International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1404
(10/2020)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Secure applications and services (2) – Distributed ledger technology security

Security assurance for distributed ledger technology

Recommendation ITU-T X.1404

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES (1)	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols (1)	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES (2)	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1319
Smart grid security	X.1330–X.1339
Certified mail	X.1340–X.1349
Internet of things (IoT) security	X.1360–X.1369
Intelligent transportation system (ITS) security	X.1370–X.1389
Distributed ledger technology security	**X.1400–X.1429**
Distributed ledger technology security	X.1430–X.1449
Security protocols (2)	X.1450–X.1459
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699
QUANTUM COMMUNICATION	
Terminologies	X.1700–X.1701
Quantum random number generator	X.1702–X.1709
Framework of QKDN security	X.1710–X.1711
Security design for QKDN	X.1712–X.1719
Security techniques for QKDN	X.1720–X.1729
DATA SECURITY	
Big Data Security	X.1750–X.1759
5G SECURITY	X.1800–X.1819

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1404

Security assurance for distributed ledger technology

Summary

Recommendation ITU-T X.1404 defines three levels of security assurance for the distributed ledger technology (DLT) in order to facilitate design and development of security assurance mechanisms. It further defines ten security assurance components encompassing the security assurance and specifies criteria and guidelines for achieving each of the three levels of a security assurance component. Finally, it also provides a mapping between specific threats and security assurance components and a mapping between specific security capabilities and security assurance components.

Distributed ledger technology (DLT) is defined as a shared digital ledger, which is a continually updated list of all transactions. The assurance of DLT is defined as the degree of confidence that the process or deliverable meets defined characteristics or objectives. An assurance level could be considered as a quantitative expression of assurance agreed among the relevant parties.

There is a need for specifying criteria and guidelines for achieving each of the three levels of a security assurance component: data integrity, data confidentiality, credential management, identity proofing of users, entity authentication, authorization, data obfuscation, consensus mechanism strength, smart contract and personally identifiable information (PII) data protection. To facilitate the design and development of security assurance mechanisms, this Recommendation is based on three levels of security assurance.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1404	2020-10-29	17	11.1002/1000/14450

Keywords

Consensus mechanism, distributed ledger technology, integrity, proof of work, security assurance.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

NOTE

INTELLECTUAL PROPERTY RIGHTS

Table of Contents

Recommendation ITU-T X.1404

Security assurance for distributed ledger technology

1 Scope

This Recommendation defines three levels of security assurance for the distributed ledger technology (DLT). It further defines ten security assurance components encompassing security assurance and specifies criteria and guidelines for achieving each of the three levels of a security assurance component. Finally, it also provides a mapping between specific threats and security assurance components and a mapping between specific security capabilities and security assurance components.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1401] Recommendation ITU-T X.1401 (2019), *Security threats of distributed ledger technology*.

[ITU-T X.1402] Recommendation ITU-T X.1402 (2020), *Security framework for distributed ledger technology*.

[ISO/IEC 27034-3] ISO/IEC 27034-3:2018, *Information technology – Application security – Part 3: Application security management process*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 block [b-ITU-T X.1400]: Individual data unit of a blockchain (see clause 3.1.2), composed of a collection of transactions and a block header.

3.1.2 blockchain [b-ITU-T X.1400]: A type of distributed ledger which is composed of digitally recorded data arranged as a successively growing chain of blocks with each block cryptographically linked and hardened against tampering and revision.

3.1.3 Byzantine fault tolerance [b-ITU-T X.1400]: Property that enables a system to continue operating properly even if some of its components fail or existence of intentional bad actors.

3.1.4 consensus [b-ITU-T X.1400]: Agreement that a set of transactions is valid.

3.1.5 consensus mechanism [b-ITU-T X.1400]: Rules and procedures by which consensus is reached.

3.1.6 control [b-ISO/IEC 27000]: Measure that is modifying risk.

NOTE 1 – Controls include any process, policy, device, practice, or other actions which modify risk.

NOTE 2 – Controls may not always exert the intended or assumed modifying effect.

3.1.7 credentials [b-ITU-T X.1311]: Set of security-related information consisting of keys, keying materials, and cryptographic algorithm-related parameters permitting successful interaction with a security system.

3.1.8 data integrity [b-ITU-T X.800]: The property that data has not been altered or destroyed in an unauthorized manner.

3.1.9 distributed ledger [b-ITU-T X.1400]: A type of ledger, that is shared, replicated, and synchronized in a distributed manner.

