International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1402

(07/2020)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

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

Security framework for distributed ledger technology

Recommendation ITU-T X.1402

International Telecommunication Union

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.1402

Security framework for distributed ledger technology

Summary

Distributed ledger technology (DLT) is usually seen as a peer –to-peer distributed ledger based on a group of technologies for a new generation of transactional applications, which maintains a continuously growing list of cryptographically secured data records against tampering and revision. DLT can help establish trust, accountability, transparency and efficiency while streamlining business processes.

However, DLT is also facing security challenges and threats specific to DLT systems and DLT application scenarios. Based on analysis of security threats and security requirements to DLT, Recommendation ITU-T X.1402 describes security capabilities that could mitigate the related security threats and specifies a security framework methodology to determine how to use these security capabilities to mitigate security threats to a specific DLT system.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1402	2020-07-22	17	11.1002/1000/14251

Keywords

Blockchain, distributed ledger technologies, security framework, security threats.

* 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

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.1402

Security framework for distributed ledger technology

1 Scope

This Recommendation lists security threats to distributed ledger technology (DLT) and analyses security requirements and security capabilities that could mitigate these threats. However, detailed description of security threats to DLT lies outside the scope of this Recommendation. A security framework methodology is provided to give guidance on how to use security capabilities to mitigate or defend against security threats to DLT applications and services.

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.1158] Recommendation ITU-T X.1158 (2014), *Multi-factor authentication mechanisms using a mobile device*.

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

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 address [b-ITU-T FG DLT D1.1]: Identifier for entity(ies) performing transactions or other actions in a blockchain or distributed ledger network.

3.1.2 application [b-ITU-T Y.2091]: A structured set of capabilities, which provide value-added functionality supported by one or more services, which may be supported by an API interface.

3.1.3 block [b-ITU-T FG DLT D1.1]: individual data unit of a blockchain (see 3.1.4), composed of a collection of transactions and a block header.

NOTE – A block may be considered immutable and considered as a digital entity described in clause 3.2.2 in [b-ITU-T X.1255], however, it can be applied to other networks or other computational facilities.

3.1.4 blockchain [b-ITU-T FG DLT D1.1]: 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.

3.1.5 capability [b-ITU-R M.1224]: The ability of an item to meet a service demand of given quantitative characteristics under given internal conditions.

3.1.6 consensus [b-ITU-T FG DLT D1.1]: Agreement that a set of transactions is valid.

3.1.7 distributed ledger [b-ITU-T FG DLT D1.1]: A type of ledger that is shared, replicated, and synchronized in a distributed and decentralized manner.

3.1.8 identity [b-ITU-T X.1257]: Set of attributes related to an entity.

NOTE – Within a particular context, an identity may have one or more identifiers to allow an entity to be uniquely recognized within that context.

3.1.9 Merkle tree [b-NISTIR 8202]: A data structure where the data is hashed and combined until there is a singular root hash that represents the entire structure.

3.1.10 node [b-ITU-T FG DLT D1.1]: Device or process that participates in a distributed ledger network.

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

3.1.11 private key [b-ITU-T X.509]: (In a public-key cryptosystem) that key of an entity's key pair which is known only by that entity.

3.1.12 public key [b-ITU-T X.509]: That key of an entity's key pair which is publicly known.

3.1.13 public distributed ledger system [b-ISO 22739]: DLT system which is accessible to the public for use.

3.1.14 private distributed ledger system [b-ISO 22739]: DLT system that is accessible for use only to a limited group of DLT users.

3.1.15 smart contract [b-ITU-T FG DLT D1.1]: 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.

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

3.1.17 transaction [b-ITU-T FG DLT D1.1]: Whole of the exchange of information between nodes. A transaction is uniquely identified by a transaction identifier.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 keystore: A file, which is encrypted with a password, used to store private key entries or certificate entries.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CA Certification Authority

CAS Component Attack Surface

CC Component Class

CD Component Domain

CV Component Vulnerability

DDoS Distributed Denial of Service

DHT Distributed Hash Table

DLT Distributed Ledger Technology

DLTS DLT System

DLTS-P DLTS Protection

DLTS-R DLTS Risk

DPoS Delegate Proof of Stake

HSM Hardware Security Module

ID Identifier

IP Internet Protocol

KYC Know Your Customer

MFA Multi-Factor Authentication

PBFT Practical Byzantine Fault Tolerance

P2P Peer-to-Peer

PoW Proof of Work

PoS Proof of Stake

ZKP Zero-Knowledge Proof

5 Conventions

None.

