CashWeb

v1.0.1

Shammah Chancellor **沙马. 常瑟罗**
shammah.chancellor@cashweb.io

Harry Barber **哈利. 巴博**
harry.barber@cashweb.io

David Schlesinger 大卫. 西列新哥
david@cashweb.io

2020 年 8 月 23 日

Abstract

本文将介绍一套开放式通信标准的协议，这套协议可以在无需仲裁的情况下限制消息滥发。发送的每一条消息都需要附加少量的虚拟币，以此来避免内容审核的需要。所有的消息都经过加密来保障用户的隐私。使用了类似于 XMPP 和 SMTP 的自治网络拓扑结构，提供非常好的可扩展性，以此达到能替代现有的通信基础设施的水准

1 介绍

1.1 历史

早期的互联网使用者，将互联网看做一个便宜且能快速分享想法，获得反馈的平台。因此，像 Usenet[7][10], Email[15][16][17][8], and XMPP[12][11][13][14] 等系统都是构建在分布式平台上的。

然而，由于这些系统内在设计的局限性，接受消息总是比发送消息显著地消耗更多的资源。消息接收方的开销包括：信息处理，存储，还有时刻的关注。这样的系统会导致大量的垃圾信息传播。这些开销也导致了我们需要一个中心的系统去发现和过滤这样的信息，而用户则不得不委托第三方来管理他们的网络身份和通讯，以此来获得使用的便利。这样一来，用户就在一定程度上失去了对自己的隐私和网络身份的控制。

2017 年 6 月，Facebook（脸书）拥有 20 亿用户。2018 年 10 月，谷歌的电子邮件服务拥有 15 亿用户[19]，Cloutfare 公司处理了互联网上 10% 信息的路由[1]。2020 年 6 月，谷歌，苹果，微软占据了 85% 的邮件客户端市场[9]。我们现在非常依赖于很少的几个公司来可靠诚实的管理关键的互联网服务。而且这些关键的服务并不是这些公司的主要收入来源。

中心化的平台给用户带来了非常一致的体验，也导致了去中心化的系统无法与之竞争。然而，随着比特币[6] 的发展，现在的技术已经有可能构建去中心化并且能达到用户期望的系统。CashWeb 协议就是充分利用虚拟币来提供无缝的上网体验，让用户能掌控自己的网络身份和隐私。

1.2 中心化的力量

1.2.1 身份管理

过去，网络服务提供商（ISPs）向客户提供电子邮件服务。当用户搬家，或者选择更换服务商的时候，他们的电邮地址也需要随之更改（例如，john.doe@sonic.net)。使用像谷歌，微软等全球公司的电邮服务就给用户省去了这种麻烦，也不用担心遗漏重要的电邮。

当自己的电邮账号托管给第三方之后，数字身份的验证就变得不是很直接了。也就导致企业和用户开始依赖于这些第三方以进行重要的通信和和维持数字身份。现在，这些已经成为我们最根本的网络身份，并且被用来作为登陆大多数网站的认证机制。如今，一旦失去对电子邮件的访问权限的话，将会给生活带来极大的不便，甚至产生严重的后果。维持对电子邮件的访问有时候并不是我们完全可以控制的，譬如密码被窃取了，或者邮件提供商因为某些原因中止服务。

如果我们因为个人原因想要更换邮件服务商的话，这会是个很艰巨的任务。当用户想让服务提供商承担责任的时候，可能发现自己无能为力。自己的网络身份完全由自己控制是非常重要的。

1.2.2 垃圾信息

电子邮件的设计初衷是作为人与人以及机器与机器之间的通信系统，但如今大部分的电子邮件是机器发给人的。这些邮件大部分是无用的广告，它们需要消耗计算资源，也需要人花时间去评估和删除。大型的中心化的电邮提供商，譬如 Gmail[a]和 Hotmail[b]，在监测到给大量用户发送类似邮件的情况的时候，可以将那些邮件过滤掉。

