计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(11): 2871−2891, 2026

基于同意的隐私保护访问控制模型及 TLA+描述与验证

马　丽[1]　王昌晶[2]　章　琳[3]　甘丽新[3]

[1]（江西科技师范大学信息工程学院　南昌　330038）

[2]（江西师范大学人工智能学院　南昌　330022）

[3]（江西科技师范大学人工智能学院　南昌　330038）

（mali@jxstnu.edu.cn）

Consent-Based Access Control Model for Privacy-Preservation and Its TLA+ Specification and Verification

Ma Li[1], Wang Changjing[2], Zhang Lin[3], and Gan Lixin[3]

[1]（*School of Information Engineering*, *Jiangxi Science & Technology Normal University*, *Nanchang* 330038）

[2]（*School of Artificial Intelligence*, *Jiangxi Normal University*, *Nanchang* 330022）

[3]（*School of Artificial Intelligence*, *Jiangxi Science & Technology Normal University*, *Nanchang* 330038）

Abstract　To prevent malicious access to and misuse of privacy-sensitive data, data collectors must ensure users' explicit consent and autonomous control over the processing of their personal data, as mandated by modern data protection regulations. In existing approaches, users' consent is mostly bound directly to data processing purposes. However, existing approaches suffer from vague semantics in consent derivation, weak alignment with the operational logic of data processing, and a lack of effective conflict resolution mechanisms, making it difficult to obtain legally valid and practically enforceable consent. To address these issues, a consent-based access control model is proposed, in which consent permissions are incorporated into the access decision process, and all personal data operations are abstracted as actions serving specific purposes. At the semantic level, the relationship between purposes and data operations is clarified, and an integrated semantic linkage among purpose, action, and consent is established. Following the principles of least privilege and deny-by-default, an asymmetric consent inheritance rule and a consent computation method tailored to hierarchical purpose structures are defined, thereby enabling fine-grained and context-aware access decisions. The model is formally specified in TLA+ (temporal logic of actions plus), and its core security properties are verified through model checking, ensuring both soundness and consistency. Experimental results on representative consent scenarios demonstrate that, compared with existing representative methods, the proposed approach can effectively resolve consent conflicts, reduce unintended over-authorization, and achieve better decision quality and compliance.

Key words　consent permissions; purpose semantics; access control; model checking; TLA+

收稿日期：2025-09-02；**修回日期**：2026-06-29

基金项目：国家自然科学基金项目（62462037, 62262027, 62361030）；江西省高校人文社会科学研究项目（JC22115）；江西省教育厅科学技术研究项目（GJJ2201330）；江西省主要学科学术与技术带头人培养项目（20232BCJ22013）

This work was supported by the National Natural Science Foundation of China (62462037, 62262027, 62361030), the Jiangxi Provincial University Humanities and Social Sciences Research Project (JC22115), the Science and Technology Research Project of Jiangxi Provincial Department of Education (GJJ2201330), and the Cultivation Project for Academic and Technical Leader in Major Disciplines in Jiangxi Province (20232BCJ22013).

通信作者：王昌晶（wcj771006@163.com）

摘 要 为防范隐私数据被恶意访问与滥用,数据收集方需保障用户对个人数据处理的明确同意与自主控制权。现有方案大多将用户同意与数据处理目的直接绑定,存在同意推导语义模糊、脱离业务逻辑、缺少冲突消解机制等问题,无法形成合法有效的用户同意。针对上述问题,提出一种基于同意的访问控制模型,将同意权限纳入访问决策流程,将个人数据操作抽象为服务于特定目的的动作。模型从语义层面明确目的与数据操作的关联,搭建“目的—动作—同意”一体化语义链路;遵循最小权限与拒绝优先原则,设计适配目的层级结构的非对称同意继承规则及同意计算方法。采用 TLA+(temporal logic of actions plus)完成模型形式化建模,经模型检测验证了核心安全属性。实验表明,相较于现有代表性方法,该方法可有效化解同意冲突,取得更优结果。

关键词 同意权限;目的语义;访问控制;模型检测;TLA+

中图法分类号 TP309

DOI: 10.7544/issn1000-1239.202550653 **CSTR:** 32373.14.issn1000-1239.202550653

物联网、智慧城市与移动互联网技术的发展使得个人数据的收集和共享日益普遍。拥有更多的数据可以使企业获得更高的市场预测能力及竞争力,但同时也会给用户(或称数据主体)带来隐私侵犯、数据泄露的风险。隐私是个人决定自己的信息何时、如何以及在多大程度上向他人透露的权利[1]。仅依靠单一技术手段难以彻底解决隐私问题,完整的隐私保护体系通常需要法律法规、社会规范、市场机制与技术措施的综合作用[2]。近年来,欧盟《通用数据保护条例》(General Data Protection Regulation, GDPR)、美国《加州消费者隐私法》(California Consumer Privacy Act, CCPA)以及《中华人民共和国个人信息保护法》等隐私法规相继出台,明确强调“知情同意”“目的限制”“最小必要”等原则,要求在系统设计阶段就保障数据主体对个人数据的知情权与控制权[3-6]。因此,在保证数据主体权利的前提下,实现个人数据的安全、合规与高效利用,已成为隐私计算与数据治理领域的重要研究问题。

安全使用和共享个人数据的方法大致分为 2 类:一类通过数据匿名化[7]、差分隐私[8]、安全多方计算[9]、同态加密[10] 等技术,使共享方无法从共享数据中识别任何个人数据,从而不再适用隐私法规,同时确保敏感或个人数据免受未经授权的访问、披露或滥用。然而,此类技术需要额外的数据处理资源,并且由于信息数量和质量的损失,可能会降低数据的价值。另一类在保留原始数据用于高质量分析的前提下,通过访问控制与合规机制对“谁”“为何目的”“访问哪些数据”进行精细约束,以落实隐私法规对目的限制与同意的要求[11−21]。随着数据分析需求的增长以及隐私法规对“可解释、可撤回同意”的要求,如何在第 2 类路线中围绕“目的”与“同意”设计可证明、安全且可扩展的访问控制模型,成为研究焦点。

现有基于目的的访问控制与同意管理工作为细粒度隐私保护提供了重要思路,如基于目的的访问控制(purpose-based access control, PBAC)模型及其扩展[11−17]。一些工作将目的组织成层次结构并利用继承规则进行同意推理[11-12,18,22-24];另一些工作将同意以记录或区块链智能合约等方式纳入系统以支持审计与可追溯性[20,23,25]。然而,大多数现有方法在目的语义形式化、刻画目的间多样化关系,以及同意继承与冲突处理等方面存在以下不足:1)目的关系常被简化为单一的包含关系,忽略了目的与应用业务逻辑之间的关联,难以精确刻画概括与前提 2 类语义;2)在同意与拒绝的继承规则上多采用对称或单向向下传递,导致用户在表达允许哪些目的及禁止哪些目的时出现逻辑矛盾或非预期的“过度同意”或“过度拒绝”甚至是“全同意”或“全拒绝”结果[26];3)对用户动态隐私意愿(如撤销同意)缺乏系统化的更新与验证机制,难以保证长期运行下的安全性与合规性[27]。

为弥补上述不足,本文提出一种新颖的基于同意的访问控制(consent-based access control, CBAC)模型。CBAC 的设计目标是:在形式化层面精确刻画目的、动作与同意之间的语义关联;区分并形式化 2 类关键的目的关系,即概括关系与前提关系,并基于这 2 类关系给出非对称的同意及拒绝继承规则;将数据主体的同意权限与数据收集者的角色权限统一纳入同一访问判定框架,从而支持细粒度的同意计算、冲突消解与动态同意演化。

与以往工作相比,CBAC 在目的语义的形式化、同意继承规则的证明性处理及可验证实现上具有以下创新:

1)CBAC 为一种新的基于同意的访问控制模型。

在该模型中，从动作时序语义出发，将实现目的的具体动作视为达成目的的操作，并把“允许动作执行”的权限与数据主体的同意直接对应，构建目的—动作—同意的语义链路，从而使同意推理具有操作语义基础。

2）提出了新的同意继承理论。在动作时序逻辑框架下形式化定义概括关系与前提关系，证明 2 类关系下同意与拒绝的非对称继承定理，并基于最小权限与拒绝优先原则提出一致且可证明的同意计算与冲突消解方法。

3）使用形式化方法对 CBAC 模型进行了验证。给出 CBAC 的动作时序逻辑（temporal logic of actions plus, TLA+）规约并利用模型检测工具验证若干关键安全性质（包括同意与授权的双重判定、继承性与撤销同意传播性），同时通过与代表性方法的对比实验评估在真实业务语义下对用户隐私偏好的符合性。

1 相关工作

1.1 隐私访问控制模型

传统访问控制模型，如自主访问控制（discretionary access control, DAC）、强制访问控制（mandatory access control, MAC）、基于角色访问控制（role-based access control, RBAC）等，主要关注资源保护和职责分离，对隐私场景下的隐私权利、同意、处理目的之间的关联关系缺乏专门刻画，难以直接满足 GDPR 等隐私法规对“知情同意”“目的限制”等原则的要求。因此，近年的研究多在既有访问控制框架上叠加隐私约束，实现细粒度的个人数据访问控制。这些研究大致可分为基于 RBAC 的扩展模型和基于属性的访问控制（attributed-based access control, ABAC）的扩展模型 2 类。

在 RBAC 方向上，Agrawal 等人[1]引入“目的”概念，开启了数据库层面的隐私保护研究。Byun 等人[11-12]提出的 PBAC 模型成为后续工作的基础。此后大量研究在 RBAC 中引入目的约束，通过“角色—目的”绑定实现细粒度授权[13-17]。除此之外，Kebede 等人[18]从“目的限制”原则出发，形式化多种目的关系并据此推导基于目的的权限；Krstić等人[26]尝试将 GDPR 中的目的限制和同意概念与 RBAC 安全策略集成；Xue 等人[28]提出目的感知访问控制（purpose-aware access control, PAAC），将目的细分为数据处理目的和数据操作目的，并设计自动目的分析算法；Patil 等人[29]将目的与角色结合，在关系数据库中实现更精细的数据保护；Sultan 等人[30]将溯源信息与属性增强的 RBAC（attributes enhanced role-based access control, AERBAC）结合，通过溯源辅助目的限制。

上述基于 RBAC 的工作普遍将“目的”视为访问控制条件的一部分，隐私策略主要由数据收集者制定，数据主体的隐私意愿多以策略参数或配置形式被嵌入，而不是被单独建模为一种可计算、可验证的“同意权限”。在数据多方共享、目的结构复杂的场景下，难以保证数据主体的真实同意在各方访问决策中得到一致且可追踪的执行。

在 ABAC 方向上，Molina 等人[19]尝试将“处理目的”映射为主体属性，将“预期目的”建模为对象属性、将“访问目的”建模为环境属性，通过主体属性、对象属性与环境属性的组合实现细粒度控制。这类方法在策略表达能力和动态性方面更为灵活，但同样没有把数据主体可撤回、可细粒度表达的同意表示为访问控制中的权限，数据主体的同意决策与数据收集者的访问控制策略往往交织在统一的属性策略中，二者难以清晰解耦。

此外，也有工作从系统架构角度将“目的限制”和“同意管理”与访问控制框架结合。例如，Daoudagh 等人[4]针对智慧城市场景提出融合同意管理与访问控制的解决方案；Popoola 等人[20]在智能家居医疗环境下，通过在区块链上发布及订阅智能合约来记录用户的同意偏好；马丽等人[21]初步将“同意权限”融入 RBAC，并对基于用户同意的隐私保护协议进行形式化描述和验证；Hajszan 等人[27]提出细粒度共享策略，将同意嵌入数据层面的访问控制中，以请求定制化的数据子集替代完整文件访问。但这些工作要么缺乏目的形式化语义描述，要么未在统一模型中同时描述隐私角色、目的结构与数据主体隐私偏好，对“同意权限—角色权限—访问判定”的一体化建模和可验证性仍有提升空间。

与上述研究相比，本文提出基于同意的访问控制模型，将数据主体的同意权形式化为“同意权限”，并通过建立“隐私角色—目的”关系与数据收集者的角色权限相结合，在访问控制模型层面实现了双重授权，为目的语义建模和同意计算提供了统一基础。

1.2 同意计算

在隐私保护领域，同意计算大致可以分为粗粒度和细粒度 2 类。粗粒度方法多采用隐私政策文本或选择加入（opt-in）（或选择退出（opt-out））机制直接获取用户同意，再据此进行访问控制决策[31-35]。然而，隐私政策通常表述冗长复杂，用户难以真正理解其含义；数据主体对某项服务整体选择“同意”或“拒

绝”,既不能反映个性化隐私偏好,也容易引起“同意疲劳”[36-38]。为提升细粒度,近年来逐渐出现将“同意”与数据处理目的绑定,并借助目的层次结构进行同意计算的研究。

Harkous 等人[39]将 opt-in(或 opt-out)选择与具体目的绑定,将“同意”视为数据主体对数据处理目的的选择权,但未进一步分析目的之间的逻辑关系,难以基于目的语义进行系统的同意推理。Byun 等人[11-12]采用目的树结构,将父子关系解释为概括关系;虽然其未显式引入“同意权限”,但“允许目的”(或“禁止目的”)的设定可以看作对“同意”(或“拒绝”)的隐含表达。Byun 等人还将同意的继承定义为“允许可以由父目的向其子目的传递、禁止可以由父目的向其子目的及其父目的传递”。后续多篇基于目的树的工作[14-17]均采用了 Byun 等人的同意继承思路。然而,由于目的缺乏统一的形式化语义解释,这类单一父子关系下的继承规则在处理复杂冲突时容易产生自相矛盾的结果。

Kebede 等人[18]区分了 4 种目的关系(特殊、概括、前提条件、兼容),并尝试据此推导不同目的上的权限与可继承同意,但未对“拒绝”的继承与冲突处理给出系统机制。Hjerppe 等人[22]使用 is-a 与 has-a 关系表达目的层次,并提出规则:允许和禁止均由父目的向其子目的传递。该规则难以反映数据主体对某些子目的“单独拒绝”的细粒度隐私偏好。Ni 等人[13]、Robol 等人[24]和 Olca 等人[31]的工作也多采用“父目的同意或拒绝统一向下继承”的处理方式,未充分区分不同类型的目的关系。

从工程实现角度看,Park 等人[25]将自然语言同意信息转化为 0/1 标志,通过矩阵运算实现批量匹配,其同意来源主要是初始勾选,不考虑目的间关系及隐含同意挖掘;Daudén-Esmel 等人[23]基于区块链设计“收集同意”和“处理同意”智能合约,通过目的层次将父目的同意或撤销同意传递给子目的,采用统一向下继承策略;Konstantinidis 等人[37]允许数据主体用“同意约束”描述不允许的数据共享组合,侧重查询层面的自动过滤;Sassetti 等人[40]通过隐私感知同意页面实现用户主动授权,让用户选择允许哪些数据使用目的、共享方等信息;Rohini 等人[41]在医疗场景中采用目的树管理处理目的,认为同意父目的即可推定对所有子目的同意,同意继承方式类似于 Kebede 等人[18]的方法。上述方案在同意记录、传播和审计方面具有实践价值,但在区分不同类型的目的关系、统一刻画目的—动作—同意的语义关联,以及在形式化框架下验证同意继承与冲突处理机制等方面仍相对薄弱。

