ETSI GR PDL 014 V1.1.1 (2022-10)

Permissioned Distributed Ledger (PDL); Study on non-repudiation techniques

Disclaimer

The present document has been produced and approved by the Permissioned Distributed Ledger (PDL) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGR/PDL-0014_non_repud_tech

Keywords
interoperability, scalability, security, smart contract

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Permissioned Distributed Ledger (PDL).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document covers the non-repudiation challenges in Permissioned Distributed Ledgers (PDLs), the non-repudiation strategies/technologies, and their viability in PDLs. It also defines the limitations in non-repudiation strategies in PDLs and possible future directions.

The present document discusses PDL based end-to-end architecture that provides non-repudiation. This includes non-repudiation for input and output data for a PDL, such as external PDLs and smart contracts.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IETF RFC 4270: "Attacks on Cryptographic Hashes in Internet Protocols".

NOTE: Available at https://datatracker.ietf.org/doc/html/rfc4270.

[i.2] David Chaum: "Blind Signatures for untraceable Payments".

NOTE: Available at http://blog.koehntopp.de/uploads/Chaum.BlindSigForPayment.1982.PDF.

[i.3] Masahiro Mambo, Keisuke Usuda, Eiji Okamoto: "Proxy Signatures for Delegating Signing Operation".

NOTE: Available at https://dl.acm.org/doi/pdf/10.1145/238168.238185.

[i.4] ETSI GS PDL 012: "Permissioned Distributed Ledger (PDL); Reference Architecture".

NOTE: Available at https://www.etsi.org/deliver/etsi_gs/PDL/001_099/012/.

[i.5] D. Boneh: "Aggregate Signatures", in Encyclopedia of Cryptography and Security, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, p. 27.

[i.6] ETSI TS 133 303 (V14.1.0): "Universal Mobile Telecommunications System (UMTS); LTE; Proximity-based Services (ProSe); Security aspects (3GPP TS 33.303 version 14.1.0 Release 14)".

[i.7] D. He, J. Chen, and R. Zhang: "An efficient identity-based blind signature scheme without bilinear pairings", Comput. Electr. Eng., vol. 37, no. 4, pp. 444-450, Jul. 2011, doi: 10.1016/j.compeleceng.2011.05.009.

[i.8] T. Peacock, P. Y. A. Ryan, S. Schneider, and Z. Xia: "Verifiable Voting Systems", Comput. Inf. Secur. Handb., pp. e293-e315, 2013, doi: 10.1016/B978-0-12-803843-7.00090-9.

[i.9] D. A. Wijaya, J. Liu, R. Steinfeld and D. Liu: "Monero Ring Attack: Recreating Zero Mixin Transaction Effect", 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018, pp. 1196-1201, doi: 10.1109/TrustCom/BigDataSE.2018.00165.

[i.10] Goldreich, Oded, and Yair Oren: "Definitions and properties of zero-knowledge proof systems". Journal of Cryptology 7.1 (1994): 1-32.

NOTE: Available at https://www.wisdom.weizmann.ac.il/~oded/PSX/oren.pdf.

[i.11] Manoj Kumar Chande, Cheng-Chi Lee & Chun-Ta Li (2018): "Cryptanalysis and improvement of a ECDLP based proxy blind signature scheme", Journal of Discrete Mathematical Sciences and Cryptography, 21:1, 23-34, DOI: 10.1080/09720529.2017.1390845.

[i.12] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab: "Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks", in European conference on Wireless Sensor Networks. Springer, 2008, pp. 305-320.

[i.13] ETSI GS PDL 011: "Permissioned Distributed Ledger (PDL); Specification of Requirements for Smart Contracts' architecture and security".

NOTE: Available at https://www.etsi.org/deliver/etsi_gs/PDL/001_099/011/01.01.01_60/gs_PDL011v010101p.pdf.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

auditability: ability of an object to undergo a thorough examination and evaluation

NOTE: Generally, auditability is measured against criteria defined by certain authority, such as the PDL governance.

governance: collection of rules and tools that control the behaviour and function of a PDL Platform (see ETSI GS PDL 012 [i.4]).

identifiable natural person: one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person

mainchain: primary non-dependent chain which forms the PDL network

PDL participants: nodes which form the PDL network

personal data: any information relating to an identified or identifiable natural person

sidechain: sub-chain which is dependent on a mainchain

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API Application Programmable Interface

BP	Bilinear Pairing
CA	Certificate Authority
CRC	Cyclic Redundancy Check
ECDSA	Elliptic Curve Digital Signature Algorithm
GDPR	General Data Protection Regulation
I/O	Input/Output
IoT	Internet of Things
MD5	Message Digest 5
NFT	Non-Fungible Token
NR	Non-Repudiation
NRD	Non-Repudiation of Delivery
NRE	Non-Repudiation of Emission
NRO	Non-Repudiation of Origin
NRR	Non-Repudiation of Receipt
NRS	Non-Repudiation of Submission
NRT	Non-Repudiation of Transport
PKI	Public Key Infrastructure
RSA	Rivest-Shamir Adleman
SHA	Secure Hashing Algorithm
TCP	Transmission Control Protocol
TTP	Trusted Third Party
ZKP	Zero Knowledge Proof

4 Introduction to Non-Repudiation Techniques

4.1 Definition

IETF RFC 4270 [i.1] defines non-repudiation as:

> *"A Security service that provides protection against false denial of involvement in a communication."*

Distributed ledges inherently implement non-repudiation through strategies such as digital signatures. Also, every node in the network keeps a record copy; therefore, it is theoretically unrealistic to deny a digitally signed transaction. However, in some situations, it is required to verify data integrity. For example, in the cases of smart contract offloading [i.2].

