# 面向移动边缘计算的密钥管理协议

蒋京玮 [1)] 汪定 [2),3)] 张国印 [1)] 陈志远 [1)]

[1)](哈尔滨工程大学 计算机科学与技术学院，哈尔滨 150001)

[2)](南开大学 网络空间安全学院，天津 300350)

[3)](天津市网络与数据安全技术重点实验室 (南开大学), 天津 300350)

**摘 要** 移动边缘计算(Mobile Edge Computing，MEC)将互联网服务环境和云计算技术在网络边缘相结合，为移动用户提供高带宽、低延迟的计算和存储服务。在 MEC 网络中，移动用户和服务器之间组成一个群组进行通信和数据传输，一旦密钥泄露，将会造成严重的安全隐患和经济损失。通过设计密钥管理协议，可以实现密钥的分发、更新和存储等功能来保护 MEC 网络隐私安全。然而传统的密钥管理协议不适合 MEC 网络。因此需要结合 MEC 网络的物理特性和实际需求，设计面向 MEC 的密钥管理协议。本文的研究分为两部分：首先构造了一个非交互透明零知识可验证多项式委托并基于该技术设计了一种支持通信秘密可验证、秘密份额可动态更新、服务器节点可动态变化的门限秘密共享协议，保护 MEC 服务器私钥的安全；然后，提出一种支持用户自由进出，保障网络通信安全的 MEC 通信密钥管理协议，保护 MEC 网络中移动用户的通信和数据安全。进一步通过形式化证明和混合仿真游戏对所提协议进行了严格的安全证明，结果表明本文密钥管理协议满足前向安全和后向安全，并可抵抗各类已知攻击。与相关协议的对比结果显示, 新协议在提高安全性的同时，保持了较高的效率，适于 MEC 网络环境。

**关键词** 移动边缘计算；可验证秘密共享；透明零知识可验证多项式委托；动态更新；密钥管理

**中图法分类号** TP309.08 **DOI 号：**

# Private key management scheme for mobile edge computing

JIANG Jing-Wei[1)] WANG Ding[2),3)] ZHANG Guo-Yin[1)] CHEN Zhi-Yuan[1)]

[1](College of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

[2](College of Cyber Science, Nankai University, Tianjin 300350)

[3](National Engineering Laboratory of Mobile Network Security (Nankai University), Tianjin 300350)

**Abstract** Mobile edge computing (MEC) combines the Internet service environment and cloud computing technology at the edge of the network to provide mobile users with high bandwidth and low latency computing and storage services. In the MEC network, mobile users and servers form a group for communication and data transmission. Once the private key of the server or the data encryption key of mobile users is leaked, it will cause serious security risks and economic losses, and this loss is often irreversible. By designing the key management protocol for MEC, the functions of key distribution, update, and storage can be realized to protect the privacy and security of the MEC network. However, the node structure in MEC changes dynamically in real-time: on the one hand, with the continuous development of MEC, the servers in the MEC network will be added or replaced; On the other hand, mobile users in the MEC network will frequently join or exit the network. Therefore, key management protocols need to ensure forward security and backward security, and also need to consider resisting collusion attacks in the distributed key management protocol, which brings challenges to how to design the private key management scheme for MEC.

In recent years, researchers have proposed some distributed network key management protocols, but these key

本课题得到国家自然科学基金 No. 62172240、京津冀基础研究合作专项（22JCZDJC00717）资助。蒋京玮, 男, 1994 年生, 博士生, 主要研究领域为密码与信息安全，E-mail: jiangjingwei@hrbeu.edu.cn，手机号：18646571532。汪定（**通信作者**），男，1985 年生，教授, 博士生导师，主要研究领域为密码学与信息安全, E-mail: wangding@nankai.edu.cn。张国印，男，1962 年生，教授, 博士生导师，主要研究领域为网络信息安全与嵌入式系统，E-mail: zhangguoyin@hrbeu.edu.cn。陈志远, 男，1978 年生，副教授, 研究领域为信息和网络安全、模型检测，E-mail: chenzhiyuan@hrbeu.edu.cn。

management protocols are not suitable for MEC networks. Because there are a large number of mobile devices in the MEC network, the traditional network key management protocol is difficult to meet the needs of MEC network users for frequent free movement, device location identification, and low communication delay, and it is difficult to be directly applied to MEC network. To realize the MEC server to generate authentication certificate, communication key, and broadcast key through the private key, manage the mobile users logging in to the MEC network, and ensure that the mobile users and MEC server can use the encryption key for secure communication and data transmission. Therefore, it is necessary to design a key management protocol for MEC based on the physical characteristics and actual requirements of the MEC network. The research of this paper is divided into two parts: firstly, a non-interactive transparent zero-knowledge verifiable polynomial delegate is constructed, and based on this technology, a threshold secret sharing protocol supporting verifiable communication secret, dynamic update of secret share, and dynamic change of server node is designed to protect the security of MEC servers private key; Then, a MEC network key management protocol is proposed to support users' free access and ensure the security of network communication, to protect the communication and data security of mobile users in MEC network. Then we strictly prove the security of the proposed protocol through formal proof and hybrid simulation games. The results show that the private key management scheme for MEC proposed in this paper meets forward security and backward security, and can resist all kinds of known attacks. Compared with related protocols, the results show that the new protocol not only improves security but also maintains high efficiency, which is suitable for the MEC network environment.

**Key words** Mobile edge computing; verifiable secret sharing; transparent zero knowledge; dynamically updatable; private key management

## 1. 引言

移动边缘计算(Mobile Edge Computing，MEC)是一种使用无线接入网为移动用户提供互联网技术和云计算能力的一种新型的计算方式[1],被认为是现代蜂窝基站研究的关键问题之一，为 5G 通信的发展提供了重要的技术支持[2]。如图 1 所示[1,2]，一个典型的 MEC 网络由多个 MEC 服务器共享计算、存储和带宽容量。与传统的网络架构模式相比，MEC 具有以下三点明显的优势：

1) MEC 可以将计算和存储能力部署在网络边缘，由于 MEC 服务器与用户或信息源的地理距离更近，因此 MEC 可以直接处理用户的请求，极大的降低了通讯时延，提高了通讯效率[3,4]。
2) MEC 服务器可以在本地执行部分计算卸载和缓存，从而减少核心网的计算和传输消耗[5]，降低 MEC 网络中传输信息的能量消耗。
3) 部署在无线接入网的 MEC 服务器可以感知链路信息并调度带宽资源，优化用户体验[6]。

因此，MEC 在计算密集型应用，如增强和虚拟现实、智能视频加速、车联网以及物联网等方面有着广泛的应用[5]。然而，在 MEC 网络中，由于设备异构性强且网络环境复杂，存在信息安全隐患，需要密码学手段保护 MEC 网络的信息安全。

在密码学系统中，密钥作为安全传输、身份认证[7-10]以及数据加密的基础，为用户的隐私安全提供了有力的保障。一旦密钥丢失或泄露，将会造成严重的损失，并且这种损失往往是不可逆转的。因此需要私钥管理协议保证私钥安全。在 MEC 网络中，MEC 服务器可以通过私钥生成认证证书、通信密钥和广播密钥对登录到 MEC 网络的移动用户进行管理。一旦私钥丢失或泄露，敌手将可以伪造身份、随意授权、造成用户隐私的泄露。所以需要研究一种适用于 MEC 的私钥管理协议对 MEC 服务器的私钥进行保护。门限秘密共享是一种解决私钥管理问题的可行方法。

然而随着 MEC 的不断发展，将会出现越来越多的 MEC 服务器为移动用户提供服务，同时也会出现 MEC 服务器故障导致更换服务器的情况。为了保证服务器私钥的安全，在设计 MEC 服务器私钥的秘密共享协议时需要考虑以下两个问题：

1) 门限秘密共享协议应该支持服务器节点在规定的时间间隔中动态的加入和退出 MEC 网络。
2) 在服务器节点动态变化的同时，应该考虑到敌手可能进行的腐化攻击问题。在满足$(t,n)$门限安全假设的前提下，敌手可能最多同时控

制网络中的 $t$ 个服务器节点和 $t$ 个新加入的服务器节点，突破门限安全假设，获取私钥。

从整个 MEC 网络的角度看，MEC 网络主要包含 MEC 服务器和移动用户两种角色[2]。移动用户可与 MEC 网络中的任何 MEC 服务器进行通信，请求 MEC 服务器响应并处理计算或存储请求。在完成用户身份认证后，MEC 服务器需要通过私钥生成用户通信密钥和广播密钥，通过广播密钥可以向用户分发流量加密密钥[11-13]。此时，MEC 网络中的移动用户和服务器可以进行安全通信和数据传输。然而， 由于 MEC 网络中存在海量的移动设备，现有基于云计算的网络密钥管理协议[11-13]难以满足 MEC 网络用户对频繁自由移动、设备位置识别以及低通信延迟等需求[14]，难以直接应用于 MEC 网络中。为了实现移动用户和 MEC 服务器之间地相互通信和传输数据，需要研究面向 MEC 的通信密钥管理协议。在设计面向 MEC 的通信密钥管理协议时，需要在保证 MEC 通信和数据传输安全基础上，考虑以下两点问题：

1) MEC 服务器相比与网络中心的云服务器，计算能力有限。在 MEC 网络中，移动用户可能会频繁的加入或退出，而用户频繁的加入或退出网络会导致 MEC 服务器需要频繁的生成或更新通讯密钥、广播密钥和流量加密密钥，将会增加 MEC 服务器的计算负担[13]。
2) 面向 MEC 的通信密钥管理协议需要保证前向安全性和后向安全性，即新加入的移动用户不能通过密钥解密之前的 MEC 网络中的消息，而离开的移动用户无法使用之前的密钥解密退出后 MEC 网络中的消息。

基于上述考虑，本文将面向 MEC 的密钥管理方案分为两个部分：通过研究一种支持通信秘密可验证、秘密份额可动态更新、服务器节点可动态变化的门限秘密共享协议保障 MEC 服务器私钥安全；并设计一种支持用户自由进出，保护网络通信安全的 MEC 通信密钥管理协议，保障 MEC 网络中移动用户的隐私和信息安全。

### 1.1 相关工作

Shamir[15]和 Blakley[16]在分别基于 Lagrange 插值法和映射几何理论提出了秘密共享的概念。 为了防止可信中心造成的安全隐患，Ingemarsson 等人[17]提出无可信中心的秘密共享协议。 在该秘密

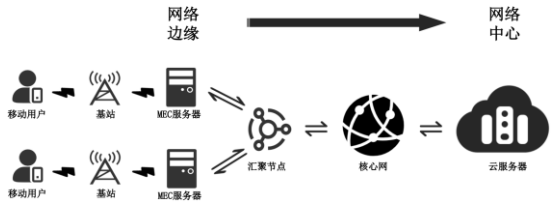


图 1 MEC 网络架构

共享中，二元多项式可以直接为成员之间提供会话密钥[18]，并且可以通过虚化[19]技术在人员动态更新时暂时提高秘密份额的安全门限，进一步提高协议的安全性和实用性，从而得到了广泛的关注，可以用来构造秘密共享协议[20]。

早期的秘密共享要求分发者和参与者都是诚实的，而实际上秘密共享协议无法避免外部攻击和内部欺骗的情况发生[21]。外部攻击是指敌手骗取参与者的秘密份额，从而恢复出秘密；内部欺骗分为两种情况：一种是分发者提供给参与者无效的秘密份额；另一种是参与者提供假的秘密份额，使秘密恢复失败。构造一个可验证、可动态更新的秘密共享协议可以解决上述问题[22]。

为保证秘密份额的有效性，Chor 等人[21]提出了可验证秘密共享协议，但是该协议通讯复杂度较高，且只能验证分发者的正确性而不能杜绝参与者的欺骗行为。 Harn 等人[22]基于中国剩余定理提出一种无计算假设和前提条件的可验证秘密共享协议，但该协议中仅参与者可以验证秘密份额的正确性。 Zaghian 等人[23,24]基于非齐次线性递归提出了公开可验证的秘密共享协议，这两个协议在合适的门限条件下效率高于 Schoenmakers 等人[25]提出的可验证秘密共享协议，但该协议不支持用户的动态加入和退出，因此无法适应 MEC 网络中移动用户频繁加入或退出网络的情况。

Herzberg 等人[26]提出了一种动态秘密共享协议，在保持秘密 $s$ 不变的前提下，通过定期更新参与者的秘密份额，增强协议的安全性。 Maram 等人[27]使用维度转换技术提出一种支持动态委员会成员动态变更的秘密共享协议，但是该协议验证使用的零知识证明需要可信初始化，不适合分布式环境，难以高效运行。此外，该算法在验证阶段使用了双线性对运算，实际验证效率较低。

在可验证的秘密共享协议中，零知识证明[28]是普遍使用的验证手段。Pedersen[29]将非交互式零知识证明引入秘密共享，提出了基于多项式的秘密共享协议。早期的零知识证明通过概率可验证

明构造[30,31]。随着零知识证明的进一步发展，基于二次算数程序[32-35]的 zk-SNARKs(zero-knowledge succinct non-interactive arguments of knowledge)协议[36-42]被广泛用于数字货币[43]和智能合约[44,45]中。Kate 等人[46]基于 zk-SNARKs 构造了一个秘密共享协议，然而该协议中的零知识证明需要一个可信中心执行初始化来生成结构化引用字符串(Structured Reference Strings, SRS)，该操作使证明效率降低，内存消耗增大，并且难以扩展到大型语句。虽然 zk-SNARKs 可以采用多方安全计算[47]生成具有普适性[48]和可更新[49]的 SRS，但这个算法操作十分复杂，且花销极高。难以应用于设备计算能力有限的 MEC 网络中。

为进一步提升零知识证明的运行效率，研究人员基于不同的技术来去除可信初始化，进而构建透明的零知识证明协议。在离散对数困难问题下，研究人员提出了几种透明零知识证明协议[50-53]。其中 Bunz 等人[53]提出的 Bulletproofs，可以高效的批处理多秘密证明，十分适合在区块链中使用。但其使用的范围证明技术存在承诺范围的限制。Ames 等人[54]通过使用“(MPC)-in-the-head”[55-57] (multi-party computation)技术，构建了一种交互式谕言机证明(interactive oracle proofs, IOPs)，并提出了透明零知识证明 Ligero。但该算法只能使用对称密钥操作，验证时间和电路大小呈拟线性关系。同样不适合在 MEC 网络中使用。

在 IOPs 下，Sasson 等人[58]基于 RAM(random-access memory)模型提出了一种可扩展的透明零知识证明 Stark，该协议的验证时间与 RAM 程序呈线性关系，但是验证的扩展速度是呈指数级增长的。Bootle 等人[59]基于哈希函数加密提出了一种高效的透明零知识证明协议，该协议的承诺协议无需同态性，且证明和验证的时间均为线性。Ben-Sasson 等人[60]在 IOPs 的模型下使用哈希函数加密构造透明零知识证明协议 Aurora，进一步提高了证明和验证的执行效率。

Zhang 等人[61]通过可验证多项式委托协议将分层算术电路[62]扩展到一个证明系统。在此基础上，研究人员通过 Cramer-Damgard 变换[63]和随机隐蔽多项式[64]构造了各具特色的零知识证明系统[65-68]。Zhang 等人[69]通过结合分层算术电路[62]提出了一种简洁透明的零知识证明协议。该协议不需要使用门限和线性大小的公钥。通过 Merkle 树承诺，避免模幂运算和双线性对运算，是目前 IOPs 下证明生成速度最快的交互式证明协议。

网络密钥管理协议[11,12]可以用来保护网络的通信密钥。在 MEC 网络中，移动设备异构性强，网络环境复杂，所以密钥管理协议具有高复杂度和通信开销，因此现有的网络密钥管理协议不适合 MEC[14]。为了支持移动用户跨区通信，Daghighi 等人[13]提出一种分级组密钥管理协议，该协议降低了用户跨区移动时带来的通信开销，并保证了用户移动情况下的前向安全和后向安全。考虑到通信中包含的敏感数据，Challal 等人[79]提出了一种基于雾计算的群组密钥管理协议，该协议允许移动用户在弹性安全域内共享敏感服务。具体应用于 MEC 网络中，Li 等人[81]提出了一种基于区块链的 MEC 密钥管理协议。该协议通过子网区块链存储移动用户的公钥，高层区块链验证身份的方式，保障用户可以进行安全通信和数据加密的基础上，允许用户进行提供跨区通信。

### 1.2 研究动机与贡献

网络密钥管理[11-13]可以在 MEC 网络的设备之间实现密钥分发、密钥更新和密钥存储等操作的密码学技术。在设计面向 MEC 的密钥管理协议时，需要适应分布式网络，考虑安全性、高效性和可扩展性，同时还需要降低协议的能耗开销[14]。

在适合分布式网络方面，由图 1 中可以看出，在 MEC 网络中服务器节点与网络中心的云服务器通信是复杂的，存在延时大、能耗大的问题。所以在密钥管理中，应该减少网络中心的云服务器参与计算和存储，MEC 服务器私钥的生成、存储以及秘密份额的更新都由 MEC 服务器自行完成。

在安全方面，在密钥管理协议安全性的基础上，还需要考虑设备节点的动态性。在私钥的管理方面，需要考虑 MEC 服务器的动态变化带来的安全问题。另一方面在 MEC 网络中，密钥管理协议需要考虑移动用户频繁的加入或退出带来的计算负担。面向 MEC 的密钥管理协议需要保证前向安全性和后向安全性，并且可以抵抗合谋攻击。

在效率方面，私钥管理协议基于可验证秘密共享协议[19,21,27,29]，然而可信初始化、双线性对、模幂运算等复杂的技术和运算在 MEC 网络中并不适用。需要考虑使用一种无需可信初始化的零知识可验证多项式委托协议(zero-knowledge Verifiab-

le Polynomial Delegation, zk-VPD)进行秘密份额验证。对 MEC 中移动用户的密钥管理，需要使用轻量级的对称密钥进行通信和数据加密。

