古阿斯的戒指：研究智能合约犯罪的未来

Ari Juels1) Ahmed Kosba2) Elaine Shi3)

1)(Institute of Computer Science, Cornell University, America)

2)( Institute of Computer Science, University of Maryland, America)

3)( Institute of Computer Science, Cornell University, America)

摘 要 由于它们具有匿名性，并且消除了值得信任的中介，类似比特币的加密数字货币已经在商业领域和某些区域取得成长或者刺激了发展。不幸的是，这其中的一些发展是犯法的，比如洗钱，犯罪市场以及勒索软件。下一代诸如以太坊的加密数字货币将会包含丰富的脚本语言来支持智能合约，一种独立地对交易起到媒介作用的程序。在这篇论文中，我们探索到智能合约引起新的犯罪生态系统的风险。特别地，我们将展示所谓的智能合约犯罪（CSCs）如何能够促使机密信息的泄漏，加密数字货币密钥的盗窃以及各种现实世界中的犯罪行为。（谋杀，纵火，恐怖主义我们将说明对机密泄漏来说，智能合约犯罪对于现有的脚本语言，例如以太坊中的脚本语言，是高效而可实现的。我们还将说明它对加密数字货币密钥的盗取，能通过使用原始的SNARKs算法实现，该算法已经可以在这些语言中表达，并且可以预期有效的支持语言扩展。我们同样说明了已认证的数据服务，一个智能合约系统的新特征，使得智能合约犯罪在现实犯罪中运用更加容易。（例如财产犯罪）为了实现将智能合约应用于有益的目标的诺言，我们的结果强调了制定政策和科技安全防卫手段的迫切。

关键词：智能合约犯罪；以太坊

古阿斯的戒指是一个虚构的神话故事，哲学家柏拉图在他的《理想国》第二册中提及过。它授予他的主人任意隐身的能力。——维基百科，“古阿斯的戒指”

“「当戴上这枚戒指时」没有人会拒绝从超市里安全地偷出任何本不属于他的东西，或是进入他人家中任意躺在他人身旁，甚至杀害他想杀的人或者帮助他越狱。”——柏拉图，《理想国》第二册（2.360b）译者：Benjamin Jowett

1引言

类似于比特币的加密数字货币移除了在现金业务方面对于可信任第三方的需求并且在个体之间提供匿名的转账（更准确地说：伪名）。虽然加密数字货币对很多应用都极具吸引力，但是它也有自己的黑暗面。以臭名昭著的丝绸之路【31】为先例，比特币引发了勒索软件【6】、洗钱【38】和不法贸易的增长。

新兴的加密数字货币，例如以太坊（还有合约方【45】和智能合约【1】之类的系统），甚至能提供比比特币更丰富的功能。他们支持智能合约，一个在图灵机的加密数字货币脚本语言中代表程序的术语。在一个例如以太坊的完全分布系统中，智能合约在没有被信任的第三方存在的情况下实现了一般公平交换（原子交换），并因此有效保证了对成功转达的信息和服务的支付。考虑到这样智能合约系统的灵活性，我们可以预期他们不仅会激发新型有益的服务，同时也会引发新型犯罪。

我们将智能合约中协助犯罪的分配型智能合约称为犯罪智能合约（CSCs）。其中一个例子就是（私人）密匙偷窃。比如，一个CSC可能会支付酬金来换取（机密）目标密匙sk，例如权限认证的私人数字签名密匙。

本文将会探索如下的几个关于密匙的问题：相比于加密数字货币（比特币），CSC能否造成更大范围的重大新型犯罪？这样的犯罪有多可行？还有相比于传统的网络系统，CSC能给犯罪分子提供怎样的密匙优势？探索这些问题对分辨威胁和设计对策都至关重要。

1.1犯罪智能合约难题

想要成为犯罪分子的人在建立csc时会面临两个基本挑战。首先，对于一个给定的犯罪，CSC是否可用并不是完全明显的。本文中将CSC的一种关键属性称为佣金合理性，意为执行CSC之后能同时确保犯罪和对犯罪者相称的支付发生或不发生，而要确定这一点是极具挑战性的（我们在本文中正式将佣金公平定义为对于个体来说的）。公平交易对确保佣金公平性是必要但不充分的：我们将展示CSC构建公平交易的同时仍然让第三方利益集团骗取利益。因此正确构建CSC是十分精细的工作。

第二，即使原则上一个CSC能够被建成，考虑到现存智能合约系统中有限的伪代码（例如以太坊），现在CSC作案是否实际可行尚不明朗。借此我们想表达的是CSC可以在没有过度负担的计算需求下被执行，而这种特征在某些智能合约系统中（例如：以太坊）同时也意味着对CSC征收不可接受的高执行费。

下面的例子阐释了这些挑战。

例1A （密匙折衷合约） 订约人C发布了一则要求偷窃并运送受害权限发布机构（CA）CertoMart的签名密匙skv的悬赏。C提出给予$reward于犯罪者P，作为（机密）运送CA私人密匙的回报。

为了确保密匙和回报使用比特币的公平交易，C和P需要使用可信任的第三方机构直接沟通，而这样会提高被背叛的或被执法机构发现的几率。他们也可以使用信誉系统互相审查，但大多数情况下这样的系统是已经被执法机构渗透了的。相反的，一个分散化的智能合约能够实现自我制约的交易公平。对于密匙偷窃，下面的案例是可能使用CSC密匙偷窃序列的。

例1B （密匙折衷CSC） C产出一个私人/公有的密匙对（skc，pkc）并且使用公用密匙pkc和pkv（CertoMart的公用密匙）来启动密匙偷窃序列。密匙偷窃序列等待一个特定的犯罪者P输入一对（ct，π），π是ct = encpk C [skV]完好建立的一个零知识证明。密匙偷窃序列之后核实π序列并且在成功后将报酬$reward发送给P。订约人C之后即可下载并破解ct来得到折衷密匙skv。

密匙偷窃序列在C和P之间建立了公平交易机制，在且只在P发送回可用的密匙后给出回报（通过π证明后），消除了对可信任第三方的需要。但这并不是交易公平的，因为skvict是否存在价值不能被保证。CertoMart可以通过先发制人，废除自己证书的有效性并最终自己获得C给出的$reward来使这种契约无效化。

正如上述的，本文的要点是展示对于类似密匙偷窃序列的CSC，罪犯是如何绕过这样的问题并最终成功建立佣金公平的CSC。（对于密匙妥协，如果密匙被无效化了，实现合约的无效化是必要的。）除此之外，我们将会展示使用现存的加密数字货币工具或现阶段加密数字货币实现的特征能够有效的实现CSC的使用。（例如：zk-SNARKS【22】）

1.2本篇论文

我们证明了在智能合约系统中构建佣金公平的CSC来进行三种犯罪活动是或将会是可能的：

1. 机密文件的泄露/买卖
2. 私人密匙的偷窃，还有
3. “名片“犯罪，广泛的物理世界犯罪（谋杀、纵火等）

事实上，CSC在原则上是可能的这个事实并不奇怪。以前，然而，我们并不清楚CSC是否实际或能否广泛使用。正如我们佣金公平的CSC显示，构建CSC并不像它看起来那样简单，但新的加密技术和智能合同工的设计方法，可以使之可行甚至实用。此外，犯罪分子对CSC的使用无疑会超越这篇论文或大多社区能够预期的。

因此，我们的工作就是展示社会考虑构建针对CSC的防御系统势在必行。假借匿名实施的犯罪行为构成了采用比特币的主要障碍。然而，很少有在关于数字加密货币【16】的公共论坛上讨论刑事契约的，并且以太坊的发布在2015年七月。只有在CSC生命周期的早期认识到它，社区才能及时的制定对策，并充分认识到分布式智能契约的美好前景。

虽然我们关注的重点是阻止恶行，皆大欢喜的是我们提出技术也能用于创造社会效益合同。我们探索构建CSC的技术和尖端加密工具的使用，例如，知识简洁的非交互式参数（SNARKs），在CSC中。像社会效益合同的设计一样，CSC的构建需要密码系统和佣金公平的精心组合。

总而言之，我们的贡献有：

•**犯罪智能合同**：我们使用图灵在下一代加密数字货币中完成的脚本语言启动对于CSC的研究。我们探索CSC中三种不同的犯罪类型：在第四节将讨论的泄露信息（例如：预发布的好莱坞电影），第五节将讨论的密匙的妥协/盗窃（例如：CA签名密匙），和“名片”犯罪，如使用第六节中被称为“验证数据提要”的数据源实施的暗杀（下面将会提到）。我们探讨制定此类刑事合同所面临的挑战，并展示（预期）抵制中和和实现佣金公平的新技术。我们强调，因为佣金公平指的是非正式的，缔约方获得“预期”的效用，一个应用程序特定的度量公平，对给定的CSC有一种特定佣金公平。因此，我们正式为我们在网上的完整版本CSC指定佣金公平[ 42 ]。

•**概念证明**：为了证明更复杂的CSC是现实的，我们报告我们探索的（在各自的部分）对干CSC研究的实施。在现存的智能合约语言系统中我们的CSC对泄密是很有效率的（例如以太坊）。那些钥匙盗窃和“名片”犯罪分别依赖效率和目前由加密数字货币社区现阶段实现的可靠性。

•**对策**：我们将在本文的第七节中简要讨论如何阻止CSC的大规模增加。简而言之，要使效率最大化，CSC必须被广告，使其具有可检测性，并给予社区警惕。由于CSCs会降低加密货币的市场价值，所以矿工有一个经济动机，不要把CSC交易包括在内。 因此，意识和强大的检测策略可能会提供有效的一般防御。 我们工作的一个重要贡献是展示这种对策的必要性，并在智能合约系统（如以太坊）中激发对其实施的探索。

我们还简单地在线完整版[42]讨论如何成熟的技术，如信任的硬件根源（如英特尔SGX [40]）和程序混淆可以丰富可能的CSC的空间 - 当然，他们当然可以有益智能的合同。