已有工作中,也有尝试将“同意”更直接地写入模型结构。例如马丽等人[21]提出将同意权限融入 RBAC,但由于未考虑目的结构带来的同意继承问题,对同意权限的授予过于简单,无法解决同意冲突问题。

总体来看,现有基于目的的同意管理和访问控制方法在以下方面存在不足:其一,在描述目的关系时缺乏对概括关系和前提条件关系的系统区分,同意与拒绝的继承语义模糊,容易引发“全拒绝”或“全同意”等非预期同意结果;其二,很少将数据主体的同意权形式化为“同意权限”,并与数据收集者的角色权限在同一访问控制模型中加以统一建模,难以在模型层面清晰刻画基于角色和基于同意的双重授权结构;其三,缺乏在形式化框架下对同意继承、冲突处理以及撤销同意传播等动态性质的系统验证,难以保证系统在长期运行和复杂共享场景中持续满足安全与合规要求。为此,本文提出了 CBAC 模型,给出了上述问题的解决方案。

2 预备知识

TLA+是一种由 Lamport 开发的、适于描述并发与分布式系统的形式化规约语言[42]。TLA+将系统建模为状态机,其中一个状态机由它可能的初始状态(initial states)与一组动作(actions)来刻画。一个状态是对所有变量的一种赋值。一个动作表达了新、旧状态之间的某种关系(relation),可用包含带撇变量(表示新状态)与不带撇变量(表示旧状态)的公式来描述。本节只将本文用到的 TLA+的语法和含义简单地列举如表 1 所示。具体内容可以参见文献 [42]。

TLC 是遍历 TLA+规约有穷实例的状态空间以验证规约正确性的模型检测工具。TLC 可以处理如下标准格式的规约:$Init \wedge \Box[Next]_{vars} \wedge Temporal$,其中 $Init$ 表示初始谓词,$Next$ 表示后继状态动作,$vars$ 表示所有变量的元组,$Temporal$ 表示活性条件的时态公式。

3 基于同意的访问控制模型

3.1 系统架构

本文先从系统层面对基于同意的访问控制所包含的功能组件、信息流与运行流程进行说明,为后续的形式化建模与验证提供工程背景。

Table 1 Relevant Syntax Symbols in the TLA+

表 1 TLA+中相关语法符号

类别	语法符号	含义
集合	SUBSET S	集合 S 的子集集合
函数	$[x \in S \mapsto e]$	函数 f，使得对应 $x \in S$，$f[x] = e$（其中 e 为表达式）
	$[f \text{ EXCEPT } ![e_1]=e_2]$	除了 $f[e_1] = e_2$ 之外，函数 f 等于函数 f
记录	$e.h$	记录 e 的 h 字段
	$[h_1 \mapsto e_1, \cdots, h_n \mapsto e_n]$	h_i 字段为 e_i 的记录
	$[h_1 : S_1, \cdots, h_n : S_n]$	h_i 字段属于 S_i 的所有记录
	$[r \text{ EXCEPT } !.h=e]$	除了 $\hat{r}.h = e$ 之外，函数 $\hat{r}$ 等于函数 r
元组	$e[i]$	元组 e 的第 i 个元素
	$\langle e_1, \cdots, e_n \rangle$	第 i 个元素为 e_i 的 n 元组
构造	IF p THEN e_1 ELSE e_2	如果 p 为真，则为 e_1，否则为 e_2
	CASE $p_1 \to e_1 \square \cdots \square p_n \to e_n$	多分支选择结构，若某个 p_i 为真，则为 e_i
	LET $d_1 \triangleq e_1 \cdots d_n \triangleq e_n$ IN e	给出 IN 后 e 的某局部变量、函数（或常量）的序列定义，返回 e
动作运算符	e'	动作执行到新状态下 e 取值
	$[A]_e$	$[A \vee (e' = e)]$（其中 A 表示动作）
	UNCHANGED e	$e' = e$
时态运算符	$\square p$	p 总是为真
	$\diamond p$	p 最终为真

整个访问控制系统包含 3 类核心管理器：访问控制器(access controller, AC)、同意决策器(consent decider, CD)和授权决策器(authorization decider, AD)。其总体交互流程如下：

1)AC 作为策略实施点，拦截外部访问请求，收集请求属性(主体标识符、请求角色、请求目的、数据标识符等)，并按需向 CD 和 AD 发起决策请求。

2)CD 维护并计算数据主体的同意目的集合(以同意目的图或同意目的库的形式)，在接收到 AC 的同意查询时返回是否同意。CD 同时负责接收用户隐私偏好的更新并重新计算同意目的。

3)AD 依据角色-目的映射关系判断请求者是否具有基于角色的访问授权。AD 与 CD 的结果共同决定最终访问决策。

CD 的作用如图 1 所示：收集数据主体(即用户)的隐私偏好并保存在一个专门的数据库(隐私偏好库)中，并根据数据收集者提供的目的库计算出数据主体的同意目的库，以便对访问请求做出同意判断；当用户更新自己的隐私偏好时，CD 重新计算同意目的，并对同意目的库进行更新。

隐私数据访问控制流程如图 2 所示。隐私数据的请求者在获得访问权限之前提出访问请求，明确告知访问的数据类型以及访问目的是什么。AC 收到请求之后(步骤①)，将根据请求者的角色以及请求的目的，查询隐私数据副本库(步骤②)，获取数据主体的标识符(identity, ID)(步骤③)；AC 再向 CD 发出同意请求(步骤④)，CD 在收到请求之后查询该 ID 对应的同意目的库(步骤⑤)，收到同意目的集合(步骤⑥)后，CD 将结果反馈给 AC(步骤⑦)；如果反馈的结果是同意，则 AC 继续向 AD 发出授权请求(步骤⑧)，否则直接拒绝请求者(步骤⑫)，流程结束；AD 根据访问者的角色查询角色-目的映射库(步骤⑨)，获得该角色可访问的目的集合(步骤⑩)；AD 进一步判断访问目的是否在可访问目的范围内，将结果反馈给 AC(步骤⑪)；如果 AD 判断的结果是可授权的，则 AC 接受访问请求，否则拒绝请求者(步骤⑫)，流程结束。

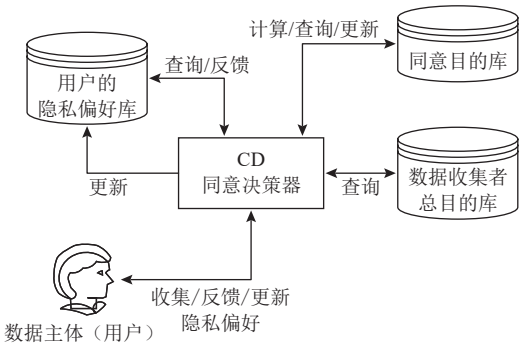


Fig. 1 Consent decider

图 1 同意决策器

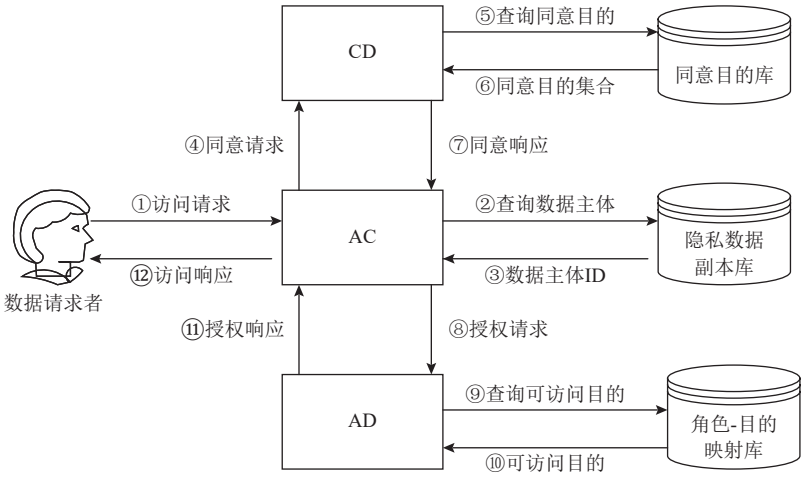


Fig. 2 Privacy data access control process

图 2 隐私数据访问控制流程

图 1 与图 2 分别展示了同意决策器的内部职责及隐私数据访问控制的总体流程。

为支持实际部署, 整体架构可与可扩展访问控制标记语言(extensible access control markup language, XACML)[43-44]集成: AC 对应 XACML 的策略实施点(policy enforcement point, PEP), CD 与 AD 合并实现策略决策点(policy decision point, PDP)功能, 策略管理点(policy administration point, PAP)和策略信息点(policy information point, PIP)提供策略与属性源。上述架构描述侧重功能与信息流, 3.2 节将基于该架构给出 CBAC 模型定义, 以便进行理论推导与模型检测。

3.2 模型语法

在系统架构基础上, 本节对 CBAC 模型的要素、动作-目的语义、目的关系以及同意/拒绝的继承性质给出形式化定义, 为后续 TLA+规约及模型检测提供理论基础。

本节以抖音隐私政策[45]为例, 将该政策的收集目的以及收集者的角色进行了分析和刻画。由于该政策文本过于复杂, 本节只给出了部分目的以及角色与目的之间的对应关系, 如图 3 所示。本文将收集目的之间因为业务逻辑构建的逻辑关系以层次结构表示。每个数据处理者都被分配一个具体的角色, 例如内部员工、广告商、安全服务商等, 且这些角色都与 1 棵子目的树对应。

角色-目的分配关系表示了数据提供者(即数据主体)对隐私数据处理者的授权, 其中每个角色可以分配多个目的, 这些目的可以构成 1 个或多个目的子树。如果该角色下的处理者又将获得的数据副本与其他应用实体(如第三方、合作方)共享, 则该角色

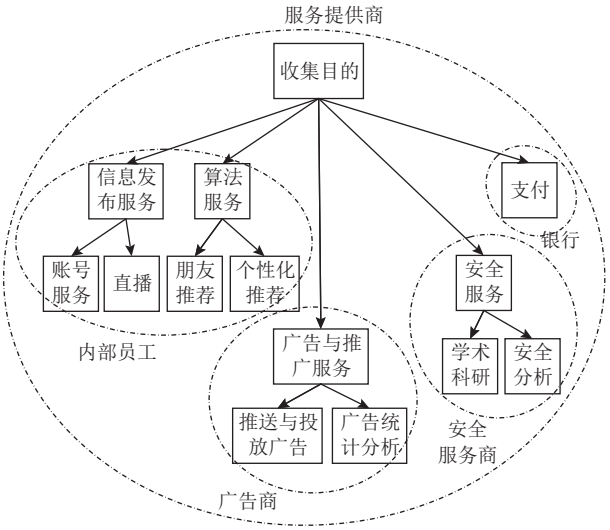


Fig. 3 An example of the correspondence between partial roles and purpose trees in Douyin's privacy policy

图 3 抖音隐私政策中部分角色与目的树对应举例

充当了数据提供者, 其他应用实体充当了数据处理者。因此, 数据提供者和数据处理者构成了 1 个角色层次, 其下层的角色(即数据处理者)可以访问的目的只能是其上一级角色(即数据提供者)目的树的子集。这样无论这些角色如何与其他应用实体共享, 其可访问的目的均被限制在有限的范围内。

如图 4 所示, 假设目的集合为 E={"信息发布", "广告与推广", "支付", …, "财付通"}, 则图 4 所示角色"服务提供商"所分配的目的集合即为 E, 其"内部员工"所分配的目的集合为 E_1, 满足 $E_1 \subseteq E$, 其下级角色"内容审核员"和"客服主管"所分配的目的集合分别为 E_1' 和 E_1'', 满足 $E_1' \subseteq E_1$ 及 $E_1'' \subseteq E_1$。

定义 1. CBAC 模型定义。

$$\mathcal{M} = (U, R, E, Ob, Op, P, RH, RA, EH, UA, UD, pur), \quad (1)$$

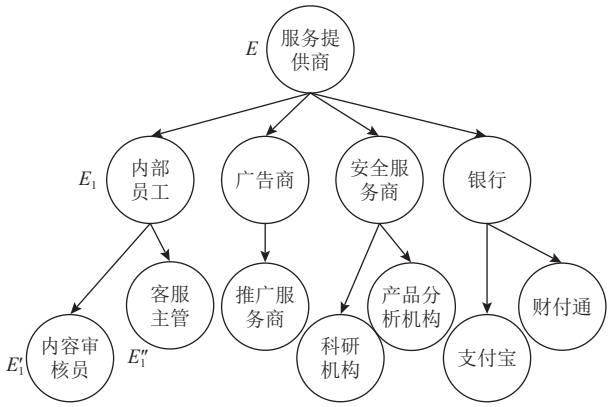


Fig. 4 An example of partial role hierarchy structure in Douyin's privacy policy

图 4 抖音隐私政策中部分角色层次结构举例

其中：U 为用户的集合；R为隐私角色集合；E 为目的集合，且 $E \subseteq N \times D$，N 表示目的名称集合，D 表示隐私数据集合；Ob 表示对象的集合，$U \subseteq Ob$；Op 表示隐私操作的集合，例如读（read）、写（write）和删除（delete）等；$P \subseteq U \times Op \times E$表示权限的集合，且 $P = P_C \cup P_R$，$P_C \subseteq P$表示同意权限集合，下标 C 表示同意（consent），$P_R \subseteq P$表示角色权限集合，下标 R 表示角色（role）；$RH \subseteq R \times R$表示隐私角色之间的层次关系，对 $r, r' \in R$，如果 $(r,r') \in RH$，则 r 比 r'高级，也可用 $r \geqslant r'$表示；$RA \subseteq R \times E$表示隐私角色与目的的分配关系；$EH \subseteq E \times E$表示目的之间的层次关系，对 $e, e' \in E$，如果 $(e,e') \in EH$，则 e比 e'高级，也可用 $e \geqslant e'$表示；$UA \subseteq U \times R$表示用户和隐私角色的分配关系；$UD \subseteq U \times D$表示用户与隐私数据之间的拥有关系；$pur: R \to 2^E$表示隐私角色所分配的目的集合，对隐私角色 $r \in R$，$pur(r)$表示 r 所分配的目的集合。

CBAC 模型中的隐私角色 R可以分为数据主体（data subject, DS）、数据收集者（data collector, DC）和数据处理者（data processor, DP）三类。对用户 u，在应用系统中的角色可能是 DP，但当有其他用户希望共享用户 u 的数据副本时，u 又充当 DC 角色；$pur(r)$反映了通过用户隐私角色获得的可访问目的。在具体实现时，这些角色和目的既可以按 RBAC 方式直接建模，也可以在属性驱动的策略语言（如 XACML）中通过主体、对象、环境属性来编码，但这并不改变 CBAC 在模型层面对“角色权限+同意权限+目的语义”的统一抽象。