最后在计算开销方面， MEC 网络中服务器和移动用户都具有动态特性，设计密钥管理协议需要避免 1-affect-n 的情况，即避免一个设备的加入或退出需要群组内所有设备配合计算的情况[13,77,92]。

基于上述考虑，本文将针对 MEC 网络的分布式架构，复杂的网络环境和用户频繁动态变化等特点[14]，分别提出面向 MEC 服务器和通信密钥管理协议。本文的贡献可分为四个方面：

1) 为了构造高效的验证协议，本文基于 Zhang 等人[59]提出的 Virgo 协议造了一种非交互式透明零知识可验证多项式委托协议。该协议无需可信初始化，适用于无可信中心的分布式网络，且该方案无需模幂运算，可为 MEC 服务器私钥管理协议提供高效的验证协议。
2) 为了保障 MEC 服务器私钥的安全，本文基于二元多项式和透明零知识证明协议设计了一个面向分布式环境下的可验证动态私钥管理协议，将 MEC 服务器私钥分布式存储，避免单点故障造成私钥丢失或泄露。该协议在保障私钥安全的基础上，允许私钥管理服务器动态变化，而无需重新生成私钥。
3) 为了方便 MEC 网络中的移动用户高效安全的进行通信和数据加密，本文设计了一个无中心的通信密钥管理协议，通过 MEC 服务器为移动用户生成通信密钥和数据加密密钥。协议允许移动用户频繁加入或退出网络，避免 1-affect-n。协议满足前向安全和后向安全。
4) 通过形式化证明和安全性分析，证明了非交互零知识证明协议具有完备性、可靠性和零知识性；证明了私钥管理协议支持秘密份额验证，允许私钥秘密份额管理服务器动态变更，抵抗各类已知的攻击；证明了密钥管理协议可以安全高效的管理移动用户，为移动用户的通信和数据加密提供安全保障，协议具有前向安全性和后向安全性。

### 1.3 组织结构

在第 2 节介绍了安全模型及评价指标；在第 3 节回顾了构造透明零知识证明协议所用到的基本技术，并提出一种非交互式的透明零知识可验证多项式委托协议；在第 4 节给出了非交互式的透明零知识可验证多项式委托协议的形式化分析；在第 5 节基于二元多项式提出了一种可验证动态私钥管理协议用于 MEC 服务器的私钥管理；并在第 6 节分析了私钥管理协议的安全性；在第 7 节提出了面向 MEC 通信密钥管理协议；并在第 8 节给出了密钥管理协议的安全性分析；在第 9 节分析了上述协议的效率；在第 10 节与相关协议进行对比；最后在第 11 节对本文的工作进行总结。

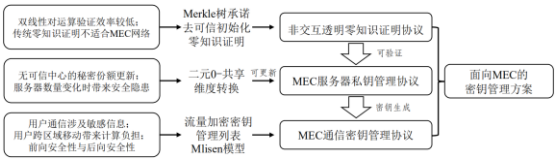


图 2 主要行文逻辑

特别地，本文提出三个协议之间的关系如图 2 所示。首先面向 MEC 服务器的私钥管理，通过第 3 节构造一个安全的非交互透明零知识证明为构造的 MEC 服务器私钥管理协议提供一个适合分布式网络的高效的验证秘密份额的方法。并在第 5 节提出一种 MEC 服务器私钥管理协议。MEC 服务器私钥的安全管理，可以为 MEC 通信密钥生成提供安全保障。在 7 节提出面向 MEC 移动用户通信密钥的管理协议。本文第 3、5、7 节提出的三个协议为递进关系，最终由 MEC 服务器私钥管理协议和 MEC 通信密钥管理协议构成了一个完整的面向 MEC 的密钥管理方案。因此，本文在第 10 节中分别给出了三个方案与相关协议的对比。

## 2 安全模型及评价指标

基于文献[13][15] [27][59]，本节将提出零知识可验证多项式委托协议、面向 MEC 服务器私钥管理的二元多项式门限秘密共享协议以及面向 MEC 通信密钥管理协议的安全模型及评价指标，作为第 3、5、7 节提出协议的安全假设，并为第 4、6、8 节中针对协议的安全性分析和证明提供理论支持。本文中所使用的符号及含义如表 1 所示。

**表 1 符号定义**

| 符号 | 含义 |
|---|---|
| $\pi$ | 证据 |
| $A$ | 概率多项式时间的敌手 |
| $\mathcal{A}$ | 概率多项式时间的算法 |
| $MS_i$ | 第 $i$ 个 MEC 服务器 |
| $MS_i'$ | 更新后第 $i$ 个 MEC 服务器 |
| $W_f$ | $f$ 正确计算的证据 |

| | |
|---|---|
| $H(\cdot)$ | 计算哈希值 |
| $\varepsilon$ | 可忽略的值 |
| $Proof(\mu, W_{f(t)})$ | 计算证明算法 |
| $Verify\left(LDT, m_p, MT, GKR\right)$ | 计算验证算法 |
| $RS$ | Reed-solomon 编码 |
| LDT | 低阶测试 |
| HMS | 诚实服务器 |
| $B(x,y)$ | $\langle t, 2t\rangle$阶二元多项式 |
| $Q(x,y)$ | $\langle t, 2t\rangle$阶二元更新多项式 |
| σ | 秘密份额更新次数 |
| $U_i$ | 第 i 个 MEC 中的用户 |
| $Uek_i$ | 第 i 个用户的加密密钥 |
| $Bek_k$ | 第 k 次更新的广播密钥 |
| $Tek_k$ | 第 k 次更新的流量加密密钥 |
| $ID_{U_i}$ | 移动用户 ID |
| $ID_G$ | $MS_i$管理的区域 ID |
| Mcl | 现有成员列表 |
| Mdl | 离开成员列表 |
| $\{m\}K$ | 用对称密钥 K 加密消息 m |
| $a \to b$ | 单点传输 |
| $A \Rightarrow x$ | 广播传输 |

### 2.1 零知识可验证多项式委托

令$\mathbb{F}$是一个有限域，$\mathcal{F}$是$\mathbb{F}$上的$\ell$-变元多项式族，$d$是变元多项式次数. 用$W_{\ell,d}$表示$\mathcal{F}$中的单项式且$N = \left|W_{\ell,d}\right| = (d+1)^{\ell}$。对于$f \in \mathcal{F}$和$t \in \mathbb{F}^{\ell}$，一个零知识可验证多项式委托包含以下算法[69]：

**参数生成阶段：**$pp \leftarrow zkVPD.KeyGen(1^{\lambda})$；

**承诺阶段：**$com \leftarrow zkVPD.Commit(f, r_f, pp)$；

**验证阶段：**$((y,\pi);\{0,1\}) \leftarrow \langle zkVPD.Open(f, r_f), zkVPD.Verify(com)\rangle(t, pp)$。

下面给出零知识可验证多项式委托的定义[69]：

**定义 1：**零知识可验证多项式委托满足以下属性：

**完整性：**对任意多项式$f \in \mathcal{F}$和$t \in \mathbb{F}^{\ell}$，$pp \leftarrow zkVPD.KeyGen(1^{\lambda})$，$com \leftarrow zkVPD.Commit(f, r_f, pp)$，满足：

$$\Pr[\langle zkVPD.Open, zkVPD.Verify\rangle(t, pp) = 1] = 1$$

**可靠性：**对任意概率多项式时间的敌手$A$，$pp \leftarrow zkVPD.KeyGen(1^{\lambda})$，下式的概率是可忽略的：

$$Pr\begin{bmatrix}(f^*, com^*, t) \leftarrow A(1^{\lambda}, pp)\\ ((y^*, \pi^*); 1) \leftarrow \langle A, zkVPD.Verify\rangle(t, pp)\\ com^* = zkVPD.Commit(f^*, pp)\\ f^*(t) \neq y^*\end{bmatrix}$$

**零知识性：**对安全参数$\lambda$，多项式$f \in \mathcal{F}$，$pp \leftarrow zkVPD.KeyGen(1^{\lambda})$，概率多项式时间算法$\mathcal{A}$和模拟器$S = (S_1, S_2)$，考虑以下面两个实验算法$\mathrm{Real}_{\mathcal{A},f}$ (pp)和$i\mathrm{deal}_{\mathcal{A},S^{\mathcal{A}}}$ (pp)：

$\mathbf{Real}_{\mathcal{A},f}$ **(pp)**：

1）$com \leftarrow zkVPD.Commit(f, r_f, pp)$

2）$t \leftarrow \mathcal{A}(\mathrm{com}, \mathrm{pp})$

3）$(y,\pi) \leftarrow \langle \mathrm{zkVPD.\ Open}\left(f, r_f\right), \mathcal{A}\rangle(t, \mathrm{pp})$

4）$b \leftarrow \mathcal{A}(\ \mathrm{com}\ , y, \pi, \mathrm{pp})$

5）输出 $b$

$\boldsymbol{i}\mathbf{deal}_{\mathcal{A},S^{\mathcal{A}}}$ **(pp)**：

1）$com \leftarrow S_1(1^{\lambda}, \mathrm{pp})$

2）$t \leftarrow \mathcal{A}(\mathrm{com}, \mathrm{pp})$

3）$(y,\pi) \leftarrow \langle S_2, \mathcal{A}\rangle(t_i, \mathrm{pp})$随机谕言机访问$y = f(t)$

4）$b \leftarrow \mathcal{A}(\ \mathrm{com}\ , y, \pi, \mathrm{pp})$

5）输出 $b$

对任意概率多项式时间算法$\mathcal{A}$和所有多项式$f \in \mathbb{F}$，存在模拟器 S 使得

$$\left|\Pr\left[\mathrm{Real}_{\mathcal{A},f} = 1\right] - \Pr\left[\mathrm{Ideal}_{\mathcal{A},S^{\mathcal{A}}} = 1\right]\right| \leq \mathrm{neg}l(\lambda)$$

### 2.2 秘密共享协议

在面向 MEC 服务器私钥的二元多项式秘密共享协议中，一个边缘区域的 MEC 服务器作为一个管理委员会分布式的管理每个 MEC 服务器私钥。令 MS 表示 MEC 服务器，$\{MS_i\}_{i=1}^{n}$表示 $n$ 个 MEC 服务器节点，每个节点$MS_i$持有一个秘密份额$s_i$。

**定义 2 :**在 Shamir 的*(t,n)*门限秘密共享中[15]，秘密 $s$ 被保存在一个单变元多项式$f(x)$中，且$f(0) = s$。将$f(x)$分成 $n$ 个秘密份额分发给 $n$ 名不同的成员。在秘密重构时，只需要超过 $t$ 名成员提供秘密份额，就可以通过 Lagrange 插值法恢复秘密多项式。

定义 2 意味着，在一个安全的*(t,n)*门限秘密共享协议中[15]，敌手无法同一时间控制超过 $t$ 个节点。为了应对敌手的泄露攻击，秘密共享协议需要定期的更新秘密份额。此外，秘密共享协议应该允许 MEC 服务器是动态的，即 MEC 服务器可以根据实际需求添加或更换。与主动更新秘密共享协议[26]一致，面向 MEC 服务器私钥的秘密共享协议在固定的时间间隔后执行主动更新协议。在本文中称每个时间间隔为纪元，两个相邻纪元切换的时间点称为转换阶段。转换阶段包含在上一个纪元中，当转换阶段结束时进入新纪元。

在秘密共享协议部分，敌手是强大且积极的。敌手随时可能腐化 MEC 服务器节点，一旦节点被敌手腐化，则会泄露存储的信息并丧失计算能力，直到当前纪元结束。此外，敌手可能会在转换阶段同时腐化旧纪元中的 MEC 服务器和新加入的 MEC 服务器，在*(t,n)*门限秘密共享协议的安全要求下，最多可以控制 $2t$ 个节点。需要在转换

阶段使用维度转换技术[27]暂时提高门限，使门限大于 $2t$，以保证门限秘密共享协议的安全。门限转换技术需要在非对称二元多项式的基础上使用。在基于二元多项式的秘密共享中，成员可以通过身份索引确定二元多项式的一个自变量，将二元多项式转化为单变量多项式，执行*(t,n)*门限秘密共享。一个二元多项式的定义如下：

**定义 3:** 令$a_{i,j} \in GF(p)$, 对$\forall i,j \in [0,t-1]$，一个 *t*-1 阶二元多项式可以表示为[18]：

$$\begin{aligned} F(x,y) = a_{0,0} &+ a_{1,0}x + a_{0,1}y + a_{1,1}xy + a_{2,0}x^2 \\ &+ a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y \\ &+ a_{2,2}x^2y^2 + \cdots \\ &+ a_{t-1,t-1}x^{t-1}y^{t-1} \bmod p, \end{aligned}$$

如果多项式的系数$a_{i,j}$满足$a_{i,j} = a_{j,i}$，其中$i,j \in [0,t-1]$，则称该二元多项式为对称二元多项式，否则称其为非对称二元多项式。

基于上述假设条件，一个面向 MEC 服务器私钥的二元多项式秘密共享协议满足下述定义。

**定义 4:** 对任意的概率多项式敌手$A$，一个面向 MEC 服务器私钥二元多项式秘密共享协议满足[27]

安全性：如果$A$在任何纪元中都无法腐化超过 $t$ 个节点，那么$A$就无法获得 MEC 服务器的私钥。

完整性：如果在两个相邻纪元，$A$无法腐化超过 $t$ 个节点，那么在转换阶段完成后，诚实节点的秘密份额将被正确的计算且私钥保持不变。

### 2.3 面向MEC的通信密钥管理协议

Daghighi 等人[13]在文献[77]的基础上提出了一种动态安全组通信的密钥管理协议设计，本文沿用该文章中的部分假设。具体如下：

- *Uek* 所有者列表是安全的。
- 多广播[78]能提供可靠的密钥传递机制。
- 通信参与设备可以安全的存储加密密钥。
- 在移动用户的角度密钥管理者都是可信的。
- 假设移动成员可以通过 RSA 或 MAC 以合法的身份通过认证，登录到 MEC 网络。
- MEC 服务器可以在网络中心的云服务器不参与的情况下生成用户加密密钥 *Uek*。
- MListen 模型可以发现用户自行离开的行为。

为了刻画上述假设下 MEC 网络中通信密钥管理协议面对的敌手能力，本文对面向 MEC 的通信密钥管理协议的安全模型做出如下定义：

通信模型：在 MEC 网络中，密钥管理层由多个 MEC 服务器(Mobile edge computing Server, MS)组成。移动用户可以使用身份认证令牌登录到 MEC 服务器。多个 MEC 服务器共同管理登录的移动用户，为拥有合法身份认证令牌的用户提供用户加密密钥(Uesr encryption key, Uek)和广播密钥(Broadcast encryption key, Bek)。移动用户在通过初始认证后被允许加入或退出网络。此外，MEC 服务器共同维护 *Uek* 所有者列表，通过记录移动用户的信息，减少用户频繁进出时的计算需求[13]。最后 MEC 生成流量加密密钥(Traffic encryption key, Tek)并用广播密钥加密后共享给所有用户[77]。

敌手能力：敌手 $A$ 通过预言机查询的方式与协议的各个参与方进行交互。通过预言机查询刻画敌手的能力。主要有以下几种：

— Execute($MS_i, U_i$)：敌手 $A$ 在协议正常运行的情况下，通过随机预言机进行查询并输出 MEC 服务器$MS_i$和移动用户$U_i$之间进行传输的数据。

— Send($R, m$)：敌手 $A$ 向通信模型中的任意角色 $R$ 查询 $m$ 并输出角色 $R$ 对 $m$ 的响应。

— Test($R$)：当敌手 $A$ 向任意角色 $R$ 进行查询时，系统随机选取$c \leftarrow \{0,1\}$。当$c = 1$时，角色 $R$ 返回真实值给敌手 $A$，否则角色 $R$ 返回一个与真实值 bit 长度相同的随机字符串给敌手 $A$。

**定义 5：**一个安全的面向 MEC 通信密钥管理协议应该满足以下 4 个属性[77]：

- 机密性：为了防止敌手的窃听攻击[84]，在通信时，移动用户使用密钥加密信息，保证 MEC 网络中移动用户通信和数据具有机密性。
- 完整性：为了防止敌手篡改信息，需要验证数据未被敌手篡改，保证数据具有完整性。
- 前向安全：新加入的移动用户不能通过 MEC 服务器生成的密钥解密之前网络中的消息。
- 后向安全：离开 MEC 网络的移动用户不能通过密钥解密离开之后 MEC 网络的消息。

**定义 6：**设一个函数$f: \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$是一个单向函数，安全参数为$\lambda$，对任意敌手 $A$，存在一个可忽略的函数$negl$，使得：

$$Adv^{OW} = \Pr_{x \leftarrow \{0,1\}^\lambda}\left[A\left(1^\lambda, f(x)\right) \in f^{-1}(f(x))\right] \le negl(\lambda)$$

## 3 非交互透明零知识证明

在秘密共享协议中，可以使用非交互零知识

承诺和证明秘密值，以保证秘密份额的有效性。但传统的非交互零知识证明需要可信初始化生成随机引用字符串。一方面降低了协议运行效率，另一方面可信第三方的存在和无中心的分布式系统也存在矛盾。本节首先回顾透明零知识证明的构造技术，并基于 NIROA 模型[71]构造一种非交互透明零知识可验证多项式委托(zero knowledge Verifiable Polynomial Delegation, zkVPD)，进一步提高秘密共享协议中秘密份额的验证效率。因为非交互透明的 zkVPD 发送的消息为 Merkle 树的认证路径，不会泄露除证据以外的任何消息，因此零知识信道可以使用公开信道。