3.1.10 distributed ledger technology [b-ISO 22739]: Technology that enables the operation and use of distributed ledgers.

NOTE – This is the technology underlying Bitcoin and other crypto currencies.

3.1.11 hardware security module [b-ISO 13491-1]: Secure cryptographic device that provides a set of secure cryptographic services, e.g., key generation, cryptogram creation, PIN translation, and certificate signing**.**

3.1.12 hash function [b-NISTIR 8202]: A function that maps a bit string of arbitrary length to a fixed-length bit string. Approved hash functions satisfy the following properties:

1. One-way: It is computationally infeasible to find any input that maps to any pre-specified output, and
2. Collision resistant: It is computationally infeasible to find any two distinct inputs that map to the same output

3.1.13 identity proofing [b-ISO/IEC 24760-1]: Verification based on identity evidence aimed at achieving a specific level of assurance.

NOTE 1 – Identity proofing is typically performed as part of enrolment. Identity evidence can also be needed during maintenance of registered identity information, e.g., recovery of a user account.

NOTE 2 – Typically, identity proofing involves a verification of provided identity information and can include uniqueness checks, possibly based on biometric techniques.

NOTE 3 – Verification for identity proofing is usually based on an enrolment policy that includes specification of the verification criteria of the identity evidence to be provided by the entity.

NOTE 4 – The verified identity information obtained when performing identity proofing can be included in the registration and can serve to facilitate future identification of the entity.

3.1.14 node [b-ITU-T X.1400]: Device or process that participates in a distributed ledger network.

NOTE – A node can store a complete or partial replica of the distributed ledger.

3.1.15 off-chain [b-ISO/TC 307]: Related to a blockchain system, but located, performed or run outside that blockchain system.

3.1.16 on-chain [b-ISO/TC 307]: Located, performed or run inside a blockchain system

3.1.17 personally identifiable information (PII) [b-ISO/IEC 29100]: Any information that (a) can be used to identify the PII principal to whom such information relates, or (b) is or might be directly or indirectly linked to a PII principal.

NOTE –To determine whether a PII principal is identifiable, account should be taken of all the means which can reasonably be used by the privacy stakeholder holding the data, or by any other party, to identify that natural person.

3.1.18 proof of stake [b-ITU-T X.1400]: Consensus process, where an existing stake in the distributed ledger system (e.g., the amount of that currency that you hold) is used to reach consensus.

3.1.19 proof of work [b-ITU-T X.1400]: Consensus process to solve a difficult (costly, time-consuming) problem that produces a result that is easy for others to correctly verify.

NOTE – Producing a proof of work can be a random process with low probability so that a lot of trial and error is required on average before a valid proof of work is generated. Bitcoin uses the Hash cash proof of work system.

3.1.20 secure cryptographic device [b-ISO 13491-1]: Device that provides physically and logically protected cryptographic services and storage such as a PIN entry device (PED) or hardware security module (HSM), and which may be integrated into a larger system, such as an automated teller machine (ATM) or point of sale (POS) terminal.

3.1.21 security assurance [b-ISO/IEC TR 15443-1]: Grounds for justified confidence that a claim about meeting security objectives has been or will be achieved.

3.1.22 smart contract [b-ITU-T X.1400]: Program written on the distributed ledger system which encodes the rules for specific types of distributed ledger system transactions in a way that can be validated and triggered by specific conditions.

3.1.23 threat [b-ISO/IEC 27000]: Potential cause of an unwanted incident, which may result in harm to a system or organization.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 DLT actor: Actors involved in the distributed ledger technology (DLT) system.

3.2.2 DLT service provider: A service provider that offers a distributed ledger technology (DLT) based service to another by means of one of its provided service interfaces.

3.2.3 DLT user: A user that consumes or uses a service provided by another component. A component may be a provider of some services and a consumer of others.

3.2.4 DLT user group: A set of users in a distributed ledger technology (DLT) system.

3.2.5 Security assurance component: One of the different parts of which a security assurance is composed.

3.2.6 security assurance level: Degree of confidence in security assurance.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AC Access Control

ATM Automated Teller Machine

CM Credential Management

CMS Consensus Mechanism Strength

DC Data Confidentiality

DI Data Integrity

DLT Distributed Ledger Technology

DO Data Obfuscation

DP PII Data protection

EA Entity Authentication

HSM Hardware Security Module

IP Identity Proofing

LoSA Level of Security Assurance

PII	Personally Identifiable Information
POS	Point Of Sale
PoW	Proof of Work
SC	Smart Contract

5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "is prohibited from" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview of security assurance for DLT

This clause specifies three levels of security assurance (LoSA) for each security assurance component: low, medium and high. The security assurance components of interest include data integrity (DI), data confidentiality (DC), credential management (CM), identity proofing (IdP) of users, entity authentication (EA), access control (AC), data obfuscation (DO), consensus mechanism strength (CMS), smart contract (SC) and personally identifiable information (PII) data protection (DP). Each LoSA describes the degree of confidence in the respective process leading up to and including the relevant process itself.

