Decentralized Semantic Identity

José G. Faísca
COPELABS*
ECATI, Universidade Lusófona

José Q. Rogado
COPELABS*
ECATI, Universidade Lusófona

ABSTRACT
This paper examines a semantic approach for identity management, namely the W3C WebID, as a representation of personal information, and the WebID-TLS as a decentralized authentication protocol, allowing individuals to manage their own identities and data privacy. The paper identifies a set of important usability, privacy and security issues that needs to be addressed, and proposes an end to end authentication mechanism based on WebID, JSON Web Tokens (JWT) and the blockchain. The WebID includes a personal profile with its certificate, and the social relationship information described as the RDF-based FOAF ontology. The JWT is a standardized container format to encode personal related information in a secure way using "claims". The distributed, irreversible, undeletable, and immutable nature of the blockchain has appropriate attributes for distributed credential storage and decentralized identity management.

CCS Concepts
Information systems~World Wide Web • Security and privacy • Computer systems organization~Distributed architectures

Keywords
semantic web; WebID; identity; authentication; blockchain; decentralization; DNS; json web tokens; P2P file system

1. INTRODUCTION
This paper describes the components and interplay of a decentralized semantic identity proposal, which has been developed as a proof of concept for distributed digital identity scenarios. The research which is at the origin of this paper introduces a fully decentralized identity approach [1] based on the blockchain concept [2]. The semantic representation of personal information is facilitated by WebID profiles [3], which are published in a peer-to-peer (P2P) file system. Besides, the WebID profile is used for authentication, in a decentralized authentication paradigm, where JSON Web Tokens [4] are used to encode personal related information. In section 2, the case of the DNS and WebID identity approaches are discussed. In section 3, Namecoin, the blockchain supporting the proposed solution is briefly described. In section 4, the Interplanetary File System (IPFS), the chosen repository for user profile storage, is examined. In section 5, a decentralized authentication implementation is presented, using the JWT concept and the other artifacts previously described. Finally section 6 presents a summary.

SEMANTiCS 2016, September 12-15, 2016, Leipzig, Germany
ACM 978-1-4503-4752-5/16/09.
http://dx.doi.org/10.1145/2993318.2993348

2. SELF-SOVEREIGN IDENTITY
Centralized digital identity ties the individuals to the administrative domains of the identity provider. A self-sovereign identity, in opposition, is created and maintained by individuals [5], for their own specific usage.

2.1 Domain Name System
A self-sovereign identity may take the form of domain names registered at a domain name registrar, accredited by ICANN [6], the main governing body for domain names or other organizations such as OpenNIC [7]. By registering and using a domain name in the Domain Name System (DNS) resource records, an individual can create an online identity. However, the DNS is a critical yet centralized component of the Internet [8], since DNS servers are generally controlled by governments and/or large corporations, and therefore may take advantage of their power to hijack, censor, or spy on users' Internet usage.

2.2 WebID
Another self-sovereign identity may take the form of a WebID profile document, a best-practice conceived in order to simplify the creation of user public online identity. WebID enables global identification and authentication in a distributed manner, by combining Linked Data and asymmetric cryptography. A WebID is a Uniform Resource Identifier (URI) with an HTTP or HTTPS scheme, which denotes a profile document describing an individual. The WebID URI must be one that dereferences to a profile document the user controls. The following WebID URI is an example of an identifier that refers to (denotes) an individual, or more generally an agent:

```
http://bob.example.com/id#me
```

In essence, a WebID profile is a web resource document consisting of data, structured according to the Resource Description Framework (RDF) model [9]. The description of the agent can be performed in any suitable vocabulary, but the Friend of A Friend (FOAF) [10] emerged as the industry standard vocabulary for that purpose.

The WebID specifications also defines the WebID-TLS, a standard for authentication over TLS (Transport Layer Security). This authentication mechanism relies on a customized RSA X.509 client certificate [11]. Such a certificate may either be signed by a Certificate Authority (CA) or self-signed. The WebID profile document includes the certificate public key that is used in the authentication process, among other items. The WebID profile contains a description of an RSA public key, which is associated to the WebID by using the **cert:key** property from the W3C certificate ontology.