定义 2. 给定 CBAC 模型 $\mathcal{M}$，对 $u \in U$，$op \in Op$及 $e \in E$，用 $p = (u, op, e)$表示权限，则有：

1）如果 $p \in P_C$，则表示 u 基于目的 e 执行操作 op 是同意授权的，简称 op 是同意的，用公式 $c_A(op)$和 $c_E(e)$分别表示动作 op 和目的 e 都是同意的（其中 A 表示动作集合）；

2）如果 $p \in P_R$，则表示 u 基于目的 e 执行操作 op 是角色授权的，简称 op 是角色授权的；

3）如果同时满足 $p \in P_C$且 $p \in P_R$，则有 $p \in P$，即 u 基于目的 e 执行 op 是授权的，简称 op 是授权的；否则，称 op 是非授权的。

定义 3. 已知 CBAC 模型 $\mathcal{M}$，访问请求 $q \in Q$表示为 $q = (u, r, e)$，其中 $u \in U$表示请求者，$r \in R$表示请求者的隐私角色，$e \in E$表示用户 u 的访问目的，且 $e = (n, d)$，其中 $n \in N$为访问目的名，$d \in D$表示请求访问的数据。用 Q 表示请求的集合。

定义 4. 令 $A = \{a, b, \cdots\}$为动作集合，其中 $Op \subseteq A$，动作的组合可以定义为

$$\alpha, \beta ::= a \mid (\alpha;\beta) \mid (\alpha + \beta) \mid (\alpha \mid \beta) \mid \varepsilon, \tag{2}$$

其中：a 表示原子动作；$(\alpha;\beta)$表示动作 α 和 β 的顺序执行；$(\alpha+\beta)$表示 α 和 β 的非确定性选择执行；$(\alpha|\beta)$表示 α 和 β 同时执行；ε表示空动作。

定义 5. 令 Φ为公式的集合，给定 CBAC 模型 $\mathcal{M}$及动作 $\alpha \in A$，状态公式 $\varphi \in \Phi$定义如下：

$$\begin{aligned}\varphi ::= {} & f \mid user(u) \mid data(d) \mid role(r) \mid purpose(e) \mid req(q) \\ & \mid pms(p) \mid pms_C(p) \mid pms_R(p) \mid ud(u,d) \\ & \mid ua(u,r) \mid rh(r,r') \mid ra(r,e) \mid eh(e,e') \\ & \mid c_E(e) \mid d_E(e) \mid c_A(\alpha) \mid d_A(\alpha) \\ & \mid Consent(q) \mid Refuse(q) \mid Auth(q) \mid Deny(q) \\ & \mid Accept(q) \mid Reject(q) \\ & \mid \neg\varphi \mid \varphi \to \varphi \mid \varphi \wedge \varphi \mid [\alpha]\varphi \mid \Box\varphi \mid \Diamond\varphi \mid \bot,\end{aligned} \tag{3}$$

其中：f 表示描述系统中一般对象的原子公式；公式 $user(u)$表示用户 $u \in U$；$data(d)$表示 $d \in D$；$role(r)$表示 $r \in R$；$purpose(e)$表示 $e \in E$；$req(q)$表示 $q \in Q$；$pms(p)$表示 p 为权限，即 $p \in P$；$pms_C(p)$表示 p 为同意权限，即 $p \in P_C$；$pms_R(p)$表示 p 为角色权限，即 $p \in P_R$；$ud(u,d)$表示 $(u,d) \in UD$；$ua(u,r)$表示 $(u,r) \in UA$；$rh(r,r')$表示 $(r,r') \in RH$；$ra(r,e)$表示 $(r,e) \in RA$；$eh(e,e')$表示 $(e,e') \in EH$；对 $e \in E$，$c_E(e)$表示同意目的 e，$d_E(e)$表示 $\neg c_E(e)$，即拒绝 e；$c_A(\alpha)$表示同意执行动作 α，$d_A(\alpha)$表示 $\neg c_A(\alpha)$，即拒绝执行动作 α；对请求 $q \in Q$，$Consent(q)$和 $Refuse(q)$分别表示同意请求 q 和拒绝请求 q，$Auth(q)$和 $Deny(q)$分别表示角色授权 q 和角色拒绝 q，$Accept(q)$和 $Reject(q)$分别表示接受和拒绝访问请求 q；$[\alpha]$表示动作 α执行的模态算子；$[\alpha]\varphi$表示执行动作 α后 φ为 TRUE；$\Box\varphi$表示 φ一直为 TRUE；$\bot$表示常量 FALSE。

符号“$\vee$”“$\to$”也在本文中使用，与经典逻辑含义相同。$\Diamond\varphi$为 $\neg\Box\neg\varphi$的缩写形式。

为了后面的证明需要,本文引用关于动作的部分公理[46]:

公理 1. $[\alpha;\beta]\varphi \equiv [\alpha]([\beta]\varphi)$。

公理 2. $[\varepsilon]\varphi \equiv \varphi$。

根据公理 1 和公理 2 可知,对任意动作 β 有 $[\beta;\varepsilon]\varphi \equiv [\varepsilon;\beta]\varphi \equiv [\beta]\varphi$。

在实际业务系统中,相同动作名在不同业务语义下,往往对应不同的执行前提条件与执行效果;若未依据动作拟达成的目的对同名动作做语义区分,会在同意推理过程中引发语义歧义。以基础操作 read 为例,该动作在各类业务场景中用以实现截然不同的目的:当执行 read 的目的为“内容浏览”时,该动作指代调取文件并展示内部数据,执行前提为用户完成登录身份核验且持有对应资源的查看权限,执行效果为文件内容渲染呈现于交互界面;当执行 read 的目的为“对外数据共享”时,该动作特指为向第三方分发数据而开展资源读取操作,执行前提为获取用户许可或满足合作合同中的合规约定,执行效果为原始数据向第三方主体完成传输交付。

为消除“同名动作但语义不同”引起的二义性,并为目的—动作—同意建立操作语义基础,本文将动作视为带属性的实体。对任意 $\alpha \in A$,引入状态公式 $Pre(\alpha)$ 和 $Post(\alpha)$,分别表示动作 α 的执行前提条件与执行效果。前提 $Pre(\alpha)$ 刻画动作在执行前必须满足的原子条件(包括触发来源、调用者角色、环境约束,以及“目的作为前提”的描述);效果 $Post(\alpha)$ 则刻画动作执行完成后必然成立的目的相关断言。形式上以模态规则表示为

$$Pre(\alpha) \to [\alpha]Post(\alpha)。 \tag{4}$$

在上述约定下,同名动作若在前提或效果上存在差异,可视为不同的动作。

定义 6. 令 L 为由原子公式及其否定构成的文字(literal)集合。定义映射 $LIT: \Phi \to 2^L$,将任意状态公式 $\varphi \in \Phi$ 映射为其包含的原子文字集合。对任意 $\alpha \in A$,记 $LIT(Pre(\alpha))$ 与 $LIT(Post(\alpha))$ 分别为 $Pre(\alpha)$ 与 $Post(\alpha)$ 的文字集。该映射用于后续对前提及效果是否包含某目的特征断言的可判定性检验。

定义 7. $\forall e \in E$,用 $\lambda(e)$ 表示 e 达成的特征原子断言。对 $\alpha \in A$, $e \in E$, α 的执行是为了达成 e,记为 $For(\alpha,e)$,当且仅当存在关于 α 的规则 $Pre(\alpha) \to [\alpha]Post(\alpha)$ 且满足 $\neg\lambda(e) \in LIT(Pre(\alpha))$ 和 $\lambda(e) \in LIT(Post(\alpha))$。

直观含义:$\lambda(e)$ 在执行 α 前为 FALSE,执行后为 TRUE; $For(\alpha,e)$ 成立表示动作 α 的执行会导致目的 e 的达成(e 可作为 α 的后置目的)。

若 $\lambda(e) \in LIT(Pre(\alpha))$,则 e 可被视为 α 的前提目的(即动作执行依赖于 e 已达成)。

上述建模使得同名动作通过不同的前提条件和执行效果去二义性,为目的作为前提或效果提供了形式基础,防止出现一个动作达成多种目的的情况。因此,本文只讨论 1 个动作达成唯一一个目的的场景,即 $LIT(Post(\alpha)) = \{\lambda(e)\}$。

定义 8. 给定 CBAC 模型 $\mathcal{M}$,对目的 $e, e' \in E$:

1)如果存在动作 α,满足 $For(\alpha,e')$,且有 $Pre(\alpha) \to [\alpha]\lambda(e')$ 且 $\lambda(e) \in LIT(Pre(\alpha))$,则称 e 为 e' 的前提目的,或 e' 称为 e 的依赖目的,用 F 表示前提关系集合,则 $(e',e) \in F$,用公式 $f_F(e',e)$ 表示,且 $F \subseteq EH$;

2)如果 e 的达成蕴含了 e' 的达成,即满足公式 $\lambda(e) \to \lambda(e')$,则称 e' 为 e 的概括目的,或 e 为 e' 的特化目的,用 G 表示概括关系集合,则 $(e',e) \in G$,用公式 $f_G(e',e)$ 表示,且 $G \subseteq EH$, $G \cap F = \varnothing$。

e' 为 e 的概括目的时,根据定义 8, e' 的达成相当于在 e 的达成基础上执行了空动作 ε,则可以认为存在关于 ε 的规则 $\lambda(e) \to [\varepsilon]\lambda(e')$。当 e' 是多个特化目的 $e_1, e_2, \cdots, e_k (k \geqslant 2)$ 的概括目的时,用规则 $\lambda(e_1) \vee \lambda(e_2) \vee \cdots \vee \lambda(e_k) \to [\varepsilon]\lambda(e')$ 描述空动作 ε。并且,当 $k \geqslant 2$ 时,称 e' 为复合目的;当 $k = 1$ 时, e' 为原子目的。

如果目的 e' 没有特化目的,且有 0 个或 n 个前提目的 $e_1, e_2, \cdots, e_n (n \geqslant 1)$,则 e' 也称为原子目的。

具备 F 关系和 G 关系的目的之间构成了一条有向边。这样目的关系可以形成一种图结构,我们称之为目的图(purpose graph, PG),用 $PG = (E, R_E)$ 表示,其中 E 表示目的结点集合, $R_E \subseteq E \times E$ 表示连接目的结点有向边集合,且有 $R_E = R_F \cup R_G$,其中 R_F 表示 F 关系组成的边, R_G 表示 G 关系组成的边。对 $e_1, e_2 \in E$,如果 e_1 是 e_2 的依赖目的,则有一条由 e_1 指向 e_2 的有向边,即 $(e_1,e_2) \in E_F$,则 $f_F(e_1,e_2)$ 成立;如果 e_1 是 e_2 的概括目的,则有一条由 e_1 指向 e_2 的有向边,即 $(e_1,e_2) \in E_G$,则 $f_G(e_1,e_2)$ 成立。没有父结点的目的结点称为总目的(general purpose, GP)结点,没有子结点的目的结点称为叶子(leaf)结点。总目的为复合目的,叶子结点为原子目的。

例如,图 5 简化显示了抖音平台部分收集目的图,其中方框内仅含目的名称(省略了收集的数据类型),并且在有向边上标注了 G 和 F 关系。在抖音平台上,“直接登录”和“刷脸登录”因为动作和收集的隐私数据都不同,是不同的目的,但无论哪种目的达成,系统都会给用户分配一个 ID 号以最终实现“登录”,因此“登录”是“直接登录”和“刷脸登录”的概括目的。

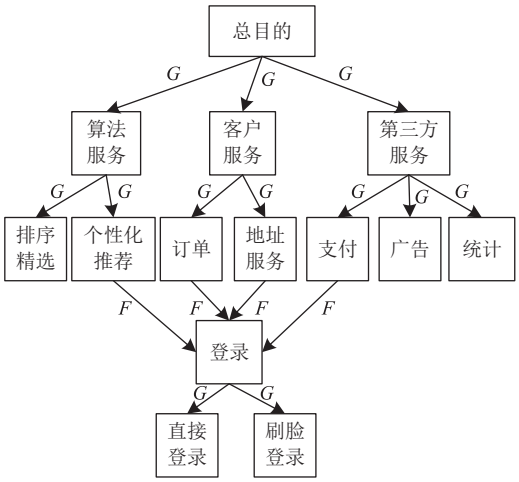


Fig. 5 Example of Douyin's privacy policy purpose graph

图 5 抖音隐私政策目的图示例

并且“刷脸登录”只以采集用户的脸部信息为前提条件，不需要以其他目的为前提，因此“刷脸登录”为原子目的。类似地，“直接登录”也是原子目的。同时，图 5 还显示，“登录”为“个性化推荐”“订单”“地址服务”“服务升级”的前提目的，用公式表示为 f_F(“个性化推荐”,“登录”)、f_F(“订单”,“登录”)、f_F(“地址服务”,“登录”) 和 f_F(“支付”,“登录”)，其余的父子目的构成了概括关系。

本文假设数据收集者的所有收集信息的目的都以目的图的形式表示出来。数据收集者在收集个人数据时会向数据主体明确收集目的(即收集及处理数据的意图)。数据主体可以根据这些目的提出允许目的(allowed intended purpose, AIP)和禁止目的(prohibited intended purpose, PIP)这 2 种类型的隐私偏好[11-12]，这样可以让数据主体更准确地把握自己的意图。并且当一个目的既为允许目的又为禁止目的时，采取禁止目的优先级高于允许目的原则，以解决产生的冲突。给定 CBAC 模型，数据主体的隐私偏好用二元组 (AIP, PIP) 表示，其中 AIP 表示允许目的集合，PIP 表示禁止目的集合，且有 $AIP \subseteq E$，$PIP \subseteq E$。当数据主体提交 (AIP, PIP) 之后，我们可以结合目的图自动推理出数据主体的(consent purpose graph, CPG)，而这样的 CPG 就是数据收集者必须遵守和维护的数据主体隐私政策。

