非交互式分布式密钥生成和密钥重新共享

詹斯。格罗斯[1]

jens@dfinity.org
缺陷基础

草案
2021 年 03 月 16 日

摘要。我们提出了一种非交互的公开可验证的秘密共享方案，其中经销商可以构建字段元素的秘密共享，并秘密但可验证地将共享分配给多个接收器。我们还开发了一个非交互式的公开可验证的重新共享方案，其中现有的秘密共享股份持有者可以创建一个新的秘密共享同一秘密，并以机密但可验证的方式将其分发给一组接收者。

公钥可以与以提升到秘密字段元素的组元素的形式共享的秘密相关联。我们使用我们的可验证的秘密共享方案来构建一个非交互式分布式密钥生成协议，它创建这样一个公钥以及离散对数的秘密共享。我们还构建了一个非交互式分布式重共享协议，它保留了公钥，但创建了密钥，并将其交给一组接收器，这可能与原始股份持有人重叠，也可能不会重叠。

我们的协议建立在一个新的基于配对的 CCA 安全公钥加密方案与向前保密。因此，我们的协议可以为参与者使用静态公钥，但仍然提供妥协保护。该方案使用分块加密，这是有代价的，但成本被我们的密文仅由源组元素而没有目标组元素组成所获得的节省所抵消。在我们的协议中，通过扩展到单接收机加密方案，可以进一步节省效率，其中密文比单接收机密文小 5 倍。

非交互式密钥管理协议部署在互联网计算机上，以方便使用阈值 BLS 签名。该协议提供了一个简单的接口，可以远程创建为一组接收器创建密钥共享密钥，在密钥持有者发生更改时刷新秘密共享，并提供针对移动对手的主动安全。

1、产品简介

因特网计算机托管运行子网（碎片）的节点集群，这些节点托管被称为容器的有限状态机（高级智能合同）。子网使用 BLS 签名对与最终用户之间的所有通信进行身份验证，这是自动发生的，因此单元开发人员不需要直接处理身份验证机制。运行子网的节点使用阈值加密技术来创建 BLS 签名，这样子网就可以有弹性地危及几个节点。互联网计算机通过分配每个子网一个永久的公钥，使密钥管理很容易。运行子网的节点可能会随着时间的推移而改变，因为其中一些节点被删除进行维护或升级，或者有其他原因来改变互联网计算机的拓扑结构，但即使在这些更改下，子网也有相同的公钥。这项工作背后的动机是开发密钥管理协议，使它容易旋转一个新的子网和提供参与节点的秘密共享子网的签名密钥，并使现有参与者容易重新共享的密钥新节点的组成子网的变化。希望这些协议都是非交互式的，因为它消除了传入节点在加入子网之前参与该协议的需要，并支持在互联网

等异步网络上的部署。

1.1 我们的贡献

我们构建了非交互式分布式密钥生成和密钥重新共享协议，支持密钥的密钥。分布式密钥生成协议允许一组经销商参与对阈值 BLS 签名方案的公共验证密钥的创建，以及该密钥在一组接收器下加密的密钥的秘密共享。一组已经持有密钥秘密共享密钥的节点可以使用分布式密钥重共享协议充当经销商，创建同一密钥的新随机秘密共享，并加密接收节点的公钥下的共享。

这些协议被设计为非交互式的，这简化了它们的使用和通信模式。每个经销商在不与其他参与者互动的情况下创建一个交易，并发布它。每个接收方都可以验证一个交易是否正确，而无需与其他参与者进行交互。给定一组商定的已验证的交易，接收方可以推导出 BLS 签名方案的公钥，并检索他们共享的秘密签名密钥。

我们的协议背后的基本思想是使用现有的信息理论方案进行沙米尔秘密共享和重新共享和加密消息。然后，我们使用非交互式的零知识证明来确保每个处理，本质上是一批密文，都是正确的。由于任何人都可以验证 NIZK 证据，这意味着交易是可公开验证的，每个人都会就交易是否有效达成一致。这消除了在现有中出现的可验证的秘密共享方案中的交互需要，只有接收器可以验证她的份额是否有效，因此需要抱怨如果经销商给她一个坏的份额，这反过来要求所有参与者等待，看看谁抱怨，是否有足够大的法定人数继续。

为了简化密钥管理，每个节点都有一个静态公加密密钥。然而，为了提供一些防止妥协的保护，我们使用具有正向保密的加密，其中时间被划分为时代，解密密钥可以进化，以便未来时代的密文可以被解密，但过去时代的密文不能被解密。构建基于正向保密的配对加密方案的一般范式是使用基于层次身份的加密方案，并根据树形层次推导出不同时代的密钥。这些前向安全加密方案使用目标组作为消息空间。然而，在我们的例子中，明文是构成沙米尔秘密共享的字段元素。如果我们直接在目标组中编码字段元素，NIZK 证明就会变得很笨拙。相反，我们将明文分成小块，可以在指数中加密，然后使用婴儿步巨步算法提取。虽然分块会产生开销，但它也可以对明文编码我们希望接收器获取和解码的目标组元素的源组预映像，并进行解码。为了探索这一想法，我们开发了一种新的基于配对的前向保密加密方案，其中密文由源群元素组成。由于源组元素比目标组元素要小，这就提供了一个显著的节省，抵消了分割明文所带来的开销。

我们新的前向安全加密方案有一个额外的好处，它的结构是，针对不同公共密钥的密文可能共享随机性。共享随机性提供了显著的性能改进，而一个块的单接收器密文由 3 个小源组元素和一个双尺寸源组元素组成，在许多接收器上摊销，我们可以将成本几乎降低到每个接收器的一个小源组元素。

最后，它也是促进有效的 NIZK 证明的一个设计目标，因为它们给验证器带来了计算负担。由于前向安全加密方案的加密算法主要使用指数，我们处于开发施诺尔式证明和应用菲亚特-沙米尔启发式使其非交互的有利环境中。我们得到的 NIZK 证明是紧凑的，并且小于密文。使用范围证明来证明加密块的大小是很自然的，但是范围证明构造起来很复杂。相反，我们选择了近似的范围的证明。在这

里，验证器不能保证一个块在一个给定的小范围内，只是保证这个块在一个足够小到可以被蛮力解密的集合中，也就是说，我们只是给出了可解密性证明。我们构造了一个新的、简单的可解密性证明。虽然证明系统本身很简单，但分析并不那么容易，因为证明系统没有完全的完整性，因此我们使用拒绝抽样来使完整性误差可以忽略不计。

1.2 相关工作

在全文中，我们将提供进一步的改进，并将我们的工作与关于秘密共享和分布式密钥生成的广泛文献进行比较。

2 初步工作

2.1 符号

在下面，我们写了 y：=x 来分配 y 的值 x。从一个概率分布中抽样，我们写下了 y.d。当 D 从一个集合中随机抽取样本时，我们写点。对于重复独立和均匀随机抽样，我们写 yi…，yU S.

设 A 是一种可以随机化的算法。我们编写 y.A(x)来分配 y 在输入 x 上运行的 a 的输出。如果该算法是确定性的，那么只有一个可能的输出，我们有时会写 y：=A(x)。我们也可以把随机算法看作是基于输入 x 和一些随机性 r 的确定性计算，在这种情况下，我们可以写出 y：=A(x；r)在讨论算法时，我们将经常要求它们是有效的。虽然上下文依赖于这意味着什么，所以我们不会提供效率的明确定义。.

我们写了 A对于一种可以访问 oracleO 的算法，它可以被认为是一个子例程。甲骨文为算法提供了一些接口，它可以发送输入并从甲骨文中返回输出。例如，如果 H 是一个哈希函数，我们可以写 A对于一种可以调用消息 m 上的哈希函数并返回消息摘要 H(m)的算法。

函数可以参数化，哈希函数例如具有输出长度入=256。通常，我们使用入来表示一个表示对象大小的参数，例如，A可以是整数 e 的位长度。这取决于上下文是更自然地考虑一个值作为协议的部分设置的参数，还是将它作为算法的显式输入更自然。

在考虑安全问题时，我们通常会为对手写 A。安全性是通过实验来定义的，其中我们精确地定义了攻击者对系统操作的能力和访问权限。我们说，对手的优势是它在实验中获胜的概率。举个例子，让我们找一个攻击者来猜测公平硬币的价值。这里的优势将是

Adv(A)=|2Pr[b*° A；bU(0,1}：b*=b]-1|，

出口额(A)	
b*u A b	//攻击者输出一个猜测
U(0,1}	//，我们扔了一个硬币
如果 b*=b 返回，	//T 表示已成功
其他返回	//丄表示出现失败

与符号中的隐式实验在一起
而对手的优势是 Adv(A)=|2Pr[Exp(A)=T]-1|。

我们有时会区分完美的安全，没有攻击者获得任何优势；攻击者不能获得显著优势的统计安全；而在计算安全方面，我们相信，基于特定的假设，没有实际的计算有界攻击者获得显著优势。

2.2 字段、组和配对

我们写了 Z 字 对于模 p 的整数。在本文中，p 将永远是一个已知的素数，因此 Zp 是一个有限域(有时也会写成 Fp)。我们假设 Zp 与场元素的规范表示为整数 0,1，…因此有有效的算法来计算场运算。我们为字段的乘法子群写 Z*，即，{1，……，p-1}。我们写 y：=xmodp 来分配给 y 的 x 模 p 的规范表示。

我们为一组已知的素阶 p 写 G。所有的素阶群都是循环的，所以使用乘法符号 G=(1，g，……，g^{1}}对于某些发电机 g，其中单位为 1=g。当我们提到群时，我们总是假设它们有已知的素阶、群元素的规范和紧凑表示，以及计算群运算和决定成员资格的有效算法。

我们使用基于配对的密码学，其中我们有两个源群 Gi，G2 和一个已知素数 p 的目标群 Gt。配对（在密码学术语中，偏离标准数学术语）是一个非退化双线性映射 e：GixG2TG。这意味着如果 gi、g2 是 Gi、G2 的生成器，那么 e(gi、g2)生成 Gt，对于所有 a、beZp，我们有 e(gf、g我们要求这个配对是有效的可计算的。出于安全目的,我们将假设源群 G1 和 G2 之间不存在非平凡的有效可计算的同态，也就是说,我们正在加尔布雷斯、帕特森和 Smart 的分类中使用 III 型对[IIS08]。我们遵循的约定是，G1 是具有组元素最紧凑的表示和最有效的计算的源群。

在实现中，我们使用 BLS12-381 实例化组大小-涇 2 的配对组 ^{255}。源群 Gi、G2 是椭圆曲线，具有大小为 q 涇 2 的基场 ^{38i} 目标群 Gt 是 F*12 的一个 p 阶乘法子群。1

2.3 哈希函数

我们依靠加密哈希函数来压缩数据。一个标准的密码哈希函数是一个有效的可计算的函数 H：(0,1}*t(0,1}ˆ它接受任意长度的输入，并将其映射到一个 a 位字符串，其中入是一个固定的参数。SHA-256 就是一个例子，它将任意长度的字符串散列成 256 位的字符串。[2]

但我们不会只是使用固定长度的输出。我们让 H 人：(0,1}*t(0,1}是一个将任意长度的输入映射到指定长度为 A 的字符串的哈希函数。我们有时也映射并让 Hg：(0,1}*tG 是一个哈希函数，它映射到组 G。我们可以映射到一个字段，并让 Hz：(0,1}Zp 是一个映射到 Zp 的哈希函数。我们隐式地假设在哈希函数中使用域分隔符，以确保它们特定于这里提出的方案，而不是在其他地方使用。

2.4 沙米尔的秘密共享和拉格朗日插值

阈值秘密共享使具有秘密 s 的经销商能够创建共享，这样任何 t 共享都足以计算秘密 s，而 t-1 共享则没有透露有关秘密的信息。

Shamir 秘密共享是一种流行的 Z 领域的秘密共享计划。这个想法是选择一个随机度 t-1 多项式 a(x)，这样一个（0）=，让共享是 si=a（1），…，s=a(n)。[3]这种秘密共享背后的想法是，平面上具有不同=坐标的任何 t 点都唯一地通过它们确定一个度 t-1 多项式 a(x)，并允许重建秘密的=a（0）。另一方面，给定具有不同非零 x 坐标的 t-1 点，它们可能有 p 多项式，每个都产生不同的秘密，所以这些点不会泄露关于秘密的任何信息。

为了正式地描述沙米尔秘密共享，让我们首先定义 Zp 上的拉格朗日插值多项

---

[1]有关选择组的一般信息，请参见 https:〃hackmd.io/@benj 小齿轮/bls12-381。

严格地说，域是长度高达 2 位的字符串，但这实际上相当于任意长度的字符串。

[3]将 Shamir 秘密共享推广到 Zp 中使用任何 n+1 不同的索引作为秘密和 n 共享是很简单的，但为了简单起见，我们只使用 0,1，…，文章中的 n。

式。给定一套，I={ii，　　有不同的指数
并且，我们定义了一个索引 ijeI

$$^{L}j^{(x)}=n\ 三^{\hat{}国防部}p.$$

我们观察到，所有的拉格朗日多项式都有“t-1”度，并满足 k=j 和 L%的(ik)=1(i)=0 中的 k=j。因此，给定的点和股份（ii，a(ii)），（it，a(it)），我们看到

$$a(x)=\quad Lf.(x)a(ij)元件\ p。$$

通过秘密共享，其中 sˆ=a(ii)，…，s=a(it)这意味着共享的秘密 s=a（0）可以重建为

$$s=£Lj（0）s\%modp。$$

它提供了以下（n、t）-Shamir 秘密共享的算法：

共享（n、t、s）T（si、...、s_n）：给定 eZp 设置一个°：=。选择 ai，…，在
-i　　*Zp*
并定义一个(x)=E 二。a&x国防部 p 返回(si，...、s)：=(a（1），...，a(n))。

重建(I，$机 Si)：给定一组不同的指数 1<ii<…<<并共享 Si1 ，返回

s：= Lf.(0)其他文件。

秘密分享给接收者带来了一个问题：她得到了正确的分享吗？交易商可能会给她一个与交易不对应的坏股份，或者给不同的接收人如此多的假股份，以至于他们与实际交易不对应。费尔德曼提出了可验证的秘密共享(VSS)来解决这个问题。他的计划使用了订单组 g。经销商将股票与公共组元素等=一起分配 g0,...，在=g 接收器我应该得到共享 Si=a(i)，现在可以检查，因为一个正确的共享满足

$g^{i}=g^{a(i)}=$

许多可验证的秘密共享方案都使用相关的想法。 n A.
然后，他们让接管人提出投诉，以防他的份额有误。事实证明，有一个单独的投诉阶段，我们将在密钥共享方案之上构建的密钥分发方案具有交互式。因此，我们将在本文的文章中构建一个可公开可验证的秘密共享(PVSS)方案，其中它可以立即对每个人进行验证，而不仅仅是接收方，无论共享是否正确。

2.5 重处理

假设我们已经有一个(n，t)-Shamir 秘密共享一个秘密，但希望将其转换为一个(n，t)-Shamir 秘密共享相同的秘密。为此，我们可以使用一个重新共享计划，该计划旨在由持有原始股份并将新的股份给 n'接收器。这个想法是，经销商我创建了一个(n，t)秘密子分享她的秘密分享。然后她给出了 n分股分配给各自的接管人。一旦接收器有了她的 t 子共享，她就可以使用拉格朗日插值将它们重新组合到一个新的秘密共享。在形式上，我们定义了以下打算在秘密共享上运行的重新共享方案)：

重新排列（n，t，s”：返回（s%i，...、s%/）-共享（n，t，s“。
组合(I，s^，…，s^j)：返回 sj-重建(I，s^j，…，s^j)。

重要的属性是所有正确的(n,t)秘密共享(s)一个秘密的 sezp,正整数 t‘<nJ,索引集 IC[1…n]大小的 t 和索引集 JC[1。n’]大小的 t'我们已经有了

“因为我本人：(siji，...，sj)-Reshare(n‘，f，s” -
(我，s^j，……，s^j) ：s‘=s=1.
年代-重建(J，sj，…，sj)

要看到这一点的一个关键观察结果是，拉格朗日插值是线性的。原始的秘密共享可以使用拉格朗日插值重新组合来重建这个秘密。但是，将相同的重建过程应用于子共享，也会产生相同秘密的秘密共享。因此，只要接收方知道交易商的指数，他们就可以使用拉格朗日插值在本地重建他们的新股票的份额。若要看到这一点，我们可以计算出：

s‘=£ £ （0）(£&(0&J
丸 eJ 如 eJ 电 I /

y? (o) I y? (°)膈 (o)Si =s

有几个特殊用途的重新共享方案，例如，如果经销商组与接收器组相同，他们可以联合创建 0 的秘密共享并将其添加到他们现有的共享中，以获得一个新的(n，t)秘密共享。或者他们可以通过在其股票中添加一个(n，t+1)秘密共享 0 来增加阈值。但为了简单起见，我们将只使用上面给出的重新共享方案。

2.6 施瓦茨拉链引理

稍后我们将开发零知识证明，其中一个共同的主题是测试多项式恒等式。施瓦茨-Zippel 引理在多项式恒等式的随机测试中是有用的，并指出对于多元多项式 f(xi，…，x_n)e Zp[xi，...，x_n任何集合的总度 $_p$

x、…、xn问：f(x1，...，xn)=0modp]<g。I^{s}I

3 个签名

为了实现完整性起见，我们回顾了签名方案(KGen、签名、签名)的安全定义。如果我们都有的话，这是完全正确的

Pr[（vk、sk）QKGen；aJ 符号（sk、m）：SigVfy（vk、m、a）=T]=1。

在自适应选择的消息攻击下，对手对抗强大存在的不可避免性的优势是

Adv(A)=Pr[（vk、sk）QKGen；（m*、a*）QA$_{a}$(vk)：签名（m*，a*）=T 和（m*，a**）eQ]，其

中甲骨文符号输入 m 返回 q 符号（sk，m）并存储 Q：=

Q U{(m，a)}。存在性不可避免性的标准概念是通过将成功条件放松到 m*牛 Q|m 来实现的，其中为 Q|是 Q 中的消息集。

3.1 BLS 签名

BLS 签名[BLS04]的工作原理如下：

设置：我们假设签名方案的所有用户都使用约定的公共参数，包括描述组 Gi、G2、已知素序 p 的 Gt、G2 的生成器 g2 和配对 e：GixG2TGt。这些参数还包括一个哈希函数：（0.1}*tG”
KGen：样品 sk.Z_q，设置 vk：= ，并返回（vk、sk）
标志（sk，m）：返回 a：= (m)斯克
签名（vk、m、a）：如果是 vkeG2，aEGi 和

e(H_a(m)，vk)=e(a，g2)

返回 T，否则将返回⊥。

我们很容易看到 BLS 的签名方案是完全正确的。稍后，当我们考虑 BLS 签名的阈值版本时，我们认为存在存在的不可避免性。然而，我们注意到重要的一点是，BLS 签名是唯一的。

独特性。如果满足验证算法的给定消息上最多有一个有效签名，则签名方案称为具有唯一签名。对于所有 vk、m、ai、a2 与 SigVfy(vk、m、ai)=T 和 SigVfy(vk、m、a2)=，必须是 ai=a？.

