Enhancing Non-Repudiation Protocols with Knowledge Graphs

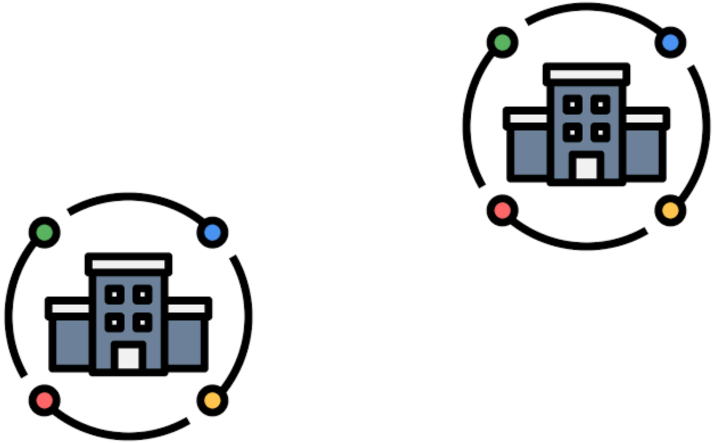
STSM Research Funded by DKG Cost Action (CA19134)

Biagio Boi
University of Salerno
bboi@unisa.it

DKGs for Non-Repudiation

Motivations

Let's imagine a dispute between Organization A and Organization B. Org. A states that Org. B accessed sensitive content at a given time, while Org. B refuses this statement.



DKGs for Non-Repudiation

Motivations

Let's imagine a dispute between Organization A and Organization B. Org. A states that Org. B accessed sensitive content at a given time, while Org. B refuses this statement.

Who's right? How this can be demonstrated?

DKGs for Non-Repudiation

Motivations

Let's imagine a dispute between Organization A and Organization B. Org. A states that Org. B accessed sensitive content at a given time, while Org. B refuses this statement.

Who's right? How this can be demonstrated?

Who is the judge?

DKGs for Non-Repudiation
Motivations

Non-repudiation is a security principle that ensures a party in a communication cannot deny the authenticity of their actions, such as sending a message or completing a transaction.

Non-repudiation in systems has always been a requirement for systems requesting a high level of security and auditability. These properties are particularly important for sensitive scenarios such as use cases 2.3 (legal data), 4.5 (healthcare), etc.

However, current non-repudiation protocols face critical challenges: there is no widely accepted standard for representing non-repudiation, particularly in **decentralized** environments, where two organizations communicate without having a **trust authority**.

DKGs for Non-Repudiation
Benefits

1. A standardized way for saving logs ensures consistent, **interoperable**, and tamper-proof records that can be easily verified and audited across different systems, independently from representation used (RDF, XML, JSON-LD, etc.)

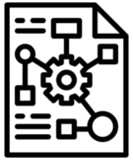
2. Signature applied over the graphs ensure **data integrity**, verifiability, and tamper-proof. Also in this case the signature is applied over the semantics rather than the single representation.

3. DKGs enables machine-to-machine (M2M) communication as well as user friendly visualization of relationships, by enabling seamless interaction and understanding of complex data connections, making the users more aware of actions performed.

DKGs for Non-Repudiation

Challenges

Application of DKGs for guaranteeing non-repudiation introduces huge advantages, anyway the implementation poses three main challenges.



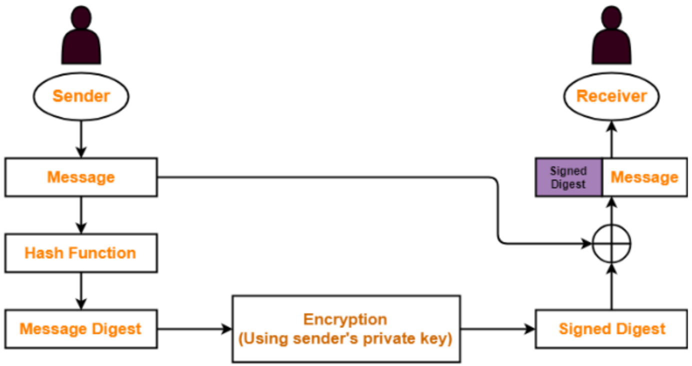
Data Canonization and Signature

Private Key management

Decentralized Communication

Data Canonization and Signature

Typical message signature involves the hashing of the message and signature applied over the resulting digest. When translating this to KG we need to take into account canonization of data.