### 3.1 校验和与单变元校验和

校验和协议是交互式证明中应用最广泛的一种基本协议，它用来计算多项式$f: \mathbb{F}^{\ell} \to \mathbb{F}$在$\sum_{b_1,\cdots,b_\ell \in \{0,1\}} f(b_1,\cdots,b_\ell)$上的和。因为有$2^{\ell}$种组合所以直接计算需要指数级时间。在 Lund 等人[72]提出的校验和协议允许验证者 V 将求和计算委托给一个无限计算能力的证明者 P，且 P 可以提供正确的求和给 V。在校验和协议的最后，V 需要选择随机点$r \in \mathbb{F}^{\ell}$，并访问其在谕言机中$f$的对应值。协议的证明大小和验证时间均为$O(d\ell)$。当$\epsilon = d\ell/|\mathbb{F}|$时，校验和协议满足完备性和可靠性。

单变元校验和[60]允许验证者验证一个单变元多项式在素域$\mathbb{F}$的子集$\mathbb{H}$上和的正确性，即证明者发送$\mu = \sum_{a\in\mathbb{H}} f(a)$给验证者，验证者通过单变元校验和验证$\mu$的正确性。该性质依赖于以下引理：

**引理 1** [73]：令$\mathbb{H}$是素域$\mathbb{F}$上的乘法陪集，$g(x)$是$\mathbb{F}$上次数小于$|\mathbb{H}|$的单变元多项式。则有

$$\sum_{a\in \mathrm{H}} g(a) = g(0)\cdot|\mathbb{H}|$$

由引理 1 可知，若$\sum_{a\in\mathbb{H}} f(a)$的次数小于$k$，则对于$Z_{\mathbb{H}}(x) = \prod_{a\in\mathbb{H}}(x-a)$，存在次数小于$|\mathbb{H}|$的单变元多项式$g(x)$和次数小于$k-|\mathbb{H}|$的单变元多项式$h(x)$，存在唯一分解$f(x) = g(x) + Z_{\mathrm{H}}(x)\cdot h(x)$。因此要验证$\mu$的正确性，可以通过运行 LDT(Low Degree Test)协议[60]和 RS(Reed-Solomon)编码，将问题转化为验证$\mu$的合理约束[69]：

$$p(x) = \frac{|\mathbb{H}|\cdot f(x) - |\mathbb{H}|\cdot Z_{\mathrm{H}}(x)\cdot h(x) - \mu}{|\mathbb{H}|\cdot x}$$

的次数小于$|\mathbb{H}|-1$即可。

### 3.2 GKR协议

令 C 是$\mathbb{F}$上深度为 $D$ 的分层电路，第$i$层的输入是由$i+1$层计算得到的即第 0 层为输出层 D 层为输入层，GKR 协议一层一层执行。证明者接受输出后，在第一轮，证明者和验证者运行校验和协议把输出归约到上一层；在第$i$轮，证明者和验证者通过校验和协议将第$i-1$层的输出归约到第$i$层；最后协议在作为输入的 D 层结束，使得验证者可以直接验证输入的正确性。如果验证通过，则验证者信任输出结果[62]。但是 GKR 协议存在两个问题，一是验证者可以本地计算输入层的值，二是每一层的校验和都泄露的当前层的值[69]。通过零知识可验证多项式委托对输出层进行承诺并隐藏每一层的值[64]，使 GKR 协议具有零知识性. 对每一层$i$，令$Z_i(x) = \prod_{i=1}^{s_i} x_i(1-x_i)$，当$x \in \{0,1\}^{s_i}$时，$Z_i(x) = 0$。证明者选取二元多项式$\mathrm{h}_i(x_1, z)$，有：

$$V_i = g_i(x)\left(x_1,\ldots,x_{s_i}\right) + Z_i\left(x_1,\ldots,x_{s_i}\right)\sum_{z\in\{0,1\}} h_i(x_1,z)$$

因为$h_i$是证明者随机选取的，所以验证者不知道每一层的值。此外证明者需要选取随机多项式$\gamma_i = \sum_{x,y\in\{0,1\}^{\Omega_{i+1}},z\in\{0,1\}} \delta_i(x,y,z)$隐藏校验和协议。验证者在$\mathbb{F}$上随机选取点$\widetilde{a_i}$, $\widetilde{b_i}$, $\widetilde{c_i}, u^{(i)}, v^{(i)}$，计算$Z_i\left(u^{(i)}\right), Z_i\left(v^{(i)}\right)$，每层由如下公式计算[62]：

$$\widetilde{a_i}V_i\left(u^{(i)}\right) + \widetilde{b_i}V_i\left(v^{(i)}\right) + \widetilde{c_i}\gamma_i = \sum_{x,y\in\{0,1\}^{s_{i+1}},z\in\{0,1\}} f_i'\left(V_{i+1}(x), V_{i+1}(y), h_i\left(u_1^{(i)},z\right), h_i\left(v_1^{(i)},z\right), \delta_i(x,y,z)\right) \quad (1)$$

这样运行 GKR 协议，验证者就可以直接验证输入结果，若协议通过，则验证者信任输出结果。

### 3.3 零知识可验证多项式委托

在 Shamir 的秘密共享协议中，秘密被嵌入到一个单变元多项式$f(x)$的常数项中，且每个参与者$P_i$的秘密份额等于多项式$f(T_i)$的函数值。因此可验证多项式委托使秘密共享实现恶意安全。

为了避免使用双线性对和模幂运算等繁重的计算工具，可以使用 Merkle 树对向量作承诺[69]。

Merkle 树[74]是一种基于哈希函数的抗碰撞性质构造的密码学原语，其包含以下算法：

- $root_c \leftarrow MT.Commit(c)$
- $(c_{idx}, \pi_{idx}) \leftarrow MT.Open(idx, c)$
- $(1,0) \leftarrow MT.Verify(root_c, idx, c_{idx}, \pi_{idx})$

基于 Merkle 树的良好性质，一个高效的交互式零知识可验证多项式委托协议如下[69]：

令$\mathbb{F}$是一个有限域，$\mathcal{F}$是$\mathbb{F}$上的$\ell$-变元多项式

族，$d$是变元多项式次数。用$W_{\ell,d}$表示$\mathcal{F}$中所有的单项式且$N=\left|W_{\ell,d}\right|=(d+1)^{\ell}$。对于$f\in\mathcal{F}$和$t\in\mathbb{F}^{\ell}$，一个 zkVPD 包含以下三个阶段算法[69]：

**参数生成阶段：**$pp\leftarrow zkVPD.KeyGen(1^{\lambda})$；

**承诺阶段：**$com\leftarrow zkVPD.Commit(f,r_f,pp)$，选取随机多项式，用来加密单变量多项式。使用 Merkle 树承诺加密多项式和随机多项式；

**验证阶段：**$((y,\pi);\{0,1\})\leftarrow\langle zkVPD.Open(f,r_f),zkVPD.Verify(com)\rangle(t,pp)$，通过 LDT 协议[60]和 GKR 协议[62]验证多项式值的正确性。

该算法在参数生成阶段不使用门陷；此外在承诺阶段通过使用 Merkle 树，从而避免使用双线性对运算，极大的提升了协议的运行效率；在承诺阶段使用加密多项式并在验证阶段隐藏了多项式的次数，保证了协议的零知识性[65]. 但是该协议是交互式协议，不适用于秘密共享中秘密份额的验证，下面我们将基于交互式透明零知识证明协议[69]和 NIROA 模型[71]构造一个非交互透明零知识可验证多项式委托。

### 3.4 非交互透明零知识可验证多项式委托

基于文献[69]的零知识证明协议和文献[71]的 NIROA 模型，一个非交互透明 zkVPD 构造如下：

设$\lambda$是安全参数，$\mathbb{F}$是素域。$C:\mathbb{F}^n\to\mathbb{F}$是$\mathbb{F}$上的 $t$ 阶分层算术电路。$H_1$和$H_2$是抗碰撞的哈希函数。令$\mathbb{F}$是一个有限域，$\mathcal{F}$是$\mathbb{F}$上的二元多项式族，$d$是变元多项式次数。用$W_{2,d}$表示$\mathcal{F}$中所有的单项式且$N=\left|W_{2,d}\right|=(d+1)^2$。对于$f\in\mathcal{F}$和$t\in\mathbb{F}^2$，对一个$d$次二元多项式$f(x,y)$，定义向量$T=(W_1(t),\cdots,W_N(t))$，有$f(t)=\sum_{i=1}^{N}c_iW_i(t)$；

运行参数生成算法：

$pp\leftarrow zkVPD.KeyGen(1^{\lambda})$，证明者在$\mathbb{F}$中选择一个乘法陪集$\mathbb{H}$使得$|\mathbb{H}|=N$；在$\mathbb{F}$中选择一个乘法陪集$\mathbb{L}$使得$|\mathbb{L}|=O(|\mathbb{H}|)>2|\mathbb{H}|$且$\mathbb{H}\subset\mathbb{L}\subset\mathbb{F}$。定义$\mathbb{U}=\mathbb{L}-\mathbb{H}$；

运行承诺算法：

$com\leftarrow zkVPD.Commit(f,r_f,pp)$，对多项式$f(x,y)=\sum_{i=1}^{N}c_iW_i(x)$，通过向量$\boldsymbol{c}$和$\boldsymbol{T}$的插值找到$\mathbb{H}$上的唯一的单变量多项式$l(x)$：$\mathbb{F}\to\mathbb{F}$，使得$l|_{\mathbb{H}}=(c_1,c_2,\cdots,c_N)$；证明者随机选取$k$阶多项式$r(x)$和$2|\mathbb{H}|+k-1$阶多项式 $s(x)$，定义$Z_{\mathbb{H}}(x)=\prod_{a\in\mathbb{H}}(x-a)$，令$l'(x)=l(x)+Z_{\mathbb{H}}(x)\cdot r(x)$。计算$l'|_{\mathbb{L}}$并运行 Merkle 树算法，得到$root_{l'}\leftarrow MT.Commit(l'|_{\mathbb{L}})$和$root_s\leftarrow MT.Commit(s|_{\mathbb{U}})$；

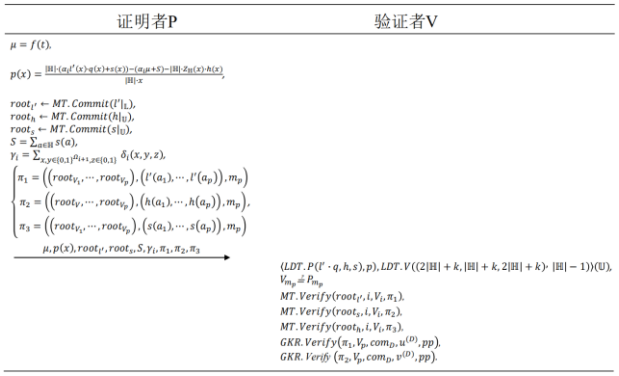


图 3. 非交互零知识可验证多项式委托

运行验证算法：

$((y,\pi);\{0,1\})\leftarrow\langle zkVPD.Open(f,r_f),zkVPD.Verify(com)\rangle(t,pp)$，证明者构造证明。计算多项式的值$\mu=f(t)$；计算$T=\left(W_1(t),\cdots,W_N(t)\right)$，通过快速傅里叶变换[75]得到唯一的单变元多项式$q(x):\mathbb{F}\to\mathbb{F}$，使得$q|_{\mathbb{H}}=T$；计算$S=\sum_{a\in\mathbb{H}}s(a)$；计算合理约束

$$p(x)=\frac{|\mathbb{H}|\cdot(\alpha_i l'(x)\cdot q(x)+s(x))-(\alpha_i\mu+S)-|\mathbb{H}|\cdot Z_{\mathbb{H}}(x)\cdot h(x)}{|\mathbb{H}|\cdot x}.$$

随机选取$\gamma_i=\sum_{x,y\in\{0,1\}^{\Omega_{i+1}},z\in\{0,1\}}\delta_i(x,y,z)$。证明者计算$m_0=H_2(PP,)$，$i\in[1,p]$执行以下操作：

1) 计算验证者随机值$\alpha_i=H_1(m_{i-1},root_{l'})$；

通过$\alpha_i$计算唯一分解$\mathrm{V}_i(x)=\alpha_i l'(x)\cdot q(x)+\mathrm{s}(x)=g_i(x)+Z_{\mathbb{H}}(x)\cdot h(x)$，其中$g_i(x)$的次数小于$|\mathbb{H}|$，$h_i(x)$的次数小于$|\mathbb{H}|+k$；计算$h|_{\mathbb{U}}$和$root_h\leftarrow MT.Commit(h|_{\mathbb{U}})$；

2) 运行 Merkle 树算法，得到

$$root_{c_i}\leftarrow MT.Commit(c_i);$$

3) 计算$m_i=H_2\left(root_{V_i},m_{i-1}\right)$。$\beta=H_1\left(m_p\right)$，有$\mathrm{V}=\left\{\alpha_1,\cdots,\alpha_p,\beta\right\}$。证明者访问随机谕言机，计算证明路径$l'(a_i)=MT.Open(l'|_{\mathbb{L}},i)$，$h(a_i)=MT.Open(h|_{\mathbb{U}},i)$，$s(a_i)=MT.Open(s|_{\mathbb{U}},i)$。令

$$\begin{cases}\pi_1=\left(\left(root_{V_1},\cdots,root_{V_p}\right),\left(l'(a_1),\cdots,l'(a_p)\right),m_p\right)\\ \pi_2=\left(\left(root_{V},\cdots,root_{V_p}\right),\left(h(a_1),\cdots,h(a_p)\right),m_p\right)\\ \pi_3=\left(\left(root_{V_1},\cdots,root_{V_p}\right),\left(s(a_1),\cdots,s(a_p)\right),m_p\right)\end{cases}$$

将$\mu,p(x),root_{l'},root_h,root_s,S,\gamma_i,\pi_1,\pi_2,\pi_3$发送给验证者。此时验证者获得了验证将用到的全部信息。验证者在合理约束$p(x)$上运行 LDT 协议$\langle LDT.P((l'\cdot q,h,s),p),LDT.V((2|\mathbb{H}|+k,|\mathbb{H}|+k,2|\mathbb{H}|+k),|\mathbb{H}|-1)\rangle(\mathbb{U})$，若运行失败则返回$0$，否则返回$1$。在 LDT 协议的最后，验证者计算

$m_0 = H_2(PP)$， $i \in [1,p]$执行以下操作：

1)计算验证者随机值$\alpha_i = H_1(m_{i-1}, root_{l'})$;

2)计算$m_i = H_2(root_{V_i}, m_{i-1})$

若验证者得到的$m_p$与证明者发送的$m_p$不等，则返回 0，否则返回 1。然后验证者运行

$$MT.Verify(root_{l'}, i, V_i, \pi_1),$$
$$MT.Verify(root_s, i, V_i, \pi_2),$$
$$MT.Verify(root_h, i, V_i, \pi_3),$$

若运行失败则返回 0，否则返回 1。

在 LDT 协议结束后，验证者运行 GKR 协议，在$\mathbb{F}$中随机选择$\widetilde{a_i}, \widetilde{b_i}, \widetilde{c_i}$, 根据式(1)可得$u^{(D)}$和$v^{(D)}$，运行$GKR.Verify\left(\pi_1, V_p, \mathrm{com}_D, u^{(D)}, pp\right)$和$GKR.Verify\left(\pi_2, V_p, \mathrm{com}_D, v^{(D)}, pp\right)$。若失败返回 0，否则 GKR 验证协议运行成功，返回 1。

记证明生成算法为$Proof(\mu, W_{f(t)})$；验证算法为$Verify\left(LDT, m_p, MT, GKR\right)$。

综上所述，一个非交互透明零知识可验证多项式委托协议流程如图 3 所示。

## 4 透明零知识证明协议分析

本节针对上一小节中提出的非交互透明零知识证明协议进行安全性分析。

**定理 1**：验证协议是一个具有完整性、可靠性和零知识性的非交互式可验证多项式委托协议。

**证明：完整性**：与文献[69]中协议 4 的完整性证明类似。由本文引理 1 可知，对于定义的$l'(x)$和$q(x)$，如果计算值$\mu = f(t)$，则有$\mu = \sum_{a\in H} l'(a) \cdot q(a) = \sum_{a\in H} (l(a)\cdot q(a) + Z_{\mathbb{H}}(a)\cdot r(a)\cdot q(a)) = \sum_{a\in H} g(a) = g(0)\cdot|\mathbb{H}|$，因此其合理约束$p(x) = (|\mathbb{H}|\cdot(\alpha_i l'(x)\cdot q(x) + \mathrm{s}(x)) - (\alpha_i \mu + S) - |\mathbb{H}|\cdot Z_{\mathbb{H}}(x)\cdot h(x))/|\mathbb{H}|\cdot x = (g(x)-g(0))/x$的次数为$|\mathbb{H}|-1$。满足 LDT 协议和 GKR 协议完整性要求。

**可靠性**：令$\varepsilon_{\mathrm{LDT}}, \varepsilon_{\mathrm{MT}}, \varepsilon_{\mathrm{GKR}}, \varepsilon_{\mathrm{NIROA}}$分别表示 LDT 协议、Merkle 树、GKR 协议和 NIROA 的完整性误差，因为交互式的可验证多项式委托具有可靠性[59]。基于 Virgo[69]，令$\lambda$表示安全参数，令$k$表示 LDT 协议中检查点的个数，$m$表示一轮证明的查询次数，考虑以下情况的恶意证明者：

① $\nexists l^* \in RS[\mathbb{L}, |\mathbb{H}|+1]$ 使得 com = MT.Commit $(l^*|\mathbb{L})$，即$com$不是一个有效的承诺。在验证者调用 Merkle 树验证算法时，$com$不是 Merkle 树的根，验证者通过验证的概率小于$\varepsilon_{\mathrm{MT}}$。如果$\exists l^{**} \notin RS[\mathbb{L}, |\mathbb{H}|+1]$使得 com = $MT.Commit\,(l^{**}|\mathbb{L})$，如果证明者运行 Merkle 树打开承诺的点$v_i^*$和计算的证明路径$l^{**}(a_i)$都不相等，即$v_i^* \neq l^{**}(a_i)$，则验证者通过验证的概率小于$\varepsilon_{\mathrm{MT}}$。如果唯一分解$q_i^* \neq q(a_i)$，即没有通过 GKR 协议，则验证者通过验证的概率小于$\varepsilon_{\mathrm{GKR}}$。如果$l^{**}(x)\cdot q(x) \notin RS[\mathbb{L}, 2|\mathbb{H}|+1]$，则验证者通过验证的概率小于$\varepsilon_{\mathrm{LDT}}$。如果验证者在 LDT 协议后计算$m_p^*$与证明者发送的$m_p$不相等，则验证者通过验证的概率小于$\varepsilon_{\mathrm{NIROA}}$。