Therefore, non-repudiation, particularly in PDLs, can be defined as:

> *"Verification techniques and strategies that can provide secure proof that the data entered to/from the PDL are from a valid source and is unaltered, that is, same as entered by the source."*

For example, when capturing temperature data, it can be confirmed that the data captured from the thermometer is unaltered, but it cannot be guaranteed that the thermometer is accurate. In such a case, strategies such as device identity will play a key role.

> EXAMPLE: Laboratory calibration confirmation and data, will be associated with the device (e.g. thermometer) identity.

Permissioned Distributed Ledgers (PDLs) provide accountability to the transactions through their inherent properties such as transparency. Historic transactions provide an audit trail for a future audit and produces undeniable records. However, in an end-to-end scenario, a PDL will not be a solitary entity and will include other functional components such as oracles and external data storage.

4.2 Types of Non-Repudiation

4.2.1 Introduction

In distributed ledgers, the role of non-repudiation is to collect evidence, verify and authenticate the source of data. In this clause, the types of non-repudiation relevant to PDLs are discussed.

4.2.2 Non-Repudiation of Origin (NRO)

Non-Repudiation of Origin (NRO) is an application layer consideration and proves that the data is from the source claimed by the message.

PDLs often take data inputs from various data sources, for instance, via oracles or in/directly from the devices. In such a situation, the authenticity of the data source will need to be verified.

4.2.3 Non-Repudiation of Emission (NRE)

Non-Repudiation of Emission (NRE) is a network layer consideration. It provides proof that the data sent is accurate and unaltered while being sent to and stored on the PDL.

In PDLs, this problem may occur when a user sends a valid message and a malicious party in the middle tampers with the message.

EXAMPLE: A user sends a bid through smart contract execution, a malicious user changes the bid as per latter's advantage.

4.2.4 Non-Repudiation of Receipt (NRR)

Non-Repudiation of Receipt (NRR) is the false denial of the receipt of a message. Typically, Permissioned Distributed Ledgers are inherently resilient to NRR because of their distributed nature. As long as the majority (as required by the consensus mechanism) of nodes receive the message correctly, it will be distributed to all the other nodes even if they maliciously deny the receipt from the sender.

4.2.5 Non-Repudiation of Submission (NRS)

Generally Non-Repudiation of Submission (NRS) provides proof that the sender submitted the data for delivery.

Since Permissioned Distributed Ledgers are implemented on the public Internet they are prone to transaction delay and network layer congestion. For example, a remote device sends data to a PDL node, and the transaction is delayed due to network conditions. In certain cases, such delay may render the data invalid. Non-Repudiation of Submission offers a set of tools that can prove that a transaction is valid.

Additionally, NRS may resolve some security vulnerabilities for the PDLs related to receipt of data from external resources. For instance, when the data or a smart contract is sent by an external storage or oracle, it is prone to malicious activity (e.g. virus) in the PDL. NRS provides the sender and the recipient the ability to verify the integrity of the data and the proof of submission.

4.2.6 Non-Repudiation of Delivery (NRD)

NRD provides the sender a set of tools that can prove that data was submitted and delivered to the recipient even if the recipient denies the receipt and/or fails to act upon the data received.

In PDLs, typically NRD is addressed in the context of Trusted Third Party (TTP) and provides the proof that the data was handed to the TTP or a Delivery Agent for delivery. Yet, distributed ledgers, in particular, PDLs advocate distributed trust and there is no TTP by the definition of PDL. Despite the distributed trust, PDLs should still offer NRD to enable data and smart contract efficiency and distribution. Specifically, in situations where the smart contracts are stored on third-party managed external storage.

4.2.7 Non-Repudiation of Transport (NRT)

Non-Repudiation of Transport (NRT) provides the proof that the data was sent by the sender and transported by the delivery agent (e.g. transport channel). NRT is different from NRD, due to the fact that there may be several transport entities involved in one end-to-end message delivery.

4.3 Generalized Non-Repudiation Scenarios

Unreliable Communication Channel: The sender sends a transaction that is dropped/delayed due to poor connectivity conditions or malicious activities on the transmission channel such as Man-in-The-Middle Attack. This may affect both the sender and the recipient because it may cause delay or non-approval of transactions and consequently may result in monetary losses for the parties.

Malicious Sender: The device user or sender is malicious and sends wrong, late or no data. In the example of smart contract external storage, it may include both the owner and the user of the device [i.2].

Malicious Receiver: The receiver of the data is being malicious and denies the receipt of the data. In a PDL scenario, the receiver is expected to be PDL nodes. However, this problem may arise when data traverses an intermediate object/entity such as an API.

4.4 Non-Repudiation Process

By definition, the non-repudiation, is a technique to generate proof that, at a later date, both the service and receiver can use to ensure proper operation of the PDL. Four phase processes for non-Repudiation are defined in [i.3] as follows:

Evidence Generation

The evidence of the message is generated by the respective participants of the system, for example, a sender or receiver. This evidence will later be used by the parties to verify that the data was transmitted/received by the other participant and prevent them denying generation/receipt of the data.

Evidence Transfer and Storage

The evidence generated by the sender/receiver are expected to be stored securely in local/governance-controlled storage.

Evidence Verification

The evidence of the transfer (of assets or an entity) is expected to be verifiable.

Dispute Resolution

In the case of PDLs, dispute resolution can be handled by the governance.