定理 1。BLS 签名方案具有唯一的签名。

证明文件。如果 SigVfy(vk、m、a)=T，我们有 vkeG2、aeGi 和 e(Hˆi(m)、vk)=(a、g2)。由于 g2 是 G2 的生成器，因此签名验证方程具有唯一的解 a，因此对于同一消息和验证密钥不能有两个不同的签名 ai=a2。 口

从签名的唯一性可知，对于 BLS 特征，存在不可避免性和强存在不可避免性是等价的。

存在不可的。在给定 gfg 的假设下，在随机 oracle 模型中是不可原谅的；和 g？很难计算，我们在这里不能证明，相反，我们将在第二节。8 证明了在不同假设下的阈值 BLS 签名的安全性。

4 个阈值签名

阈值签名方案使 n 个潜在贡献者中的 t 个签名者能够协作地签名一条消息。在本文中，我们将使用非交互式的阈值签名方案，其中每个签名者都可以自己在消息上生成一个签名共享。给定 t 签名共享，然后它们将被组合到消息上的数字签名。我们通过描述下面的组成有效算法来定义阈值签名的语法。阈值签名方案以参数和键作为输入。稍后，我们将描述用于生成参数和密钥的分布式密钥生成协议，但在这里，它们只是被认为是理所当然的。

VKVfy(t、vk、shvki、shvk)tb：确定阈值在具有 t 的密钥包上，验证密钥 vk 和共享验证密钥 shvk1 上，如果密钥包被认为有效，shvkn 返回 T，否则返回⊥。它只能在 t 时返回 t，n 是具有 t<n 的正整数

SKVfy(sk，shvk)t ⊥：确定性算法，如果 sk 被认为是关于共享验证密钥 Shvk 的有效的共享签名密钥，则在共享签名密钥上 sk 返回 T，否则返回⊥。

SigShare(sk，m)tsh：确定性或随机算法，给定共享签名密钥 sk 和消息 me(0,1}*生成签名共享 sh。

SigShVfy(shvk，m，sh)tb：确定性算法，给定共享验证密钥 shvk，如果签名共享有效，消息 m 和签名共享 sh 返回 T，否则返回⊥。

签名组合(I，Shi，…t：确定性算法，取一组不同的指标 Ii<…<它和 t 签名共享 Shi，…并将它们组合到一个签名 a。

SigVfy(vk、m、a)tb：确定性算法，给验证键 vk、消息 me(0、1}*和签名，如果签名有效，则返回 T，然后返回⊥。

正确性。在以下情况下，生成未指定密钥的阈值签名方案完全正确：

—有效的密钥包具有在正确范围内的阈值。如果 VKVfy(t，vk，shvki，...，shvkn)=T，那么 te[1…n]。

-有效的共享签名密钥将生成有效的签名共享。对于所有的 sk，shvk，m，SKVfy(sk，shvk)=T

Pr[sh-SigShare（sk、m）：SigShVfy（shvk、m、sh）=T]=1。

—为不同指数的阈值组合有效的股份可产生有效的签名。对于所有的 vk，shvki，...，shvkn，我，shi1，…，shˆt 其中 VKVfy(t,vk,shvki,...,shvkn)=T，我是一组不同的索引 1<ii<…<<，

对于所有我 I：SigShVfy(shvki，m，sh“=T

(I，sh^，…，shQ：SigVfy(vk，m，a)=T]=1。

独特性。唯一签名属性可以定义为标准签名方案，因为它仅依赖于 SigVfy。

不可性。我们在第 19 节中定义了不可原谅性。8 用于具有相关的分布式密钥生成算法的阈值签名方案。

4.1 BLS 阈值签名

我们现在描述了具有未指定密钥生成的 BLS 阈值签名。我们重用了标准 BLS 签名的签名验证算法，但启用了密钥被秘密共享到多个密钥的情况。这些共享签名密钥可以用于生成签名共享，并且如果有足够的签名共享，可以将它们组合到签名中。因此，以下算法可以看作是单一签名者 BLS 签名算法的一种替代方案。

设置：公共参数包括已知素数 p 的组 Gi、G2、Gt 和配对 e：GixG2TGt，以及 Gi、G2 的生成器 Gi、G2。这些参数还包括一个哈希函数 H_{G1}：(0,1}*t G「

VKVfy(t、vk、shvki，...、shvk)：检查 te[1…n]和 vk，shvki，...，shvkE G2.设置 shvko：=vk 和 I=(0，……，t-1}。对于 j=t，……，n 检查是否

$$日本=单位^{\iota}(7)$$
伊文

如果所有的检查都通过，则返回 T，否则将返回⊥。

SKVfy（sk、shvk）：如果是 skEZ和 shvk=返回 T，否则返回⊥ SigShare(sk，m)：返回.

sh：=Hgi(m)。

签名（shvk、m、sh）：如果 shvkEG2、shEGi 和

e（Hci(m)、shvk）=（sh、g2）

返回 T，否则将返回⊥。

签名组合功能（I，　，……，shi)：解析 I 作为一组不同的索引 ii<…<它作为 Gi 中的元素。返回

s%：

和希……，sh^

答：=和=。

$\iota\iota\zeta\ I$

签名（vk、m、a）：检查是否有 vkEG2、aEGi 和

e(Hi(m)，vk)=e(a，g2).

如果所有检查都通过，则返回 T，否则则返回⊥。

性能优化。与其使用拉格朗日插值，还可以通过检查 卩仁来对 vk，shvki，...，shvkn 进行快速随机检查？，在哪里(e，...，e)是一个随机向量，它正交于有效(n，t)秘密分割的离散对数空间。这使得验证是非确定性的，并且需要对定义进行轻微的修改，因为我们不再得到完全的正确性。

定理 2。BLS 阈值签名方案完全正确。

证明文件。如果七屯[1..n]在 VKVfy(t、vk、shvki、shvk)中的阈值检查）失败，验证返回丄。

让 sk，shvk(sk，sk，shvk)=T。这意味着 skeZ和 shvk=g2。因为 SigShare(sk，m)返回 sh=(m)我们现在有了这个

共享验证，即，

e(H(m)，*shvk*）=（*H*(m)，病）=e(H(m)，仞)=(sh，g

让 vk，shvki，...，shvkn 与一组不同索引 1<ii<…<t<，签名共享 shˆ，…，sh，和一个消息 m。如果 VKVfy(t，vk，shvki，...，shvkn)=T，它意味着有一个度 t-1 多项式 a(X)eZp[X]，这样 shvki=g“”fori=0，…，n 使用 shvko：=vk。我给我们的条件是(shi，shvki)=T，shi，shvkieg2 和 e(shi，g2)=(=(m‘g“”)，这意味着 shi=Hgi(m)。对于不有效的签名共享，签名共享组合算法将生成一个=riie/。自

n sh扩`=n(HGi(m)政
国际尔　　国际尔

我们看到了 e(H*i(m)，vk)=e(H*i(m)，g“)=e(Hi(m)，g2)=e(a，g2)。这意味着签名(vk、m、a)=T。口

5 具有正向保密密钥的公钥加密

在第 2 节中。7 我们提出了我们的分布式密钥生成和分布式密钥重共享协议。在这些协议中，经销商将向接收方加密秘密共享。所有各方都有长期的公共加密密钥，这使得管理公钥很容易，因为它们是静态的，但在其一生中带有妥协的风险。我们将通过使用前向保密的加密来部分减轻这种风险，这意味着如果对手学习了解密密钥，就有可能将未来的密码文解密给这个参与者，而不是过去的密文。对手可能发送恶意制作的密文是完全可信的，所以我们也希望加密方案是安全的，以抵御所选择的密文攻击。在本节中，我们将使用基于配对的加密技术，逐步构建一种新的 CCA 安全的多接收人公共密钥加密方案。

5.1 决定性的假设

我们的加密方案依赖于一个决策问题，即生成器 gi、g2、e(gi，g2、g2)，其中 e 是 III 型配对。我们假设这些群是在参数中一起给出的

与 fo，fiheG2 放在一起。我们在下面的实验中定义了决策问题，以及对手 A 对决策问题的优势，分别定义为 Adv(A)：=|Pr[Expo(A)=T-ExpJA)=T|。[4]

[4] 目的是 A 指定树的树的高度 叶子的叶子。稍后我们将定义基于树的加密，其中发送者将消息加密到树中的叶子；持有树中节点的解密密钥的接收方派生叶子的解密密钥，以获取明文。我们的加密方案可以通过选择树更大的分支因子来优化，例如，选择 4 个树将使公共参数大小减少到 1/2，解密密钥大小减少到 3/4。然而，决策假设和由此产生的加密方案变得有点难以描述。

费用(A)

T1，...，T\ (0,1}

x，r，s U Zp

对于每个 je[1。选择 pjUZp

／ *ri*、……、*T*，*g*，gi0、fo、..。，fnh，(fo *h*\

A「{gE(f.-厂广，f }J

下面的定理表明，假设有足够的熵在 fo，……从攻击者的角度来看，f\，h。

定理 3。决策假设适用于一般群模型中，当 fo，……，f\，h 是均匀随机选择的，即，任何只使用通用群操作的攻击者，其中有一定数量的优势可以忽略不计。

证明文件。在通用群模型中，攻击者可以将现有群元素乘以在同一组中构造新的群元素，使用配对将两个现有源组元素映射到目标组，并测试组中的相等性。由于可以使用配对将源组中的相等式提升到目标组，而不会失去一般性，我们可以在源组元素 Ai 上假设对手，…，A*eGi*、*Bi*、……、*B*eG2 和目标组中没有元素，构造了形式的对积等式

1=e(n AU 项，n Bj)...

让我说，……，a，bi，……，bn 是离散对数。等式适用当且仅当

0=£ui，iai£vj，ibj+----------+£“gai£bjmodp。

因此，我们可以推理对手在可用群元素的离散对数中基于这些二次方程学习有用信息的能力。

让加，……，机，n 是 fo 凡 h 的离散对数。现在我们还将证明，当 Gi 和 G 中的群元素的离散对数在©o，...、©人、n、x、r、s、pi 中被视为形式多项式时……(ci)，对手不能使用通用群操作来构造在 c 中不平凡的配对积等式。这是很容易看到的，当 c=ci，因为任何这样的双线性等式都可以被重写成这个形式

e(ci、*)=e（*、*） e（*，*），

其中未指定的条目不依赖于 ci。由于 ci 只出现在左边，当我们取离散对数时，我们得到了形式的形式等式

日志 ci。*= ... ，

其中形式变量 logci 只出现在左侧。为了使等式成为真，我们必须有左侧的形式对数 ci0 来取消形式变量对数，因此对手使用的配对积方程是 e(ci、1)=（*、*)*(*，*），没有其他条目。这意味着 ci 根本不被使用，因此平等没有告诉对手任何关于它的信息。现在我们开始证明对手不能使用通用群操作来构造形式 e(co、*)=e（*、*）e（*、*）的非平凡双线性等式。

当 cb=co=gX 时 对手可以使用 Gi 中的通用群操作来构造具有离散对数的组元素

$$u=u_c xr+ui+u_x x+Ur\ r+u_s s+ \qquad u_{Pj}pj$$

我[i...A]

对于已知的字段元素，u°，ui，ux，....使用 G2 中的通用群操作，对手可以构造具有离散对数形式的组元素

_ （ ʳf 工：=o ᵀⁿⁿᵀ⁰, n, r, s((©oˆ2i=i^{r}iˆi^{jr} + 〃ˢ)
\ ˆSjG[i..A](ˢj$^{(x+}$©oᵀ⁽¹⁻*Tj)©j^{p}j)ˆEi..A'ij©i^{p}j+^{s}nPj $\eta\rho_j$))

要在 co 中具有非平凡相等，我们可以不丧失一般性地假设它是 e 形式(co，*)的目标群中的相等_____e，如果右手边不涉及 c。所以在取了离散对数之后，问题是对手是否可以找到多元多项式 v___0 和 u(1)，v(1)，...such，那是

xrv 次 (1)

每个 u（ ‘）的你都在哪里_____ 0.现在我们将分析各种项的系数，以表明不存在这样的多项式，即，如果等式在不确定的 x、r、s、pi…，p 人然后 v___ 0.

术语 xrˆPj 可以在（1）的左侧出现为 xr-v ₙpj.对条款中可能出现的产品的检查 显 示 xrˆPj 不能出现在上
在（1）的右手边，当我们记住，通过设计的=0。这意味着这个词不能出现在右手边的任何地方的
(1).因此，术语 xrnpj 的唯一可能的系数是 0。我们得出结论，Vnpj=0 对于所有可能选择的 j。

术语 xr 如 pj 表示 ie[1...A]\(j}可以出现在（1）的左侧，作为 xr-vi，j 饥 pj。再次检查 En=i 中条款的产品 显示
不能出现在(1)的右侧。因此，术语 xrˆipj 的唯一可能系数为 0。我们得出结论，所有的 vij=0。

术语 xrˆpj 告诉我们所有 Vj=0。

术语 xrˆon告诉我们 vˆ，ˆ，r，s=0。

xrn 一词告诉我们，v $_{注册}$ = 0.

术语 xr©i 可以出现在（1)的左侧，也可以出现在(1）的右侧，作为? /0xv0((由 c-\‘丁。由。不过，我们也有，例如 UXP 号传 rs=0，给出了（1）右侧的 xs 的非平凡系数。美国产品条款的检验(表明这个项没有其他系数，所以我们的右手边具有作为 xs 的非平凡系数。由于（1）的左边是 xr-v，所以我们不能有一个术语 xs。因此，我们得出结论，v 饥=0 为所有 i=0，……，A。

以上分析显示，（1）左侧的许多系数为零，左侧剩下的是 vixr。我们只能点击右边的 xr，例如 v（(1）的幻与产品形式的 r-(x+(©o+k©j 丿 pj 丿。
所以必须至少有一个 j，我们有一个非平凡的和 w=例如，然而，这也给了我们 wr^opj 和 w(1-Tj 丿 r©jpj 在（1）的右侧。这些术语不出现在（1）的左侧，所以我们必须在右侧取消它们。我们可以使用这些术语的唯一另一种方法是通过形式的 pj- +丁渺讨+股份有限公司)。要取消操作后一种乘积的系数必须之和为-w。要取消术语 r 如 pj 的系数，那么我们需要 w(1-Tj 丿 r©jpj-wpjTj 饥 r=0)。这意味着 w=0，所以它毕竟不能不是平凡的。我们得出了 vi=0，因此在假设中定义群元素离散对数的形式多项式上的所有等式都排除了如的使用

现在的最后一个问题是，对手是否可以幸运并构造一个二次等式，它对于不确定的 Wo、、© 人、n、x、r、s、pi 中的形式多项式无效，……，p 人，但碰巧适用于挑战中的具体值。然而，正如在使用通用群模型的各种优步假设中所指出的那样，这意味着对手已经在计算值为零的不确定值中构造了一个非平凡的低度多元多项式。通过施瓦茨-Zippel 引理，这种情况发生的可能性可以忽略不计。口

5.2 二进制树加密

[5]我们首先为单个接收器定义二进制树加密。在 BTE 中，参数指定一个高度为 A 的二元树。加密算法采用一个明文，并将其加密到树中的一片叶子上。对每个叶可以关联一个解密密钥，并且该解密密钥的持有者可以恢复明文。还有一些与内部节点相关联的解密密钥。使用内部节点的解密密钥，可以派生出该节点的任何子节点的解密密钥。这意味着，如果您拥有根目录的解密密钥，那么您可以派生出所有叶子的解密密钥。但是，如果您没有根目录的解密密钥，您只能解密所解密密钥的子树中与叶子相关的密文。[6]

BTE 方案具有以下有效的算法：

设置：参数包括二叉树的消息空间 M 和高度 A。我们将写入树中节点的路径为...”与 t<A。因此，根节点有一个空路径，t=为 0，而对于叶 t=A.
KGent（pk、dk）：随机密钥生成算法，为树的根生成公用密钥和解密密钥
KVfy(pk)tb：确定性密钥验证算法，如果认为公钥有效，则返回 T，否则返回丄
衍生物(dk)$_{t1}$ ，”)t dk$_{t..}$：随机更新算法
...节点的解密密钥和一点 t 返回节点的解密密钥 ti…r^。如果 t>A 或“哒(0、1}或其他问题，派生算法返回丄。
我们将在整个论文中使用 dk 是解密密钥的约定，而 dk^，……丁仑是一对(“…”，dk)，明确表示节点 ti…“解密密钥属于。

---

[5] 更大的分支因素是可能的，并以定义和描述 BTE 的复杂性为代价进行适度的性能改进。

[6] 二进制树加密[CHK07]与基于层次身份的加密密切相关。主要的区别是，在 HIBE 中，恒等式可以是任意的字符串，而这里我们有一个非常小的“恒等式”空间。另一个小的区别是，在我们的工作中，我们总是加密到一个叶子，而不是树中的某个内部节点(由于这个原因，允许 0 标记叶子，例如。要求身份信息为非零）。

Enc(pk、m、ri、……、r)tc：随机加密算法，给定一个公共密钥、消息和一个叶子，在失败时返回一个密文或丄，例如，如果其中一个输入格式错误。

Dec(dkTi，...，T 入，c)tm：确定性解密算法，给定密文和解密密钥，在发生错误时返回明文 meM 或丄。

正确性。在以下情况下，树形加密方案完全正确：

-诚实生成的密钥验证为有效。

公关[（pk、dk）JKGen：KVfy(pk)=T]=1

-正确生成的密文解密到原始明文。I.e.，对于所有 mE e(0,1}

Pr，（pk、dk）-KGen；c-Enc（pk、m、tit 人） ... ♦ym

dkr—衍生物(dk)；……，dk“丁入-衍生物(dk..₇₁)；m₂—12 月(dk₁..，c)

选择明文攻击下随机叶的不可分辨性。对于一个有状态的对手 A，我们定义了它在选择的明文攻击下不可区分的优势为 Adv(A)=|Pr[Exp₀(A)=T]-Pr[Exp](A)=T|，实验所在

出口”

（*pk*、*dk*）——*KGen*

TI，……，*T 人©(0,1}*

对于 j=1，……，A

dkr..j_10-Derive(dk，1……写_1,0)

dkT1 ... Tj_11©~衍生物(dk) *TI ... Tj_1*，

(m0·m1) —A(Pk，T *I ... Tn*，(dkT1...Tj_1(1-Tj)jg[1..a])

交流（pk、mb、T1t 人）...

如果 b=b*返回 T，否则将返回丄

施工过程。

设置：参数指定配对组 G1、G2、Gt、消息空间 M=[-R..S]CZp它足够小，可以通过蛮力、组元素 fo、f1、...、f\、heG2 和树的高度入来搜索。

KGent(y，dk)：选择 x©Zp 并计算 y♦=gf。 选择 p©Zp，让 dk♦=(g，*g2f0，f*，...，f，*h*，返回方向（y，dk）。

KVfy(pk)tb：如果 pk=和 eG1 返回 T，否则返回丄

衍生产品(dkT1...互_1，%)tdk”...：给定

$$dk_{T1}\ldots T\pounds\_1=(T1\ \ldots\text{”}—1',a,b,\ dg,\ldots,\ dA,\ e)e(0,1\}\ X\ G1X\ G2\ +$$

$$dk_{\tau_1\ldots\tau_i}:\ \ (\tau_1\ldots\tau_\ell,a\cdot g_1^{\delta},b\cdot d_\ell^{\tau_\ell}\cdot(f_0\prod_{i=1}^{\ell}f_i^{\tau_i})^{\delta},d_{\ell-1}\cdot f_{\ell+1}^{\delta},\ldots,d_\lambda\cdot f_\lambda^{\delta},c\cdot h^{\delta}).$$

和 Tge(0,1}选择 6 个©Zp 并返回

... 环境（y、m、T1t 人）c：给定 yeG1、meM、T1t 人（0、1}... A 选择 r，s©并返回

c♦=”ge，gs，(为斗 f；jh，。

12c(dk”..T 入，c)Tm：解析

dk“..T 入=(T1　　a，b，e)e(0,1}^ 乂 G1 乂 G/

和 c=(C、R、S、Z)eG；xG/。断言 e（R、f）e（S、h）=（g_1，Z)。计算结果 M：=（C、gi）-e（R、b）-e(a，Z)-e(S，e)。

搜索 m=eM，使 M=(gi，gi)。如果一切都成功地返回 m，否则就返回上。

定理 4。树加密方案完全正确。

证明文件。密钥生成算法会生成 y：=gf。 所以=yGi 总是持有的 KVfy(pk)。

为了看到我们有正确的解密，首先让我们观察到我们推导了特定形式的解密密钥。让一个公钥 y=gf 有一个匹配的解密密钥

dk 钛……T£_i=(T1 ... T£—1,a, b,　　...，d|，e)

=“1，...Te-1，g，gl(f。*頒)，f“..，f，hj

是给出的。对于 0<入和一点点”，当我们推导出一个新的解密密钥 dk 时 r...。衍生物

$$dk_{\tau_1\ldots\tau_\ell} := \left(\tau_1\ldots\tau_\ell, a\cdot g_1^{\delta}, b\cdot d_\ell^{\tau_\ell}\cdot\left(f_0\prod_{i=1}^{\ell} f_i^{\tau_i}\right)^{\delta}, d_{\ell+1}\cdot f_{\ell+1}^{\delta}, \ldots, d_\lambda\cdot f_\lambda^{\delta}, e\cdot h^{\delta}\right)$$