In general the operation uses the RDF Dataset Normalization Algorithm transform an input document into its canonical form. The cannonical representation is then hashed and signed with a detached signature algorithm.

Data Canonization and Signature

Typical message signature involves the hashing of the message and signature applied over the resulting digest. When translating this to KG we need to take into account canonization of data.

In general the operation uses the RDF Dataset Normalization Algorithm transform an input document into its canonical form. The cannonical representation is then hashed and signed with a detached signature algorithm.

Canonized KG

```
<did:example:456> <https://example.org/examples#degree> _:c14n0 .
<http://example.gov/credentials/3732> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<https://example.org/examples#UniversityDegreeCredential> .
<http://example.gov/credentials/3732> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<https://www.w3.org/2018/credentials#VerifiableCredential> .
<http://example.gov/credentials/3732> <https://w3id.org/security#proof> _:c14n1 .
<http://example.gov/credentials/3732> <https://www.w3.org/2018/credentials#credentialSubject> <did:examp
<http://example.gov/credentials/3732> <https://www.w3.org/2018/credentials#issuanceDate> "2020-03-
10T04:24:12.164Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://example.gov/credentials/3732> <https://www.w3.org/2018/credentials#issuer> <https://example.com,
_:c14n0 <http://schema.org/name> "Bachelor of Science and Arts"^^<http://www.w3.org/1999/02/22-rdf-synta
_:c14n0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <https://example.org/examples#BachelorDegree>
_:c14n2 <http://purl.org/dc/terms/created> "2019-12-11T03:50:55Z"^^<http://www.w3.org/2001/XMLSchema#da
```

Private Key Management

Non-Repudiation algorithms rely Public Key Infrastructure (PKI) to guarantee signatures and checks over them. A secure storage is needed to preserve malicious usage of these keys.

Aries Askar is a secure storage and a key management service suitable for use with Hyperledger Aries agents and possibly other digital trust agents

It leverages biometric authentication to protect private keys and offers all the crypto-suite needed to sign and encrypt data.

Decentralized Communication

Decentralized communication is much more critical without a centralized authority responsible for manage certificates and create trust among partecipants.

DIDComm offers a strong security againist attackers, and it is also able to create a narrowed channel between communication parties.

It is based on Elliptic Curve Diffie Hellman (ECDH) exchange of XChaCha20Poly1305 key used for the encryption.

Decentralized Communication

It is based on Elliptic Curve Diffie Hellman (ECDH) exchange of XChaCha20Poly1305 key used for the encryption.

The public key, which are managed by users individually, are published in a DIDDocument under the control of the user itself.

```
{
  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:example:jklmnopqrstuvwxyz1",
  ...
  "keyAgreement": [
    {
      "id": "did:example:jklmnopqrstuvwxyz1#key-1",
      "type": "X25519KeyAgreementKey2019", // external (property value)
      "controller": "did:example:jklmnopqrstuvwxyz1",
      "publicKeyBase58": "9hFgmPVfmBZwRvFEyniQDBkz9LmV7gDEqytWyGZLmDXE"
    }
  ],
  ...
}
```

Use Case

Non-Repudiable Resource Request

A concrete application of the proposed protocol has been investigated during STSM period. Such a protocol has been applied to decentralized authentication, fully leveraging decentralized identifiers and KGs.

Parties are identified by their respective DIDs, each of them mapped into a DIDDocument.