比较讽刺的是，跟我们日常打交道的公司也在发送越来越多的市场推广邮件。电邮服务商一般都不会过滤这些邮件，虽然这样做违反了他们只给用户提供有价值信息的初衷

与此同时，我们很多重要的和熟人之间的通信也迁移到了其他平台，譬如手机短信，Telegram，Messenger，Whatsapp，Twitter，和 Signal。这些平台为了能控制垃圾信息，会要求用户至少提供电话号码，或者电子邮箱来注册账号。如果一个账号在平台上产生了大量无用的内容时候，账号就会被限制，甚至删除，关联的电话号码或者电子邮箱也会被永久禁用。然而，这也意味着屏蔽垃圾信息会不可避免的和现实世界的个人关联起来。

1.2.3 后果

大型服务给我们提供了很多便利，这些中心化的服务提供商也需要获得收入来支付他们的运维开销，并获得利润。很多服务提供商提供“免费”的电邮和网站资源。

> “如果你没付钱，那你就是被卖的产品。”
>
> - 罗伯特. 丹尼而森

有些服务商也提供付费的电邮服务，其实就是把隐私保障作为一个卖点。即便如此，这些服务提供商也仍然可以和免费服务商一样有权限读取用户的数据。在经济利益的驱动下，他们可能在保护隐私的伪装下出售用户的信息。就算不管这些付费服务商的职业道德如何，那些和免费邮件服务商之间的电邮依然会被索引分类，来服务于广告。

在非电邮的系统里面，我们的身份越来越和我们的电邮地址或者电话号码绑定在一起。这就意味着所有其他的账户和数字交互之间有明显的联系。能够把一个用户的所有数据联系在一起，形成完整的个人资料，这对广告商来说非常有价值。

[a] `https://mail.google.com`

[b] `https://outlook.live.com`

事实上，很多公司（例如，LiveRamp）会从不同的服务商那里购买数据，并基于电邮和电话号码把数据联系在一起。设备的指纹也被用来跟踪网络浏览历史，并用来和电邮及电话号码关联，这样用户即便用多账户登陆访问也能被关联起来。

这样做的目的是为了提供非常精确的广告投放。这对那些想要寻找产品的用户可能是有帮助的。然而，这些数据也能被用来达到很多其他目的，譬如跟踪政治上的反对派的网上行踪。这类用途并不在本文的讨论范围内。

2 CashWeb 协议

2.1 原理

CashWe 会依据以下的原则来提供替代现有系统的方案：

2.1.1 简洁

CashWeb 必须提供一种需要很少专业知识的解决方案来吸引大量的用户。对于没兴趣自己做硬件配置的用户，必须可以让第三方来提供托管和运营服务。

同样的，基础的协议也必须尽可能的简单，从而能吸引大量的开发者来参与

2.1.2 易迁移性

用户必须能控制自己的网络身份，并且有能力从一个服务提供商迁移到另一个。当用户有能力可以方便地切换服务商以后，用户也就可以让服务商对他们的行为承担职责，也会促进良性竞争。

2.1.3 可恢复性

在身份丢失的时候，必须有办法能妥善的恢复。为了能将认证从一个服务商迁移到另一个服务商，将使用非对称密码技术，这样身份认证就不是 CashWeb 服务提供商的职责了。如果一个用户丢失了手机或者电脑，而且另一个人获得了他们的私钥，用户的身份也必须可以恢复。

2.1.4 安全和隐私

任何第三方，包括 CashWeb 的服务提供商，都不能获取消息的内容，以此来保护用户的隐私。所有用户和用户之间的通信都默认会被加密，并使用已确立的加密标准，如“高级加密标准”（AES）和“椭圆曲线加密”（ECC）。

