CASHWEB

A PREPRINT

Shammah Chancellor
shammah.chancellor@cashweb.io

Harry Barber
harry.barber@cashweb.io

July 20, 2020

ABSTRACT

CashWeb is a suite of protocols providing an open messaging standard. It uses a federated network topology, similar to XMPP and SMTP, to provide its scaling characteristics. It is unique in that it provides end-to-end encryption and abuse mitigation without the need for central moderation.

1 Introduction

1.1 History

Many early adopters of the internet saw it as a platform for inexpensive, and fast, sharing of new ideas. Systems like Usenet[1][2], Email[3][4][5][6], and XMPP[7][8][9][10] were envisioned as decentralized platforms for this purpose.

However, due to the economics of network messaging, the cost of message processing is placed on the recipient with respect to processing power, storage, and mental attention. We are increasingly relying on centralized systems to identify and filter these messages.

Eventually, users delegated authority over their online identities and communications to third-parties in exchange for convenience. This has resulted in a loss of their privacy and control over their identities.

As of June 2017, Facebook reported 2 billion users. By October 2018, GMail reported 1.5 billion active users[11] and Cloudflare routed 10% all internet traffic[12], June 2020, Google, Apple, and Microsoft held a total of 85% of total market share of email clients[13].

This has all happened rather passively. Users have allowed centralization to happen simply due to convenience. A few major inconveniences that have driven internet users to abandon their rights are: identifying important communications amongst unsolicited marketing advertisements, and maintaining the address where they receive communications. The design of the CashWeb protocol is centered around restoring user authority over their digital identities and privacy.

1.1.1 Spam

While email was envisioned as a person-to-person and machine-to-machine messaging system, the vast majority of emails are now machine-to-human. These messages largely consist of useless advertisements, but require processing power and human attention to evaluate and identify for deletion. Large-scale centralized email providers (e.g. Gmail[1] and Hotmail[2]) benefit from their message volume by being able to identify similar messages sent to a large number of different customers and filter them out.

Ironically, companies that users actively do business with are sending more "opt-in" marketing messages. These are largely served unfiltered by these centralized platforms, despite the original promise of these platforms to provide users with only high-value communications. Email has largely become a historical record for our business notifications.

[1] https://mail.google.com
[2] https://outlook.live.com

As a result of Spam, much of our high-value communications with personal acquaintances have moved to other digital platforms like SMS, Telegram, Messenger, WhatsApp, Twitter, and Signal. However, in order to dissuade spam on these platforms, they require providing a telephone number, or email, or both, in order to have an account.

If a "user" starts producing too much unwanted content on these systems, the account is restricted or deleted. The pinned phone number, or email address, is permanently banned. These identifiers are difficult to obtain due to the restrictions around sign-ups placed by email providers, and the cost associated with purchasing a phone number.

1.1.2 Identity Management

In the early days of the internet, internet service providers (ISPs) also offered email services to their customers. However, when a user moved addresses or wanted to switch service providers, their email address would change due to the reliance on routing information provided within the address (e.g. john.doe@sonic.net).

Our important correspondences (e.g. Bills, Pay statements, Second-factor authentication) have moved to email. The need to provide all these services with a new email address is a major inconvenience. In order to avoid this inconvenience, many internet users have opted to have email addresses with providers which were independent from their ISPs.

1.2 Consequences of Centralization

1.2.1 Loss of Identity

Due to the delegation of address management, our digital identities have become more stable. As a result, businesses and users have begun to rely on them for more and more important communication and as a digital identity.

Now, loss of access to our email has become a life-altering event with significant consequences. Such a loss may result due to a compromised password, or a judgement by the provider.

Additionally, should we want to migrate email providers for personal reasons, it has become an insurmountable task. Users may find themselves effectively powerless to hold their service providers accountable.

1.2.2 Loss of Privacy

Centralized providers must generate revenue to maintain their infrastructure and generate a profit. However, many users when offered a free product vs one that costs money opt for free services. As such, many of these services provide access to email for free in return for the ability to advertise to the user in a controlled way. When compared to the deluge of spam users would otherwise receive, this is an acceptable trade-off.

The product these companies offer is not actually email, but marketing access to other businesses. In order to provide the best possible product, they use the email contents in order to generate digital profiles for users which are then used to pair us with advertisements. This in turn allows for higher profit margins.

