left|\Pr\left[G_0 \Rightarrow 1\right]-\Pr\left[G_1 \Rightarrow 1\right]\right|\leqslant\mathrm{Adv}_{\mathrm{HMAC,q}}^{\mathrm{prf}}(\mathcal{B})$$

游戏跳转 2($G_1 \rightarrow G_2$)：将 $C_{\mathrm{tk}_{i-1}}=\mathrm{Enc}(\mathrm{tk}_i,\mathrm{tk}_{i-1})$ 替换为随机字符串 $C_{\mathrm{tk}_{i-1}} \leftarrow \{0,1\}^{\mathrm{len(tk)}}$。基于对称加密方案的 IND-CPA 安全性，存在 PPT 敌手 $\mathcal{C}$ 使

$$\left|\Pr\left[G_1 \Rightarrow 1\right]-\Pr\left[G_2 \Rightarrow 1\right]\right|\leqslant\mathrm{Adv}_{\mathrm{E,q}}^{\mathrm{ind\text{-}cpa}}(\mathcal{C})$$

在最终游戏 G_2 中，挑战者返回全随机的三元组 $\left(\mathrm{mid}_i,C_{\mathrm{tk}_{i-1}},\mathrm{tPk}\right)$，其分布与 $\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},0}$ 中返回的完全相同，因此

$$\Pr\left[G_2 \Rightarrow 1\right]=\Pr\left[\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},0} \Rightarrow 1\right]$$

综上分析，可得

$$\begin{aligned}&\mathrm{Adv}_{\mathrm{TDFTM}}^{\mathrm{p\text{-}conf}}(\mathcal{A})=\left|\Pr\left[\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},1} \Rightarrow 1\right]-\right.\\&\left.\Pr\left[\mathrm{PSConf}_{\mathrm{TDFTM}}^{\mathcal{A},0} \Rightarrow 1\right]\right|=\left|\Pr\left[G_0 \Rightarrow 1\right]-\Pr\left[G_2 \Rightarrow 1\right]\right|\\&\leqslant\mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{prf}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{E},q}^{\mathrm{ind\text{-}cpa}}(\mathcal{C})\end{aligned}$$

引理 1 证毕。

引理 2 用户追踪机密性。在定理 1 的假设下，TDFTM 对于任何 PPT 用户视图敌手 $\mathcal{A}$，都有

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\mathrm{u\text{-}conf}}(\mathcal{A})=0$$

证明 在 TDFTM 中，接收者视图包括接收验证生成的溯源密钥 $\mathrm{tk}'=\mathrm{H}((\mathrm{tPk}+\mathrm{Pk_s})\cdot\mathrm{sk_r})$、数据标识符 $\mathrm{mid}'=\mathrm{HMAC}(\mathrm{tk}',m)$、接收数据 m 和临时公钥 tPk。上述数据由本地计算生成，与是否为新创建或转发数据无关。由于溯源密钥 tk′ 与前一跳溯源元数据无交集，且 HMAC 与加密过程不泄露历史路径信息，故敌手在挑战位 b=0 与 b=1 下得到的接收者视图分布完全相同。因此，有

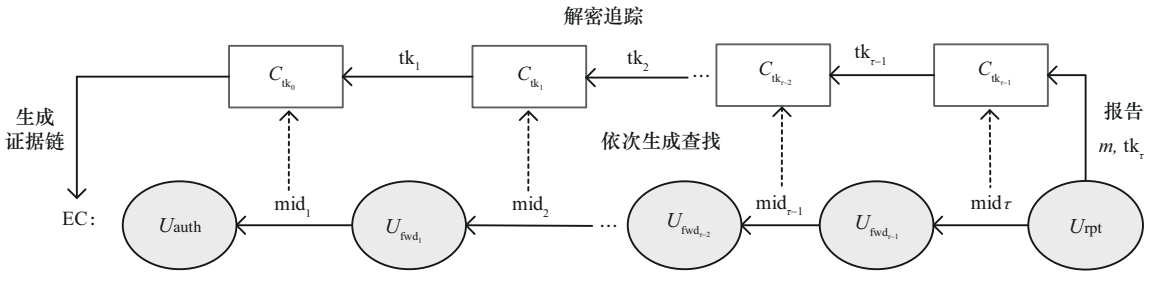


图 4 EC 生成流程

$$\left|\Pr\left[\mathrm{UTrConf}_{\mathrm{TDFTM}}^{\mathcal{A},1}\Rightarrow 1\right]=\Pr\left[\mathrm{UTrConf}_{\mathrm{TDFTM}}^{\mathcal{A},0}\Rightarrow 1\right]\right|\cdot$$

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{u-conf}}(\mathcal{A})=0$$

引理2证毕。

结合引理1和引理2，可得

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\mathrm{conf}}(\mathcal{A})\leqslant \mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{prf}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{E},q}^{\text{ind - cpa}}(\mathcal{C})$$