(dk)r..r i，Tg)我们通过选择 6-Z 来实现它 p 和设置

请注意，派生的解密密钥具有相同类型的格式，只有到节点的路径更长，并且具有随机性 p+6。因为我们在密钥生成后，从表格 dk=的根节点的解密密钥开始(gp，glf0，fp，…，fp，h)，并且所有的推导都保留了该形式，我们通过归纳法得到，

$$\text{dk 钛 } T\backslash=^{r} I \quad \begin{matrix} \text{I } {}^{T}I\cdot {}^{p''}, IL, {}^{g}I = \sigma_1\ldots \prod_{i=1}^{r} i^{(r}0[] \text{声尼上}, {}^{f}\pounds+1, \ldots, \text{丿人}^{h}\text{ I} \\ \backslash \end{matrix} \quad / \text{e(0,1}\}^{p}\text{xG1xG|.}$$

以这种方式推导出的叶的解密密钥将是该形式

在叶子^1 的公钥 y 下的 meM 的加密。Tp e(0，1}使用随机性 r，seZ。给我们一个形式的密文

*c=(C，R，S，Z)=　　gm，g\，g1，f*f：\hjeg；xG/。*

解密算法首先断言 e(R，foHi=iFT)-e(S，h)=e(gi，Z)。使用由加密算法产

e（g1、fo fi)-e(g1，h)=e(gi，(fo　须)h)，

生的密文，这个断言是

这是由于配对的双线性。

接下来，解密算法计算 M=（=、仞）-e（=、b）-e(a，Z)-e(S，e)。插入我们得到的密文和解密密钥中的值

$$M=e(\text{矿评也})-e(g1,\text{謨}(fo\quad fT))\,T\ -e(gp,\ (\text{fonfT})^{r}h^{s})-e(g1,\ h^{p})^{-1}$$

$$=e(g1,g2)^{m}\text{-e(gX}^{r},g2)-e(g1,gX)^{-1}-e(g1),\ \prod_{i=1}^{\lambda}f_{i}^{\tau_{i}})^{\rho})\qquad -^{e(}g1,\ ^{(f}o\ n_{i=1}f?\ )$$

$$-e(gP,\ h^{s})\bullet e(g1,\ h^{p})^{-1}=e(g1,\ g_{2})^{m}$$

由于 meM 和消息空间大小适中，解密算法可以找到并返回 m。　　口

定理 5。在决策假设下，树加密方案是基于 CPA 安全的。

证明文件。假设我们有一个具有优势 e 的叶子对手 A，那么我们就可以构造一个对手 A！以黑盒的方式使用 A，只消耗适度的额外资源，与决策假设相比具有 e/2 的优势。我们现在描述 A！，它得到 co=gf'或 C1 个 G1 与其他组元素一起输入，并试图决定是哪一种情况。

T1，……，TA，GF，G1，GF，FO，……，F，*h，fo!]\f:　')h**

A

“，成(信息 •

设置 y：=gX

对于 j=1，……，A

设置

$$(1\text{-Tj})^{:}=(di,\ g*^{(f}o^{f}j\text{預}\bullet\text{压}=^{1(f}i^{Pj)T}\text{'},\ ^{f}j+1,\ ...,\ ^{f}A,\ \text{砂})$$

(no，a1)七-A(r. ，了I.•了A，f*k？i...Tj—i（I-）] jG[1--A])

b° {0,1}

b '。A(c_6“gf”，“g”g1，（对于：=1f？　)$^{r}h^{s}$)

如果 b '=b 返回 b“”，否则将返回一个统一的随机位

然后通过检查，给定其输入，A 可以计算它所使用的相关组元素。为了分析 A '的成功概率，让我们首先观察到一个随机选择的 x 的 y=gX，就像在 rleaf-IND-CPA 实验中一样。

此外，无论是在这里还是在这个实验中　　)

作为 ti 的统一随机解密密钥分发…Tj_i(1-Tj)。

如果是 b=0，我们有 cb=gf[r]第二次调用 A 的输入是随机加密。由于 A 有优势 e，并且有 50%的机会有 b=0，所以我们得到了优势 e/2。

否则，如果 b=1，cb 只是一个随机的组元素。这给了公关[z]=b]=1/2，所以 A 返回一个均匀随机的位。这不会加或减优势，所以加起来 A 得到 e/2 优势。 口

5.3 多接收机二进树加密

我们现在定义了多接收器 BTE，其中发送器有多个明文地址到不同的接收器。我们可以对单接收机树加密方案进行并行重复，对不同接收机加密多个接收机，但随机性的重用可以使加密方案更有效，与简单的重复方案相比，我们的增益几乎是 5 倍。

多接收机 BTE 具有以下有效的算法：

设置：参数为具有 2 的树指定消息空间 M 和高度入[A]叶子的叶子。我们将树中的一个节点的路径写入力……使用 t<入和叶子 t=入

KGent（pk、dk）：随机密钥生成算法，它可以生成根目录的公用密钥和解密密钥

KVfy(pk)tb：确定性密钥验证算法，如果认为公钥有效，则返回 T，否则返回丄

Derive(dkTi...Tg_i，t)tdkn…^)：随机更新算法，给出节点 tit 的解密密钥^_i 返回节点 T1T£_]T£的解密密钥。 ... . . .

Enc(pki，mi，……、pkn、m[b] ... ，tit 人)tc：随机加密算法，给定公共密钥和发送给其所有者的密钥和消息，叶子返回密文（在失败时返回丄，例如，如果其中一个输入格式错误）

Dec(i，dkTi…T 入，c)tm：确定性解密算法，给定一个密文，一个索引解密和一个解密密钥在错误时返回明文 meM 或丄。

正确性。如果：-诚实生成的密钥验证为有效，则树加密方案完全正确。

公关[（pk、dk）-KGen：KVfy(pk)=T]=1

-正确生成的密文解密到原始明文。I.e.，为所有米，…，mneM，ti，...，t 人(0,1}和 pki，…，pki_i，pki[+]我……KVfy(pkj)=T

Pr (pk，dk)。KGen；pki：=pk；c.Enc(pki，mi，……，pkn，m[rL]... ，[n] Tit 人)dk”-Derive(dk)；...，dkTi……t[r]—Derive(dkTi...t[r_I)；M[E]—Dec(i，dkTi...T 入，c).[z]

(A)=T]-公关(A)=T]|，实验所在

Exp 时

（pk、dk）——KGen

© (0,1}

对于 j=1，……，A

dk[r]…叼-1o-Derive(dk[r]..,0)

[dk]钛……Tj_11 七[^密生值(dk)]i...Tj_1，[1)]

(pk|，m|，...，pki-i，米-i，m，m» i，。.. ，pk，m)

七[^(pk]，丁 1 丁人，…[dk]T1 ... Tj_1(1-Tj)] jG[1..A])

如果（mi，……，mi[0)]，m 里[10]，...，mQ 屯 M[n+1]或者有一个格式错误的键 KVfy(pkQ=丄或一

个键碰撞 p&=pk 或 p&=pk 为 k=0 返回丄 c-Enc(pk1，m1，..。，pk，mi[b)]，。。.，pkn，m”，T1t\)...

b'—A(c)

如果 b=b' 返回 T，否则将返回丄

请注意，虽然如果一个公钥重复，加密方案可能是正确的，但重要的是，相同的公共密钥不重复，因为对手可能使用密文中的相关性来区分。

施工过程。现在我们给出了一个多接收机 BTE 方案的构造。我们的构造几乎与单个接收器相同，除了我们使用相同的随机性 r，s 来加密到多个公钥。为了证明多接收机 BTE 是安全的，我们希望将其简化为单个接收机 BTE 方案的安全性。为了实现安全证明，这意味着我们需要从单个接收机密文模拟到多接收机密文的扩展，为了实现这种简化，知道公钥的离散对数是很有用的。因此，我们在每个公钥上增加一个离散对数知识的证明。

设置：参数指定消息空间 M=[-R..S]CZ。它足够小，可以通过蛮力、组元素 fo，f1，...、fx、heG 和树的高度入来搜索。

这些参数还提供了离散对数的模拟提取 NIZK 证明知识的设置，见第二节。6.3.

*KGent（pk、dk）：选择 x©Z。*并计算 y：=gf。 生成离散对数 x 作为©©|的知识证明。（y；x）并设置 pk：=（y，n）。选择 p©Zp，让 dk：=(gf，gffP，ff，……，f：，h)。返回系统（pk、dk）。

KVfy(pk)tb：解析 pk=(y，n)。如果 yeG1 返回 PVfy(y，n)，否则返回丄

Derive(dkT1..E_1，”)tdk^。E: 给定的

$$dkT1...Tg\_1=(T1...Tg-1,\ a,\ b,\ dg,\ ...,\ d_{A},\ e)e(0,1\}\ \text{‘}^{-1}\ \mathbf{x\ Glx\ G\ “’}+^{2}$$

然后“选择 6 个 Z。并返回

$$dk_{\text{※}}: =(T1...r_{,}\ a\text{-}g:\ ,\ \&\text{-}d]^{e}\text{-}(\text{fo}\ \text{卩厅})_{\text{®}},\ d_{\text{•}}\text{-}f\pounds_{,}\ ...,\ d_{\text{x}}\text{-f},\ \text{e-h}^{\text{6}}).$$

i=1

pki，m 里，..。，pk_{n}，m_{n} ... ，给定的输入 pki=(防，兀“与 ye

$$\text{“通用汽车”}_{\text{f}}\text{…新 g？”、}(^{\text{f0}}\text{S、厅})^{\text{hs}}\text{gi、}$$

c：

GI 和 MIEM 和 TI…T 人(0,1}选择 r、s-Zp，然后返回
12 月（i、dkTi、..Tx、c）Tm：解析

$$dk\text{”..}T\text{入=(ti…“abe)e(01)}^{\text{A}}x\ Gi\ x\ G/$$

和 c=(Ci、……、C，R，S，Z)e G^+2x G。断言 e（R、fonL1f：‘）-e（S、h）=（gi、Z）。假设我是[1]。n]计算

$$M\text{：=e（Ci、g2）-e（R、b）}_{\text{T}}\text{e(a，Z)-e(S，e)，}$$

搜索 m=eM，以便搜索 M=(gi，g2)。如果一切都成功地返回 m，否则就返回丄。

定理 6。假设数据日志的 NIZK 证明系统具有完全的完整性，多接收机树加密方案是完全正确的。

证明文件。密钥生成算法生成 y：=g®和 n-程序生成器(y；x)。证明系统的完美完备性意味着 PVfy(y，n)=T 持有的 KVfy(yk)中的检查，我们也有 yeGi。

为了看到我们有正确的解密，请观察到对于任何 ie[1…n]将加密算法的输出限制在(Ci、R、S、Z)会给我们一个密文，就像早期的单接收机树加密方案一样。此外，该解密算法的工作原理与单接收机树加密方案相同。从单接收树加密方案的完美正确性出发，我们对多接收树加密方案也具有完美的正确性。 口

定理 7。假设单接收树加密是可提取的，并且 NIZK 证明，多接收树加密方案是可提取的。

证据草意图。让 A 成为 rleav-IND-CPA 的对手，对抗多接收机除了 rleaf-IND-CPA 实验中的诚实密钥之外还输出 n-1 公钥的 TBE 密钥。然后我们就可以构造出 Ak 和 A 还有 A！对手分别反对证明系统或单接收机 TBE 方案的零知识或模拟可提取性性质，从而

$$\textbf{Adv(A)<AdVzk(Azk)+AdVse(Ase)+Adv}_{\text{s}}\text{ingle-TBE(A}^{\text{Z}}\text{).}$$

这三种类型的对手以黑箱的方式使用 A，并且只消耗适度的额外资源。

用我们的具体加密方案编写了 rleaf-IND-CPA 实验，我们得到了下面的实验。与此同时，我们指出了实验 Exp^，**支出**隐式地定义了一个零知识的对手 Ak 和一个模拟可提取性的对手 Ae（用于从多个证明中同时提取）。

出口表(A)

x U Zp

y：=-产品（y；x） //在出口£中**(A)、有经验的；**(A)代替运行 n-模拟日志(y)

pk：=(y，n)p U Zp dk：=(gp，gXfPW，...，公里)
U(0,1}
对于 j=1，……，A
dk,叼-1oU 衍生物(dk,..,0)
dk钛……Tj_1 f 派生值(dk) i...Tj_1，1) (pk\，m\，...，pki-\，米-\，米0，m»1，pki,i，m,i，。..，pk,，m,)
1 丁人，…. dk T1 ... Tj_1(1-Tj)jG[1..A])
如果有 j，mj 牛 M 或 KVfy(pkj)=丄或 pkj=pk 返回丄
//支出支出 e(A)j=1，…==提取(yj，新 j)，如果 yj=g？返回丄 r，sUZpc：=(yMf1，……、yn、gi、g%(f 在^=1 抒上)hs)b'uA(c)如果 b=b 返回 T，否则返回丄

实验结果之间的差异 s(A)和 Exp£s(A)是诚实的公钥是否有真实的证据还是模拟证明。如果有差异，则对手将选择最好的 b 并运行实验 Expb(A)/ExpZs(A)能区分这两种情况，并用它来打破证明系统的零知识性质。具体地说，实验与对手 A 一起成为一个联合的零知识对手 Azk，它将实例 y 与证人 x 一起产生实例，然后在结果的证明上，n 试图区分它是否通过实验和 A 被模拟。更改为我们提取对手密钥离散对数的情况，对成功概率的影响可以忽略不计，否则我们可以使用联合实验和对手来打破证明系统的模拟可提取性。但在这里，我们必须小心，因为这一事实依赖于对许多重新绕组的仔细分析，我们将在完整的版本中扩展这个证明。

最后，我们为利用 Exp 中的任何优势的单接收机 TBE 构建了一个对手；s(A)破坏单接收机 TBE 的实验。

A ‘ ...，T*{dk”..T3 i(t-TJ)。N)(m O m ‘1)

n.模拟日志(y)
pk：=(y，n)
(pk，m，…...，英里，m 里)，。。.，pkn，mQ—A(pk，T1...t，{dk,(i-T3)返回 mO：=mi，
还有米：=mi)4(C，R，S，Z)—b
如果对于任何 j，我们有 KVfy(pkj)=丄或 mj=丄返回丄
用于 je[1..n]\{i}
解析 pkj=(y，新泽西州)
如果 j=，我从证明 Xj 中提取出那个 y=gx 如果失败，则返回丄
如果提取集 Cj：=RXjg-3
词号：=C
返回 A（C1、....、C，R，S，Z)

在本阶段，该实验与单接收机实验完全相同。所以我们有广告s(A)=广告公司(A ‘)。为了再次看到这一点，在分析中需要一些注意，以确保在证人提取中使用的重绕组在实验中对手操作的两个阶段都有意义。口

5.4 多接收器树加密与消息空间 Zp

对于 meZp，计算 M=e 的离散对数(gi，g2)通常是不可行的 m 而解密则需要太长时间。自然的解决方案是使用分块加密。拿 m 来写，写成 m=Yj=imjBj-1 与块 mje[0..其中块大小上的

绑定 B 足够小，可以通过[0..强制搜索。[7]

让我们正式写下在消息空间 M=的多接收器 BTE 方案上构建的消息空间 Zp 的多接收器 BTE 方案的细节。

设置：参数包括基本多接收机树加密方案的设置和正整数 B、m，以便[0…B-1]CM 和 p<B.

KGen‘t（pk、dk）：返回（pk、dk）。KGen

KVfy‘(pk)tb：返回 KVfy(pk)

衍生/(dkTi...;;TE_i，)Tdk‘：返回衍生物(dkTi...;;TE_i，”)

Enc‘(pki，mi，...，pkn，mn，ti，...，t 人)Tc/：给定 mi，...，mneZp 将它们分割成碎片 mi，je[0..B-1]，以便 mi=52^=1m%jB^{j-i}。对于 j=1，…，m 设置 Cj-Enc(pki，mi，j，...，pkn，mn，j，ti，…，t 人)。返回 c/：=(ci，...，c_m)

12 月”（i，dk）解析 C=(ci，……，c)和 j=1，…，m 计算 mi，j.Dec(i，dkTi...T 入，cj)。如果有任何 mj=丄返回丄，否则返回 m/：=Em=imi，jB国防部长 p

[7]使用步步算法使搜索更快。

定理 8。分块多接收机树加密方案具有完美的正确性。

证明文件。根据消息空间 M 的多接收树加密方案的完全正确性，块 m*e[0…运行时检索的 CM..，(c1，c)）返回在 Cj 中加密的数据块。因为加密算法选择了数据块，所以 mi= 我们看到了那个 m= 口

定理 9。Z 的分块多接收机树加密方案对“注册会计师”是安全的。

证明文件。从一个混合论证可以看出，对手 A 反对分块多接收机树加密方案可以用黑盒的方式来构建对手 A! 与多接收机树加密方案的消息空间 M=[-R…，给我们 AdvM

口

让我们用组元素的重新排列写出完整的加密算法，以观察加密过程和由此得到的密文可以分为两部分。第一部分依赖于公钥和明文，第二部分依赖于叶子，这两部分都依赖于随机性。我们还写出了解密算法，以观察到第一部分和叶子的第二部分是唯一的。

环境（pk1，　。，pk，m)tc=(c1、C2)：选择 ■)•••，m<~ 和计算 ci：=Enci(pki，mi，pkn，mn；ri，si，…，r，s)和 C2：=Enc2(T1，...，；ri，si，……，r，s），其中

-Enci(pki，mi，...，pkn，mn；ri，si，……，r，s)首先解析 pki，…，作为 pki=(yi，ni)与 yeGi 和块 mi，…，mneZ如 mi=ERimijB使用数据块 m%je[0..B-1]。

它返回 ci：=(Ci，i，...，Cn，，*Ri，Si，...，Rm，S*)e G“，其中

词、j =y? j = 9? *j =9i.*

—Enc2(Ti，...，T*ri，si，...，r，Sm)在蒂，……，t 人(0,1}返回 c：(字，……，Zm)eGm，其中

$$Z_j := \left(f_0 \prod_{i=1}^{\lambda} f_i^{T_i} \right)^{r_j} h^{s_j}.$$

如果一切正常，请返回 c：=(ci，C2)。如果输入输入 ci=⊥或 c2=⊥，加密算法返回 c：=⊥。

Dec(i，dkTi...T 入，c)tm：解析 c= （ci、C2） 以及 Ci = ‘...■...■..m)。看看所有的 j=1，……，我们有

$$e(9i, Zj) = \prod_{i=1} 尸)-财, h)。$$

假设 1<i<n 解析 dk=(ti……a、b、e)e(0、1}如在多接收机树加密方案中的 M。对于 j=1，……，m 计算

$$M_j := e(C_i, j, g2) , e(R_j, b)-i , e(a, Z_j), e(S_j, e)-i$$

并搜索 mj=eM(gi，g2)。如果一切都成功了，请返回 m：=E^=1mjB■odp，否则返回⊥。

下面的定理表明，一个格式良好的密文的第二部分是唯一的。

定理 10。对于任何密文 c=(ci，C2)和叶 Ti…t 人不解密到⊥，不能有其他的 c2=C2，这样 C=(ci，c2)也非琐碎地解密关于“……”，

证明文件。为了使解密算法返回规则明文而不是错误丄，密文对 c=(ci，c2)必须如 ci 算法中所述解析为 ci=(Ci，i，...，$C_{n,m}$、*Ri*、*Si*、……、R_m，S_m)和 *c2=(字，……，Z_m).*解密算法检查所有 j=1，…，我们有 e(gi，Zj)=(Rj，fo 卩打 fT)，e(Sj，h)。由于 gi 是 Gi 的生成器，所以没有不同的 Zj=Jj 也满足这个等式。因此，解密算法将在任何 c2=c 上返回丄？.　　口

5.5CCA 安全多接收机公用密钥加密

保密性

具有前向保密的多接收机加密方案包括以下有效算法：

设置：参数指定消息空间 M 和最大时代数 $T=2^{Xr}$

KGent(pk，dko)：随机密钥生成算法，为时代 t=0 生成公钥和解密密钥

KVfy(pk)tb：确定性密钥验证算法，如果认为公钥有效，则返回 T，否则返回丄

KUpd(d&T)tdkT+i：随机更新算法，给出时代 t 的解密密钥，返回 t+1 的解密密钥。如果 t+1=T，它返回丄，我们将在整个论文中假设 dk，表示它打算的时代 t。

Enc(pki，mi，…c：随机加密算法，给出 n 个公共密钥和消息以及一个时代返回一个密文（在失败时，例如，如果其中一个输入格式错误，返回丄）。

Dec(i，dkw，c，t)tm：确定性解密算法，给出索引 i 的解密密钥 dkt 和时代 t 的密文，在出现错误时返回明文 meM 或丄。它总是返回丄，以防 t 牛[t ‘..t-1]。

正确性。在以下情况下，具有前向保密性的加密方案完全正确：

-诚实生成的公钥验证为有效。

公关[（pk、dko）—KGen：KVfy(pk)=T]=1

-正确生成的密文解密到它们的原始明文。I.e.，对于所有的米，，米 $_{n}$E<，0<‘<t<T 和 pki，…，pki_i，pki$_{+}$i，... ，pk_{n} 其中，KVfy(pkj)=T

Pr (pki，dko)JKGen；c.Enc)(pki，mi、..。，pk_{n}，m_{n}，t) dkiJKUpd(dko)；……、dk/JKUpd(dk，_i)；12 月 12 日(i，dk/，c，t)

在 CCA 攻击下的不可分辨性。我们将有状态选择的密文对手 A 的显著优势定义为 Adv(A)=|Pr[Expo(A)=T]-Pr[Exp1(A)=T]|，其中

Exp 时

（pk、dko）JKGen

存储 dk。

(pki，英里…pki_i，mi_i，m 里，m 里，pki，m 里，...，pkn，m”，t)J A12 月（月）

pki：=pk

$_{b}$IAKUpd，腐败，12 月(c)

如果（mi，……，mi$^{0)}$，m 里^{1}），...，m”)E M^{n+1} 或者有无效的公钥 KVfy(pkj)=丄，或者对于 k=0，与 pkk=pk 的公钥发生冲突。T-1]返回丄(b)

行驶里程：=英里

cJEnc(pki，mi.…，pkn，$m_{r比}$，

如果<损坏或(c，t)EQ 返回丄 Ifb=b′ 返回 T，否则将返回丄

库垫	已损坏	12 月（i、c、T）
获取最后一个 dkT	获取最后一个 dk，	获取最后一个 dk，/
dk‘+iJKUpd(dkT)	^{T}已损坏：—^{T}	Q：=Q u{(c，t)}
存储 dk，+i	返回 dk_{T}	返回 12 月（i、dk、/、
如果是 t+1=T，则忽略进一步的调	忽略其他呼叫	

用，其中解密查询可能有任意的(i、c、t)。特别地，解密函数可以在可能不匹配 n 个接收器的挑战密文的维度上调用。

施工过程。我们现在提出了一个多接收机 fs-cca 安全的加密方案。它建立在 Z 的多接收机树加密方案之上，我们使用在观察中引入的符号，多接收机 BTE 加密算法可以分为两部分，一个部分独立于叶子，另一个部分独立于明文。在 fs-CCA 安全加密方案的安全证明中，我们依赖于随机甲文模型。

设置：参数指定消息空间 Z_{p} 和最大的时代数 T=2$_{在}$和一个哈希函数 H：(0,1}*t(0,1}$_{从}$.