2 背景和相关工作

新兴的分散式加密货币[52,59]依赖于一种新型的区块链技术，其中矿工们不仅达成了关于数据的共识，而且关于计算。 松散地说，比特币区块链（即矿工）验证了交易并存储了一个全局分类账，这个分类账可能被模拟为一个公共存储区，其完整性依赖于正确执行底层分布式共识协议。 比特币支持区块链执行的有限范围的可编程逻辑。 但是，它的脚本语言是有限制的，而且难以使用，正如先前在比特币上构建智能合同类应用程序所显示的那样[23,17,7,53,46]。

当区块链（即矿工）所执行的计算被推广到任意的图灵完备逻辑时，我们可以获得更强大的通用智能合约系统。 这种分散式智能合约系统的第一个体现是最近推出的以太坊[59]。 非正式地，在这样的系统中的智能合同可以被认为是自主执行的代码，其输入和输出可以包括钱。 （下面我们给出更多的形式主义。）业余爱好者和公司已经建立了以太坊或分支来开发各种智能合约应用，如安全和衍生品交易[45]，预测市场[5]，供应链起源[11]和人群 集资[2,47]。

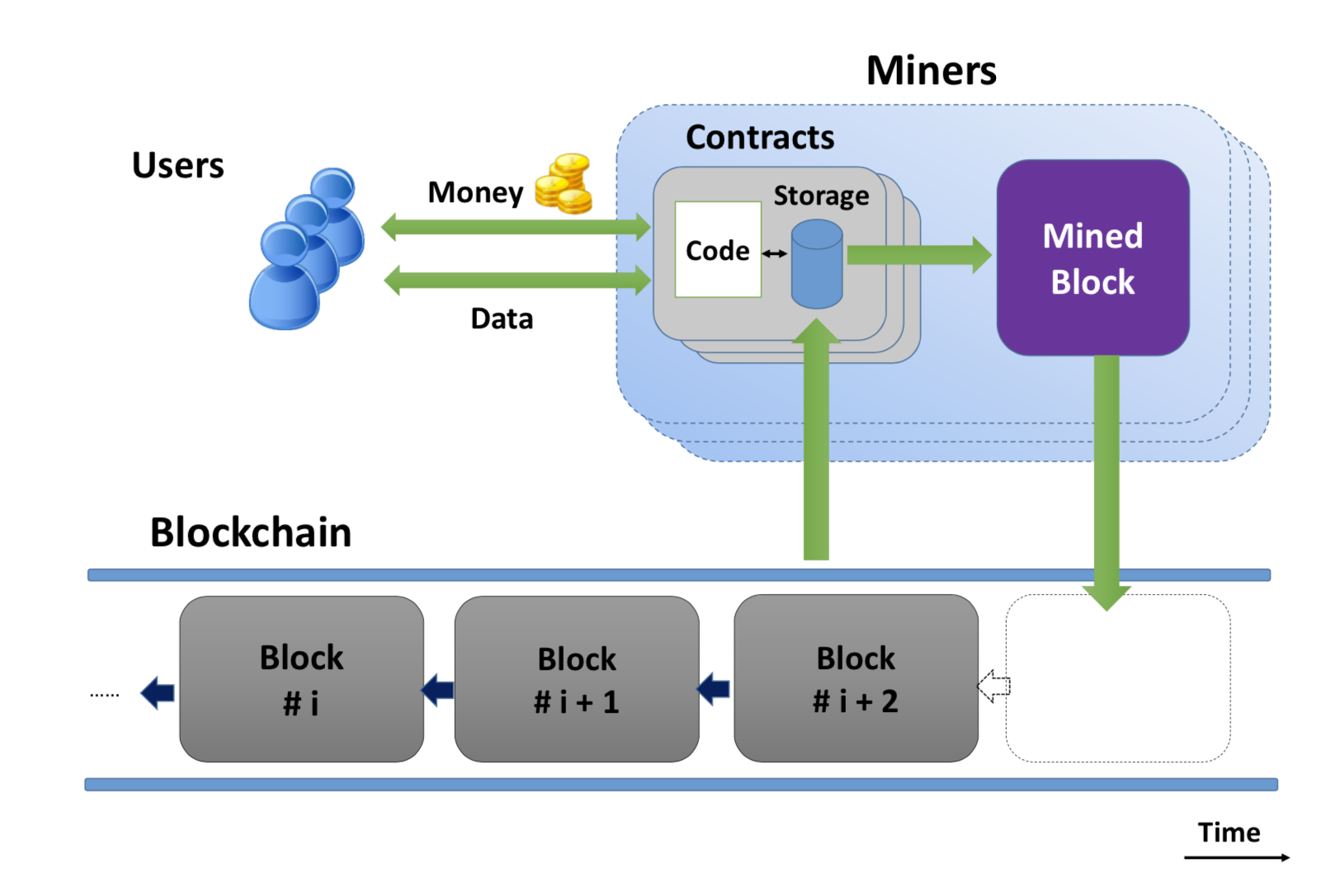
图1

图1显示了一个智能合约系统的高级架构，该系统通过比特币或以太坊等分散式加密货币实例化。 当使用加密货币的潜在共识协议是安全的时，大多数矿工（通过计算资源测量）被假定为正确执行合约的可编程逻辑。

加油站。 分散式智能合约系统的现实实例依靠天然气保护矿工免受拒绝服务的攻击（例如运行无限合同）。 天然气是一种交易费用，大体上说，与合同的运行时间成正比。 在本文中，虽然我们并没有明确地以智能合约的形式表示天然气，但我们试图将程序逻辑从合约中分离出来，尽可能优化，以保持天然气和低交易费用。 例如，我们建议的一些合同涉及在用户端执行的程序逻辑，而不会损失安全性

2.1智能合约：好与坏

分散型智能合约有许多有益的用途，包括实现各种各样的新型金融工具。 正如比特币在交易中所做的那样，在一个分散的智能合约系统中，共识系统强制自动执行合同; 没有一个实体或一小组实体可以干涉合同的执行。 由于合同是自我执行的，它们消除了对可信中介或信誉系统的需求，以降低交易风险。 与传统的加密货币比如比特币相比，分散式智能合约具有这些优势：

•互相不信任的各方之间进行公平交流，用可编程逻辑表达丰富的合同规则。 这个功能可以防止各方通过中止交换协议来作弊，同时也消除了对第三方中间人物理交会和潜在欺骗的需求。

•最大限度地减少各方之间的交互，减少不必要的监测和跟踪机会。

•通过允许经纪人提供智能合同系统以外的物理和其他事件（例如股票报价单，天气预报等）提供经认证的数据馈送（证明）来增强与外部状态的交易。这些在以太坊中处于起步阶段，但其可用性 正在不断增长并使用可信硬件的承诺来刺激他们的部署[61]。

不幸的是，除了他们所有的好处，这些财产有一个黑暗的一面，可能有助于犯罪，因为：

•公平交换使得互相不信任的犯罪团伙之间的交易成为可能，消除了当今脆弱的声誉系统和/或潜在的欺骗或执法的第三方中介机构的需要[55,39]。

•尽量减少互动，使执法部门难以监测的非法活动更为困难。 在某些情况下，对于我们提供的防盗和电话卡CSC，犯罪分子可以建立一个合同并避开，让它自主地执行而不需要进一步的交互。

•与外部国家进行丰富的交易，将可能的该系统的合作关系的范围扩大到例如物理犯罪（恐怖主义，纵火，谋杀等）。

由于分散式智能合约系统通常继承比特币的匿名性（匿名），因此它们对于犯罪活动也具有类似的保密性。 因此，广义而言，分散式智能合约系统所具备的能力有助于新的地下生态系统和社区。

2.2数字现金和犯罪

比特币和智能合约并不代表最早出现的加密货币。David Chaum [29]在一篇开创性的论文中介绍了匿名电子现金。 Naccache和冯·索姆斯指出，匿名货币会导致“完美的犯罪”，比如执法中难以追查的绑架行为[57]。这种观察促进了公平的盲签名的设计或电子现金的“托管”[26,58]，这使得可信的第三方能够链接身份和支付。 传统电子现金计划中的这种联系是可能的，用户在收回匿名现金时确定自己的身份，而不是像比特币这样的假名加密货币。

勒索软件自1989年以来一直在公众视野中出现[18]。 对勒索软件的一个主要的密码病毒[60]的“改进”就是使用比特币[44]，因为CryptoLocker勒索软件据称已经勒索了数亿美元的赎金[25]。暗杀市场对匿名数字现金使用首先在1995 - 6年的文章“暗杀政治”中提出。

对已有比如洗钱[51]，比特币盗窃[49]以及丝绸之路[31]等非法市场的比特币犯罪的广泛研究，。Meiklejohn等人 [49]注意到比特币是假名，而混合的机制旨在赋予比特币匿名性，并不以大量货币操作，一般来说，犯罪分子现在很难以匿名方式兑现。

另一方面，罗恩和沙米尔提供的证据表明，联邦调查局没有找到丝绸之路运营商“恐怖海盗罗伯茨“（罗斯·乌布里希特）的大部分比特币控股权，甚至在扣押他的笔记本电脑之后[56] M¨oser, B¨ohome, and Breuker [51]发现，他们无法成功地对所研究的三种混合品中的两种交易进行匿名化处理，这表明监管者在打击洗钱方面的主要工具“知道你的客户” 证明在过程中很难强制执行于加密货币。 越来越多NIZK的使用证明匿名加密货币的建议[20,33,50]，以及至少一个目前处于商业部署早期阶段[13]，这些建议为犯罪分子具有更强的匿名性提供了保证。

3 符号和威胁模型

我们采用Kosba等人的正式区块链模型[43] 作为背景，我们将在本节中对该模型进行高级描述。 我们使用该模型来指定我们论文中的密码协议; 这些协议包含犯罪智能合同和相应的用户方协议。

我们采用Kosba等人的正式区块链模型。[43]。 作为背景，我们将在本节中对该模型进行高级描述。 我们使用该模型来指定我们论文中的密码协议; 这些协议包含犯罪智能合同和相应的用户方协议。  
智能合约模式中的协议。如上所述，我们的模型将合同作为特殊的一方来处理，这个特殊的一方被委托来强制执行正确性而不是隐私。（实际上，合同是由网络强制执行的。）发送给合同的所有消息及其内部状态都是公开的。契约与用户和其他契约交互通过交换消息（也被称为交易）。以账户余额形式表示的资金记录在全球分类账（区块链）中。合同可以访问和更新分类帐，以实现由假名公钥代表的用户之间的资金转移。