WebID-TLS is based on TLS client certificates, allowing the protocol to work on nearly all browsers [12]. Although support for client certificates is already built into modern web browsers, the user interfaces for this feature is inconsistent between browsers, and in many cases is complex, overloading the user with technical details that few understand. WebID-TLS developers are entirely

*COPELABS - Association for Research and Development in Cognition and People-centric Computing. Building U First Floor, University Lusofona Campus. Campo Grande 388, 1749-024 Lisboa

dependent on browser vendors and getting browser vendors to agree on new features is a long process. In addition, some of the successful web technologies have evolved outside of the web browser.

The WebID-TLS protocol, by placing public keys in a public profile, makes it is straightforward to deactivate a compromised identity certificate, but requires that relying parties communicate with the server to verify certificates, revealing information about the identity owner's activities. Besides, users may choose to locate their WebID profile at a URI for which they do not have full control. The issue of how a user controls their identity credentials is often the weakest link in a security system. Users may choose weak passwords, share or lose private keys, etc. We may argue that users worried about privacy would host their WebID profile only on a self-hosted system that they can control (such as a home server). In that case, having all identification requests recorded makes it much simpler for a user to monitor suspicious activity patterns. The counter-argument is that very few average users have the desire or technical skills to set up and run their own servers. As a result, it seems that mainstream adoption of WebID will be dependent on hosting service providers, and in that case, the traceability baked into the WebID protocol may become a potential privacy concern.

The WebID is based on non-permanent URI identifiers and is subject to recycled identifiers usage if a profile is hosted by a third-party service. Since WebID users are strictly identified by their profile URIs, if Alice's profile host discard her identifier after she closes her account, all Bob needs to do is to make a new WebID profile at the same URI to gain access to any web accounts still linked with that WebID. It is not a requisite to have Alice's private key, once Bob has replaced the public keys in the WebID profile with his own.

This topic is amplified if an organization or person is hosting their own profile server under their own domain name, and for some reason releases the domain. In that case, whoever takes possession of the domain name, gains effective control over all the identities formerly related with the domain. In fact, the WebID specifications rely on a central component, the DNS, a central point of failure. This centralization generates vulnerability.

3. THE NAMECOIN

The Namecoin (NMC) [13], the first fork from Bitcoin (BTC), is a cryptocurrency that implements a decentralized namespace and its own blockchain transaction database [14]. It offers the same features as the BTC, with the addition of a name/value store that can be used to maintain arbitrary data such as users' online identities, or the censorship-resistant top level Dot-Bit (.bit) domain, which is independent of ICANN. An identity can be composed of an e-mail addresses, namecoin / bitcoin addresses, gpg key (fingerprint), and other additional fields. All NMC records and transactions are protected using strong Elliptic Curve Cryptography, therefore making it appropriate for secure distributed credential storage and decentralized identity management. In the scenario presented in this paper, the NMC identity will contain the WebID URI, and the subsequent proposed authentication mechanism is described in section 5.

4. THE INTERPLANETARY FILE SYSTEM

In this approach, the InterPlanetary File System (IPFS) [15] is used to publish the WebID profile. The IPFS is a novel P2P version-controlled file system, that implements a content-addressed block storage, where all data is modeled as a generalized Merkle DAG. Every computer running IPFS is a node in a "swarm" of peers, communicating through an incentivized block exchange, therefore avoiding single points of failure.

IPFS creates a self-certifying namespace based on a distributed hashtable, thereby providing a content-addressed block storage model, with content-addressed hyper links. The significant change from more conventional access paradigms is that IPFS is searched for content, while HTTP, for instance, is searched for location. Instead of querying a centrally-controlled location for the WebID profile, a distributed network of computers is queried for the content that is supposed to be in the profile. When a file id.rdf representing the user profile is added to IPFS, it is allocated a new "name", which is represented by its hash, e.g.:

```
QmS9MpEspg1UPGiNhnZ5Wa7JSicfBotJzDk7xMFh7QA4x6
```

That hash is guaranteed by an irreversible cryptographic function to represent the contents of the file. If the file is changed, the hash becomes invalid. When the IPFS network is queried for that hash, it efficiently discovers the nodes that have the data using a DHT (Distributed Hash Table) algorithm, retrieves it, and verifies that the hash corresponds to the correct data.