② $\exists l^* \in RS[\mathbb{L}, |\mathbb{H}|+1]$ 使得 $com = MT.Commit\,(l^*|\mathbb{L})$。令$c^* = l^*|_{\mathbb{H}}$，$f^*(x) = \sum_{i=1}^{N} c_i^* W_i(x)$，则有$com = Commit\,(f^*, pp)$。如果$\mu^* \neq f^*(t)$，则$\mu^* \neq \sum_{a\in\mathbb{H}} l^*(a)q(a)$。由引理 1 可知，对所有的$h \in RS[\mathbb{L}, |\mathbb{H}|+1], p^* \notin RS[\mathbb{L}, |\mathbb{H}|-1]$，有：

$$\sum_{a\in\mathbb{H}} (p^*(a)\cdot a)$$
$$= \sum_{a\in\mathbb{H}} (|\mathbb{H}|\cdot l^*(a)\cdot q(a) - \mu^*)/|\mathbb{H}|$$
$$= \sum_{a\in\mathbb{H}} (l^*(a)\cdot q(a)) - \mu^* \neq 0。$$

因此如果证明者计算的$h$和$l^*$与 Merkle 树打开的承诺值不一致时，验证者通过验证的概率小于$\varepsilon_{\mathrm{MT}}$。如果计算的唯一分解$q_i^* \neq q(a_i)$，即没有通过 GKR 协议，则验证者通过验证的概率小于$\varepsilon_{\mathrm{GKR}}$。如果$l^*\cdot q \in RS[\mathbb{L}, 2|\mathbb{H}|+1]$，不论$h \notin RS[\mathbb{L}, |\mathbb{H}|+1]$还是$p \notin RS[\mathbb{L}, |\mathbb{H}|-1]$，则验证者通过验证的概率小于$\varepsilon_{\mathrm{LDT}}$。如果验证者在 LDT 协议后计算得到的$m_p^*$与证明者发送的$m_p$不相等，则验证者通过验证的概率小于$\varepsilon_{\mathrm{NIROA}}$。

在上述情况的约束下，一个证明者运行非交互式可验证多项式委托协议的错误概率为$O(\varepsilon_{\mathrm{LDT}} + \varepsilon_{\mathrm{MT}} + \varepsilon_{\mathrm{GKR}} + \varepsilon_{\mathrm{NIROA}})$，又因为[54,60,69,73]

$$\varepsilon_{\mathrm{LDT}} = O(\mathrm{L}/\mathbb{P}) + \mathrm{negl}(k), \varepsilon_{GKR} = O(\log^2 N/|\mathrm{F}|),$$
$$\varepsilon_{\mathrm{MT}} = \mathrm{negl}(\lambda), \varepsilon_{\mathrm{NIROA}} = \mathrm{negl}(\lambda) + 3(m^2+1)2^{-\lambda}$$

可得可靠性误差为$\mathrm{negl}(\lambda) + 3(m^2+1)2^{-\lambda}$，当$\lambda$足够大时，可靠性误差可忽略，协议具有可靠性。

**零知识性**：设安全参数为$\lambda$，多项式$f \in \mathcal{F}$，$pp \leftarrow zkVPD.KeyGen(1^\lambda)$，$\mathcal{A}$是概率多项式时间算法，构造模拟器$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$，令$\mathcal{S}_1$中的$l'_{\mathrm{sim}}$和 zkVPD.Commit 中的$l'$是均匀分布。记真实协议为$\mathrm{Real}_{\mathcal{A},f}$ (pp)，模拟协议为$\mathrm{Ideal}_{\mathcal{A},\mathcal{S}}$ (pp)。有：

$\boldsymbol{com \leftarrow \mathcal{S}_1(1^\lambda, \mathrm{pp})}$：选择一个随机多项式$l'_{\mathrm{sim}}(x) \in RS[\mathbb{L}, |\mathbb{H}|+\kappa+1]$。计算$l'_{\mathrm{sim}}|_{\mathbb{u}}$，输出

$$\mathrm{root}_{l'_{\mathrm{sim}}} \leftarrow MT.Commit\,(l'_{\mathrm{sim}}|_{\mathbb{U}}).$$

$\boldsymbol{\mathcal{S}_2(t, \mathrm{pp})}$：谕言机访问$\mu = f(t)$。计算$T = (W_1(t), \cdots, W_N(t))$，通过快速傅里叶变换得到唯一

的单变元多项式$q(x):\mathbb{F}\to\mathbb{F}$，使得$q|_{\mathbb{H}}=T$。

随机选择$2|\mathbb{H}|+\kappa-1$次多项式$s_{sim}(x)$；计算$S_{sim}=\sum_{a\in\mathbb{H}}s(a)$，输出$\mathrm{root}_{S_{sim}}\leftarrow MT.Commit(S_{sim}|_{\mathbb{U}})$。计算$m_0=H_2(PP,)$，$i\in[1,p]$；计算随机值$\alpha_i=H_1(m_{i-1},root_{l'})$。令$LDT.\mathcal{S}$是 LDT 协议模拟器，$LDT.\mathcal{S}$对每个$\alpha_i$生成$p^*(x)\in RS[\mathbb{L},|\mathbb{H}|-1]$并计算$h_i$使得

$$p^*(a_i)=\frac{|\mathrm{H}|\cdot\left(\alpha l'_{sim}(a_i)\cdot q(a_i)+s_{sin}(a_i)\right)-(\alpha\mu+S_{sim})-|H|\cdot\mathbb{Z}_H(a_i)h_i}{|H|\cdot a_i}$$

插入$h_i$可得多项式$h_{sim}$，输出

$$root_{h_{sim}}\leftarrow MT.Commit\ (h_{sim}|_{\mathbb{U}});$$

运行 Merkle 树算法，得到

$$root_{c_i}\leftarrow MT.Commit(c_i);$$

证明者访问随机谕言机，计算证明路径

$$\begin{cases}\left(l'_{sim}(a_i),\pi_i^{l'_{im}}\right)\leftarrow MT.Open\ (i,l'_{sim}|_U)\\ \left(h_i,\pi_i^{h_{sim}}\right)\leftarrow MT.Open(i,h_{sim}|_U)\\ \left(s_{sim}(a_i),\pi_i^{s_{sim}}\right)\leftarrow MT.Open(i,s_{sim}|_U)\end{cases}$$

发送$\{\mu,p^*(a_i),root_{l'_{sim}},root_{h_{sim}},root_{S_{sim}},S_{sim},\pi_i^{l'_{im}},\pi_i^{h_{sim}},\pi_i^{s_{sim}}\}$给模拟验证者$V^*$。此时模拟验证者$V^*$获得了验证将用到的全部信息等待$V^*$验证。

在模拟器中$\mu$，$q|_{\mathbb{H}}$与真实协议没有区别，随机值$\alpha_i$的计算不包含有用的信息，所以$p^*(a_i)$和真实协议中的合理约束$p(x)$无法区分。$S_{sim}$和 $s$ 是随机选取的且对其进行的承诺和计算是无法区分的。由于真实协议中的$r(x)$是 $k$ 次随机多项式，所以计算$l'(x)$的证明路径时产生的 $k$ 独立随机分布的，与模拟器中计算证明路径无法区分。同理，模拟验证者$V^*$接收到$h_i$和$s_{\mathrm{sim}}$的证明路径同样与真实协议无法区分。因为模拟验证者$V^*$无法区分真实信息和模拟器信息，所以模拟验证过程也与真实协议无法区分。综上所述，模拟协议$\mathrm{Ideal}_{\mathcal{A},\mathcal{S}}(\mathrm{pp})$和真实协议$\mathrm{Real}_{\mathcal{A},f}(\mathrm{pp})$满足：

$$\left|\Pr\left[\mathrm{Real}_{\mathcal{A},f}(\mathrm{pp})=1\right]-\Pr\left[\mathrm{Ideal}_{\mathcal{A},\mathcal{S}}(\mathrm{pp})=1\right]\right|\le\mathrm{negl}(\lambda).$$

# 5 MEC 服务器私钥管理协议

本节将针对 MEC 网络环境下，基于二元多项式门限秘密共享协议[27]和透明 zkVPD 提出一种面向动态 MEC 服务器的私钥管理协议。设 MEC 网络中有 $n$ 个 MEC 服务器作为私钥管理的节点。记 MEC 服务器为$MS$，$\{MS_i\}_{i=1}^n$表示 MEC 网络中的第 $i$ 个节点。由 2.2 的分析，本文基于$\langle t,2t\rangle$阶非对称二元多项式构造可验证动态秘密共享协议。

## 1.1 私钥生成协议

基于二元多项式[83]和 zkVPD 构造私钥生成算法。令$i,j\in\{1,\cdots,n\}$，具体协议如下：

a) 每个$MS_i$选取一个$\langle t,2t\rangle$阶多项式$f_i(x,y)$，令$S_i=f_i(0,0)=a_{0,0}^{(i)}$。$MS_i$计算$s_{ij}=f(i,j)$并运行证明生成算法$Proof(\mu,W_{f(t)})$计算$W_{f(i,j)}$。发送$s_{ij}$和$W_{f(i,j)}$给$MS_j$。

b) 每个$MS_j$收到$MS_i$发送的$s_{ij}$和$W_{f(i,j)}$后，运行$Verify(LDT,m_p,MT,GKR)$。如果对i的索引验证失败，则$MS_j$广播投诉$MS_i$。

c) 当$MS_i$收到来自$MS_j$的投诉时，需要重新计算$s_{ij}$和$W_{f(i,j)}$并发送给$MS_j$。

d) 当$MS_i$在 b 阶段收到超过 $t$ 个投诉或使用伪造的值应答 c 阶段的投诉，则 MEC 服务器标记其为故障，在纪元更新时进行替换。

在上述协议结束后，未被标记为故障的服务器被标记为诚实服务器$HMS$。$HMS_i$计算其私钥$sk_i=\sum_{j\in HMS}s_{ji}$。此时，MEC 服务器的主私钥$MSK=\sum_{i\in HMS}S_i$且主私钥没有被直接计算得出。可以通过门限签名技术[82]对 MEC 网络用户进行认证。本文的研究重点不在于此，因此不再进行展开。

## 1.2 *(t,n)*门限二元多项式秘密共享协议

拥有私钥的 MEC 服务器在纪元更新后，使用*(t,n)*门限二元多项式秘密共享协议管理私钥。

1. 生成秘密份额

为了防止单点故障导致私钥泄露，将私钥通过秘密共享的方式分享给 MEC 中的其他服务器分布式管理，提高私钥的安全性。具体流程如下：

a) $MS_i$选取$\langle t,2t\rangle$阶二元多项式$B(x,y)=\sum_{i=0}^{t}\sum_{j=0}^{2t}a_{ij}x^iz^j$且$B(0,0)=sk_i$

b) MEC 服务器计算秘密份额$B(j,y)$并运行$Proof(\mu,W_{f(t)})$计算证据$W_{B(j,y)}$。

c) 发送$B(j,y)$和$W_{B(j,y)}$给服务器$MS_j$。

d) 服务器$MS_j$收到$B(j,y)$和$W_{B(j,y)}$后运行$Verify\left(LDT,m_p,MT,GKR\right)$。如果验证失败，则$MS_j$要求$MS_i$重新计算并发送。

e) 如果验证通过，则服务器保存$B(j,y)$作为$MS_j$私钥共享的秘密份额。

2. 秘密重构

当服务器$MS_i$需要使用私钥时，需要至少$t+1$个服务器提供其秘密份额，$MS_i$使用 Lagrange 插值

法重构二元多项式$B(x,y)$。具体流程如下：

a) $MS_i$向 MEC 服务器$MS_j$发出重构私钥请求，$MS_j$收到请求后响应，并使用秘密份额$B(j,y)$计算 $B(j,0)$，并运行证明生成算法$Proof(\mu,W_{f(t)})$计算证据$W_{B(j,0)}$。

b) $MS_j$发送$B(j,0)$和$W_{B(j,0)}$给$MS_i$。

c) 当$MS_i$收到 $B(j,0)$和$W_{B(j,0)}$后运行算法验证算法$Verify\left(\mathrm{LDT},\mathrm{m_p},\mathrm{MT},\mathrm{GKR}\right)$。如果验证失败，要求$MS_i$重新计算并发送。

d) 当$MS_i$接收并验证成功至少$t+1$个值时，通过 Lagrange 插值法计算$B(x,0)$。计算$sk_i=B(0,0)$，$MS_i$完成私钥重构。

3. 秘密份额更新

当一个纪元结束时，需要 MEC 服务器对秘密份额进行更新，提高秘密共享协议的安全性。基于 zkVPD 和文献[26]的思想构造主动更新协议。在纪元周期结束时，需要至少$2t+1$个$MS_i$响应生成更新多项式$Q(i,y)$。具体流程如下：

a) 在纪元结束时进入转换阶段，至少$2t+1$个$MS_i$随机生成$\langle t,2t\rangle$阶多项式$f(x,y)$使得$f(0,0)=0$。计算$f(i,j)$和$W_{f(i,j)}$。

b) 发送$f(i,j)$和$W_{f(i,j)}$给 MEC 服务器$MS_j$。

c) 当服务器$MS_j$接收到$f(i,j)$和$W_{f(i,j)}$时运行$Verify\left(LDT,m_p,MT,GKR\right)$。如果验证失败，要求$MS_i$重新计算并发送。

d) 当$MS_j$接收并验证成功至少$2t+1$个值时，计算$t$阶多项式$f'(x,j)=\sum_{i=1}^{2t+1}f(i,j)+f(x,0)$。即$Q(0,j)=\sum_{i=1}^{2t+1}f(i,j)$。

e) 计算$f'(i,j)$和$W_{f'(i,j)}$，并发送给$MS_i$。

f) 当服务器$MS_i$接收到$f'(i,j)$和$W_{f'(i,j)}$时运行$Verify\left(LDT,m_p,MT,GKR\right)$。如果验证失败，要求$MS_j$重新计算并发送。

g) 当$MS_i$接收并验证成功至少$2t+1$个值时，重构$2t$阶多项式$Q(i,y)$。

h) $MS_i$更新$B'(j,y)=B(j,y)+Q(i,y)$。

4. MEC 服务器节点更新

在纪元即将结束进入转换阶段时，可能有一些服务器节点已经停止提供服务或退出 MEC 网络，此时需要新增加或更换 MEC 服务器，记更新的服务器为$MS_i'$。一方面 MEC 网络需要通过密钥生成协议为新服务器生成私钥，并提供分布式存储，保障新服务器私钥的安全。另一方面，要为更新后的服务器节点分发秘密份额。此外，需要考虑转换阶段敌手可能在门限安全假设的条件下，控制新增服务器和原网络服务器节点以突破门限，从而获取用户私钥信息。基于 zkVPD 和文献[27]提出的维度转换技术构造 MEC 服务器的节点更新协议。具体流程如下：

a) $2t+1$个$MS_i$响应阶段转换，计算 $B(i,j)$和$W_{B(i,j)}$并发送给$MS_j'$，$MS_j'$验证通过后计算私钥$sk_i=\sum_{j\in HMS}s_{ji}$。并通过其中$t$个值使用 Lagrange 插值法计算$B(x,j)$。

b) MEC 网络中$2t+1$个服务器运行秘密份额更新协议，因为此时秘密份额为$B(x,j)$，重构$t$阶更新多项式$Q(x,j)$。

c) $MS_j'$计算 $B'(x,j)=B(x,j)+Q(x,j)$。

d) 在转换阶段结束后，进入新的纪元$MS_j$计算$B'(i,j)$和证据$W_{B'(i,j)}$，发送给$MS_i$。

e) 在$MS_i$收到并过验证$2t+1$个节点后，计算$B(j,y)$作为$MS_j$私钥共享的秘密份额。

此时，若纪元中敌手 $A$ 腐化的 MEC 服务器节点不超过 $t$ 个，且在转换阶段腐化了不超过 $t$ 个新加入网络的 MEC 服务器节点，则敌手在转换阶段无法获取未被腐化的 MEC 服务器私钥。这个结论的证明将在节 6.2 安全性分析中给出。

## 6. 私钥管理协议分析

### 6.1 正确性分析

基于二元多项式构建的*(t,n)*门限共享协议的正确性证明与传统的协议类似[15]，本节将分析秘密份额更新协议的正确性。由第四节的分析可知，验证协议可以验证计算的正确性。

**定理 2：**运行秘密份额更新协议后，服务器仍然可以通过更新后的秘密份额恢复私钥。

**证明：**在第σ此运行更新协议后，新秘密份额$B^{\sigma}(j,y)=B^{\sigma-1}(j,y)+Q^{\sigma}(i,y)$。当有超过 $t$ 个人提供其秘密份额时，重构的$\langle t,2t\rangle$阶二元多项式为：

$$B^{\sigma}(x,y)=\sum_{j=1}^{t}B^{\sigma}(j,y)\prod_{i=1,j\neq i}^{t}(x-i)/(j-i)$$
$$=\sum_{j=1}^{t}(B^{\sigma}(j,y)+Q^{\sigma}(i,y))\prod_{i=1,j\neq i}^{t}(x-i)/(j-i)$$
$$=\sum_{j=1}^{t}(B(j,y)+Q^{1}(i,y)+\cdots+Q^{\sigma}(i,y))\prod_{i=1,j\neq i}^{t}(x-i)/(j-i)$$
$$=B(x,y)+Q^{1}(x,y)+\cdots+Q^{\sigma}(x,y),$$