定理1得证。

定理2　不可否认性。假设HMAC函数是抗碰撞的安全PRF，哈希函数H抗碰撞，且椭圆曲线离散对数问题（ECDLP, elliptic curve discrete logarithm problem）在底层群上是困难的，则对于任何PPT敌手A，存在PPT敌手B和C使

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{non - rep}}(\mathcal{A})\leqslant \mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{cr}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{ECC},q}^{\mathrm{ecdlp}}(\mathcal{C})$$

证明　以下将根据安全游戏$\mathrm{NonRep}_{\mathrm{TDFTM}}^{\mathcal{A}}$中敌手2种否认情况进行分析。

1) 发送否认：敌手提交$(\mathrm{ID_U}^*,m^*,\mathrm{tk}^*)$，试图使计算生成的数据标识符$\mathrm{mid}^*=\mathrm{HMAC}(\mathrm{tk}^*,m^*)$存在于DB中，且对应条目的发送者为$\mathrm{ID_U}^*$，但状态表中$\mathrm{WasSent}(\mathrm{ID_U}^*,-,\mathrm{mid}^*,m^*)=\mathrm{false}$。

由于系统在每次发送操作时设置$\mathrm{WasSent}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)=\mathrm{true}$，因此满足上述条件必须依赖于敌手伪造一组不同于真实数据的(tk^*,m^*)使$\mathrm{HMAC}(\mathrm{tk}^*,m^*)=\mathrm{HMAC}(\mathrm{tk},m)$，但基于HMAC函数的抗碰撞性，找到这样的碰撞是困难的。即存在PPT敌手$\mathcal{B}$成功构造发送否认的优势上界为$\mathrm{Adv}_{\mathrm{HMAC}}^{\mathrm{cr}}(\mathcal{B})$。

2) 接收否认：敌手提交$(\mathrm{ID_U}^*,m^*,\mathrm{tk}^*)$，试图使计算生成的数据标识符$\mathrm{mid}^*=\mathrm{HMAC}(\mathrm{tk}^*,m^*)$存在于DB中，且对应条目的接收者为$\mathrm{ID_U}^*$，但状态表中$\mathrm{WasRec}(-,\mathrm{ID_U}^*,\mathrm{mid}^*,m^*)=\mathrm{false}$。

由于系统在数据完整性验证通过后设置$\mathrm{WasRec}(\mathrm{ID_s},\mathrm{ID_r},\mathrm{mid},m)=\mathrm{true}$，敌手必须伪造一个有效的$\mathrm{tk}^*$，使得$\mathrm{HMAC}(\mathrm{tk}^*,m^*)=\mathrm{mid}^*$匹配与数据库记录，且$\mathrm{tk}^*$应在未持有接收方私钥$\mathrm{sk}_{\mathrm{ID}_{u^*}}$的情况下构造。

在TDFTM中，溯源密钥的验证公式为$\mathrm{tk}'=\mathrm{H}((\mathrm{tPk}+\mathrm{Pk_s})\cdot\mathrm{sk_r})$，为了构造合法的$\mathrm{tk}^*$，敌手需从$\mathrm{Pk_r}=\mathrm{sk_r}\cdot G$反推出$\mathrm{sk_r}$，这将面临解决ECDLP的计算难题。因此，存在PPT敌手$\mathcal{C}$成功构造发送否认的优势上界为$\mathrm{Adv}_{\mathrm{ECC}}^{\mathrm{ecdlp}}(\mathcal{C})$。

综合上述2种情况，可得

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{non - rep}}(\mathcal{A})\leqslant \mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{cr}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{ECC},q}^{\mathrm{ecdlp}}(\mathcal{C})$$

证毕。

定理3　溯源可审计性。假设HMAC函数是抗碰撞的安全PRF，哈希函数H抗碰撞，ECDLP在底层群上是困难的，且对称加密方案$\mathrm{E}=(\mathrm{Enc},\mathrm{Dec})$满足IND-CPA安全性，则对于任何PPT敌手$\mathcal{A}$，存在PPT敌手$\mathcal{B}$、$\mathcal{C}$和D使

$$\mathrm{Adv}_{\mathrm{TDFTM}}^{\text{tr - audit}}(\mathcal{A})\leqslant \mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{cr}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{ECC},q}^{\mathrm{ecdlp}}(\mathcal{C})+\mathrm{Adv}_{\mathrm{E},q}^{\text{ind - cpa}}(\mathcal{D})$$