6.1 Level of security assurance

Table 1 provides the overall concept for security assurance comprising objectives and description for each assurance level. Based on this concept described in Table 1, the different assurance components will be described in clause 8.

Table 1 – Overall concept of the LoSA of DLT

Level	Objectives	Description
LoSA1 – low	To achieve low level of security assurance	Minimal confidence in the respective security assurance component of DLT.
LoSA2 – medium	To achieve medium level of security assurance	Some confidence in the respective security assurance component of DLT.
LoSA3 - high	To achieve high level of security assurance	High confidence in the respective security assurance component of DLT.

6.2 Selecting the appropriate level of security assurance

Selection of the appropriate LoSA should be based on a risk assessment of the transactions or services based on DLT. By mapping impact levels to LoSAs, organizations managing DLT and providing services based on DLT can determine what LoSA they require. Table 2 indicates possible consequences and potential impacts of security failure of the various LoSAs.

Table 2 – Overall concept of the LoSA of DLT

Possible consequences of security failure	**Potential impact of security failure by LoSA**		
	1	**2**	**3**
Inconvenience, distress or damage to standing or reputation	Low	Substantial	High
Financial loss or agency liability	Low	Substantial	High
Harm to the organization, its programs or public interests	N/A	Low/Substantial	High
Personal safety	N/A	Low	Substantial/High
Civil or criminal violations	Min	Low/Substantial	High
NOTE – N/A=Not Applicable; Low=Low; Sub=Substantial; High=High			

Determination of what constitutes low, substantial, and high risk depends on the risk criteria defined by the organization using this Recommendation for each of the possible consequences.

The risk assessment of a transaction may be conducted as a part of an organization's overall information security risk assessment (e.g., [b-ISO/IEC 27001]) and should focus on the specific need for security in the transactions being contemplated. The risk assessment shall address risk related to the security assurance component. The results of the risk assessment shall be compared to the three LoSAs. The LoSA that best matches the results of the risk assessment shall be selected.

7 DLT actors

The actors involved in the DLT include users, DLT nodes, DLT service providers, and user groups. These actors may belong to a single organization or separate organizations. Figure 1 illustrates actors in DLT.

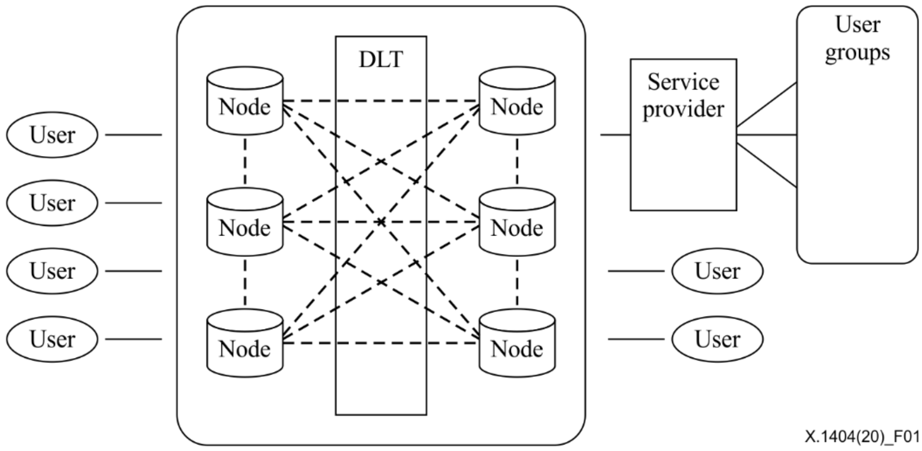

Figure 1 – Actors in DLT

A node is an individual system within the distributed ledger. Some of nodes are called a full node which stores the ledger data, passes along the data to other nodes, and ensures that newly added blocks are valid. A service provider is a component that offers a DLT based service to another by means of one of its provided service interfaces. A user is a component that consumes or uses a service provided by another component. A component may be a provider of some services and a consumer of others. A user group is a set of users in the DLT system.

