DWN Decentralized Identity Authenticated Key Exchange Protocol

@

Abstract

Decentralized Identifier (DID)-based privacy is a technique for enabling secure and private communication between parties using DID's as the primary means of identification. DID's are decentralized, self-sovereign identifiers that allow individuals and organizations to securely and privately store and manage their own identity information.

One potential approach to implementing DID-based privacy is through the use of Authenticated Key Exchange (AKE) protocols. AKE protocols enable two parties to establish a secure, authenticated channel for communication by exchanging keys and verifying the authenticity of the keys through the use of a trusted third party or a secure protocol.

In the context of DID-based privacy, AKE protocols can be used to establish a secure, authenticated communication channel between parties using DID's as the primary means of identification. This allows parties to communicate in a secure and private manner, while also ensuring that the identity of the parties is verifiable and cannot be spoofed.

Overall, the use of DID's in combination with AKE protocols provides a promising approach to enabling secure and private communication between parties, while also enabling decentralized and self-sovereign identity management.

Introduction

Decentralized identity refers to a system in which individuals and organizations can securely and privately store and manage their own identity information, rather than relying on centralized identity providers such as governments or large tech companies. This allows individuals to have more control over their personal data and ensures that their identity information is not subject to the risks and vulnerabilities associated with centralized systems, such as data breaches or unauthorized access.

There are several key benefits to decentralized identity systems. First, they provide greater privacy and security for individuals, as their identity information is not stored in a centralized location that could potentially be hacked or accessed without their consent. Additionally, decentralized identity systems can be more inclusive, as they allow individuals who may not have access to traditional identity verification processes, such as those without government-issued identification, to create and manage their own identity.

Decentralized identity systems can be implemented using various technologies, such as blockchain, distributed ledger technology, and self-sovereign identity frameworks. These technologies enable the creation of decentralized identity networks, in which individuals and organizations can securely store and manage their own identity information and share it with others as needed.

In conclusion, decentralized identity systems offer a more secure and private alternative to traditional centralized identity systems. By empowering individuals to control and manage their own identity information, decentralized identity systems can provide greater privacy and security for individuals, as well as enable more inclusive identity verification processes.

Decentralized identity systems can be used in conjunction with InterPlanetary File System (IPFS) to store and manage identity information in a decentralized manner. IPFS is a distributed file system that allows for the storage and sharing of data in a peer-to-peer network, rather than on a central server. This means that data stored on IPFS is decentralized and can be accessed by anyone on the network.

In the context of decentralized identity, IPFS can be used to store identity information, such as personal documents, in a decentralized manner. This allows individuals to have greater control over their identity information and ensures that it is not subject to the risks and vulnerabilities associated with centralized systems. Additionally, IPFS can be used to share identity information with others in a secure and private manner, as the data is encrypted and stored in a decentralized manner. This can be particularly useful for individuals and organizations seeking to verify their identity online, as it allows them to securely share their identity information with others without relying on centralized identity providers.

How we apply the protocol

There are several different types of Authenticated Key Exchange (AKE) protocols, each with its own unique mathematical formulation. Here is an example of a basic AKE protocol, known as the ‘Diffie-Hellman’ key exchange, which uses modular arithmetic to enable the exchange of a shared secret key between two parties:

1. Alice and Bob agree on a prime number p and a primitive root g.
2. Alice selects a private key and computes A = g^a mod p. Alice sends A to Bob.
3. Bob selects a private key b and computes B = g^b mod p. Bob sends B to Alice.
4. Alice computes the shared secret key s = B^a mod p.