该设置包括在上一节中定义的多接收器树加密方案的组元素 fo，...、fx、heG，其中为入=入 t+入 h。KGent（pk、dko）：选择 x.Zp 并计算 y：=g®- 生成 n 。

产品（y、x） 然后让 pk：=(y，n)。选择 p Zp 并设置 dk :=(g，g®f$_{n}$，*ff*、、、*f*|、*h*让 dko：=(0，(dk})和返回(pk，dk。)。

KVfy(pk)tb：解析 pk=(y，n)，如果 yeG1 返回 PVfy$_{d}$|og(y，n)，否则返回丄

KUpd(dk)，t ）*t dk*$_{T}$/：如果是 t‘牛，则返回丄[t+1……T-1]。解析 t=...，r\$_{T}$在二进制文件和 dk 中，=(t，(。灼…无}t$_{r}$)，其中 TT 是一组最小的节点，这样它们的子树正好覆盖了[t]中的叶子。T-1](TT 最多有入 t 节点)。设 TT/是子树覆盖叶子的最小节点集 $_{7}$和对于所有新的 t、...reTT/\TT 推导出一个子键 dk-。T.返回

$^{dk}T^{Z}$=（^{T}，$^{(dk}$钛……£}我…T$_{£}$等等$_{7}$／）

Enc(pki，mi，……，pk$_{n}$，m，t)tc：解析 t=　　t、……t 人了在二进制文件中。选择 r、，si，……，r$_{n}$，s_{n}个 Zp 和计算值

中国人：=Enci(pki，mi，...，pk$_{n}$，m$_{n}$；1，．．．，m)。

计算 t 人了+i……t 人：=H(pki，...，pk$_{n}$，ci，t)和

C2：=Enc2(钛、……、t；ri，si，..。，　m)。

如果一切都成功，则返回 c：=(ci，C2)，否则将返回丄。

Dec(i，dk，/，c，t)tm：解析 c=(ci，c2)和 t=Ti…t 人了。计算 t 人了+i…塔：=H(pki，…，pk$_{n}$，ci，t）。假设 t[t‘…T 入并返回 i，dk，i…T，c)。如果有什么东西失败了，请返回丄。

定理 11。多接收机加密方案具有完美的正确性。

证明文件。Zp 的底层多接收机 BTE 方案具有完美的正确性，这也意味着在丁次丁+i…的完美正确性…ta 碰巧等于 H(pki，……，pk$_{n}$，ci，t）。因此，我们所谓的 fs-CCA 安全加密方案具有完美的正确性。　口

定理 12。多接收机加密方案是 fs-CCA 安全的。

证明文件。假设 A 是一个具有优势的 fs-CCA 对手 e。我们假设各方都将 H 当作一个随机的神谕。在随机甲骨文模型中，哈希函数 H 返回一个随机字符串 tat+i…当查询新输入时。随机甲骨文可能是可编程的（良性可编程定义：编程查询首先返回一个随机值，然后有人可以程序的输入值），我们不排除一个能够编程哈希函数只要它不能找到碰撞。

在用我们的具体加密算法编写 fs-CCA 实验时，对手的优势是 e=|Pr₀(A)=T]-Pr[Exp1(A)=T]|，其中

出口表(A)

（*pk*、*dko*）*JKGen*
存储 dko
(pki，　…，我，我，我，m 里，pki，m 里，...，pk，m
pki：=pk
如果（mi，……，mi⁰⁾，m 里¹)，...，m_n)屯 M^{n+1} 或者有无效的公钥 KVfy(pkj)=⊥，或者对于 k=0，与 pkk=pk 的公钥发生冲突。T-1]返回⊥(b)
行驶里程：=英里
*1，．．．，*mJZpci：=Enci(pki，mi，。..、pkn、mn、ri、si、……、

解析 t=^i..。 ᵀ. . *i，...，*)
C2：=Enc2（Ti，。..，T_A；ri，si，.….，
ʳm，ˢm)
C :=（ci、C2）
*5*___ ^KUpd，腐败，12 月　(c
如果<损坏或(c，t)G() 返回⊥如果
b=b' 返回 T，否则将返回⊥

库垫 获取最后一个 dk_T dk+iJKUpd(dk) 存储 dk_{T+}i 如果值为 t+1=T，则忽略进一步的调用	已损坏 获取最后一个 dk_T/ ᵀ已损坏：--ᵀ 返回 dk_T 忽略其他呼叫	12 月（i、c、t） 获取最后一个 dk_T/ Q：=Q u{(c，t)} 12 月返回（i，dk)/，c，t)

．．．观察到，在随机甲骨文模型中，对手无法区分我们首先选择 t 入了+i 的区别。。t 人均匀随机，然后在选择 ci 后假装（本质上是随机编程），当 A 查询(ch，t)，har 希函数返回 t 入了+i.t 人 So，我们看到如果我们运行以下修改，优势是不变的

实验结果

出口”

查询随机 oracle 的输出 r\$_{T+1}$, $(0,1\}^{A}$

（*pk*、*dko*）*jKGen*

存储 dkg

(pki、ml、。。。., pki_, mi_, m 里, m 里, pki, m 里, ..., pk, m

如果(皿, ……, 英里$^{0)}$, m 里^{1}), ..., 风)屯 M^{n+1} 或者有无效的公钥 KVfy(pkj)=丄，或者对于 k=0，与 pkk=pk 的公钥发生冲突。T-1]返回丄(b)

行驶里程：=英里

質 1, ^{s}1, •…, 質 m, ^{s}m ^{Z}p

ci：=Enci(pki，mi，。..、pkn、mn；ri、si ^{r}m, ^{s}m)

解析 t=^i..。 T 人了

将随机神谕编程为 H(pki，……，pkn，ci，t)：=t 人了+it 人 C2：=Enc2(Ti，。.. . . . ，T_{A}；ri，si，.…., ^{r}m, ^{s}m)

C := (Ci、C2）

K-^KUpd，腐败，Decj)

如果已损坏的<t 或(c，t)GQ 返回丄

如果 b=b 返回 T，否则将返回丄

库垫	已损坏	12 月（i、c、t）
获取最后一个数据 $_{T}$	获取最后一个 dk_{T}/	获取最后一个 dk_{T}/
dk_{T}+i-KUpd(dk)	^{T}已损坏：— ^{T}	Q：=Q U((C，T)}
存储 dk_{T}+i	返回 dk_{T}	12 月返回（i，dk）$_{T}$/,
如果值为 t+1=T，则忽略进一步的调用	忽略其他呼叫	

唯一的小区别可能是，如果 A 尝试编程随机神谕，并命中相同的输出，或者之前意外地命中了相同的输入，这不太可能是由于密文中的熵。

接下来,观察到我们可以用概率为 1/T 的方式提前猜出 t。所以选择 Ti.t 人了 J(0,1}.. 如果 A 输出另一个历元，则返回一个均匀随机的位 b ‘。这意味着我们有电子科技的优势。

我们现在可以在修改后的实验中使用黑盒的 A 来构建一个 rleef-IND-CPA 对手 A！ 它有同样的优势，只使用适度的更多的资源。我们期待着 A！ 至少拥有与 A 相同的随机神骨文，即，如果 A 可以编程随机神骨文，A 可以！。构造的主要思想类似于 IBE 的[BCHK07]CCA 转换，即只要叶子匹配挑战密文 c 和挑战时代 t 的解密密钥仍然是秘密，我们原则上可以揭示所有其他叶子的解密密钥，仍然不可区分，这些其他叶子足以回答解密挑战。要了解这一点，请注意，如果解密神谕是在一个挑战 d=(ci，c2)，不匹配挑战密文 c=(ci，c？)在第一部分和挑战时期 t 中，然后因为他们唯一地定义了第二部分 c2，我们有 c=c′ 因为 c2 是由 ci 和 t 唯一决定的，因此成功条件(c、t、牛 Q 不再成立。

}

A$^{(pk}$, ^{T}1, • • • , ^{T}A, $_{Z}$*i i(i-$_{Z}$j)* jG[1--A])

（*pki，mi*、.…、*m：0*）、*m：*^{1})，…，T. ‘)’-^Kupd，腐败，12 月(pk)

如果 t ‘=不返回丄

在查询 Kupd，A！ 跟踪增量 ttt+1

如果查询损坏，树解密密钥足以构造解密密钥ˆ鄉顷明

在查询 Dec（i、c、t）中，叶的树解密密钥足以解密。回路（m0，m；）：=（m：，m：，

″）

返回 A(c)

对于概率 1/T，我们正确地猜测了历元，并保持了优势，所以总体来说，我们至少在打破我们的基础多接收机 TBE 方案方面有 e/T 优势。 口

6 个非交互式的零知识证明

6.1 随机神谕模型

我们使用 Fiat-Shamir 启发式来创建 NIZK 证明，这意味着证明者将使用哈希函数在证明中创建挑战。我们的证明系统将在随机 oracle 模型中是安全的，其中当证明健全度和零知识时，哈希函数被建模为一个随机函数。I.。，该函数不是一个确定性哈希函数，我们允许该函数返回一个上域中的一致随机值。

设 H：*tY 是一个具有上域 Y 的哈希函数。展望未来，在我们的 NIZK 证明中，我们将使用 Y=Z 形式的上域，*Y=(0,1}*对于可变尺寸的 A，正如我们已经在签名 Y=中使用的那样，当我们需要区分具有不同上域的哈希函数时，我们会编写 Hy。我们将限制可编程性来保证 y 是均匀随机的，这限制了模拟器的能力，并使我们的结果比零知识更强。因此，为了捕获这个有利的可编程的随机神谕模型，我们要求使用哈希函数的算法通过以下神谕来实现。

0	*Q*	程序
关于输入端的信息 x： 如果 O(x)=y=丄返回 y y—H(x) O(x)：=y Q：=Q u(x} 返回 y	激活时返回年年 并等待一个输入 在等待的输入 x： 如果 xeQUQprog 返回丄 q执行程序：=q程序 U(x) O(x)：=y；返回 T	在输入 x 如果是 x∈q 程序 返回丄 其他呼叫 y—O(x) 并返回 y

我们使用的约定是，当甲骨文的值尚未定义时，O(x)=丄。从 Q=Qprog=0 开始很方便，但请注意，我们的方案使用的参数生成可能与哈希函数的使用相纠缠，所以这可能是一个错误的假设。我们所希望的是，没有进行过太多的查询，即，q=|Q|+|Qprog|数量适中，并且输入有足够的最小熵，不太可能到达之前的查询之一。由于我们的具体 NIZK 证明，为了简单起见，我们假设从 Q=Qprog=0 和 q=0 开始。

我们将构建方案，其中甲骨文调用必须按顺序执行。如果一个 Oracle 调用的输出是勿，我们查询 H（勿，其他输入）来获得勿+“因为 y 是不可预测的，这保证了创建勿+的调用是在创建切的调用之后出现的。我们在以下引理中正式化了这个保证（省略了生日悖论的简单风格的证明）。

引理 1。设为 $A^{O,\text{操作程序}}$ *成为一个对手，创建多达 q 的查询-响应对 O(xi)=yi，…，O(x_q)=y_q，无论是通过进行直接查询，还是通过对甲骨文进行编程。A 对某些 i<j 使用 Xi=(yj，*)的概率为 2*
小于。。

由于引理 1 的结果，我们可以在概率分布上几乎没有什么差别，假设我们正在与“链限制”对手合作，它们不会按顺序创建链哈列值。

6.2NIZK 证明

我们通过三种有效的算法定义了一个有效的可判定的二元关系 R（它可能依赖于全系统的参数）的一个非交互式的零知识证明。算法使用哈希函数，我们在安全证明中将其视为随机神谕。

证明。（实例，证人）tn：随机算法，在一个实例和证人上返回一个证明 n（或错误符号丄）
项目页 。(实例，n)tb：确定性算法，在一个实例和证明 n 上，如果证明被认为是有效时返回 T，否则返回丄 If，验证算法得到丄作为输入，无论是在实例，证明，还是从甲骨文，它都返回丄。
模拟工作。（实例）tn：在实例上返回模拟证明 n 的随机算法。

我们说，NIZK 证明系统是完全完整的，如果对所有的（例如，证人）eR

证明证明。（例如，证人）：PVfy。(例如，n)=T]=1。

更一般地说，我们将完整性对手 A 的优势定义为

公关（实例，证人）-A。 n-证明文件。（例如，证人）：
（例如，证人）eR 和 PVfy°(例如，n)=丄

在我们稍后的分块证明中，我们限制证人组有关系
R完成的，R—

对于一个零知识的对手 A，我们将其优势定义为

|2Pr[b—{0，1}；b‘—A■：=b‘]-1/，

其中（实例，证人)神谕作为证明(实例，证人）=证明。（例如，证人）　　以及　　证人，证人）　　=
模拟工作■(实例)，而在(实例，见证）牛 R 上，它返回丄，无论是 b=0 还是 b=1。

对于一个模拟稳健的对手 A，我们定义了它的优势超过 par 为

Pr[（实例，n）—A■ 实例屯 Lr 和 PVfy■(例如，n)=T]。

请注意，让对手访问 Oprog 意味着它可以运行模拟 因此，在任意实例上模拟证明。然而，我们只考虑一个假实例实例牛 Lr 的证明，如果验证独立于随机甲骨文被编程的点，即它没有被模拟。

对于模拟可提取性的对手 A，我们定义了它与黑盒提取器提取物相比的优势为

Pr （例如，n）-A^0.0磅；证人的引渡人。。。（例 ■(例如，证人）：
如，证人）R 和 PVfy°　　-Qprog（例如，n）=T

提取器看到所有甲骨文查询 A 的文字记录，并被允许倒带 A，并在随机甲骨文中以新的随机性再次看到所有的查询和响应。

6.3 离散对数的证明

我们使用一个标准的施诺尔证明来获得离散对数的知识。我们想使用它是模拟可提取的。

关于离散对数知识的 NIZK 证明

设置:带有发电机 gi 的已知素数阶 p 的组 gi。在实现中的哈希函数 Hz 将实例化一个 oracleO。
但在安全证明中被建模为一个随机的甲骨文。

实例：yeGi
声明：了解 y 的唯一离散对数
见证人：xeZ$_p$这样的 y=g®
证明⁰（例如，证人）：
—选择 rUZp 并计算 a：=
—计算 e：=O（y、a）
—计算+：=

-让证明为 n=(a，z)eGixZ$_p$
—返回 n（并删除在证明期间创建的中间信息）PVfy⁰（例如，n）：
—检查实例和证明的格式是否正确地使用组元素 y、eGi 和字段元素 zeZp
—计算 e：=O（y、a）
-如果所有的检查都通过了，则返回 T

y^{ka}_{i}

否则，您将通过返回⊥来拒绝

安全性

定理 13。离散对数对任意甲骨文知识的证明系统都是完整的，在随机甲骨文模型中模拟可提取和零知识。

*证明文件。完美完整性：*ya=(g®)ₖ(gi)=gj=gi。

零知识：模拟器问可编程随机的神谕一个挑战 eUZp。然后模拟器选择 z 个 Zp 并设置 a：=giy。模拟器通过编程随机甲骨文返回 e 的输入(y、a)来完成模拟证明。在实证明和模拟中，z 都是均匀随机的，正如 e 一样。给定这两个值，值 a 由验证方程确定，因此实证明和模拟证明具有相同的概率分布。

模拟数据的可提取性。沿着格罗斯的路线前进。 口

6.4 提供正确的秘密共享的证明

展望未来，我们将构建非交互式分布式密钥生成方案，其中经销商使用我们的前沿安全多接收机加密方案对秘密共享进行加密。多接收机密文组元素 Rj，Ci，j 具有唯一定义的离散对数 Rj=g“和 Ci，j=yrg。从这样的密文中，任何人都可以计算组合的组元素

$$r=\prod_{i=1}^{m} \text{‘「’}, \quad Ci=\prod_{i=1}^{m} cj, \ldots, \quad c=\prod_{j=1}^{m} cr.$$

这些组元素唯一地定义了 reZp 和 si，……，s=---=-g 于的关系 r=%i 以及 $_i$ ，

在一个正确的处理中，每个 Si=a(i)=a”。我们现在给出了一个关于这个等式的 NIZK 证明……我们的想法是使用哈希函数来计算一个挑战 xeZp，我们用它将实例压缩到

$$c：=\prod_{2=1}^{nn} c\text{？}=\prod_{i=1} 妒-g\ 尹$$

如果这个说法是真的，我们就有了

$$\text{£六个}^{2} \quad = \prod_{i=1}^{n} \text{£} \quad \prod_{k=0}^{/t-1} (\text{£}a_k) \quad \text{、} \quad \text{x‘} \quad = \prod_{k=0}^{t-1} \text{£}a_k \quad \prod_{i=1}^{/n} I\text{£} \quad i^k x.)$$

如果该语句是假的，即，有任何 s.=a(i)，那么最多有 n 个可能的 x 值，其中等式意外地保持不变。因此，在压缩步骤之后，我们可以用第二个随机的 Oracle 挑战 x 来进行一个标准的施诺尔风格的证明'压缩的 s=£^=1s^x² 与 卩£的离散对数相同(压-=0 ″