3.3 模型语义

本节给出 CBAC 的语义模型，在此模型基础上进一步解释同意与目的之间的语义关系。

定义 9. 已知 CBAC 模型 $\mathcal{M}$，CBAC 的语义模型

$$M = (S_0, S, A, \delta, \{R_\alpha\}, \{C_\alpha\}, \{D_\alpha\}), \tag{5}$$

其中：S_0 为初始状态集合；S 为非空的状态集合，该集合由描述 CBAC 的模型 $\mathcal{M}$ 中元素的断言组成，其中 $\forall e \in E$，存在未达成 e 的状态 US_e 和达成 e 的状态 S_e 这 2 类状态，且 $US_e \subseteq S$，$S_e \subseteq S$，其中 $\neg\lambda(e) \in US_e$，$\lambda(e) \in S_e$；A 表示动作的集合；$\forall \alpha \in A$，$R_\alpha \subseteq S \times S$ 表示动作 α 上的可达关系；对任意的 $s \in S$，存在至多 1 个状态 $s' \in S$ 满足 $(s, s') \in R_\alpha$，记为 $R_\alpha(s, s')$；$\delta: S \times \Phi \to$ {TRUE,FALSE} 表示赋值函数，对任意的状态 $s \in S$ 和公式 $\varphi \in \Phi$，$\delta(s, \varphi) =$ TRUE 表示将真值赋值给 s 中的 φ，也可记为 $s \vDash \varphi$；对状态序列 s_0，s_1，…，s_i，…，s_j，…，有 $s_i \vDash \Box\varphi$ 当且仅当 $\forall s_j (j \geqslant i)$ 有 $s_j \vDash \varphi$；$s_i \vDash \Diamond\varphi$ 当且仅当 $\exists s_j (j \geqslant i)$，$s_j \vDash \varphi$；$s_i \vDash \varphi \to [\alpha]\varphi'$ 当且仅当 $s_i \vDash \varphi \& (\exists s_j \in S)(j \geqslant i)((s_i, s_j) \in R_\alpha \Rightarrow s_j \vDash \varphi')$，其中 φ 和 φ' 均为不含模态算子的状态公式；$C_\alpha \subseteq S \times S$ 为动作 $\alpha \in A$ 的同意状态迁移关系；$D_\alpha \subseteq S \times S$ 为动作 $\alpha \in A$ 的拒绝状态迁移关系；有 $C_\alpha \cap D_\alpha = \varnothing$。

对每个动作 $\alpha \in A$，我们规定 R_α 为自反和传递的。关系 C_α 和 D_α 是对状态迁移关系的一种约束，其中 $C_\alpha(s, s')$ 说明从 s 到 s' 的直接迁移关系是被同意的，而 $D_\alpha(s, s')$ 说明 s 到 s' 的直接迁移关系是被拒绝的。

φ 在模型 M 中为真，如果对模型 M 中每个 $s \in S$ 为真，记为 $M, s \vDash \varphi$，φ 是有效的(记为 $M \vDash \varphi$)；φ 在每个模型 M 中都为真，如果存在 M 和 $s \in S$，有 $M, s \vDash \varphi$，则 φ 是可满足的。

这里进一步给出定义 7 的语义解释。给定模型 M，对动作 $\alpha \in A$ 和目的 $e \in E$，e 的特征断言 $\lambda(e)$ 及 α 的规则 $Pre(\alpha) \to [\alpha]Post(\alpha)$，则 $M, s \vDash For(\alpha, e)$ 当且仅当 $\exists (s, s') \in R_\alpha$ 且满足：

1) $s \in US_e$ 且 $s' \in S_e$，其中 $\neg\lambda(e) \in US_e$，$\lambda(e) \in S_e$，或者

2) $M, s \vDash Pre(\alpha)$ 且 $M, s' \vDash Post(\alpha)$，其中 $\neg\lambda(e) \in LIT(Pre(\alpha))$，$LIT(Post(\alpha)) = \{\lambda(e)\}$。

下面遇到动作规则描述的时候，我们采用 2) 的解释。

定义 10. 给定模型 M，动作 $\alpha \in A$，以及 α 实现的目的 $e \in E$，有：

1) $M, s \vDash c_A(\alpha)$ 当且仅当 $(\exists s' \in S)\ (C_\alpha(s, s') \& R_\alpha(s, s'))$；

2) $M, s \vDash d_A(\alpha)$ 当且仅当 $(\exists s' \in S)\ (R_\alpha(s, s')) \& (\forall s' \in S)(R_\alpha(s, s') \Rightarrow D_\alpha(s, s'))$；

3) $M, s \vDash c_E(e)$ 当且仅当 $(\exists \alpha \in A)\ (For(\alpha, e) \& M, s \vDash c_A(\alpha))$；

4) $M, s \vDash d_E(e)$ 当且仅当 $(\exists \alpha \in A)\ (For(\alpha, e)) \& (\forall \alpha \in A)(For(\alpha, e) \Rightarrow M, s \vDash d_A(\alpha))$。

注：$c_A(\alpha)$ 和 $d_A(\alpha)$ 也可以等价地解释为：

1) $M,s\vDash d_A(\alpha)$当且仅当$\exists W\subseteq S$, $M,s\vDash[\alpha]W$, 即在状态s下拒绝执行α, 当且仅当, α的执行会迁移到“违反了数据主体的隐私保护意愿”的状态, 用W来表示这些状态的集合;

2) $M,s\vDash c_A(\alpha)$当且仅当$M,s\vDash\neg d_A(\alpha)$, 即状态$s$下同意执行动作$\alpha$, 当且仅当, 该动作没有被拒绝。

3.4 同意继承性

在传统的RBAC模型中, 角色层次关系蕴含着用户角色以及权限的继承关系, 类似地, CBAC模型中目的层次结构也蕴含着同意的继承关系。从定义10可知, 对动作$\alpha\in A$及目的$e\in E$, CBAC的语义模型M及任意的状态s, 如果$M,s\vDash For(\alpha,e)$, 则动作的同意授权与目的同意是等价的。接下来我们通过目的层次上目的同意继承性质来给出同意权限的继承性质。

定理 1. G关系继承性。给定CBAC的语义模型M, 对目的e和e', 任意状态$s\in S$, 如果$M,s\vDash f_G(e',e)$, 则

1) $M,s\vDash c_E(e)\rightarrow c_E(e')$;

2) $M,s\vDash d_E(e')\rightarrow d_E(e)$。

证明.

1)根据定义10可知$M,s\vDash c_E(e)\Rightarrow(\exists\alpha\in A)(For(\alpha,e)\&M,s\vDash c_A(\alpha))$, 且 $M,s\vDash c_A(\alpha)\Rightarrow(\exists s'\in S)(C_\alpha(s,s')\&R_\alpha(s,s'))$。

假设关于α的规则为$Pre(\alpha)\rightarrow[\alpha]\lambda(e)$且$\neg\lambda(e)\in LIT(Pre(\alpha))$, 则对状态对$(s,s')$满足$C_\alpha(s,s')$和$R_\alpha(s,s')$, 有$M,s\vDash\neg\lambda(e)$及$M,s\vDash\neg\lambda(e')$(因为在状态$s$下实现目的$e$和$e'$的动作还未执行), 且$M,s'\vDash\lambda(e)$。根据定义8可知$\lambda(e)\rightarrow\lambda(e')$, $\forall w\in S$, $M,w\vDash f_G(e',e)$推断出如果$M,w\vDash\lambda(e)$, 则$M,w\vDash\lambda(e')$。因此对状态s'有$M,s'\vDash\lambda(e')$。

针对目的e', 我们可以构造一个动作β满足$For(\beta,e')$, 因为目的e和e'有G关系, 因此β可以定义为$\beta=\alpha;\varepsilon$(基于公理1和公理2, 空动作的执行不改变状态公式的值)。则对β有$(s,s')\in C_\beta$, 则有$M,s\vDash\neg\lambda(e')$且$M,s'\vDash\lambda(e')$, 根据前面的推论, 状态对(s,s')满足$C_\alpha(s,s')$和$R_\alpha(s,s')$, 且由于状态s'不会被ε改变, 因此也有$C_\varepsilon(s',s')$和$R_\varepsilon(s',s')$成立, 这样有$C_{\alpha;\varepsilon}(s,s')$和$R_{\alpha;\varepsilon}(s,s')$成立, 因此根据定义10, 有$M,s\vDash c_A(\beta)$, 则有$M,s\vDash c_E(e')$。

2)根据定义10, $M,s\vDash d_E(e')\Rightarrow(\exists\beta\in A)(For(\beta,e'))\&(\forall\beta\in A)(For(\beta,e')\Rightarrow M,s\vDash d_A(\beta))$。已知$M,s\vDash f_G(e',e)$, 类似1)的方法, 我们可以定义动作$\alpha$满足$For(\alpha,e)$且$\beta=\alpha;\varepsilon$, 则有$d_A(\beta)\equiv d_A(\alpha;\varepsilon)$。

根据$d_A(\beta)$的语义, $M,s\vDash d_A(\beta)\Rightarrow\exists W\subseteq S$, $M,s\vDash[\beta]W$, 即$M,s\vDash[\alpha;\varepsilon]W$, 根据动作公理1和公理2, 可知: $\exists W\subseteq S$满足$M,s\vDash[\alpha;\varepsilon]W\Rightarrow M,s\vDash[\alpha]([\varepsilon]W)\Rightarrow M,s\vDash[\alpha]W(*)$。假设$M,s\vDash c_A(\alpha)$, 则$M,s\vDash\neg d_A(\alpha)$, 即$\nexists W\subseteq S$, 使得$M,s\vDash[\alpha]W$, 与前面的结论(*)矛盾, 即$M,s\vDash d_A(\alpha)$。因此$M,s\vDash d_E(e)$。 证毕.

定理 2. F关系继承性。给定CBAC语义模型M, 对目的e和e', 状态$s\in S$, 如果$M,s\vDash f_F(e',e)$, 则有:

1) $M,s\vDash c_E(e')\rightarrow c_E(e)$;

2) $M,s\vDash d_E(e)\rightarrow d_E(e')$。

证明.

1)由$M,s\vDash c_E(e')$可知$\exists\alpha\in A$满足$For(\alpha,e')$且$M,s\vDash c_A(\alpha)$。定义α的规则为$Pre(\alpha)\rightarrow[\alpha]\lambda(e')$且$\neg\lambda(e')\in LIT(Pre(\alpha))$。对状态$s\in S$, $M,s\vDash c_A(\alpha)\Rightarrow(\exists s'\in S)(C_\alpha(s,s')\&R_\alpha(s,s'))$使得$M,s\vDash\neg\lambda(e')$且$M,s'\vDash\lambda(e')$。对目的$e$和$e'$, $M,s\vDash f_F(e',e)\Rightarrow M,s\vDash\lambda(e)$。假设$M,s\vDash\neg c_E(e)$, 即$M,s\vDash d_E(e)$, 则存在动作$\beta$满足$For(\beta,e)$且$M,s\vDash d_A(\beta)$。这样, 对任意状态都有$M,s\vDash\neg\lambda(e)$成立, 与之前的结论$M,s\vDash\lambda(e)$矛盾。因此有$M,s\vDash c_E(e)$成立。

2)由$M,s\vDash d_E(e)$可知$\exists\beta\in A$满足$For(\beta,e)$且$M,s\vDash d_A(\beta)$, 因此有$(\exists s'\in S)(R_\beta(s,s'))\ \&(\forall s'\in S)(R_\beta(s,s')\Rightarrow D_\beta(s,s'))$。假设关于$\beta$的规则为$Pre(\beta)\rightarrow[\beta]\lambda(e)$且$\neg\lambda(e)\in LIT(Pre(\beta))$。这样, 如果$\beta$被拒绝执行, 则$\forall s\in S$都有$M,s\vDash\neg\lambda(e)$。

当$M,s\vDash d_E(e)$时, 假设$M,s\vDash c_E(e')$。对目的e', $\exists\alpha\in A$满足$For(\alpha,e')$且$M,s\vDash c_A(\alpha)$。定义α的规则为$Pre(\alpha)\rightarrow[\alpha]\lambda(e')$且$\neg\lambda(e')\in LIT(Pre(\alpha))$, 由$M,s\vDash c_A(\alpha)$可知$(\exists s'\in S)\ (C_\alpha(s,s')\&R_\alpha(s,s'))$, 则有$M,s\vDash\neg\lambda(e')$。由$M,s\vDash f_F(e',e)$可知, 如果$\alpha$是同意执行的, 则必有$M,s\vDash\lambda(e)$, 这与前面的结论$M,s\vDash\neg\lambda(e)$矛盾, 这样有$M,s\vDash d_E(e')$。 证毕.

定理1和定理2可以进一步推广到多个子目的的情况。给定$PG=(E,R_E)$和CBAC语义模型M, 已知PG上的目的结点$e'\in E$及其子结点e_1, e_2, …, $e_n\in E$, 则$\forall e_i(1\leqslant i\leqslant n)$, 有:

1)如果$M,s\vDash f_F(e',e_i)$成立, 则有

① $M,s\vDash c_E(e')\rightarrow c_E(e_1)\wedge\cdots\wedge c_E(e_n)$;

② $M,s\vDash d_E(e_1)\vee\cdots\vee d_E(e_n)\rightarrow d_E(e')$。

2)如果$M,s\vDash f_G(e',e_i)$成立, 则有

① $M,s\vDash c_E(e_1)\vee\cdots\vee c_E(e_n)\rightarrow c_E(e')$;

② $M,s\vDash d_E(e')\rightarrow d_E(e_1)\wedge\cdots\wedge d_E(e_n)$。

定理1和定理2表示了G关系中的父子目的继承关系: 如果子目的同意, 则父目的也同意; 如果父目的拒绝(即不同意), 则子目的拒绝, 即, G在目的层次结构中同意向上继承, 拒绝则向下继承。F关系中的父子目的继承关系: 如果父目的同意, 则子目的

也同意；如果子目的拒绝，则父目的也拒绝，即，F 在目的层次结构中同意向下继承，拒绝向上继承。本文的继承理论与其他研究不同，主要对比请见第 6 节。

3.5 推理规则

已知 CBAC 模型 $\mathcal{M}$，对数据主体的隐私偏好 (*AIP*, *PIP*)，用 AIP^+ 和 PIP^+ 分别表示 *AIP* 和 *PIP* 的闭包，$\forall e \in E$，则有如下公理成立：

公理 3. $e \in AIP \rightarrow e \in AIP^+$。

公理 4. $e \in PIP \rightarrow e \in PIP^+$。

根据定理 1 和定理 2，对目的 e 和 e'，用 *EIP* 表示用户同意目的集合，则我们根据用户的隐私偏好 (*AIP*, *PIP*)，可以用

$$EIP = C(AIP, PIP) \tag{6}$$

计算用户的同意目的集合，其中 C 表示同意计算算法，该算法包含如下的规则：

规则 1. $(\exists e' \in AIP: f_F(e', e)) \rightarrow e \in AIP^+$。

规则 2. $(\exists e \in PIP: f_F(e', e)) \rightarrow e' \in PIP^+$。

规则 3. $(\exists e \in AIP: f_G(e', e)) \rightarrow e' \in AIP^+$。

规则 4. $(\exists e' \in PIP: f_G(e', e)) \rightarrow e \in PIP^+$。

规则 5. $EIP \equiv AIP^+ \backslash PIP^+$。

规则 6. $e \in EIP \leftrightarrow c_E(e)$。

规则 7. $e \notin EIP \leftrightarrow d_E(e)$。

其中规则 1 和规则 2 分别表示 F 关系的目的同意的向下继承和拒绝的向上继承；规则 3 和规则 4 分别表示 G 关系的目的同意的向上继承和拒绝的向下继承；规则 5 表示 *EIP* 的计算方法，体现了拒绝优先原则；规则 6、规则 7 表示了对目的 e 的同意和拒绝与 *EIP* 之间的关联，其中规则 7 是对最小权限原则的实现。