除了默认的安全性，用户必须能够进一步利用现有的覆盖网络（如：Tor）来加强安全性。

2.1.5 无需许可

协议必须是开源的，必须有维护良好的参考实现和文档。这个模式可以让潜在的开发者很容易加入，也利于 CashWeb 的生态建设。

基础协议是开源并公开维护的，具体的软件实现可以不需要开源。

2.1.6 可扩展的隐私

协议应当允许用户决定他们自己想要的隐私级别。基础层需要提供协议机制来让用户来决定保密的内容，同时给大多数用户提供合理的默认设置。

2.1.7 简洁性和可扩展性

最基本的功能集必须非常简单，从而能获得普遍地应用。同时又需要能方便地扩展，允许企业能在不影响现有用户和软件客户端的前提下提供更复杂的功能。

2.1.8 易集成性

协议应当基于大多数用户和软件开发者熟悉的标准网络技术，如 HTTP。

2.2 核心概念

为了达成以上的需求，CashWeb 协议围绕以下的核心概念：

2.2.1 网络标准

CashWeb 系统遵循现有的网络标准，从而可以比较容易且快速地与现有的协议和基础设施整合。“持有者”类型的代币将和现有的虚拟币支付标准一起被广泛使用，来验证访问那些已经用虚拟币购买的资源。消息的收发会使用 HTTP/2[5] 和 WebSockets[3]，从而能使用现有的网络设施和服务。

2.2.2 虚拟币

为了能让通信安全，并实现 CashWeb 系统的无许可性，就需要让任何个体都没办法发送大量未经请求的消息。所有消息的发送，都需要发送者同时支付给接受者一点虚拟币。为了实现这个需求，就必须把支付作为设计的核心。

传统的系统需要可信赖的第三方，以及和传统银行系统较复杂的整合。有远见的人士，如 Hal Finney（哈而.菲尼）考虑过用“可重用的工作验证”（RPoW）来解决这个问题。不幸的是，菲尼最初的设计需要中心化的管理 RPoW 的代币，所以并不切合实际。然而，这个想法现在可以通过点对点的现金系统如比特币（BTC）来实现。

使用虚拟币，而不是和传统银行整合，非常切合安全、开放、无需许可的通信。同一个密钥既可以用来收发资金，也可以用来加密消息。

不幸的是，比特币网络无法支持一个广为使用的消息平台的交易量。大多数的其他虚拟币系统也并不打算支持除去垃圾信息外的电子邮件量级的交易量。有些能支持这个交易量的系统又采用中心化管理的经济政策，那样的化，管理者就会有权限制消息的收发。

因此，我们选择了使用比特现金（BCH），原因是其路线图非常切合 CashWeb 项目的需求。比特现金的路线图非常强调全球范围内即时交易的重要性，并构建于中本聪的代币化的工作验证（PoW）概念。

2.2.3 身份

用户的身份是假名化的，并和一个公钥联系在一起。只需要一个主密钥，就可以非常方便且便宜地生成一串公开身份。每个身份相关联的密钥都可以通过给比特现金网络上矿工很小的付费来确认。这些微小的付费包含了一个可进行数字验证的身份核实来证明给任何第三方进行了付费。

同时，这些假名也可以在未来的一个时刻被证明是从另一个隐藏密钥所产生的。有了这种证明，系统就能撤销和假名相关的某个密钥。也就能在无需建立新的信任的情况下，通知假名的联系人。具体的身份识别机制的细节会放在详细的协议规范里面，协议是可以根据需求来做进一步的扩展的。

2.2.4 消息格式

所有在 CashWeb 系统里面的消息都使用协议缓存（Protocol Buffer）[18] 的消息格式。如今，协议缓存已经被广泛应用，在很多程序设计语言中都能容易的实现，并序列化成二进制数据。

2.3 基础设施

2.3.1 密钥服务器

CashWeb 协议包括一个密钥服务器网络，他们提供公开和分布式的元数据注册表。注册表是用来保存和密钥相关的少量元数据。密钥服务器的所有数据在网络上被复制来对抗审查。点对点的协议能让数据最终保持一致。