3.1威胁模型

本文中我们采用如下的威胁模型。

•区块链：因为准确性被信任，但不是隐私。 我们假设区块链总是正确地存储数据并执行计算而且始终可用。 然而，区块链将所有内部状态公开给公众，并且不保留私人数据。

•任意恶意的合同当事人。 我们假设合同当事人是相互不信任的，他们只是为了最大限度地发挥自己的利益。 它们不但可以随意偏离规定的协议，还可以过早地退出协议

•网络对敌手的影响。 我们假设区块链和玩家之间的消息是在有限的延迟内传递的，即不是永久丢弃（玩家总是可以重新发送恶意矿工下的交易）。然而，在我们的模型中，敌手立即接收并可以任意地重新排序消息。 在现实生活中分散的加密货币中，获胜的矿工设置了消息处理的顺序。 敌手可能与某些矿工勾结或影响节点之间的信息传播。 正如我们在第5节中所显示的那样，对于钥匙盗窃合同，消息重新排序可以实现一个佣金公平的CSC必须防止的急剧的攻击。

我们采用的正式模型（在本节后面进行回顾，由Kosba等[43]完整描述）涵盖了我们威胁模型的所有上述方面。

3.2安全定义

一个CSC如果要实现佣金公平需要两个条件：

•正确定义佣金公平。 没有普遍的正式的佣金公平定义：它是特定的应用，因为它取决于罪犯（和犯罪者）的目标。因此，对于每个CSC，我们在在线完整版[42]中通过UC风格的理想功能来指定相应的委托 - 公平的定义。只是指定一个正确的理想功能本身往往是挑战！我们在第五部分和在线完整版本[42]中说明了这个挑战，其中一个天真密匙功能代表了看起来正确但实际上是失窃的钥匙盗窃合同。

•正确的协议实施。为了证明CSC是佣金公平的，我们必须证明它的（现实世界）协议模拟了相应的理想功能。我们在密码学文献中采用的标准通用可组合（UC）模拟范式[28]中对我们描述的CSC进行了证明，而不是任意恶意的合同对手以及可能的网络攻击者。我们的协议也可以抵御对手的堕落，例如企图在不向对方支付的情况下放弃。一般来说，存在中止的公平性在分布式计算的标准模型中是不可能的[32]。最近的一些作品表明正确、可用和意识到时间进程的区块链可以强化对中止方的财务公平性[23,43,17]。 特别是，当合同失效时，区块链可能导致中止方向诚实方给予存款。

3.3符号公约

我们现在解释一下写合同的一些符号约定。在线完整版[42]给出了一个热身的例子。

•货币和分类帐。我们使用分类帐[P]来表示P方在全局分类账中的平衡。为了清楚起见，以$符号开头的变量表示货币，但其他行为与普通变量相似。与以太坊的蛇形语言不同的是，在我们的正式表示法中，当一个合约从P方得到一定数额的金额时，这只是信息传递，此时没有货币转移。当合同在分类帐上执行操作时，转账只会起作用。

•假名。 各方可以使用假名来获得更好的匿名性。特别是一方可以生成任意多的公钥。在我们的记号系统中，当我们提到一个利益方时，P代表利益方的化名。 我们采用的正式区块链模型[43]提供了一个合同包装器，用于管理为合同建立认证通道所需的假名生成和消息签名。 这些细节是从主合同计划中抽象出来的。

•定时器。 时间在按照轮次进展。在每一轮的开始，合约的定时器功能将被调用。变量T编码当前时间。

•入口点和变量范围。 合同可以有多个入口点，每个入口点在接收到相应的消息类型时被调用。因此入口点的行为就像接收到消息时所调用的函数调用一样。假定所有变量都是全局范围的，但有以下例外：当入口点表示“在收到来自某一方P的消息后”时，这允许注册一个新的P方。一般来说，合同对任何一方 与他们互动。当收到来自P的消息（没有关键字“some”）时，利益方P表示一个固定的方 - 并且一个格式良好的合同已经定义了P.

这个符号系统[43]不仅是为了方便而设计的，而且还被赋予了与通用可组合性框架兼容的精确的，正式的含义[28]。 我们引用读者[43]的正式建模细节。虽然我们在线完整版[42]的证据依赖于这种支持形式主义，但没有它，主体仍然是可以理解的。

4 CSC秘密泄露

作为智能合约权力的第一个例子，我们展示了一个现有类型的比特币犯罪合同如何能够作为一个智能合同变得更强大和功能更强大，并可以在以太坊实际执行。

在支付过程中泄露秘密是由比特币引发的的非法行为之一，即公开披露。 最近创建的网站Darkleaks [3]（一种受到资助的维基解密）是一个分散的市场，为各种各样的秘密进行公众泄密，包括“好莱坞电影，商业秘密，政府机密，专有源代码，工业设计如药品或国防，[等]“。

直觉上来说，我们在这个定义中定义佣金公平性意味着承包商C在规定的时间限制内完全泄露了一个秘密（正式的定义请参见在线完整版[42]）。正如我们所示，Darkleaks强调比特币不能支持佣金公平。我们展示了一个CSC实际上如何以高概率实现佣金公平的范例。

4.1Darkleaks

在Darkleaks系统中，承包商C希望出售一段内容M，将其分割成n个段{mi} n i = 1的序列。 在一个时间（块高）Topen预先指定的C，一个随机选择的K段的子集Ω⊂[n]被公开作为样本，以吸引捐助者/购买者，而这将有助于购买M公共泄漏。当C确定捐助者集体支付了足够的价格时，C解密剩余的部分以供公开发布。参数triple（n，k，Topen）由C设置（其中n = 100，k = 20是推荐的默认值）。

为了确保公平交换M付款而不需要双方直接交流，Darkleaks在比特币脚本语言之上实现了一个（智能的）协议。 主要思想是，对于Ω中没有显示为M样本的M的给定部分，捐助者使用公钥pki向比特币账户ai支付。 段mi在密钥κ= H（pki）（其中H = SHA-256）下被加密。为了从账户ai中获得奖励，C被比特币交易协议强制披露pki; 从而自动消费的行为使社区解密。

我们在网上完整版[42]给进一步的细节。

缺点和漏洞。 Darkleaks协议有三个主要的缺陷/漏洞，这些缺陷/漏洞似乎源于比特币脚本语言的基本功能限制，在构建合同时没有双方之间的直接通信。前两者削弱了佣金公平，而第三种则限制了职能。

1.延迟公布：C可以避免购买者/捐助者的付款，并且在M失去价值之后释放M的未公布部分。例如，C可以隐藏上线前电影的一部分，或者发布前的工业设计等等。

2.选择性隐瞒：C可以选择放弃支付选定的细分市场，不公开。例如，C可能泄漏并收集所有泄漏电影的付款，但最后几分钟（很可能不会出现在样本Ω中）显着降低了泄漏电影的价值。

3.只有公开泄漏：Darkleaks只能公开泄露秘密。它不允许公平交换私人泄漏，即为了交换在购买者P的公共密钥下加密的秘密M的支付。

另外，Darkleaks有一个基本的协议的缺陷：

4.奖励盗窃：在Darkleaks协议中，对应于pki的比特币私钥是从mi派生的; 特别是ski = SHA-256（mi）。因此，M的来源（例如，泄漏电影的受害拥有者）可以获窃取得由C接收的奖励。（另外，当C要求奖励时，接收交易的恶意节点可以解密mi，计算= SHA-256（mi），并且可能通过竞争性交易来侵蚀网络来窃取奖励[36]）。

最后一个问题很容易通过伪随机地或随机地生成段加密密钥的集合{κi} n i = 1来解决，这在我们在CSC设计中会做的。

备注：在任何货物由随机样本表示的协议中，不仅仅是Darkleaks，C可以向M中插入少量无价值或重复的分段。以不可忽略的概率，这些不会导致样本Ω看起来无效，因此必然只能提供对M的全局有效性的弱保证。k和n越大，这种攻击的风险就越小。 对这类人为验证的证明和/或自动化的方法进行形式分析是一个超出本文范围的有趣问题，但是在评估这种CSC的端到端安全性方面是非常重要的。

4.2一个通用的功用漏洞CSC

我们现在提出一个智能合约，用黑匣子密码原语来实现公开泄密。（我们后来提出了有效的实现）。这个合同通过在预定的时间内强制披露M，来克服Darkleaks协议（延迟发布）的限制1或者立即退还买方的资金。它通过确保M以全有或全无方式显示来解决限制2（选择性扣留）。（我们稍后解释如何实现私人泄漏并克服限制3。）

再次，我们考虑设置C的目标是在揭示样本段M \*之后卖给M公开发布。

非正式的协议描述 ：非正式地，协议涉及以下步骤：

•创建合同：卖家C用随机生成的主密钥msk的加密来初始化智能合约。主密钥用于为段{mi} n i = 1生成（对称）加密密钥。C向合约提供密码承诺c0：= Enc（pk，msk，r0）msk。 （为了满足我们安全性证明的狭义技术要求，这个承诺是在可信任设置步骤期间创建的公钥pk下的随机性r0的加密。）主密钥msk可以用来解密M的所有泄露段。

•上传加密的数据：对于每个i∈[n]，C生成加密密钥κi：= PRF（msk，i），并且将第i个段加密为cti =encκi[mi]。C将所有加密段{cti}i∈[n]发送到合同（或者为了高效，提供与存储提供者（例如，对等网络）一起存储的副本的散列）。有兴趣的购买者/捐赠者可以下载M的片段，但是不能解密它们。

•**挑战**：合约生成一个随机挑战集Ω⊂[n]，现在在以太坊的实践中基于最近块的散列。另一个未来的可能性是一些众所周知的随机性来源，例如NIST随机性信标[9]，也许通过认证的数据馈送中继。

•**收集捐款**： 在捐赠期间，潜在购买者/捐赠者可以使用显示的密钥{κi}i∈Ω来解密相应的片段。如果他们喜欢解密的细分市场，他们可以把钱捐给合同作为泄漏的贡献。