The IPFS hash represents immutable data and encourages data persistence, but a mechanism is still needed to find the latest hash representing the user profile. This is possible by storing a reference to an IPFS hash under the namespace of a peer ID (node public key), so that the content has a name attached to it, and changes can be republished using the same name. The following example represents the IPFS content-address hyper link to the id.rdf file, where the hash corresponds to a peer ID:

```
/ipns/QmPtGiNVxEzSRTGxfjgxLpRWttx6nbF1EqFs7TFgcc9sNA/id.rdf
```

This content-address hyper link type is proposed as a new WebID profile URI based on IPFS. By using this URI type, assigning a name to a mutable (changeable) profile becomes possible, therefore solving the previously mentioned recycled identifiers problem, since a permanent URI identifier is now used.

Since the IPFS implementation provides an HTTP gateway, the content-address hyper link will have the following form:

```
http://gateway.example.com/ipns/QmPtGiNVxEzSRTGxfjgxLpRWt
tx6nbF1EqFs7TFgcc9sNA/id.rdf
```

This way an URI equivalent to the WebID profile URI is obtained, as described in the W3C WebID specifications.

5. DECENTRALIZED AUTHENTICATION

The concept of tokens containing claims is applied to provide a way for clients to authenticate every request without having to maintain a session or repeatedly send login credentials.

Claims are individual statements about the subject made by the issuer. A JSON Web Token (JWT) is a standard means of representing those claims. In a JWT, a claim is encoded as a JSON object that is digitally signed and optionally encrypted. The JWT is self-descriptive, which means that it contains all necessary information about the token itself and consists of three main components: a header object, a claims object, and a signature. These three components are encoded using base64, then concatenated with periods as separators.

In a common JWT pattern, the client code merely stores the token, it does not create it. The server creates the JWT on valid authentication (e.g. username & password login), adds custom claims and expiry limit and returns it to the client. The client just stores and sends the JWT back to the server which validates the

token by decoding it with a secret key. The client would store the JWT in a cookie or local storage.

However, in its standard form, a JWT does not provide a mechanism for end to end authentication. One of the missing blocks includes key distribution. This implementation proposes a solution for this problem, by using the Namecoin blockchain as a distributed credential storage in order to verify the JWTs.

Managing private keys securely is complex [16]. However this complexity is already being addressed in BTC. With BTC, private keys provide control over digital money – no one can move it without the owner's consent. In the same way, NMC private keys provide control over digital identities - no entity or individual can change other users public data, use their usernames or control the release of their private data without their consent. NMC addresses are directly derived from the same elliptic-curve cryptography standards used by BTC [17], and transactions are authenticated using digital signatures. The signatures and public keys are published as part of the publicly available blockchain to prevent double-spending.

Clients authenticate with their NMC identity, which means that they need to prove to an application/service that they control this specific NMC identity. They do so by signing a token, and therefore securely linking all related settings and data to a session. This way, the only information that clients need to disclose to a third party is their NMC identity. The proof-of-concept developed so far currently uses the NMC blockchain, but other decentralized key-value store solutions may be used.

Using the NMC virtual currency, names corresponding to online identities are registered with the decentralized NMC network. The client uses a name in a form "id/username" and the server a "d/domain" (.bit).

The WebID profile document is published in the IPFS P2P file system and contains a description of an NMC wallet address, which is associated to the WebID by using the **cc:wallet** property from the ontology for cryptocurrencies [18]. Figure 1 represents a sample WebID profile document published in the IPFS, containing a description that relates a person to a NMC wallet address.

```
@prefix : <http://www.w3.org/ns/auth/cert#>
@prefix <http://www.w3.org/2001/XMLSchema#>
@prefix foaf: <http://xmlns.com/foaf/0.1/>
@prefix rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix cc: <https://w3id.org/cc#>

<#me> a foaf:Person;
  foaf:name "Bob";
  foaf:knows <https://alice.example.edu/card#i>;
  cc:wallet "namecoin:N2pGWAh67TWpWmEFrFxsRQkQubbczJSKi8";
```

Figure 1: WebID profile containing an NMC wallet address

The user NMC identity contains the WebID URI for that profile. Figure 2 represents a sample NMC profile document (in JSON format) containing an IPFS permanent URI that represent the WebID URI.

```
{"name":"Bob","uri":
"/ipns/QmPtGiNVxEzSRTGxfjgxLpRWttx6nbF1EqFs7TFgcc9sNA/id#me"
}
```

Figure 2: NMC user profile containing the WebID URI

The relation between the WebID URI and the WebID profile document is illustrated in figure 3.