证明　以下将根据安全游戏$\mathrm{TrAudit}_{\mathrm{TDFTM}}^{\mathcal{A}}$中敌手4种攻击情况进行分析。

1) 数据替换攻击：敌手提交$(\mathrm{ID_U}^*,m^*,\mathrm{tk}^*)$，试图使数据标识符$\mathrm{mid}^*=\mathrm{HMAC}(\mathrm{tk}^*,m^*)$存在于DB中且对应发送者为$\mathrm{ID_U}^*$，但状态表存在记录$\mathrm{Audit}(\mathrm{ID_U}^*,-,\mathrm{mid}^*,m')=\mathrm{true}$且$m'\neq m^*$。为此敌手必须构造一组$(\mathrm{tk}^*,m^*)\neq(\mathrm{tk}',m')$以满足$\mathrm{HMAC}(\mathrm{tk}^*,m^*)=\mathrm{HMAC}(\mathrm{tk}',m')$，等价于在HMAC函数上构造碰撞。因此，敌手成功构造数据替换攻的优势上界为$\mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{cr}}(\mathcal{B})$。

2) 身份替换攻击：敌手提交$(\mathrm{ID_U}^*,m^*,\mathrm{tk}^*)$，试图使得$\mathrm{mid}^*$对应于发送者$\mathrm{ID_U}^*$向接收者$\mathrm{ID_r}^*$发送的数据，但状态表存在记录$\mathrm{Audit}(\mathrm{ID_U}^*,\mathrm{ID_r}',\mathrm{mid}^*,m')=\mathrm{true}$且$\mathrm{ID_r}'\neq\mathrm{ID_r}^*$。为此敌手可尝试2条路径：构造HMAC函数碰撞使得不同接收方信息混淆，或在未知接收者私钥的情况下伪造有效的溯源密钥tk^*通过验证。前者对应HMAC函数抗碰撞难题，后者等价于求解$\mathrm{tk}=\mathrm{H}((\mathrm{tPk}+\mathrm{Pk_s})\cdot\mathrm{sk_r})$即解决ECDLP难题。因此敌手成功构造身份替换攻击的优势上界为$\mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{cr}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{ECC},q}^{\mathrm{ecdlp}}(\mathcal{C})$。

3) 路径伪造攻击：敌手试图伪造EC，使其包含某一跳未在状态表中记录的伪造数据流转记录。在TDFTM中，EC基于密态溯源密钥链$\{C_{\mathrm{tk}_0},C_{\mathrm{tk}_1},C_{\mathrm{tk}_2},\cdots,C_{\mathrm{tk}_i}\}$构建，敌手伪造路径必须解密现有$C_{\mathrm{tk}_{i-1}}$获取前一跳$\mathrm{tk}_{i-1}$或伪造加密指针$C_{\mathrm{tk}}{}^*$使其通过平台溯源验证。前者可归约至对称加密方案的IND-CPA安全性，后者归约至ECDLP或HMAC函数抗碰撞安全性。因此敌手成功构造路径伪造攻击的优势上界为$\mathrm{Adv}_{\mathrm{HMAC},q}^{\mathrm{cr}}(\mathcal{B})+\mathrm{Adv}_{\mathrm{ECC},q}^{\mathrm{ecdlp}}(\mathcal{C})+\mathrm{Adv}_{\mathrm{E},q}^{\text{ind - cpa}}(\mathcal{D})$。

4) 流转记录篡改攻击：敌手试图篡改DB中的流转记录或插入伪造记录，破坏审计过程的完整性。具体包括替换已有键值对应的元数据内容，或

构造新的合法记录(mid*, tmd*)。由于平台采用HMAC函数计算mid并以其为唯一键值，敌手必须构造碰撞(tk, m)以生成已有键值或伪造C_{tk}*通过审计验证。此过程规约至HMAC函数抗碰撞与对称加密IND-CPA安全性。此外，溯源过程将验证用户身份的前后一致性，一旦出现异常情况将自动终止溯源。因此敌手成功构造流转记录篡改攻击的优势上界为$\mathrm{Adv}^{\mathrm{cr}}_{\mathrm{HMAC},q}(\mathcal{B})+\mathrm{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathrm{E},q}(\mathcal{D})$。

综合上述4种情况，可得

$$\mathrm{Adv}^{\mathrm{tr\text{-}audit}}_{\mathrm{TDFTM}}(\mathcal{A})\leqslant\mathrm{Adv}^{\mathrm{cr}}_{\mathrm{HMAC},q}(\mathcal{B})+\mathrm{Adv}^{\mathrm{ecdlp}}_{\mathrm{ECC},q}(\mathcal{C})+\mathrm{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathrm{E},q}(\mathcal{D})$$

证毕。

5 仿真与性能对比分析

本节将从安全性、计算开销、通信开销与存储成本对TDFTM与同类方案进行对比分析，并仿真说明TDFTM的性能优势。

5.1 安全性对比

本节分别从完整性、机密性、隐私性、不可否认性、全路径溯源和溯源可审计性等方面将TDFTM与文献[5,10,29,32-33]中的方案进行安全特征对比，如表2所示。根据方案采用密码原语的安全强度、是否与身份信息和数据操作强绑定、溯源信息保护内容和范围以及是否具备防伪造审计能力等方面综合考虑，将安全等级分为强、中、弱。