•**接受**。 如果已经收集了足够的钱，C就解除合同的msk（发送密文和msk的随机性）。如果合同成功地证明了解体，那么所有的捐款都支付给C.合同因此强制公平地交换msk。（如果合同在没有发布msk的情况下到期，所有捐款都将被退还。）

合同。 我们提出的用于实施这个公开泄漏协议的CSC PublicLeaks在图2中给出。对应的用户方在上面非正式地解释（并且可以从合同推导出）。

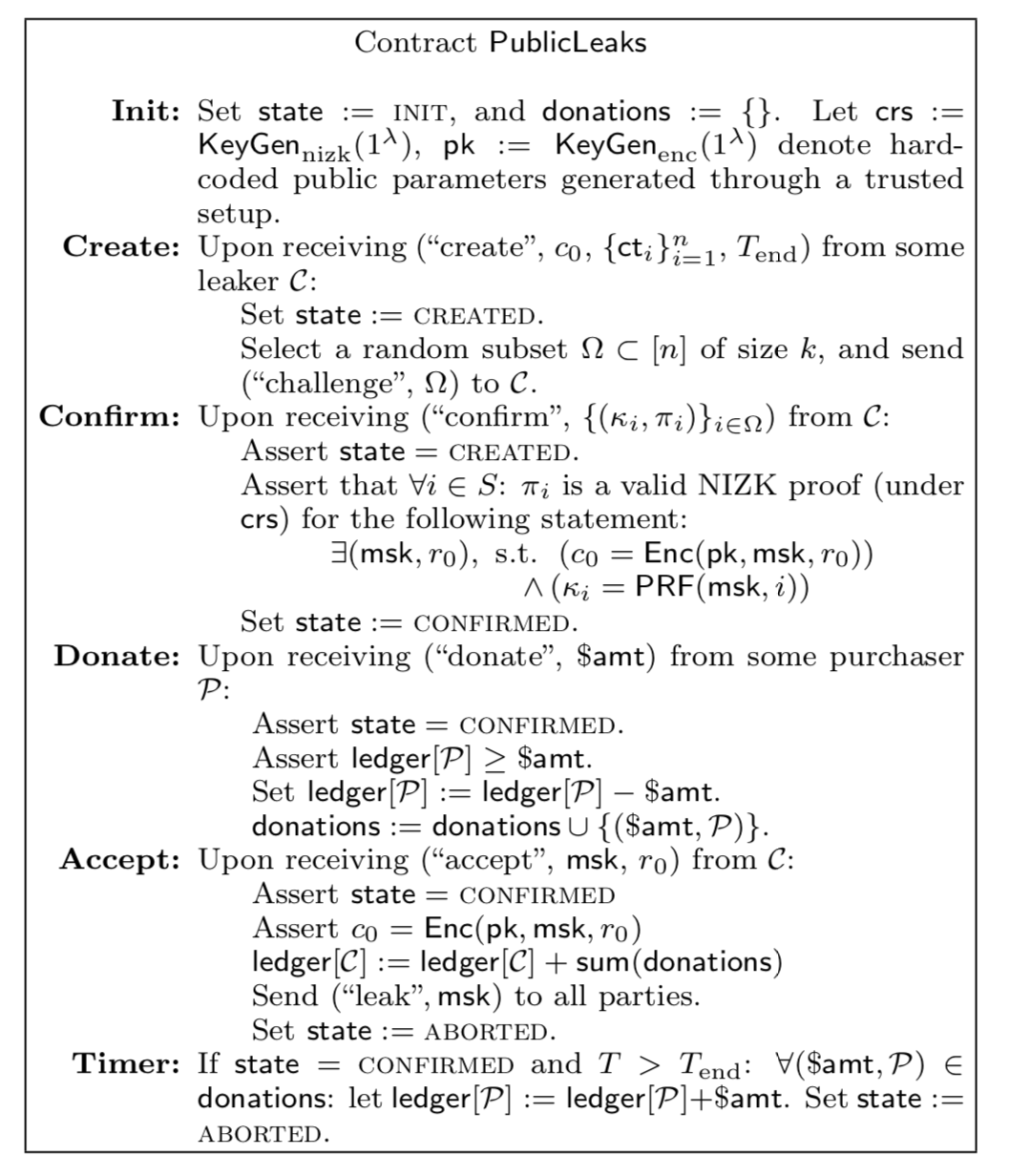


图2

4.3佣金公平：正式定义和证据

在线完整版本[42]中，我们给出了一个正式的公开泄漏的公平性定义的定义（上面非正式解释），作为一个理想的功能。 我们还证明，PublicLeaks认识到这一功能，假设所有显示的细分都是有效的 - 通过在PublicLeaks中对M进行随机抽样，强制执行的属性的概率很高（但不是压倒性的）。

4.4优化和以太坊实施

正式指定的合同PublicLeaks以黑匣子方式使用通用密码原语。我们现在给出一个实用的，优化的版本，依靠随机预言模型（ROM），消除了可信任的设置，同时也获得了更好的效率，并易于与以太坊集成[59]。

一个实际的优化：在合同创建过程中，C为i∈[n]选择随机κi$←{0,1}λ，计算

c0：= {H（κ1,1），...，H（κn，n）}。

主密钥简单地是msk：= {κ1，...，κn}，即，该组哈希预图像。和PublicLeaks一样，每个分段mi仍然会被加密为cti：= enck [mi]。（由于技术原因，为了在安全性证明中实现可模拟性，这里enck [mi] = mi⊕[H（κi，1，“enc”）|| H（κi，2，“enc”）...， H（κi，z，“enc”）]适合于较大的z值。）

C向智能合约提交c0。当用集合Ω挑战时，C向合约显示{κi}i∈Ω，然后通过散列和与c0比较来验证它的正确性。接受捐款，C揭示了整个msk。

这个优化的方案比我们通用的黑盒结构PublicLeaks渐近效率低，因为主密钥在线段n的数量上是线性的。但是对于典型的，实际的文档集合大小（例如n = 100，DarkLeaks推荐的），它更有效率。

以以太坊为基础的实施：为了证明使用现有技术实施泄漏合同的可行性，我们使用“大蛇”合约语言[10]，实施了以太坊[59]上的PublicLeaks合同版本。我们的在线完整版[42]详细说明了完整的实现。

我们实现的版本依赖于上述的实际优化。作为一个技术性的问题，以太坊目前还没有出现支持定时器激活的功能，所以我们实施了定时器，使购买者/捐助者明确提款，而不是接收自动退款。

这个公开泄漏以太坊合同是非常有效的，因为它不需要昂贵的密码操作。它主要依靠散列（SHA3-256）来生成随机数并验证散列约定。存储条目（加密密钥需要的）和散列操作的总数是O（n），同样，Darkleaks推荐n = 100（实际上散列函数调用需要几微秒，例如3.92μscs被测量 在核心i7处理器上。）

4.5延伸：私人泄露

如上所述，Darkleaks的缺点是它无法支持私人泄漏，其中C只向买方P出售秘密。在线完整版[42]中，我们将展示PublicLeaks如何为此目的而进行修改。基本思想是C不直接揭示msk，而是在msk上为买主P提供一个密文ct = encpk P [msk]，同时证明ct是正确形成的。我们描述一个黑箱变体，其安全性可以通过与PublicLeaks基本相同的方式得到证实。我们还描述了一个实用的变体，它将Chaum和Pedersen [30]的一个可验证的随机函数（VRF）（用于生成{κi} ni = 1）与Camensich和Shoup的一个可验证加密（VE） 证明ct的正确性）。这个变体可以在今天使用测试版支持以太坊中的大数运算。

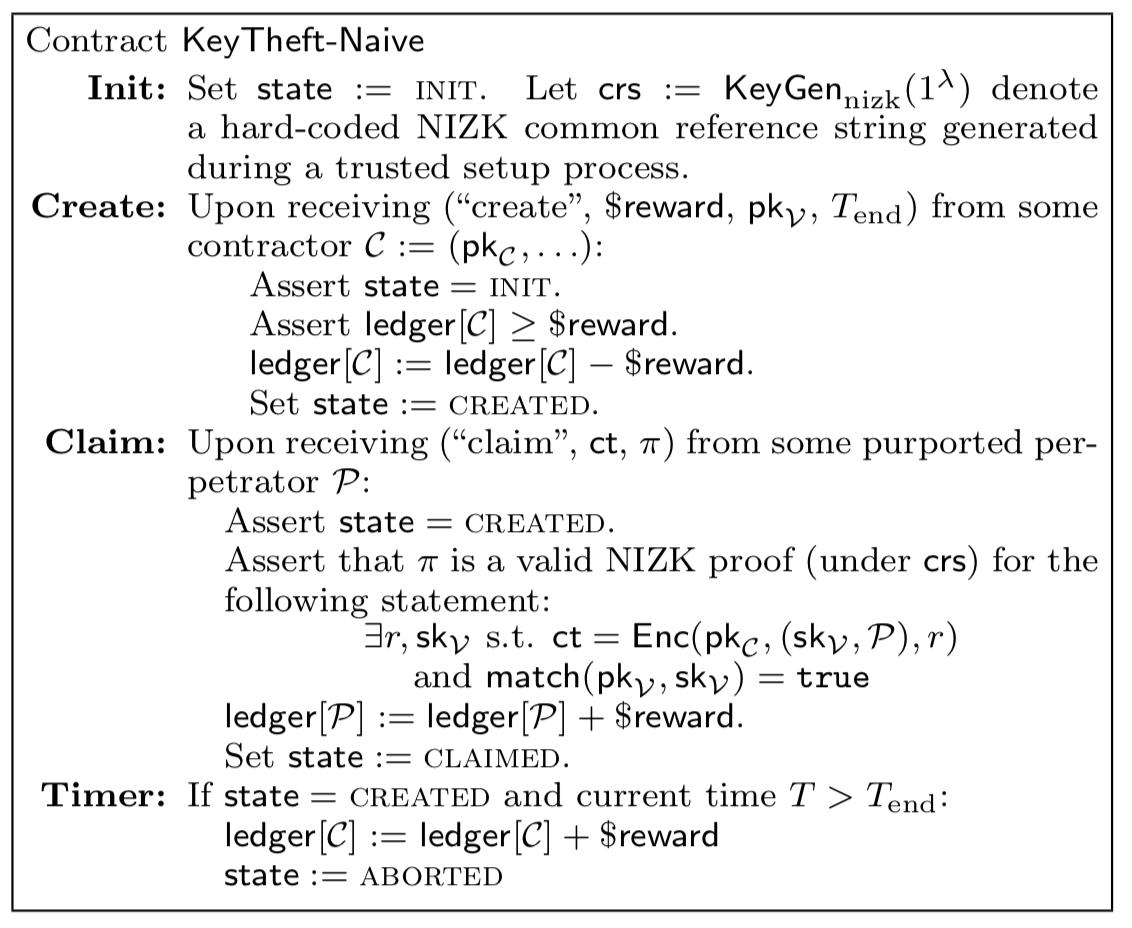


图3

5 密匙妥协的CSC

在引言中的例子1b描述了一个CSC，奖励犯罪者P向受害者V的被盗密钥skV（在这种情况下是具有公共密钥pkV的证书颁发机构（CA））递送C.。回想一下C生成一个私钥/公钥加密对（skC，pkC）。合同接受P一对（ct，π）的索赔。如果π是证明ct = encpkC [skV]且skV是与pkV相对应的私钥，那么它向P发送奖励$奖励。