A distributed ledger is an electronic data that has been replicated, shared, synchronized and stored by consensus in physically separate multiple places (e.g., states, organizations, etc.).

8 Security assurance for DLT

Clause 8 specifies criteria and guidelines for achieving each of three levels of a certain security assurance component: data integrity, data confidentiality, credential management, identity proofing of users, entity authentication, access control, data obfuscation, consensus mechanism strength, smart contract and PII data protection.

The criteria are incremental, that is, criteria for the lower level of security assurance will be included those for the higher level of security assurance.

8.1 Data integrity

There are many consensus algorithms (e.g., proof of stake, proof of work, etc.) that help provide agreement that a set of transactions is valid. Integrity of a distributed ledger could be preserved by using a consensus algorithm and its associated hash function, which is a critical component to implement consensus mechanism. Hashing is a process of calculating a relatively unique output (also referred to as a hash digest) for an input of nearly any size (e.g., a file, text, image, etc.). Hash functions are designed to be one-way; calculating the hash digest of an input is simple but reconstructing the input from the digest is significantly more difficult and collision-resistant so that it is computationally infeasible to find two inputs which result in the same digest. Additionally, the smallest change of input, even a single bit, will result in a completely different output hash digest.

A block consisting of hash chains is an append-only data structure where data are bundled into blocks that include a hash digest of the previous block's data within the newest block. This data structure provides evidence of tampering because any modification to a block's data will change the hash digest recorded by the following block.

A Merkle tree [b-Merkle] is a data structure where the data are hashed and combined until there is only a singular root hash that represents the entire structure.

Blocks are chained together through each block containing the hash value of the former block in every block, thus forming the distributed ledgers. If a previously published block was tampered with, it would have a different hash. This in turn would cause all subsequent blocks to also have different hashes since they include the hash of the previous block. This makes it possible to easily detect and reject any changes to previously published blocks by checking the hash value.

There are three types of threats to data integrity in a DLT system:

1) an attacker violates data integrity by directly altering (part of) the DLT system;

2) a set of federated members updates blocks without informing the other members;

3) multiple federations of members collude to a malicious altering of (part of) the DLT system.

Once data has been written in a block, if we assume a majority of nodes or stakeholders are honest, the probability of data alteration decreases exponentially over time. Data integrity is indeed strictly related to this probability. The less likely that data can be tampered with, the stronger the integrity guarantees that can be claimed. However, being dependent on time introduces critical risks to data integrity. Data integrity on DLT cannot be simply seen as a simple property, but it should be as a more complex, quantitative concept. This amounts to taking multiple factors into account, including the time elapsed and parties' awareness.

The assurance level of data integrity mainly depends on the strength of hash functions which are used to calculate the Merkle tree.

LoSA for data integrity can apply to any type of DLT: permission or permissionless DLT. Table 3 provides a LoSA for data integrity.

Table 3 – LoSA for data integrity

Security assurance level for data integrity	
LoSA[DI] 1	• The Merkle tree is used to generate a hash structure in a hierarchical manner for all transaction data. • Hash functions for generating Merkle tree root data, which is publicly available and safe from existing cryptanalysis attacks are used to generate the Merkle tree's root data.
LoSA[DI] 2	• The data in each block of a DLT database is stamped with a time stamp. • The digital signature is used to prove the origin and integrity of the data and protect the recipient node of the data against forgery of third parties. • Hash functions and digital signature which are publicly available and safe from existing cryptanalysis attacks and are selected from lists of approved hash functions by authoritative entities are used.
LoSA[DI] 3	• Hash functions and digital signature which are which are secure against both quantum and classical computers are used. The digital signature is used to protect the sender node against forgery of the recipient node. • It is recommended for multi-signature with more than one key to be used to authorize a DLT transaction.

8.2 Data confidentiality

The assurance level of data confidentiality depends on the strength of the cryptographic algorithm. If encrypted data are stored in the distributed ledger, the strength of the cryptography algorithm is critical to data confidentiality.

Table 4 provides a LoSA for data confidentiality. LoSA for data confidentiality can apply to any type of distributed ledger technology: permission or permissionless DLT. A risk assessment is carried out to identify the required level of protection which will help determine the necessary type, strength and quality of cryptographic algorithm to be used.