表2结果显示，文献[5]采用图论追踪法，适用于明文数据流转场景，由于未针对加密数据提供保护机制，无法满足密态数据流转的机密性和隐私保护需求，且明态存储的溯源信息存在被篡改风险。文献[29]的对称密钥设计未建立流转操作与用户身份绑定机制，难以有效抵抗用户抵赖攻击。文献[10]在文献[29]的基础上引入公钥签名机制，提供了身份认证和举报不可否认性，但开销显著增加。文献[32]聚焦于密态消息的授权控制与访问策略，但其机制仅支持单跳流转的可信验证，缺乏全路径溯源能力与完整的链式审计保障。文献[33]设计了E2EE群组消息传递中的分层治理框架，通过加密协议确保治理行为的隐私性和完整性，但其共享状态设计使得数据的历史状态及溯源过程对所有群组成员可见，存在信息泄露风险。

相较于现有方案，TDFTM在保障数据流转完整性与机密性的基础上，通过发送者、接收者与数据的三元密码学绑定，以及椭圆曲线密码学的私钥安全性，实现了用户操作行为的不可抵赖性；通过溯源密钥加密机制保障了数据传播路径和用户的隐私安全；基于密态密钥链生成了不可伪造的溯源证据链，实现了数据溯源的审计保障。综上，TDFTM的整体安全性优于其他方案。

5.2 计算开销对比

为评估TDFTM的计算效率，本节将TDFTM与同类方案的计算开销进行对比分析。

文献[10]采用数字签名、伪随机函数等密码学技术构建溯源机制，与本文具备相同的安全特性，数据流转溯源过程包括数据发送、接收和追踪3个阶段。由于2种方案均基于端到端加密系统构建，故参考文献[10]中端到端加密操作的计算开销模拟结果。方案相关密码学算法的计算开销如表3所示，其中，T_{h}表示哈希，T_{hmac}表示消息认证码，T_{ecm}表示ECC点乘，T_{sym}表示对称加/解密，T_{sig}表示签名，T_{vf}表示验签，$T_{\mathrm{E2EE\,Tx}}$表示端到端加密发送，$T_{\mathrm{E2EE\,Rx}}$表示端到端加密接收。

假设数据经k次转发后由用户报告并完成溯源。基于表3中的执行时间，评估各方案在用户端和服务器端执行的密码学运算次数及其计算开销，详情如表4所示。图5直观展示了数据经过10次转发并溯源后，不同方案的计算开销对比情况。

表2 安全特征对比

安全特征	文献[5]	文献[10]	文献[29]	文献[32]	文献[33]	TDFTM
完整性	有	有	有	有	有	有
机密性	无	有	有	有	有	有
隐私性	无	强	强	强	中	强
不可否认性	有	有	无	有	有	有
全路径溯源	有	有	有	无	有	有
溯源可审计性	弱	强	强	弱	中	强

表3　　相关密码学算法的计算开销

密码学算法	平均时间/μs
T_{h}	1.88
T_{hmac}	3.84
T_{ecm}	25.48
T_{sym}	4.98
T_{sig}	38.96
T_{vf}	97.81
$T_{\mathrm{E2EE\ Tx}}$	198.69
$T_{\mathrm{E2EE\ Rx}}$	174.42