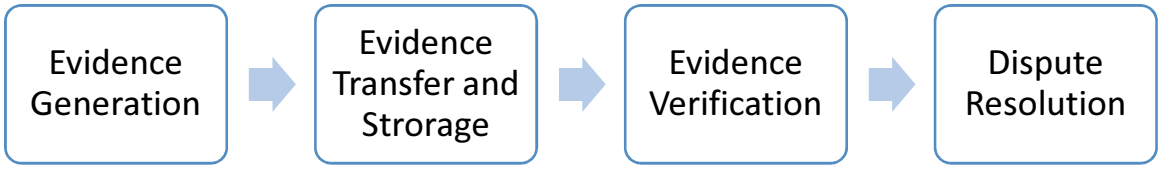


Figure 1: Non-repudiation process

Table 1: Examples of non-repudiation and possible solutions

Challenges/Objects	Example Scenarios where challenges occur	Possible solutions
Mismatched data - sent by the originator and received by the recipient are inconsistent or not the same	NRE, NRD	The sender sent the data and the receiving party received that but the proofs are mismatching, or not same. In this scenarios, the dispute can be launched and governance can step-in to resolve the issue.
Delayed data - data arrived at the destination after the allotted period of time	NRT, NRD, NRE	Timestamps can be recorded in a hashed form together with the data. Sometimes, timestamps can be encrypted part of the data. This problem applies to the *time-critical data.*
Tampered data (both accidental and malicious) - the data sent is tampered by the sender, receiver or a man-in-the-middle	NRE, NRD	Regular device health checks, and monitoring of transmission delays/losses can help the parties to identify the real reason of data tampering. In PDLs, the governance, can also take compliance actions against the parties with poor transmission record and may set standards for connection and security requirements. Multiple transmission ways can also be adopted to ensure the integrity of data.
Erroneous data - the data sent is incorrect or not complying with the agreed/set standards	NRE	Here regular device checks and governance compliance strategies can help.
Missing/incomplete data - the data is not sent at all/or missed by the communication channel	NRE, NRD, NRD	Governance compliance strategies and regular node checks can help.
Data Re-ordering - data arrives in a sequence different than it was sent	NRE, NRD, NRT	Application layer protocols for reordering the packets, at a cost of additional delays.

5 Objects of Non-repudiation

5.1 Introduction

In PDLs, data is written to a ledger by internal and external participants. This includes PDL nodes and oracles; the main objective is that the data integrity is maintained. In this clause, the key properties for a non-repudiation mechanism are highlighted.

5.2 Pre-requisite

5.2.1 Evidence Recovery

In networks, the evidence (e.g. receipt) are sent through a wireless/wired link. These links may suffer from disruption or performance issues that may cause data loss, which may result in delayed or lost evidence. In some applications, such delays in evidence arrival may cause the stakeholders losses such as a delayed bid. Therefore, the evidence generated by either party needs to be recoverable with correct parameters (e.g. time original evidence was sent).

5.2.2 Redact

5.2.2.1 Introduction

In business applications particularly a document may include numerous details such as stakeholders' personal information and previous business dealings. When a number of parties involved in a dealing or business, the whole document cannot be disclosed to all the parties, nor the non-repudiation of a complete document is required. For example, in a loan application, non-repudiation of finances may be required, and stakeholders' personal records may be irrelevant and can be kept hidden through mechanisms such as encryption.

A good non-repudiation mechanism ought to provide *Redaction*. That is participants are allowed to disclose the relevant part of the data for non-repudiation purposes only and the rest of the document (the document for non-repudiation) can be hidden/masked. The non-repudiation proof will be for a revealed/disclosed part of the document (the document for non-repudiation) only.

5.2.2.2 Difference between Data Masking and Data Redacting

The difference between data redaction and data masking is that the former hides the sensitive details such as credit card information. Data Masking may cause the risk on digital platforms because nullified attributes may cause unpredictable results (e.g. divide by zero).

Data Redacting changes the data by replacing part or all of the actual data and other techniques such as encryption. The result is an attribute with readable but meaningless information for interceptor.

Both techniques are useful for GDPR compliance purposes.

5.2.3 Robustness

A non-repudiation mechanism is expected to be unbreakable by advanced systems such as quantum computing. Such that a malicious party will not be able to break the security mechanism and produce false non-repudiation proofs under any circumstances.

5.2.4 Performance

The non-repudiation protocol is needed to be computation and space efficient. In a PDL scenario, performance metrics such as latency, bandwidth and computational power of the devices (e.g. PDL nodes, external storage) should be taken into account when adopting a NR protocol.

5.2.5 Transparency and Auditability

Distributed ledgers (both permissioned and permissionless) are transparent by definition and design, which is one of the reasons for their wide adoption. To maintain the property of transparency, non-repudiation protocols used by PDL networks, need to be transparent and auditable. The code and algorithms of a NR protocol will be freely auditable and open-sourced.

5.2.6 Coalition Resistance

To maintain the trustworthiness of a PDL platform, NR protocols adopted by such platform should be coalition resistant. It should discourage any collusion between the PDL stakeholders in a manner that may influence the decisions of a consensus or governance.

5.2.7 Evidence

Evidence produced by a NR protocol is secure, non-changeable and transparent. It is generated in a timely manner and communicated securely over a communication channel.

Most of the existing non-repudiation systems use receipts as an evidence of delivery. Using cryptographic methods (e.g. CRC) can ensure that recipient has received message unaltered. In such a case NR protocol will ensure that the receipts generated are unforgeable, untamperable and generated with minimal network resources. Moreover, the receipts are communicated in a timely manner. The receipts can provide details such as time, date and other metrics defined by the PDL governance.

5.2.8 Fairness

A non-repudiation protocol should not give advantage, in terms of, for example, computational power and memory, to one party over another. Governance can ensure that non-repudiation algorithms adopted by the PDL platform are adequately implemented by all the internal and external participants. However, in some situations, governance may need to ask the nodes to make necessary upgrades to enable a more robust non-repudiation algorithm. The point is, no one party, under any circumstances, gets benefited over another.

5.2.9 Order Preserving

In case of TCP communication, data may arrive in different order. Indeed, packet headers allow reassembly of the packets and enable to correct the order of messages at the receiver's end. This property is also required to be available at the non-repudiation layer. It is important that the non-repudiation technique is able to verify the order of the messages sent by the source.

5.2.10 Protection Granularity

The non-repudiation proofs can be verified at the byte or message level or periodically (e.g. every second or every 10 bytes or so). The governance of the PDL may specify the required granularity at the time of the PDL initialization.