Resolution
$DID_U = did{:}web{:}user \mapsto (g^u, pk(sk_U))$
$DID_A = did{:}web{:}app \mapsto (g^a, pk(sk_A))$
$DID_I = did{:}web{:}issuer \mapsto (g^i, pk(sk_I))$
$DID_V = did{:}web{:}verifier \mapsto (g^v, pk(sk_V))$

Table 1: Mapping of DIDs using Web method resolution.

Use Case

Non-Repudiable Resource Request

A concrete application of the proposed protocol has been investigated during STSM period. Such a protocol has been applied to decentralized authentication, fully leveraging decentralized identifiers and KGs.

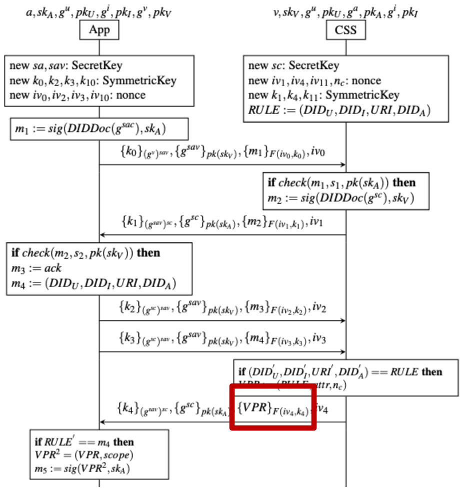
Resolution
$DID_U = did{:}web{:}user \mapsto (g^u, pk(sk_U))$
$DID_A = did{:}web{:}app \mapsto (g^a, pk(sk_A))$
$DID_I = did{:}web{:}issuer \mapsto (g^i, pk(sk_I))$
$DID_V = did{:}web{:}verifier \mapsto (g^v, pk(sk_V))$

Table 1: Mapping of DIDs using Web method resolution.

Use Case

Non-Repudiable Resource Request

The first step of the proposed protocol is the authentication request from the Org. 1 (taget) to the user (agent), which is intermediated by the Org. 2 (client).



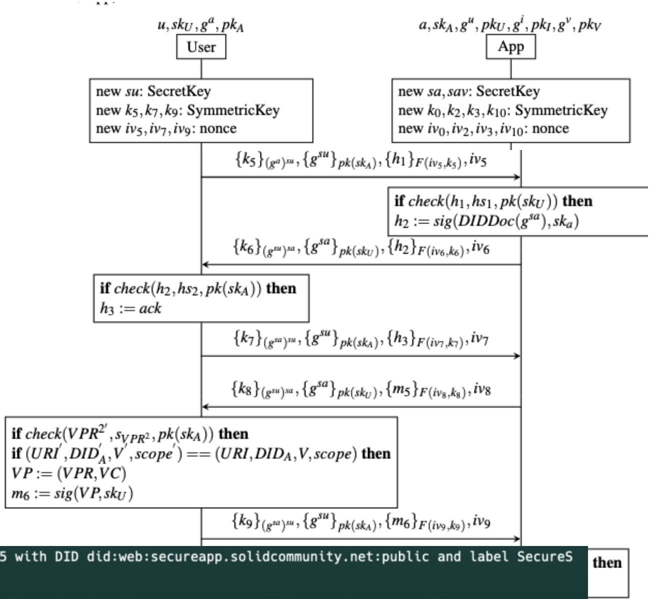

```json
{
  "@context": [
    "https://bboi.solidcommunity.net/public/schemas/2024/presexchange.jsonld"
  ],
  "type": [
    "VerifiablePresentationRequest"
  ],
  "presentation_definition": {
    "id": "32f54163-7166-48f1-93d8-ff217bdb0653",
    "input_descriptors": [
      {
        "id": "unisa_student",
        "name": "University of Salerno Demo",
        "purpose": "Demonstrate to be a student from the University of Salerno to access this POD",
        "constraints": {
          "fields": [
            {
              "path": [
                "$.credentialSubject.degree",
                "$.credentialSubject.claims.degree"
              ]
            }
          ]
        }
      }
    ],
    "format": {
      "ldp_vc": {
        "proof_type": [
          "JsonWebSignature2020",
          "Ed25519Signature2018",
          "EcdsaSecp256k1Signature2019",
          "RsaSignature2018"
        ]
      },
      "ldp_vp": {
        "proof_type": [
          "Ed25519Signature2018"
        ]
      },
      "ldp": {
        "proof_type": [
          "RsaSignature2018"
        ]
      }
    },
    "requestACP": {
      "type": [
        "ACPContext"
      ],
      "target": "http://localhost:3000/my-pod/test-folder/test-resource.txt",
      "owner": "did:web:raw.githubusercontent.com:biagioboi:CommunitySolidServer:main",
      "issuer": "did:web:secureissuer.solidcommunity.net:public",
      "client": "did:web:secureapp.solidcommunity.net:public",
      "agent": "did:web:bboi.solidcommunity.net:public"
    }
  },
  "options": {
    "challenge": "zFWfzGWq0UtAuZOiwKwV0g==",
    "domain": "https://bboi.solidcommunity.net/definition/ourProtocol#test"
  }
}
```