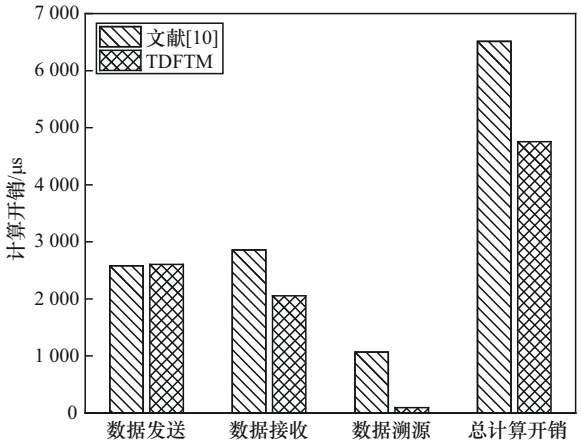


图5　不同方案的计算开销对比

表4　　计算开销对比

方案	用户端		平台服务器端	总计算开销
	数据发送	数据接收	数据溯源	
文献[10]	$kT_{\mathrm{sig}} + kT_{\mathrm{h}} + 3kT_{\mathrm{sym}} + kT_{\mathrm{hmac}} + kT_{\mathrm{E2EE\ Tx}}$	$kT_{\mathrm{vf}} + kT_{\mathrm{hmac}} + 2kT_{\mathrm{sym}} + kT_{\mathrm{E2EE\ Rx}}$	$kT_{\mathrm{vf}} + (k+1)T_{\mathrm{hmac}} + kT_{\mathrm{sym}}$	$kT_{\mathrm{sig}} + kT_{\mathrm{h}} + 2kT_{\mathrm{vf}} + (3k+1)T_{\mathrm{hmac}} + 6kT_{\mathrm{sym}} + kT_{\mathrm{E2EE\ Tx}} + kT_{\mathrm{E2EE\ Rx}}$
TDFTM	$kT_{\mathrm{h}} + kT_{\mathrm{sym}} + kT_{\mathrm{hmac}} + 2kT_{\mathrm{ecm}} + kT_{\mathrm{E2EE\ Tx}}$	$kT_{\mathrm{h}} + kT_{\mathrm{hmac}} + kT_{\mathrm{ecm}} + kT_{\mathrm{E2EE\ Rx}}$	$(k+1)T_{\mathrm{hmac}} + kT_{\mathrm{sym}}$	$2kT_{\mathrm{h}} + 2kT_{\mathrm{sym}} + (3k+1)T_{\mathrm{hmac}} + 3kT_{\mathrm{ecm}} + kT_{\mathrm{E2EE\ Tx}} + kT_{\mathrm{E2EE\ Rx}}$

图5结果表明，TDFTM在数据发送阶段的计算开销与文献[10]相近，但在数据接收和数据溯源阶段明显低于后者。方案的总计算开销包括用户端和服务器端在流转和溯源2个阶段的计算开销总和。数据经10次流转并溯源后，文献[10]方案总计算开销为 $10T_{\mathrm{sig}} + 10T_{\mathrm{h}} + 20T_{\mathrm{vf}} + 31T_{\mathrm{hmac}} + 60T_{\mathrm{sym}} + 10T_{\mathrm{E2EE\ Tx}} + 10T_{\mathrm{E2EE\ Rx}} \approx 6\,513.53$ μs，TDFTM总计算开销为 $20T_{\mathrm{h}} + 20T_{\mathrm{sym}} + 31T_{\mathrm{hmac}} + 30T_{\mathrm{ecm}} + 10T_{\mathrm{E2EE\ Tx}} + 10T_{\mathrm{E2EE\ Rx}} \approx 4\,751.74$ μs，TDFTM总计算开销较文献[10]减少了27%。这是因为文献[10]使用数字签名并加密签名以此绑定数据与发送者身份，将验证流转信息真实性与完整性的责任转移至数据接收者，使接收阶段的验签操作增加了计算开销，而TDFTM基于椭圆曲线算法生成收发双方身份密钥和临时密钥对，并采用基于哈希的消息认证码生成函数设计了溯源密钥在用户端的动态生成与验证机制，避免了额外的签名与验签操作，在实现数据流转可信性验证的同时降低用户端的计算负担。

在溯源阶段，文献[10]采用逐跳回溯记录机制，需在每一步验证发送用户签名以确保溯源链未被篡改，因此计算开销随溯源链长度而大量增加，而TDFTM通过对密态溯源密钥的递归解密直接定位可信的数据来源，未引入额外的密码学操作，有效降低了服务器端的计算开销。

5.3　通信开销与存储成本对比

为评估TDFTM的通信开销与溯源信息存储成本，本节对比了TDFTM和文献[10]的通信开销和数据库存储成本。通信开销中涉及的元数据长度如表5所示，其中，L_{h}表示哈希摘要长度，L_{hmac}表示消息认证码长度，L_{ecm}表示ECC点乘长度，L_{sym}表示对称加/解密长度，L_{ran}表示随机数长度，L_{sig}表示签名长度，L_{ID}表示ID长度，L_{t}表示时间戳长度。基于表5评估各方案在流转和溯源阶段总通信开销与数据库存储成本，详情如表6所示。图6直观展示了各方案通信开销与存储成本的对比情况。

表5　　元数据长度

元数据	长度/bit
L_{h}	256
L_{hmac}	256
L_{ecm}	160
L_{sym}	128
L_{ran}	256
L_{sig}	512
L_{ID}	16
L_{t}	32

表 6　通信开销与存储成本对比

方案	通信开销/bit	存储成本/bit
文献[10]	2 240	672
TDFTM	960	416

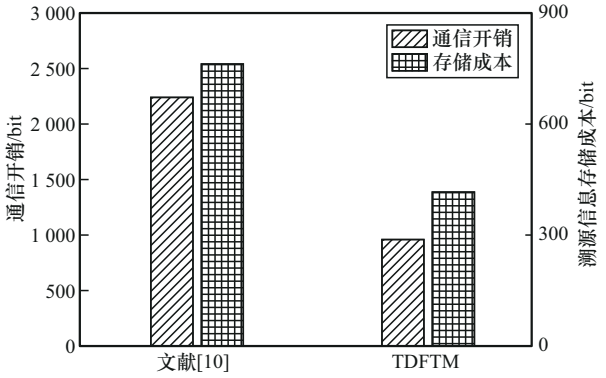


图 6　方案通信开销与存储成本对比

表 6 结果表明，TDFTM 在通信开销方面较文献[10]降低了 57%，主要原因在于文献[10]需要在数据发送者与平台之间频繁交换数字签名，导致较高的通信开销。在溯源信息存储成本方面，文献[10]需额外存储密态公钥及密态签名数据以支持溯源时的用户验证，因此 TDFTM 较文献[10]减少了 38%存储成本，从而降低大规模场景下通信存储负担。

