(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119922024 A
(43)申请公布日 2025.05.02

(21)申请号 202510419271.9

(22)申请日 2025.04.03

(71)申请人 西华大学
地址 610039 四川省成都市金牛区土桥金周路999号

(72)发明人 刘志才 杨淋 熊玲 何王俊辰 李文豪 梁星宇 曾晟珂 陈鹏

(74)专利代理机构 成都正象知识产权代理有限公司 51252
专利代理师 杜燕

(51)Int.Cl.
H04L 9/40(2022.01)
H04L 9/32(2006.01)

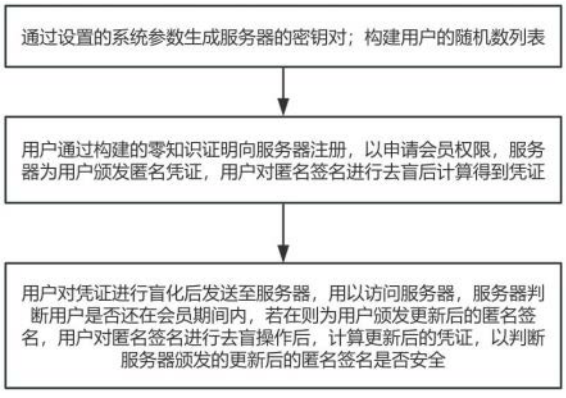
权利要求书3页 说明书5页 附图1页

(54)发明名称

面向大模型即服务的一体化匿名认证和分级访问方法

(57)摘要

本发明涉及隐私保护技术领域,公开面向大模型即服务的一体化匿名认证和分级访问方法,包括步骤:通过设置的系统参数生成服务器密钥对;构建用户的随机数列表;用户通过构建的零知识证明、承诺值和申请的权限向服务器注册,以申请会员权限,服务器为用户颁发匿名凭证,用户对匿名签名进行去盲后计算得到凭证;用户对凭证进行盲化后发送至服务器,用以访问服务器,服务器判断用户是否还在会员期间内,若在则为用户颁发更新后的匿名签名,用户对匿名签名进行去盲操作后,计算更新后的凭证,以判断服务器颁发的更新后的匿名签名是否安全。本发明通过不可链接性和匿名性,确保同一用户在不同会话中的认证信息无法被关联,能有效隐藏用户真实身份信息。



CN 119922024 A

1.面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:包括以下步骤:

步骤1,通过设置的系统参数生成服务器的密钥对;构建用户的随机数列表;

步骤2,用户通过构建的零知识证明向服务器注册,以申请会员权限,服务器为用户颁发匿名凭证,用户对匿名签名进行去盲后计算得到凭证;

步骤3,用户对凭证进行盲化后发送至服务器,用以访问服务器,服务器判断用户是否还在会员期间内,若在则为用户颁发更新后的匿名签名,用户对匿名签名进行去盲操作后,计算更新后的凭证,以判断服务器颁发的更新后的匿名签名是否安全。

2.根据权利要求1所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤1中,设置的系统参数包括:

给定安全参数1^{λ},λ用于确定系统的安全级别和相关参数的大小;

服务器生成双线性映射e:$G_1 \times G_2 \rightarrow G_T$,其中$G_1$、$G_2$和$G_T$均为p阶的循环群;$G_1$的生成元有g、w,$G_2$的生成元有$\tilde{g}$;

选择抗冲突哈希函数H:{0,1}*→Z_p*,其中Z_p*表示模p的整数乘法群;

公共参数表示为pp=(e,G_1,G_2,G_T,p,g,w,$\tilde{g}$,H)。

3.根据权利要求2所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤1中,通过设置的系统参数生成服务器的密钥对的步骤,包括:

服务器选择(x,y_1,y_2,y_3,z)←Z_p*,计算:

$$(X, Y_1, Y_2, Y_3) = (g^x, g^{y_1}, g^{y_2}, g^{y_3});$$

$$(\tilde{X}, \tilde{Y}_1, \tilde{Y}_2, \tilde{Y}_3) = (\tilde{g}^x, \tilde{g}^{y_1}, \tilde{g}^{y_2}, \tilde{g}^{y_3});$$

$$Z = w^{\frac{1}{z}};$$

服务器生成私钥SK_M=(X,y_1,y_2,y_3,z);

服务器生成公钥$PK_M = (Y_1, Y_2, Y_3, \tilde{X}, \tilde{Y}_1, \tilde{Y}_2, \tilde{Y}_3, Z)$。

4.根据权利要求3所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤1中,构建用户的随机数列表的步骤,包括:

根据用户Ui的剩余会员天数Ld,将随机数列表划分为12行,每行对应的剩余会员天数Ld为[0,30),[30,60),[60,90),...,[330,360),依次用TP1、TP2、...、TP12来表示剩余会员天数所在区间的类型;在初始状态下,用户Ui的随机数列表中每行的伪随机数d_{sid}均为空白。

5.根据权利要求4所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤2中,用户通过构建的零知识证明向服务器注册,以申请会员权限的步骤,包括:

用户Ui选择一个随机数t←Z_p*,以及自身的IDi;

计算承诺值:

$$cmt_{Ui,1} = g^t Y_1^{IDi};$$

构建零知识证明:

$$\prod_{Ui}^{1}: Pok\{(IDi): cmt_{Ui,1} = g^{t}Y_{1}^{IDi}\};$$

其中,Pok为用于解释集合的符号;

用户申请的权限为ACL,ACL代表用户申请的会员权限,用户支付相应权限的金额;

用户Ui将$cmt_{Ui,1}$、$\prod_{Ui}^{1}$、ACL发送给服务器。

6.根据权利要求5所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤2中,服务器为用户颁发匿名凭证的步骤,包括:

服务器接收到$cmt_{Ui,1}$、$\prod_{Ui}^{1}$、ACL后,验证零知识证明$\prod_{Ui}^{1}$,若验证成功则说明该用户是拥有对应ID的合法用户;

零知识证明验证通过后,判断用户Ui申请的权限与支付的金额是否匹配,若不匹配,则变更为用户实际支付金额对应的权限ACL`,并生成ACL`对应的访问天数val;若匹配,则生成ACL对应的天数val;

选择两个随机数$(r,u)\leftarrow Z_{p}^{*}$,计算伪随机数$d_{sid}=H(u,Lt)$,其中Lt为当前注册的时间,H为抗冲突哈希函数,令RT=(val,Lt);

服务器计算用户Ui的匿名签名$\hat{\sigma}$:

$$\hat{\sigma} = (\hat{\sigma_{1}}, \hat{\sigma_{2}}) = (g^{r}, (cmt_{Ui,1} \cdot XY_{2}^{d_{sid}}Y_{3}^{RT})^{r});$$

服务器将$\hat{\sigma}$、d_{sid}、RT发送给用户Ui。

7.根据权利要求6所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤2中,用户对匿名签名进行去盲后计算得到凭证的步骤,包括:

用户Ui接收到匿名签名$\hat{\sigma}$后,对其进行去盲操作,用户Ui计算凭证cred:

$$cred = (\sigma_{1}, \sigma_{2}) = (\hat{\sigma_{1}}, \hat{\sigma_{2}}(\hat{\sigma_{1}})^{-t}) = (g^{r}, (g^{t}Y_{1}^{IDi}X \cdot Y_{2}^{d_{sid}}Y_{3}^{RT})^{r}(g^{r})^{-t});$$

并通过检查下式是否成立来验证签名的合法性:

$$e(\sigma_{1}, \tilde{X}\tilde{Y}_{1}^{IDi}\tilde{Y}_{2}^{d_{sid}}\tilde{Y}_{3}^{RT}) = e(\sigma_{2}, \tilde{g});$$

其中,e代表双线性映射操作;

验证通过后用户Ui存储cred、d_{sid}、RT。

8.根据权利要求7所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤3中,用户对凭证进行盲化后发送至服务器,用以访问服务器的步骤,包括:

用户Ui选择三个随机数$(\alpha,\theta,\beta)\leftarrow Z_{p}^{*}$;

计算盲化凭证cred`:

$$cred` = (\sigma_{1}^{`}, \sigma_{2}^{`}) = (\sigma_{1}^{\alpha}, (\sigma_{2} \cdot \sigma_{1}^{\theta})^{\alpha});$$

计算承诺值:

$$cmt_{Ui,2} = g^{\beta}Y_{1}^{IDi};$$

生成一个零知识证明:

$$\prod_{Ui}^{2}: \mathrm{Pok}\{(IDi): cmt_{Ui,2} = g^{\beta} Y_1^{IDi}\};$$

用户Ui将cred`、$cmt_{Ui,2}$、$\prod_{Ui}^{2}$、$d_{sid}$、RT、问题query以及当前认证时间Lt`发送给服务器。

9.根据权利要求8所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤3中,服务器判断用户是否还在会员期间内,若在则为用户颁发更新后的匿名签名的步骤,包括:

服务器在接收到cred`、$cmt_{Ui,2}$、$\prod_{Ui}^{2}$、$d_{sid}$、RT、query、Lt`后,计算Lt`-Lt=d,其中d代表认证时间与注册时间之间的间隔天数,判断d是否超过val;

若d超过val,则提醒用户Ui会员已过期,需重新支付;

若d未超过val,则计算val-d=Ld,其中Ld代表用户Ui可访问服务器的剩余天数;检查随机数列表中剩余天数Ld的类型一行中的伪随机数d_{sid}是否存在,如果存在则代表该盲化凭证cred`是已经使用过的凭证,并拒绝提供服务;如果不存在则将d_{sid}加入所在类型中,并提供服务;

服务器验证零知识证明$\prod_{Ui}^{2}$;验证通过后,服务器选择两个随机数(ru`,k)←$Z_p$*更新$d_{sid}$,$d_{sid}$*=H(ru`,Lt),d_{sid}*为更新后的伪随机数;

服务器计算更新后的匿名签名$\hat{\sigma}$`:

$$\hat{\sigma}` = (\hat{\sigma}_1`, \hat{\sigma}_2`) = (g^{k}, (cmt_{Ui,2} XY_2^{d_{sid}*} Y_3^{RT})^{k});$$

并将问题query的答案和更新后的匿名签名$\hat{\sigma}$`、d_{sid}*、Ld返回给用户Ui。

10.根据权利要求9所述的面向大模型即服务的一体化匿名认证和分级访问方法,其特征在于:所述步骤3中,用户对匿名签名进行去盲操作后,计算更新后的凭证,以判断服务器颁发的更新后的匿名签名是否安全的步骤,包括:

用户Ui接收到匿名签名$\hat{\sigma}$`后,对其进行去盲操作,更新自己的凭证$cred* = (\sigma_1*, \sigma_2*)$,其中:

$$\sigma_1* = \hat{\sigma}_1` = g^{k};$$

$$\sigma_2* = \hat{\sigma}_2` \cdot \hat{\sigma}_1`^{-\beta} = (cmt_{Ui,2} XY_2^{d_{sid}*} Y_3^{RT})^{k} \cdot \hat{\sigma}_1`^{-\beta}$$

$$= (g^{\beta} Y_1^{IDi} XY_2^{d_{sid}*} Y_3^{RT})^{k} (g^{k})^{-\beta};$$

用户Ui检查下式是否成立:

$$e(\sigma_1*, \tilde{X}\tilde{Y}_1^{IDi}\tilde{Y}_2^{d_{sid}*}\tilde{Y}_3^{RT}) = e(\sigma_2*, \tilde{g});$$