5.2.11 Digital Signatures

5.2.11.1 Introduction

Digital signatures provide a method to verify the authenticity of a document digitally. The sender signs the hash of the document/data, to be signed, with its private key and sends the document/data along with the corresponding public key. Generally, in distributed ledgers, participants (i.e. nodes) use digital signatures to verify the Non-Repudiation of Origin (NRO). Such non-repudiation enables trust in the distributed ledgers.

5.2.11.2 Considerations

5.2.11.2.1 Hashing and Signing Algorithm

The document/data is hashed and encrypted with sender's private key producing a digital signature. There are several hashing and signing algorithms available for digital signatures such as MD5, ECDSA, RSA and SHA and others. The choice of signing algorithm depends on factors such as computational power and required level of security.

In PDLs, however, governance approved hashing and signing algorithms will enable the consistency and uniformity in the PDL system.

5.2.11.2.2 Hashing and Key Sizes

The consideration of hashing algorithm leads to the consideration of key size. Hashing algorithms vary with key sizes, for instance, SHA-3, supports, several key sizes such as 20 and 32 bytes. Signing algorithms such as RSA need to support appropriate key sizes (i.e. > 3 072 bits) to ensure secure signing of the data.

It is up to the governance of the PDL to choose a suitable key size with a suitable hashing and signing algorithm. Note that hash can be keyed or non-keyed, the reliability of the protocol depends on the final digest length. Governance may also consider such factors whilst choosing hashing and signing algorithms.

5.2.11.2.3 Certificate Authority (CA)

Digital signatures need public/private key pairs. Certificate authorities can be used to verify the validity of public keys. They maintain the mapping between the public key and key holder. Although self-issued public/private keys are widely used, it is a good practice to use CA-registered keys to ensure the genuinity and traceability of the identity.

In PDLs, several models for a Certificate Authority (CA) are possible. A CA can be a subsidiary of the governance, or an external entity accepted by the governance for key management. In the situations of external participants, the external governance may issue the keys which may be acceptable by the other PDL network. The governances may maintain a list of approved/recommended external certificate authorities.

5.2.12 Types of Digital Signatures

5.2.12.1 Introduction

There are several types of digital signature schemes proposed by the industry and the academia. Yet, all the schemes have their pros and cons. In this clause, some of the widely-discussed schemes are highlighted.

5.2.12.2 Aggregate Signatures

Aggregate signature algorithm combines the digital signatures of two or more parties and produces a single signature which validates all the signatures. In a PDL scenario, aggregate signatures can be useful for validating the external data. For example, an API can take the input from several data sources, to ensure the validity of data and combines all the signature and produce an aggregate signature. Yet, aggregate signatures conceals the identity of the signers, it is difficult to trace the individual signers in this case.

5.2.12.3 Group Signatures

A group member signs the document on behalf of the whole group. Group signatures removes the link between the signer of the document and its public key, hence, even if the adversary acquires the private keys of the group members, they will not know, which group participant, in fact, signed the document. Generally, group signatures are useful when the anonymity of the signers is required. The signer is known to the group and is selected by an algorithm.

5.2.12.4 Ring Signatures

Ring signatures are similar to group signatures, however, in Ring Signatures, signers are selected randomly from the group. None of the participants of the group have the knowledge about the actual singer beforehand, therefore they provide higher degree of privacy in the PDL network. Due to the random selection of the signer, Ring Signatures provide high degree of anonymity for the signer.

5.2.12.5 Blind Signatures

Blind Signatures are proposed by David Chaum in his foundational paper: "Blind Signatures for Untraceable Payments" [i.2]. The idea of the Blind Signatures is to anonymize the sender. To that end, the signers sign the message without knowing the contents of the message which can be later submitted with the original message to prove the non-repudiation of contents.

5.2.12.6 Proxy Signatures

In Proxy Signatures, an original signer delegates the signing authority to another party, generally referred as 'Proxy'. The signature can later be verified with the information of actual and the proxy signer. Proxy Signatures are useful in several use cases, for example, in the situation of resource limited devices (e.g. IoT), which cannot perform computation intensive digital signature algorithms and may delegate the signing task to a dedicated proxy.

Three different types of proxy signatures are discussed by the Mambo et al [i.3]:

1) **Full Delegation:** the proxy signer is given and uses the same credentials as the original signer. In the situations of malicious proxy signer, it is difficult to identify the culprit due to the identical signing credentials.

2) **Partial Delegation:** the proxy signer is given signing credentials which are derived from the original signer's credentials but are different from them. The proxy signer can be identified through Partial Delegation.

3) **Warrant Delegation:** a participant who be entrusted is identified and given the right to act as a proxy signer on behalf of the original signer.

5.2.13 Evaluating Signature Schemes

5.2.13.1 Introduction

There are three main metrics for evaluating signature schemes:

1) Computation overhead (computational time cost).

2) Communication overhead (messaging and distribution).

3) Bandwidth overhead (data transmission).

There are two different types of schemes available for digital signatures.

5.2.13.2 Bilinear Pairing (BP) based schemes

Bilinear pairing has high computational cost [i.5].

5.2.13.3 Non BP based schemes

Depending on the digital signature algorithm, Non-BP based schemes may have lower computational costs.

5.2.13.4 Overheads due to Mathematical operations

Typically, digital signatures are driven by the number of scalar multiplications and modular exponential operations performed during the signature calculations. This may lead to computational overheads.