Note: The VPR, which is the request of credentials, is a Knowledge Grarph.

Use Case

Non-Repudiable Resource Request

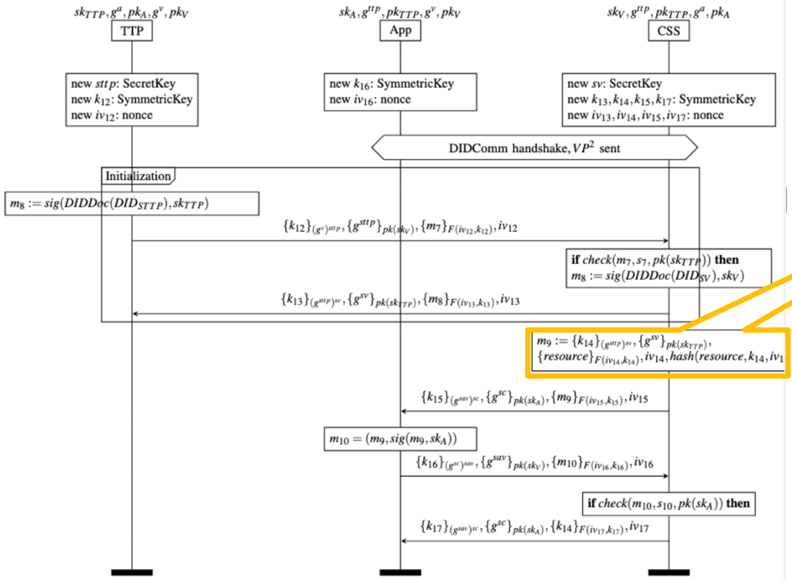
Before to exchange the VPR a connection is established. The VPR is then signed by the client and forwarded to the agent.




```
I received the following VPR (Connection Record: 8987fdfa-7203-4a56-928b-3d8362a1cff5 with DID did:web:secureapp.solidcommunity.net:public and label SecureS
olidApppppppppp):
ACP Policy:
For the target: http://localhost:3000/my-pod/test-folder/test-resource.txt with the client (app): did:web:secureapp.solidcommunity.net:public
```

Use Case
Non-Repudiable Resource Request

The resource is delivered in an encrypted and signed way. This operation is needed to guarantee non-repudiation of origin (NRO).