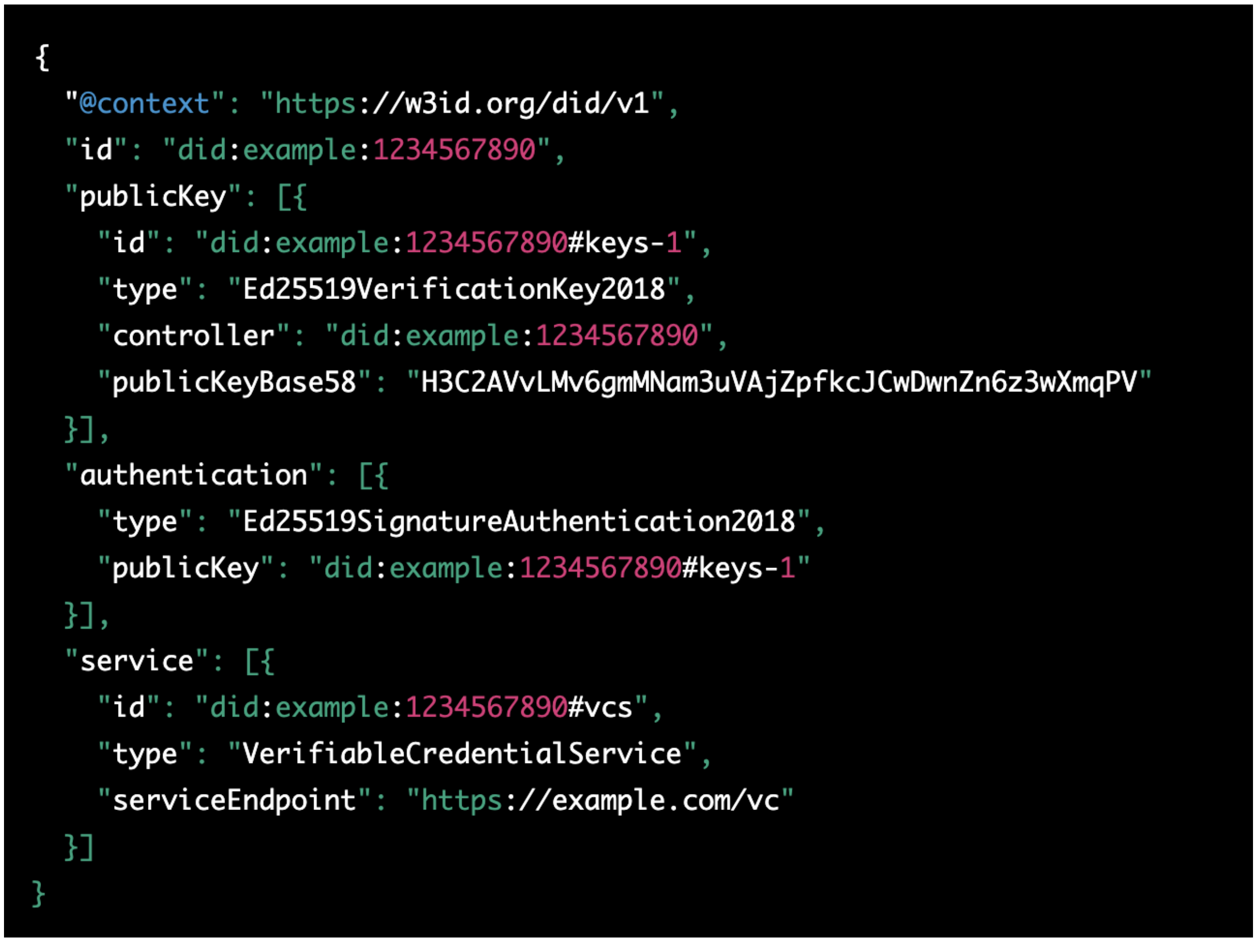
5. Bob computes the shared secret key s = A^b mod p.

In this protocol, Alice and Bob are able to exchange a shared secret key s without revealing their private keys a and b to each other. This allows them to establish a secure, authenticated communication channel.

There are also more advanced AKE protocols that use more complex mathematical formulations, such as elliptic curve cryptography and zero-knowledge proofs. These protocols can provide additional security and privacy features, such as resistance to key substitution attacks and the ability to prove knowledge of a secret without revealing it.

Defining an example of a DID document in JSON representation

A Decentralized Identifier (DID) document is a JSON-formatted document that contains



```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:example:1234567890",
  "publicKey": [{
    "id": "did:example:1234567890#keys-1",
    "type": "Ed25519VerificationKey2018",
    "controller": "did:example:1234567890",
    "publicKeyBase58": "H3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
  }],
  "authentication": [{
    "type": "Ed25519SignatureAuthentication2018",
    "publicKey": "did:example:1234567890#keys-1"
  }],
  "service": [{
    "id": "did:example:1234567890#vcs",
    "type": "VerifiableCredentialService",
    "serviceEndpoint": "https://example.com/vc"
  }]
}
```

information about a DID, including its unique identifier, public keys, and service endpoints. Here is an example of a DID document in JSON representation:

In this example, the DID document is associated with the DID:
”did:example:1234567890" and contains a public key, an authentication method, and a service endpoint. The public key is used for verifying the authenticity of the DID, and the authentication method specifies how the DID can be used for signing and verifying messages. The service endpoint is a URL that can be used to access services related to the DID.

DID documents can also contain other types of information, such as proof purposes and proof types, which specify how the DID can be used for proof of ownership or other types of verifications. Additionally, DID documents can include additional public keys, authentication methods, and service endpoints as needed.

Preliminaries

Verifiable Credentials

A DID verifiable credential is a digital document that is issued by a trusted entity and verifiable using a Decentralized Identifier (DID). Verifiable credentials are used to represent and prove claims about an individual or organization, such as their identity, qualifications, or membership in a particular group.