为正确的秘密共享提供的 NIZK 证明

设置：已知序数 p 的 G1、G2、Gt 组，具有配对 e：G1xG2TGt 和发电机 g1、g2、e（g1、g2）。

组元素 heG2\{1}。在实现中的哈希函数 Hz 将实例化一个 oracleO但在安全证明中被建模为一个随机的甲骨文。

实例：y1，……，yneG1，Ao=g 芽，-…，在-1=g”和[8]

$$^{r}_{g}1\text{-}g1, \qquad\qquad ^{r}_{g}n\text{-}g1$$

语句：实例中的离散对数满足 i=1，……，n

$$\text{硅 } i^{\,t-1}_{k=0} \qquad \text{万崎 国防部长 p}$$

证人：r，S1，……，SneZ$_p$满足 s.=a(i)，其中 a(i)=E；=0aki^{k}国防部 p。[9]

证明[10]（例如，证人）：

—计算 x：=0（实例）

—生成随机 a、p 个 Zp 和计算

$$^{F}=g^{p}, \qquad\qquad \text{，A=g2，} ^{Y}=\text{(n 必 j■g?}$$

[8] 实例只指定 2n 个+t 组元素，指数不是实例的一部分，而是由组元素唯一定义并供以后参考。

[10] 维克多指出，规律的健全可能就足够了

—计算 x'：=O(x，F，A，Y)

$$Z_{r} \quad rx^{l}\text{+p 型号} \qquad Za \quad x\hat{}_{\circ}^{n}{}_{pi=1}{}^{s}iX^{\varepsilon}\text{ +a 国防部}$$

—正在进行计算

-让证明为 n=(F，A，Y，Zr，z_{a})e Gi x G2x Gi x Zjp

—返回 n（并删除在证明期间创建的中间信息）PVfy^{0}（例如，n）：

-检查实例和证明是否正确使用组元素 yi、…、yn、R、Ci、…….、Cn、F、YeGi、Aq、…、A(_1、AeG2 和字段元素 Zr、ZaeZ$_{p}$ 如预期

—计算 x：=O（实例）和 x：=O(x，F，A，Y)

$$B^{\kappa_{r}} \text{-F=gi} \qquad (f[A*j \qquad i^{k}x^{j} \mid^{x} \text{-A=gZ}$$

-进行验证

以及

$$\left/n \quad \backslash i=1 \right. \qquad \left(n^{c}n^{-y}=-gz^{a} \right)^{x}{}_{/} \qquad n \quad i=1$$

—如果所有检查都通过，则返回 T，否则返回上将被拒绝

正确秘密共享证明系统的成本

通信：不考虑挑战，证明大小是 Gi 中的 2 个组元素，G2 中的 1 个组元素，Zp 中的 2 场元素

证明计算：证明计算在 Gi 中以 n 范围的多指数为主。

验证器计算：验证器计算主要是 G2 中的 t 宽多指数，Gi 中的两个 n 宽多指数，以及计算 Zp 中的 tn 宽内积(对于大 t 和 n 可以使用 FFts 来减少)。

安全性

定理 14。对于任何神谕和模拟声音，正确秘密共享的证明系统都是完整的 在随机甲骨文模型中，知识为零。

证明文件。完美完备性：在一个格式良好的实例上，我们看到证明器确实用 Gi、G2 和 Z 中的元素计算了一个格式良好的证明 ，如预期。对于任何甲骨文的输出，xeZp 和 x'eZ 写出这三个验证方程，我们看到它们确实满足于一个诚实构造的证明，即，

(g2 户或 j . 加=g；牛 5+[a]

在第二个等式中使用证人有=E[illegible] 、以及

/n \ [x] /n\ [p] /n\ [rx*p]

m(y)-m）-ga=（mj -gX*[illegible]

*可编程随机神谕模型中的统计零知识：模拟器首先调用 0（实例）来获得一个挑战 xeZp。然后它调用 0prog 来获得第二个挑战 x'*e Zp.该模拟器可以均匀地随机选取 U Zp
并计算满足三个验证方程的唯一群元 F、YeGi、AeG2。它通过调用 Oprog(x、F、A、Y)来完成神谕的编程，它将神谕编程为具有 O(x、F、A、Y)=x'，除非以前使用过(x、F、A、Y)，在这种情况下，编程失败并返回丄。

现在让我们论证，模拟证明与真实证明是不可区分的。首先观察到，在这两

p=z[r] —*x'*月模型 *a=z*[a] —*x* ‘E[n, i=1] *a(i)x*[z] 国防部 p。

种情况下，我们得到均匀随机的 xeZp 和 x'e Zp.现在，请进行定义
自 Zr 和 z 之后 在仿真中选择了均匀随机的方法，实证明和模拟证明都具有均匀随机的 p 和 a。现在给定这些值，F、A 和 Y 由三个验证方程唯一定义。所以真实的证明和模拟证明有完全相同的分布，唯一的问题是如果(x、F、A、Y)之前被查询，从那时起模拟器在尝试编程时失败。然而，Zr，z 的随机选择 个 Zp 意味着 F、A 是一致和独立随机的，因此遇到以前的查询的风险最多是
随机神谕模型中的统计模拟稳健性：考虑一个对手试图生成一个实例 yi，…，Cn，其中有 sˆ=(i)和有效证明 n。如果实例或证明格式错误，证明将被拒绝，因此我们可以假设对手返回一个格式良好的实例和证明 n=(F，A，Y，zr，ze 的 xG2xGixZ*如果对手编程了甲骨文或(x、F、Y、A)定义不认为它是成功的伪造，它可能只是一个模拟证明，所以让我们假设 x，x[i]起源于查询 O（实例）和 O（x、F、A、Y）。

现在，对手可以尝试三种策略。第一个策略是要很幸运，然后进行一个查询 x' 在第一个查询 x.O（实例）之前的。O（x、F、A(Y)。然而，这意味着对手必须提前猜测 x，而通过引理 1 2 ，对手找到反向顺序查询对的机会小于舞。

第二种策略是希望一个 x，这样=i=a(i)x[i]即使有一个=a(i)。从年代起!，...，s_n 多项式 a(i)由实例定义，在每个查询 O（实例）中，施瓦茨-zippel 引理将成功的机会最多限制为 n/p。 因此，对手成功获得这样一对实例和挑战 x 的机会被 qo 限制

P ‘ n- n -

最后，它可能是 En=i6=En=ia(i)x但是对手在选择 F，A，Y 后得到一个随机挑战 x ‘=O(x，F，A，Y)，它可以用粉，Za 回答，所以证明是有效的。取前两个验证方程的离散对数，我们可以看到=x ‘r+p 和=x’ Enia(i)x[i] +a.让月所以 Y=Y[.

i(yxi)最后一个验证方程告诉我们

$$\prod_{i=1}^{n} \quad \prod_{i=1}^{n} \quad n(\text{年 } gi \text{ ‘产} \prod_{i=1}^{n}$$

有 y 的部分取消了，留下了我们

$$\prod_{i=1}^{n} \text{ng } \text{严“-gi=gZ}_{a}-$$

取离散对数基，我们得到了

$$\text{x ‘》《六*+月=x’》a(i)x}_{i}\text{+a，} \quad \sum_{i=1}^{nn} \quad \sum_{*=1}$$

它只有 1/p 的机会比选择 x ‘是真的。通过多达 qo 的尝试，对手的成功概率小于岛。

总的来说，一个组成 q 甲骨文查询的对手完全有可能得到 $q \cdot \quad q^{2}$ 击败由舞+限定的模拟可靠性 ₄号<等，当几乎没有失去一般性的 qo>n。 口

6.5 正确的分组的证明

经销商必须提供证据，证明可以在交易中解密密文，以便接收者可以恢复他们共享的签名密钥。我们使用多接收机转发安全加密方案，其中密文大部分是可公开验证的。唯一的问题是，明文应该被分割成小块，并提取接收器计算离散对数所需的块。因此，如果接收器应该提取的块太大，并且我们需要一个 NIZK 证明系统，它可以确保所有块的大小都适中，那么接收器就会有问题。

我们首先观察到，我们不需要考虑完整的密文，其中只有部分内容对于证明加密的块有正确的大小至关重要。每个接收器我都看到一组埃尔伽玛尔密文，据说里面有 m 块她的明文。我们的
想证明密文是有效的元素加密到中等大小的明文块，。所以接收器可以提取它们并计算她的完整纯文 Si=£m1si，jmodp。每个埃尔伽玛尔密文（Rj、Ci、j）可以唯一写为 ，y“)。在我们的加密技术中在方案中，我们在解密过程中使用密文上的配对来计算 e(gi，g2)，如果经销商是诚实的，那么 s%je[0..B-1]和接收器可以强行搜索 s%j。

我们想避免一个不诚实的经销商使用 sj，这是不能被暴力提取的。这样做的一个策略是使用范围验证来证明 s%je[0..B-1]。然而，精确的范围证明对于素阶组来说是昂贵的。相反，我们的目标是建立一个放松的范围样的证明，这将显示有一个小的厶 ij，这样厶 ijs%j 属于一个中等大小的范围。这足以表明 sj 可以被提取，因为接收器现在可以做一个蛮力搜索一个合适的厶 ij，带我们到这个范围内。由于乘法因子厶 ij 和范围的增加，蛮力搜索不像在一个诚实的处理中那样有效，但如果在范围之间有适度的差异，它仍然是可行的。

正确块证明背后的想法是并行做许多小块证明。每一个子证明都有适度的可靠性，但它们共同地使得证明者作弊的机会可以忽略不计。每个子证明都将使用一个挑战 [0..E—1]。采用线性组合

$$\prod_{i=1}^{n}\prod_{j=1}^{m} C_{i,j} \quad \prod_{i=1}^{n}$$

我们可以推导出匹配的编码的随机性 卩爲 Rj，七为每个山季，所以我们有一个独特的决定 。现在，我们可以了
（目前不用担心零知识）要求证明者揭示 Z=En=1、m=1eijsij。如果证明者是诚实的，s%je[0..因此 z_s 在[0..S]范围内，其中 S>nm（E-1）（B-1）。现在，一个不诚实的证明者提供 z 呢 $_s$ e[0..S]？好吧，如果证明者有机会这样做，那么每一个(i、j)都存在两组挑战(e1、1、eij、en、m)和($e_{1;1}$，只有在(i，j)条目中有所不同，其中证明者显示了羽和 Z_s 在 0..S]范围内。如果证明者揭示了正确的 z_s，Z_s 这就意味着 $^{(e}$ij $^{-e}$ij$)^{s}$ij=£(i*j*) “j*%j*¯E “j*)^{e}i*j*^{s}i*j*= $^{-Z}_{s}$，所以（当
与 Aij=e%j-e “e[1；我们有那个 Ajjsije[-S..S]。重复时间，降低入境时欺诈的风险(i，j)到 e<E~-。

下一个问题是如何在每个并行运行中强制证明器显示正确的 z。，k=欧盟 152j=1eij，ksij。为了以一种高效通信的方式验证这一点，我们使用一个挑战 x 来批处理 t 并行运行在一起，以一次性显示 Zs、k ‘s 的正确性。

我们还必须避免揭露证人身上的秘密价值观。所以我们通过在揭示之前添加致盲因子晚来证明零知识

$$n\ m$$

$$s，k \quad /I/I\ e\ 礼^{j\ kk}_{F}\text{-i=1j=1}$$

现在，如果氣太大了，我们最终会得到 z，k 太大

了，这意味着我们不能再保证可以在指数中进行蛮力搜索。另一方面，如果晚太小，也许它不能很好地隐藏总和。为了解决这个问题，我们使用拒绝抽样[Gro05，Lyu09]。考虑随机选择 ak。计算结果为 z，k 属于范围[-S；Z+S-1]。我们可以将此范围分为两个不相交的部分[0...Z-1]和[-S；Z+S-1]\。在范围内[0…ak 的随机选择使每个 z$，k 的可能性同样大。此外，每个可能的和 Es%j 的概率与随机 ak 着陆 z$，k 在[0，Z-1]内的概率完全相同。所以想法是验证者检查每个 z$，k 在范围内。Z-1]，如果不是，验证者以新的随机性重新启动整个证明，然后再次尝试。重新启动不会泄露信息，因为任何总和 EUiE-ii，j，kSi，j 导致 z$，k 的概率相等，并且在 FiatShamir 启发式中是不可见的，因为它都发生在证明者的局部。[11]在运行 k 之外着陆范围的风险最多是 Z，这意味着我运行它最多是号。通过仔细选择参数 Z，我们可以确保重新启动的风险足够低，使期望的证明者很少重新启动，而且范围[0..Z-1]足够小，当给定 e%je[1-Z..Z-1]。

关于分块的 NIZK 证明

设置：参数指定带有发电机的级数顺序 p 的组 Gi

参数包括安全参数入和正整数 n、m、£、B、E、S、Z，使得 E=2“”/们、S=nm(B-1)(E-1)和 20S<Z<p2~(Z=20S，或者如果我们想将拒绝风险降低到 1/2 以下，我们可以选择更大的 Z），和 A=*nm£\X/^]*。

我们让 O 成为一个 oracle，在实现中将被实例化为哈希或扩展输出函数 H，在安全证明中将被建模为一个随机的 oracle。

实例：Gi 中的组元素

Yi，……，yn，...基体，...=，m= -

离散对数不是实例的一部分，但它们是由组元素唯一决定的，并被指示以供以后参考。

[11]虽然拒绝的机会是所有的和相同的，有可能一些和需要稍微快或更慢的计算，所以我们可以使用常数时间算法来防止时间泄漏，尽管在实际中，因为我们只终止一次语句不太可能足够的信息泄露给一个敌对的优势。

语句：实例的离散对数满足所有 i=和 j=1。有厶疆[1；这样的

$$\triangle 集^{s} i，\ j^{[1}—Z..Z^{-1]}$$

证人：离散对数，，r。、si，i，……、s满足所有 si，je[0..B-1]的约束
证明⁰（例如，证人）：
¯选择你的©⁶我，比，……，以©[—S; Z—¹],月 i，...，烷©ᶻp
—正在进行计算

双=gf[1],　　慈氏=舟疣[1]，...，玖=g 甲，堡=y0g?

——查询 O(instance，yo，Bi，Ci，...、Bg、a；A_e)，并将输出解析为 ᵉi¡i，i' ...，ᵉm，n，g[0E.. —[1]
—正在进行计算

$$\sum_{i=i}^{n}\sum_{j=i}^{m} +_b i,\qquad \sum_{i=i}^{n}\sum_{j=i}^{m}$$

并检查它们是否属于这个范围。如果他们没有，选择新鲜的 bi，……，bg©[-S；Z-1]，然后重试一次，以获得最多的入尝试。如果入尝试失败，请通过返回 n=丄来中止。
-选择乔，……，ˆn©Zp
—正在进行计算

做=gi 吗，　　$D_n = g_1^{\delta_n}$　　n 必 $\sum_{i=0}^{n}$

—查询⁰(ei，i，i，...，ᵉn，m出'ᶻs，i，...，ᶻs，g，DO，...，Dn，ʸ) 要获取 xe{0,1}入

$$\sum_{j=1}^{m}\sum_{k=1}^{\ell} e \qquad = \sum_{j=1}^{m}\sum_{k=1}^{\ell} e_{n,j,k} r_j x^k + \delta_n,\quad z_\beta = \sum_{k=1}^{\ell} \beta_k x^k + \delta_0$$

—正在进行计算
■= ■，...，　　DO，...，Dn，匕 s，i，...，r，i，...，r，n，　　r，i　　i，j，k r j* + ʲi，　　)
—删除在证明和返回期间创建的所有中间信息
n
项目页⁰（例如，n）：
-请检查该实例是否属于 Gn■并解析 n=
(yo，...，zf)e Gf+ⁿ+²x zp+ⁿ⁺ⁱ
-检查 z，i...，z，£e[0..Z—1]
——通过查询 O 来计算 ei，i，i，...、Cn、m、£和 x

-进行验证

^{h}R 尹-Di=gjw，……，有 r 户=1E”$^{k}D=gI^{rn}$ $_{j=i}$ $j=i$ h $B^{x}-D_{0}=gZ$ $_{k=i}$

以及

$$\left(\begin{array}{l} \backslash x^{k} n\ m \\ nn^{c}\ j\ I-n^{c^{x}-r}=n\ 妒，，•疗-g\ 尹 \end{array}\right.\begin{array}{l} \backslash \\ i=1 j=1 \end{array}\quad \begin{array}{l} £ \\ ) \end{array}\quad \begin{array}{l} n \\ k=1 \end{array}\quad \begin{array}{l} \\ i=1 \end{array}\quad \begin{array}{l} \\ .£ \end{array}\quad \begin{array}{l} \\ {}_{c=1^{x}s,\ k^{k}} \end{array}$$

—如果所有检查通过返回 T 来通过接受，否则通过返回上进行拒绝

证明尺寸、证明程序计算和验证器计算

通信：证明由 2 个+n+3 组元素和 n+1+1 字段元素组成，我属于较小范围[0..Z-1]。
证明计算：证明者计算 yis 的 gi 指数，yis 指数(忽略 ak 的小指数，期望重复小于 2 倍)，以及 n 全尺寸场乘法加上较小气讷元素的 nmt 场乘法。验证器计算：验证器成本主要是 tnm 宽的多指数到 ei、j、k 大小的指数(相当于 nm 范围的多指数)和 nm 宽的多指数(忽略大约 2t+n 指数和 ei、j、k 大小元素的 nmt 场乘法)。

安全性

定理 15。非交互式证明系统具有完整、健全、零知识性。

*证明文件。随机神谕模型的统计完整性：由于所有⑶册 E[0…E-1]和证人中所有 si，j，kE[0…B-1]，我们有所有 k=1，…，tEUiE；=i*i、j、ksi、jE[0..nm（E-1）（B-1）]C[0..S）。在值 yo，Bi 中有很多熵…，证明者选择的 Bˆ，所以之前进行的第一次甲骨文查询的概率最多是 qo/p₈。在每次重试中，证明者选择新的 Ci，…，Cˆ和多达入查询之间碰撞的风险是有限的。假设没有与先前的输入发生冲突，查询返回一致的随机值 e%j，k，它们独立于选择的 ai，……，ag。证明者均匀地随机选择所有的 ak，意思是

$$\Pr \quad \begin{array}{l} 纳米的 \\ s，k \end{array} \quad \underline{y?\ y?}\ {}^{e}i，j，k 細 {}^{\#}k^{e[0}..z^{-1]}\ i=i\ \ j=i$$

有了这样的选择，证明程序重新启动的风险就低于铲<1。在入这样的运行之后，重新启动的概率被 2 限制再加上遇到碰撞的风险可以忽略不计。

现在，检查验证者成功运行所产生的证明，我们看到实例和证明的 Gi 和 Z 中的组元素格式正确。此外，根据成功运行的定义，ˆe[0..Z—1]。将证明元素插入验证方程中，我们看到它们都得到了满足

$$\prod_{j=1}^{m}(g_1^{r_j})^{\sum_{k=1}^{\ell} e_{n,j,k}\, c^k}\cdot g_1^{\delta_n} \quad g_1^{z_{r,n}}, \quad \prod_{k=1}^{\ell}(g_1^{\beta_k})^{c^k}\cdot g_1^{\delta_0} \quad g_1^{z_B}$$

以及

$$\left(\begin{array}{l} \backslash x^k \\ g?\ \bullet n\ (\overset{n\ m}{舟}\text{殍})\ **\text{-}n\ 必\text{=}n\ \overset{\backslash}{舟}\text{-}疗\text{-}\overset{\pounds}{g}\ 尹\ i\text{=}1 j\text{=}1 \end{array}\right. \qquad \begin{array}{ll} n & n \\ ) & k=1 \end{array} \qquad i=0 \qquad i=1$$

*基于 Gi 中的 DDH 假设的随机甲骨文模型中的计算零知识：模拟器统一选择随机的 yo，Bi，Ci，...、B£、Ce 个 Gi。然后查询ˆ(实例，你，Bi，Ci，……，Ce；A)*来迎接挑战∎接下来，模拟器查询 Oprog（入）以获得一个挑战 x。然后它选择了 zi，...，z，£U[0……Z-1]和 Zr，i，……，z，*n，z ˆ个 Zp.*现在它计算唯一的组元素，…可以满足 n+2 验证方程的 Y。最后，它调用 Oprog(ei，i，i，...，en，m，e，zs，i，...，zs，e，Do，...，Dn，Y)来完成编程∎*1,1,1,...，0,...，*∎入)=ₓ（除非编程失败）。

为了看到模拟是好的，首先观察到，由于随机元素中的大量熵，我们遇到先前的甲骨文查询、遇到早期的查询或对随机甲骨文编程失败的 pOi+pnrr 风险要小。假设我们避免到达前面的查询点，元素 ei，j，k 是一致随机的，x 也是如此，这也可能是实证明的情况。