Table 4 – LoSA for data confidentiality

Security assurance level for data confidentiality	
LoSA[DC] 1	• Symmetric encryption is used to defend against data leakage risk. • Encryption algorithms which are publicly available and safe from existing well-known cryptanalysis attacks are used.
LoSA[DC] 2	• Asymmetric encryption is used based on the public and private key of the node. • Encryption algorithms which are publicly available and safe from existing well-known cryptanalysis attacks and are selected from lists of approved algorithms by authoritative entities are used.
LoSA[DC] 3	• Encryption algorithms which are secure against both quantum and classical computers are used.

8.3 Credential management

A credential is an object or data structure that binds an identity - via an identifier or identifiers - and (optionally) additional attributes, to at least one authenticator possessed and controlled by a DLT user.

LoSA for credential management can apply to any type of distributed ledger technology: permission or permissionless DLT. Table 5 provides a LoSA for DLT credential management.

Table 5 – LoSA for credential management

Security assurance level for credential management	
LoSA[CM] 1	• Credential information such as private key is managed in secure storage methods. • Credentials are stored in clear text, hard-coded in the software, or stored in plain-text on local machines.
LoSA[CM] 2	• Credentials are stored in cipher-text on local machines encrypted with a password or other external storage device encrypted with a password.
LoSA[CM] 3	• Credentials are stored in a hardware security module (HSM) with additional access control such as fingerprint.

At LoSA[CM]3, credentials are stored in a hardware security module (HSM) to achieve a high level of assurance. This is required when a large scale of damage is expected. LoSA[CM]3 is highly resistant to key extortion attacks or key disclosure. For example, a credential that is used on a DLT associated with expensive transactions may require LoSA[CM]3. Key pairs can be generated or stored within HSM. When an application raises a query for the use of the private key, the HSM performs the query within the HSM, then sends the output of the query.

8.4 Identity proofing of user

Identity proofing (IP) establishes that a subject is who they claim to be. In case IP is required especially for the permissioned distributed ledger, Table 6 provides a LoSA for DLT IP.

Table 6 – LoSA for identity proofing of user

Security assurance level for identity proofing	
LoSA[IP] 1	• There is no requirement to link the applicant to a specific real-life identity. • Any attributes provided in conjunction with the subject's activities are self-asserted or are to be treated as self-asserted. • Self-asserted attributes are neither validated nor verified.
LoSA[IP] 2	• Either remote or in-person identity proofing is required. • Identifying attributes to have been verified in person or remotely. • There is evidence that supports the real-world existence of the claimed identity and verifies that the applicant is appropriately associated with this real-world identity.
LoSA[IP] 3	• In-person identity proofing is required. • There is physical presence which is required for identity proofing. • Identifying attributes are verified by an authorized and trained credential service provider representative.

8.5 Entity authentication

Digital authentication is the process of determining the validity of one or more authenticators used to claim a digital identity. in the case entity authentication is required, especially for a permissioned DLT system, Table 7 provides a LoSA for DLT authentication.

Table 7 – LoSA for entity authentication

Security assurance level for entity authentication	
LoSA[EA] 1	• This level should provide some assurance that the claimant controls an authenticator registered to the subscriber. • The nodes in the distributed ledger can be authenticated by a trusted third party. • A single factor authentication mechanism is required.
LoSA[EA] 2	• This level should provide some assurance that the claimant controls an authenticator registered to the subscriber. • A trusted certificate authority (CA) can be used to issue certificates for every node joining the DLT system to prevent Sybil attack. • A two-factor authentication mechanism using a wide range of available authentication technologies is required.

Table 7 – LoSA for entity authentication

Security assurance level for entity authentication	
LoSA[EA] 3	• This level should provide some assurance that the claimant controls an authenticator registered to the subscriber. • Nodes can store some trustworthy Internet protocol addresses and deploy a mechanism to check the misbehaving nodes in the DLT network to prevent an eclipse attack. • A two-factor authentication mechanism with the use of tamper-resistant hardware devices for the storage of all secret or private cryptographic keys is used for entity authentication.

For LoSA for entity authentication, a single factor authentication mechanism is LoSA[EA]1 regardless of the use of tamper-resistant hardware devices.

8.6 Access control

Access control is a term used to describe the capability to restrict the use of services accessing data to users who have been previously authorised. Authorization is the right or permission that is granted to an entity to access a system resource. In the case access control is required to write data in the ledgers, especially for the permissioned distributed ledger, Table 8 provides a LoSA for DLT access control.