因为更新多项式满足:

$$Q^{\sigma}(x,y)=\sum_{j=1}^{t}Q^{\sigma}(i,y)\prod_{i=1,j\neq i}^{t}(x-i)/(j-i)$$
$$=\sum_{j=1}^{t}f'(x,j)\prod_{i=1,j\neq i}^{t}(y-j)/(i-j)$$

$\prod_{i=1,j\neq i}^{t}(x-i)/(j-i)$
$=\sum_{j=1}^{t}(\sum_{i=1}^{2t+1}f(i,j)+f(x,0))$
$\prod_{i=1,j\neq i}^{t}(x-i)/(j-i)\prod_{i=1,j\neq i}^{t}(y-j)/(i-j)$
$\prod_{i=1,j\neq i}^{t}(x-i)/(j-i)$，

因为$f(0,0)=0$，所以当$(x,y)=(0,0)$时，有$Q^{\sigma}(x,y)=0$，$\sigma=1,2,\cdots,\mathrm{M}$。重构私钥的服务器$MS_j$计算私钥，有$B^{\sigma}(0,0)=B(x,y)+Q^{1}(x,y)+\cdots+Q^{\sigma}(x,y)=B(x,y)+0+\cdots+0=sk_j$。

### 6.2 安全性分析

本协议在信息发送时，采用安全信道，以避免外部敌手。因为使用非交互零知识可验证多项式委托协议具有完备性、可靠性和零知识性，可以避免出现内部成员欺骗的情况。容易看出私钥生成和$(t,n)$秘密共享协议满足定义 2 的条件，具有安全性和完整性满足定义 4 的安全。本节首先讨论在转换阶段运行的秘密份额更新协议和服务器节点更新协议的安全性和完整性，并给出证明。

首先分析 PPT 敌手 $A$。在纪元更新时，进入转换阶段，敌手 $A$ 除了公共参数和服务器身份索引外可能获得可以获取的信息包括：

- 在上一纪元的转换阶段，被腐化服务器的秘密份额$B(x,j)$和在本纪元转换阶段之前，被腐化服务器的秘密份额$B(j,y)$，以及被腐化服务器发送的秘密份额$f(i,j)$和$W_{f(i,j)}$。
- 在当前纪元的转换阶段，新加入的服务器接收和发送的$B(i,j)$、$W_{B(i,j)}$和$B(x,j)$
- 在当前纪元的转换阶段，新增服务器运行更新秘密份额计算的$f(i,j)$、$W_{f(i,j)}$和$Q(x,j)$
- 在转换阶段结束后，进入新纪元，被腐化服务器计算和接收的$B'(i,j)$和$W_{B'(i,j)}$。

因为使用了透明 zkVPD 协议，所有计算的多项式和随机多项式均由 Merkle 树承诺，仅有根节点和对应路径上的哈希值被公开，敌手 $A$ 无法获得 Merkle 树承诺中获得额外信。记被敌手腐化的服务器为 $\mathcal{C}MS_i$。则对敌手 $A$，有以下引理。

**引理 2：**如果在纪元中敌手 $A$ 腐化的 MEC 服务器节点不超过 $t$ 个，且在转换阶段腐化了不超过 $t$ 个新加入网络的 MEC 服务器节点，则敌手在转换阶段无法获取未被腐化的 MEC 服务器私钥$sk$。

证明：在最坏情况下，敌手 $A$ 会腐化 $t$ 个原 MEC 服务器节点和 $t$ 个新加入网络的 MEC 服务器节点，由上面敌手可获取信息的分析可知，此时敌手获取的秘密份额包括原 MEC 服务器节点的秘密份额$B(j,y)$和密钥转换阶段 $2t$ 个 MEC 服务器节点的秘密份额$B(x,j)$。由定义 2 可知，在$\langle t,n\rangle$门限秘密共享中，敌手不能通过$B(j,y)$获取服务器$MS_j$的私钥$sk_j$。在 5.4(b)中，秘密多项式转换为 $2t$ 阶，假设被腐化的节点的索引为$i\in[1,2t]$，敌手计算$B(x,y)=\sum_{j=1}^{2t}\mathrm{B}(x,j)\prod_{i=1,j\neq i}^{2t}(x-i)/(j-i)$，可得$\langle t,2t-1\rangle$阶多项式$B(x,y)$。此外，在为新加入网络的 MEC 服务器$MS_j'$，$j\in[p,q]$生成私钥的阶段，传输份额数据可以用如下矩阵表示：

$$\begin{pmatrix} B(1,p) & \cdots & B(1,t') & \cdots & B(1,q) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ B(2t+1,p) & \cdots & B(2t+1,t') & \cdots & B(2t+1,q) \end{pmatrix}$$

其中矩阵中行表示服务器$MS_i$发送的份额，列表示新增服务器$MS_j'$收到的份额。由信息传输矩阵可以看出，敌手可以获取矩阵中的$k$行，$k\leq t$，计算新增服务器$MS_j'$的私钥$sk_j=\sum_{k\in A}B(i,j)\neq\sum_{i\in 2t+1}B(i,j)$。故即使在最坏情况下，敌手也无法在转换阶段获取未被腐化 MEC 服务器的私钥。

**定理 3：**对任意的概率多项式敌手$A$，秘密份额更新协议具有安全性和完整性。

证明：完整性：对任意一个$2t+1$阶 $t$ 变元多项式$Q(x,j)$对应了唯一的$\langle t,2t\rangle$阶二元多项式。又因为$Q(0,0)=0$。由定理 2 可知，对于随机生成的$Q(x,y)$可以隐藏$\langle t,2t\rangle$阶二元多项式$B(x,y)$且不改变秘密 s。即更新协议具有完整性。

与引理 2 的情况相同，在转换阶段敌手最多可以获取 $2t$ 个$2t+1$阶 $t$ 变元多项式$Q(x,j)$。计算$Q'(x,y)=\sum_{j=1}^{2t}\mathrm{Q}(x,j)\prod_{i=1,j\neq i}^{2t}(x-i)/(j-i)$可得$\langle t,2t-1\rangle$阶多项式$Q'(x,y)$。此时无法使用$Q'(x,y)$更新多项式$B(x,j)$。即敌手无法正确计算$B'(x,y)=B(x,y)+Q(x,y)$。即更新协议具有安全性。

**定理 4：**对任意的概率多项式敌手$A$，在连续的纪元中服务器节点更新协议具有安全性和完整性。

**证明：**由引理 2 可知，在转换阶段敌手无法获取未被腐化的 MEC 服务器私钥$sk$。由定理 3 完整性可知，$Q(x,y)$是随机生成的，$B'(x,y)=B(x,y)+Q(x,y)$独立与$B(x,y)$。此时，敌手无法计算得到$B'(x,y)$。最后， MEC 服务器计算$B'(i,j)$和证据$W_{B'(i,j)}$，传输数据可以用如下矩阵表示：

$$\begin{pmatrix} B'(1,1),W_{B'(1,1)} & \cdots & B'(1,2t+1),W_{B'(1,2t+1)} & \cdots & B'(1,n),W_{B'(1,n)} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ B'(n,1),W_{B'(n,1)} & \cdots & B'(n,2t+1),W_{B'(n,2t+1)} & \cdots & B'(n,n),W_{B'(n,n)} \end{pmatrix}$$

其中矩阵中行表示服务器$MS_i$发送的数据，列表示新增服务器$MS_j$收到的数据。在最坏情况下敌

手仅能获得上述数据传输矩阵中 $2t$ 行的数据，计算 $B(j,y')=\sum_{j=1}^{2t}\mathrm{B}(j,y')\prod_{i=1,j\neq i}^{2t}(y'-i)/(j-i)$，因为$B(j,y')$是$2t-1$阶 $2t$ 变元多项式，有$B(j,y')\neq B(j,y)$。在转换阶段结束后，进入新纪元。此时根据定义 2 可知，在$\langle t,n\rangle$门限秘密共享中，敌手不能通过$B(j,y)$获取服务器$MS_j$的私钥$sk_j$。

# 7 MEC 通信密钥管理协议

本节面向 MEC 网络，研究移动用户与 MEC 服务器之间通信和数据传输的密钥管理协议。如图 1 所示，可以看出在一个 MEC 网络中存在不同的区域，不同区域内的移动用户通过不同的基站传输信息和数据，通过不同的 MEC 服务器处理信息和数据。MEC 服务器与各区域移动用户之间的通信均为公开信道上，因此在设计 MEC 通信密钥管理协议时，密钥和通信数据都需要加密传输。此外，根据参考文献[13,77,92]对移动用户行为的分析，在 MEC 网络中一个区域中的移动用户是可以随时移动到另一个区域的，如图 4 所示。

在 MEC 网络中，移动用户的计算能力有限，所以移动用户的通信和数据传输使用对称密钥进行加密保护[77]。MEC 服务器在认证移动用户身份后，为其生成各类密钥。因为 MEC 服务器使用了二元多项式密钥管理协议分布式的管理服务器私钥，$MS_i$和$MS_j$之间可以生成对称密钥$B_{ij}$用于相互通信[18]。此外，MEC 服务器需要生成三种密钥：

- $Uek_i$：由$MS_i$生成，是$MS_i$和其管理区域移动用户$U_i$之间共享的唯一密钥。用于移动用户与$MS_i$之间的安全单点通信。
- $Bek_i$：由$MS_i$生成，并发送给其管理区域的移动用户，用来加密广播的$Tek$。
- $Tek$：由 MEC 服务器共同生成，是一个 MEC 网络中是所有移动用户和服务器共享的唯一对称密钥，用来加密通信和数据传输。

## 7.1 密钥的建立和分发

本节使用现有研究协议[13,79,81]中广泛采用的 PRF-HMAC-SHA-256[85]为生成密钥提供安全的伪随机函数，用来其确保数据的真实性且在传输过程中不会被敌手修改。移动用户$U_i$登录 MEC 网络，并通过门限身份认证的方式[80]获得身份验证令牌。$MS_i$通相同的密钥生成函数 PRF-HMAC-SHA-256[85] 为使用认证令牌 $AutToken=\{Token, Aut_{U_i}\}$请求服务的$U_i$生成对称密钥$Uek_i$。记伪随机函数 PRF-HMAC-SHA-256 算法为$H(\cdot)$，通过[86]的算法 5，$HMS$ 中的诚实 $MS_i$生成管理密钥$K_m$，并计算对称密钥：

$$Uek_i = H_1\,(K_m||Aut\mathrm{Token}_{U_i}||ID_G) \quad (1)$$

$MS_i$并通过$U_i$的公钥加密后发送给$U_i$。$U_i$收到$\{Uek_i\}\mathrm{PK}_{U_i}$后使用私钥解密并保存密钥$Uek_i$。在$Bek$更新后，服务器可以通过$Uek_i$将更新后的$Bek$发送给$U_i$。$MS_i$通过服务器私钥$sk_i$生成 $Bek_i$：

$$Bek_i = H_2\,(sk_i||ID_G||text) \quad (2)$$

当同一纪元中，需要更新时$Bek_i$时可以使用原来的$Bek_i$和私钥生成新的$Bek_i'$。即

$$Bek_i' = H_2\,(sk_i + Bek_i||ID_G||text)$$

在初始阶段，选取由于缺乏可信中心生成 $Tek$，需要至少$t+1$个$MS_i$提取主私钥[83]$g^{Msk}$，并计算流量加密密钥

$$Tek = H_3\,(g^{Msk}||K_m||text) \quad (3)$$

其中$g$是离散对数困难问题下循环群的生成元，$text$ 中包含各类参数。并加密发送给每个 MEC 服务器[18]。当$Tek$需要更新时，所有$MS_i$计算:$Tek' = H_3\,(Tek||K_m||text)$。

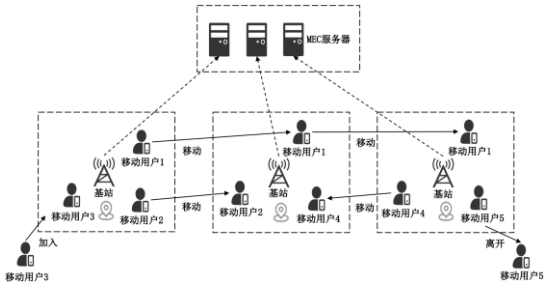


图 4 MEC 网络的移动用户行为

## 7.2 管理列表

为了避免$U_i$的移动导致的频繁密钥更新，本文引入管理列表[13,79,87]来记录动态用户的移动情况。每个$MS_i$维护自己的$Mdl$。每次成员到新区域时，记录移动用户的$ID_{U_i}$,$ID_G$及移动到的区域。

当$U_i$进入 $MS_i$管理区域时，$MS_i$查找其$Mdl$确认该$U_i$是返回用户还是新用户。如果$U_i$是返回用户，则$MS_i$将无需执行移动密钥更新协议。此外，$MS_i$还需要维护$Mcl$，记录当前$MS_i$管理区域$U_i$的信息。

## 7.3 面向MEC的密钥管理协议

使用不同应用的移动用户被分配在不同的 MEC 网络中。例如：智能视频用户和车联网用户

用户被分别分配在不同的 MEC 网络，他们无法随意获取其它 MEC 网络的通信和数据[5]。当$U_i$加入 MEC 网络后，每个 MEC 网络都有自己的$Tek$。为了保持前向安全性和后向安全性，当移动用户加入或离开 MEC 网络时，需要更新$Tek$。更新有以下三种情况[13,77]：

- 用户加入：当有一个新的移动用户加入$MS_i$管理区域时，需要$MS$更新$Bek$和$Tek$。以保证前向安全。例如图 4 中的移动设备 3。
- 用户移动：当一个移动用户从一个$MS_i$管理区域转移到$MS_j$管理区域时，$Tek$无需更新，但是$MS_j$需要重新生成$Bek_j$。例如图 4 中的移动设备 1、移动设备 2 和移动设备 4。
- 用户离开：当一个移动用户离开 MEC 网络时，需要$MS$重新生成$Tek$，另外，离开用户所在区域的管理服务器$MS_i$需要更新$Bek_i$。此时，$MS$通过$Bek$向 MEC 网络中的移动用户发送新的$Tek$。例如图 4 中的移动设备 5。

由上述分析，基于文献[11-13]的密钥管理协议构造思想，结合 MEC 网络的实际情况，一个面向 MEC 的密钥管理协议三种情况的协议如下：

**情况 1**：用户加入密钥管理协议

$U_i$通过验证后获得身份认证令牌，使用身份认证令牌加入 MEC 网络。

a) $U_i$向所在区域管理者发送加入请求：
$U_i \to MS_i: \{AutToken_{U_i}||join||pk_{U_i}\}sk_{U_i}$

b) $MS_i$验证请求，如果通过则生成$Uek_i$
$MS_i: \{\{AutToken_{U_i}||join\}sk_{U_i}\}pk_{U_i}$
$MS_i \to U_i: \{Uek_i||ID_{U_i}||ID_{MS_i}||text\}pk_{U_i}$

c) 为了保证前向安全，更新$Tek$和$Bek$：
$MS: Tek' = H_3\ (Tek||K_m||text)$
$MS_j \Longrightarrow U_j: \{ID_{U_i}||ID_{MS_j}||Tek'\}\ Bek_j$
$MS_i \Longrightarrow U_i: \{ID_{U_i}||ID_{MS_i}||Tek'||Bek_i'\}\ Uek_i$

**情况 2**：用户移动密钥管理协议

a) $U_i$向$MS_i$和目标区域$MS_j$发送移动请求：
$U_i \to MS_i: \{ID_{U_i}||ID_{MS_i}||ID_{MS_j}\}Uek_i$
$U_i \to MS_j: \{ID_{MS_i}||ID_{MS_j}||Aut_{U_i}||Token\}Uek_i$

b) 目标区域$MS_j$收到请求后，需要对$U_i$进行身份认证[13,92]，计算$Uek_i^* = H(K_m||Aut_{U_i}||ID_{MS_i}||\text{Token})$，并验证$Uek_i^* \stackrel{?}{=} Uek_i$。如果验证通过，则$MS_j$授权$U_i$访问。同时检查$Mdl$，如果没有记录$U_i$，则更新$Bek$：
$MS_j \to U_i: \{ID_{U_i}||ID_{MS_j}||Bek_i'\}\ Uek_i$
$MS_j \Longrightarrow U_j: \{ID_{U_i}||Bek_i'\}\ Bek_i$
如果$U_i$在$Mdl$中则发送当前$Bek$：
$MS_j \to U_i: \{ID_{U_i}||ID_{MS_j}||Bek_i\}\ Uek_i$

c) $MS_j$通知$MS_i$移动成功，$MS_i$和$MS_j$同时更新$Mcl$和$Mdl$。
$MS_j \to MS_i: \{ID_{U_i}||ID_{MS_i}||ID_{MS_j}\}B_{ij}$

**情况 3**：用户离开密钥管理协议

a) 当$U_i^*$离开$MS_i$时，向$MS_i$发送离开通知。
$U_i^* \to MS_i: \{ID_{U_i^*}||leave\}Uek_i$

b) $MS_i$收到离开通知后，更新$Tek$和$Bek_i$，以保证后向安全，更新$Mcl$和$Mdl$。
$MS_i \to U_i: \{ID_{U_i}||ID_{MS_i}||Tek'||Bek_i'\}Uek_i$

c) $MS_i$通知其它$MS$。$MS$收到后检查$Mdl$，如果$MS^*$中存在$U_i^*$则更新$Bek^*$
$MS^* \to U_i: \{ID_{U_i}||ID_{MS^*}||Tek'||Bek_i'\}Uek_i$

d) 其它$MS_j$向其管理的成员发送$Tek_{k+1}$
$MS_j \Longrightarrow U_i: \{ID_{U_i}||ID_{MS_j}||Tek'\}\ Bek_i'$