Table 2: Comparison of Digital Signature Types

Digital Signature	Participant Anonymity	Performance	Security
Aggregate Signature	Yes	Based on BP, therefore computational cost is high [i.12].	Depend on the algorithm and dependent cryptographic hardness assumptions.
Group Signature	Local	Depends on the algorithm ETSI TS 133 303 [i.6].	Depend on the algorithm and group manager.
Ring Signature	Yes	Efficient (i.e. low computational overheads). Other schemes relying on RSA or BP may lead to higher computational overheads.	Depend on the algorithm and dependent cryptographic hardness assumptions. Recent studies identified the different users in a ring [i.9].
Blind Signature	Yes	Most schemes are based on bilinear pairing [i.7], therefore they have higher computational overheads. Some Blind Signature algorithms do not rely on bilinear pairing and are more efficient [i.8].	Depend on the algorithm and dependent cryptographic hardness assumptions.
Proxy Signature	No	Certificate based proxy-signature schemes can be more efficient in terms of computational time overhead but heavily rely on TTP or PKI. Certificateless proxy signature schemes reduce the dependence of TTP but may incur higher computational time overhead if BP or other complicated cryptographic operations are applied.	Dependent on algorithm used and cryptographic hardness assumptions. In 2018, the proxy signature is proved to be insecure [i.11].

5.3 Smart Contracts

Smart contracts are auto-executable codes and installed on the PDLs. The participants can execute/activate a smart contract with some actions, such as read/write data to the PDL. The executions (e.g. read/write data proofs) can be recorded to the PDL and verified later.

EXAMPLE: If a PDL node consumes/uses data from an external, third-party, oracle the data is vulnerable to being malicious or inaccurate. In such a scenario, non-repudiation can be achieved through execution of a smart contract when data is accessed or exchanged with an external party. Such contract can record/apply access conditions and penalties in case of denial of non-repudiation. Depending on the PDL-type, certain policies can be implemented as part of a smart contract to enable enforcement.

5.4 Oracles

Oracles inherently provide non-repudiation methods for the external data entered to the PDL. Generally, they work as intermediary between the data sources and the PDL. They can verify, authenticate and interpret the external data to make it PDL-ready. They are generally trusted by the PDL governance due to their methods and procedures for data processing. However, to add an additional layer of security and integrity, in PDLs, governance can maintain a list of entrusted/reputable oracles and their associated security keys.

Oracles may need an additional security clearance, that may be obtained from a PDL governance-defined certificate authority.

5.5 Trust Anchors

Trust anchor is an entity that provides the root of the trust to stakeholders. The stakeholders trust an object such as a key or certificate due to reputation of the issuer - the issuer is considered to be the trust anchor. The most common type of trust anchor is a certificate authority.

5.6 Governance

Governance of a PDL can take the compliance strategies to ensure the non-repudiation within the PDL network. Generally, PDL-related decisions are the responsibility of the governance, which may introduce strategies and techniques to ensure the security and integrity of the PDL such that no party can misbehave (e.g. deny the non-repudiation). For example, the governance may have a list of approved oracles, and penalties in the situations of contract breach.

6 Scenarios

6.1 Introduction

Typically, in PDLs, non-repudiation is inherited, techniques such as digital signature and access control (specific to PDL-type) provide methods of non-repudiation. In an end-to-end PDL network, in which several external and internal sources (e.g. data sources and certificate authorities) are likely to be involved, however, non-repudiation can happen in the following ways.

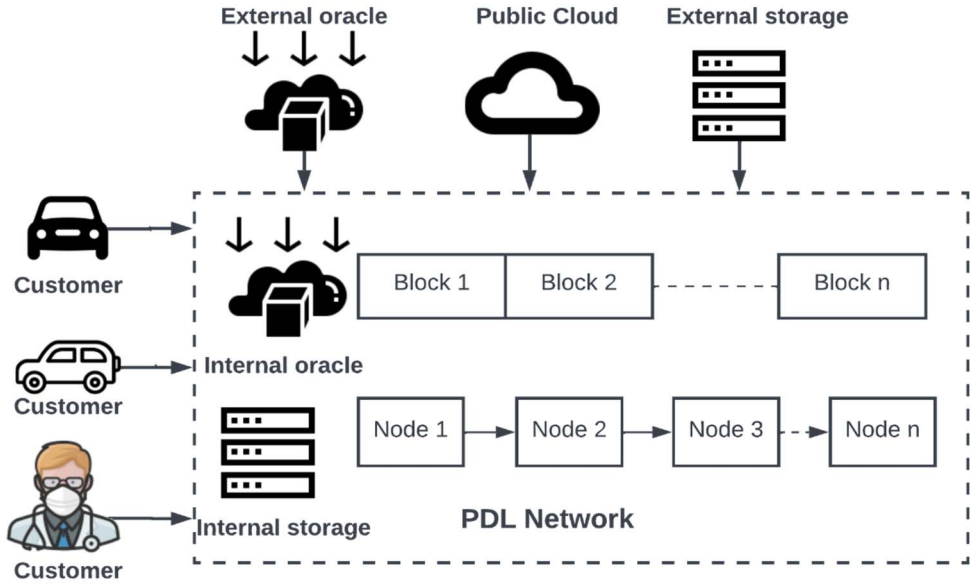


Figure 2: Generalized PDL scenario with internal and external data inputs

6.2 Attacks to Data Communication

In a typical scenario, a PDL exchanges information, both internal to the PDL and with external sources. The data is carried through various communication channels, for example, radio channel, private and public networks. Such communication channels can be interfered with. For example, eavesdropping, data corruption and impersonation. Such attacks are not unique to PDLs and may happen in any type of communication. PDLs should identify such situations and take appropriate actions, as PDLs are immutable, any data/information written to them cannot be changed.

EXAMPLE: In PDLs, the communication between an oracle and node can be intercepted by a malicious party. Any modifications to the receipt will, indeed, provide NRO but not the non-repudiation of contents.