若成立则用户Ui存储cred*、d_{sid}*、RT;如果不成立则说明服务器颁发的匿名签名不安全。

面向大模型即服务的一体化匿名认证和分级访问方法

技术领域

[0001] 本发明涉及隐私保护技术领域,特别涉及一种面向大模型即服务的一体化匿名认证和分级访问方法。

背景技术

[0002] 大型语言模型(LLMs,Large Language Models)在文本理解和生成任务中被广泛应用于各个领域,成为提升生产效率的重要工具。随着各大型语言模型的普及,用户隐私保护问题日益凸显,尤其是在用户与模型进行交互时,隐私泄漏的风险显著增加。用户通常依赖大型语言模型服务提供商所提供的在线AI即服务(AIaaS,AI as a Service)进行交互,这种模式下,用户的身份信息、交互数据以及行为模型可能会被服务提供商收集和存储。由于服务提供商能够通过用户的交互行为、搜索记录等数据构建详细的用户画像,用户的隐私安全面临严峻挑战。

发明内容

[0003] 本发明的目的在于通过盲签名机制,实现在注册和认证阶段对用户身份的双重保护特性,即不可链接性和匿名性,确保同一用户在不同会话中的认证信息无法被关联,并且能有效隐藏用户的真实身份信息,提供一种面向大模型即服务的一体化匿名认证和分级访问方法。

[0004] 为了实现上述发明目的,本发明实施例提供了以下技术方案:

面向大模型即服务的一体化匿名认证和分级访问方法,包括以下步骤:

步骤1,通过设置的系统参数生成服务器的密钥对;构建用户的随机数列表;

步骤2,用户通过构建的零知识证明向服务器注册,以申请会员权限,服务器为用户颁发匿名凭证,用户对匿名签名进行去盲后计算得到凭证;

步骤3,用户对凭证进行盲化后发送至服务器,用以访问服务器,服务器判断用户是否还在会员期间内,若在则为用户颁发更新后的匿名签名,用户对匿名签名进行去盲操作后,计算更新后的凭证,以判断服务器颁发的更新后的匿名签名是否安全。

[0005] 与现有技术相比,本发明的有益效果:匿名性,不可链接性,认证性。

[0006] 匿名性:当用户向服务器注册时,服务器会向用户颁发一个匿名签名,在用户使用该匿名签名进行认证之前,要对该匿名签名进行去盲操作,得到凭证cred,认证时使用盲化后的cred`进行认证,用户无需透露真实身份,即可完成注册和认证操作。

[0007] 不可链接性:在注册阶段,系统通过零知识证明技术保护用户的身份信息,使外来攻击者无法建立与用户的关联。在认证过程中,系统采用基于属性的匿名认证协议,确保攻击者无法通过认证过程追踪用户身份;同时,服务器端采用匿名签名保护机制,确保无法通过分析不同凭证来识别同一用户。

[0008] 认证性:是指验证用户的合法性和真实性,本方案中服务器对用户进行认证,确保只有合法用户才能访问其服务,从而在根本上保护系统的安全性;认证性是访问控制的核

心基础,只有通过认证的用户才能被赋予相应的权限,进而访问系统资源或执行特定操作。这一机制有效防止了非法用户的冒充和未经授权的访问,为系统的安全性和可靠性提供了重要保障。

附图说明

[0009] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本发明的某些实施例,因此不应被看作是对范围的限定,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他相关的附图。

[0010] 图1为本发明方法流程图。

具体实施方式

[0011] 下面将结合本发明实施例中附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。通常在此处附图中描述和示出的本发明实施例的组件可以以各种不同的配置来布置和设计。因此,以下对在附图中提供的本发明的实施例的详细描述并非旨在限制要求保护的本发明的范围,而是仅仅表示本发明的选定实施例。基于本发明的实施例,本领域技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0012] 应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。同时,在本发明的描述中,术语“第一”、“第二”等仅用于区分描述,而不能理解为指示或暗示相对重要性,或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。另外,术语“相连”、“连接”等可以是元件之间直接相连,也可以是经由其他元件的间接相连。

[0013] 实施例1:

本发明通过下述技术方案实现,如图1所示,面向大模型即服务的一体化匿名认证和分级访问方法,包括以下步骤:

步骤1,通过设置的系统参数生成服务器的密钥对;构建用户的随机数列表。

[0014] 步骤1-1,设置参数:

给定安全参数1^{λ},λ用于确定系统的安全级别和相关参数的大小,比如λ决定循环群的阶p的比特长度,还能确保抗冲突哈希函数的抗冲突性足够强。

[0015] 服务器(即大型语言模型服务器)生成双线性映射$e:G_1\times G_2\rightarrow G_T$,其中$G_1$、$G_2$和$G_T$均为p阶的循环群;$G_1$的生成元有g、w,$G_2$的生成元有 $\tilde{g}$。

[0016] 选择抗冲突哈希函数$H:\{0,1\}^{*}\rightarrow Z_p^{*}$,其中$Z_p^{*}$表示模p的整数乘法群,即集合{1,2,...,p-1},p是一个素数。

[0017] 公共参数表示为$pp=(e,G_1,G_2,G_T,p,g,w,\tilde{g},H)$。

[0018] 步骤1-2,生成密钥:

服务器选择$(x,y_1,y_2,y_3,z)\leftarrow Z_p^{*}$,计算:

$$(X,Y_1,Y_2,Y_3)=(g^{x},g^{y_1},g^{y_2},g^{y_3});$$

$$(\tilde{X},\tilde{Y}_1,\tilde{Y}_2,\tilde{Y}_3)=(\tilde{g}^{x},\tilde{g}^{y_1},\tilde{g}^{y_2},\tilde{g}^{y_3});$$

$$Z=w^{\frac{1}{z}};$$

服务器生成私钥$SK_M=(X,y_1,y_2,y_3,z)$；

服务器生成公钥$PK_M=(Y_1,Y_2,Y_3,\tilde{X},\tilde{Y}_1,\tilde{Y}_2,\tilde{Y}_3,Z)$。

[0019] 步骤1-3,生成随机数列表:根据用户Ui的剩余会员天数Ld,将随机数列表划分为12行,每行对应的剩余会员天数Ld为[0,30),[30,60),[60,90),...,[330,360),依次用类型TP1、TP2、...、TP12来表示,每行的更新时间依次为1月、2月、...、12月。在初始状态下,用户Ui的随机数列表中伪随机数d_{sid}均为空白。

[0020] 表1 随机数列表

类型	剩余会员天数 Ld	伪随机数d_{sid}	更新时间/月
TP1	0 ≪Ld<30	d_{sid}	1
TP2	30 ≪Ld<60	d_{sid}	2
TP3	60 ≪Ld<90	d_{sid}	3
TP4	90 ≪Ld<120	d_{sid}	4
TP5	120 ≪Ld<150	d_{sid}	5
TP6	150 ≪Ld<180	d_{sid}	6
TP7	180 ≪Ld<210	d_{sid}	7
TP8	210 ≪Ld<240	d_{sid}	8
TP9	240 ≪Ld<270	d_{sid}	9
TP10	270 ≪Ld<300	d_{sid}	10
TP11	300 ≪Ld<330	d_{sid}	11
TP12	330 ≪Ld<360	d_{sid}	12

步骤2,用户通过构建的零知识证明向服务器注册,以申请会员权限,服务器为用户颁发匿名凭证,用户对匿名签名进行去盲后计算得到凭证。

[0021] 步骤2-1,用户Ui选择一个随机数$t\leftarrow Z_p*$,以及自身的IDi(数字),计算承诺值:

$$cmt_{Ui,1}=g^{t}Y_1^{IDi};$$

构建零知识证明:

$$\prod_{Ui}^{1}:Pok\{(IDi):cmt_{Ui,1}=g^{t}Y_1^{IDi}\};$$

其中,Pok为用于解释集合的符号。

[0022] 用户申请的权限为ACL,ACL代表用户申请的会员权限,比如一日会员、月度会员、季度会员、年度会员;用户支付相应权限的金额。

[0023] 用户Ui将$cmt_{Ui,1}$、$\prod_{Ui}^{1}$、ACL发送给服务器。

[0024] 步骤2-2,服务器接收到$cmt_{Ui,1}$、$\prod_{Ui}^{1}$、ACL后,验证零知识证明$\prod_{Ui}^{1}$,若验证成功则说明该用户是拥有对应ID的合法用户。零知识证明验证通过后,判断用户Ui申请的权限与支付的金额是否匹配,若不匹配,则变更为用户实际支付金额对应的权限ACL`,并生成ACL`对应的访问天数val;若匹配,则生成ACL对应的天数val。选择两个随机数$(r,u)\leftarrow Z_p*$,计算伪随机数$d_{sid}=H(u,Lt)$,其中Lt为当前注册的时间,H为抗冲突哈希函数,令RT=(val,Lt)。然后服务器计算用户Ui的匿名签名$\hat{\sigma}$:

$$\hat{\sigma}=(\hat{\sigma_1},\hat{\sigma_2})=(g^{r},(cmt_{Ui,1}\cdot XY_2^{d_{sid}}Y_3^{RT})^{r});$$

即：

$$\hat{\sigma_1} = g^r;$$

$$\hat{\sigma_2} = (cmt_{Ui,1} \cdot XY_2^{d_{sid}} Y_3^{RT})^r;$$

服务器将$\hat{\sigma}$、d_{sid}、RT发送给用户Ui。

[0025] 步骤2-3,用户Ui接收到匿名签名$\hat{\sigma}$后,需要对其进行去盲操作,用户Ui计算凭证cred:

$$cred = (\sigma_1, \sigma_2) = (\hat{\sigma_1}, \hat{\sigma_2}(\hat{\sigma_1})^{-t}) = (g^r, (g^t Y_1^{IDi} X \cdot Y_2^{d_{sid}} Y_3^{RT})^r (g^r)^{-t});$$

并通过检查下式是否成立来验证签名的合法性：

$$e(\sigma_1, \tilde{X}\tilde{Y}_1^{IDi}\tilde{Y}_2^{d_{sid}}\tilde{Y}_3^{RT}) = e(\sigma_2, \tilde{g});$$

其中,e代表双线性映射操作;验证通过后用户Ui存储cred、d_{sid}、RT。

[0026] 步骤3,用户对凭证进行盲化后发送至服务器,用以访问服务器,服务器判断用户是否还在会员期间内,若在则为用户颁发更新后的匿名签名,用户对匿名签名进行去盲操作后,计算更新后的凭证,以判断服务器颁发的更新后的匿名签名是否安全。

[0027] 用户Ui在访问大型语言模型服务器时,将问题query以及凭证cred发送给给大型语言模型服务器,服务器对用户Ui的凭证cred验证通过后,则将问题的答案发回给用户Ui。详细来说：

步骤3-1,用户Ui选择三个随机数(α,θ,β)←Z_p*,并计算盲化凭证cred`：

$$cred` = (\sigma_1`, \sigma_2`) = (\sigma_1^{\alpha}, (\sigma_2 \cdot \sigma_1^{\theta})^{\alpha});$$

计算承诺值：

$$cmt_{Ui,2} = g^{\beta} Y_1^{IDi};$$

生成一个零知识证明：

$$\prod_{Ui}^{2}: Pok\{(IDi): cmt_{Ui,2} = g^{\beta} Y_1^{IDi}\};$$

用户Ui将cred`、$cmt_{Ui,2}$、$\prod_{Ui}^{2}$、$d_{sid}$、RT、query以及当前认证时间Lt`发送给服务器。

[0028] 步骤3-2,服务器在接收到cred`、$cmt_{Ui,2}$、$\prod_{Ui}^{2}$、$d_{sid}$、RT、query、Lt`后,计算Lt`-Lt=d,其中d代表认证时间与注册时间之间的间隔天数,判断d是否超过val;

若d超过val,则提醒用户Ui会员已过期,需重新支付;