元数据将用户公钥的哈希值来建立索引，并包含了公钥，数据信息，一个能验证数据完整，可靠，不可否认的数字签名。元数据可以通过提供有效的数字签名来更新。

数据上传到密钥服务器是由“支付证明协议”（POP protocol）来保护的。这提供了一种将链上数据和具体更新联系在一起的方法，这样可以让密钥服务器的数据复制具有抵抗“拒绝服务攻击”（DoS）的能力。

定制的 CashWeb 密钥服务器跟 GPG 基础设施相比较，有以下的优点：

- 从一开始就考虑了对抗 DDoS 的机制，有更简单可靠的设计。
- 使用 HTTP2，交互时简单得多。立刻和市场现有的负载平衡器兼容。
- 数据内容的格式比现有的密钥服务器更简洁。同时，在给定的一个地址相关的数据里面，也可以提供 X.509 的认证。

CashWeb 密钥服务器可以有很多应用，虽然我们这里的主要用途是记录一个指向特定的管理用户信息的中继服务器。任何能访问密钥服务器网络及其中一个主机的用户，都可以查询密钥服务器网络的地址，然后重定向到某个特定的中继服务器来启动通信。

密钥服务器的另一个功能是可以帮助用户在丢失上网密钥的情况下让密钥作废。密钥服务器可以用无需信任的方式把新的密钥发布给已有的联系人。这就允许了比特币的密钥轮换，这也解决了自比特币白皮书问世以来，所有虚拟币的一个明显的缺陷。

2.3.2 中继服务器

中继服务器达到了 POP 和 SMTP 服务器加在一起的目的。它们帮助客户接受消息，做一些基本的完整性验证。另外也存放个人简介信息，包括头像和一些其他信息。虽然目前中继服务器仅提供消息传送，个人的名字，和图标，但其软件可以很容易地进行扩展，来提供很多有用的功能，例如状态信息，微博等等。

密钥服务器和中继服务器是因为不同的关注点而区分开的：

- 密钥服务器提供全局的复制，从而让少量的没有加密的数据能够抗审查。
- 中继服务器只服务特定的用户，所以可以比较经济地为大量加密以后的用户数据提供托管服务。

在中继服务器上开户可以通过一个标准的 HTTP 远程调用来完成。授权和认证机制基于 POP 协议。中继服务器能够生成假名账号并产生收益。用户可以很方便使用标准的账户注册 API 来将他们的账号在不同的中期服务提供商之间迁移。

2.3.3 消息客户端

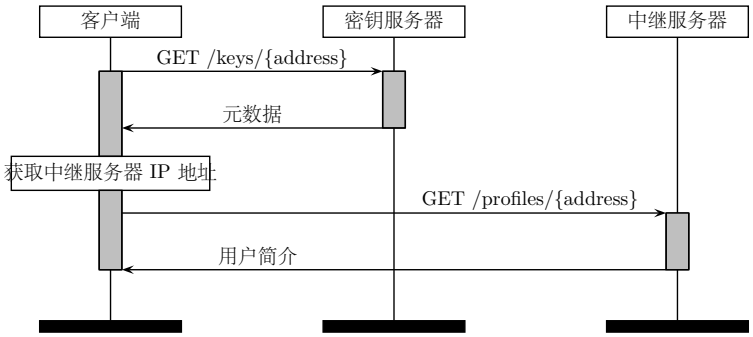
终端用户的客户端也是必须的，它能够非常方便地和系统交互。客户端需要管理收发消息相关的资金，在密钥服务器上更新用户接收消息的中继服务器，以及和中继服务器联络并处理收到的消息。

消息客户端是 CashWeb 系统最复杂的部分。它需要管理那些用于生成时间戳和其他微支付的资金。因此它同时也要承担数字钱包的职责。另外，协议的大部分工作要处理加密的结构化的数据，消息客户端也必须能分析和处理它们。无论是密钥服务器，还是中继服务器，很大程度上来说，都不需要知道客户端之间交流时使用的协议。