### 7.4 MListen模型

在实际情况中，存在未执行步骤 3(a)就离开 MEC 网络的移动用户。这将对密钥管理协议的后向安全造成影响。文献[13]中使用 MListen 作为发现群组通信中是否出现加入或离开事件的判定模型。本文将采用[13]的思想解决未执行步骤 3(a)就离开 MEC 网络的移动用户造成的问题。本节将基于文献[13,88,89,90]的结论，结合 MEC 的密钥管理协议的实际情况，针对未执行步骤 3(a)的离开用户提出一个密钥管理协议。首先给出 MListen 的思想和判定用户离开的结论[13,88,89,90]：

Almeroth 等人[88,89]指出加入通信的用户遵循泊松分布，进入速率未$\lambda$，而持续时间成指数分布，平均时间为1。一个随机选择的成员状态可以看作是一个马尔可夫过程[88]。令$i$表示所在区域的索引，$i \in [1, n]$，而索引$i = 0$表示用户不在区域内。$\lambda_{0,i}$表示一个新成员加入区域$i$。由于成员可以在不同区域移动，令$P = [p_{i,j}]$在各区域之间转移的概率且$p_{i,i} = 0$。$p_{0,i}$表示一个成员加入区域$i$，$p_{i,0}$表示一个成员离开区域$i$。结合文献[13]的分析和文献[88,89]的结论，在模型稳定状态下有：

$$\lambda_i = \lambda_{0,i} + \sum_{j=1}^{A} \lambda_j p_{j,i}$$

可知一个成员离开区域$i$的概率：

$$p_{i,0} = 1 - \sum_{j=1}^{A} p_{i,j}$$

成员加入区域通信的总速率$\lambda = \sum_{i=1}^{A} \lambda_{0,i}$。区域$i$中

成员数量$n_i = \lambda_i/\mu_i$。由 Jackson 定理[90]可得：

$$P(n_i = k) = \frac{n_i^k}{k!}e^{n_i}$$

基于该结论，在 MEC 网络中可以检测未执行 7.3 中 3(a)的移动用户。如果判定移动用户的状态为离开，则区域管理服务器检查$Mcl$，用户$U_i^*$所在区域的管理服务器$MS_i$执行以下操作：

a) $MS_i$判定$U_i^*$离开并更新$Tek$和$MS_i$的$Bek$，以保证后向安全，更新$Mcl$和$Mdl$。

$$MS_i \rightarrow U_i: \{ID_{U_i}||ID_{MS_i}||Tek'||Bek_i'\}Uek_i$$

b) $MS_i$通知其它$MS$。$MS$收到后检查$Mdl$，如果$MS^*$中存在$U_i^*$则更新$Bek^*$

$$MS^* \rightarrow U_i: \{ID_{U_i}||ID_{MS^*}||Tek'||Bek_i'\}Uek_i$$

c) 其它$MS_j$向其管理的成员发送$Tek'$

$$MS_j \Longrightarrow U_i: \{ID_{U_i}||ID_{MS_j}||Tek'\}\, Bek_i'$$

## 8 密钥管理协议安全性分析

MEC 服务器通过管理密钥和私钥份额来生成移动用户的通信密钥，本节首先分析 7.1 中生成密钥的独立性，然后基于第二节中介绍的安全模型通过形式化证明相的方式来分析第七节中面向 MEC 通信密钥管理协议的安全目标。

**定理 5：**在 7.1 节中，MEC 服务器为不同移动用户生成的用户加密密钥$Uek$是相互独立的；MEC 服务器为不同区域生成的广播密钥是相互独立的；同一用户生成的三类密钥是相互独立的。

证明：假设 SHA-256 是一种抗碰撞的哈希函数。在 7.1 节中，MEC 服务器生成的群组密钥包括$Uek$，$Bek$和$Tek$。其中$Uek$基于 MEC 服务器管理密钥$k_m$生成，$Bek$基于 MEC 服务器私钥$sk$生成，$Tek$基于 MEC 服务器管理密钥$k_m$和私钥$sk$共同生成，由 7.1 中式(1)、式(2)和式 (3)可以看出，对不同的密钥输入时不同的。

此时证明各密钥是相互独立的问题相当于证明对于不同的输入，密钥生成函数 PRF-HMAC-SHA-256[85]的输出是相互独立的。PRF-HMAC-SHA-256 算法基于 HMAC 框架构建，其安全性基于底层哈希算法的强度。即如果对于不同输入$x$，PRF-HMAC-SHA-256 密钥生成算法可以得到相关的密钥$K$，则意味着密钥生成算法可以找到 SHA-256 的整体碰撞。

因此在密钥生成算法中，MEC 服务器生成的密钥之间是相互独立的。即 MEC 服务器为不同移动用户生成的用户加密密钥$Uek$是相互独立的；MEC 服务器为不同区域生成的广播密钥是相互独立的；同一用户生成的三类密钥是相互独立的。

**定理 6：**对任意敌手 $A$，获得 MEC 网络中移动用户$U_i$的加密密钥$Uek_i$的优势为

$$Adv^{Uek}(A) \le (2\mathrm{n} + \mathrm{q} + \mathrm{e} + 3) \cdot negl(\lambda) + negl_{ADKG}(\lambda)$$

且协议各部分均满足：

前向安全：新加入 MEC 网络的移动用户不能通过 MEC 服务器生成的密钥解密之前网络中的消息。

后向安全：离开 MEC 网络的移动用户不能通过已有的密钥解密离开之后 MEC 网络的消息。

证明：$Uek_i$是移动用户$U_i$和$MS_i$之间共享的唯一密钥，假设敌手 $A$ 是以获取$U_i$的$Uek_i$为目标，并在$Bek$和$Tek$更新时，尝试在加入后获取更新前的密钥或者在离开后获取更新后的密钥，以破坏前向安全和后向安全。根据第二节中的安全假设，服务器是可信实体。敌手可以通过加入 MEC 网络，获取$Bek_i$和$Tek$。此外敌手可以知道用户和服务器的 ID。通过使用与[93]类似的技巧，定义一系列混合仿真游戏($Game_n$)。通过对对敌手 $A$ 进行分析，最后得出敌手 $A$ 通过计算和查询获取密钥的优势为零。首先定义下述事件：

—$succ_n^{K_m}$：$A$ 成功计算得到管理密钥$K_m$；

—$succ_n^{sk}$：$A$ 成功计算得到服务器私钥$sk$；

—$AskUek$：$A$ 通过对$K_m||Aut\mathrm{Token}_{U_i}||ID_G$查询$H_1$，从而成功计算出用户加密密钥$Uek$；

—$AskBek$：$A$ 通过对$sk_i||ID_G||text$查询$H_2$，从而成功计算出用户加密密钥$Bek$；

—$AskBek'$：$A$ 通过对$sk_i + Bek||ID_G||text$查询$H_2$，从而成功计算出用户的$Bek'$；

—$AskTek$：$A$ 通过对$g^{Msk}||K_m||text$查询$H_3$，从而成功计算出$Tek$；

—$AskTek'$：敌手通过对$Tek||K_m||text$查询$H_3$，从而成功计算出更新后的$Tek'$。

$Game_0$：该游戏对应敌手 $A$ 的真实攻击，即敌手计算得到其它成员的 $Uek$的优势为：

$$Adv^{Uek}(A) = Pr[succ_0^{K_m}]$$

敌手计算得到$Bek$的优势为：

$$Adv^{Bek}(A) = Pr[succ_0^{sk}]$$

$Game_1$：在该游戏中，敌手 $A$ 攻击用户加入协议，在这部分敌手需要计算或猜测$K_m$和$sk_i$进而计算其他用户的$Uek$，和更新前的$Bek$或$Tek$以破

坏协议的安全性。此时分为两种情况讨论：

—$A$是新加入的用户，此时 $A$ 可以获得的信息包括$\{Uek_A||ID_A||ID_{MS_i}||Tek'||Bek_i'\}$，由定义 6 可知，此时敌手通过$Uek_A$计算获得$K_m$的优势为：

$$Adv_{H_1}^{K_m}(A) \le negl_{K_m}(\lambda)$$

同理可知，敌手计算获得$MS_i$私钥$sk_i$优势为

$$Adv_{H_2}^{sk_i}(A) \le negl_{sk_i}(\lambda)$$

$A$ 通过$Bek_i'$和$Tek'$计算$Bek_i$和$Tek$的优势为

$$Adv_{H_2}^{Bek_i'}(A) \le negl_{Bek_i'}(\lambda)$$

$$Adv_{H_2}^{Tek'}(A) \le negl_{Tek'}(\lambda)$$

—$A$ 是老用户，此时 $A$ 拥有的信息包括$\{Uek_A||ID_A||ID_{MS}||Tek||Bek\}$，因为 $A$ 是可移动的，因此敌手最多可以拥有 MEC 网络中所有的$Bek$。在新成员加入后，$A$ 可以获取更新后的$\{Tek'||Bek_i'\}$，此时，敌手 $A$ 计算获得$K_m$和私钥$sk_i$的优势为

$$Adv_{H_1}^{K_m}(A) \le negl_{K_m}(\lambda)$$

$$Adv_{H_2}^{sk_i}(A) \le negl_{sk_i}(\lambda)$$

当 $A$ 不知道$K_m$时，构造算法$\Pi_1$执行[86]算法 5。由[86]的分析可知 $\Pr[AskUek] \le Adv^{ADKG}(n)$。此时，敌手得到其它成员$Uek$的优势为：

$$Adv^{Uek}(A) = Pr\left[succ_1^{K_m}\right]$$

$$= Pr\left[succ_0^{K_m}\right] + 2Adv_{H_1}^{K_m}(A) + Adv^{ADKG}(n)$$

$$\le Pr\left[succ_0^{K_m}\right] + negl_{K_m}(\lambda) + negl_{ADKG}(\lambda)$$

当 $A$ 不知道$sk_i$时，构造算法$\Pi_2$：$A$ 按照 5(1.1)运行，计算 t 个多项式$f(x,y)$，带入$i,j$，计算$s_{ji} = f(i,j)$，并计算服务器私钥$sk_i = \sum_{j\in HMS} s_{ji}$。

此时，$Pr[AskBek] = 1/(t \cdot poly\ (\mathrm{n}))$。即$\Pr[AskBek] \le t \cdot Adv^{poly}(n)$。敌手 $A$ 加入网络时，获得$Bek$的优势为：

$$Adv^{Bek}(A) = Pr[succ_1^{sk}]$$

$$= \Pr\left[succ_0^{sk}\right] + Adv_{H_2}^{Bek_i'}(A) + AskBek_1$$

$$\le Pr\left[succ_0^{sk}\right] + negl_{Bek_i'}(\lambda) + t \cdot Adv^{poly}(n).$$

根据第六章的分析可知，敌手 $A$ 获取$g^{Msk}$的优势为$Adv^{Msk}(A) \le negl_{Msk}(\lambda)$。因此，敌手 $A$ 加入网络时，计算获得更新前$Tek$的优势为:

$$Adv^{Tek}(A) = Pr[succ_1^{K_m}]$$

$$= Pr\left[succ_0^{K_m}\right] + 2Adv_{H_1}^{K_m}(A) + AskTek$$

$$\le Pr\left[succ_0^{sk}\right] + negl_{Bek_i'}(\lambda) + negl_{Msk}(\lambda).$$

此时计算 $K_m$ 和 $sk$ 的概率为 $Pr\left[succ_1^{K_m}\right] - Pr\left[succ_0^{K_m}\right] = Pr[succ_1^{sk}] - Pr[succ_0^{sk}] = 0$。

$Game_2$：在该游戏中，敌手攻击用户移动协议。在这个部分敌手需要计算或猜测$K_m$和$sk_i$进而计算其它移动用户的$Uek$。当 $A$ 第一次移动到新区域时，设 $A$ 移动了 $k$ 次，则 $A$ 获取新区域更新前$Bek$的优势为：

$$Adv^{Bek}(A) = \Pr[succ_0^{sk}] + k \cdot Adv_{H_2}^{Bek_i'}(A) + AskBek_1$$

$$\le Pr\left[succ_0^{sk}\right] + k \cdot negl_{Bek_i'}(\lambda) + t \cdot Adv^{poly}(n)$$

此外，$A$ 可以通过多次移动获得所有服务器的$Bek$，当敌手移动时，服务器的 Mcl 列表会记录敌手的移动轨迹，所以当 $A$ 在离开网络前，移动到进入过的服务器，则无需更新$Bek$，因为此时 $A$ 仍然保持通信状态且拥有$Tek$。此时保持前向安全性是无意义的，而且会增加网络的通信负担。在用户移动协议阶段，敌手可以获取的信息包括：

$\{Uek_A||ID_A||ID_{MS_i}||Tek||Bek_i\}$，$i = 1,\cdots,n$。

此时，敌手通过移动获得了 $n$ 个$Bek_i$，由定义 6 可知，$A$ 通过$Bek$计算获得私钥的优势为：

$$aAdv_{H_2}^{sk_i}(A) \le \mathrm{n} \cdot negl_{sk_i}(\lambda) = negl_{sk_i}^{n}(\lambda)$$

当敌手在网络中保持通信的时间足够久时，可以获得 $m$ 个$Bek'$计算获得私钥的优势为：

$$bAdv_{H_2}^{sk_i}(A) \le \mathrm{m} \cdot negl_{sk_i}(\lambda) = negl_{sk_i}^{m}(\lambda)$$

所以敌手在用户移动阶段计算私钥$sk$的优势为：

$$Adv_{H_2}^{sk_i}(A) = aAdv_{H_2}^{sk_i}(A) + bAdv_{H_2}^{sk_i}(A)$$

$$\le \mathrm{n} \cdot negl_{sk_i}(\lambda) + \mathrm{m} \cdot negl_{sk_i}(\lambda)$$

$$= negl_{sk_i}^{n}(\lambda) + negl_{sk_i}^{m}(\lambda)$$

由此可知$Pr[succ_2^{sk}] - Pr[succ_1^{sk}] = (n+m+1) \cdot Pr[succ_0^{sk}]$。此外，A 在$Game_2$中共获得了 e 个$Tek$。此时敌手得到其它成员$Uek$的优势为：

$$Adv^{Uek}(A) = Pr\left[succ_2^{K_m}\right] + (e-1) \cdot Pr\left[succ_2^{K_m}\right]$$

$$= e \cdot Pr\left[succ_2^{K_m}\right] \le e \cdot negl_{k_m}(\lambda)$$

此时，单次计算$K_m$的概率没有得到增加，即$Pr\left[succ_2^{K_m}\right] - Pr\left[succ_1^{K_m}\right] = 0$。

$Game_3$：在该游戏中，敌手攻击用户离开协议。分为两个情况讨论，首先时敌手继续保持在 MEC 网络中，此时其他用户离开，敌手没有获得额外的信息，此时$Game_3$与$Game_2$情况一致。当敌手 $A$ 离开时，敌手需要计算或猜测$K_m$和$sk_i$进而计算其他用户的$Uek$，或者计算$Bek'$和$Tek'$以破坏协议的安全性。$Game_3$分为以下两种情况讨论：

—$A$ 执行 7.3 中情况 3(a) 通知服务器离开。此时敌手计算获得管理密钥$K_m$和私钥$sk_i$，记该事件为$Tsucc_n^{K_m}$和$Tsucc_n^{sk}$。

—$A$ 未执行 7.3 中情况 3(a) 通知服务器离开。此时通过 MListen[88,89]判断敌手离开，服务器执行

7.4(a)(b)(c)。此时，敌手计算获得密钥管理密钥$K_m$和私钥，记该事件为$Fsucc_n^{K_m}$和$Fsucc_n^{sk}$。

首先，考虑$Tsucc_n^{K_m}$和$Tsucc_n^{sk}$。当 $A$ 向服务器发送离开请求后，服务器生成$\{Tek'||Bek_i'\}$。若 $A$ 获得$Uek_i$，则可以获得接收到$\{Tek'||Bek_i'\}$。设 $A$ 在 MEC 网络中共获得 $h$ 个$Tek$ 。$A$ 通过计算获得$Uek_i$的优势为：

$$\begin{aligned}&Adv^{Uek}(A)=Pr\left[succ_3^{K_m}\right]+(n-1)Pr\left[succ_3^{K_m}\right]\\&=n\cdot Adv_{H_1}^{K_m}(A)\le \mathrm{n}\cdot negl_{K_m}(\lambda)\end{aligned}$$

由此可得：

$Pr\left[succ_3^{K_m}\right]-Pr\left[succ_2^{K_m}\right]\le(\mathrm{n}-1)\cdot negl_{K_m}(\lambda)$。此时，$A$ 为获得新的消息，计算$sk$的优势与$Game_2$相同。则 $A$ 通过计算获得$Bek_i'$的优势为：

$$\begin{aligned}&Adv^{Bek_i'}(A)=Adv^{Uek}(A)+Adv^{sk_i}(A)+AskBek'\\&=Adv_{H_1}^{K_m}(A)+Adv_{H_2}^{sk_i}(A)+t\cdot Adv^{poly}(n)\\&\le negl_{K_m}(\lambda)+negl_{sk_i}(\lambda)+1/(t\cdot poly(\mathrm{n}))\end{aligned}$$

且$Pr[succ_3^{sk}]-Pr[succ_2^{sk}]=0$。

然后执行$AskTek'$获得对$Tek||K_m||text$查询$H_3$，从而成功计算出更新后的$Tek'$的优势为：

$$\begin{aligned}Adv^{Tek'}(A)&=Adv^{Uek}(A)+AskTek'+Adv^{Bek_i'}(A)\\&=2Adv_{H_1}^{K_m}(A)+Adv^{ADKG}(n)\\&\quad+Adv_{H_2}^{sk_i}(A)+t\cdot Adv^{poly}(n)\\&\le negl_{K_m}(\lambda)+negl_{ADKG}(\lambda)+negl_{K_m}(\lambda)\\&\quad+negl_{sk_i}(\lambda)+1/(t\cdot poly(\mathrm{n}))\end{aligned}$$