Some providers still provide paid email access while essentially offering privacy as a product. However, they still have access to the same data about their users as free services. There becomes a financial incentive to sell this data while maintaining a guise of privacy. Regardless of the ethics of paid providers, emails exchanged with free email providers (e.g. Gmail) are still indexes and categorized for advertising.

Again, in non-email systems, our identity is increasingly tied to our email or phone numbers. This association means that there is a clear association between all other accounts and digital interactions. Being able to collate all this data about a user, and form a more complete profile, is extremely valuable to advertisers.

Indeed, various companies purchase data across multiple services, and collate it based on emails and phone numbers. Device fingerprinting is also employed to combine this data with ones web browsing history. Device fingerprints also provide a way to associate email addresses and phone numbers should users log in to two different accounts during the same session or on the same computer.

The stated purpose of this is to provide highly-specific advertising. This may be desirable to users in finding products they want. However, are many other purposes this data is used for which are of concern. These other uses are outside the scope of this paper.

2 The CashWeb Protocol

2.1 Philosophy

In order to provide an alternative to the existing system we adhere to the following principles:

2.1.1 Simplicity

In order to support the needs of non-technical users, CashWeb must be a solution which requires minimal technical expertise. It must be possible to use third-party providers which provide hosting, and operation, of various services.

2.1.2 Migratability

Users of the CashWeb system must have control over access to their own identities, and the ability to migrate from one service provider to another. This enables them to hold service providers accountable for their actions. A name resolution system which is distributed in nature. The name resolution system must be resistant to denial of service attacks, and have no central authority.

2.1.3 Recoverability

In the case of a loss of identity, it must be possible to recover gracefully. In order to make authentication transferable from one service provider to another, asymmetric cryptography must be employed so that identification is not a responsibility of CashWeb service providers.

2.1.4 Security & Privacy

In order to protect the privacy of users, the contents of messages must not be readable by third parties, including CashWeb service providers. All communications between two parties should be encrypted by default. The systems must be compatible with existing overlay networks to provide additional security such as Tor.

2.1.5 Permissionless

While individual software implementation may be privately maintained, the protocol must allow for low barrier entry into the ecosystem. Reference implementations should be provided.

2.2 Central Concepts

In order to meet the above requirements for the CashWeb system, the following design decisions were made.

2.2.1 Web Standards

The CashWeb system adheres to the established web standards to allow quick and easy integration into existing protocols and infrastructure. "Bearer"-style tokens are used extensively in combination with existing cryptocurrency payment standards to allow authenticated access to resources to be purchased with Bitcoin tokens.

2.2.2 Cryptocurrency

In order to maintain secure communication, and the permissionless nature of the CashWeb system, it must be impractical for a single party to send large volumes of messages. All messages sent should impose a cost on the sender that is paid to the recipient. In order to support this requirement, payments need to be at the center of the design.

Traditional systems require trusted third parties and complex integration's with the traditional banking system. Visionaries like Hal Finney conceived of this problem being solved through "reusable proof of work" (RPoW). However, his original design was impractical, but ultimately made possible through the use of Bitcoin Tokens.

Unfortunately, the Bitcoin network does not support the transaction volume which would be associated with a widely used messaging system. Most other cryptocurrency systems also do not intend to support volumes on the order of email (sans spam). The ones that do support these volumes have centrally managed economic policies. Such management would give them authority over the ability to send and receive messages.

Thus, Bitcoin Cash was selected due to its roadmap being highly compatible with the requirements of the CashWeb project. The roadmap purports to desire to support the majority of payments on earth. Also, It uses a proof-of-work with no central issuing authority for tokens issuance.

Additionally, using a cryptocurrency, instead of traditional banking integrations, synergizes well with the secure communication requirement. The same keys used to send and receive funds can also be used to provide encryption for messages.

2.2.3 Identity

Each user identity is pseudonymous, and associated with a public key. These public identities can be easily an inexpensively generated from a single master key. Each identity key is acknowledged by the network via various small payments to the miners of the Bitcoin Cash network which include a cryptographically verifiable commitment to the identity.

Additionally, these pseudonyms can be made such that they can be proven to have been derived from another hidden key at a later date. Such proofs enable the specific key associated with a pseudonym to be revoked, and rotated, in a trustless way. This enables the contacts of the pseudonym to be informed without the need for re-establishing trust.

The specific details of these identity schemes are left to further detailed protocol specifications, and the protocol is extensible to future schemes should a need arise.