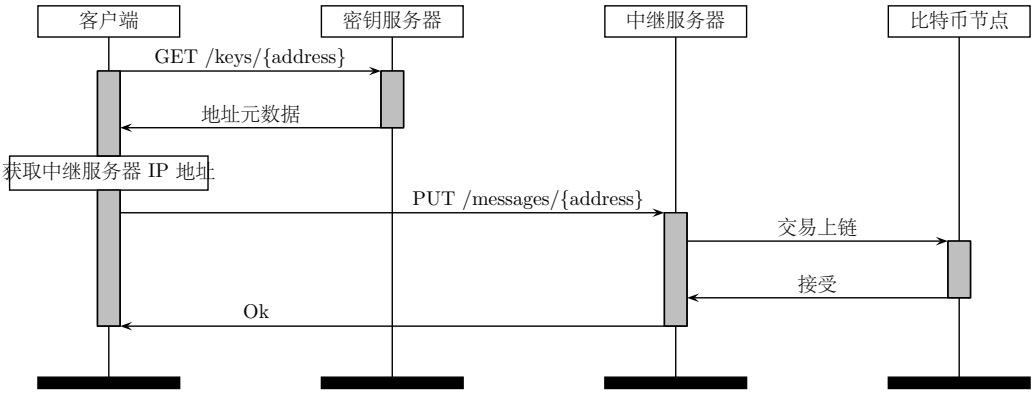
这样的设计可以在无需对底层通信机制大规模修改的前提下让功能不断进化。钱包（客户端）的开发者可以不断的增加和改进功能。钱包只需要忽略那些他们不支持的数据类型，这样就可以在不需要达到大规模共识的前提下对协议进行添加修改。

2.3.4 协议流程图

msc 获得用户简介



msc 消息发送



3 应用

3.1 标准化的授权和认证

支付证明（POP）协议可以让我们标准化，且无缝地使用现有的 HTTP 技术，并且不需要使用很复杂的支付系统和账户管理。可以半匿名的购买一个 JWT[4] API 代币，无需账户管理系统、支付前端和其他复杂的系统。

3.2 分布式的身份管理

有了标准的密钥服务器基础设施以后可以给很多应用带来好处，譬如可以提供加密安全，可更新的，通过二维码或其他方式分享联系人。信息轮转的功能可以用来做密钥撤销和轮换等重要操作。提供了完善的机制来实现分布式，且无需信任的网络身份管理。有了中立的身份管理基础设施，就会非常好地鼓励所有的网络用户来参与，这和中心化的身份管理提供商（如谷歌，脸书）形成鲜明对比。

3.3 开放的通信

支付证明协议，密钥服务器，中继服务器组合在一起，就能提供高级的隐私功能，和没有垃圾的通信。发送有附加值的结构化消息，就允许各种各样，基于一定费用的，人与人，人与机器，以及机器和机器之间的消息处理。最显而易见的应用就是点对点的支付和消息。然而，也有潜力应用到其他服务，例如，网络机器人，网络商城，Web3.0 的应用，分布式金融（Defi）协议等等。

4 结论

CashWeb 协议的目标，是针对那些一直有新的专有方法解决的基本的技术问题，提供抗审查的解决方案。CashWeb，以及它基于的虚拟币的技术，能够去除中心化的认证，身份管理，和消息发送。在无需授权金融媒介的情况下，无缝地在互联网基础设施上做支付应用。

添加支付的点对点通信会挑战现存的互联网巨头。避免了中心化的审查，CashWeb 就能给所有愿意合作的各方提供一个数字化的 “中立平台”。也会消除对 “围墙里的花园” 类型的通信网络的需求。

数字货币和消息发送相结合，最早是哈而. 菲尼（Hal Finney[2]）提出的，给所有的参与者在全球对话中提供平等的位置。这将是非常关键的一步，关键的基础设施，随着技术的进步，捍卫人权和经济自由。也有潜力去改变人类交流和思考的方式，让我们把注意力更多的集中在客观有价值的信息上。

因为 CashWeb 是运营在整合的微支付上，用户不再是商品。

References

[1] Cloudflare. 非洲网络流量增长和未来的预期. 网络文档. 2018. url: https://blog.cloudflare.com/african-traffic-growth-and-predictions-for-the-future/.