Table 8 – LoSA for access control

Security assurance level for access control	
LoSA[AC] 1	• A simple access control list is used for accessing distributed ledgers.
LoSA[AC] 2	• A role-based access control is used to allow access of the node to the permissioned distributed ledger.
LoSA[AC] 3	• A fine-grained access control, such as attributed control mechanisms is used to allow access of the node to the distributed ledger based on the attributed control mechanism.

8.7 Data obfuscation

De-identification is a general term for any process of removing the association between a set of identifying attributes and the data subject. In case where data obfuscation is required, Table 9 provides a LoSA for DLT data obfuscation.

Table 9 – LoSA for data obfuscation

Security assurance level for data obfuscation	
LoSA[DO] 1	• De-identified data are stored in the distributed ledgers.
LoSA[DO] 2	• Encrypted de-identified data are stored in the distributed ledgers.

Table 9 – LoSA for data obfuscation

Security assurance level for data obfuscation	
LoSA[DO] 3	• Hashed value of data and link to the real data are stored in on-chain in the distributed ledgers. • The real data are stored off-chain and it stores encrypted PII with a strong encryption algorithm.

8.8 Consensus mechanism

A consensus mechanism is the rules and procedures by which consensus is reached.

A key aspect of DLT technology is determining which user validates the next block. This is achieved through implementing one of many possible consensus mechanisms. For permissionless DLT networks, there are generally many nodes competing at the same time to validating the next block.

There are typical examples of consensus mechanisms: Proof of work, proof of stake, and Byzantine fault tolerant-based [b-NISTIR 8202].

In the proof of work (PoW) consensus model, a node validates the next block by being the first to solve a computationally intensive puzzle. The solution to this puzzle is the "proof" they have performed the work. In PoW, any node (miner) needs to solve a difficult cryptographic puzzle to produce a hash value nonce which is smaller than the whole information in order to collect transactions and/or propose a block which will be potentially included in the distributed ledger. The probability of validating a new block depends on the instantaneous computational power devoted to the task. As a reward for validating a block, the node (miner) will receive a certain amount of crypto asset and transaction fees.

A proof of stake is a consensus process, where an existing stake in distributed ledger system (e.g., the amount of that currency that a user holds) is used to reach consensus. The proof of stake (PoS) is based on the idea that the more stakes that a user has invested into the system, the more likely they will want the system to succeed, and the less likely they will want to subvert it. For example, the stake is often an amount of crypto asset that the DLT user has invested into the system.

In a distributed ledger system, some nodes may behave abnormally and thus cause a Byzantine fault to the system. A Byzantine fault tolerant (BFT)-based consensus algorithm is designed and implemented to solve the Byzantine fault problem, and it ensures the distributed ledger system functions normally even with abnormal nodes involved in the network. In BFT-based consensus, all nodes in the network need to participate in the consensus process which performs multiple rounds of voting and communication to reach consensus on a block. So, it is more compatible with small systems with limited nodes. Meanwhile, since BFT requires that all participants agree on the list of participants in the network, the protocol is normally only used in permissioned blockchains. For permissioned DLT, the BFT-based consensus algorithm is deterministic, and it is a more conventional approach, compared with the PoW consensus mechanism in permissionless DLT. A BFT-based distributed ledger system can offer better consistency and lower latency.

LoSA for DLT consensus mechanism can apply to any type of distributed ledger technology: permission or permissionless DLT. Table 10 provides a LoSA for DLT consensus mechanism.

Table 10 – LoSA for DLT consensus mechanism

Security assurance level for consensus mechanism	
LoSA[CMS] 1	• It is required for a consensus mechanism to detect tampering. • It is recommended for DLT consensus mechanisms to expand the computing power or stake resources as much as possible to prevent and to defend against 51% attack.
LoSA[CMS] 2	• A consensus mechanism to resist tampering is required. • Data consistency and accuracy are maintained when the number of malicious nodes (referred to as the Byzantine fault tolerance) does not exceed the theoretical value. • It is recommended for DLT consensus mechanisms to have a capability to defend against self-mining attack, such as Freshness preferred, randomly choose, and ZeroBlock described in clause 9.4.3 of [ITU-T X.1402].
LoSA[CMS] 3	• Specific solution is used to identify data tampering when the number of malicious nodes exceeds the theoretical value. • It is required for DLT consensus mechanisms to have a capability to defend double-spending attack, such as such as listening periods, inserting observers, and forwarding double spending attempts in clause 9.4.4 of [ITU-T X.1402].