最后，考虑$Fsucc_n^{K_m}$和$Fsucc_n^{sk}$。$A$ 不发送离开网络请求，但不在 MEC 网络中进行通信。此时仍然可以通过$Uek_A$接收更新的$Bek_i'$，并通过最新的$Bek_i'$得到 $Tek'$。直到被 MListen 判定离开。令 $A$ 额外接收了$p$个$Bek_i'$和$q$个$Tek'$。此时，$A$ 计算$Uek_i$的优势为：

$$\begin{aligned}Adv^{Uek}(A)&=Pr\left[succ_3^{K_m}\right]+(n+q-1)\cdot Pr\left[succ_3^{K_m}\right]\\&=(n+q)\cdot Adv_{H_1}^{K_m}(A)\\&\le(\mathrm{n}+\mathrm{q})\cdot negl_{K_m}(\lambda)\end{aligned}$$

在敌手 $A$ 接收不到新的$Bek_i'$后，$A$ 通过计算获得$Bek_i'$的优势为：

$$\begin{aligned}Adv^{Bek_i'}(A)&=Adv^{Uek}(A)+(n+p)Adv^{sk_i}(A)\\&\quad+AskBek'\\&=Adv_{H_1}^{K_m}(A)+(n+p)Adv^{sk_i}(A)\\&\quad+t\cdot Adv^{poly}(n)\\&\le negl_{K_m}(\lambda)+(n+p)\cdot negl_{sk_i}(\lambda)\\&\quad+1/(t\cdot poly(\mathrm{n}))\end{aligned}$$

在敌手 $A$ 接收不到新的$Tek_i'$后，$A$ 通过计算获得$Tek_i'$的优势为：

$$\begin{aligned}Adv^{Tek_i'}(A)&=(n+q-1)\cdot Pr\left[succ_3^{K_m}\right]\\&\quad+Adv^{Bek_i'}(A)+(n+q)\cdot AskTek'\\&=(n+q-1)\cdot Adv_{H_1}^{K_m}(A)+Adv_{H_1}^{K_m}(A)\\&\quad+(n+p)Adv^{sk_i}(A)+t\cdot Adv^{poly}(n)\\&\quad+(n+q)\cdot Adv^{ADKG}(n)\\&=(\mathrm{n}+\mathrm{q})\cdot negl_{K_m}(\lambda)\\&\quad+(n+p)\cdot negl_{sk_i}(\lambda)\\&\quad+1/(t\cdot poly(\mathrm{n}))+(n+p)\cdot negl_{ADKG}(\lambda)\end{aligned}$$

此时，$Pr[succ_3^{sk}]-Pr[succ_2^{sk}]=p\cdot Pr[succ_0^{sk}]$，且$Pr\left[succ_3^{K_m}\right]-Pr\left[succ_2^{K_m}\right]=q\cdot Pr\left[succ_0^{K_m}\right]$。

经过上述游戏$(Game_0,Game_1,Game_2,Game_3)$的分析可知，$A$ 通过执行面向 MEC 的密钥管理协议可以成功获取服务器管理密钥$K_m$的优势为：

$$\begin{aligned}Adv^{K_m}(A)&=Pr\left[succ_0^{K_m}\right]+Pr[succ_1^{sk}]-Pr[succ_0^{sk}]\\&\quad+Pr\left[succ_2^{K_m}\right]-Pr\left[succ_1^{K_m}\right]\\&\quad+Pr\left[succ_3^{K_m}\right]-Pr\left[succ_2^{K_m}\right]\\&=Pr\left[succ_0^{K_m}\right]+q\cdot Pr\left[succ_0^{K_m}\right]\\&\le(q+1)\cdot negl(\lambda)\end{aligned}$$

$A$ 通过执行面向 MEC 的密钥管理协议可以成功获取服务器私钥$sk$的优势为：

$$\begin{aligned}Adv^{sk}(A)&=Pr\left[succ_0^{sk}\right]+Pr[succ_1^{sk}]-Pr[succ_0^{sk}]\\&\quad+Pr[succ_2^{sk}]-Pr[succ_1^{sk}]\\&\quad+Pr[succ_3^{sk}]-Pr[succ_2^{sk}]\\&=q\cdot Pr\left[succ_0^{K_m}\right]\\&\quad+(n+m+1)\cdot Pr[succ_0^{sk}]+Pr[succ_0^{sk}]\\&=(q+n+m+2)\cdot Pr[succ_0^{sk}]\\&\le(q+n+m+2)\cdot negl(\lambda)\end{aligned}$$

此外，由上述四个阶段是游戏的分析，可以看出在面向 MEC 的密钥管理协议的加入阶段，$A$ 获得加入前的密钥$Bek_i$和$Tek$的优势为：

$$\begin{aligned}&Adv^{Bek}(A)+Adv^{Tek}(A)\\&\le negl_{Bek_i'}(\lambda)+negl_{Tek'}(\lambda)+2\cdot Pr\left[succ_0^{sk}\right]\\&\quad+negl_{Bek_i'}(\lambda)+t\cdot Adv^{poly}(n)\\&\quad+negl_{Bek_i'}(\lambda)\ +negl_{Msk}(\lambda)=Negl(\lambda)\end{aligned}$$

保持了前向安全。同理可知在面向 MEC 的密钥管理协议的离开阶段，$A$ 获得加入前的密钥$Bek_i$和$Tek$的优势为：

$$\begin{aligned}&Adv^{Bek}(A)+Adv^{Tek}(A)\\&=3\cdot Adv^{Uek}(A)+Adv^{Bek_i'}(A)\\&\quad+(2n+2p+1)Adv^{sk_i}(A)\\&\quad+(n+q)\cdot Adv_{H_1}^{K_m}(A)+3t\cdot Adv^{poly}(n)\\&\quad+(n+q+2)\cdot Adv^{ADKG}(n)\le Negl(\lambda)\end{aligned}$$

保持了后向安全。在移动阶段，如果 $A$ 第一次进入区域，该区域的管理服务器会更新$Bek$，此时敌手获得更新前$Bek$的优势为

$$\begin{aligned}&Adv^{Bek}(A)\\&\le Pr\left[succ_0^{sk}\right]+\ k\cdot negl_{Bek_i'}(\lambda)+t\cdot Adv^{poly}(n)\end{aligned}$$

在面向 MEC 的密钥管理协议运行阶段，A 通过计算和查询获取其它移动用户$Uek_i$的优势为：

$$\begin{aligned}Adv^{Uek}(A) = {} & Pr\left[succ_0^{K_m}\right] + Pr\left[succ_0^{K_m}\right] \\ & +2Adv_{H_1}^{K_m}(A) + Adv^{ADKG}(n) \\ & +e \cdot Pr\left[succ_2^{K_m}\right] + n \cdot Adv_{H_1}^{K_m}(A) \\ & +(n+q) \cdot Adv_{H_1}^{K_m}(A) \\ \leq {} & (2\mathrm{n} + \mathrm{q} + \mathrm{e} + 3) \cdot negl(\lambda) \\ & + negl_{ADKG}(\lambda)\end{aligned}$$

定理 6 得证。　　证毕。

## 9 本文协议效率分析

本文首先基于 Virgo[69]和 NIROA[71]技术构造一种非交互透明 zkVPD。因为构造中涉及 GKR 协议[62]，并使用 Merkle 树[74]承诺多项式，故该协议的由[69,71]可知，可计算非交互 zkVPD 协议每轮证明的时间复杂度为$O(D\log C + m\log m + C)$，验证的时间复杂度为$O(D\log C + \log^2 m)$，证明大小为$O(D\log C + \log^2 m)$。在文献[69]分析的基础上，本文将 5 种不同构造的透明零知识证明在非交互式协议下的运行效率与本文进行对比，如表 4 所示。此外，本文的协议无需可信中心初始化生成 SRS，而类似的协议[61]在证据大小$m = 2^{20}$时，可信初始化生成 SRS 需要 12.6s 的时间[69]。因此本文提出非交互式零知识可验证多项式委托协议在验证多项式计算正确性上是高效的。在本文面向 MEC 服务器的私钥管理协议中，涉及到数乘运算、多项式求值运算、拉格朗日插值运算。其中数乘运算计算量较小，可以忽略不记。令 $t$ 阶多项式求值运算的计算复杂度为$O(t)$，Lagrange 插值运算的时间复杂度为$O(t^2)$。私钥管理协议在各阶段计算和通信复杂度如表 2 所示。

在密钥管理协议中，采用对称密钥加密信息，保持了较高的效率。对密钥管理协议的影响主要来源于 1-affects-n ，即某个移动用户发生变化时，对其他用户造成影响的数量[13,77]。由于缺乏适合 MEC 网络的密钥管理，每次重新生成密钥时，通过单点传输或广播传输消息，消耗了大量网络带宽，导致密钥分发延迟和通信服务中断。

本文提出的面向 MEC 的密钥管理协议，通过分布式网络管理移动用户，分散了传输、计算和存储的压力，参考[13]的方法，引入成员管理列表$Mcl$和$Mdl$，记录了敌手的移动轨迹，减少因为用户频繁移动带来的密钥重构。本文针对 5 种使用$Tek$密钥的协议进行比较。表 3 展示了在用户移动时，所需的密钥重构的传输负载。为了方便与

表 2 私钥管理协议各部分的计算和通讯复杂度

| | 私钥生成 | 秘密共享 | 秘密重构 | 秘密份额更新 | 服务器更新 |
|---|---|---|---|---|---|
| 计算复杂度 | $O(t^2)$ | $O(t^3)$ | $O(t^2)$ | $O(t^3)$ | $O(t^3)$ |
| 通讯复杂度 | $O(t^2)$ | $O(t^2)$ | $O(t)$ | $O(t^2)$ | $O(t^2)$ |

表 3 用户移动密钥管理协议传输负载

| 协议 | $AKM \leftrightarrow DKM$ | $U \leftrightarrow AKM$ | $U \leftrightarrow DKM$ |
|---|---|---|---|
| Wong[91] | 无 | 无 | 2log n |
| Decleene [92] | 无 | 4 | 无 |
| Kiah [77] | 无 | 5 | 无 |
| Daghighi [13] | 3 | 3 | 无 |
| 本文的协议 | 无 | 3 | 无 |

[13,77,91,92]做对比分析，表 3 中称 MEC 服务器为 AKM(区密钥管理器)，称中心服务器为 DKM(域密钥管理器)。在表 3 中，[91]是集中式管理方案，不使用 AKM 进行管理，造成了较高的通信负载。而[13,77,92]均采用双层结构，需要可信中心提供$Tek$密钥。从表 3 可以看出本文协议无需网络中心云服务器参与，用户与 MEC 服务器之间也只有较低的通信负载。协议具有较高的效率。

## 10 与相关协议的对比分析

本节将本文设计的协议与相关的协议进行对比分析。其中 zkVPD 协议的效率对比结果如表 4 所示，私钥管理协议的性能对比结果如表 5 所示，密钥管理协议的性能对比如表 6 所示。本文的三个协议为递进关系，最终由 MEC 服务器私钥管理协议和 MEC 通信密钥管理协议构成了一个完整的面向 MEC 的密钥管理方案。因此，本节中分别分析了这三个方案与相关协议的对比。

zkVPD 协议的效率主要由证明时间、验证时间和证明大小来衡量。一般来说，证据大小 $m$ 最高取到$2^{22}$，而电路 $C$ 的规模可高达$2^{26}$[60]。因此可以说$m \ll C$。由表 4 可以看出，本文构造的协议证明和验证时间仅与证据大小呈对数关系，在非交互条件下仍是一种简洁透明的 zkVPD 协议，且证明效率在对比协议[43,44,48,50,56]中是最高效的。此外本文基于 Merkle 树进行承诺，减少了大量的模

表 4 透明零知识证明方案性能

| | Ames [44] | Bunz [43] | Wahby [56] | Ben [48] | Ben [50] | 本文改进的协议 |
|---|---|---|---|---|---|---|
| 证明时间 | $O(C\log C+\sqrt{C}+C)$ | $O(\log C+2C)$ | $O(C\log C+D\log C+\sqrt{m})$ | $O(C\log^2 C+\log^2 C)$ | $O(C\log C+\log^2 C+C)$ | $O(D\log C+m\log m+C)$ |
| 验证时间 | $O(C)$ | $O(C)$ | $O(D\log C+\sqrt{m})$ | $O(\log^2 C)$ | $O(C)$ | $O(D\log C+\log^2 m)$ |
| 证明大小 | $O(\sqrt{C})$ | $O(\log C)$ | $O(D\log C+\sqrt{m})$ | $O(\log^2 C)$ | $O(\log^2 C)$ | $O(D\log C+\log^2 m)$ |

表 5 私钥管理协议性能对比

| | Schultz[66] | Maram[17] | 本文的协议 |
|---|---|---|---|
| 无可信中心 | √ | √ | √ |
| 去可信初始化 | × | × | √ |
| 通讯复杂度 | $O(n^3)$ | $O(n^2)$ | $O(t^2)$ |
| 主动更新 | √ | √ | √ |
| 动态服务器 | × | √ | √ |
| 承诺算法 | Feldman | Kate | Merkle 树 |

表 6 密钥管理协议性能对比

| | Wong [91] | Decleene [92] | Kiah [77] | Daghighi [13] | 本文的协议 |
|---|---|---|---|---|---|
| 层数 | 1 | 2 | 2 | 2 | 1 |
| 分布式处理 | × | √ | √ | √ | √ |
| 无需中心管理 | × | × | × | × | √ |
| 移动密钥管理 | × | √ | √ | √ | √ |
| 防止单点故障 | × | √ | √ | √ | √ |
| 用户管理列表 | × | √ | √ | √ | √ |
| 用户移动认证 | √ | × | √ | √ | √ |

幂运算和双线性对运算，所以在证明生成和验证都具有较高的效率。该协议的证明大小仅大于子弹证明[53]。与其它几个证明大小相差较小。

表 5 从有无可信中心、验证协议是否去可信初始化、通讯复杂度、是否支持主动更新、是否支持成员动态变更以及使用了何种算法承诺多项式。从表 5 可以看出本文的性能相比与其它两种方案[27,76]，在验证阶段不需要可信中心，一方面提高了协议的运行效率，另一方面更加适合无可信中心的分布式 MEC 网络。本文的通讯复杂度低于 Schultz 等人[76]的方案且支持服务器动态变化。此外，本文的私钥管理协议使用 Merkle 树[74]承诺多项式，减少了大量的模幂运算和双线性对运算，承诺和验证效率都得到了较大的提升。

如表 6 所示，在文献[13]研究基础上，本文从协议结构层数、是否为分布式处理密钥、是否需要中心辅助管理和计算、是否能够管理移动用户密钥、是否可以防止单点故障、是否使用列表管理和是否对用户的移动进行认证 7 个方面，将本文与其它四个使用$Tek$的方案[13,77,91,92]进行了功能比较。由表 6 可以看出，本文提出的协议在分布式网络的基础上仅使用了一层结构对移动用户的数据加密密钥进行管理，且无需可信计算中心参与。此外在其它性能方面具有和其它方案向当的性能，是一种完全分布式的密钥管理协议。结合表 3 可以看出本文提出的面向 MEC 的密钥管理协议具有良好的性能和较高的效率。

## 11 结束语

随着 MEC 的快速发展和广泛应用，MEC 网络的安全问题也逐渐凸显。密钥是保护 MEC 网络通信和数据安全的基础。需要研究适合 MEC 网络的密钥管理协议，保护 MEC 网络中服务器和移动用户的密钥安全。本文的研究分为两部分：首先构造了一个非交互透明零知识可验证多项式委托

并基于该技术设计了一种支持通信秘密可验证、秘密份额可动态更新、服务器节点可动态变化的门限秘密共享协议，用来保护 MEC 服务器私钥的安全；然后本文提出一种支持用户自由进出，保障网络通信安全的 MEC 通信密钥管理协议，用来保护 MEC 网络中移动用户的通信和数据安全。通过对协议安全性分析、效率分析以及对比其它协议表明本文提出的密钥管理协议实现了更好的安全性、更高的效率并保持了良好的性能，证明了本文提出的密钥管理协议适于 MEC 网络环境。

## 参 考 文 献


[1] Shi W, Cao J, Zhang Q, et al. Edge computing: vision and challenges[J]. IEEE internet of things journal, 2016, 3(5): 637-646.

[2] Rimal B P, Van D P, Maier M. Mobile edge computing empowered fiber-wireless access networks in the 5G era[J]. IEEE Communications Magazine, 2017, 55(2): 192-200.

[3] Bastug E, Bennis M, Debbah M. Living on the edge: The role of proactive caching in 5G wireless networks[J]. IEEE Communications Magazine, 2014, 52(8): 82-89.

[4] Mehta A, Tärneberg W, Klein C, et al. How beneficial are intermediate layer data centers in mobile edge networks?[C]//2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems. Augsburg, Germany, 2016: 222-229.

[5] Hu W, Gao Y, Ha K, et al. Quantifying the impact of edge computing on mobile applications[C]//Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems. New York, USA, 2016: 1-8.

[6] Jalali F, Hinton K, Ayre R, et al. Fog computing may help to save energy in cloud computing[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(5): 1728-1739.

[7] Wang C, Wang D, Tu Y, et al. Understanding node capture attacks in user authentication schemes for wireless sensor networks [C]// IEEE Transactions on Dependable and Secure Computing, 2020, Doi: 10.1109/TDSC.2020.2974220

[8] Wang C, Wang D, Wang F, et al. Multi-factor user authentication scheme for multi-gateway wireless sensor networks[J]. Chinese Journal of Computers, 2020 43(4): 683-700.
(王晨宇，汪定，王菲菲，徐国爱. 面向多网关的无线传感器网络多因素认证协议. 计算机学报, 2020 43(4): 683-700.)

[9] Wang D, Wang P, Wang C. Efficient multi-factor user authentication protocol with forward secrecy for real-time data access in WSNs[J]. ACM Transactions on Cyber-Physical Systems, 2020, 4(3): 1-26.

[10] Chen Z, Li S, Wu Q, et al. A distributed secret share update scheme with public verifiability for ad hoc network[J]. Security and Communication Networks, 2015, 8(8): 1485-1493.

[11] Jeong I R, Lee D H. Key agreement for key hypergraph[J]. computers & security, 2007, 26(7-8): 452-458.