直觉上来说，如果一个钥匙盗窃合同是佣金公平的，如果它奖励为犯罪者P提供一个私钥，那么：（1）P负责窃取;（2）在相当长的一段时间内有效。（有关正式的定义，请参见在线完整版[42]。）

这种合同形式可以用来请求盗用任何类型的私钥，例如CA的签名密钥，SSL / TLS证书的私钥，PGP私钥等（类似的合同可能会引起滥用， 但不能完全泄露私钥，例如伪造的证书。）

图3显示了我们用于智能合约的例子1b的合约。我们让crs在这里表示一个NIZK方案的公共参考字符串，匹配（pkV，skV）表示一个算法，该算法验证skV是否是目标公钥密码系统中某些公钥pkV的相应私钥。

图3显示了我们用于智能合约的例子1b的合约。我们让crs在这里表示一个NIZK方案的公共参考字符串，匹配（pkV，skV）表示一个算法，该算法验证skV是否是目标公钥密码系统中某些公钥pkV的相应私钥。

5.1KeyTheft-Naive缺陷

KeyTheft-Naive合约由于两个缺点而未能达成委托公平。

撤销和索赔攻击：CAV可以撤销密钥skV，然后自己提交付款密钥。那么CA不但否定了合同的价值，而且实际上从中受益！ 这种撤销和索赔攻击表明，KeyTheft-Naive在确保提供可用私钥skV方面不是佣金公平的。

冲击袭击：另一个攻击是一个冲击攻击。正如第3节所述，攻击者可以任意重新排序消息 - 反映在一个加密货币中对网络层的可能的攻击。（参见正式的区块链模型[43]）。因此，从犯罪者P得到有效的要求，腐败的C可以解密并学习skV，在有效的声明之前构建另一个看似有效的自己的要求。

5.2KeyTheft-Naïve的固定缺陷

我们现在展示如何修改KeyTheft-Naive来防止上述两种攻击并实现佣金公平。

阻止撤销和索赔攻击：在对KeyTheft-Naive的撤销和索赔攻击中，V抢先撤销其公钥pkV，并用新的pk0V取而代之。如上所述，受害者可以扮演犯罪者P的角色，向合同提交skV并要求奖励。结果是C向V付了$奖励并获得一个陈旧的密钥。

我们通过在合同中添加一个称为奖励截断的特征来解决这个问题，合同接受撤销证据Πrevoke。

这个证据可以是在线证书状态协议（OCSP）的回应，表明pkV不再有效，在合同创建时未知的V的新证书（因此不存储在合同中）或证书撤销清单 （CRL）包含与pkV证书。

C可以提交Πrevoke，但为了尽量减少C的交互，KeyTheft可以向第三方提交者提供$ smallreward的奖励。奖励可能很小，因为Πrevoke对普通用户来说很容易获得。

合约然后根据关键skV保持有效的时间间隔提供奖励。让Tclaim表示提供密钥skV的时间，Tend是合约到期的时间（不能超过包含目标密钥的证书的到期时间）。设Trevoke为Πrevoke出现的时间（Trevoke =∞，如果在Tend之前没有撤销）。然后，合同赋予P奖励f（奖励，t），其中t = min（Tend，Trevoke）-Tclaim。

我们不在这里探索f的选择。然而，我们注意到，鉴于CA密钥skV可以用来伪造证书以便快速用于例如恶意软件或伪造的软件更新，其很大的价值可以在我们用δ表示的短时间间隔内实现。（一般倾向于先期实现利用价值[24]）。适当的奖励函数选择应该是前加载的，并迅速衰减。这个属性是一个自然，简单的选择。

阻止冲击袭击：为了阻止匆匆的袭击，我们将这一说法分为两个阶段。在第一阶段，P通过提交真实索赔信息的承诺表达了要求索赔的意图。P然后等待下一轮打开承诺，并显示索赔消息。（由于证明中的技术细节，承诺必须是自适应安全的;在证明中，模拟器必须能够模拟承诺而不知道承诺的字符串，并且稍后能够声明对任何字符串的承诺 s）。在现实生活中分散的加密货币中，P可能会在开放承诺之前等待多个块间隔，以更高的置信度区块链不分叉。在我们的形式主义中，一轮可以对应一个或多个块区间。

图4给出了一个关键的盗窃合同KeyTheft，它可以阻止撤销和匆忙的攻击。

5.3目标和状态的暴露

KeyTheft-Naive的不良财产其目标/受害者和国家是公开可见的。V可以从而了解它是否是KeyTheft-Naive的目标。V还观察到成功的主张，即skV是否被盗，因此可以采取明智的防御行动。例如，由于密钥撤销是昂贵和耗时的，因此V可能会等到发生成功的请求，然后才会执行撤销和声明攻击。

为了限制目标和状态的曝光，我们标记两个可能的增强KeyTheft。第一个是多目标合同，要求对一个多重受害者中的任何一个进行钥匙盗窃。其次是我们所称的封面索赔，虚假索赔，隐瞒任何真正的索赔。如图4所示，我们实现的KeyTheft是一个多目标合约，因为这种技术提供了部分目标和部分状态隐藏。

多目标合约：多目标合同征求m个潜在受害者V1，V2，...，Vm中的任何一个的私钥。有很多设置，其中不同的受害者的私钥具有相似的价值。例如，多目标合约KeyTheft可以为任何能够发行由例如Internet Explorer（其中有超过650 [37]）信任的SSL / TLS证书的CA的私钥skV提供回报。

这里的挑战是合同状态是公开的，因此合同必须能够在不知道提供哪个密钥的情况下验证有效索赔（私钥）skVi的证明，即不需要知道i。 我们的实现表明，构建zk-SNARKs这样的证明是实用的。 （承包商C本身可以通过解密skVi，生成pkVi并识别相应的受害者来轻松掌握信息。）

覆盖索赔。 由于合同的状态是公开的，受害者V知道是否已经向KeyTheft-Naive提交了成功的索赔。在单一目标合同的情况下，这尤其成问题。

不使用ct发送NIZK证明π，而是延迟提交π（以及奖励的支付）直到Tend。（也就是说，索赔作为输入（“索赔”，ct）。）这种方法隐藏ct的有效性。请注意，即使没有π，C仍然可以使用ct。

支持这种隐瞒的合同也可以支持我们称之为保障索赔的想法。保险索赔是表格（“索赔”，ct）的无效索赔，即其中ct不是skV的有效加密的索赔。C可能会提交封面索赔来掩盖合同的真实状态。因此，在创建之后，C不需要与合同交互，合同可以将时间上的小回报倾向于提交保险索赔的第三方。我们不在我们的KeyTheft版本中实施封面声明，也不将其包括在图4中。

5.4佣金公平：正式定义和证据

我们根据在线完整版[42]的理想功能定义了关键盗窃的佣金公平性[42]，并为KeyTheft提供了正式的安全证明。

5.5履行

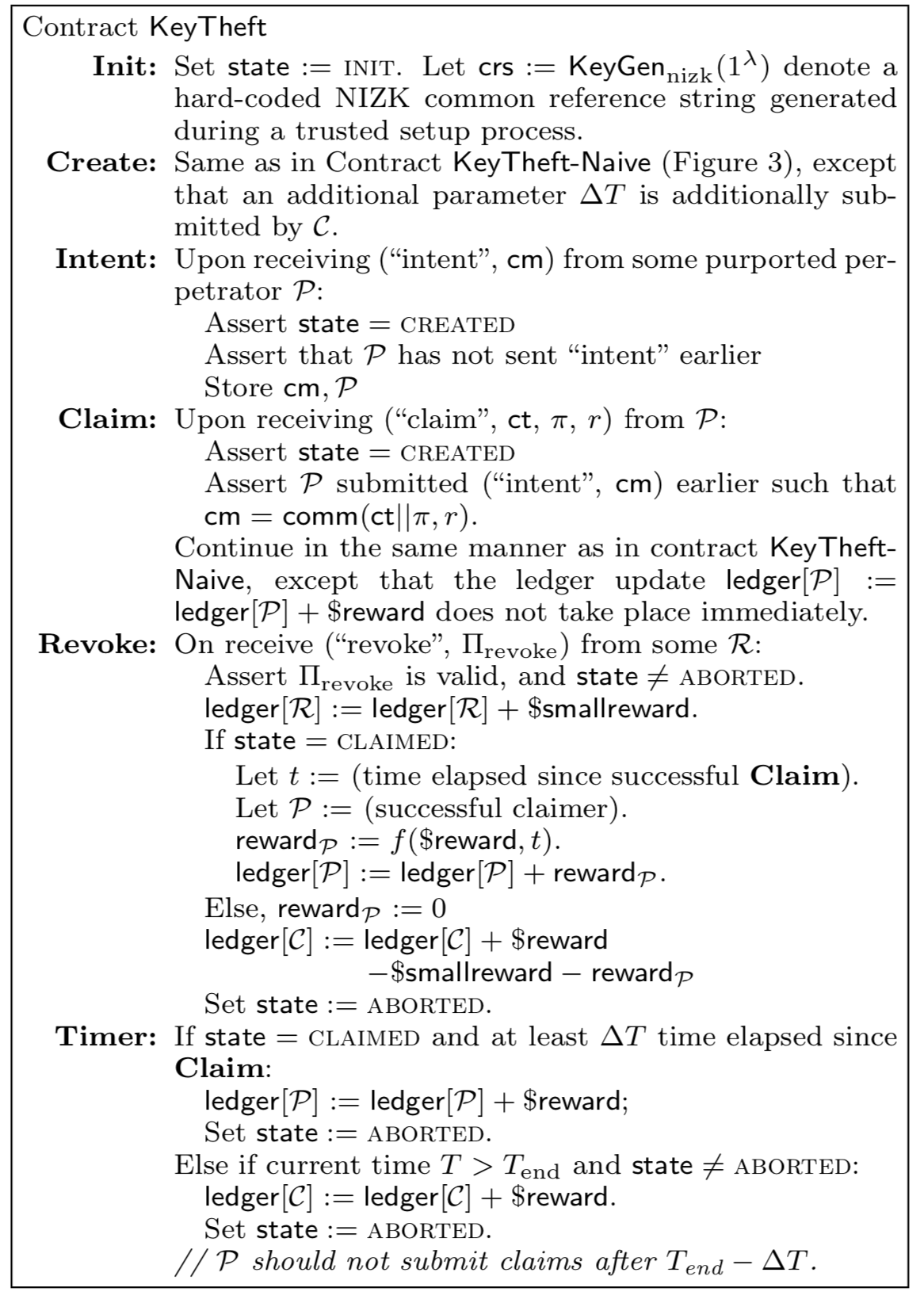
我们依靠zk-SNARKs来有效地实现上述协议。 zk-SNARKs是知识的零知识证明，简洁且非常有效地进行验证。zkSNARK的安全性比UCstyle仿真证明中所需要的要弱。因此，我们使用Hawk工作[43]中描述的一种通用转换来提升安全性，使得零知识证明确保模拟可提取的健全性。