2.2.4 Message Format

For all messages within the CashWeb system use the Protocol Buffer message format. "Protobufs" are now in wide use, easy to implement in a variety of languages, and serializable to binary.

2.3 Components

2.3.1 Keyservers

The CashWeb protocol includes a network of keyservers which provide a public & distributed metadata registry. The store is intended to track small amounts of metadata associated with cryptographic keys. This information stored on the keyservers is replicated across the network to provide censorship resistance. A peer-to-peer protocol is included which provides eventual consistency.

This metadata is indexed by the hash of ones public key and includes said public key, a body of information, and a signature covering the body providing integrity, authentication, and non-repudiation. Metadata updates are permissioned by providing valid signatures.

Uploading data to the keyserver is protected by a specific version of the "Proof-of-Payment protocol" (POP protocol) outlined above. This provides a way to anchor on-chain value to specific updates and therefore allowing DoS resistant replication across the keyserver network.

The specialised CashWeb keyserver has the following benefits over existing GPG infrastructure:

- Anti-DDoS mechanisms are considered from inception and hence we can arrive at a more simple and robust overall design.
- HTTP2 makes it significantly simpler to interact with, for example, it is immediately compatible with off-the-shelf load balancers.
- The payload format is more concise than what existing keyservers provide. However, X.509 certificates can also be provided inside an entry associated with a given address.

  The CashWeb keyserver can be used for a wide-range of applications (which are eluded to below), however our primary use-case is to record a pointer to the specific relay server managing that users messages. In this way any user with access to the keyserver network and a hosted address, may lookup the address on the keyserver network and then redirect to their specific relay server in order to bootstrap communication.

  Another possible functionality would be revocations of keys in the event that ones private key is lost. It is possible to generate the keys used for identities such that you can revoke, and notify contacts automatically, of a new key. This allows for key-rotation on Bitcoin, which has been a significant deficiency in cryptocurrencies since the Bitcoin whitepaper was written.

2.3.2 Relay Servers

Relay servers provide the combined purpose of both POP and SMTP servers. They accept messages on behalf of the clients, and verify basic integrity of these messages. They also host profile information including avatars and other information. While this server currently only provide messaging, profile names, and icons, they can easily be extended to provide status messages, microblogs, and other potentially useful functions.

The distinction here between keyservers and relay servers is made due to seperation of concerns: * Keyservers provide replication and therefore censorship-resistance for small amounts of unencrypted data. * Relay servers do not provide replication and therefore can cheaply host large amounts of encrypted personal data.

Uploading to, and pulling messages from, the relay server should be protected by the Proof-of-Payment protocol. Unlike the keyserver, the specific implementation of the POP token protocol is not enforced due to lack of replication. Clients supporting the POP token protocol should be agnostic to any variations of implementation across relay servers, this allows relay servers to provide a variety of hosting plans.

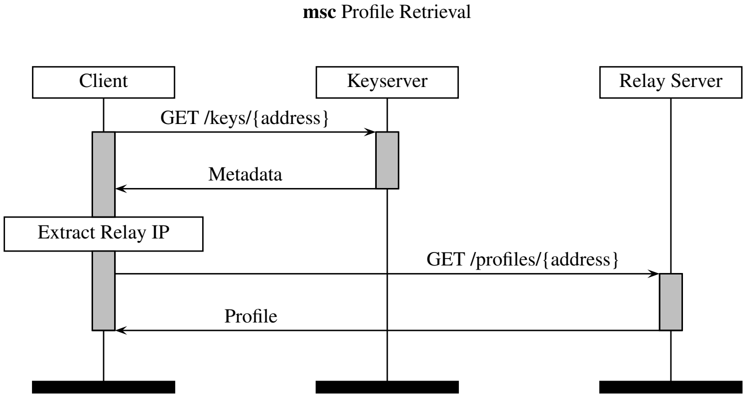
2.3.3 Messaging Client

An end-user client is necessary for interacting with this system in a easy-to-use way. The client serves to manage funds associated with sending and receiving messages, update keyservers as to which relay server the user accepts messages, and connect to and process messages received from the relay servers.