5.4　仿真测试

本节使用 Python 语言，调用 cryptography 库，针对 TDFTM 各阶段算法进行了仿真，并记录多次重复运行后的平均耗时以保证实验数据的可靠性。具体仿真实验环境配置如表 7 所示。

实验结果如表 8 所示，各算法的主要操作运行时间均处于微秒级别，展现出较高的执行效率。

表 7　仿真实验环境配置

硬件工具	版本/型号
操作系统	Windows 10
CPU	Intel(R)Core(TM)i5-8300H 2.3 GHz
内存	16 GB RAM
数据库	MySQL 8.0.36

表 8　算法运行时间

算法	平均运行时间/μs
NewMsg	64.06
FwdMsg	62.58
Plat-Process	149.78
RecMsg	43.10
ECGen	63.61

客户端开销集中在数据发送与接收验证。实验结果显示，溯源标识的生成、验证总耗时约为 126.74 μs，整体开销较低。其中，初始发送算法与转发算法中链接标记与数据标识符生成操作执行基本一致，二者耗时差异仅源于初始化部分涉及的初始溯源密钥生成操作。

服务器端的开销包括溯源元数据处理和溯源证据链生成 2 个部分。溯源元数据的处理与存储不涉及密码学运算，主要执行键值存储以及溯源标识传递，生成一条可信流转记录耗时约为 149.78 μs。溯源阶段，图 7(a)展示了溯源追踪时间随流转路径溯源长度（即转发次数）的增长趋势，整体结果表明，溯源证据链的溯源追踪时间与数据流转路径溯源长度呈线性关系，随流转路径溯源长度增长线性递增。这种线性增长趋势的原因在于溯源机制逐跳

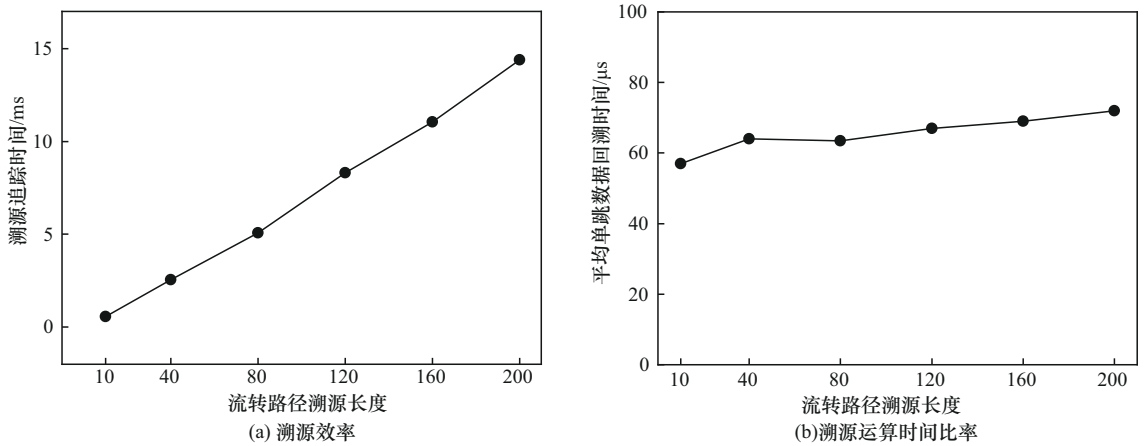


图 7　溯源追踪性能测试

遍历溯源元数据，并在每一跳执行恒定数量的密码学运算，包括一次对称密钥解密和HMAC验证。即使数据经历200次转发，溯源追踪时间仍控制在约15 ms。图7(b)展示了溯源运算时间比率，即平均单跳数据回溯时间。结果表明，单跳溯源解密与追踪操作的平均耗时约为63.61 μs，说明即使面对较大规模的数据溯源任务，仍能解析数据流转路径确保溯源过程的低时延。