有了上述规则，我们可以进一步给出授权判断规则：$\forall q = (u, r, e) \in Q$，其中 $u \in U$，$r \in R$，$e \in E$：

规则 8. $c_E(e) \rightarrow Consent(q)$。

规则 9. $d_E(e) \rightarrow Refuse(q)$。

规则10. $(\exists r' \in R: (ua(u, r) \wedge rh(r, r') \wedge ra(r', e))) \rightarrow Auth(q)$。

规则11. $\neg ua(u, r) \vee (\nexists r' \in R: (ua(u, r) \wedge rh(r, r') \wedge ra(r', e))) \rightarrow Deny(q)$。

规则12. $Consent(q) \wedge Auth(q) \rightarrow Accept(q)$，其中 $Accept(q)$ 表示 q 被接受。

规则13. $Refuse(q) \vee Deny(q) \rightarrow Reject(q)$，其中 $Reject(q)$ 表示 q 被拒绝。

规则 8 和规则 9 描述了请求者是否有同意权限；规则 10 和规则 11 描述了请求者是否有角色权限；规则 12 和规则 13 描述了系统是否接受或拒绝该请求。

Byun 等人[11-12] 对目的的集合计算方法中用 $E^{\uparrow}$ 表示 E 中所有元素以及这些元素的祖先组成的集合，用 $E^{\downarrow}$ 表示 E 中所有元素以及这些元素的后代组成的集合，其中↑表示向上继承，↓表示向下继承。如果采用这种表达方式，规则 1~4 可以重写为：

规则 14. $AIP^+ = AIP_G^{\uparrow} \cup AIP_F^{\downarrow}$。

规则 15. $PIP^+ = PIP_G^{\downarrow} \cup PIP_F^{\uparrow}$。

4 CBAC 模型的 TLA+描述

本节用 TLA+语言按照规则 1~15 对 CBAC 模型进行表示，其中包含 3 个并发进程 *Consent*, *GrantConsent*, *RevokeConsent*，其中 *Consent* 表示对任意一个访问请求进行同意计算的执行过程；*GrantConsent* 和 *RevokeConsent* 是动态同意执行过程，分别表示授予同意和撤销同意动作。符号 *E*, *R*, *U*, *UA*, *RH*, *EA*, *G*, *F* 的含义与 CBAC 模型 $\mathcal{M}$ 相同。

请求集合 $Requests \triangleq U \times R \times E$。

变量定义为 $vars \triangleq \langle pc, EIP, AIP, PIP, revoked, q, e_revoked, e_grant \rangle$。其中，*AIP* 和 *PIP* 分别表示允许目的和禁止目的的集合；*EIP* 表示同意目的集合；*pc* 用来指定进程变量；*revoked* 表示撤销同意的目的集合；*q* 表示请求变量；*e_revoked* 表示撤销同意的目的变量；*e_grant* 表示授予同意的目的变量。

1）递归函数 *DeriveAIP*[*AIP*] 求 *AIP* 的闭包：

DeriveAIP[*initialAIP*∈SUBSET *E*]≜

LET *NewAIP*≜*initialAIP* ∪

{$e \in E: \exists k \in initialAIP: \langle e, k \rangle \in G$} ∪

{$m \in E: \exists n \in initialAIP: \langle n, m \rangle \in F$}

IN IF *NewAIP*=*initialAIP* THEN *initialAIP*

ELSE *DeriveAIP*[*NewAIP*]。

2）递归函数 *DerivePIP*[*PIP*] 求 *PIP* 的闭包：

DerivePIP[*initialPIP*∈SUBSET *E*]≜

LET *NewPIP*≜*initialPIP* ∪

{$e \in E: \exists k \in initialPIP: \langle k, e \rangle \in G$} ∪

{$e \in E: \exists k \in initialPIP: \langle e, k \rangle \in F$}

IN IF *NewPIP*=*initialPIP* THEN *initialPIP*

ELSE *DerivePIP*[*NewPIP*]。

3）角色权限继承定义：

RoleClosure[*r*∈*R*]≜

LET *SubRoles*≜{$rx \in R: \langle r, rx \rangle \in RH$}；

IN IF *SubRoles*={} THEN {*r*}；

ELSE {*r*}∪UNION {*RoleClosure*[*rx*]: *rx*∈ *SubRoles*};

pur[*x*∈*R*] ≜ {*e* ∈ *E*:⟨*x*,*e*⟩ ∈ *EA*} ∪ {*e* ∈ *E*:∃*r_low*∈*RoleClosure*[*x*]:⟨*r_low*,*e*⟩ ∈ *EA*}。

4）同意计算相关定义：

IsAuthorized(*q*) ≜ ∧ ⟨*q*[1],*q*[2]⟩ ∈ *UA* ∧(∃*rz* ∈ *RoleClosure*[*q*[2]]:⟨*rz*,*q*[3]⟩ ∈ *EA*);

IsConsented(*e*)≜*e* ∈ *EIP*;

AccessDecision(*q*) ≜ IF *IsConsented*(*q*[3])∧ *IsAuthorized*(*q*) THEN "Accept" ELSE "Reject";

Consent(*q*)≜IF *q*[3]∈*EIP* THEN TRUE ELSE FALSE;

Auth(*q*)≜*IsAuthorized*(*q*);

Refuse(*q*)≜IF *q*[3]∉*EIP* THEN TRUE ELSE FALSE;

Deny(*q*)≜¬*Auth*(*q*);

Accept(*q*)≜IF *AccessDecision*(*q*)="Accept" THEN TRUE ELSE FALSE;

Reject(*q*)≜IF *AccessDecision*(*q*)="Reject" THEN TRUE ELSE FALSE。

5）总进程定义：

ProcSet≜{"Consent"}∪{"GrantConsent"} ∪ {"RevokeConsent"}。

6）变量初始化：

Init≜∧*EIP*={}∧*AIP*∈SUBSET *E*
∧*PIP*∈SUBSET *E*∧*revoked*={}∧*q*∈*Requests*
∧*e_revoked*∈*E*∧*e_grant*∈*E*
∧*pc*=[*self*∈*ProcSet*↦
CASE *self*="Consent"→"Lable1"
□*self*="GrantConsent"→"Lable2"
□*self*="RevokeConsent"→"Lable5"]。

7）*Consent* 动作定义：

*L*1≜∧*pc*["Consent"]="Lable1"
∧*EIP*′=*DeriveAIP*[*AIP*] \ *DerivePIP*[*PIP*]
∧*pc*′=[*pc* EXCEPT !["Consent"]="Done"]
∧UNCHANGED ⟨*AIP*, *PIP*, *revoked*, *q*, *e_revoked*, *e_grant*⟩;

Consent≜*L*1。

8）添加同意 *GrantConsent* 动作定义：

*L*2≜∧*pc*["GrantConsent"]="Lable2"
∧*e_grant*′=(CHOOSE *e* ∈ *E*:TRUE)
∧*pc*′=[*pc* EXCEPT !["GrantConsent"]="Lable3"]
∧UNCHANGED ⟨*EIP*, *AIP*, *PIP*, *revoked*, *q*, *e_revoked*⟩;

*L*3≜∧*pc*["GrantConsent"]="Lable3"
∧IF (*e_grant*∉*AIP*) THEN
∧IF (*e_grant*∈*PIP*) THEN
∧*PIP*′=*PIP* \ {*e_grant*}
∧*AIP*′=(*AIP*∪{*e_grant*})
ELSE
∧*AIP*′=(*AIP*∪{*e_grant*})
∧*PIP*′=*PIP*
∧*pc*′=[*pc*
EXCEPT !["GrantConsent"]="Lable4"]
ELSE
∧*pc*′=[*pc* EXCEPT !["GrantConsent"] ="Done"]
∧UNCHANGED ⟨*AIP*,*PIP*⟩
∧UNCHANGED ⟨*EIP*, *revoked*, *q*, *e_revoked*, *e_grant*⟩;

*L*4≜∧*pc*["GrantConsent"] = "Lable4"
∧*EIP*′=*DeriveAIP*[*AIP*] \ *DerivePIP*[*PIP*]
∧IF *e_grant*∈*revoked*
THEN *revoked*′ = *revoked* \ {*e_grant*}
ELSE
∧ UNCHANGED *revoked*
∧*pc*′=[*pc* EXCEPT !["GrantConsent"] = "Done"]
∧UNCHANGED ⟨*AIP*, *PIP*, *q*, *e_revoked*, *e_grant*⟩;

GrantConsent≜*L*2∨*L*3∨*L*4。

9）撤销同意 *RevokeConsent* 动作定义：

*L*5≜∧*pc*["RevokeConsent"]="Lable5"
∧IF *AIP*≠{} THEN
∧*e_revoked*′=(CHOOSE *e*∈*AIP*: TRUE)
∧*AIP*′=*AIP* \ {*e_revoked*′}
∧*revoked*′=(*revoked*∪{*e_revoked*′})
∧*EIP*′=*DeriveAIP*[*AIP*′] \ (*DerivePIP*[*PIP*]∪ *DerivePIP*[*revoked*′])
ELSE
∧UNCHANGED ⟨*EIP*, *AIP*, *revoked*, *e_revoked*⟩
∧*pc*′=[*pc* EXCEPT !["RevokeConsent"]="Done"]
∧UNCHANGED ⟨*PIP*, *q*, *e_grant*⟩;

RevokeConsent≜*L*5。

10）系统规约：

Terminating ≜ ∧ ∀*self*∈*ProcSet*: *pc*[*self*]="Done"
∧UNCHANGED *vars*;

Next ≜ *Consent* ∨ *GrantConsent* ∨ *RevokeConsent* ∨ *Terminating*;

Spec ≜ *Init* ∧ □ [*Next*]_*vars*;

Termination ≜ ◇ (∀ *self* ∈ *ProcSet*: *pc*[*self*]="Done")。

5 安全性定义与实验评估

5.1 安全性定义

隐私数据访问控制主要面临着同意权限及角色权限受侵犯的威胁。因此，如果同时满足了授权和同意 2 种控制条件，则我们认为访问控制系统是安全的。具体的，我们通过如下 8 条安全性质来展开安全性的讨论。其中性质 1~6 为基本安全性质，性质 7 和性质 8 为动态安全性质。

1）基本安全性质

性质 1. 任何时刻访问请求仅在同意且授权时被接受。

$$AcceptSafety \triangleq \forall q \in Requests:$$
$$\Box(Accept(q) \Rightarrow (Consent(q) \wedge Auth(q)))。$$

该性质表示在任何时刻，请求 q 被接受的必要条件是该请求被同意且被授权。

性质 2. 任何时刻未同意或未授权的请求必须被拒绝。

$$RejectSafety \triangleq \forall q \in Requests:$$
$$\Box((Refuse(q) \vee Deny(q)) \Rightarrow Reject(q))。$$

性质 3. 双重授权交叉影响，访问请求获得了授权，但未获得同意，则该访问请求被拒绝。

$$DualSafety1 \triangleq \forall q \in Requests:$$
$$\Box(Auth(q) \wedge Refuse(q) \Rightarrow Reject(q))。$$

性质 4. 双重授权交叉影响，访问请求获得了同意，但未获得授权，则该访问请求被拒绝。

$$DualSafety2 \triangleq \forall q \in Requests:$$
$$\Box(Deny(q) \wedge Consent(q) \Rightarrow Reject(q))。$$

性质 5. 同意目的继承性，分为 G 关系向上继承性和 F 关系向下继承性，其中

① G 向上继承性。

$$UpwardInheritance_G \triangleq \forall e_1, e_2 \in E:$$
$$\langle e_1, e_2\rangle \in G \wedge e_2 \in EIP \Rightarrow e_1 \in EIP。$$

② F 向下继承性。

$$DownwardInheritance_F \triangleq \forall e_1, e_2 \in E:$$
$$\langle e_1, e_2\rangle \in F \wedge e_1 \in EIP \Rightarrow e_2 \in EIP。$$

性质 6. 角色层次的目的继承性，即下级角色的目的必须被上级角色继承。

$$RolePurposeInheritance \triangleq \forall e_3 \in E: \forall r_high \in R:$$
$$(\forall r_low \in RoleClosure[r_high]: e_3 \in pur[r_low] \Rightarrow$$
$$e_3 \in pur[r_high])。$$

2）撤销同意性质

性质 7. 撤销同意后，相关请求最终被拒绝。

$$RevokePropagates \triangleq \forall q \in Requests, \forall e_revoked \in E:$$
$$\Box(q[3] = e_revoked \wedge e_revoked \in revoked \Rightarrow$$
$$Reject(q))。$$

解释：一旦目的被撤销同意，则所有关于该目的的访问请求都会被拒绝。其中 $q[3]$ 表示 q 的第 3 个分量，即 q 中包含的目的。

性质 8. 目的被撤销同意后，与该目的相关的继承性同意目的也将被撤销同意。

$$RevokeInheritance \triangleq \forall e_revoked \in E:$$
$$\Box(e_revoked \in revoked \Rightarrow (\forall e \in E:$$
$$(\langle e_revoked, e\rangle \in G \vee \langle e, e_revoked\rangle \in F) \Rightarrow$$
$$e \notin EIP))。$$

目的被撤销同意和禁止的区别如下：

1）目的被撤销同意指对原本已经在 *AIP* 中被允许的目的，在运行时主动收回同意，通过 *RevokeConsent* 动作（见第 4 节该动作的定义）写入 *revoked* 集合，并在性质 7、性质 8 中体现其对相关继承目的的传播效应。因此对目的 e，撤销同意是 $e \in AIP$ 转变为 $e \in revoked$ 的过程，本文对撤销同意的目的虽然未明确划入 *PIP* 集合，但在计算 *EIP* 时也是按照禁止目的来处理，这个步骤体现在 *RevokeConsent* 动作定义中。

2）目的被禁止指用户在初始隐私偏好中明确标记为不允许的目的，即目的 $e \in PIP$，通过闭包 PIP^+ 以及 $EIP = AIP^+ \backslash PIP^+$ 体现在同意计算中。如果用户的隐私偏好发生变化，也可以将原本禁止的目的改为允许，再重新计算 *EIP* 集合，这个步骤体现在 *GrandConsent* 动作定义中。

5.2 CBAC 系统部署与维护

CBAC 是一种与具体策略语言无关的访问控制模型。为了实现基于 CBAC 的访问控制系统，本文在 3.1 节提出的系统 3 类功能模块可直接映射到 XACML 架构，如图 6 所示。图 6 上层为数据请求者/应用，通过 PEP（AC）发起携带主体标识、角色、访问目的、资源和动作等属性的 XACML 请求。中间层为 XACML 引擎，其中 PDP 内部集成 AD 和 CD：AD 基于角色-目的映射及角色层次（*RA*, *RH*）进行角色授权判断，CD 基于用户隐私偏好 (*AIP*, *PIP*)、目的图（包含 *G*/*F* 关系）和同意闭包 *EIP* 进行同意判定。PDP 按 CBAC 推理规则（规则 12 和规则 13）组合 AD 和 CD 结果，即对访问请求 q，获得同意及角色授权则接受，否则拒绝；并将决策返回 PEP，由其执行业务放行或拒绝。下层为 PAP/PIP 及相关数据服务：PAP 负责管理和发布角色—目的映射、目的图及同意策略，PIP 提供用