假设现在有一个区别者，除了在同一实例上的模拟证明之外，在在真实证人上说出真实证明方面具有优势。考虑下面的混合证明者，他知道证人，但行为就像一个模拟器。对于每个 k=1，…，t 混合证明器选择 z$，k.[0。并计算皿=z$，k-EU152-1ei，j，ki，j，之后她继续成为一个真正的证明者。我们观察到，在实证明和混合证明 z$中，k 最终都是一致随机的，因为在实证明中，ei，j，k-挑战和 zs，ke。Z-1]有一个有效的 ake[-S；Z-1]，将使它发生(因为在实证明中任何收集 ei，j，ke[0。E-1]和有效证人因 z$，k 越界而被拒绝的风险相同)。接下来，我们观察到混合证明和模拟证明之间的唯一区别是在模拟证明中加密的密文(Bk、Ck)的选择

随机群元素和在混合证明中加密了 g 仪。一个标准的混合论证表明，模拟证明和混合证明之间的区分器可以用来建立一个具有优势的 ElGamal(DDH)e/n 区分器。

随机谕模型中的统计模拟稳健性：创建有效证明的欺骗证明者必须提供一个格式良好的实例和证明，包括 Gi 中正确数量的群元素和 Zp 中的字段元素，其中 zi，e[0..Z—1]。此外，由于如果该值已经被编程，验证器会拒绝，所以在假证明中使用的两个查询必须通过对 O 进行查询。

证明程序现在有三个选项可以尝试：它可以使用正确的 Zs，k=EUiEm=ii，j，k“它可能希望进行顺序查询，所以它在第一轮之前学习 x，或者它可能尝试使用 Zz，k=E^=1Ej=iei，j，k证明查询按顺序进行。我们从引理 1 中知道，中间选项的概率由彖限制，所以现在让我们来看看第一个和最后一个选项。

假设对手正在 y 上第一次查询 O 的阶段、Bi、Ci、Bg、C，以获得挑战 e，e。如果它的概率大于在有效证明中使用正确的 Zs、i、、zs、k、k=En=i、m=i，j，k+Qk，那么也必须是 Zs，i……，zsge[0..…Z-1]。现在，我们知道挑战 e%j、k 是一致随机的，所以如果概率高于 E，这意味着每对(i，j)都有两个挑战元组(ei 丄加...弭第，加...弭払皿帝)和(ei，i，k，…，eijk，…，en，m，k)，它可以使用正确的 zs，k 值。在不失去一般性的假设下，假设^，j，k>jk，相应的答案是 z，k，z'sk e[0..Z—1]。这意味着对于该索引（i、j），有一个索引 k Z..Z。

最后，假设对手在证明中使用顺序查询，至少有一个不正确的 Zs，k=EUiEm=ie%j，ki，j+Qk.让我们分析当它调用 O(ei 丄 i 时能够提供有效证明的概率，…，enmg，Zs，i，…，F，做，…，Dn，Y；入)得到挑战 x。取第一个 n

$$r,\ i \qquad \begin{matrix} g\ m \\ i,\ j,\ k \end{matrix} \qquad \begin{matrix} g\ m \\ n,\ j,\ k \end{matrix} \qquad = \sum_{k=1}^{\ell} \beta_k x$$

个+1 验证方程的离散对数

看看 yi 的指数，……在最后一个验证方程中，我们看到它们等于 yj，……，，

$$g_1^{\sum_{k=1}^{\ell}\sum_{i=1}^{n}\sum_{j=1}^{m} e_{i,j,k} s_{i,j}} \qquad g_1^{\sum_{k=1}^{\ell} \sigma_k x^k} = g_1^{\sum_{k=1}^{\ell} z_{s,k} x^k}$$

yn'"“双方平等，因此取消。在一个有效的证明中，剩下的是平等

通过施瓦茨-齐佩尔引理，随机选择 x 的这个等式保持的概率最多为灸。通过对对手的 qo 查询，我们将这类欺诈的概率限制为上限。 口

7 非交互式分布式密钥生成和密钥重新共享

分布式密钥生成协议使一组参与方能够一起生成一个公共密钥和密钥的共享。DKG 协议由一组经销商运行，他们的目标是生成一个公钥，并为一组提供匹配的密钥秘密共享的接收器。作为经销商的参与者组和作为接收人的参与者组可能是相同、重叠或不相交的。

我们希望 DKG 协议是非交互的，即，经销商只是创建交易，不再与接收器或相互交互。接收方和其他各方可以结合在一组交易中提供的公钥材料，以获取阈值签名方案的公钥。接收方还可以从交易组中检索他们对签名密钥的秘密共享。除了查看一组交易之外，接收器还不会与其他参与者进行交互。

如果已生成了一个公钥，我们可能要保留它，但要重新共享该密钥。在这种情

况下，我们假设经销商已经拥有密钥的共享，但他们希望运行一个分布式重新共享协议，为一组接收器提供新的密钥共享。我们将这两种可能性纳入我们的系统；当经销商想要创建一个新的交易时，他们调用没有输入的处理算法“-”来表示他们还没有份额，而在重新分配中，他们用他们的密钥调用处理算法。

我们有两个理由对重新分享很感兴趣。第一个原因是，有时股票持有人可能会改变，然后我们需要一种方法，让现在的股东向未来的股份持有人秘密分享。第二个原因是主动安全，即持有密钥共享的参与者定期刷新这些共享，以防止股票的偶尔泄漏随着时间的推移而积累，并导致系统妥协。

设置：参数指定了一组可能的索引，为了简单起见，我们将假设它是[1。N]和最大时期数 T。

KGent(pk，dko)：随机密钥生成算法，返回公共加密时代 t=0 初始化的密钥和私有解密密钥。

KVfy(pk)tb：确定性密钥验证算法，如果公钥有效，则返回 T，否则则返回丄。

KUpd(dk),*t dk*≠i：以 tt 的解密密钥(解密密钥唯一确定相关的 epoch)作为输入，并将其更新为 t+1 的解密密钥。如果作为输入提供的解密密钥是针对 t=T-1 的，则更新调用返回丄以指示时代已达到限制。

Deal(？sk，t，pki，...，pk，t)td：给定一个阈值和一组具有 n<n 的公钥的随机处理算法，可以生成一个给定时期的处理。它将一个密钥 sk 作为可选输入，用于重新共享交易，或者如果省略，将创建一个新的交易。

DVfy（？？　　．，*pk*确定性交易验证

如果交易 d 被认为有效，否则返回丄。它将一个共享验证密钥 shvk 作为可选输入。其目的是，shvk 可以用于测试重新共享交易，而在测试新交易时被省略。理智检查输入是很自然的，所以我们假设处理验证算法只能返回正整数 t<和<N[0。T-1]。我们还要求一个一致性属性，即包括一个可选的股票验证密钥，使处理验证比没有股票验证密钥的处理验证更严格或更严格。

(n、n、I、d\、…)t(vk、shvki、shvk)：确定性算法，给定一组不同的索引 ii<…<和相应的交易返回一个公共验证密钥 vk 和共享验证密钥 shvk、、…、shvkn。

VKVfy(t，vk，shvki，...，shvkn)tb：确定性算法，给定阈值 t 和一组验证密钥，如果被认为是有效的密钥将返回 T，否则返回丄。该算法只能在正整数<<上返回<。

滑雪检索（j、dk>，I，d\，...，dg，T)tsk：给定解密密钥、大小和匹配处理 d、、、dg 为给定索引 j 返回一个秘密共享签名密钥 sk。...

SKVfy(sk，shvk)tb：确定性密钥验证算法，如果该密钥对于共享验证密钥有效，则返回 T，否则返回丄。

正确性。该协议是正确的，如果

—密钥生成会生成有效的公钥

公关[（pk、dko）—KGen：KVfy(pk)=T]=1

-通过有效的公钥进行的交易均有效。更准确地说，如果 1<n<N 和 te[0..T-1]和 pk、，...，pkn 是有效的公钥，因此 KVFy(pki)=T 和 SKVFy(sk、shvk)=T 或(sk、shvk)=（-、-），则

(? skt，pki，...pknt)：Dvfy(? shvkt，pk\，...pknt，d)=T]ml。

—当提供可选的股票验证密钥时，交易验证将更加严格。如果为 DVfy（shvk、t、pk、、、pkn、d）... 然后是=T *DVfy(-，tpk1…pknT，td)=T*。

-有效的交易会导致有效的验证密钥。如果 t，pki，…，pkn，t 和交易，……都满足需求。，=T 和索引集 IC[1…N]有大小&然后

$$\Pr \begin{matrix} (vk、shvki、、。..(t、n、I、d、、、)。。., dg) \\ VKVfy(t、vk、shvki，...、shvkn^{l})=T \end{matrix} \quad 1.$$

-诚实的接收者应该能够从一组有效交易中检索有效的密钥。我们定义了一个检索对手 A 的优势是

(pkdko)-KGen；(jIpki…，pkndi，　　)—A 单（*pk*、*dko*）
(vk，shvki，...，shvkn)JVKCombinge(t，n，I，di，……，dg)
Pr　　滑雪试验（j，dk/，*I*，*d*\，...，*dg*，*t*)：
我 C[1..n]和|I|=和 pkj=pk 和所有交易都有效，即 DVfy(-，t，pki，…，pkn，t，di)=T 和 t′<t，但 SKVfy(sk，shvkj)=T

在每次对 Kupd 的呼叫时，甲神谕设置 dkˆ+i：=Kupd(dkT/)和 t ‘：t’ = ‘+1，并在 t’+1=t 后停止对进一步的呼叫做出反应。甲神谕通过向对手发送解密密钥来响应呼叫。[12]

验证钥匙保存。如果在重新共享后我们仍然有相同的验证密钥，则该协议将保留验证密钥。正式地，对于正整数 t<n<N，t ‘<n’<N，验证键 vk，shvki，...，shvkn 与 VKVfy(t，vk，shvki，...，shvkn)=T，索引集 I 包含 ii<…，它，时代[0…<-1]，公钥 pki，…pkn 和有效交易，…与 DVfy(shvki，札 pki，……，=T，我们有

公关[(vk ‘，shvki、……，*shvk′ₘ*/)JVKCombine(t，n，我，d\，…，dt)：vk ‘=vk]=1。

7.1、施工情况。

我们现在提出了一个具有前向保密性的非交互式分布式密钥生成和密钥重分发协议。它建立在一个隐式的 fs-CCa 安全加密方案之上，从中我们得到了密钥生成、密钥验证和密钥更新算法 KGen、KVfy 和 KUpd。它还打算用于 BLS 阈值签名，后者使用 VKVfy，SKVfy 算法来验证生成的密钥。我们在这里以一种独立的方式呈现了完整的协议，包括这些算法。

设置：包括已知质数阶 p 的组 Gi、G2、Gt，并具有配对 e：gixG2TGt 和生成器 gi、g2、e（gi、g2）。BLS 阈值签名方案使用哈希函数　：(0,1}*t G「

参数定义了最大周期次数 T=2和一个定义指数集的绑定 N<p[1…N]协议可以分配给参与者，这意味着 N 是我们在单一交易中可以拥有的最大接收器集。

该设置还指定了组元素 fo，...、fx、heG2，它们在我们的 CCA 安全加密方案中使用，具有正向保密性。组元素隐式地定义了一个函数 f：　tG2　给　出　的 f(ti，...，t 人)：=

[12] 我们假设所有的交易都相对于同一目标时代运行。这是可以放松的，可以想象交易是针对不同时期的交易，但在这种情况下，定义必须稍微调整一下。

所以我们会经常利用的。[13]作为加密方案参数的一部分，有一个块大小为 B>2。我们让 m：=\log_B(p)“|，意思是 B>p.该加密方案使用了哈希函数 H\,且参数入应满足入=入 t+入 h 的要求。该加密方案使用了第 6 节中 Gi 中一个元素的离散对数知识的仿真可提取的 NIZK 证明。

该构造利用模拟声音的 NIZK 证明来进行正确的秘密共享和我们在第 6 节中提出的正确的分块。NIZK 证明依赖于一个哈希函数 Hz：(0,1}*t Z，分块的 NIZK 证明包括计算附加参数的函数&E、S、Z、A。关于分块的 NIZK 证明也使用了一系列哈希函数 H\：(0,1}*t(0,1}，其中输出的长度 A可以由用户选择。

KGent（pk、dko）：选择 xUZ_p 和设置

$$y := 元$$

构造离散对数 ndlog 的知识的证明（y；x）。设置 pk：=（y，ndiog）。

选择 p-Zp。设置

$$dk：=(gP，gf\ fp，...，fP，h^{p})\&Gi\ x\ G\ 产$$

和 dko：=(0，dk)。

删除中间信息并返回（pk、dko）。

KVfy(pk)Tb：解析 pk=(y，ndiog)，如果 yeG1 返回 PFfydi° g(y，ndiog)=T，则返回丄。

KUpd(dk)，k)t dk≠k：在描述更新过程之前，让我们给出一个解密密钥 dk 的高级结构；。

前向安全加密方案建立在树加密方案上，为大小为 2 的二叉树加密。消息被加密到树的叶子，并且内部节点的解密密钥应该允许您派生出该节点下面的子树中的所有节点的解密密钥。具有 y=g®和节点 Ti 的公钥的解密密钥的结构。“在二叉树中的高度 0<入是

$$^{\sim T}i.\ ..^{r}£,\ g^{P\sim}\ g2 \qquad f_{\ell|}^{p}\ i，...，fP，h^{p})e(0,1\}，x\ G1x\ G_{\frac{X-£+2}{2}}$$

$$^{dk}T\backslash...T=(^{\prime}1\ ...”，a，b，\ \ d\hat{}+i，...，\ \ ^{d}\backslash,^{e)}$$

---

[13] 组元素 可以用哈希函数 H 生成■

{0,1}*T 让第三方相信我们没有任何袖子。在中的

随机的甲骨文模型，这给我们提供了一组随机的组元素（直到研磨）。

如果使用协议中的算法生成解密密钥，则将从 Zp 中统一随机选择 p。[14]

[14] 在密钥生成算法中，dk 是这种形式的二叉树根的解密密钥。

从一个解密密钥 dk,.. 通过选择 6 个 Z，可以为子树中的任何节点获得一个完全随机的解密 key.&力...乓+ˆ和设置

析 -g1，b- n d? • (fo H)气 d 牛，+1 • ，...，d_A •f，e•h

新的解密密钥具有随机性 p+6，它在 Zp 中是一致随机的。

在具有前向保密性的加密方案中，前缀 tir\ ... 表示解密密钥工作的历元。解密密钥 dk，与 te[0..因此，T-1]必须允许我们派生密钥 dk，即…T 入了，还使我们能够为高度入 t 子树中的所有后续叶派生密钥。对于任何 te[0..T-1]让 TT 是节点 t、t(与 t<入 t)的最小集，这样它们的子树就不相交，覆盖了所有的叶子。T-1]。解密密钥 dkT 的格式为...

^{dk}T := (',(dk 钛 T£}斜时间£GTˆ).[15]

密钥更新算法给定 dk 和 k 使 t+k<T。从树解密密钥集{dkˆ。E”}力……互顷它使用子树中的相关密钥推导随机解密密钥 dk，i…所有ˆ。+k\TT。然后，它会擦除中间数据并返回 ..

dk T+k=(T+k,(dk 钛 T£i 钛……T£或+k)，

如果解密密钥格式错误或 1，更新算法返回±。[16] [17]

交易(? sk、t、pki、。..，pk_n，t)T d:

将输入解析为? sk=-或 skeZp，te[1…n]与 n<N，pki=(yi，n”与 yieGi，te[0…T-1]如果失败，则返回丄。17

-如果 sk 不存在，请随机选择 sk 个 Zp

... -在二进制文件中解析 t=t、t 人了。

-设置 ao：=sk，随机选择 ai，…，a(_iUZp

-计算 si，…，si=Ek=0国防部长 p

-用 b 字符号写入每个 si，即，si=52ˆ=1si，jwishs%je [0..B—1] -

-选择随机性，ri，si，……，r_m，s_m U Zp

-计算……，硅，硅，…，Rm，^{S}m如:

q，j：= ^{y}i ‘，gj’ ‘ ^{R}j：=gi ^{S}j =9i^{3}

计算结果 ^{m}1,.…，Pn，ˆ1，1，•••，Cn，m，

— 计算 f：=f（标题） ...

-计算字、……、Z_m as Zj：=f 七 h^{Sj}

-计算 Ao：=g；……，A(_i：=g；$^{t-1}$

— 计算 r：=Em=i ，国防部长 p

—计算 R：=gj 和 Ci：= -g? ，...，Cn：=y；-g^{Sn}

— 构造一个正确的秘密共享证明

---

[15] 在密钥生成算法中，可以看到 dko 有这种形式。

[16] 因为解密密钥是完全随机运行 dk，k）相当于 dk，1)，...，dk≠k-KUpd(dk+k-i，1)所以在安全定义中，我们只使用一步更新，并通过定义 KUPd(dk 来省略跳转 k，1).

[17] 该处理算法没有显式地检查 yi 的离散对数知识的证明 n，假设这种检查已经在其他地方进行过。

共享七~PrOV6■, . . . , 0, . . . , ■1, . . . , ■1, ... , S；）

— 构造一个正确的分块证明

■, . . . , ■1, . . . , ■1,1, . . . , ■...■.., ■

-删除中间信息并返回交易记录

d：____f [C]1,1, . . . , [C]n,m,[R]1,[S]1 . . . , [R]m, [S]m)

y [Z]1, . . . , [Z]m, [A]0, . . . , [A]t_1, [n]公司所持有的股份，[n]数据块丿

DVfy(? shvktpk1..。，*pkn，t，d)：*

将输入解析为? shvk=-或? shvkeG2，tE[1…n]与 n<N、pki=(yi，ni)与不同的 yiEG1，te[0..T-1]，如果解析失败，则返回±。[18]

-检查交易的形式

d：_____/ [C]1,1, . . . , [C]n，m，[R]1,[S]1, . . . , [R]m, [S]m)

y [Z]1, . . . , [Z]m, [A]0, . . . , [A]t_1, [n]公司所持有的股份，[n]数据块丿

带有 Ci、j、Rj、SjEG1 和 Zj、AkEG2

—如果存在可选的 shvk，请检查 shvk=Ao

■.. A： ■1, . . . , n，C*1,1, ...， ■1, ... , ^^m， ■)

-设置 f：=f(t1，。 .., T)

—验证每三个参数（R1、S1、Z1）、...、（Rm、Sm、Zm）的 e（g1、Zj）[e(Rj,f)] - [e(Sj,h)]

-通过检查来验证正确的秘密共享的证据

毫米 *m*

光伏 V[Cy]股份[Cy]1, ,'''[y]n, [A]0, . . . , [A]t_1,]] [R]*j* ,]] [C]1, j ''', ,]][C]“分享)=丁

j=1 j=1 j=1

-通过检查来验证分块的证明

项目页[区块](y1，...，yn，R1，. . . , [R]m，[C]1,1, . . . , [C]*n，m，*[n]块）=T

[18] 处理验证算法假设 yi 的离散对数知识的证明 n 已经在其他地方完成了。

-如果所有检查都通过，则返回 T，否则将返回丄

VKCombione(t，n，I，做了^)：给定 1<t<n<N，和一套 I 的 I 索引 *1<ii<……<* <n 和一组交易

■j：=(...， ...， — ...)

使用所有的 Aj，[k]eG2 计算 Aq，…，在 1 点

Ak：=血^[(Q)] (j=1)

—设置 vk：=Aq

——*计算 shvki，...，shvk*[n] 如：

t—1 ⅈ=n Ak k=Q

——如果所有的工作都返回(vk，shvki，...，shvk)，否则将返回丄。

VKVfy(t、vk、shvki，...、shvkn)：我们回顾了 BLS 阈值签名方案中公钥材料的有效性条件。检查 1<和<和 vk、shvki，...、<和 G2。设置 shvko：=vk 和 J={0，…，t-1}。对于我=，…，n 检查是否

=U .