若d未超过val,则计算val-d=Ld,其中Ld代表用户Ui可访问服务器的剩余天数;若0≤Ld<30,则检查随机数列表中类型TP1一行中的伪随机数d_{sid}是否存在,如果存在则代表该盲化凭证cred`是已经使用过的凭证,并拒绝提供服务;如果不存在则将d_{sid}加入TP1中,并提供服务;同理,若30≤Ld<60,则在TP2中执行相同操作;以此类推;若330≤Ld<360,则在TP12中执行相同操作。

[0029] 步骤3-3,服务器验证零知识证明$\prod_{Ui}^{2}$;验证通过后,服务器选择两个随机数(ru`,k)←$Z_p$*更新$d_{sid}$,$d_{sid}$*=H(ru`,Lt),d_{sid}*为更新后的伪随机数;服务器计算更新后的匿名签

名$\hat{\sigma}`$：

$$\hat{\sigma}` = (\hat{\sigma}_1`, \hat{\sigma}_2`) = (g^k, (cmt_{Ui,2} XY_2^{d_{sid}*} Y_3^{RT})^k);$$

并将query的答案answer和更新后的匿名签名$\hat{\sigma}`$、$d_{sid}*$、Ld返回给用户Ui。

[0030] 步骤3-4,用户Ui接收到匿名签名$\hat{\sigma}`$后,需要对其进行去盲操作,更新自己的凭证$cred *= (\sigma_1 *, \sigma_2 *)$,其中：

$$\sigma_1 *= \hat{\sigma}_1` = g^k;$$

$$\sigma_2 *= \hat{\sigma}_2` \cdot \hat{\sigma}_1`^{-\beta} = (cmt_{Ui,2} XY_2^{d_{sid}*} Y_3^{RT})^k \cdot \hat{\sigma}_1`^{-\beta}$$

$$= (g^{\beta} Y_1^{IDi} XY_2^{d_{sid}*} Y_3^{RT})^k (g^k)^{-\beta};$$

用户Ui检查下式是否成立：

$$e(\sigma_1 *, \tilde{X}\tilde{Y}_1^{IDi}\tilde{Y}_2^{d_{sid}*}\tilde{Y}_3^{RT}) = e(\sigma_2 *, \tilde{g});$$

若成立则用户Ui存储cred*、$d_{sid}*$、RT;如果不成立则说明服务器颁发的匿名签名有问题,比如被篡改、被攻击等。

[0031] 综上所述,当用户向大型语言模型服务器注册时,服务器会向用户匿名颁发一个匿名签名,在用户使用该匿名签名进行认证之前,会对匿名签名进行去盲操作,得到凭证。使用去盲后的凭证去进行认证,在认证的时候,用户会添加一个零知识证明,用于证明用户的身份的合法性,而不需要透露用户的真实身份,从而实现匿名认证,确保认证过程中使用的签名与用户的真实身份之间无法建立任何链接,从而有效防止用户身份被追踪或揭示。通过这种方式,匿名认证技术不仅确保了用户身份的匿名性,还实现了用户在不同交互场景中的不可链接性和认证效率提升,这种设计显著提升了用户隐私保护的水平,尤其是在大模型交互等高隐私风险场景中,能够有效防止用户身份信息被滥用或泄露。

[0032] 基于随机化凭证的匿名认证机制(如零知识证明、匿名签名),用户每次访问生成唯一伪身份,相同的用户在不同的查询中,使用的身份是不同的。即使获取了用户的某次的查询信息,也无法链接到用户的真实身份。保护了用户的隐私,防止用户身份的跟踪模式和行为数据被滥用,本方法认证机制,基于盲签名技术实现,确保用户在与大模型交互时,其身份信息和行为数据无法被链接或追踪,有效解决现有基于云的大型语言模型中服务提供商会揭示用户真实身份。该方法通过动态生成不可链接的匿名凭证,使得每次用户访问大模型时都使用匿名的、无法关联的认证标识。用户的隐私数据得到了更高级别的保护,同时确保了大模型服务的安全性和可信性。