[2] Hal(哈而. 菲尼) Finney. 可重用的工作证明. 网络文档. 2004. url: https://nakamotoinstitute.org/finney/rpow/index.html.

[3] IETF. RFC6455 - WebSocket 协议. 网络文档. 2011. url: https://tools.ietf.org/html/rfc6455.

[4] IETF. RFC7519 - JSON 网络代币 (JWT). 网络文档. 2015. url: https://tools.ietf.org/html/rfc7519.

[5] IETF. RFC7540 - 超文本传输协议 2(HTTP/2). 网络文档. 2015. url: https://tools.ietf.org/html/rfc7540.

[6] 中本聪. 比特币: 一个点对点的电子现金系统. 网络文档. 2008. url: https://bitcoin.org/bitcoin.pdf.

[7] 丹. 孔，肯. 莫赤森，和查尔斯. 林西. 网闻文章格式. RFC 5536. Nov. 2009. doi: 10.17487/RFC5536. url: https://rfc-editor.org/rfc/rfc5536.txt.

[8] 史蒂夫. 赫尔和阿历克斯. 莫尔尼科夫. 对条件性存储操作或快速标示改变的重新同步的 IMAP 扩展. RFC 4551. June 2006. doi: 10.17487/RFC4551. url: https://rfc-editor.org/rfc/rfc4551.txt.

[9] 李特摩斯实验室. 电邮客户端的市场份额. 网络文档. 2020. url: https://nakamotoinstitute.org/finney/rpow/index.html.

[10] 查尔斯. 林西和拉丝. 阿尔巴里. 网闻的架构和协议. RFC 5537. Nov. 2009. doi: 10.17487/RFC5537. url: https://rfc-editor.org/rfc/rfc5537.txt.

[11] 皮特. 圣安祝. 可扩展消息和存在协议（XMPP）实时通信及现状. RFC 3921. Oct. 2004. doi: 10.17487/RFC3921. url: https://rfc-editor.org/rfc/rfc3921.txt.

[12] 皮特. 圣安祝. 可扩展消息和存在协议（XMPP）核心. RFC 3920. Oct. 2004. doi: 10.17487/RFC3920. url: https://rfc-editor.org/rfc/rfc3920.txt.

[13] 皮特. 圣安祝. 将可扩展消息和存在协议（XMPP）映射到现状和实时通信（CPIM). RFC 3922. Oct. 2004. doi: 10.17487/RFC3922. url: https://rfc-editor.org/rfc/rfc3922.txt.

[14] 皮特. 圣安祝. 端对端签名和对可扩展消息和存在协议 (XMPP) 的对象加密. RFC 3923. Oct. 2004. doi: 10.17487/RFC3923. url: https://rfc-editor.org/rfc/rfc3923.txt.

[15] 皮特. 瑞斯尼克. 互联网消息格式. RFC 5322. Oct. 2008. doi: 10.17487/RFC5322. url: https://rfc-editor.org/rfc/rfc5322.txt.

[16] 马沙尔.T. 罗斯博士 和 约翰.G. 迈尔斯. 邮局协议-第三版. RFC 1939. May 1996. doi: 10.17487/RFC1939. url: https://rfc-editor.org/rfc/rfc1939.txt.

[17] 约翰. 克冷辛博士. 简单邮件传输协议. RFC 5321. Oct. 2008. doi: 10.17487/RFC5321. url: https://rfc-editor.org/rfc/rfc5321.txt.

[18] 谷歌. 协议缓存语言规范第三版. 网络文档. 2015. url: https://developers.google.com/protocol-buffers/docs/reference/proto3-spec.

[19] 谷歌邮件. Oct. 2018. url: https://twitter.com/gmail/status/1055806807174725633.