如果所有的检查都通过，则返回 T，否则将返回丄

滑雪检索（i、dk/，K，di，……，dˆ，r)：将每个交易解析为

k，1，1，...，Ck，n，m，...，k，1，...，k，1，...，k，m，……)，与 Cki，J，Rk，J，Sk，jeG1 和 Zk，jeG2 在一起。检查 1<i<和<N。

对于 k=1，……，我定义了丁加 1=1,...，ᵣ = ᵀᵣ 和计算

k... ᴵᴵ1，... ，Pn，Ck，1，1，...， k，1，...，

k：=/(丁 k，1，...，丁 k，X)。

假设 t ‘<没有从 dkˆ的解密密钥更新中推导出一个 BTE 解密密钥

*k，1 . . . k，A，k， k， k)*X 2

对于 k=1，……，£。[18]

[18] 为了提高计算效率，可以在密钥推导中被省略随机化步骤，即，密钥推导可以使用 8=0。当 SKRetrieve 在验证正确的交易上运行时，我们有 e(gi、Zk、j)=(Rk、j、f)-e(Sk、j、h)，因此关键随机化中 8 的任何选择都会产生下面相同的 Mk、j 值。假设派生密钥和中间数据在使用后立即被删除，检索算法因此返回相同的结果，并且不会留下可以用来将几个非随机派生密钥补丁在一起以给对手优势的跟踪。

对于每个 k=1，…，£和 j=1，…，m 计算

[illegible]

然后用 Sk、je{z/A\Ae[1..婴儿步算法进行暴力搜索，使 M，j=计算

$$sk：=£sk，jB^{\sim}1$$

m

j=i

版。

解析 KC[1…n]作为不同的索引 k1<…<k^和计算

$$\text{硅}_{i}\ \prod_{j=1}^{\text{d}}\text{j（0）。}$$

擦除中间数据，如果一切顺利，请 返回 sk：=Si，否则将返回丄。

SKVfy(sk、shvk)：我们回顾了基站阈值签名方案中秘密共享签名密钥的有效性条件。如果是 Z*和 shvk=g；*ₖ 删除中间数据并返回 T，否则返回丄。

定理 16。构造正确的。

证明文件。关键键的正确性。我们从底层的 fsCCA 安全加密方案继承密钥正确性：密钥生成生成 pk=(y，ndiog)的密钥，并证明离散对数 x 的知识，从而证明 y=gX。键验证检查了 y 的离散对数的知识证明，并由于其完美的完整性，总是接受。这意味着密钥生成返回有效的密钥，Pr[(pk，dko)JKGen：KVfy(pk)=T]=1。

通过有效公钥进行的交易有效。更准确地说，如果 t<n<N 和 te[0…t-1]和 pki。，pkₙ 是有效的公钥，所以 KVfy(pki)=T 和 SKVfy(sk，shvk)=T 或(sk，shvk)=（-，-），然后我们想要

(? skt，pki，...pknt)：Dvfy(? shvkt，pki，...pknt，d)=T]=1。

首先，观察到，如果 SKVfy(sk，shvk)=T，那么 skeZp 和 shvk=g2。在处理算法中包括 sk，使 Aq=g$ₖ 在交易验证中包括 shvk，添加了检查 Aq=shvk，这是真的。因此，对于有效的共享签名密钥对（sk、shvk），交易有效的概率不受影响。剩下的是确保当我们在验证中省略 shvk 时，交易是有效的。

KVfy(pki)的前提是确保公钥是 pki=(yi，ni)与 yieGi。与其他先决条件 t<n<N，te[O..T-1]和可选的 sk，它们必须属于 Zp 才能有 SKVfy(sk，shvk)=T，我们看到处理算法的输入是如预期的。通过处理算法的步骤，NIZK 证明的完备性（除了在块证明中有小的完备性错误）意味着它终止没有

证明是丄，因为我们给有效的证人正确的秘密共享和块。产生的交易形式

d—/　C1，1，•••，Cn，m，　R1,S1，...，Rm，　Sm）
\ Z1，．．．，Zm，AO，……，在 1 点，n公司所持有的股份，n数据块丿

查看处理验证算法,前提条件 1<n<N 和 te[0..T-1]和 KVfy(pk“=T 确保它解析输入没有问题。然后，它检查处理的格式，并看到上面给出的格式对于给定的参数确实是正确的。

处理验证算法具有计算 t\所需的所有值 ≠i……t 人，然后计算 f。通过构造处理算法中的密文，我们看到检查 e(Rj、SJ，Zj)=(Rj、=)-e(SJ、h)持有，和 Rj=g[」和 Sj=gi。

最后，正确的秘密共享证明具有完全的完整性，并验证了它是否正确。除了较小的完整性错误外，分块证明也是如此。因此，处理验证算法返回 T。

*有效的交易会导致有效的验证密钥。如果 t，pki，...，pkn，t 和交易 di，..。*所有满足 DVfy(-，t，pki，...，pkn，t，dQ=T 和 IC[1..N]大小 t

公关[(vk、shvki、。..JVKCombine(t，n，I，di，…)(t，vk，shvki，。..，)=T]　　1.

要看到这一点，首先观察到从处理验证，参数是有意义的，即 1<t<N，每个处理 dk 包括元素 Ak，o，…，Ak，t-ie　　因为索引集 IC[1。N 的大小是 t
预期，组合算法计算公式，…，成功。然后它设置了 vk=Aq，我们会认为它是 shvk°，和 shvkj：=nklAk=ga，对于隐式度 t-1 多项式 a(j)和 j=1，…，n.这意味着共享验证密钥的离散对数位于度 t-1 多项式上，由于拉格朗日插值的性质，验证成功

$$a(i) \quad \sum_{j \subset J} a(j) L_j^J(i)$$

多项式
J={0…，t-1}。

诚实的接收者可以从一组有效交易中检索有效的密钥。检索对手 A 的优势是

（pk、dko）-KGen；（我、我、我、……、pkn、di、..。。，dg，t)—A(pk，dk°）
(vk、shvki、、。..(t，n，I，di，。。.，dg)
Pr　　滑雪试验（j，dk，*I、di、……，dg*）：

我 C[1..n]和|I|=和 pk=pk 和所有交易都有效，即 DVff(-，t，pki，…，pkn，t，dk)=T 和 t' <t，但 SKVfy(sk，shvki)=T

在每次对 KUpd 的调用中，甲骨文设置了 dk，，+i：=KUpd(d&T/)和 t' ：=t‘+1，并停止对进一步呼叫的反应，一次 t'+1=T。定义说甲骨文神谕

应该将解密密钥给对手，但在我们的方案中，这是一个没有意义的点，因为对手已经有 dko，派生密钥是完全随机的。

由于交易是有效的与相同的参数 t 和键 pk1，..。，*pk*和 t 时代，它们都有相同的格式

$^{d}_{k}$ (Ck，1，1，，*Ck，n，m，*^{n}k，1，^{s}k，1，.，^{g}*k，m，*^{s}*k，m*)
\$^{z}$k，1，...，^{z}k，m，Ak，0，...，Ak，t-1，^{n}k、股份、^{n}k，块丿

由于 pk=pki，它已经由密钥生成算法正确生成，其形式为 pki=(yi，ni)和 yeG1，以及解密密钥 dk/对于这个索引，在定义中定义得很好。加密方案的完美正确性也确保了 dk“能够解密密文 Ck，1,1，...、Zk、关于索引 i，如果密文是正确生成的。

所有交易必须在概率的成功条件下验证有效。因此，处理中的分块证明保证了它们的可靠性错误，即它们有 Rk，j=　和 Ck，i，j=　9*宀‘与有效的块 sk，i，j。

假设分块是正确的，对密文的其他检查意味着它们的格式良好，即对于某些随机选择 rg、zk、jez 的加密算法的有效输出。因此，检索算法成功地学习了块 sk，j，1，...，sk，j、从

结合块，检索获得 s1，j，…st，jeZ从每一笔交易中获得的收益。对于每笔交易，股票证明 nk，共享保证到其可靠性误差，即 sk，i=ak(i)，其中 ak(X)是由 Ak，o 的离散对数定义的度 t-1 多项式，Ak，t-1。因此，因为这两个验证

密钥组合算法和共享检索算法使用了由集合 I 定义的相同的拉格朗日插值，我们得到了 si=£sk，iLf$_{fc}$（0）和
石维=　fm*$^{(0)}$　满足 shvki 的要求2.因此，与 sk：=si 我们有 SKVfy(sk，shvki)=T。

交易中获得的信息。

定理 17。施工具有完美的验证钥匙保存。

证明文件。假设 VKVfy(t，vk，shvk、，...，shvk$_{n}$)=t。由于检查保证了 shvkt,...，shvkn 可以用 vk 和 shvk、，...，shvkt 的指数中的拉格朗日插值推导出 shvkn_1 它确保有一个度 t-1 多项式 a(i)，使 vk=g；$^{(0)}$ 和 shvki=g“”。因此，对于任何索引集，I 包含 t　^{L}i-(o)
1<i1<…<它<我们有 vk=卩 j=，shvki。，

现在，给定 d1…对于 shvki1，...是有效的。　我们分别从有效性检查中知道
DVfy（shvki*、t）^{z}，pk1，...，pk$_{n}$他们使用 a1，o=shvki、，...，At，o=t　，
李安。(o)
套包。由于 VKCombine 计算 Ao=Y\j=1Aj，因此我们有 Ao=vk。由于新的 vk=Ao，我们看到了 vk=vk 和验证密钥被保存了下来。

8 安全性

一个阈值贡献的 签名生成 带有 a 的方案 匹配的 说
键低 算法 协议包括 中的 明
KGen，KVfy，KUpd，交易，DVfy，VKCombion，
VKVfy，滑雪检索，SKVfv，SKVFy，签名，签名，签名，签名。我们已经有了之前已经定义了正确性、唯一性和密钥保存，并假设该方案具有这些属性。

我们希望避免使用未经授权的签名。这意味着，如果我们看到消息上的签名，这应该是因为生产它涉及了股东的阈值。参与者可以自愿参与签名创建一个签名共享，也通过腐败和做对手的投标，或未能及时擦除她的共享签名密钥或疏忽更新解密密钥，这样它落在错误的手中。

进入安全定义，我们想象一个世界，诚实的政党可能会产生新的钥匙公开，即对手知道。他们匹配的解密密钥将对对手保密，除非对方被损坏。我们将假设公钥有足够的熵是唯一的，这样一个政党就可以通过其公钥唯一地识别，我们将让 Qpk 成为诚实的政党生成的公钥集，特别是当他们创建密钥时是诚实的。在某些情况下，对手可能代表不诚实的一方传播公钥。我们不能说太多关于这些钥匙，除非我们假设互联网计算机将检查所有在流通的公钥是否形成良好。

双方可以将其解密密钥更新到一个新的时代，并删除上一个时代的解密密钥，这样他们就不再有能力在过去的时代解密消息。只要他们知道他们永远不会再解密旧时代的密文，他们就应该这样做。我们跟踪该表单(pk、dk)的公钥记录中的解密密钥)，其中 dk是 pk 时代 t 的解密密钥。在不可原谅的定义中，我们通过让对手决定何时更新解密密钥来捕获它，只要这种破坏不违反我们希望密钥管理系统工作的假设。只要对手不能违反这些条件，阈值签名方案就应保持不可原谅的状态。

双方有时可以创建具有匹配的签名密钥的新阈值签名验证密钥。我们让对手决定这种情况何时发生，但要唯一地确定谁应该做什么，对手必须提供具有匹配阈值和时代的参与者配置。这对应于我们如何配置在互联网计算机上创建子网密钥的方式，并且可以对该配置进行一些理智检查。

我们使对手能够触发创建一个诚实的新交易，建模，一个诚实的一方运行 Deal(-，t，pki，...，pk，t)，其中的值来自现有配置。这一套 Qd 包含了所有诚实的交易。后来，对手可能会将诚实的交易与它自己的任意交易结合起来，形成一个文字记录，使互联网计算机识别出一个新的公众

键。在安全定义中，我们要求对手在文字记录中包括至少一个诚实的交易，因为如果对手进行了所有的交易，他们就不能对安全做出贡献。验证引用现有配置的文字记录，并检查其完整性。如果文字记录被接受，交易合并得到一个验证密钥 vk，在 Q 注册 。然后，我们让所有诚实的接收器提取他们的秘密共享签名密钥。我们在表单的记录(pk、ik、sk、t)中跟踪这些秘密共享密钥，其中 pk 是诚实方的密钥，id 识别配置，sk 是诚实节点使用 pk 检索到的共享签名密钥，t 是批处理的时代。

在其他时候，双方希望重新共享现有的秘密阈值签名密钥。在这里，对手还必须首先创建一个配置，但这次配置必须引用先前的配置和它正在构建的现有公钥的文字记录。对于重新共享，我们允许对手触发一个诚实的交易，这次引用(pk，id!)对于一个诚实的聚会，持有相关记录为(pk，id!，sk，)的共享签名键。诚实的节点然后运行 Deal(sk，t，pki，...，pk而诚实的交易也被添加到了 Qd 之中。同样，对手可以将诚实的交易与自己的任意交易结合起来，提供一份文字记录，禁止对现有验证密钥的新的秘密共享。每当这样的重新共享发生时，接收方应该验证重新共享建立在现有的先前文字记录上，并跟踪表格的文字记录中的关键演变(vk、id、t、pki、shvk、，...、pk，*shvk*，t)。

接收方可以使用共享签名密钥对任意消息提供签名共享。在我们的模型中，我们让对手控制这种情况发生的时间以及哪一方签署哪些消息。

在互联网计算机上，一个诚实的一方会在共享签名密钥过时时擦除它。当然，只要当事一方有一个解密密钥，允许它再次恢复共享签名密钥，擦除就没有多大意义。因此，我们只能使对手要求擦除一个诚实方再也无法再恢复的共享签名密钥。

最后，我们的安全模型旨在捕捉动态腐败，因此我们使对手能够打击腐败的政党。每当对手破坏一方时，她就会了解到该党拥有的所有有趣数据，即其解密密钥和所有未被删除的共享。之后，与该方有关的记录被删除，因为该模式只跟踪诚实的各方，而对手现在有权代表该方采取行动。

我们定义了锻造对手 A 的优势

(vk m<j)《—AKGen, KUpd,Config,Deal,Transcript,Erase,SigShare, Corrupt.

vkeQvk 和签名(vk、m、a)=T

Pr 对于所有带有文字记录的 id(vk、id、t、pki、shvki，...、pkn、shvkn、t)：
(有具有 t‘<t 或 t’>t 的腐败记录(C、pki、t‘)
i 以及未被删除的共享签名密钥记录（pki、id、sk、t），>><t
或 pkieQpk，或有签名共享记录（id、pki、m） 丿

要赢得胜利，对手必须在公认的验证密钥下生成一个有效的签名，即来自至少一个诚实的文字记录中的验证密钥

在交易中。对手可以通过使用一个它知道的共享签名密钥来获得帮助来创建这样的签名，因为它破坏了一个诚实的政党，通过从一开始就控制恶意的一方，或者通过看到来自诚实的政党的消息上的签名共享。我们说，如果对于所有与验证密钥相关的索引配置和文字记录，对手没有帮助超过与配置相关的阈值 t，那么对手将获胜。本定义所指的神谕如下：

克根语

（pk、dko）JKGen

如果 pkeQpk 让实验返回 TQpk：=^{Q}**pk u**{^{pk}}

存储公钥记录（pk、dko）返回 pk

辅助衬垫(pk)

查找已存储的公钥（pk、dk）$_{\gamma}$)，如果没有此类记录或 t=T，则返回⊥ Setdkt+1-Kupd(dkT)，并更新存储的公钥记录到(pk，dk，+1)，则返回 T

配置(? id ‘、t、pki、。..，*pk$_{n}$，t)*

如果已有配置记录（*、*、t、pki、……、pkn、T），则返回⊥

如果是 id! =—让 t ‘：=-1

否则就可以找到文字记录(vk，id!，，…，t ‘)，如果没有这样的记录存在，请返回⊥

如果 te[1.n]或 te[t ‘+1…T-1]返回⊥

对于 i=1，…，n 如果 KVfy(pki)=⊥返回⊥

如果公钥发生冲突，pk=pkj 为 i=j 返回⊥

存储配置记录（id、id!，t，pki，...、pkn、t）　　//在初始化时，尚未使用 id=0

设置 id：=id+1 并返回 T

交易内容（？pk，id）

查找配置记录(id、id!、t、pki、...、pkn、t)，如果没有返回⊥ Ifpk=-letsk=-以及如果 id ‘=-返回⊥

否则，请找到共享签名键记录(pk，id ‘，sk，t’)，如果没有返回⊥ dJDeal(sk，七，pki，……，pkn，t)

让 Qd：=QdU(d}并返回 d

文字记录（id，吗？I，由，...，dg)

如果已经有表格的文字记录(vk，id，t，……)，请返回⊥

如果一个交易重复 di=dj 与 i=j 返回⊥

查找配置记录(id、id!，t，pki，...、pk$_{n}$，t)，如果没有返回⊥

如果没有诚实的交易，请返回⊥

如果设置了可选的索引吗？我是—（不包括在内）

只允许它，当配置不构建在之前的配置上，如果 id！=—返回丄
如果有一个无效的处理，其中，DVfy(-，t，pki，…，pkn，t，d“=丄返回丄集 I：={1，…，£}
其他的
查找文字记录ᶻ，*id!*，*t*ᶻ，，，…如果没有或我=ᶻ返回丄解析 I 作为一组索引 1<ii<…ig<v，如果它失败，返回丄
如果有一些地方有 DVfy(shvk£，t，pki，…，pkn，t，dj)=丄返回丄
(vk，shvki，...，shvkn)JVKCombinge(t，n，I，di，……，dg)
$^{Q}vk^{i}= {}^{Q}vk\ U^{\{vk\}}$
存储文字记录(vk、id、t、pki、shvki、……、pkn、shvkn、t)
对于 i=1，…，n 如果有公钥记录(pki，dk)与 t<t
滑雪 J 滑雪检索(i，dkˆ，I，di，…，dg)
存储共享签名键记录（pki、id、ski、t）
返回（vk、shvk，…….，shvk)

擦除（pk，id）

查找存储的公钥记录（pk、dk），如果没有返回丄
查找存储的共享签名密钥记录(pk、id、sk、t‘)，如果没有返回丄，则删除共享签名密钥记录(pk、id、sk、t’)，然后返回 T

签名共享文件（id、pk、m）

查找存储的共享签名密钥记录（pk、id、sk、t），如果没有返回丄 shJSigShare(sk、m)
存储签名共享记录（id、pk、m）并返回 sh

已损坏(pk)

找到存储的公钥记录（pk、dk），如果没有返回丄存储，则找到损坏记录（C、pk、t）
返回公钥记录（pk、dk、）和所有共享签名密钥记录（pk、id、sk、t‘）并删除公钥记录

安全定义中的正确性属性神谕保留了正确性属性，例如，KGen 将产生有效的键 pk。看看文字记录甲骨文，我们特别注意到它总是产生有效的验证密钥。I.e，如果文字记录(vk、id、t、pki、shvki、…、pkn、shvkn、t)，
这只能通过调用成绩单来发生，然后 vk 和共享验证密钥 shvki 是用一个调用来构建的(vk，shvki，shvk).VKCombine(t，n，I，di，...，dg).成绩单甲骨文确保有一个 v，1<O<v<N 和 I 包含索引 1<ii<…<
<
v 和所有交易都是有效的，即 DVfy(-、t，pki，...、pkn、T、dij)=T。正确性意味着抄本总是产生有效的公钥，即 VKVfy(t、vk、shvki，...、shvkn)=T。

配置记录由配置谕创建。配置记录可以组织成树，在那里我们用 id 标记节点。配置记录（id、-，...）是树的根。而一个配置记录(id、id!、..。。)意味着 id 是 id 的孩子！。当协议具有验证密钥保存时，当调用转换树本时，配置树中的所有节点都将获得与根节点相同的验证密钥（或仍未分配验证密钥）。

引理 2。如果协议有验证密钥保存，那么当记录本存储文字记录(vk，id，..。在配置的基础上构建。它必须是一份文字记录′，id/，……)已经存储和 vk=vk‘。

证明文件。文字记录记录只能通过调用文字记录神谕来存储。为了一份文字记录。必须已经存储一个配置记录(id、id！、..。如果 id‘=——没有什么可证明的，因为它是配置树的根，但否则我们必须争论已经有一个记录(vk，id’，..。

甲神谕确实检查有一个早期的文字记录(vk，id，0，pk，…，pk[f]，shvkV，t‘）。如前所述，由于这是一个文字记录，我们必须有 VKVfy(0，vk‘，shvk’，...，shvkˆ)=T。它还从配置和文字记录中检查 1<O<v<N 和<N1<n<N。甲神谕检查 1<ii<..。<iˆ<v 和在配置记录中(id，..。，我们只能有 t[0。最后，甲骨文检查每个 dj(shvkj，七，pki，…，pkn，t，dj)=t。这意味着所有验证密钥保存的条件都得到满足，因此当它调用(vk，shvki，...，shvkn)JVKCombon(t，n，i，di…，dg)我们得到 vk=vk‘。口

放松动态安全虽然我们的分布式密钥生成可能对动态对手是安全的，但我们的安全证明使用了一个模拟参数，它会遇到选择性开放攻击问题。即，我们想使用加密方案的 fs-CCA 安全性来隐藏诚实接收者在诚实交易中获得的股份。但是，如果我们尝试使用加密方案的不可区分性并加密不同的虚拟共享，那么在接收器损坏时，对手会发现共享是错误的，与交易中的密文不匹配，而不是共享验证密钥。这迫使我们加密正确的共享，现在我们陷入了安全证明中的模拟中。我们解决这个问题的方法是限制对手，这样我们就不必提供这些股份了。具体地说，我们限制对手，使它不能在解密时损坏一个诚实的接收者

密钥是处于一个可以解密模拟交易的时代。此外，不可能以未被删除的份额腐败一个政党。

通过这种方式限制对手，我们引入了一个诚实锁：一个集合 H，它指定了在某个时期之前不能被破坏的各方。我们定义了一个轻松的锻造对手 A 的优势

(vk m b)《—AKGen, KUpd, Config, Deal, Transcript, Erase, SigShare, Corrupt.