图4

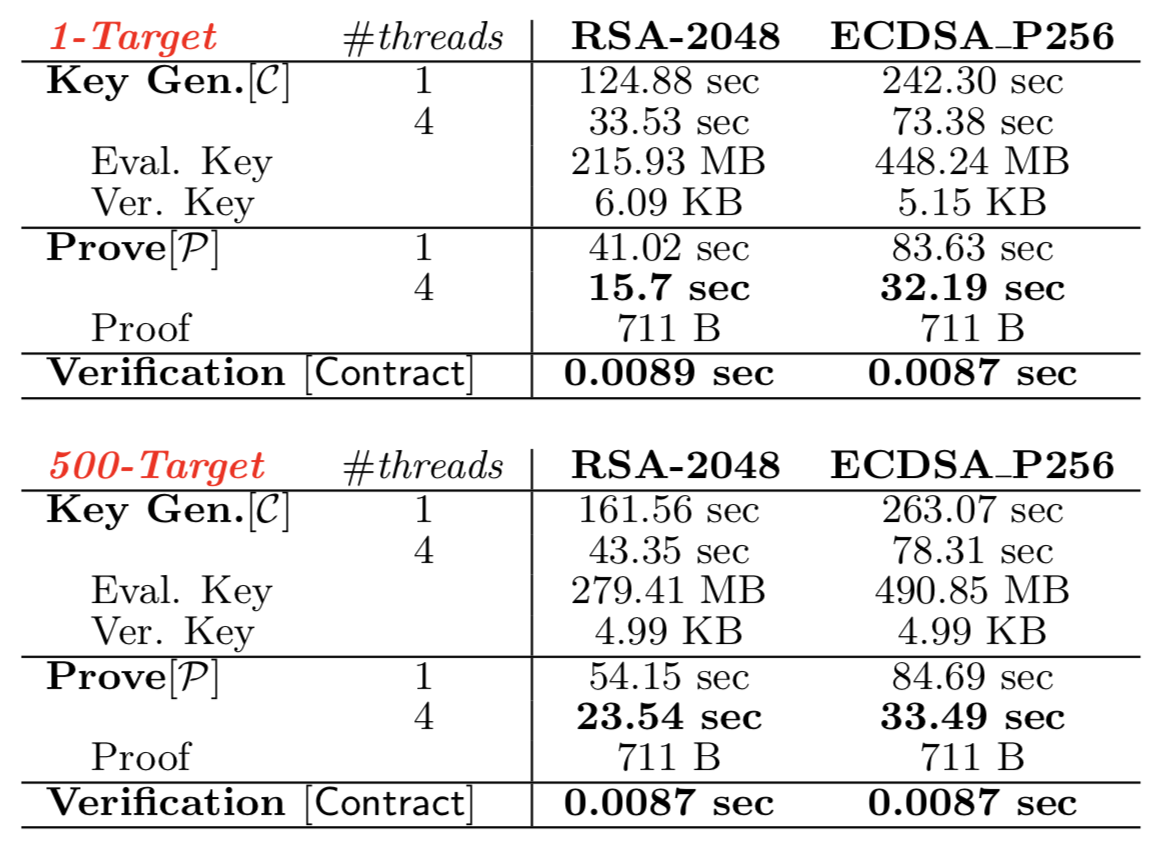
（简而言之，需要一次密钥生成阶段来生成两个密钥：一个公开的评估密钥和一个公开的验证密钥为了证明某个NP声明，一个不可信的证明者使用评估密钥来计算一个简洁的证明;验证者可以使用公开的验证密钥来验证证明，我们的案例中的验证者是合同）。在我们的实现中，我们假设密钥生成是由一个可信任的组织来执行的。否则证明人可以出示虚假陈述的有效证据。为了尽量减少密钥生成阶段的信任，可以使用安全的多方计算技术[21]。zk-SNARK电路索赔。 为了估计索赔所需的证明计算和验证费用，我们实施了上述窃取RSA-2048和ECDSA P256密钥的协议，这些密钥在当前的SSL / TLS证书中被广泛使用。该电路具有两个主要子电路：密钥检查电路和加密电路。使用RSAES-OAEP [41]和2048位密钥来实现加密电路。依靠编译器来实现这些算法的高级实现可能会产生昂贵的zk-SNARK证明计算电路。相反，我们建立了定制的电路发生器，产生更多的有效电路。然后，我们使用最先进的zk-SNARK库[22]来获得评估结果。表1显示了单目标和多目标合同的电路评估结果。实验在Amazon EC2 r3.2xlarge实例上进行，配备61GB内存和2.5 GHz处理器。

表1

结果产生了两个有趣的观察结果：i）一旦犯罪者获得TLS公钥的秘密密钥，计算zk-SNARK证明将需要不到两分钟的时间，单个或多个目标的成本低于1美元[4]合同; ii）在证明者方面使用具有500个按键的多目标合同引入的开销在最坏情况下仅为13秒。与此同时，合同的检验费用仍然与单一目标的情况相同。这是通过使用有效的Merkle树电路来实现的，该电路证明了在目标密钥集中被破坏的公钥的成员资格，同时使用与单目标电路相同**的组件。验证**撤销的证书。 上述合同中的奖励功能依赖于证书吊销时间，因此合同需要能够处理证书吊销证明（例如CRL和OCSP响应）的模块，并验证CA上的数字签名。例如，我们测量了2016年2月15日的openssl verify -crl\_check命令的运行时间，测试了[12]的撤销的证书和最后更新为[8]的CRL的大小为143KB的运行时间。平均而言，在2.3 GHz i7处理器上执行的验证时间约为0.016秒。签名算法是带有RSA加密的SHA-256，带有2048位密钥。由于OCSP响应可能比CRL小，因此OCSP的验证时间可能会更少。

多目标合同的情况。 验证单目标合同的撤销证明很简单：合同可以确定撤销证明是否与目标密钥相对应。但在多目标合同中，合同不知道哪个目标关键字对应于提交的关键盗窃证据。因此，需要证明撤销对应于被盗密钥，并且必须由C提交。

我们构建了一个zk-SNARK电路，通过这个电路C可以证明犯罪者提交的密文与被盗用的目标密钥之间的连接。为了提高效率，我们通过强制P在权利要求中应用加密之前强制P将受损公钥的秘密索引附加到秘密密钥来消除对撤销中的密钥检查子电路的需要。

表2中的评估说明了接受证明的合同所进行的验证的效率，以及构建证明的C的实用性。与权利要求的情况相反，该电路的一次性密钥生成必须独立于C完成，因此C不能欺骗合同。我们注意到我们构建的撤销电路对于目标密钥的密码系统是不变的。

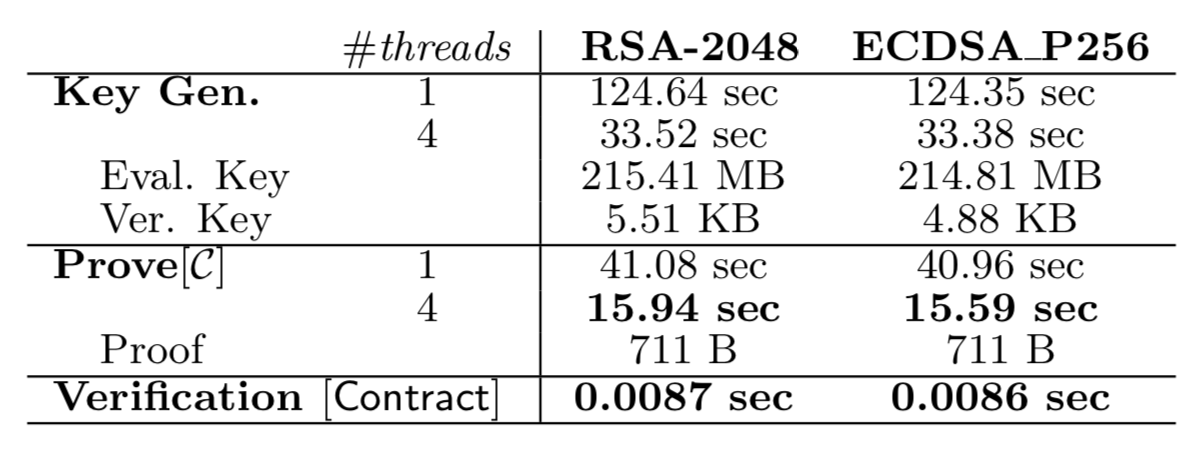


表2

6 呼叫卡犯罪

如上所述，分散式智能合约系统（如以太坊）提供支持服务，提供认证的数据馈送，数字签名的新闻证据，有关物理世界的事实等等。虽然还处于起步阶段，但这种强大的功能对于许多应用 的智能合约，并且将扩大CSCs的范围，从而广泛地涵盖物理世界中的事件，如下例所示：

例2（刺杀CSC） 承包商C发布合同刺杀参议员X的暗杀行为。合同奖励此罪行的犯罪人P.