6.3 Malicious Participants

In a typical PDL system nodes verify each other's transactions. As such, PDLs are permissioned by definition, therefore, it is unlikely, yet possible, that some nodes purposely send incorrect data and later blame a third party for the fraudulent transaction. For example, hacked/compromised machine/nodes may have sent valid data prior to being compromised and later, once compromised, send malicious data and claim to be honest considering their honest history. Other PDL participants, when seeing the fraudulent transaction from a node they perceive as honest, may approve such transaction. Such a scenario is difficult to handle, however mitigation techniques can be applied depending on the case.

The node is honest and unknowingly becomes compromised:

An honest node can be compromised without its knowing. In such a case the honest node does not know that it is sending incorrect/wrong data and there is a risk that this data will propagate throughout the network. Repudiation is possible when someone identifies that the information is corrupted, and it can then be traced back to the compromised node. Whether or not the consequences of incorrect/corrupted/inconsistent data can be reversed depends on the case.

The node is honest and knows it has become compromised:

In the event that the honest node knows that it has been compromised, it may send a message to all nodes that it has become compromised and should not be included in consensus and/or allowed to validate transactions until it has become uncompromised.

The node is malicious and pretend to be compromised:

A malicious node pretends to be honest for some time to gain trust and then performs malicious act to make larger damage claiming it has been compromised, thus not bearing any consequences.

To mitigate such problems, the governance may introduce mitigation techniques to identify the compromised/fraudulent nodes and remove them from consensus and restrict them from sending transactions. The governance may also introduce additional measures to ensure that all the participants take responsibility of the data originated by their machine/node and bear consequences if they are involved in malicious activity knowingly or unknowingly.

The node is greedy and sends the transaction without verifying its integrity:

In some cases, it is also possible that the node sends the transactions without verifying the integrity of the data provided to the node. Such situations may arise when the node has incentive to forward the transaction to the ledger such as bribery or greed.

6.4 PDL Network External Storages

6.4.1 Introduction

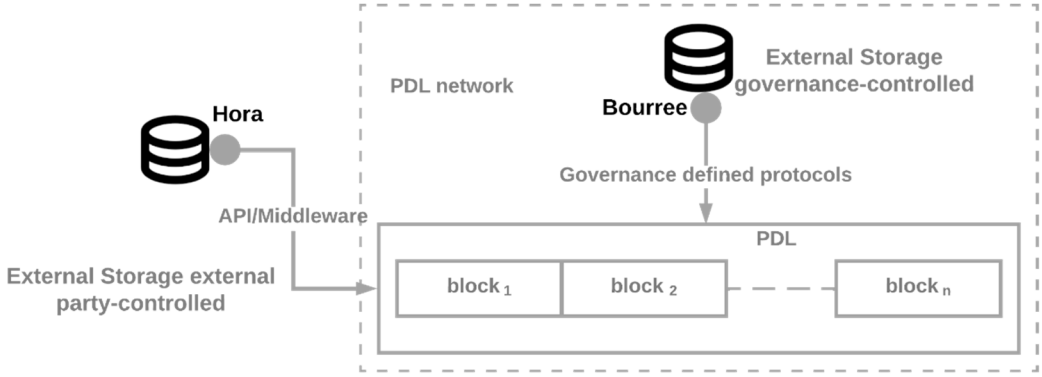
A PDL network may include several internal and external storage. The external storage can be managed by several types of entities such as trusted and non-trusted third party, internal or external PDL participants and governance nodes. Typically, external storage is a method to offload the I/O task away from the PDL consensus to enable performance benefits. External storage devices usually do not run PDL consensus, and may not be part of the chain at all. The records maintained by them may not be visible or verifiable by the PDL participants except to the nodes directly connected to them.

Typically, there are two different types of external storage scenarios possible:

- storage associated with and accessible by a specific node; and
- external/third-party external storage accessible by all participants.

Storage managed by a specific node is easy to comply with the PDL rules since the governance manages the PDL network and may take necessary actions such as compliance strategies and set protocols to ensure that the external storage cannot repudiate the input.

The problem/challenge can, however, be with external storage accessible by multiple participants, where governance may not have any control over the entity managing the storage. Such a system is vulnerable to intentional and unintentional, internal and external attacks. For example, external storage may send a tampered or delayed data and emit a fraudulent receipt. Non-repudiation in such a scenario, may be challenging due to lack of governance control. Nevertheless, governance can discourage or forbid using of such a storage. In situations where using a third-party storage is inevitable, governance approval may be required to ensure the integrity of the PDL. A possible solution can be the use of a trust anchor - an entity that is trusted by all the stakeholders of the PDL network may provide a guaranteed mechanism to verify the integrity of the external storage and the data provided by it. Additionally, this communication may happen through an intermediate oracle service to facilitate the translation and verification of the data from the external storage. The oracle service used to mediate this communication will be explicitly trusted by the trust anchor, external storage and the PDL network stakeholders. Therefore, the PDL network stakeholders will implicitly trust the external storage.



NOTE: Hora and Bourree interfaces are same as defined in ETSI GS PDL 012 [i.4].

Figure 3: External storage example

6.4.2 External Smart Contracts

Smart contracts are inherently providing non-repudiation. As such, they record all the transactions to a PDL, therefore, parties can always verify the transaction. However, problems may arise when external smart contracts (both PDL and non-PDL smart contract) send data to a PDL.

External PDLs may stop, interrupt and continue their smart contracts depending on the consensus of their local PDLs and governance. In such a scenario, if those, that is, external smart contracts are required to provide the input to local PDL, the following scenarios can happen:

- **Stopped/Terminated Smart Contracts:** If an external smart contract is stopped/terminated and has provided some data input to a PDL in the past, non-repudiation can be challenging. If this historical data later proved to be malicious, may cause disruption at the recipient PDL. Moreover, the sending PDL/smart contract may deny or repudiate the ownership of this data, such a scenario can cost receipt PDL members hefty losses such as unwanted payments.