综上，TDFTM在保证了安全性、隐私性的同时，还具有良好的性能表现与开销可行性，能够为数据流转大规模且高频的即时通信平台提供较为高效的技术支持。

6　结束语

针对现有流转溯源系统在数据安全保障、溯源范围和计算开销等方面存在的局限性，本文提出了一种针对端到端加密通信系统的数据流转可信溯源机制，应用于数据流转记录的可信生成与流转路径的可信溯源取证。基于数据收发双方身份密钥和临时密钥对设计溯源密钥的动态生成与验证机制，解决了因缺乏身份绑定和路径完整性保障导致的假冒攻击和抵赖攻击问题。基于HMAC函数的抗碰撞性，将消息内容与溯源密钥绑定，生成唯一数据标识符，并以其作为平台存储记录的键值用于映射密态溯源元数据，在满足加密通信平台隐私性需求的同时，确保记录抗篡改性与完整性验证。在数据流转过程中，采用溯源密钥加密机制动态生成并存储密态溯源密钥链，形成具备保护流转隐私的防篡改路径记录。该密钥链的解密密钥仅在平台和最终报告用户之间共享，平台根据报告用户提供的溯源数据逐级解密溯源密钥链，实现数据流转路径的可信追踪取证，从而有效平衡隐私性与溯源能力。安全性分析和仿真结果表明，TDFTM能够满足数据流转溯源的安全性目标，在计算效率与存储成本等方面优于同类方案。

TDFTM主要采用基于端对端单播通信的溯源密钥，并以链式结构记录数据流转路径，在群聊、社交传播等复杂传播环境下存在一定局限性。未来设计代表多播通信的溯源密钥生成方案，并采用基于树或者图结构的方式记录数据流转路径，以适应一对多或者多对多复杂场景的数据流转溯源，提升系统的适用性与扩展性。同时，优化树或者图结构，实现更加稳定、高效追踪所有分支路径。

参考文献：

[1] 杨学成, 陶晓波, 岳欣. 双有限异质社交网络仿真建模及实证分析[J]. 北京邮电大学学报, 2015, 38(S1): 121-124.
YANG X C, TAO X B, YUE X. Double limited and heterogeneous social network: a simulation and empirical study[J]. Journal of Beijing University of Posts and Telecommunications, 2015, 38(S1): 121-124.

[2] CHANG B, XU T, LIU Q, et al. Study on information diffusion analysis in social networks and its applications[J]. International Journal of Automation and Computing, 2018, 15(4): 377-401.

[3] PSALLIDAS F, AGRAWAL A, SUGUNAN C, et al. OneProvenance: efficient extraction of dynamic coarse-grained provenance from database query event logs[J]. Proceedings of the VLDB Endowment, 2023, 16(12): 3662-3675.

[4] 冷涛, 蔡利君, 于爱民, 等. 基于系统溯源图的威胁发现与取证分析综述[J]. 通信学报, 2022, 43(7): 172-188.
LENG T, CAI L J, YU A M, et al. Review of threat discovery and forensic analysis based on system provenance graph[J]. Journal on Communications, 2022, 43(7): 172-188.

[5] 杜娟, 苏秋月. 基于DAG的Hive数据溯源方法[J]. 信息技术与网络安全, 2020, 39(11): 31-37.
DU J, SU Q Y. Hive data provenance method based on DAG[J]. Information Technology and Network Security, 2020, 39(11): 31-37.

[6] GAO Y Z, CHEN X Y, LI B L, et al. A near real-time big data provenance generation method based on the conjoint analysis of heterogeneous logs[J]. IEEE Access, 2023, 11: 80806-80821.

[7] BARTUSEK J, GARG S, JAIN A, et al. End-to-end secure messaging with traceability only for illegal content[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2023: 35-66.

[8] HARTEL P, VAN WEGBERG R. Going dark? Analysing the impact of end-to-end encryption on the outcome of Dutch criminal court cases[J]. Crime Science, 2023, 12(1): 5.

[9] GRUBBS P, LU J H, RISTENPART T. Message franking via committing authenticated encryption[C]//Advances in Cryptology – CRYPTO 2017: 37th Annual International Cryptology Conference. Berlin: Springer, 2017: 66-97.

[10] KENNEY E, TANG Q, WU C S. Anonymous traceback for end-to-end encryption[C]//European Symposium on Research in Computer Security. Berlin: Springer, 2022: 42-62.

[11] SENGUPTA B, LI Y J, BU K, et al. Privacy-preserving network path validation[J]. ACM Transactions on Internet Technology, 2020, 20(1): 1-27.

[12] GHOSHAL D, PLALE B. Provenance from log files: a bigdata problem[C]//Proceedings of the Joint EDBT/ICDT 2013 Workshops. New York: ACM Press, 2013: 290-297.

[13] 杨彬, 高俊涛, 王志宝, 等. 基于词嵌入的元组级数据溯源方法[J]. 计算机技术与发展, 2023, 33(12): 49-57.
YANG B, GAO J T, WANG Z B, et al. A tuple-level data lineage approach based on word embedding[J]. Computer Technology and Development, 2023, 33(12): 49-57.

[14] LOU P, LU G T, JIANG X M, et al. Cyber intrusion detection through association rule mining on multi-source logs[J]. Applied Intelligence, 2021, 51(6): 4043-4057.