A DID verifiable credential contains three main components:

1. The credential subject: the individual or organization that the credential is about.
2. The credential issuer: the trusted entity that issued the credential.
3. The credential claims: the specific claims being made about the credential subject, such as their name, job title, or educational qualifications.
4.

In order for a verifiable credential to be considered trustworthy, it must be issued by a trusted entity and signed using a private key that is associated with a DID. This allows the credential to be verified by anyone using the corresponding public key, which is stored in the DID document associated with the DID.

Verifiable credentials can be used in a variety of contexts, such as to prove identity, verify qualifications, or access certain services. They provide a secure and private alternative to traditional forms of verification, such as government-issued identification, as they allow individuals to control and manage their own identity information.

Zero-Knowledge Proofs

A Zero-Knowledge Proof (ZKP) is a cryptographic protocol that allows one party (the prover) to prove to another party (the verifier) that they possess certain knowledge or information, without revealing the actual knowledge or information. ZKPs are used to establish the authenticity of a statement or claim without revealing any unnecessary information.

Here are some examples of statements that could be proven using a ZKP:

- "I know the secret password to a particular system, but I will not reveal it to you."
- "I am over the age of 18, but I will not reveal my actual age."
- "I have a degree from a particular university, but I will not reveal my transcripts or any personal information."
-

In each of these examples, the prover is able to prove that they possess certain knowledge or information without revealing the actual knowledge or information itself. This allows the prover to maintain their privacy while still establishing the authenticity of their claim.

There are various types of ZKPs, each with its own specific mathematical formulation. Some common types of ZKPs include interactive proof systems, non-interactive proof systems, and proof of knowledge. These protocols can be used in a variety of contexts, such as to verify the authenticity of digital signatures, to enable secure and private communication, and to enable decentralized and self-sovereign identity management.

Asymmetric Cryptography and ECDSA

$$y2 = x3 + ax + b$$

Asymmetric cryptography, also known as public-key cryptography, is a type of cryptographic system that uses a pair of keys – a public key and a private key – to encrypt and decrypt messages. In this system, the private key is kept secret by the owner and is used to decrypt messages that have been encrypted using the corresponding public key. The public key, on the other hand, is made available to anyone and is used to encrypt messages that can only be decrypted using the corresponding private key.

One example of an asymmetric cryptographic algorithm is the Elliptic Curve Digital Signature Algorithm (ECDSA). ECDSA is a signature algorithm that uses elliptic curve cryptography to generate a digital signature. It is widely used in various applications, such as secure communication, digital identity, and blockchain technology.

Here is an example of how ECDSA works:

1. Alice wants to send a message to Bob, but wants to ensure that the message is secure and cannot be read by anyone else.
2. Alice generates a pair of keys – a public key and a private key – using ECDSA.
3. Alice shares her public key with Bob, but keeps her private key secret.
4. Bob wants to send a message to Alice, so he uses Alice's public key to encrypt the message.
5. Alice receives the encrypted message and uses her private key to decrypt it.

In this example, Alice is able to securely send a message to Bob using ECDSA. The message is encrypted using Alice's public key, which can only be decrypted using her private key. This ensures that the message is secure and can only be read by Alice.
Here is an example formula that describes the ECDSA algorithm:

Signature Generation:

1. Select a random integer k from $[1, n-1]$, where n is the order of the elliptic curve.
6. Compute the point $R = kG$, where G is the generator point of the elliptic curve.
7. Compute $r = x(R) \bmod n$. If $r = 0$, go back to step 1.
8. Compute $s = k^{-1}(H(M) + dr) \bmod n$, where H is a hash function, M is the message being signed, and d is the private key. If $s = 0$, go back to step 1.
9. The signature is (r, s).

Signature Verification:

To verify a signature using the Elliptic Curve Digital Signature Algorithm (ECDSA), the following steps can be followed:

1. Obtain the public key Q and the signature (r, s).
2. Compute $w = s^{-1} \bmod n$, where n is the order of the elliptic curve.
3. Compute $u1 = H(M)w \bmod n$ and $u2 = rw \bmod n$, where H is a hash function and M is the message being verified.
4. Compute the point $R = u1G + u2Q$, where G is the generator point of the elliptic curve.
5. If $x(R) \bmod n$ is equal to r, the signature is valid. Otherwise, the signature is invalid.
6.

In this process, the verifier uses the public key Q and the signature (r, s) to reconstruct the point R on the elliptic curve. If the x-coordinate of R ($x(R)$) is equal to r, the signature is considered valid. This is because the signature was created using the private key d, which corresponds to the public key Q. If the x-coordinate of R does not match r, the signature is considered invalid, indicating that the message has been tampered with or that the signature was not created using the correct private key.