- **External Smart Contracts with interrupted/changed parameters:** Smart Contracts invoked/executed by the local PDL consensus, so a smart contract can be interrupted or updated its parameters with the consensus of its PDL participants. If these smart contracts had some clauses included which impacted the external smart contracts, for example, provide non-repudiation receipts every pre-defined interval, and if this parameter is changed by the interrupt command, may impact the external PDL.

- **External Smart Contracts with interrupted and/or continued operation:** Smart contracts may be interrupted for some time and continued their operations after some time. The time for which a smart contract was interrupted and may changed their parameters may affect the external inputs. For example, they may have changed the price of the currency conversion, in such a case, external PDLs will receive a different conversion rate.

All external smart contracts and the scenarios involving such situations should be regularised by the local governance. The local governance may implement appropriate compliance and standardized methodologies to ensure that external smart contracts provide non-repudiation in all historic transactions and inputs regardless of their actions.

6.4.3 External PDL Networks

PDLs may take inputs from external PDLs, which can be from varied storage types such as mainchains, sidechains and standalone storage, these storage types will be managed by external governance. Since data can be from non-PDL storage as well, for instance from an external PDL's oracle or standalone storage, the data may have passed through several sources and communication channels which may be unreliable and will not provide repudiation on the data. The governance of the external PDLs, typically, ensures that all the data sources are providing authenticated and reliable data only, and may implement strategies to ensure that all the devices in their respective network provide non-repudiation. However, this may vary from PDL to PDL and may not be adopted by all the PDL networks. Also, a malicious external PDLs may send wrong/incorrect data intentionally to a PDL and try to blame other parties.

Naturally, the external PDL should not deny the fact that they have written the data to a recipient PDL and own their inputs in all the circumstances.

Additionally, the following scenarios may occur in case of external PDL networks or malicious behaviour:

- **Storage is deleted or removed:**

  External PDLs networks may include several storage types such as sidechains, mainchains and standalone types. Since, standalone storage types are not immutable, they can be deleted without the running the consensus mechanism. However, PDLs are immutable and cannot be deleted without the consensus of their respective PDL. Nevertheless, PDL can be removed or completely deleted if all the participants delete their ledgers through the agreed consensus. If such PDL networks have provided or providing the data to other external PDLs, the non-repudiation can be challenging, because if the whole PDL network or some part of the network is wiped out, and the governance of PDL can repudiate the data input in such a case.

- **External PDL network is compromised:**

  External PDLs can be compromised through attacks such as sybil and Man-in-the-Middle attack. Scenarios such as imposter sending fraudulent transactions may compromise the integrity of the external PDL. In such a case, the recipient PDL will blame the external PDL network and will demand the non-repudiation proof regardless of their internal security status. The external PDL network cannot deny the fact the data was provided by their storage component (e.g. sidechain and standalone storage) and they will need to solve the problem at their end.

- **External PDL network is malicious:**

  External PDLs networks may act maliciously and try to blame other parties for wrong/incorrect data input, in order to avoid providing non-repudiation.

A universal non-repudiation mechanism is needed to solve the problems discussed above and to ensure that, in the future, PDLs networks cannot deny the transactions or data inputs from any past and present component of their network.

Additionally, governance of the PDL networks, can implement mitigation and compliance strategies such as maintaining a list of trusted PDL networks, to ensure that the parties providing data to their PDL cannot deny the repudiation and should take responsibility of their inputs.

6.4.4 GDPR Considerations

GDPR regulations require the data owners to have the right to be forgotten. In PDLs it may be complicated because PDL nodes are spread across different jurisdiction, and they may not be legally required to follow the GDPR requirement of data deletion. Because of immutability of the PDLs and inability to delete the data it is recommended not to store GDPR sensitive information on chain and other solutions such as off chain storage and trust anchors can be considered. However, in some mandatory circumstances, it may be required to enter personal data in a PDL, in such a scenario, the participants can be made aware of the risks associated.

Data deletion in other storage techniques such as external storage, which may be the part of the PDL network may be allowed. In such a scenario, data can be deleted by the participant or the governance on request and after taking appropriate measures. Non-repudiation in such a case can be a challenge, the record of the existence of data should be maintained.

It is important that when the participants delete or modify the data, some proof of this modification/deletion is recorded immutably in order to provide non-repudiation in future on the historic data.

6.4.5 Oracles

Oracles can be of different types for instances oracles that takes the data from PDLs to external oracles and others that provide data to PDL from external sources. Generally, as non-repudiation is concerned, oracles extract data from external data sources such as weather/stock exchange websites. They translate this data to a PDL interpretable format before submitting to the PDL. Generally, oracles are an important source of data input and output for PDLs and may be trustable by the governance due to reasons such as oracle service reputation, prior dealings and internal security algorithms implemented. In PDLs, governance can also maintain a list of trustable oracles to ensure that only authenticated data is provided to the ledger by the oracles. ETSI GS PDL 011 [i.13] specifies PDL governance to implement strategies to maintain a list of trustable oracles to ensure secure data reads. However, when a PDL network is formed by several sub-PDL networks (may involve external PDLs), it may be difficult to keep track of oracles due to their agreements with external governance and the mixed network architecture. To this end, the PDL networks can implement strategies in which the whole system (a high-level PDL network formed by several PDL networks) can be formed, only by the PDL networks that implement trustable and local governance-verified oracles.

7 Mitigation Techniques

7.1 Introduction

As discussed earlier, PDLs provide methods of non-repudiation in several ways. The techniques adopted by such digital signatures make it difficult for PDL participants to repudiate their input. The present document, however, is focused on the non-repudiation in an end-to-end PDL network, where several internal and external service providers (e.g. oracles, smart contracts) input the data. Some of them may not be controlled by the PDL network and the governance, and are therefore unlikely to comply with the PDL network's consensus. In this clause, mitigation techniques for repudiation in such a scenario, are discussed.