[15] WANG R Y, SUN D, LI G Q, et al. LogProv: Logging events as provenance of big data analytics pipelines with trustworthiness[C]//Proceedings of the 2016 IEEE International Conference on Big Data (Big Data). Piscataway: IEEE Press, 2016: 1402-1411.

[16] SIDDIQUI M S, RAHMAN A, NADEEM A. Secure data provenance in IoT network using bloom filters[J]. Procedia Computer Science,

2019, 163: 190-197.
[17] LIANG X P, SHETTY S, TOSH D, et al. ProvChain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability[C]//Proceedings of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). Piscataway: IEEE Press, 2017: 468-477.
[18] ZENG Y, ZHANG X, AKHTAR R, et al. A blockchain-based scheme for secure data provenance in wireless sensor networks[C]//Proceedings of the 2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). Piscataway: IEEE Press, 2018: 13-18.
[19] RAMACHANDRAN A, KANTARCIOGLU D M. Using Blockchain and smart contracts for secure data provenance management[J]. arXiv Preprint, arXiv: 1709.10000, 2017.
[20] DING Y P, SATO H. Sunspot: a decentralized framework enabling privacy for authorizable data sharing on transparent public blockchains[C]// International conference on algorithms and architectures for parallel processing. Berlin: Springer, 2022: 693-709.
[21] HASAN R, SION R, WINSLETT M. Preventing history forgery with secure provenance[J]. ACM Transactions on Storage, 2009, 5(4): 1-43.
[22] WANG X L, ZENG K, GOVINDAN K, et al. Chaining for securing data provenance in distributed information networks[C]//Proceedings of the MILCOM 2012 - 2012 IEEE Military Communications Conference. Piscataway: IEEE Press, 2012: 1-6.
[23] AHMED I, KHAN A, KHAN M S, et al. Aggregated signatures for chaining: a secure provenance scheme[C]//Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA. Piscataway: IEEE Press, 2016: 2012-2017.
[24] RANGWALA M, LIANG Z L, PENG W, et al. A mutual agreement signature scheme for secure data provenance[J]. Environments, 2016, 13(14): 726-733.
[25] FARAJ O, MEGIAS D, GARCIA-ALFARO J. Security approaches for data provenance in the Internet of Things: a systematic literature review[J]. ACM Computing Surveys, 2025, 57(10): 1-41.
[26] 李凤华, 孙哲, 牛犇, 等. 跨社交网络的隐私图片分享框架[J]. 通信学报, 2019, 40(7): 1-13.
LI F H, SUN Z, NIU B, et al. Privacy-preserving photo sharing framework cross different social network[J]. Journal on Communications, 2019, 40(7): 1-13.
[27] TYAGI N, GRUBBS P, LEN J, et al. Asymmetric message franking: content moderation for metadata-private end-to-end encryption[C]//Advances in Cryptology-CRYPTO 2019: 39th Annual International Cryptology Conference, Berlin: Springer, 2019: 222-250.
[28] ISSA R, ALHADDAD N, VARIA M. Hecate: abuse reporting in secure messengers with sealed sender[C]//Proceedings of the 31st USENIX Security Symposium (USENIX Security 22). Berkeley: USENIX Association, 2022: 2335-2352.
[29] TYAGI N, MIERS I, RISTENPART T. Traceback for end-to-end encrypted messaging[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2019: 413-430.
[30] PEALE C, ESKANDARIAN S, BONEH D. Secure complaint-enabled source-tracking for encrypted messaging[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2021: 1484-1506.
[31] LIAN R, MING Y L, CAI C J, et al. Nemesis: combating abusive information in encrypted messaging with private reporting[C]//Computer Security - ESORICS 2024: European Symposium on Research in Computer Security. Berlin: Springer, 2024: 247-267.
[32] JIANG P, LIU Q, ZHU L H. Purified authorization service with encrypted message moderation[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 5196-5206.
[33] NAMAVARI A, WANG B, MENDA S, et al. Private hierarchical governance for encrypted messaging[C]//Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2024: 2610-2629.

[作者简介]

谢绒娜（1976– ），女，山西永济人，博士，北京电子科技学院教授、博士生导师，主要研究方向为网络与系统安全、访问控制、密码工程。

王嘉烜（1999– ），女，江苏南京人，北京电子科技学院硕士生，主要研究方向为信息安全、数据溯源。

王文鼎（1999– ），男，四川德阳人，北京电子科技学院硕士生，主要研究方向为信息安全、云环境下完整性验证。

史国振（1974– ），男，河南济源人，博士，北京电子科技学院教授、博士生导师，主要研究方向为密码信息安全、信息论与编码理论。

周创（2001– ），男，安徽池州人，北京电子科技学院硕士生，主要研究方向为信息安全、数据溯源。

张玲翠（1986– ），女，河北故城人，博士，中国科学院信息工程研究所高级工程师，主要研究方向为网络与系统安全。