BFTKV DID Method Specification

STATUS: Working Draft, September 12, 2017

*Note: the definitions of the terms written in **bold** are parallel with the Terminology section of the DID Specification.*

ABSTRACT OF THE DID SPECIFICATION

DIDs (decentralized identifiers) are a new type of identifier intended for verifiable digital identity that is “self-sovereign”, i.e, fully under the control of the **identity owner** and not dependent on a centralized registry, identity provider, or certificate authority. DIDs resolve to **DDOs** (DID descriptor objects)—simple JSON documents that contain all the metadata needed to prove ownership and control of a DID. Specifically, a DDO contains a set of **key descriptions**—machine-readable descriptions of the identity owner’s public keys—and a set of **service endpoints**—resource pointers necessary to initiate trusted interactions with the identity owner. Each DID uses a specific **DID method**, defined in a separate **DID method specification**, to define how the DID is registered, resolved, updated, and revoked on a specific distributed ledger or network.

ABSTRACT OF THIS SPECIFICATION

This document defines the BFTKV DID method specification that includes the BFTKV DID scheme and operations. Moreover, security and privacy issues are discussed in regard to DID clients, developers and owners.

BFTKV is a Byzantine fault tolerant key value storage that uses

- PGP’s web of trust mechanism to build trust relationships between entities,
- b-masking quorums to provide Byzantine fault tolerance and
- Quorum certificates to restrict access to write operations

Implementation details and more information about the system is accessible at the following URL: https://www.github.com/yahoo/bftkv

Table of Contents

1. Introduction and Background

This document defines the BFTKV DID method specification adhering to the defined requirements in the DID specification.

BFTKV is a Byzantine fault tolerant key value storage that incorporates three concepts in its core:

- Byzantine Quorum Systems
- PGP's Web of Trust Mechanism
- Quorum Certificates

PGP's Web of Trust mechanism is a way to build trust between entities without a central authority. BFTKV takes advantage of this mechanism to build quorums and provide Byzantine fault tolerance. In addition, to prevent unauthorized write operations, BFTKV uses Quorum Certificates.

2. BFTKV DID Scheme

BFTKV DID scheme adheres to the following ABNF:

```
bftkv-did           = "did:bftkv:" PGPFingerprint
PGPFingerprint      = 40*(char)
char                = ALPHA / DIGIT
```

`PGPFingerprint` is the fingerprint of the public PGP key. We use the term "fingerprint" as defined by PGP.

3. BFTKV DID Operations

Prerequisites

BFTKV consists of nodes that can dynamically join and leave the system (in addition to consistently available nodes). Since the system is based on PGP's web of trust mechanism, **owner**s that will store/publish DDOs SHOULD create a PGP key pair (public and private) and use these keys for BFTKV DID operations.[1]

After the key pair is created, the keys should be marked as trusted (signed) by a quorum of the members of the BFTKV system. To achieve this, the public PGP key is sent to the system and

[1] In the future, BFTKV MAY provide other means of accessing BFTKV DID operations, such as via HTTP.

the signatures are gathered by the client. After this step, the client can join the system and perform BFTKV DID Operations detailed below.

3.1. Create

BFTKV utilizes the signature field to store the signature over the DDO template part (i.e., @context, id, and owner), and BFTKV node key (verification key). The “signatureValue” field is a base64 encoded string which contains those information. It is opaque from DDI applications. BFTKV provides a verification method.

BFTKV DDO Template Example:

```
{
  “@context”: “https://example.com/did/v1”,
  “Id”: “did:bftkv:uiduiduid”,
  “Owner”:[{
    “Id”: “did:bftkv:uiduiduid#key1”,
    “Type”: [“cryptographickey”],
    “Expires”: “2017-02-08T16:02:20Z”,
    “publickeyBase64”: “…”
  }],
}
```

Complete DDO Example:

```
{
  < same as the template >

  “Signature”: {
    “Type”: “BFTKVsignatures”,
    “Creator”: “http://bftkv_123:7501”,
    “signatureValue”: “...”
  }
}
```

Create operation is done by using BFTKV’s Write function and setting DID-DDO as the key-value pair to be submitted.

```
bftkv_did.Write(DID, DDO)
```

3.2. Read/Verify

A BFTKV DID record can be looked up directly by the DID using a standard BFTKV API that simply takes the DID and will return the DDO.

```
DDO := bftkv_did.Read(DID)
```

BFTKV guarantees data integrity and freshness of the value.

3.3. Update

BFTKV DID records can be updated directly using a standard BFTKV API that simply takes the DID and the updated DDO.

```
bftkv_did.Write(DID, DDO)
```

BFTKV guarantees only the original writer of DDI can update the same DDI with a new value (updated DDO). See the BFTKV design for details.

3.4. Delete/Revoke

Delete/Revoke operation for a DID can be achieved by writing null to DDO.

4. Security Considerations

4.1. Transport security

BFTKV uses PGP message encryption scheme to protect the message from: eavesdropping, replay, message insertion, deletion, modification, and man-in-the-middle attacks.

4.2. Potential denial of service attacks

The transport security mitigates DoS type attacks in some extent. Also, TOFU (Trust On First Use) will prevent massive unauthorized writes.

4.3. Residual Risks

The quorum system mitigate errors at some nodes. See the BFTKV design for details.

4.4. Integrity protection and update authentication

BFTKV has two update modes: TOFU and WriteOnce. With the TOFU policy, the key/value slot can be modified by the original writer. BFTKV uses quorum certificate to recover from the key-loss situation. See the BFTKV design for details.

4.5. Burdens

Known as “load” in the quorum system. BFTKV uses two separated quorum systems to minimize the load at each node: authority and r/w quorums. See the BFTKV design for details.

4.6. Uniqueness of DID

BFTKV uses the PGP fingerprint as “specific-idstring”, which is the hash value of the public key. As long as private / public key pairs are generated uniquely the fingerprint has to be unique. So is DID.

4.7. Private key management

BFTKV uses “GPG secring” to keep private keys, which MUST be securely stored at each BFTKV node. **Owners** are responsible for keeping their private keys secure.

4.8. Identity Owner Authentication and Verifiable Claims

BFTKV can utilize “email proof” when the BFTKV ID is bound to the email address. The actual method is out of scope for BFTKV.

5. Privacy Considerations

BFTKV DID with its Security Considerations, aims to provide a usable and secure environment for DIDs, owners and other system entities. In addition to this, BFTKV DID does not store any personally-identifiable information. However, DID owners might need to take precautions to reduce the correlation risks. These include sharing a private DID for each relationship, providing unique information in each DDO, as suggested by the DID Specification.