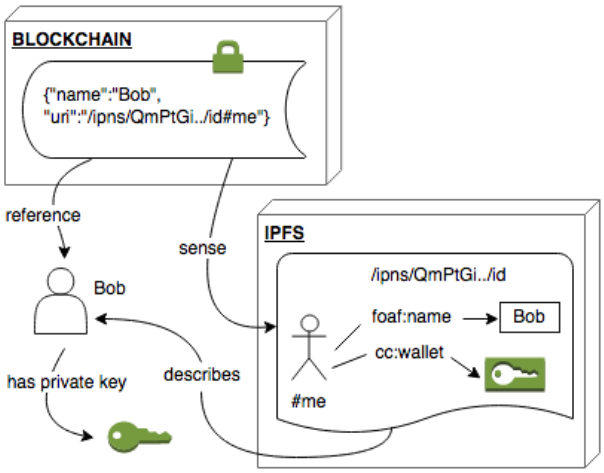

Figure 3: The WebID URI and WebID profile relation

The implemented authentication scheme is illustrated in figure 4. In this scenario, a JWT is signed using elliptic-curve asymmetric encryption. This means the token creators need a private and public key pair. Only the token creator should have access to the private key. The public key needs to be accessible to the receiver of the token. Moreover, the ownership of a name is associated with a possession of a corresponding private key. The client uses the private key to generate a self-signed JWT, adds custom claims and expiry date and sends them to the server. Since the password is replaced by a digital signature, this scheme does not involve server account creation and user credentials management, thus significantly increasing the security level of the exchange. The server confirms that the JWT was signed by the private key associated to the token issuer (id/username) and the token expiry limit. The client may verify the legitimacy of the server if the server uses a .bit domain. This novel JWT identity management pattern allows authentication outside the control of any single entity.

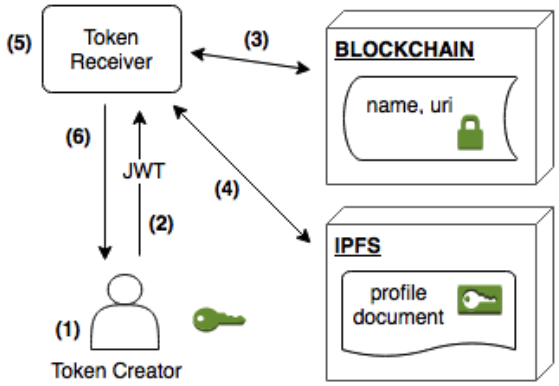

Figure 4: JWT decentralized interplay pattern

Step 1 - A client creates a JWT token and encodes a series of claims in the claims object:

- The "**iss**" (issuer) claim contains the NMC public online identity, prefixed with "id/" (e.g. id/Bob) that identifies the principal that issued the JWT. Use of this claim is mandatory.

- The "**exp**" (expiration time) claim contains the time after which the JWT must not be accepted for processing. Use of this claim is optional.
- Other custom claims can be added.

The JWT header declares that the encoded object is a JSON Web Token that is signed using a cryptographic algorithm:

```
{"typ":"JWT","alg":"ES256"}
```

The "typ" (type) header parameter is used to declare that the data structure is a JWT. The "alg" (algorithm) header parameter value "ES256" is used to indicate that the JWT crypto segment was signed with the Elliptic Curve Digital Signature Algorithm (ECDSA) using the P-256 curve and the SHA-256 hash algorithm (the digital signature scheme of NMC).

The client will then use the NMC private key associated with the issuer claim in order to sign the token on behalf of their identity.

If the receiver makes use of a .bit/NMC domain the client may verify the legitimacy of the receiver. The NMC decentralized blockchain performs validations equivalent to those of a Certification Authority (CA), which means that in this context, a CA is not necessary [19].

Step 2 - The HTTP standard "Authorization" request-header field is used to send the JWT.

Step 3 - The receiver looks at the issuer token claim, retrieves the WebID URI associated with the issuer identity from the NMC blockchain and associated JSON.

Step 4 - The receiver retrieves the WebID Profile associated with the issuer identity from the IPFS.