8.9 Smart contract

A smart contract is a program written on the distributed ledger system which encodes the rules for specific types of distributed ledger system transactions in a way that can be validated, and triggered by specific conditions (e.g., Ethereum's smart contracts, Hyperledger Fabric's chain-code). The smart contract is automatically executed by nodes within the distributed ledger system. The results of the execution are validated by consensus and recorded on the distributed ledger.

Smart contract automation reduces costs, lowers risks of errors, mitigates risks of fraud and potentially streamlines many business processes.

LoSA for DLT smart contracts can apply to any type of distributed ledger technology: permission or permissionless DLT. Table 11 provides a LoSA for DLT smart contract.

Table 11 – LoSA for DLT smart contract

Security assurance level for smart contract	
LoSA[SC] 1	• The smart contract is required to be developed to comply with secure design and secure coding practices as defined in [ISO/IEC 27034-3]. The smart contract should be checked with open source based publicly available security audit tool.
LoSA[SC] 2	• The smart contract developer should be checked with static and dynamic analysis-based security audit tool. • The smart contract is required to be securely verified before installation into nodes, e.g., installation package code signature verification. • It is recommended to customize highly controlled or simplified virtual machines to run smart contracts by having a capability such that accessing system resources, accessing memory directly and interacting with file systems should be forbidden in the customized virtual machine, described in clause 9.5.4 of [ITU-T X.1402].

Table 11 – LoSA for DLT smart contract

Security assurance level for smart contract	
LoSA[SC] 3	• The smart contract developer should be checked with a dynamic analysis security audit tool recommended by publicly authorized organizations.

8.10 Personally identifiable information (PII) data protections

In theory, a distributed ledger system can have two types of storage to store PII, on-chain storage and off-chain storage, although on-chain storage is strongly discouraged.

When PII data are stored on-chain, it may lead to negative effects, i.e., direct or indirect identification of a PII principal, as the distributed ledger increases in size and as transactions are added, and the accumulated data within the distributed ledger itself is used to link to external databases. In addition, advanced analysis and profiling capabilities can also lead to direct or indirect identification of a PII principal.

An intermediary practice holds that symmetric encryption is mandated by the system's design to store data on the distributed ledger [b-FG DLT D4.1]. As the on-chain data can be accessed by anyone who has permission for the ledger, this case introduces a risk of data breach stored in DLT exposed to threats such as loss of keys, brute force attack or cryptanalysis attack.

Where PII is held off-chain, privacy and PII protection can be addressed by adopting the [b-ISO/IEC 29100] approach. DLT systems typically use hashes of data to allow the actual data to be held off-chain while a record of the data, confirming the existence of the data at a certain moment in time and its provenance and authenticity and enabling verification of its integrity, is held on the distributed ledger. This facilitates large related data to be held off-chain while the integrity of the data referenced is maintained using the hashing function on the data. Identifiers can be used to point to PII held off-chain, where these identifiers are not derived from the data itself and can probably only have a one-way relationship.

LoSA for DLT PII data protection can apply to any type of distributed ledger technology: permission or permissionless DLT. Table 12 provides a LoSA for DLT PII data protection.

Table 12 – LoSA for DLT PII data protection

Security assurance level for data protection	
LoSA[DP] 1	• The capability of DLT system is to store PII in the off-chain with appropriate access control and to store hashed value of that PII in the on-chain.
LoSA[DP] 2	• The capability of DLT system is configured to store data containing PII encrypted off-chain with access control.
LoSA[DP] 3	• The capability of DLT system is configured to store data containing PII encrypted off-chain with granular access control on stored data

9 Threats and capabilities related to DLT security assurance

The threats and controls for distributed ledger technology are described in [ITU-T X.1401] and [ITU-T X.1402], respectively. The specific threat is described in clause 5 of [ITU-T X.1401], and the capability in clause 9 of [ITU-T X.1402]. The relation between threats and security assurance components are described in Annex A.

Annex A

Mapping between threat items and security assurance components

(This annex forms an integral part of this Recommendation.)

Table A.1 provides mapping between security assurance components in this Recommendation and the threat items in [ITU-T X.1401].