Here is the same process expressed using mathematical notation:

Given:

- Public key Q
- Signature (r, s)

Compute:

- w = s^-1 mod n
- u1 = H(M)w mod n
- u2 = rw mod n
- R = u1G + u2Q
-

If x(R) mod n = r, the signature is valid.
Else, the signature is invalid.

In this process, the verifier uses the public key Q and the signature (r, s) to reconstruct the point R on the elliptic curve. If the x-coordinate of R (x(R)) is equal to r, the signature is considered valid. This is because the signature was created using the private key d, which corresponds to the public key Q. If the x-coordinate of R does not match r, the signature is considered invalid, indicating that the message has been tampered with or that the signature was not created using the correct private key.

DWN Authentication & Messaging Protocol

Proposed authentication protocol using decentralized identity, asymmetric cryptography, and zero-knowledge proofs:

1. Alice and Bob agree to use a decentralized identity system for authentication or messaging.
10. Alice and Bob each create a DID and a corresponding pair of keys – a public key and a private key – using asymmetric cryptography.
11. Alice and Bob exchange their DID's and public keys.
12. Alice wants to authenticate herself to Bob. She creates a message and signs it using her private key.
13. Alice sends the signed message and her DID to Bob.
14. Bob uses Alice's DID to retrieve her public key from the decentralized identity system.
15. Bob uses the public key to verify the authenticity of the signed message.
16. If the message is authentic, Bob can be confident that it was sent by Alice.
17. To further enhance the security of the authentication process, Alice and Bob can use a zero-knowledge proof to prove the authenticity of their DID's without revealing any unnecessary information.
18.

In this protocol, Alice and Bob are able to authenticate each other or exchange secure and private messages using their DID's and asymmetric cryptography. The use of a decentralized identity system and a zero-knowledge proof allows them to securely and privately establish their identity without revealing any unnecessary information. This protocol could be used in a variety of contexts, such as to enable secure communication or to access secure systems or services.

Decentralized Web Node Service (DWN)S

The DWN (decentralized web node) is connected to a decentralized web network and is used to store and transmit data in a decentralized manner.

Decentralized web networks, also known as distributed networks or peer-to-peer networks, are networks that are not controlled by a central authority and are instead made up of a large number of nodes that are connected and communicate with each other directly. Decentralized web networks are designed to be resistant to censorship, downtime, and data breaches, as there is no central point of control or failure.

Decentralized web nodes can be used to store and transmit a variety of types of data, such as documents, images, videos, and websites. They can also be used to run decentralized applications (dApps) and to participate in decentralized governance processes.
Decentralized web nodes are an important part of the decentralized web, as they enable the distributed storage and transmission of data and enable decentralized systems to function. By participating in a decentralized web network, individuals and organizations can contribute to the decentralization of the internet and help to create a more open, secure, and transparent online environment.

An decentralized web node can serve web applications from a user's secure node by hosting the web application on the decentralized web node and making it available to users through the decentralized web network.

To do this, the user would first need to create their identity using DWNs which will set up a decentralized web node on their smart device using a L2 application integrated with DWN.

This will involve the users (DWN)S joining a decentralized web network, providing a peer-to-peer network and decentralized storage network.

Once the identity is setup the decentralized web node is secure and available, the user can run self hosted applications on the node. These applications can then be accessed by users through the decentralized web network by using a decentralized web address on the decentralized web gateway.

By hosting a web application on a users decentralized web node, the user can ensure that the application is served from a secure and private environment. The decentralized nature of the web node also ensures that the application is resistant to censorship, downtime, and data breaches, as there is no central point of control or failure.

Overall, hosting web applications on decentralized web nodes allows users to securely and privately serve web applications to others through the decentralized web, while also contributing to the decentralization of the internet.

Personal identification information (PII)

PII data of participants in a carbon credit producing project could be secured using DWN decentralized identity as follows:

1. The carbon credit producing project could create a decentralized identity (DID) on the DWN platform for each participant in the project. The DID would act as a secure and verifiable digital identity for each participant, allowing them to prove their identity and access their carbon credit information.

2. The project could then use the DWN DID's to store and secure the PII data of each participant. This data could include information such as the participant's name, contact information, and any other relevant identifying information.

3. The PII data could be encrypted using the participant's DID, so that only the participant themselves would have access to it. This would ensure that the data is kept secure and private, and that it can only be accessed by the participant themselves or with their permission.
4. The project could also set up secure access controls for the PII data, using the DWN platform. This could include requiring two-factor authentication or other security measures to ensure that only authorized parties have access to the data.

By using decentralized identity to secure the PII data of participants in a carbon credit producing project, the project can ensure that the data is kept secure and private, while also allowing participants to access and manage their own data in a secure and verifiable way.