合同暗杀者从犯罪者P输入一，提前指定暗杀的细节（日期，时间和地点）。为了认领）。为了认领这个奖励，P在暗杀之后解除vcc。为了验证P的说法，刺客在当前事件上搜索一个经过验证的数据馈送，以确认参议员X的暗杀事件，确认其细节符合vcc。

这个例子还说明了我们称之为呼叫卡的用法，用cc表示。 电话卡是被执行的犯罪的不可预知的特征（例如，在例子2中，一天，一次，一次）。呼叫卡以及认证的数据馈送可以支持各种CSC的通用框架。

基于电话卡的CSC的通用结构如下。P提前向合同提供一个对cc卡的承诺vcc。在犯罪后，P证明cc对应于vcc（例如，取消vcc）。合同是指一些可靠的和认证的数据馈送，以验证：（1）犯罪已经发生（2）电话卡cc符合犯罪。如果两个条件都满足，合同就付给P奖励.

直观地说，我们将佣金公平定义为：如果P负责执行委托犯罪，P就会得到奖励。（正式定义在线完整版[42]）

更详细地说，令CC是一组可能的呼叫卡，cc∈CC表示一个呼叫卡。如上所述，预计在智能合约系统（如以太坊）周围将出现经过身份验证的数据馈送生态系统。我们将数据馈送建模为源S的一个序列对，其中（s（t），σ（t））是时间t的发射。这里s（t）∈{0,1} \*是在时间t发布的一段数据，而σ（t）是相应的数字签名; S具有用于签名/验证签名的相关私钥/公钥对（skS，pkS）。

请注意，一旦被创建，呼叫卡合同就不需要C的进一步交互，使执法机构难以使用后续的网络传输追踪C。

6.1事例：网站污损合同

作为一个例子，我们指定一个简单的CSC SiteDeface用于网站的修改。承包商C指定一个网站的网址被黑客攻击，并声明要显示（例如，stmt =“匿名，我们是军团，我们不会原谅...”和url = whitehouse.gov.）

我们假设一个验证网站内容的数据馈送，即s（t）=（w，url，t），其中w是网页内容的表示，t是时间戳，在合约时间内表示为简单。（为了高效，w可以是对页面内容的散列和指针）这种馈送可以采取例如被黑网站（例如，zone-h.com）的档案的数字签名版本的形式。函数SigVer表示签名验证操作。

作为例子参数化，我们可以让CC = {0,1} 256，即，cc是一个256位的字符串。行为人P简单地选择呼叫卡cc $←{0,1} 256和承诺vcc：= commit（cc，P;ρ），其中commit表示承诺方案，并且ρ∈{0,1} 256是随机字符串（在实践中，考虑到支持SHA-256，HMACSHA256是在以太坊易于实现的合适选择）。P通过揭示所有提交的参数来被解除。

CSC（ 网）站污损如（图5）所示。为了清楚起见，我们使用的示例被简化了。我们假设在这个例子中，发布的网页将只包含电话卡和声明，但是可以在发布的网页中支持任意丰富的内容。

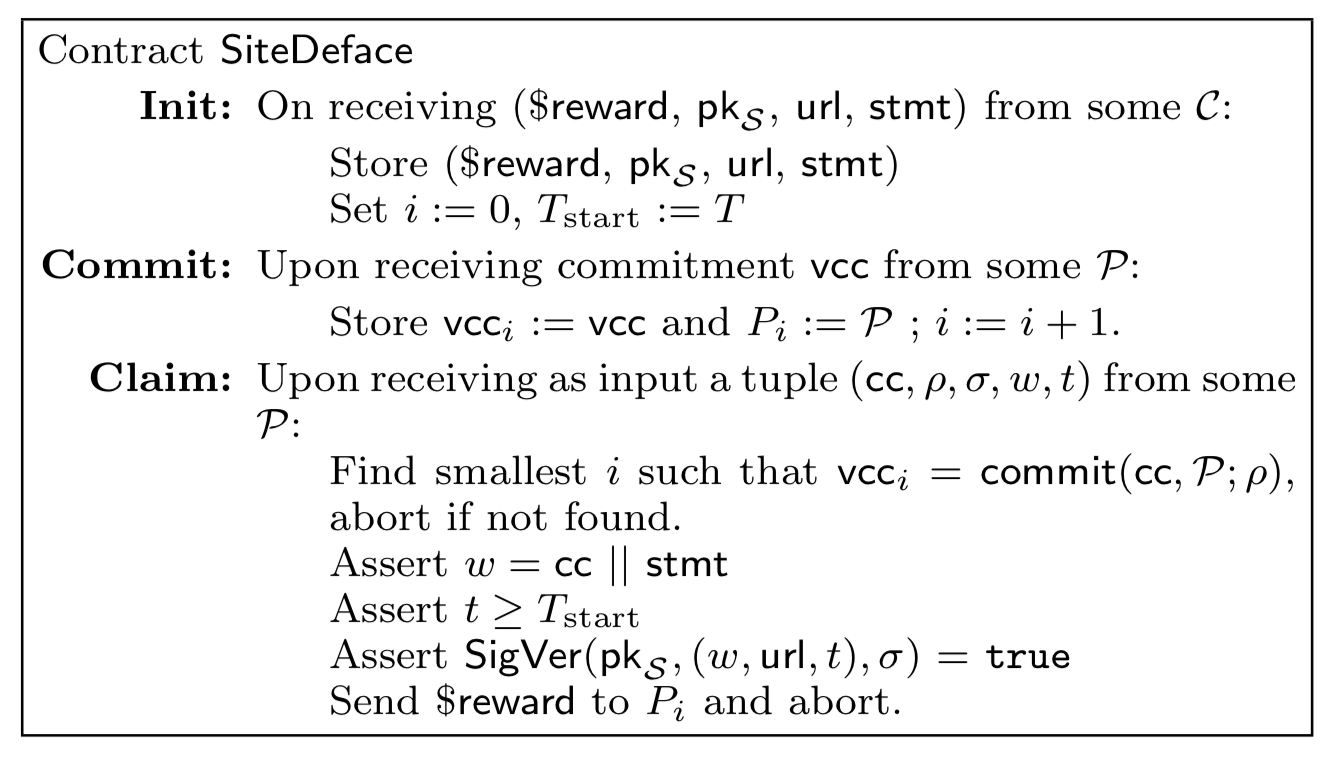


图5

备注。网站污损可以通过将P生成为数字签名来实现。然而，我们的实施也容纳短的低熵呼叫卡cc，这对于一般的呼叫卡CSC是重要的。见在线完整版[42]。

执行： 给定一个认证的数据馈送，实现网站污损将是直接和有效的。主要的开销在于Claim模块，在这个模块中，契约计算两个哈希值并验证检索到的网站数据上的提要签名。如第4节所述，哈希函数调用可以在非常短的时间（4μsec）内计算，而检查签名会更加昂贵。例如，如果检索到的内容是100KB，则合同仅需要大约10msec来验证RSA-2048签名。

6.2佣金公平：正式定义

我们在线完整版[42]给出了普通呼叫卡CSC的佣金公平的正式定义。

我们没有提供安全证明，因为这需要物理世界系统的建模，而这在本文范围之外。

6.3其他的呼叫卡犯罪/数据提供腐化

承包商C可以使用类似于网站污损的CSC来招揽许多其他罪行，例如暗杀，殴打，破坏，劫持，绑架，拒绝服务攻击和恐怖袭击。犯罪者P必须能够指定由经认证的数据馈送可靠报告的电话卡。

然而，对于基于主叫方的CSC的自然对策是针对敌手，即试图中立CSC的一方，以破坏认证的数据馈送或数据源为手段提供无效数据。在涉及高度危险的CSC的情况下，这样的方法可能是可行的。例如，如果有一个资金充足的CSC要求暗杀公职人员，可能会说服数据提供者或新闻来源发布虚假的目标死亡报告。

相反，如果C关心在一个来源中抑制信息，它当然可以创建一个引用多个来源的CSC，例如多个新闻提要。只有当这些事件的某个部分（例如大多数）被报告时，该CSC才可能会将事件视为真实的。 这种来源的多样化会使腐败变得更具挑战性。（对于良性合同来说，为数据腐败提供弹性也是一个好主意）

我们在线完整版讨论围绕着电话卡CSC构建的一些常见问题[42]。

7 对应方法

我们工作的主要目标是强调针对新兴的智能合约系统（如以太坊）开发针对CSC的对策的重要性。我们将简要地在这里讨论这个挑战。

诸如列入“污染”硬币/交易的黑名单 - 已知犯罪用途的黑客已经提出了诸如比特币这样的加密货币。第二部分提到的一个积极主动的选择是早期（集中式）电子现金系统中的身份托管思路，有时被称为“基于受托人的追踪”[26,58]。受托人追查计划允许可信任方（“受托人”）或此类当事人的法定人数追查本应保持匿名的货币交易。然而，在分散的加密货币中，用户不会向当局登记身份，许多人反对这样做。用户可以自愿注册，只选择只接受他们认为适当注册的货币。尽管如此，隐藏硬币的想法一直没有得到加密货币社区的接受，因为它破坏了基本的类似现金的可替代性[14,48]，而基于受托人的追踪也会有类似的缺点。还不清楚哪些实体应该被授予执行黑名单或注册用户的权限。