户隐私偏好库、同意目的库/EIP缓存、角色/目的/属主等属性数据; PEP, PDP, CD/AD的操作统一记录到审计与日志系统, 用于合规审计与追溯。整套架构在属性驱动的XACML框架中实现CBAC的“角色—目的—同意”统一决策。

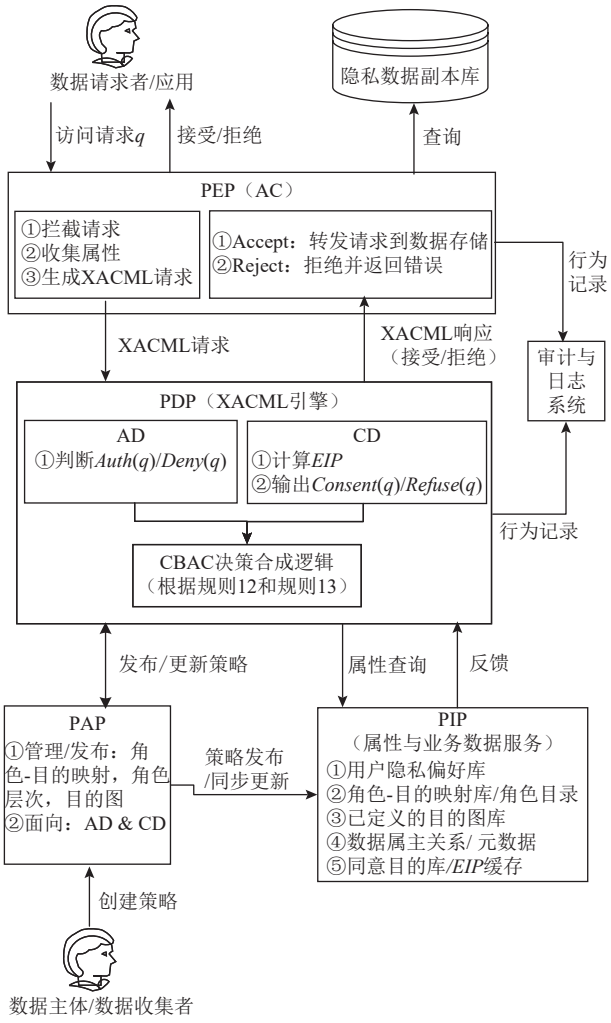


Fig. 6 An example of CBAC model-based access control system deployment

图6 基于CBAC模型的访问控制系统部署示例

CBAC系统的维护包括3个方面:

1)建模。系统上线前, 数据收集者完成目的图的建模工作。

2)日常运维。成本主要体现在偏好更新和同意重计算上, 对系统整体影响有限。

3)长期维护与合规。与仅依赖自然语言隐私政策的方式相比, CBAC提供了“自然语言隐私条款→目的图→TLA+形式规约→模型检测”的技术链条, 使得隐私策略可以在形式化层面被描述和验证。当业务需求或法规要求发生变化时, 数据收集者可以在目的图和TLA+规约层面增量更新策略, 并通过模型检测工具快速检查更新后的配置是否仍满足预期的安全性质, 从而降低长期演进过程中的策略错误和合规风险。

因此, 在实现CBAC系统时, 配置负担主要体现在2个层面:

1)数据收集者在系统上线前需要识别业务相关的收集/使用目的, 构建规模为几十至上百个结点的目的图, 并标注概括关系G和前提条件关系F, 同时配置角色-目的分配关系。该部分工作通常由隐私合规和技术人员协同完成, 后续仅在业务拓展或策略调整时进行增量维护。

2)普通用户只需在较高层次目的上选择允许或禁止(*AIP*, *PIP*), 系统会根据目的图结构和同意继承规则自动推导细粒度同意结果, 用户界面形式可与常见的“隐私设置”或“同意管理”页面保持一致, 不会显著增加操作复杂度。

本文接下来只关注CBAC模型关键性质和算法的验证。

5.3 实验及评估

5.3.1 计算复杂度分析

为合理设计实验规模, 并解释后续实验配置的取舍, 本节从理论上分析同意计算、角色授权计算以及基于TLC的模型检测在目的数量和关系规模上的复杂度。

1)同意计算复杂度

令目的集合为E, 概括关系集合为$G \subseteq E \times E$, 前提关系集合为$F \subseteq E \times E$, 二者均为目的图上的有向边集合。*DeriveAIP*/*DerivePIP*的递归定义基于目的图中的这些有向边进行推理, 每一轮计算中, 需要基于当前*AIP*/*PIP*集合, 对所有有向边进行遍历并更新同意目的集合, 其时间复杂度与边集规模($|G|+|F|$)线性相关。单轮递推时间复杂度为$O(|G|+|F|)$, 在最坏情况下迭代轮数不超过$|E|$。因此, 1次同意计算所消耗的时间为

$$T_{EIP} = O(|E| \times (|G| + |F|))。 \tag{7}$$

由$|G|+|F| \leqslant |E|^2$可知, 仅依赖目的数量的最坏情况复杂度为

$$T_{EIP} = O(|E|^3)。 \tag{8}$$

说明在固定目的图结构下, 同意计算算法在目的规模增大时仍保持多项式时间复杂度。

2)角色授权复杂度

除同意计算外, 每个访问请求还需要进行角色授权判定*Auth*/*Deny*, 其核心操作是对请求角色r求

其角色层次闭包 *RoleClosure*(*r*), 并在闭包结果中检查是否存在某个角色与请求目的 *e* 的角色-目的映射关系。令角色集合为 *R*, 角色层次关系 $RH \subseteq R \times R$。根据第 4 节中 *RoleClosure* 的递归定义可知：对任意给定角色 *r*, *RoleClosure*(*r*) 实质上是在 *RH* 所诱导的有向图上做一次从 *r* 出发的可达性遍历, 其时间复杂度与角色数和边数线性相关, 为

$$T_{RoleClosure}(r) = O(|R| + |RH|)。 \tag{9}$$

对单个访问请求 $q = (u, r, e)$, 授权判定 *IsAuthorized*(*q*) 只需计算 1 次 *RoleClosure*(*r*); 在 *RoleClosure*(*r*) 与角色-目的映射 *RA* 上做 1 次集合包含/连接检查（时间复杂度至多为 $O(|RoleClosure(r)|) \leqslant O(|R|)$）。因此, 对单个请求的授权判定开销为

$$T_{Auth}(q) = O(|R| + |RH|)。 \tag{10}$$

将 T_{Auth} 与 T_{EIP} 对比可以看到, 在典型系统中, 角色数及角色层次边数 ($|R|, |RH|$) 远小于目的数及目的关系边数 ($|E|, |G| + |F|$), 且授权判定不涉及递归闭包重计算（只对单个角色做 1 次遍历）, 故在总体访问决策中, *Consent*/*Refuse* 的 *EIP* 计算仍是主要影响因素, 而 *Auth*/*Deny* 带来的附加开销相对较小。由此, 对每个请求 *q*, 总体时间复杂度为 $T(q) = O(|E| \times (|G| + |F|)) + O(|R| + |RH|)$, 在 $|E| \gg |R|$ 的现实配置下, 可近似视为由同意计算主导。

从 1）和 2）的分析可以看出访问请求的 *Consent*/*Refuse* 与 *Auth*/*Deny* 判定主要涉及集合运算和角色层次闭包计算, 计算开销较小。因此, 在典型系统规模下, CBAC 可以支持包含大量目的和关系的目的图, 具有良好的可扩展性。

3）模型检测复杂度分析

在采用 TLC 进行模型检测时, 系统可达状态空间的规模 $|S|$ 由用户数 $|U|$、角色数 $|R|$ 以及目的数 $|E|$ 共同决定。根据第 4 节给出的 TLA+规约, 可以得到系统理论上可达状态数目的一个上界：

$$|S| \leqslant O(2^{4|E|} \times |U| \times |R| \times |E|^3), \tag{11}$$

其中指数项 $2^{4|E|}$ 主要由 4 个目的子集（*AIP*, *PIP*, *EIP*, *revoked*）的组合, 体现了目的数 $|E|$ 对状态空间规模的指数级影响。式（11）反映了 TLC 模型检测的状态空间随目的数呈指数级增长。因此我们在面对实际隐私政策时将不可避免地遇到状态空间爆炸问题。

因此, 从理论复杂度角度可以预见：在目的数量较大、关系结构复杂的真实应用场景下, 直接对完整目的图进行模型检测将面临显著的状态空间挑战。5.3.2 节将结合抖音隐私政策这一具体案例, 对该问题进行实证量化, 并说明为何需要对目的图规模进行分解。

5.3.2　实际隐私政策规模分解

为了评估 CBAC 模型在真实场景中的适用性, 本节选取抖音隐私政策作为实验原型。从该隐私政策中抽取与个人数据处理直接相关的目的, 共得到约 90 个目的结点, 并依据业务逻辑和隐私条款之间的依赖关系构建了对应的目的图, 如图 7 所示。该目的图包含数量可观的概括关系边集 *G* 和前提关系边集 *F*, 整体结构远比图 5 显示的目的图示例复杂。

将 $|E|$ 及典型的 $|U|$, $|R|$ 配置代入 5.3.1 节给出的状态空间上界估计后可以发现, 即便在较为保守的情况下, 理论可达状态数仍然达到极大的数量级。初步实验表明, 在现有硬件条件（CPU、内存等）和 TLC 默认配置下, 直接对完整 90 个目的的抖音目的图进行模型检测, 将导致状态数和内存占用迅速超过可承受范围, 难以在有限时间内完成全部安全性质的验证。

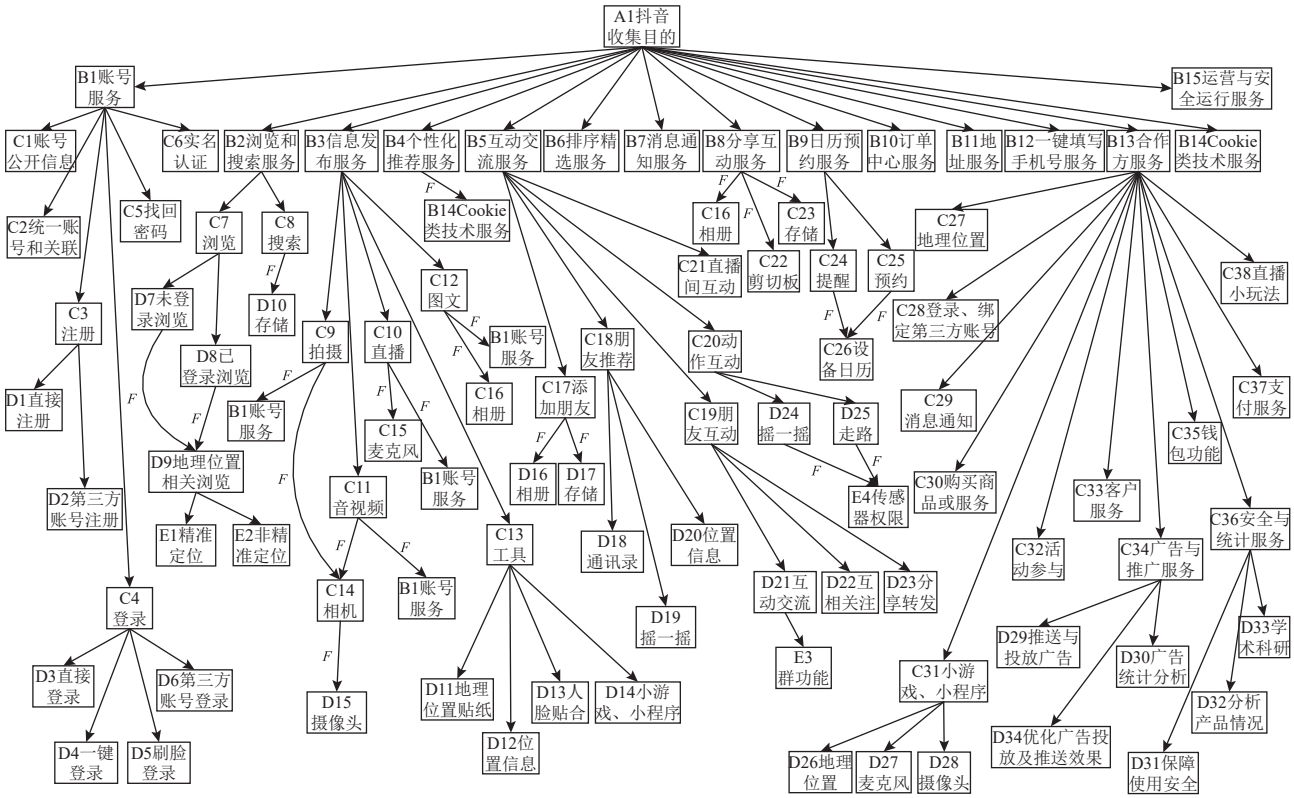
因此, 为了验证 CBAC 的安全性, 本文对基于抖音隐私政策构建的目的图进行了结构化分解。分解的基本思路是：在遵循业务逻辑和隐私条款语义的前提下, 将原始目的图按功能模块或角色-目的子树划分为若干子图, 每个子图包含若干紧密相关的目的结点以及完整的概括关系（*G*）和前提关系（*F*）。这样既能显著降低单次模型检测的目的数量（用 $|E'|$ 表示, 其中 E' 为分解后子图的目的集合）, 又能保留同意继承和冲突处理所依赖的关键结构。

在具体实验中, 本文选取了若干具有代表性的子图, 其规模约为 $|E'| \in [8, 11]$（可根据实际实验配置精确给出）, 并在每个子图上分别配置不同的用户数 ($|U|$)、角色数 ($|R|$), 通过逐步增加 $|E'|$, $|U|$, $|R|$ 的方式观察状态数和检测时间的变化趋势。将 5.3.1 节中的复杂度分析缩放到子图规模, 可以看到同意计算的理论复杂度由 $O(|E| \times (|G| + |F|))$ 缩减为 $O(|E'| \times (|G'| + |F'|))$, 而 TLC 的状态空间上界也由 $O(2^{4|E|})$ 量级降至 $O(2^{4|E'|})$ 量级, 在当前硬件条件下变得可操作。

这种目的图分解并非简单的“删减目的”, 而是在保持典型 *G*, *F* 关系和同意继承语义的前提下, 对原始图进行语义保持的拆分。因此, 分解后的实验结果依然能够反映 CBAC 模型在真实隐私策略下的可扩展性特征。