[0033] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

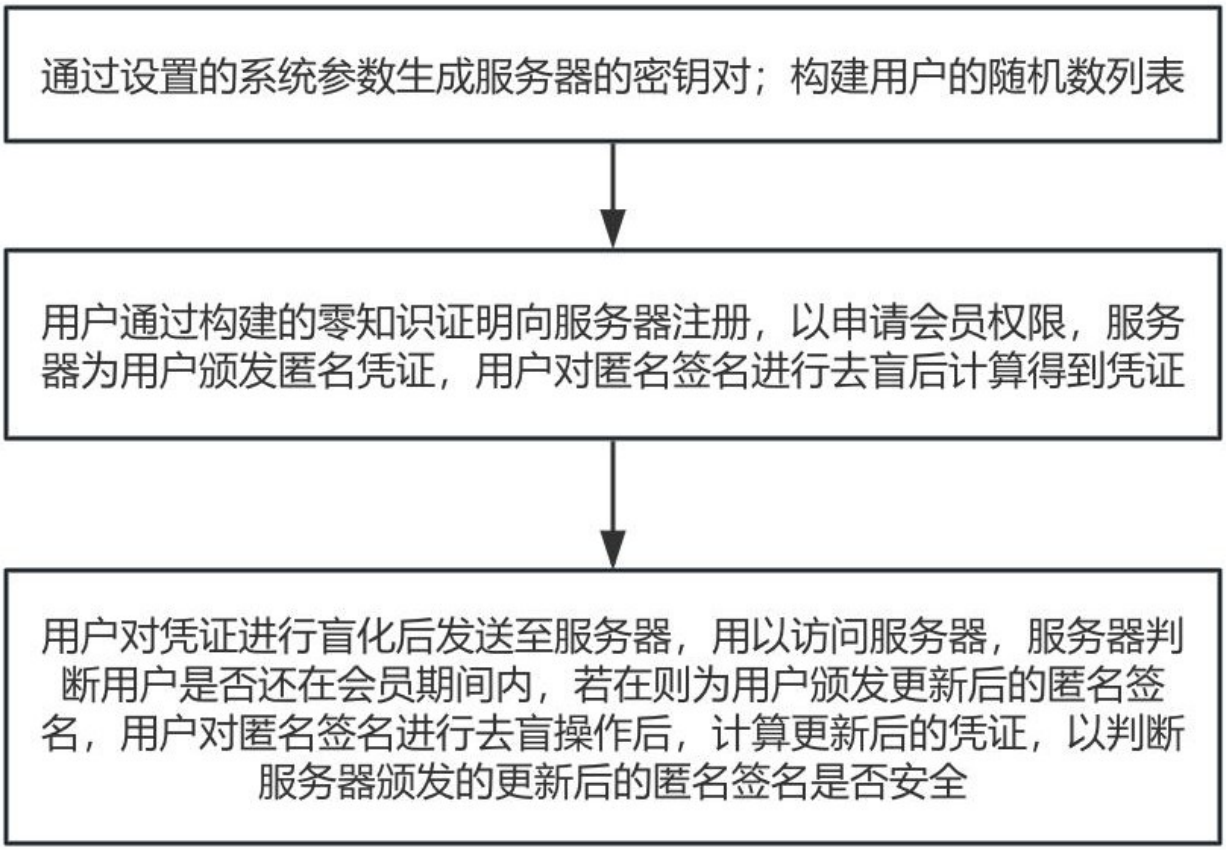

通过设置的系统参数生成服务器的密钥对；构建用户的随机数列表
用户通过构建的零知识证明向服务器注册，以申请会员权限，服务器为用户颁发匿名凭证，用户对匿名签名进行去盲后计算得到凭证
用户对凭证进行盲化后发送至服务器，用以访问服务器，服务器判断用户是否还在会员期间内，若在则为用户颁发更新后的匿名签名，用户对匿名签名进行去盲操作后，计算更新后的凭证，以判断服务器颁发的更新后的匿名签名是否安全

图 1