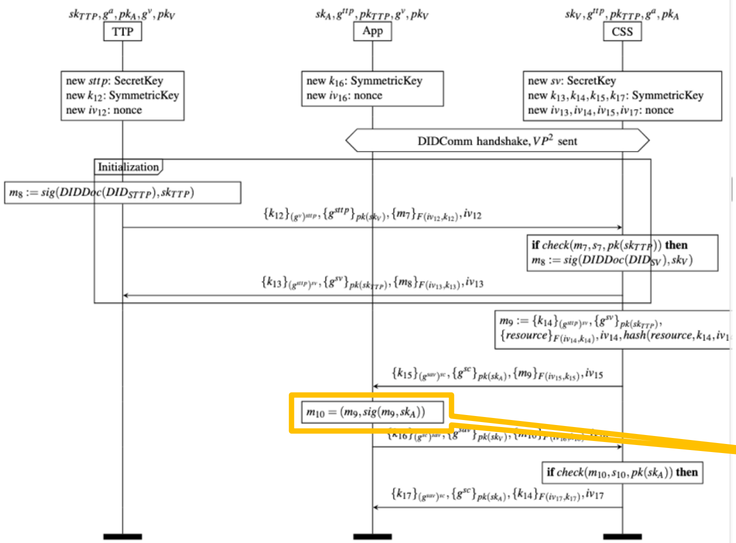
Non-repudiation of origin (NRO)

{"protected":"eyJlbmMiOiJ4Y2hhY2hhMjBwb2x5MTMwNV9pZXRmIiwidHlwIjoiSldNLzEuMCIsImFsZyI6IkFub25jc
JDMHp5OWIweW1UUzV3aFdoMEVVT0tBbE9yQkdZYi1Xdm5qd1NpNVZhSkp6WmlfRVcyNHJDLXIyNUlnSlJHS0tqTnNtTEhFS
I2N1pMNnZ0R0d1d2hOcHNnaGJBckdQenR0c25DM2JDb3Y3Q1NIcTRpMXc1SyJ9fV19","ciphertext":"j9TWKO1JyQaSK
76Qm9XekJt6QlCNlDfZtSrNUFu43nHCD851LwzgeQx1Ei-i5qx2bB3BZ6tEPdSjpQ1rfcKzaqT93J19MbRzdBxmgeYcQaCP
GTFOkdpzOZm_bjqEc4CYmeerkZT2sq6-UJ01QWzB4ImT5qXuohS7oOQcml6alf9e5Kbz3xGF2GnocHq32qj56iynPmZsmbC
OsfjUH_gvbfA4gX9SWg_hBClIIpw0Sbo2qOKR7d4oQ30D-pQpFkJ50YfdPKetskudqnDkx1iIsV0HrdjTipbsCLoN2P03NR
P6koy93Btl9GFFHqIRUGxT8_tpw72fcDd4WBqGWJrAX9pWBsO2AjtylyjlqUt6vDrolSPokHfdgj5-UN9ZrLSTdZJsSjPmF
76UakR0QvP_sRHGGo9Jj7j2yO1HxvSlzRwE4PHDhqmYVbxa9EPMeR4JrbzYQ","iv":"fySs3Zk2exgrK1YY","tag":"UZ
765c5e172fb13077bf0edfbc724b03449"}

Use Case

Non-Repudiable Resource Request

To obtain access to decrypted resource, the client must sign a message which represents the non-repudiation of reciept (NRR) that demonstrates the app really accessed the resource. At this point the server release the key for the decryption.

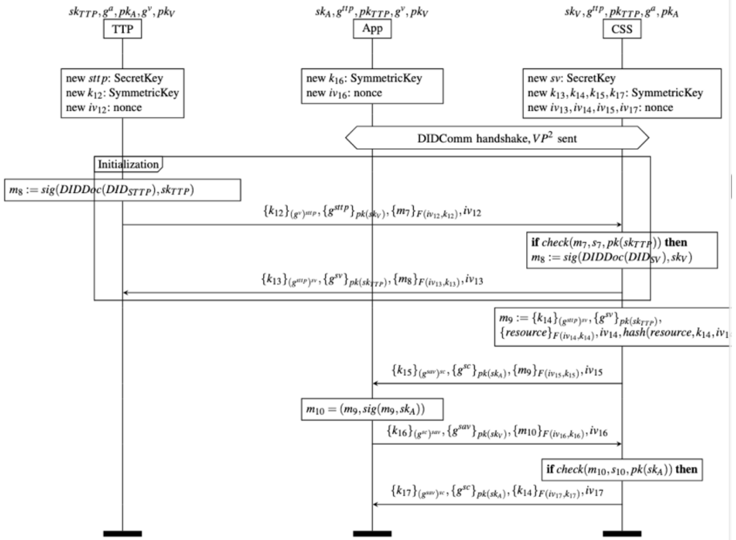


Non-repudiation of reciept (NRR)