5.3.3　实验结果与评估

本节分别对 *Consent*, *RevokeConsent*, *GrantConsent* 这 3 类动作进行模型检测, 其中动作 *Consent* 和 *GrantConsen* 用来验证 CBAC 模型是否满足 5.1 节给



注：1）每个矩形框表示一个目的，每个目的均有编号，其中 A，B，C，D 表示层号，数字表示同一层的编号；2）图上只标注了 *F* 关系，其余未标注的关系默认为 *G* 关系；3）方框内目的省略了收集的数据类型。

Fig. 7 Purpose graph of Douyin

图 7 抖音目的图

出的基本安全性质(性质 1~6)，*RevokeConsent* 用来验证动态撤销性质(性质 7 和性质 8)。实验所采用的硬件环境为 12th Gen Intel® Core™ i7-127000 2.10 GHz，内存为 128 GB。软件环境为 Windows 10 系统，模型检测套件采用 TLA+ tools(含 TLC 模型检测工具)1.8.0 版本。每类动作的安全性测试都用 14 个线程来进行，结果的统计方式包括遍历的状态数量、发现的不同状态数以及检测花费的时间。在配置实验环境时，每次实验模型配置都按照用户数、角色数和目的数的递增进行，而 *AIP* 和 *PIP* 则是让模型检测工具随机选取 *E* 的子集，这样所有的组合都在测试范围内，保证所有的路径都无遗漏。在输入模型配置时，我们抽象了具体的目的名称，用符号 e_1，e_2，…，表示，保留了目的图的结构和关系。

实验结果如下：

1) *Consent* 安全性验证结果

表 2 给出了不同配置规模下验证 *Consent* 动作满足基本安全性的实验结果。

从表 2 可以看出，目的数的增加对整个测试状态的数量影响较大。测试过程没有出现反例提示，说明模型满足基本安全性质。模型的描述将权限的判断全部用函数的形式进行定义，动作比较单一，因此状态图直径保持在 2 左右。

Table 2 Security Verification Results of *Consent* Action

表 2 *Consent* 动作安全性验证结果

TLC 模型	状态数	不同状态数	检测时间/s
(10,5,9)	353 894 400	235 929 600	1957
(10,5,10)	1 572 864 000	1 048 576 000	13 793
(10,5,11)	7 072 854 926	4 765 987 726	83 933
(10,6,9)	424 673 280	283 115 520	2 334
(10,6,10)	1 887 436,800	1 258 291 200	16 213
(10,6,11)	8 538 644 400	5 770 403 760	110 334
(10,7,9)	495 452 160	330 301 440	2 813
(10,7,10)	2 202 009 600	1 468 006 400	20 699
(11,5,9)	389 283 840	259 522 560	2 327
(11,5,10)	1 730 150 400	1 153 433 600	16 657
(11,5,11)	8 026 877 017	5 489 323 097	74 805

注：括号中的 3 个数字分别表示用户数、角色数和目的数。

2) *GrantConsent* 安全性验证结果

由于目的层次存在同意目的的继承性质，因此，即使将初始不在 *AIP* 集合中的目的添加进 *AIP* 中，也

不能保证最终该目的被同意。因此，这里对执行 *GrantConsent* 动作之后基本安全性质是否满足进行测试。结果如表 3 所示，其中状态图直径为 4。

Table 3 Security Verification Results of *GrantConsent* Action

表 3 *GrantConsent* 动作安全性验证结果

TLC 模型	状态数	不同状态数	检测时间/s
(5,4,8)	188 743 680	104 857 600	1 109
(5,4,9)	943 718 400	519 045 120	6 313
(5,4,10)	4 613 734 396	2 516 582 296	38 713
(5,5,8)	235 929 600	131 072 000	1 405
(5,5,9)	1 179 648 000	648 806 400	8 246
(5,5,10)	5 767 167 999	3 145 727 999	48 566
(6,4,8)	226 492 416	125 829 120	1 314
(6,4,9)	1 132 462 080	622 854 144	7 838
(6,4,10)	5 536 481 280	3 019 898 880	52 342
(6,5,8)	283 115 520	157 286 400	1 697
(6,5,9)	1 415 577 600	778 567 680	9 744
(6,5,10)	6 920 601 600	3 774 873 600	59 124

注：括号中的 3 个数字分别表示用户数、角色数和目的数。

表 3 结果表明验证的过程中未出现反例，说明本模型执行 *GrantConsent* 动作后能满足基本安全性质。

3）*RevokeConsent* 安全性验证结果

本实验测试撤销同意性质，包括性质 7 和性质 8。测试结果如表 4，其中所有的状态图直径为 2。

Table 4 Security Verification Results of *RevokeConsent* Action

表 4 *RevokeConsent* 动作安全性验证结果

TLC 模型	状态数	不同状态数	检测时间/s
(5,4,8)	178 544 640	94 658 560	461
(5,4,9)	897 269 760	470 596 480	2 774
(5,4,10)	4 405 862 400	2 308 710 400	17 850
(5,5,8)	223 180 800	118 323 200	568
(5,5,9)	1 121 587 200	590 745 600	4 141
(5,5,10)	5 507 328 000	2 885 888 000	38 559
(6,4,8)	214 253 568	113 590 272	664
(6,4,9)	1 076 723 712	567 115 776	4 129
(6,4,10)	5 287 034 880	2 770 452 480	27 968
(6,5,8)	267 816 960	141 987 840	1 256
(6,5,9)	1 345 904 640	708 894 720	6 545
(6,5,10)	6 608 793 600	3 463 065 600	41 124

注：括号中的 3 个数字分别表示用户数、角色数和目的数。

以上测试的结果均未出现反例，说明本模型满足性质 7 和性质 8。

以上实验结果表明在分解后的子图规模下，所有配置均未出现安全性质的反例，CBAC 模型在该范围内能够满足预期的安全要求。

从可扩展性角度看，随着目的数 $|E'|$ 从 8 逐步增加到 10 或 11，在 $|U|$，$|R|$ 相对固定的情况下，TLC 报告的状态总数和不同状态数均呈显著增长趋势，检测时间也相应增加。这一现象与 5.3.1 节中基于 $O(2^{4|E|}\times|U|\times|R|\times|E|^3)$ 的状态空间复杂度分析相吻合，说明理论预测与实际模型检测结果是一致的。同时，在本文选取的子图规模及硬件配置下，检测时间仍然处于可接受范围之内，表明通过合理分解可以在工程实践中有效控制模型检测的成本。

综上可以得出：1）CBAC 同意计算算法在目的数量和关系规模扩展时保持多项式时间复杂度，具备良好的算法可扩展性；2）尽管基于完整抖音隐私政策构建的目的图在当前硬件条件下难以直接完成模型检测，但通过语义保持的目的图分解，可以在较小规模的子图上验证 CBAC 模型的安全性质，并观察到与理论分析一致的状态空间增长趋势。这表明，所提出的模型和同意计算方法在真实隐私策略背景下具有较强的工程可行性和推广潜力。

6 相关研究比较

由于现有文献未给出同意计算的形式化方法，为了与这些文献进行比较，我们基于本文的形式化理论将现有文献的同意计算方法抽取出来，从概括关系 G、前提关系 F、同意继承性和同意计算结果等多个角度进行比较，并给出实验结果，从理论和实验 2 个角度来说明本文的方法与现有方法的不同。

已知目的集合 E 和用户初始的隐私偏好 $(AIP_{\mathrm{I}}, PIP_{\mathrm{I}})$，其中 AIP_{I} 表示用户允许目的集合，PIP_{I} 表示用户禁止目的集合（下标 I 表示初始的（initial））。根据式（6），最终计算出同意目的集合 EIP。表 5 表示已有研究与本文在同意计算方法在理论上的不同之处。为了在表示上一致，所有的概括、包含关系都用 G 表示。其中 AIP_G，AIP_F，PIP_G，PIP_F 中的下标 G 和 F 分别用来表示基于关系 G 和 F 的继承闭包集合。

为了更加直观地对比这些方法，我们将抖音隐私政策作为实例，以抖音目的图（图 7）为实验对象进行上述方法的实验对比，其中抖音目的图规模为：$|E|$=90，$|G|$=79，$|F|$=22。

表 5 中的方法 1，5，7 包括了 G 和 F 这 2 种关系，方法 3，4 和 6 只有 G 关系。在处理方法 3，4，6 时，我

Table 5 Comparison of Related Studies on Consent Calculation

表 5 同意计算的相关研究对比

方法序号	方法出处	初始目的集合	同意计算方法	目的关系	同意继承性质	同意权限
1	文献 [18]	$(AIP_1,\varnothing)$	$AIP_G^{\downarrow}\cup AIP_F^{\downarrow}$	G, F	G 和 F 关系同意向下继承	无
2	文献 [4, 20, 25]	$(AIP_1,\varnothing)$	AIP_1	无	无继承	有
3	文献 [41]	$(AIP_1,\varnothing)$	$AIP_G^{\downarrow}$	G	同意向下继承	无
4	文献 [13, 24, 31]	$(AIP_1,\varnothing)$	$AIP_G^{\downarrow}\backslash PIP_G^{\downarrow}$	G	同意和拒绝都向下继承，且拒绝优先	无
5	文献 [22]	(AIP_1,PIP_1)	$(AIP_G^{\downarrow}\cup AIP_F^{\downarrow})\backslash(PIP_G^{\downarrow}\cup PIP_F^{\downarrow})$	G, F	同意和拒绝都向下继承，且拒绝优先	无
6	文献 [11−12, 14]	(AIP_1,PIP_1)	$AIP_G^{\downarrow}\backslash(PIP_G^{\downarrow}\cup PIP_G^{\uparrow})$	G	同意向下继承，拒绝向下和向上继承	无
7	本文	(AIP_1,PIP_1)	$(AIP_G^{\uparrow}\cup AIP_F^{\downarrow})\backslash(PIP_G^{\downarrow}\cup PIP_F^{\uparrow})$	G, F	G 关系同意向上继承，拒绝向下继承；F 关系同意向下继承，拒绝向上继承	有
8	文献 [21]	收集目的集合 E	$\{e\|e\in E\}$	无	无	有

注：↑表示向上继承，↓表示向下继承。

们将原本为前提条件的目的全部列为总目的的子目的，以保证所有的方法都基于相同目的规模开展计算。

做实验时，我们同时考虑了 *AIP* 和 *PIP* 的规模选择以及用户对某些目的的偏好问题，设置 *AIP* 集合偏基础服务（例如包括账号服务、浏览和搜索服务、信息发布服务和互动交流服务等），*PIP* 集合偏广告/合作方/位置等敏感用途，以覆盖目的图中的主要 *G*/*F* 关系。具体地，*AIP* 覆盖从稀疏到中等密度，取 $|AIP|\in\{4, 8, 12, 16, 20, 24, 30\}$，其中 4~12 代表“强隐私需求”的用户，选少量高层或核心服务目的；16~30 代表“普通”用户。而 *PIP* 的规模从“不禁止”到“相对激进”，但为了防止全拒绝的现象，让 *PIP* 规模小于 *AIP* 的规模。取 $|PIP|\in\{0, 4, 8, 12, 16, 20\}$，其中 0 为对比“只有允许目的，无显式禁止目的”时各方法的差异；4~8 反映真实用户常见情况（只禁止少数广告/第三方/位置定位等）；12~20 为增加冲突，观察哪些方法开始出现大面积全拒绝或异常现象。

图 8 给出了 $|AIP|$ 和 $|PIP|$ 变化之后各种方法同意目的数量 $|EIP|$ 的变化趋势，其中方法 7 为本文采用的方法。实验结果表明方法 7 和其他方法区别非常明显。尽管采用同意计算的方法可以避免“全同意”的粗粒度结果，但实验结果可以显示，现有的同意计算方法仍存在很多问题，分析如下：

1）从图 8 的整体趋势来看，除了方法 6，$|EIP|$ 随着 $|AIP|$ 规模的增长而增长。但采用了同意算法之后，方法 1, 3, 4, 5 在最小规模环境下（$|PIP|=0$, $|AIP|=4$）计算出 $|EIP|$ 已经是 $|AIP|$ 的 10 倍以上，而且无论 $|AIP|$ 和 $|PIP|$ 的规模如何变化，$|EIP|$ 的结果只增不减。说明采用这些算法的企业可以获得远超用户想象规模的同意目的来采集数据。

2）尽管方法 4, 5 都采取了拒绝优先原则，但随着 $|PIP|$ 规模的增加，这 2 种方法计算出来的 $|EIP|$ 趋势未受到显著的影响，说明单一向下继承机制会导致过度同意的现象，无论是 *G* 关系还是 *F* 关系。

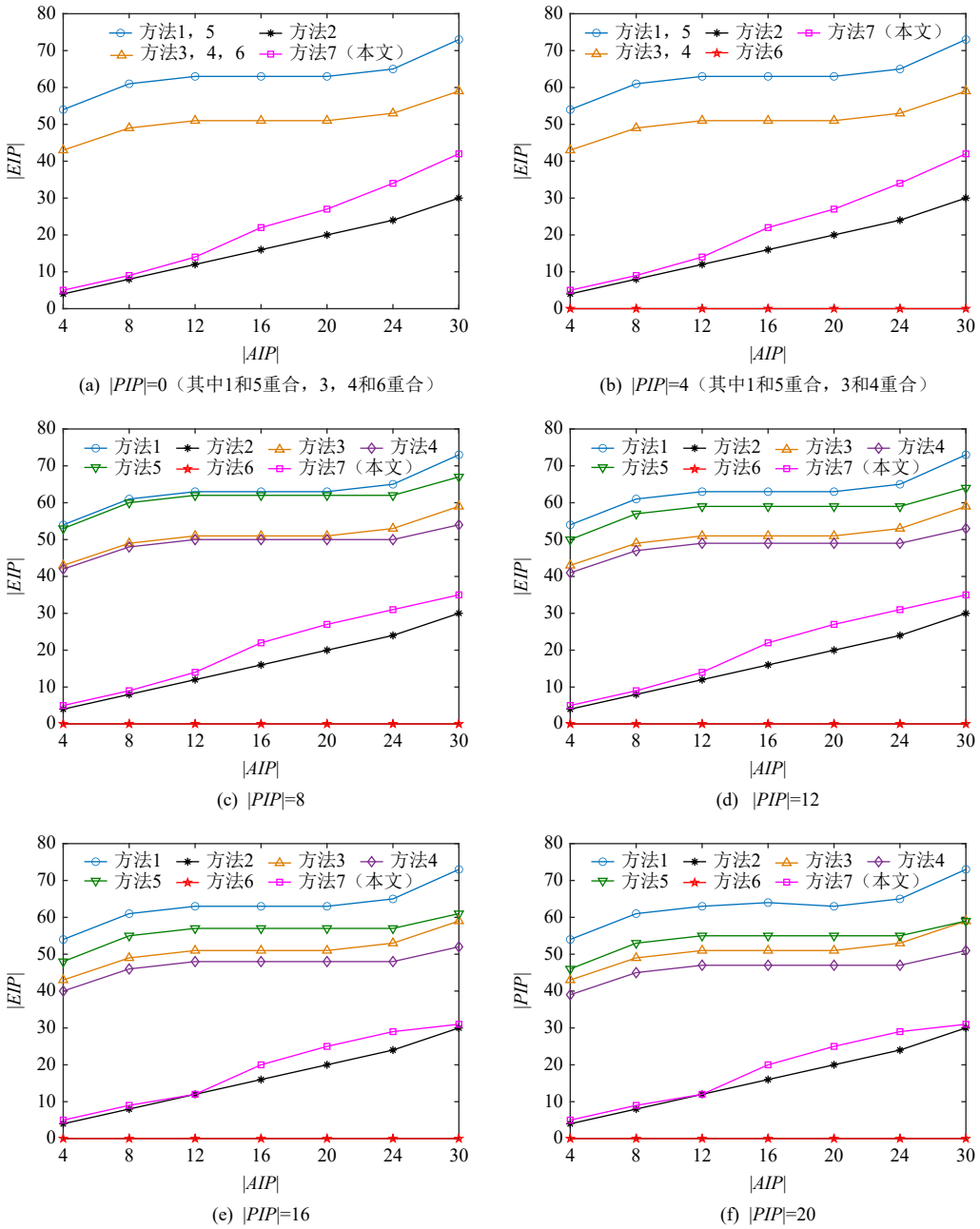
3）方法 2 没有继承机制，直接将用户允许的目的作为最终的同意结果。虽然这种思想比较符合 GDPR 只收集“明确同意”的要求，但没有考虑禁止目的造成的冲突。在实际应用中冲突问题没有解决，会导致同意难以判断。