[12] Sciancalepore S, Capossele A, Piro G, et al. Key management protocol with implicit certificates for IoT systems[C]//Proceedings of the the 13th Annual International Conference on Mobile Systems, Applications, and Services, New York, USA, 2015: 37-42.

[13] Daghighi B, Kiah M, Iqbal S, et al. Host mobility key management in dynamic secure group communication[J]. Wireless Networks, 2018, 24(8): 3009-3027.

[14] Ibrahim M H. Octopus: an edge-fog mutual authentication scheme[J]. International Journal of Network Security, 2016, 18(6): 1089-1101.

[15] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.

[16] Blakley G. Safeguarding cryptographic keys [C]// Proceedings of National Computer Conference. New York, USA: ACM Press, 1979: 313-317.

[17] Ingemarsson I, Simmons J. A protocol to set up shared secret schemes without the assistance of a mutually trusted party[C]//Workshop on the Theory and Application of of Cryptographic Techniques. Berlin, Heidelberg: Springer, 1990: 266-282.

[18] Harn L, Hsu F. (t, n) multi-secret sharing scheme based on bivariate polynomial[J]. Wireless Personal Communications, 2017, 95(2): 1495-1504.

[19] Baron J, El Defrawy K, Lampkins J, et al. Communication-optimal proactive secret sharing for dynamic groups[C]//International Conference on Applied Cryptography and Network Security. Berlin, Heidelberg: Springer, 2015: 23-41.

[20] Cachin C, Kursawe K, Lysyanskaya A, et al. Asynchronous verifiable secret sharing and proactive cryptosystems[C]//Proceedings of the 9th ACM Conference on Computer and Communications Security. New York, USA, 2002: 88-97.

[21] Chor B, Goldwasser S, Micali S, Awerbuch B. Verifiable secret sharing and achieving simultaneity in the presence of faults [C]// Proceedings of the 26th IEEE Symposium on Foundations of Com-puter Science. Portland, USA, 1985: 383-395.

[22] Harn L, Fuyou M, Chang C C. Verifiable secret sharing based on the Chinese remainder theorem[J]. Security and Communication Networks, 2014, 7(6): 950-957.

[23] Zaghian A, Bagherpour B. A Fast Publicly Verifiable Secret Sharing Scheme using Non-homogeneous Linear Recursions[J]. The ISC International Journal of Information Security, 2020, 12(2): 91-99.

[24] Bagherpour B, Janbaz S, Zaghian A. A fast non-interactive publicly verifiable secret sharing scheme[C]// 17th International ISC Conference on Information Security and Cryptology, Tehran, Iran, 2020: 20227088.

[25] Schoenmakers B. A simple publicly verifiable secret sharing scheme and its application to electronic voting[C]//Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1999: 148-164.

[26] Herzberg A, Jarecki S, Krawczyk H, et al. Proactive secret sharing or: How to cope with perpetual leakage[C]//Annual international cryptology conference. Berlin, Heidelberg: Springer, 1995: 339-352.

[27] Maram S, Zhang F, Wang L, et al. Churp: Dynamic-committee proactive secret sharing[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York, USA, 2019: 2369-2386.

[28] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems[J]. SIAM Journal on computing, 1989, 18(1): 186-208.

[29] Pedersen T. Non-interactive and information-theoretic secure verifyable secret sharing[C]//Annual international cryptology conference. Springer, Berlin, Heidelberg, 1991: 129-140.

[30] Kilian J. A note on efficient zero-knowledge proofs and arguments[C]// Proceedings of the 24th Annual ACM Symposium on Theory of Computing, New York, USA, 1992:436-445.

[31] Micali S. Computationally sound proofs[J]. SIAM Journal on Computing, 2000, 30(4): 1253-1298.

[32] Groth J. Short pairing-based non-interactive zero-knowledge arguments[C]//International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer, 2010: 321-340.

[33] Ishai Y, Kushilevitz E, Ostrovsky R. Efficient arguments without short PCPs[C]//Twenty-Second Annual IEEE Conference on Computational Complexity. San Diego, USA, 2007: 278-291.

[34] Gennaro R, Gentry C, Parno B, et al. Quadratic span programs and succinct NIZKs without PCPs[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer, 2013: 626-645.

[35] Lipmaa H. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments[C]//Theory of Cryptography Conference. Berlin, Heidelberg: Springer, 2012: 169-189.

[36] Ben-Sasson E, Chiesa A, Genkin D, et al. SNARKs for C: Verifying program executions succinctly and in zero knowledge[C]//Annual cryptology conference. Berlin, Heidelberg: Springer, 2013: 90-108.

[37] Ben-Sasson E, Chiesa A, Tromer E, et al. Scalable zero knowledge via cycles of elliptic curves[J]. Algorithmica, 2017, 79(4): 1102-1160.

[38] Braun B, Feldman A J, Ren Z, et al. Verifying computations with state[C]// Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles. Pennsylvania, USA, 2013:341-357.

[39] Costello C, Fournet C, Howell J, et al. Geppetto: Versatile Verifiable Computation[C]// 2015 IEEE Symposium on Security and Privacy. San Jose, USA, 2015:253-370.

[40] Fiore D, Fournet C, Ghosh E, et al. Hash First, Argue Later: Adaptive Verifiable Computations on Outsourced Data[C]// Acm SIGSAC Conference on Computer & Communications Security. Vienna, Austria, 2016:1304-4316.

[41] Parno B, Howell J, Gentry C, et al. Pinocchio: Nearly Practical Verifiable Computation[C]// Proceedings of the 2013 IEEE Symposium on Security and Privacy. Berkeley, USA, 2013: 13597232.

[42] Wahby R, Setty S, Ren Z, et al. Efficient ram and control flow in verifiable outsourced computation[C]// NDSS Copyright 2015 Internet Society. DOI: 10.14722/ndss.2015.23097

[43] Ben-Sasson E, Chiesa A, Garman C, et al. Zerocash: Decentralized Anonymous Payments from Bitcoin[C]// 2014 IEEE Symposium on Security and Privacy (SP). Berkeley, USA, 2014:459-474.

[44] Bowe S, Chiesa A, Green M, et al. Zexe: Enabling decentralized private computation[C]//2020 IEEE Symposium on Security and Privacy. San Francisco, USA, 2020: 947-964.

[45] Kosba A, Miller A, Shi E, et al. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts[C]// IEEE Symposium on Security and Privacy. San Jose, USA, 2016:839-858.

[46] Kate A, Zaverucha G, Goldberg I. Constant-Size Commitments to Polynomials and Their Applications[C]// International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer, 2010:177-194.

[47] Ben-Sasson E, Chiesa A, Green M, et al. Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs[C]// 2015 IEEE Symposium on Security and Privacy. San Jose, USA, 2015:15308310.

[48] Groth J, Kohlweiss M, Maller M, et al. Updatable and universal common reference strings with applications to zk-SNARKs[C]// Annual International Cryptology Conference. Cham: Springer, 2018:698-728.

[49] Maller M, Bowe S, Kohlweiss M, et al. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings[C]//Proceedings of the 2019 ACM SIGSAC Confe-rence on Computer and Communications Security, London, UK. 2019: 2111-2128.

[50] Bayer S, Groth J. Efficient zero-knowledge argument for correctness of a shuffle[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer, 2012: 263-280.

[51] Bootle J, Cerulli A, Chaidos P, et al. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting[C]//Ann-ual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer, 2016: 327-357.

[52] Groth J. Linear algebra with sub-linear zero-knowledge arguments[C]//Annual International Cryptology Conference. Berlin, Heidelberg: Springer, 2009: 192-208.

[53] Bünz B, Bootle J, Boneh D, et al. Bulletproofs: Short proofs for confidential transactions and more[C]//2018 IEEE Symposium on Security and Privacy. San Francisco, USA, 2018: 315-334.

[54] Ames S, Hazay C, Ishai Y, et al. Ligero: Lightweight sublinear arguments without a trusted setup[C]//Proceedings of the 2017 ACM sigsac conference on computer and communications security. Dallas Texas USA, 2017: 2087-2104.

[55] Chase M, Derler D, Goldfeder S, et al. Post-quantum zero-knowledge and signatures from symmetric-key primitives[C]//Pro-ceedings of the 2017 ACM SIGSAC conference on computer and com-munications security. Dallas Texas USA , 2017: 1825-1842.

[56] Giacomelli I, Madsen J, Orlandi C. Zkboo: Faster zero-knowledge for boolean circuits[C]//25th Usenix security symposium. Austin, USA, 2016: 1069-1083.

[57] Ishai Y, Kushilevitz E, Ostrovsky R, et al. Zero-knowledge from secure multiparty computation[C]//Proceedings of the thirtyninth annual ACM symposium on Theory of computing. San Diego California USA , 2007: 21-30.

[58] Ben-Sasson E, Bentov I, Horesh Y, et al. Scalable, transparent, and post-quantum secure computational integrity[J]. International association of cryptological research, 2018, 2018: 46.

[59] Bootle J, Cerulli A, Ghadafi E, et al. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability[C]//International Confer-ence on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2017: 336-365.

[60] Ben-Sasson E, Chiesa A, Riabzev M, et al. Aurora: Transparent succinct arguments for R1CS[C]//Annual international conference on the theory and applications of cryptographic techniques. Cham: Springer, 2019: 103-128.

[61] Zhang Y, Genkin D, Katz J, et al. vSQL: Verifying arbitrary SQL queries over dynamic outsourced databases[C]//2017 IEEE Symposium on Security and Privacy. San Jose, USA, IEEE, 2017: 863-880.

[62] Goldwasser S, Kalai T, Rothblum N. Delegating computation: interactive proofs for muggles[J]. Journal of the ACM, 2015, 62(4): 1-64.

[63] Cramer R, Damgård I. Zero-knowledge proofs for finite field arithmetic, or: Can zero-knowledge be for free? [C]//Annual Internation-al Cryptology Conference. Berlin, Heidelberg: Springer, 1998: 424-441.

[64] Chiesa A, Forbes M A, Spooner N. A zero knowledge sumcheck and its applications[J]. arXiv preprint arXiv:1704.02086, 2017.

[65] Zhang Y, Genkin D, Katz J, et al. vRAM: Faster verifiable RAM with program-independent preprocessing[C]//2018 IEEE Sympo-sium on Security and Privacy. San Francisco, USA, 2018: 908-925.

[66] Wahby R, Tzialla I, Shelat A, et al. Doubly-efficient zkSNARKs without trusted setup[C]//2018 IEEE Symposium on Security and Privacy. San Francisco, USA, 2018: 926-943.

[67] Xie T, Zhang J, Zhang Y, et al. Libra: Succinct zero-knowledge proofs with optimal prover computation[C]//Annual International Cryptology Conference. Cham: Springer, 2019: 733-764.

[68] Zhang Y, Genkin D, Katz J, et al. A Zero-Knowledge Version of vSQL[J]. International association of cryptological research. 2017(1): 1146-1169.

[69] Zhang J, Xie T, Zhang Y, et al. Transparent polynomial delegation and its applications to zero knowledge proof[C]//2020 IEEE Symposium on Security and Privacy. San Francisco, USA, 2020: 859-876.

[70] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems[C]//Conference on the theory and application of cryptographic techniques. Berlin, Heidelberg: Springer, 1986: 186-194.

[71] Ben-Sasson E, Chiesa A, Spooner N. Interactive oracle proofs[C]//Theory of Cryptography Conference. Berlin, Heidelberg: Springer, 2016: 31-60.

[72] Lund C, Fortnow L, Karloff H, et al. Algebraic methods for inter-active proof systems[J]. Journal of the ACM, 1992, 39(4): 859-868.

[73] Byott N P, Chapman R J. Power sums over finite subspaces of a field[J]. Finite Fields and Their Applications, 1999, 5(3): 254-265.

[74] Merkle C. A certified digital signature[C]//Conference on the Theory and Application of Cryptology. New York: Springer, 1989: 218-238.

[75] Cormen H, Leiserson E, Rivest L, et al. Introduction to algorithms[M]. MIT press, 2009.

[76] Schultz A, Liskov B, Liskov M. Mobile proactive secret sharing[C]//Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing. Toronto, Canada, 2008: 458-458.

[77] Kiah M, Martin M. Host mobility protocol for secure group communication in wireless mobile environments[C]//Future Generation Communication and Networking. Jeju, Korea, 2007, 1: 100-107.

[78] Srinivas V, Ruan L. An efficient reliable multicast protocol for 802.11-based wireless LANs[C]//IEEE International Sympo-sium on a World of Wireless, Mobile and Multimedia Networks & Workshops. Kos, Greece, 2009: 1-6.

[79] Challal Y, Benhamida Z, Nouali O. Scalable Key Management for Elastic Security Domains in Fog Networks[C]//IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises. Paris, France, 2018: 187-192.

[80] Zhang Y, Xu C, Li H, et al. PROTECT: efficient password-based threshold single-sign-on authentication for mobile users against perpetual leakage[J]. IEEE Transactions on Mobile Computing, 2020, 20(6): 2297-2312.

[81] Li J, Wu J, Chen L, et al. Blockchain-based secure key management for mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2021. DOI: 10.1109/TMC.2021.3068717

[82] Gennaro R, Goldfeder S, Narayanan A. Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security[C]//In-

ternational Conference on Applied Cryptography and Network Security. Cham: Springer, 2016: 156-174.

[83] Wu Q, Chen H, Li Z, et al. On a practical distributed key generation scheme based on bivariate polynomials[C]//7th International Conference on Wireless Communications, Networking and Mobile Computing. Wuhan, China, 2011: 1-4.

[84] Wang D, Li W, Wang P. Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks[J]. IEEE Transactions on Industrial Informatics, 2018, 14(9): 4081-4092.

[85] Kelly S, Frankel S. Using hmac-sha-256, hmac-sha-384, and hmac-sha-512 with ipsec. RFC 4868, May, 2007.

[86] Kokoris E, Malkhi D, Spiegelman A. Asynchronous Distributed Key Generation for Computationally-Secure Randomness, Consensus, and Threshold Signatures[C]//Proceedings of the 2020 ACM SIG-SAC Conference on Computer and Communications Security. Virtual Event USA , 2020: 1751-1767.

[87] Esposito C, Ficco M, Castiglione A, et al. Distributed group key management for event notification confidentiality among sensors[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 17(3): 566-580.

[88] Almeroth C, Ammar H. Collecting and modeling the join/leave behavior of multicast group members in the Mbone[C]//Proceedings of 5th IEEE International Symposium on High Performance Distributed Computing. Syracuse, NY, USA, 1996: 209-216.

[89] Almeroth C, Ammar H. Multicast group behavior in the Internet's multicast backbone (MBone)[J]. IEEE communications Magazine, 1997, 35(6): 124-129.

[90] Ballerini R. Probability, stochastic processes, and queueing theory: The mathematics of computer performance modeling[J]. Journal of the American Statistical Association, 1996, 91(434): 913-915..

[91] Wong K, Gouda M, Lam S. Secure group communications using key graphs[J]. IEEE transactions on networking, 2000, 8(1): 16-30.

[92] DeCleene B, Dondeti L, Griffin S, et al. Secure group communications for wireless networks[C]// MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force. McLean, USA, 2001, 1: 113-117.

[93] Wang D, Wang P, Lei M. Cryptanalysis and improvement of gateway-oriented password authenticated key exchange protocol based on RSA[J]. Acta Electronica Sinica, 2015, 43(1): 176.
(汪定, 王平, 雷鸣. 基于 RSA 的网关口令认证密钥交换协议的分析与改进. 电子学报, 2015, 43(1): 176.)

**JIANG Jingwei**, born in 1994, Currently, he is ursuing his Ph.D. de-gree at Harbin Engin-eering University, China. His primary research interests are in cryptography and information security. In particular, his research focus is cryptology and information security.

**WANG Ding,** born in 1985, received his Ph.D. degree in Information Security at Peking University in 2017. He is currently a Professor of College of Cyber Science, Nankai University, and also serves as the deputy director of the Tianjin key laboratory of Network and Data Security Technology. As the first/corresponding author, he has published more than 60 papers at venues like ACM CCS, USENIX Security, NDSS, IEEE DSN, ESORICS, ACM ASIACCS, ACM TCPS, IEEE TDSC and IEEE TIFS. Nine of them have been recognized as “ESI highly cited papers”. His Ph.D. thesis receives the “ACM China Doctoral Dissertation Award” and “China Computer Federation (CCF) Outstanding Doctoral Dissertation Award”. He has been involved in the community as a TPC member or PC Chair for over 60 international conferences such as ACM CCS 2022、PETS 2022, ACSAC 2021/2020, ACM AsiaCCS 2022/2021 、 ICICS 2018-2022, IFIP SEC 2018-2021, and INSCRYPT 2018-2020. His research interests focus on authentication protocol and password security.

**ZHANG Guo-yin**, born in 1962, Ph.D, Professor at Harbin Engineering University. His research interests include Network information security and embedded system.

**CHEN Zhi-yuan**, born in 1978. PhD. Associate professor at Harbin Engineering University. His main resea-rch interests include information and network security model checking and modal logic.

## Background

This paper studies two parts: firstly, a non-interactive transparent zero knowledge verifiable polynomial delegate is constructed, and based on this technology, a threshold secret sharing protocol is designed to protect the private key of MEC server; Then, this paper proposes a MEC network key management protocol to support users' free access and ensure network communication security, which is used to protect the communication and data security of mobile users in MEC network. Through the security analysis, efficiency analysis and comparison with other protocols, it shows that the key management protocol proposed in this paper achieves better security, higher efficiency and maintains good performance. It is proved that the key management protocol proposed in this paper is suitable for MEC network environment.

Our research group has devoted a lot of effort in password security and user authentication scheme. We have published some papers at prestigious venues including IEEE TDSC, computer networks, information science.

This research is partial supported by the National Natural Science Foundation of China under Grant No. 62172240, and by the National Key Research Program of China under Grant No.2018YFB08036-05