*vk e Q*k 和签名（vk、m、a）=T

以及所有带有文字记录的 id(vk，id，t，pki，shvki，...，pk，*shvk*，t)：有一个具有|H|>的配置记录 Pr (id，…，H)n-t 和

我有一个腐败记录(C，pki，t ‘)与 t<t，或带有 t>和一个未删除的共享签名密钥记录(pki、id、sk、t)、pkieQpk，或者有一个签名共享记录(id、pki、m)，我们在其中修改配置和损坏神谕

或

配置(? id、t、pki、……、pkn、t、H)

如果已有配置记录（*、*、t、pki、……、pkn、t），则返回丄

如果 id=-lett ‘：=-1

否则就可以找到以前的文字记录。，t ‘)，如果没有返回丄

如果 t/{1，……，n^}或 t/{t ‘+1，……，T-1}返回丄

对于 i=1，…，n 如果 KVfy(pki)=丄返回丄

如果公钥发生冲突，pki=pkj 为 i=j 返回丄

解析 H={<，pk^}与 1<ii<…<%<n

如果有 pki，没有公钥记录(pki。，dk_{Γ}。_)返回丄

商店诚实记录(H，pki1，t)，…，(H，pki^，t)

存储配置记录（id、id!，t，pki，...、pkn、t、H）　　//关于初始化 id=0

设置 id：=id+1 并返回 T　　//Wlog 顺序标识符

已损坏(pk)

对于所有诚实的记录（H、pk、t）

如果有一个存储的公钥记录（pk、dk；/）与 t'<t 返回丄

如果有存储的共享签名密钥记录（pk、id、sk、t ‘）与 t<t 返回丄找到公钥记录(pk、dk；)，如果没有返回丄 Stooe 则损坏记录(C、pk、t)

返回公钥记录（pk、dk；）和所有共享签名密钥记录（pk、id、sk、t ‘），并删除公钥记录

我们应该如何解释这个宽松的安全定义？它说对手当调用交易（没有失去一般性我们可以假设交易发生在配置调用后)必须提供一个列表 H 的公钥将保持诚实直到解密密钥已经更新过去这个时代和所有之前秘密共享签名密钥已经删除，但这是现实吗？这取决于我们认为在世界上是现实的腐败类型。如果对手能在任何地方腐败任意一方，那么不，但在这种情况下，我们为什么会相信一个特定的是
现实的吗？另一方面，如果我们相信对手在任何时候都有一组政党，它可能会实际地腐败。特定类型机器上的漏洞，数据中心操作员在接下来的几个时期内容易行贿)，这是一个非常现实的静态损坏和瞬时动态损坏之间的安全模型。从交易到使用的延迟越短，双方将解密密钥升级到新时代，删除旧的阈值共享越短，放松的安全措施就越现实。这并不是说这在全世界都是现实的：也许每个数据中心运营商都有一个代价，他们将被贿赂并迅速采取行动；在这种情况下，对手可能针对任何节点，即使无法承担腐败的费用。当然，对手也可能会尝试一次 DoS 的攻击来阻止该系统，并扩展她可以腐败某人的窗口。

定理 18。我们的阈值 BLS 签名协议放宽了动态安全性。

证据草意图。正如前面所述，这也适用于放松的安全定义，配置可以被组织成树，其中每个配置都是树的根或引用早期配置的文字记录。此外，我们已经证明了我们的协议具有验证密钥保存，因此同一配置树中的所有转录本都与根配置的转录本具有相同的验证密钥。它还可以表明，对于在至少一个诚实经销商的参与下创建的验证密钥，它们成为由一个诚实经销商创建的不同配置树中的验证密钥的概率也可以忽略不计。因此，我们只需要关注对手针对唯一定义的配置树伪造签名的情况。

另一个有用的观察结果是，找到一个对两种不同配置有效的处理的概率可以忽略不计。配置甲骨文中的检查确保了没有元组 . ，*pk*，t)出现在两种配置中。计算元素的数量 Aq， 在交易中，我们看到 t 一定是这样的
在所有有效的配置中都是如此。此外，处理中的密文唯一地决定了 tj...，t 人，而后者又唯一地决定了 t 和消息摘要 t ...，T。由于消息摘要在密文验证中重新计算，抗碰撞性意味着 pki，……，pkn 是唯一确定的。

第三个有用的观察结果是，来自诚实交易的密文不能被复制。在一个诚实的交易中，加密的秘密共享定义了一个唯一的阈值 t。如上所述，它也与 pk1 绑定，…，pkn 和 t。因此，诚实经销商创建的密文与一个独特的配置绑定。现在，甲骨文专门检查所有的交易都是不同的。这意味着对手只能通过调整其中一个零知识证明来成功地在同一配置中复制密文。然而，由于它们是模拟的声音，所以这对对手来说是不可能的，除非对手知道被加密的完全秘密共享。

现在，让 A 成为一个锻造对手，它创建表单（-，...）的多达 qconfig 配置请求，即对于新的配置。如果概率至少为 1/q 配置，我们可以预先猜出具有特定验证密钥的配置树的根索引。如果一个人成功地伪造了这个特定索引的验证密钥，我们会保留它，否则我们会让实验失败。假设 Adv(A)是原始实验中的优势，而 Adv‘(A)是修改实验中我们猜测指数的优势。然后是 Adv(A)<qconfig-Adv‘(A)。所以在 g 配置的紧密损失，我们可以从现在开始分析优势 Adv(A)定义为

Pr 存在具有\H|>....、H)的配置记录(id、...n-t 和
{ 有具有 t‘<t 或 t’>t 的腐败记录(C、pki、t‘t)
i 以及未被删除的共享签名密钥记录（pki、id、sk、t），　　>⟨t

图 1]；（vk、m、a）-A

有记录（vk、id *)和(idg -、*)和签名(vk、m、a)=T
对于所有带有文字记录的 id(vk，id，t，pki，shvki，…，pk，*shvk，t)*
或 pki 牛 Qpk，或有签名共享记录(id、pk、m)　　丿

观察到，由于文字甲骨文的构造，记录(vk，id‘，*)意味着 vkeQ这样早期的条件是冗余的，因此被省略了。

．我们通过改变诚实各方从诚实交易中获得股份的方式，开始对修改后的实验进行分析。回想一下，甲骨文运行了一个诚实的交易协议(sk，t，pki，...，pkn，t)，创建一个记录的 deQd 这个交易建立在诚实的秘密共享和块加密上，将个人共享转发到公钥 pki…，pkn。只有在对这些公钥有预先注册的配置，并且配置的注册意味着所有公钥都有效时，才会进行处理。对于每个诚实生成的公钥 pkeQpk，都满足加密方案的正确性条件。因此，一个诚实的交易，解密任意更新的解密密钥匹配一个诚实生成公钥 pkieQpk，公钥可以正确解密（因为块有正确的大小），这样做你得到密钥匹配 si=E在这个特殊的处理中使用（通过构造块的明文）。因此，我们可以修改实验，使 Deal 创建的诚实交易存储他们的密文与匹配的明文一起，这碰巧是匹配 Ao 的秘密共享，……在交易中的 At-i。并相应地，每当滑雪检索(i，dk，I，di，...，dg)在文字记录中遇到这样一个密文，地址是一个诚实和未损坏的公钥 pkeQpk，它只是查找匹配的明文，而不是运行解密算法。完美的正确性确保了这种查找始终与解密相匹配，因此其优势保持不变。

接下来，让我们修改处理甲骨文交易，以模拟每当它用根 idg 引用树中的节点时，它在诚实处理中创建的密文的证明。新交易的甲骨文神谕是

交易方式ᶻ（？pk，id）

查找记录（身份证！！！！！　．，pk_n，t，H），如果没有返回⊥
如果 pk=-letsk=-和如果 id！=—返回⊥
否则就找到记录(pk，id！，sk，t‘)，如果没有返回⊥
如果 id 不在植根于 idsuce 的子树中
　dJ 交易(sk、t、pki、……、pkn、t)
其他的
　dJDeal(sk，t，pki，……与模拟证明“共享”，块
让 Qd：=QdU(d}并返回 d

从证明系统的零知识性质可知，这种变化对对手优势的差异可以忽略不计。

将焦点转向对抗性交易，让我们看看甲骨文是如何代表诚实的接收者处理它们的。甲骨文检查所有交易是否有效，这意味着敌对交易必须有正确格式的密文，并接受 NIZK 证明，以获得正确的秘密共享和分块。现在，让我们修改成绩单甲骨文让实验返回⊥以防有一个密文不是由诚实的处理甲骨文但有有效的 NIZK 证明，这样

解密一个诚实的 pkeQpk 失败或结果明文不匹配预期给 Ao，……在交易。由于分块证明表明我们的前向保密加密方案具有正确生成的密文，并且加密方案具有完美的正确性，因此未能解密意味着我们违反了分块的 NIZK 证明的可靠性。同样地，如果明文与 AO 定义的秘密共享不符，…，A(_i，那么我们违反了 NIZK 证明的合理性。从模拟的可靠性可以看出，这种情况发生的风险可以忽略不计，因此对手的优势几乎保持不变。

正如之前所说的，对手不能从诚实的交易中复制密文，所以我们可以认为从敌对的交易中进行共享签名密钥检索是使用解密神谕。所有诚实的接收方都可以解密敌对的交易，并通过正确的秘密共享证明，他们的共享与该接收方的公共共享验证密钥相匹配。根据对手优势的定义，Adv ‘(A)最多可以有腐败的经销商，因此 n>不是诚实的接收者。因此，从诚实的接收者的股份可以使用拉格朗日插值来重建与公共股份验证密钥基础的股份匹配的 t-1 多项式。使用组合股份验证密钥时使用的拉格朗日插值系数，我们现在知道一个 t-1 多项式，它表示反向创建交易对股票的贡献。我们可以利用它来取消对手对密钥生成所做的任何贡献。

现在，让我们将 BLS 签名方案中的哈希函数 Hˆ(m)建模作为随机谕。我们可以看到，对哈希函数的每个查询都是返回一个具有诉 JZ 的组元素 gf。如果我们编程随机神谕给我们这样的值，我们可以把卬看作是随机已知的场元素。我们现在可以改变代表诚实的政党分享信息的方式。对于必须共享签名消息的 pkeH，与其使用其秘密共享 Si=a(i)来签名，我们可以将签名视为使用 g*”，然后返回

因此，我们可以跟踪锁定诚实的政党的明文，而是跟踪他们股票 g“初的指数。

在这个阶段，加密方案和 NIZK 证明不再发挥作用。对抗性的交易也不一样，因为我们可以提取完整的秘密共享并取消它。此外，我们可以毫不丧失一般性地假设在每一份文字记录中有一个诚实的交易，只是在其他交易中扮演一个诚实的“对手”。剩下的是一个对手，在每个文字记录中看到一个诚实地处理组元素 Aq，A1 和秘密的ˆSi对于已损坏或可损坏的接收器。而对于诚实的政党，对手只看到 g“1，…，g-1但使用它对随机神骨文的控制来创建他们的签名共享。我们将展示，我们可以在这个场景中嵌入一个 DH 问题实例，这样成功的攻击者可以用来破坏 1MDH 实例。III 型对中的 1MDHP 设置可以表示为提供随机组元素 gf 和 gYO、GYO、……、g 献给对手。攻击者现在可以了
自适应地要求该形式的线性组合(r°，……，rg)e接收作为响应 g。“銘。攻击者获胜，如果在 q 查询后，她能够计算 gf。...，g也就是说，她需要计算一个比允许询问的查询数量更不同的海尔曼组元素。

好吧，让我们看看如何将 1MDH 问题嵌入到系统，以便使用成功的攻击者来解决 1MDH 问题。我们预先猜测，攻击者将使用随机神谕查询的索引来创建成功的伪造，如果我们猜测在减少的紧密性有限损失，将中止攻击。然后我们得到 1MDH 挑战 gf 和 g“，g；。，...，g：。，g；其中 q 是与相关配置树诚实处理的诚实锁定方数量的上限。现在，对于已猜测的配置，我们选择了 g*作为诚实的经销商对验证密钥的贡献，这将被敌对的交易修改，给 vk=gˆ对于一些已知的 aQ。在每一笔诚实的交易中，都会有一些腐败的政党有指数，我，我们只是随机选择了股票 Si1，…，Si。然后，在选择了这些之后，我们将为第一个 m=-c-1 诚实锁定指数 ji，……，j。选择以前未使用的元素 g；1,...，g；并将本交易中这些当事人的股份定义为未知的 Sj1=*+i，……，Sj从我们的 1MDH 挑战中，我们知道匹配的公钥材料 gˆ$^{k+1}$，...，gˆ$^{k+m}$)。现在，对于伪造的消息 m，我们编程的甲骨文返回 g来自 1MDH 的挑战。我们可以使用 1MDHOracle 来回答此消息的所有共享签名查询。对于所有其他消息，我们只对随机神骨文进行编程，给我们一个随机指数，使我们能够签署所有共享签名查询。同样，我们也可以很容易地消除敌对性交易的贡献。所以我们现在使用 1MDHoracle 得到了对协议的完美模拟。在
在一天结束时，对手现在可以返回伪造的签名，使得 e(b、g2)=(H(M)、vk ‘)=(g1、g2° $^{+aoa}$° ,32)-

由于°是已知的，这意味着我们可以计算

$$a-(g1)f^{\circ} = \quad -$$

所有的共享签名查询都与 70 个线性无关，因此计算这个值会导致对另一个 DH 实例的解决方案。 口

8.1 主动式安全性。

由于公钥仍然使用了很长一段时间，可以合理地假设每个股东在此期间都存在重大的妥协风险，无论是由于小故障还是恶意软件。如果秘密共享的寿命也很长，它们可能会一个接一个地泄露，如果对手收集了足够多的信息，她可以签署任意的信息。为了应对这种影响，很自然地实施防御，如监控各方的行为以检测偏离正常行为，或定期重置以清除未检测到的恶意软件。我们使用移动对手模型来模拟这种错误或恶意行为，在该模型中，对手在任何给定时代的时候都可能会损害各方，但不能同时控制太多的各方。I..，对手可能会在一个时代妥协给一些政党，但如果她已经达到了腐败的门槛，她也必须放弃对一些政党的控制，然后恢复诚实。在移动对手模型中，每一方都可能在某个时间点受到损害，只是同时不会有太多。

如果我们把妥协建模为对手完全控制一个政党，那么积极的安全保护就没有太大希望了。一旦对手得知一个解密密钥，它就可以解密未来一方作为接收方的所有交易。但是我们可以使用主动的安全性来防止攻击，比如由于故障导致的秘密共享的意外泄露、不影响解密密钥的非严重攻击导致的恶意行为，或者长期解密密钥受到安全硬件保护的系统。为了防止这种移动对手，股东可以使用非交互式分布式密钥重新共享协议来定期刷新他们的共享并删除他们的旧共享。一旦某一特定时代有足够的股份被删除，该时代的剩余股份将永远不足以重建密钥。这样，我们就可以将对移动对手的攻击面减少到一个时代的持续时间。

现在，虽然我们不给移动攻击者访问解密密钥，但恶意行为仍然可能涉及获得不同密文的解密；例如，即使解密密钥受到安全硬件的保护，移动对手可能会要求解密一些密文。因此，我们将安全模型更改为，对手可以危及参与者，以学习共享签名密钥，并获得对解密的访问权限

甲骨文组织。在形式上，我们定义了访问解密甲骨文的优势

(vk， 。A KGen，KUpd，配置，DealjTranscriptjErasejSigShare，Dec，腐败。

vk e Q,k 和签名（vk、m、a）=T

Pr 以及所有带有文字记录的 id(vk，id，t，pki，shvki，...，pk，*shvk，t)：*
（有具有 t‘<t 或 t’>t 的腐败记录（C、pki、t‘））

i 以及未被删除的共享签名密钥记录（pki、id、sk、t），>⟨t
或 pkieQpk，或有签名共享记录（id、pki、m） J

安全证明可以适用于覆盖这种更强大的安全形式。因此，我们获得了针对具有 CCA 的移动对手的主动安全——加密方案也可以学习给定时期的一些共享签名密钥。

确认情况

许多人指导了设计选择，建议改进文章，并在互联网计算机上实现协议，我们特别要感谢简·卡梅尼什、安德里亚·塞鲁利、大卫·德勒、马努·德里弗斯、玛丽亚·杜博维茨卡娅、本·林恩、马克西米利安·墨菲、格雷戈里·内文、弗朗茨-斯蒂凡·普雷斯、巴托斯、维克多·舒普和比约恩·塔克曼。

参考文献

bbg05。丹博内，泽维尔博因和金金。基于层次结构标识的、具有恒定大小的密文的加密。在罗纳德·克莱默，编辑，密码学进展 2005，第 24 届密码技术理论与应用国际会议，奥尔胡斯，丹麦，2005 年 5 月 22-26 日，论文集，计算机科学课堂讲稿第 3494 卷，第 440-456 页。施普林格，2005 年。

公元前 07 年。丹博尼，跑卡纳蒂，沙哈勒维，和乔纳森卡茨。从基于身份的加密中获得的选择密码文本安全性。爵士爵士。，36(5):1301-1328，2007.

第 04。丹博内，本林恩，和霍瓦夫沙查姆。来自威尔配对的短签名。J.密码。，17(4):297-319,2004 年。

第 87 页。保罗·费尔德曼。一种不可交互式可验证秘密共享的实用方案。1987 年 10 月 27-29 日，美国加利福尼亚州洛杉矶，第 28 届计算机科学基础年度研讨会，第 427-437 页。IEEE 计算机学会，1987 年。

gps08。史蒂文 D.加尔布雷斯，肯尼斯 G.帕特森，和奈杰尔 P.斯马特。为密码学家提供的配对。自由决定。应用程序。数学方法。，156(16):3113-3121，2008.

02 级。詹斯。格罗斯。在通用可组合性框架下评估投票方案的安全性。密码学电子打印档案，报告 2002/002,2002 年。,

05 级。詹斯。格罗斯。同态加密的一种可验证的秘密洗牌。密码学电子打印档案，报告 2005/246 年，2005 年。

左 09。卢巴什耶夫斯基。具有中止功能的菲亚特-萨米尔：应用于基于格子和保理的签名。松井三井，编辑，密码学进展-准丙烯酸 2009，第 15 届密码学与信息安全理论与应用国际会议，日本东京，2009 年 12 月 6-10 日。论文集，计算机科学课堂讲稿第 5912 卷，第 598-616 页。施普林格，2009 年。

沙 79。阿迪。沙米尔。如何分享一个秘密。提交。acm，22（11）：612-613,1979 年。

9 证人可以通过使用指数来检验 R 匹配 r、每对 yi、Ci 匹配 r、Si 和每个 卩 Ak 匹配 Si 来验证。