4）方法 6 除了图 8（a）表现与 1, 3, 4, 5 相同的趋势之外，其余的实验结果均为 0（与横坐标重合），说明该方法自身存在矛盾。由于方法 6 未给出禁止目的合理的语义解释，导致“禁止目的的父目的是否也是禁止目的”这个问题的答案十分模糊。如果回答是，则根据方法 6 向上和向下 2 个方向的继承机制，父目的会将“拒绝”向上传递给其父目的，同时向下继续传递给其子目的，最终导致整颗目的树被拒绝（本实验的结果证明了这一点）；如果回答否，但该父目的本身又被包含在禁止目的集合中，则自相矛盾。

5）方法 7 表明基于 *G* 和 *F* 同意继承理论的细粒度计算之后，$|EIP|$ 的规模始终与用户明确同意的意图保持一致，避免出现过度同意现象。因为方法 7 考虑了同意冲突，虽然与方法 2 的结果比较相近，但能够给出准确的同意结果，相比之下效果更加理想。说明可以通过更细粒度的算法控制数据的滥用。

6）表 5 显示方法 8 没有探讨目的之间的关系，缺乏细粒度的同意计算方法，因此这里没有给出相应的实验对比。

综上，CBAC 模型给出了 *G* 关系和 *F* 关系 2 种目的层次在同意继承上的具体解释，对用户的隐私意图给出细粒度的同意判断，并且依据拒绝优先原则解决冲突问题。相比其他同意计算方法，CBAC 在不



(a) |*PIP*|=0（其中1和5重合，3，4和6重合）　(b) |*PIP*|=4（其中1和5重合，3和4重合）

(c) |*PIP*|=8　(d) |*PIP*|=12

(e) |*PIP*|=16　(f) |*PIP*|=20

Fig. 8　Experimental comparison between our study and existing research on consent calculation results

图 8　本文与现有研究在同意计算结果上的实验对比

增加用户额外负担的前提下，能够更精确地反映个性化隐私偏好，降低“过度同意”与“过度拒绝”的风险，对提高系统的合规性与可用性具有实际意义。

7　结束语

本文提出的 CBAC 模型通过形式化目的—动作—同意的语义关联并区分概括（*G*）与前提（*F*）两类关系，给出了基于非对称继承的同意计算与冲突消解规则，并以 TLA+规约与模型检测验证了若干关键安全性质。理论分析与实验表明，CBAC 在语义明确性、继承一致性与形式化可验证性方面较现有方法具有综合优势。在未来工作中将重点推进按数据类型的细化策略、自动化目的图抽取及更大规模的工程化评估。

作者贡献声明：马丽负责模型设计、实验、论文撰写及修改；王昌晶负责整体规划并提出指导意见；章琳和甘丽新负责隐私政策文本的语义分析及目的图的构建。

参 考 文 献

[1] Agrawal R, Kiernan J, Srikant R, et al. Hippocratic databases [C]//Proc of the 28th Int Conf on Very Large Databases. San Francisco, CA: Morgan Kaufmann, 2002: 143−154

[2] Lessig L. Code and Other Laws of Cyberspace[M]. New York: Basic Books, 1999

[3] Zhao Jingxin, Yue Xinghui, Feng Chongpeng. Survey of data privacy security based on general data protection regulation[J]. Journal of Computer Research and Development, 2022, 59(10): 2130−2163 (in Chinese)
(赵景欣, 岳星辉, 冯崇朋. 基于通用数据保护条例的数据隐私安全综述[J]. 计算机研究与发展, 2022, 59(10): 2130−2163)

[4] Daoudagh S, Marchetti E, Savarino V, et al. Data protection by design in the context of smart cities: A consent and access control proposal[J]. Sensors 2021, 21(21): 7154

[5] Tokas S, Owe O. A formal framework for consent management [G] // LNCS 12159: Proc of the 40th IFIP WG 6.1 Int Conf on Formal Techniques for Distributed Objects, Components, and Systems. Berlin: Springer, 2020: 169−186

[6] Pesch P J. Drivers and obstacles for the adoption of consent management solutions by ad-tech providers[C]//Proc of the 2021 IEEE European Symp on Security and Privacy Workshops (EuroS&PW). Piscataway, NJ: IEEE, 2021: 269−277

[7] Fieschi A, Hirmer P, Stach C, et al. Characterising and categorising anonymization techniques: A literature-based approach [C] //Proc of the 11th Int Conf on Information Systems Security and Privacy (ICISSP). Porto, Portugal: SCITEPRESS 2025: 107−118

[8] Zhao Yuqi, Yang Min. Review of differential privacy research[J]. Computer Science, 2023, 50(4): 265−276(in Chinese)
(赵禹齐, 杨敏. 差分隐私研究进展综述[J]. 计算机科学, 2023, 50(4): 265−276)

[9] Liu Fengxia, Zheng Zhiming, Shi Yexuan, et al. A survey on federated learning: A perspective from multi-party computation[J]. Frontiers of Computer Science, 2024, 18(1): 181336

[10] Podschwadt R, Takabi D, Hu Peizhao, et al. A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption[J]. IEEE Access, 2022, 10: 117477−117500

[11] Byun J W, Bertino E, Li Ninghui. Purpose based access control of complex data for privacy protection [C] // Proc of the 10th ACM Symp on Access Control Models and Technologies. New York: ACM, 2005: 102−110

[12] Byun J W, Li Ninghui. Purpose based access control for privacy protection in relational database systems[J]. The VLDB Journal, 2008, 17(4): 603−619

[13] Ni Qun, Bertino E, Lobo J, et al. Privacy-aware role-based access control [J]. ACM Transactions on Information and System Security, 2010, 13(3): Article 24

[14] Kabir M E, Wang Hua, Bertino E. A role-involved purpose-based access control model[J]. Information Systems Frontiers, 2012, 14(3): 809−822

[15] Colombo P, Ferrari E. Enforcement of purpose based access control within relational database management systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(11): 2703−2716

[16] Masoumzadeh A, Joshi J B. PuRBAC: Purpose-aware role-based access control [C] // Proc of the 2008 OTM Confederated Int Conf on the Move to Meaningful Internet Systems. Berlin: Springer, 2008: 1104−1121

[17] Pallas F, Ulbricht M R, Tai S, et al. Towards application-layer purpose-based access control [C] // Proc of the 35th Annual ACM Symp on Applied Computing. New York: ACM, 2020: 1288−1296

[18] Kebede M G, van Binsbergen T, van Engers T, et al. Towards a purpose-based access control model derived from the purpose limitation principle [C] // Proc of the 36th Annual Conf on Legal Knowledge and Information Systems: JURIX 2023. London: SAGE, 2023: 143−148

[19] Molina M, Betarte G, Luna C. Consent validation for personal data access control using ABAC [C] // Proc of the 13th Latin-American Symp on Dependable and Secure Computing. New York: ACM, 2024: 30−31

[20] Popoola O, Rodrigues M, Marchang J, et al. A critical literature review of security and privacy in smart home healthcare schemes adopting IoT & blockchain: Problems, challenges and solutions[J]. Blockchain: Research and Applications, 2024, 5(2): 100178

[21] Ma Li, Jiang Huowen, Peng Yun. Formalization and verification of privacy preserving protocol based on user consent [J]. Acta Electronica Sinica, 2023, 51(7): 1842−1849(in Chinese)
(马丽, 姜火文, 彭云. 基于用户同意的隐私保护协议形式化描述与验证[J], 电子学报, 2023, 51(7): 1842−1849)

[22] Hjerppe K, Ruohonen J, Leppänen V. Extracting LPL privacy policy purposes from annotated web service source code[J]. Software and Systems Modeling, 2023, 22(1): 331−349

[23] Daudén-Esmel C, Castellà-Roca J, Viejo A. Blockchain-based access control system for efficient and GDPR-compliant personal data management[J]. Computer Communications, 2024, 214: 67−87

[24] Robol M, Breaux T D, Paja E, et al. Consent verification monitoring [J]. ACM Transactions on Software Engineering and Methodology, 2023, 32(1): Article 2

[25] Park H, Oh H, Choi J K. A consent-based privacy-compliant personal data-sharing system[J]. IEEE Access, 2023, 11: 95912−95927

[26] Krstić S, Nguyen H, Basin D. Model-driven privacy[J]. Proceedings on Privacy Enhancing Technologies, 2024, 2024(1): 314−329

[27] Hajszan T, Staudinger M, Miksa T. You shall not pass (without consent): Enforcing data sovereignty with solid pods [J]. ACM Transactions on the Web , 2026, 20(1): Article 12

[28] Xue Tao, Wen Yu, Luo Bo, et al. SparkAC: Fine-grained access control in Spark for secure data sharing and analytics[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 20(2): 1104−1123

[29] Patil S K, Sagare S C, Shelar A S. Use of purpose and role-based access control mechanisms to protect data within rdbms[J]. International Journal of Software Innovation, 2020, 8(1): 82−91

[30] Sultan S, Jensen C D. Ensuring purpose limitation in large-scale infrastructures with rovenance-enabled access control[J]. Sensors

2021, 21(9): 3041

[31] Olca E, Can O. DICON: A domain-independent consent management for personal data protection[J]. IEEE Access, 2022, 10: 95479–95497

[32] Xu Zhiyu, Jiao Tengyun, Wang Ziyuan, et al. AC2M: An automated consent management model for blockchain financial services platform[C] // Proc of the 2021 IEEE Int Conf Smart Data Services (SMDS). Piscataway, NJ: IEEE, 2021: 33–41

[33] Román-Martínez I, Calvillo-Arbizu J, Mayor-Gallego V J, et al. Blockchain-based service-oriented architecture for consent management, access control, and auditing[J]. IEEE Access, 2023, 11: 12727−12741

[34] Pathmabandu C, Grundy J, Chhetri M B, et al. ICME: An informed consent management engine for conformance in smart building environments [C] // Proc of the 29th ACM Joint Meeting on European Software Engineering Conf and Symp on the Foundations of Software Engineering. New York: ACM, 2021: 1545−1549

[35] Debackere L, Colpaert P, Taelman R, et al. A policy-oriented architecture for enforcing consent in solid [C] //Proc of the Web Conf 2022. New York: ACM, 2022: 516−524

[36] Haring M, Tiefenau E, Tiefenau C, et al. I have not understood but agree: Studying informed consent in the context of the German COVID−19 contact tracing app [J]. ACM Transactions on Computer-Human Interaction, 2025, 32(3): Article 30

[37] Konstantinidis G, Holt J, Chapman A. Enabling personal consent in databases[J]. Proceedings of the VLDB Endowment, 2021, 15(2): 375−387

[38] Abrusio J. The (in)efficacy of consent for the processing of personal data[J]. Humanities and Rights Global Network Journal. 2024, 6(1): 180−205

[39] Harkous H, Fawaz K, Lebret R, et al. Polisis: Automated analysis and presentation of privacy policies using deep learning [C] // Proc of the 27th USENIX Conf on Security Symp. Berkeley, CA: USENIX Association, 2018: 531−548

[40] Sassetti G, Sharif A, Sciarretta G, et al. Assurance, consent and access control for privacy-aware OIDC deployments [C] // Proc of the 37th Annual IFIP WG 11.3 Conf: Data and Applications Security and Privacy. Berlin: Springer, 2023: 203−222

[41] Rohini K R, Rajakumar P S, Geetha S. Smart patient consent management model for health information exchange based on blockchain technology[J]. Journal of Computer Science, 2024, 20(7): 730−741

[42] Lamport L. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers [M]. Boston: Addison-Wesley Publishing Company, Incorporated, 2002

[43] Rissanen E. eXtensible Access Control Markup Language (XACML) Version 3.0 [S]. Billerica: Organization for the Advancement of Structured Information Standards (OASIS), 2013

[44] Diaz, J P, Mendoza, F A. Authorization models for IoT environments: A survey[J]. Internet of Things, 2025, 29: 101430

[45] Douyin Data Security and Privacy Protection Center. Douyin privacy policy[EB/OL]. [2023-10-05]. https://aweme.snssdk.com/draft/douyin_agreement/privacy.html?(in Chinese)
(抖音数据安全与隐私保护中心. 抖音隐私政策[EB/OL]. [2023-10-05]. https://aweme.snssdk.com/draft/douyin_agreement/privacy.html?)

[46] Meyer J J C. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic[J]. Notre Dame Journal of Formal Logic, 1988, 29(1): 109−136



Ma Li, born in 1974. PhD, lecturer. Member of CCF. Her main research interests include access control and privacy preserving.

马　丽, 1974 年生。博士，讲师。CCF 会员。主要研究方向为访问控制、隐私保护。



Wang Changjing, born in 1977. PhD, professor. Distinguished member of CCF. His main research interests include high-confidence software and intelligent software.

王昌晶, 1977 年生。博士，教授。CCF 杰出会员。主要研究方向为高可信软件、智能化软件。



Zhang Lin, born in 1982. PhD, professor. Her main research interests include artificial intelligence, pattern recognition, and image processing.

章　琳, 1982 年生。博士，教授。主要研究方向为人工智能、模式识别、图像处理。



Gan Lixin, born in 1982. PhD, associate professor. Member of CCF. Her main research interests include natural language processing, data mining, and image processing.

甘丽新, 1982 年生。博士，副教授。CCF 会员。主要研究方向为自然语言处理、数据挖掘、图像处理。