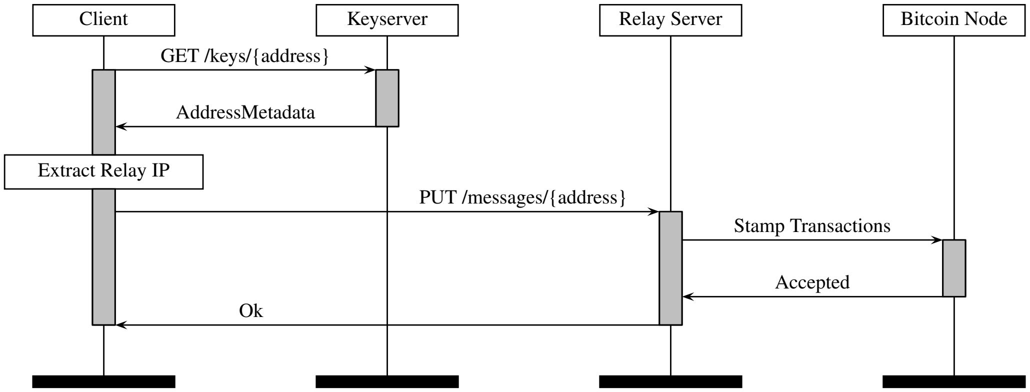
The messaging clients are the most complex portion of the CashWeb system. The majority of the protocol is handled via encrypted structured payloads that wallets needs to parse and process. Both the keyserver and the relay server are for the most part ignorant as to the protocols that wallets may use to talk to each other.

This allows the functionality to evolve gracefully over time without needing major changes to the underlying infrastructure. Wallet developers may add and evolve functionality over time. Wallets need only ignore payload types they don't understand, while enabling protocol changes without needing wide-scale consensus on additions.

2.3.4 Protocol Flow



msc Message Send



3 Applications

3.1 Proof-of-Payment Protocol

The Proof-Of-Payment (POP) protocol enables the standardized use of HTTP APIs as traditionally used in the technology industry, but without the use of complicated billing infrastructure and account management. It allows the purchasing pseudnonymously of a JWT API token, without requiring accounts, billing frontends, or other complicated infrastructure.

3.2 Keyserver Infrastructure

Having a standard keyserver infrastructure benefits a range of applications. Such as cryptographically secure, but updatable, contact exchange via QR codes or other mediums. The ability to rotate this information allows for important key revocation, and rotation events, for end-users. It provides a comprehensive mechanism for managing an online identity.

3.3 CashWeb Infrastructure

The combination of the POP protocol, keyservers, and relay servers, allows for advanced privacy features and SPAM-free communication. The ability to send structured messages with attached value, allows for all kinds of fee-based machine-to-machine message queue processing. The most obvious application of this is peer to peer payments and messaging. However, the potential for other interesting services such as robots, which exchange value as well as information, can be imagined.

4 Conclusion

The CashWeb protocols aim to be censorship-resistant, like the underlying cryptocurrency technology. Attaching payments to P2P communications enables the decentralization of existing power structures, and removing the control that intermediaries have over humans talk to one another. This mechanism, originally imagined by Hal Finney[14], provides equal footing in participating in global dialog. This is a critical step forward, and critical infrastructure, in maintaining human rights as technology continues to evolve. It also has the potential to change the way humans communicate with one another and think.
[15]

References

[1] Dan Kohn, Ken Murchison, and Charles Lindsey. Netnews Article Format. RFC 5536, November 2009.

[2] Charles Lindsey and Russ Allbery. Netnews Architecture and Protocols. RFC 5537, November 2009.

[3] Pete Resnick. Internet Message Format. RFC 5322, October 2008.

[4] Dr. Marshall T. Rose and John G. Myers. Post Office Protocol - Version 3. RFC 1939, May 1996.

[5] Dr. John C. Klensin. Simple Mail Transfer Protocol. RFC 5321, October 2008.

[6] Steve Hole and Alexey Melnikov. IMAP Extension for Conditional STORE Operation or Quick Flag Changes Resynchronization. RFC 4551, June 2006.

[7] Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 3920, October 2004.

[8] Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 3921, October 2004.

[9] Peter Saint-Andre. Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM). RFC 3922, October 2004.

[10] Peter Saint-Andre. End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP). RFC 3923, October 2004.

[11] Gmail, October 2018.

[12] Cloudflare. African traffic growth and predictions for the future. Web document, 2018.

[13] Litmus Labs. Email client market share. Web document, 2020.

[14] Hal Finney. Reusable proofs of work. Web document, 2004.

[15] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Web document, 2008.