7.2 Reputation-based Solutions

Reputation, in the present document, refers to the service providers' (e.g. node, oracle service, external PDL participants) prior dealing or dealings with the PDL network. In a PDL environment, the service providers who regularly send disputed data to the network, may be observed closely and then handled appropriately (e.g. blacklisting) by the PDL network.

Several different types of reputations can be maintained by the PDL network.

Table 3

Term	Definition	Indicative thresholds or boundary conditions (note)	Advisory
Positive Reputation	The device *can be* trusted, *without* a doubt, for the non-repudiation mechanisms.	if a node or device provides the receipts/proof accurately and timely 99,999 % or above of the time.	The devices can be audited only on periodic basis (e.g. quarterly).
Acceptable Reputation	The device *can be* trusted *with reasonable* doubt for the non-repudiation mechanisms.	if a node or device provides the receipt/proof accurately and timely between below 99,999 % and above 90 %.	The devices can be audited on periodic basis (e.g. quarterly) with some additional audits.
Grey Reputation	The device may be trusted but *with significant doubt* and if consensus rules allow the device input *may be ignored.*	if a node or device provides the receipt/proof accurately and timely between above 80 % and below 90 % of the time.	These types of the devices may be monitored closely for future breaches. The audit may be performed more frequently than the positive nodes.
Negative Reputation	The device *cannot be* trusted and thus *cannot* provide data input to the PDL network.	if a node or device provides the receipt/proof accurately and timely less then 80 % of the time.	These types of devices may be stopped by the PDL network immediately and comprehensive audit of the communication channel and the device may be performed before they are allowed back to provide the data.
NOTE: The reputation thresholds in the table are indicative only and will depend on the PDL network and the type of the industry they are applied in. The above indicative figures may be adjusted to address the specific industry guidance.			

Blacklisting, for example, can be applied to below acceptable reputation devices. Such compliance strategies can be enforced by the PDL native support such as governance or the PDL consensus. However, malicious PDL participants may exploit this to eliminate/exclude their unwanted/less-favoured service providers through collusion with other PDL participants. For example, a group of PDL participants repudiate the receipt of data from a service provider often to intentionally damage their reputation and exclude them from the system. This may result in other malicious activities such as bribery to provide the data to the network.

Distributed ledgers can be used to mitigate such a problem, for example, the service provider can execute a smart contract as soon as they provide the data. Also, not all the participants may not be given the right to score the service providers and authenticated and governance nodes can only rate the services and service providers.

7.3 Periodic Audits

Periodic audits on devices enable the PDL network to keep up to date with possible vulnerabilities in the input devices, for instance, possible data leaks and communication channels. Such audits also may ensure that service providers are using appropriate mechanisms for providing non-repudiation and the appropriate security mechanisms are used by the devices. Also, they may verify that the devices' adopted mechanisms such as receipt generation and key exchange are performing accurately. Such audits, by a third-party PDL participant, would ensure that the devices implement appropriate security methods, are well-maintained, and are not compromised. Claims of data breach, will be handled through respective governance rules/guidelines. The frequency of these periodic audits may be subjected to governance and specific industry-environment.

7.4 Incentivisation

The service providers can be incentivised to provide a valid non-repudiation proof such as loyalty points. With every certain checkpoint achieved, for instance, 99 % non-repudiation proofs, a device may be given an award (e.g. some free transactions) or token (e.g. NFT).

7.5 Governance Role

The governance policies can keep track of all the repudiations, their reasons, and timestamps. A separate, off-chain database can be maintained to store such historic data. The governance can also implement enforcement strategies, such as penalties and awards to ensure non-repudiation in the PDL network. Other strategies may include provide reports of the devices to other PDL networks, thus creating inter-dependency and strengthening the reputation validity.

7.6 Trusted Third Party (TTP)

Generally, in distributed ledgers, trust is distributed among PDL participants only and trusted third parties are not allowed. However, as a supplementary strategy, in some cases, trusted third parties may be allowed to maintain track of data inputs to the PDL network to enable the inter-PDL trust and reputations. In some cases, it may also be beneficial to delegate the confirmation of receipt generation to a trusted third party. Examples would be situations when PDL network nodes or devices cannot confirm the rightfulness and origin of the data due to reasons such as computational overheads and time constraints, and a TTP may perform these tasks instead.

The use of TTP, and specific situations when and where they can be used, has to be agreed through consensus by the PDL participants.

Trusted Anchors may be considered as a special type of TTPs, however, trusted anchors may not be considered as a trusted third party due to the fact that trusted anchors do not take part in core PDL activities such as consensus operations.

7.7 Zero Knowledge Proof (ZKP)

Zero Knowledge Proofs (ZKPs) are mathematical techniques based on probability theory, used for the verification of data without revealing the data itself. The three distinguished properties of Zero Knowledge Proofs (ZKPs) [i.10] are:

- **Completeness:** If the statement is true, the prover can convince the verifier that it is true.
- **Soundness:** If the statement is false, the prover cannot fool the verifier to think it is true.
- **Zero Knowledge:** The verifier gets no visibility of data whatsoever but it is able to determine whether it is true or false.

Since no actual data is shared in the verification process, ZKPs are ideal for privacy related tasks. They enable privacy in privacy-constraint systems such as permissionless distributed ledgers. However, ZKPs are based on probability theory with some uncertainties, a remote chance of non-repudiation exists in them. The governance of a PDL network may implement ZKP with these considerations and known risks.

8 Recommendations

Based on the study in the present document, it is recommended follow the guidelines herewith:

1) **Design:** Design PDL networks taking into consideration that PDL-based networks are vulnerable to non-repudiation challenges, as they involve several internal and external inputs and outputs.
2) **Management:** Manage the vulnerabilities on all layers.
3) **Mitigation:** Governance and members of the PDL can implement mitigation strategies such as reputation-based mechanisms to enable non-repudiation in the PDL network.

History

Document history		
V1.1.1	October 2022	Publication