然而，我们观察到，加密货币的大多数用户监视和/或抑制犯罪活动在经济上是有利的，这可能降低接受度并因此降低市场价值。这个观察激发了例如区块链联盟[15]的创建，其使命是打击区块链上的犯罪活动。类似的，像以太坊这样的加密货币的核心开发者已经表明，类似于对仇恨言论的审查[54]，过滤区块链内容是可取的。

将比特币交易识别为犯罪行为是具有挑战性的，因为交易本身没有关于支付上下文的信息相比之下，如果确定为CSC（例如暗杀合同），则CSC是自我控制的对象。扭转CSC二进制文件可能是一个具有挑战性的问题，但我们注意到，为了让CSC发挥作用，就像我们上面的例子一样，他们的部署者必须做广告，提请注意合同的性质。（例如，一个想要暗杀的承包商必须找到一个刺客）因此，我们的假设是，一个高度警惕的加密货币社区可以检测到许多CSC的存在，并将被激励来过滤或清除相关的交易。一个简单的潜在机制是矿工在被知名的社区作为CSC加以压制时，可以省略交易。

更积极的方法也是可能的，我们称之为受托人可中和的智能合约。智能合约系统的设计可以使权力机构，法定人数或合适的一般系统参与者有权从区块链中移除合同。与传统的基于受托人的保护相比，这种方法将具有很大的优势，因为它不需要用户注册身份。这个想法是否适合加密货币社区，是否可以确定广泛接受的权威集合，当然是开放的问题，正确的技术支持机制也是如此。

总而言之，我们在这里的工作对于加密货币社区敏感CSC的威胁非常重要，从而能够建立监控结构并制定适当的对策。我们认为，在分散式智能合约生态系统（如以太坊）的发展初期，提高认识至关重要。

8 结论

我们已经证明犯罪智能合约在智能合约影响下的去中心化货币上具有实用性。我们呈献了机密，密钥盗窃，和呼叫卡犯罪，并表明，在智能合约系统（如以太坊）中给予适当支持的情况下，它们可以利用现有的加密技术有效实施。现今，合约“公众泄露“与它的私有转化能够给予一种名为“大蛇”的以太坊脚本语言以适当支持。密钥盗窃仅需要适度的，已经能够设想的操作码为zk-SNARKs的有效部署提供支持。拥有了足够丰富的实时数据更新服务生态系统，更加标准的完美的犯罪将成为可能，更多的智能合约犯罪更加具有可能性。

我们分析道智能合约在去中心化的数字加密货币中拥有大量有前途的，合法的应用。因此禁用智能合约既不明智，很大程度上也不可能。这个在我们的工作中出现的，急迫的开放性问题是，如何制定保障措施来应对这种危险的智能合约滥用，同时继续支持他们的有益应用。

鸣 谢

这项工作是由国家科学基金会资助CNS-1314857，CNS-1330599，CNS-1453634，CNS-1514261，CNS-1514163，CNS-1518765，帕卡德奖学金，斯隆奖学金，Google学院研究奖和VMWare Research 奖。我们非常感谢Rafael Pass，abhi shelat，Jonathan Katz和Andrew Miller对密码构造和形式主义的有用的技术讨论。我们特别感谢拉斐尔·帕斯指出了这次

参 考 文 献

[1]  http://www.smartcontract.com.

[2]  http://koinify.com.

[3]  https://github.com/darkwallet/darkleaks.

[4]  AmazonEC2pricing.  http://aws.amazon.com/ec2/pricing/.

[5]  Augur. http://www.augur.net/.

[6]  Bitcoin ransomware now spreading via

spamcampaigns.  http://www.coindesk.com/  bitcoin- ransomware- now- spreading- via- spam- campaigns/.

[7]  bitoinj. https://bitcoinj.github.io/.

[8]  CRL issued bby Symantec Class 3 EV SSL CA - G3.  http://ss.symcb.com/sr.crl.

[9]  NISTrandomness.beacon.https://beacon.nist.gov/home.

[10]  Serpent.https://github.com/ethereum/wiki/wiki/Serpent.

[11]  Skuchain. http://www.skuchain.com/.

[12]  Verisign revoked certificate test page.  https://test.sspev.verisign.com:  2443/test- SPPEV- revoked- verisign.html. Accessed: 2015-05-15.

[13]  Zcash. Referenced Aug. 2016 at z.cash.

[14]  Mt. Gox thinks it’s the Fed. freezes acc based on “tainted”  coins. (unlocked now).  https://bitcointalk.org/index.php?topic=73385.0, 2012.

[15]  BlockchainAlliance.www.blockchainalliance.org, 2016.

[16]  Ethereum and evil. Forum post at Reddit;url:  http://tinyurl.com/k8awj2j, Accessed May 2015.

[17]  M. Andrychowicz, S. Dziembowski, D. Malinowski, and  L. Mazurek. Secure Multiparty Computations on Bitcoin.  InS&P,2013.

[18]  J. Bates. Trojan horse: AIDS information introductory  diskette version 2.0,. In E. Wilding and F. Skulason,  editors, Virus Bulletin, pages 3–6. 1990.

[19]  J. Bell. Assassination politics.  http://www.outpost-of-freedom.com/jimbellap.htm, 1995-6.

[20]  E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers,  E. Tromer, and M. Virza. Zerocash: Decentralized  anonymous payments from Bitcoin. In S & P. IEEE, 2014.

[21]  E. Ben-Sasson, A. Chiesa, M. Green, E. Tromer, and  M. Virza. Secure sampling of public parameters for succinct  zero knowledge proofs. In S & P, 2015.

[22]  E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza.  Succinct non-interactive zero knowledge for a von Neumann  architecture. In USENIX Security, 2014.

[23]  I. Bentov and R. Kumaresan. How to Use Bitcoin to Design  Fair Protocols. In CRYPTO, 2014.

[24]  L. Bilge and T. Dumitras. Before we knew it: an empirical  study of zero-day attacks in the real world. In CCS, 2012.

[25]  V. Blue. Cryptolocker’s crimewave: A trail of millions in  laundered Bitcoin. ZDNet, 22 December 2013.

[26]  E. F. Brickell, P. Gemmell, and D. W. Kravitz.  Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In SODA, volume 95, pages 457–466, 1995.

[27]  J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In CRYPTO ’03. 2003.

[28]  R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In FOCS, 2001.

[29]  D. Chaum. Blind signatures for untraceable payments. In CRYPTO, pages 199–203, 1983.

[30]  D. Chaum and T. P. Pedersen. Wallet databases with observers. In CRYPTO’92, pages 89–105, 1993.

[31]  N. Christin. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In WWW, 2013.

[32]  R. Cleve. Limits on the security of coin flips when half the processors are faulty. In STOC, 1986.

[33]  G. Danezis, C. Fournet, M. Kohlweiss, and B. Parno. Pinocchio Coin: building Zerocoin from a succinct pairing-based proof system. In PETShop, 2013.

[34]  K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. https://eprint.iacr.org/2015/460.

[35]  P. T. et al. Darkwallet on twitter: “DARK LEAKS coming soon. http://t.co/k4ubs16scr”. Reddit: http://bit.ly/1A9UShY.

[36]  I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In FC, 2014.

[37]  E. F. Foundation. EFF SSL observatory. URL: https://www.eff.org/observatory, August 2010.

[38]  A. Greenberg. ’Dark Wallet’ is about to make Bitcoin money laundering easier than ever. http://www.wired.com/2014/04/dark- wallet/.

[39]  A. Greenberg. Alleged silk road boss Ross Ulbricht now accused of six murders-for-hire, denied bail. Forbes, 21 November 2013.

[40]  Intel. Intel software guard extensions programming reference. Whitepaper ref. 329298-002US, October 2014.

[41]  J. Jonsson and B. Kaliski. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, 2003. RFC 3447.

[42]  A. Juels, A. Kosba, and E. Shi. The ring of gyges: Investigating the future of criminal smart contracts. Cryptology ePrint Archive, Report 2016/358, 2016. http://eprint.iacr.org/2016/358.

[43]  A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In S & P. IEEE, 2016.

[44]  V. Kotov and M. Ra jpal. Understanding crypto-ransomware. Bromium whitepaper, 2014.

[45]  A. Krellenstein, R. Dermody, and O. Slama. Counterparty announcement.  https://bitcointalk.org/index.php?topic=395761.0, January  2014.

[46]  R. Kumaresan and I. Bentov. How to Use Bitcoin to  Incentivize Correct Computations. In CCS, 2014.

[47]  D. Mark, V. Zamfir, and E. G. Sirer. A call for a temporary  moratorium on “The DAO” (v0.3.2). Referenced Aug. 2016  at http://bit.ly/2aWDhyY, 30 May 2016.

[48]  J. Matonis. Why Bitcoin fungibility is essential. CoinDesk,  1 Dec. 2013.

[49]  S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko,  D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: characterizing payments among men with no names. In IMC, 2013.

[50]  I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In S & P, 2013.

[51]  M. Moser, R. Bohme, and D. Breuker. An inquiry into money laundering tools in the bitcoin ecosystem. In eCRS, 2013.

[52]  S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. http://bitcoin.org/bitcoin.pdf, 2009.

[53]  R. Pass and a. shelat. Micropayments for peer-to-peer currencies. Manuscript.

[54]  M. Peck. Ethereum developer explores the dark side of Bitcoin-inspired technology. IEEE Spectrum, 19 May 2016.

[55]  K. Poulsen. Cybercrime supersite ‘DarkMarket’ was FBI sting, documents confirm. Wired, 13 Oct. 2008.

[56]  D. Ron and A. Shamir. How did Dread Pirate Roberts acquire and protect his bitcoin wealth? In FC. 2014.

[57]  S. V. Solms and D. Naccache. On blind signatures and perfect crimes. Computers Security, 11(6):581–583, 1992.

[58]  M. Stadler, J.-M. Piveteau, and J. Camenisch. Fair blind  signatures. In Eurocrypt, pages 209–219, 1995.

[59]  G. Wood. Ethereum: A secure decentralized transaction  ledger. http://gavwood.com/paper.pdf, 2014.

[60]  A. Young and M. Yung. Cryptovirology: Extortion-based  security threats and countermeasures. In S & P, 1996.

[61]  F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi.  Town Crier: An authenticated data feed for smart contracts. In ACM CCS, 2016. (To appear.).