Use Case

Non-Repudiable Resource Request

To obtain access to decrypted resource, the client must sign a message which represents the non-repudiation of reciept (NRR) that demonstrates the app really accessed the resource. At this point the server release the key for the decryption.

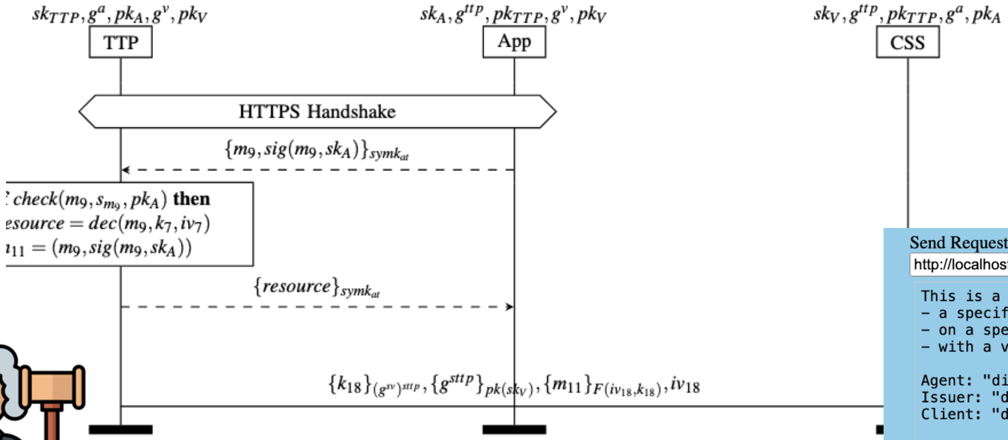


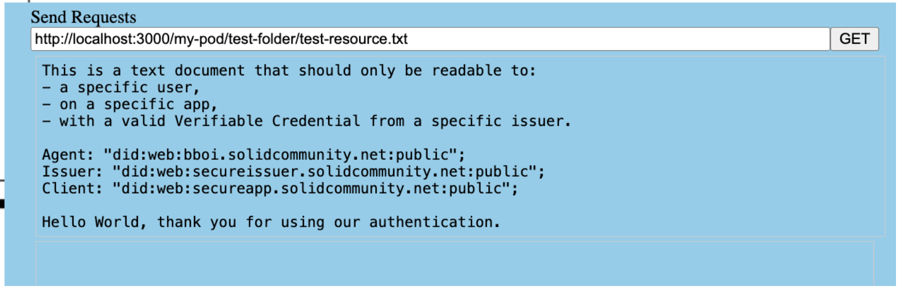
And if the server is malicious and sends the wrong key? Then we delivered NRR without really accessing the resource !!!!

Use Case

Non-Repudiable Resource Request

To overcome this problem, the key used for ciphering the resource is shared among CSS and TTP, meaning that the App can ask to the TTP to decrypt the message in case the CSS acts malicously.





Conclusion

We have demonstrated how Knowledge Graphs can be a powerful tool to enhance non-repudiation protocols, especially in decentralized systems where trust and security are critical. By providing a standardized, tamper-proof representation of actions, they enable secure, verifiable interactions

While the proposed solution offers significant benefits, challenges remain, particularly in the areas of private key management and data canonization. Future work will focus on refining these aspects and exploring broader applications in decentralized authentication.

Thank you!

Questions?

Biagio Boi
Enhancing Non-Repudiation Protocols with Knowledge Graphs

EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY
Funded by the European Union