Table A.1 – Mapping between security assurance components and threat items

Assurance component in this Recommendation	Threat item in [ITU-T X.1401]
8.1 Data integrity	6.1.1 Consensus mechanism threat 6.1.3 Virtual machine threat 6.1.4 Cryptographic hash algorithm threat 6.1.5 Asymmetric cryptographic algorithm threat 6.1.6 Threat from practical Quantum computers 6.1.2 Smart contract threats
8.2 Data confidentiality	6.1.5 Asymmetric cryptographic algorithm threat 6.1.6 Threat from practical Quantum computers 6.3.4 Transaction threats
8.3 Credential management	6.3.1 Account data and transaction data threat 6.3.2 Private key leakage threat 6.3.3 Private key loss threat
8.4 Identity proofing of users	6.2.3 Node identity threats
8.5 Entity authentication	6.2.1 Node routing table threat 6.2.3 Node identity threat 6.3.1 Account data and transaction data threat 6.3.2 Private key leakage threat
8.6 Access control	6.3.1 Account data and transaction data threat
8.7 Data obfuscation	6.3.1 Account data and transaction data threat
8.7 Consensus mechanism	6.1.1 Consensus mechanism threats
8.9 Smart contract	6.1.2 Smart contract threat 6.1.3 Virtual machine threats
8.10 PII data protection	6.3.1 Account data and transaction data threat

Table A.2 provides mapping between security assurance components in this Recommendation and the capabilities in [ITU-T X.1402].

Table A.2 – Mapping between security assurance components and security capabilities

Assurance components in this Recommendation	Security capabilities in [ITU-T X.1402]
8.1 Data integrity	9.2.1 Merkle tree 9.2.2 Time stamp 9.2.3 Digital signature 9.4.1 Consensus mechanism 9.5.3 Multi-signature
8.2 Data confidentiality	9.2.4 Data encryption
8.3 Credential management	9.2.5 Security storage
8.4 Identity proofing of user	9.5.1 Identity Authentication
8.5 Entity authentication	9.3.1 Routing attack defence 9.3.2 Sybil attack defence 9.3.3 Eclipse attack defence 9.5.3 Multi-signature
8.6 Access control	9.5.2 Authorization
8.7 Data obfuscation	9.5 Application security
8.8 Consensus mechanism	9.4.1 Consensus mechanism 9.4.2 51% attack defence 9.4.3 Selfish mining attack defence 9.4.4 Double-spending attack defence
8.9 Smart contract	9.5.4 Smart contract security design
8.10 PII data protection	9.2.1 Merkle tree 9.2.4 Data encryption

Bibliography

[b-ITU-T X.800]	Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications.*
[b-ITU-T X.1311]	Recommendation ITU-T X.1311 (2011), *Information technology – Security framework for ubiquitous sensor networks*.
[b-ITU-T X.1400]	Recommendation ITU-T X.1400 (2020), *Terms and definitions for distributed ledger technology*.
[b-ISO/IEC TR 15443-1]	ISO/IEC TR 15443-1:2012, *Information technology – Security techniques – Security assurance framework – Part 1: Introduction and concepts*.
[b-ISO/IEC 24760-1]	ISO/IEC 24760-1:2019, *IT Security and Privacy – A framework for identity management – Part 1: Terminology and concepts*.
[b-ISO/IEC 27000]	ISO/IEC 27000:2016, *Information technology – Security techniques – Information security management systems – Overview and vocabulary*.
[b-ISO/IEC 27001]	ISO/IEC 2700:2013, *Information security management*.
[b-ISO/IEC 29100]	ISO/IEC 29100:2011, *Information technology – Security techniques – Privacy framework.*
[b-ISO 13491-1]	ISO 13491-1:2016, *Financial services – Secure cryptographic devices (retail) – Part 1: Concepts, requirements and evaluation methods.*
[b-ISO 22739]	ISO 22739:2020, *Blockchain and distributed ledger technologies – Vocabulary*.
[b-ISO/TC 307]	ISO/TC 307:2016, *Blockchain and distributed ledger technologies*.
[b-DFS-Glossary]	ITU-T Focus Group Digital Financial Services, Digital Financial Services (DFS) Glossary, 2017.1.
[b-ENISA]	ENISA, *Distributed Ledger Technology & Cybersecurity Improving Information Security in the Financial Sector*, December 2016.
[b-FG DLT D4.1]	Focus Group on Application of Distributed Ledger Technology (2019), *Distributed Ledger Technology Regulatory Framework.* https://www.itu.int/en/ITU-T/focusgroups/dlt/Pages/default.aspx
[b-Merkle]	Merkle tree at https://en.wikipedia.org/wiki/Merkle_tree.
[b-NISTIR 8202]	NISTIR 8202:2018, *Blockchain Technology Overview*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2021