Step 5 - The receiver acts as a JWT validator by checking the JWT signature. For example, if the following token is sent:

```
{"typ":"JWT","alg":"ES256"}.{"iss": "id/Bob"}.[signature]
```

The receiver validates that the token was signed by the NMC address private key associated with id/Bob. The private key is mathematically related to the NMC addresses generated for the NMC wallet that owns the id/Bob. The NMC address is a 160 bit hash of the public portion of a private/public ECDSA key pair.

This process does not reveal the NMC private key to the receiver or anyone else, but is used to prove that the client is indeed the owner of the NMC public online identity.

Step 6 – The receiver sends response to the client.

6. SUMMARY

In this paper, an authentication scheme based on WebID, JSON Web Tokens (JWT) and the blockchain was presented. An alternate decentralized identity management implementation, based on the Namecoin blockchain, is described, on which the blockchain is used to register the user's WebID URI and domain names. To achieve this goal, a new WebID profile URI based on IPFS P2P file system is created, allowing the use of permanent URI identifiers. The JSON Web Token concept is used as a way to authenticate every request, without having to maintain a session or repeatedly send login credentials.

The proposed authentication scheme is outside the control of any single entity and can be applied to a number of different use cases. The security of the proposal is tied to the security of the blockchain, provided that the necessary precautions are taken in the generation, storage and use of the blockchain private keys.

7. REFERENCES

[1] Weitzner, D. June 2007, Whose name is it anyway? decentralized identity systems on the web. IEEE Internet Computing.

[2] Zysking, G. and Nathan, O. May 2015, A. Pentland, Decentralizing privacy: using blockchain to protect personal data. 36th IEE Symposium on Security and Privacy Workshops (SPW).

[3] WebID Incubator Group, WebID specifications, https://www.w3.org/2005/Incubator/webid/spec/

[4] Jones, M., Bradley, J., Sakimura, N. et all. May 2015, JSON Web Token (JWT), RFC7519. Internet Engineering Task Force (IETF).

[5] Clippinger, J. 2015, A new kind of social ordering: self-sovereignty, autonomous trust and P2P parity. ID3

[6] ICAAN, https://www.icann.org/

[7] OpenNIC, https://www.opennicproject.org/

[8] Deegan, T., Crowcroft, T. and Warfield, A. October 2005, The main name system: an exercise in centralized computing. ACM SIGCOMM Computer Communication Review, Volume 35, Number 5.

[9] RDF 1.1 Primer, http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/

[10] FOAF Vocabulary Specification 0.99, http://xmlns.com/foaf/spec/

[11] Internet X.509 Public Key Infrastructure Certificate and CRL Profile, Internet Engineering Task Force (IETF), http://www.ietf.org/rfc/rfc2459

[12] Hackett, M. and Hawkey, K. May 2012, Security, privacy and usability requirements for federated identity. Web 2.0 Security & Privacy 2012, IEEE Computer Society Security and Privacy Workshops.

[13] Kalodner, H., Carlsten, M., Ellenbogen, P., Bonneau, J. and Nayaranan, A., June 2015, An empirical study of namecoin and lessons for decentralized namespace design. WEIS 2015, 14th Annual Workshop on the Economics of Information Security.

[14] Bonneau, J., Miller, A., Clark, J., Nayaranan, A., Kroll, J. and Felten, E. May 2015, Sok: research perspectives and challenges for bitcoin and cryptocurrencies, 2015 IEEE Symposium on Security and Privacy (SP).

[15] Benet, J. July 2014, IPFS – Content addressed, versioned, P2P file system. Cornell University Library, http://arxiv.org/abs/1407.3561v1

[16] Dolev,D. And Yao, A. March 1983, On the security of public key protocols. IEEE Transactions on Information Theory, Volume:29 (2): 198–208.

[17] Bos, J., Halderman, J., Heninger, N., Moore, J., Naehrig, M. and Wustrow, E. March 2014, Elliptic curve cryptography in practice. 18th International Conference on Financial Cryptography and Data Security (FC 2014).

[18] Ontology for cryptocurrencies, https://w3id.org/cc

[19] Garman, C., Green, M. and Miers, I. 2014, Decentralized anonymous credentials, network and distributed system security. NDSS Symposium.