6 Overview

DLT is a digital system for recording and sharing data across multiple nodes. Each node in DLT contains the exact same data record. Unlike traditional databases, there is no central administrator or centralized data storage in DLT. According to a series of cryptographical solutions, DLT provides integrity, veracity and consistency of data in different nodes that is maintained by anonymous participants without any need for trust across one or more institutions. DLT is used in many fields based on its characteristics, e.g., implementing DLT in a logistics information-tracing scenario, the information-tracing system can take each node in a logistics flow as a DLT node and store the corresponding logistics information at each node to prevent information from being tampered with and guarantee the credibility of the information.

Although DLT is popular due to its features for establishing trust, accountability and transparency, there are security threats in adopting DLT. Some threats are directed at networks, others at data. Therefore it is necessary and useful to summarize security requirements in different categories based on analyses of security threats. Based on analysis of these threats and requirements, a set of high-level security capabilities is identified. In addition, this Recommendation provides a security framework methodology on how to use these security capabilities to mitigate security threats to a DLT system.

7 Security threats to DLT

This clause analyses security threats in different DLT application scenarios.

a) Private key leakage

A private key is used to sign data and confirm the ownership of the data. When the private key of a node is stolen, the identity of the node may be counterfeited.

A detailed threat description is provided in clause 6.3.2 of [ITU-T X.1401].

b) Data leakage

Data in DLT nodes may be searched or accessed by an unauthorized entity.

A detailed threat description is provided in clause 6.3.1 of [ITU-T X. 1401].

c) Distributed denial of service (DDoS) attack

The nodes may be subject to DDoS attacks resulting in a malfunction.

A detailed threat description is provided in clause 6.2.2 of [ITU-T X.1401].

d) 51% Attack

By controlling 51% of the computing resources of an entire network, a new fork of the blockchain can be created that replaces the original fork of blockchain as the main blockchain.

A detailed threat description is provided in clause 6.1.1 of [ITU-T X.1401].

e) Double-spending attack

A double-spending attack leads to an incorrect transaction. By controlling 51% of the computing power of an entire network, a new fork can be created and the transaction can experience rollback. Then the transaction needs the same money twice and the attacker can steal the money of the first transaction.

A detailed threat description is provided in clause 6.1.1 of [ITU-T X.1401].

f) Selfish mining attack

A selfish mining attack is a method for mining pools to waste other nodes computing resources and increase malicious node returns by unfair means.

A detailed threat description is provided in clause 6.1.1 of [ITU-T X.1401].

g) Sybil attack on network

A Sybil attack on a network is one where a malicious node weakens or destroys the redundant backup mechanism by simulating or controlling multiple nodes.

A detailed threat description is provided in clause 6.2.3 of [ITU-T X.1401].

h) Routing attack

In the networks that use a distributed hash table (DHT), the attacker pretends to be a normal node of the network, and forwards other nodes' routing requests to non-existent or incorrect nodes.

A detailed threat description is provided in clause 6.2.4 of [ITU-T X.1401].

i) Smart contract attack

Each node in the network runs programs on its own local machine, this entails some obvious security risks.

A detailed threat description is provided in clause 6.1.2 of [ITU-T X.1401].

j) Eclipse attack

An eclipse attack allows an attacker controlling a sufficient number of Internet protocol (IP) addresses to monopolize all connections to and from a victim DLT node. The attacker can then exploit the victim for attacks on DLT mining and consensus systems, including *n*-confirmation double spending, selfish mining, and adversarial forks in the DLT system.

A detailed threat description is provided in clause 6.2.1 of [ITU-T X.1401].

8 Security requirements for DLT

Based on the analysis of security threats outlined in clause 7, different security threats have various targets, e.g., some threats are directed at networks, others at data. According to different targets to which security threats are directed, security requirements are analysed from the following four aspects: data security, network security, consensus security and application security.

8.1 Data security

Security of private keys, transaction data and privacy data are all parts of data security. Data security has two meanings:

– the security of data itself, which mainly refers to using cryptographic algorithms to achieve data authenticity, confidentiality and integrity;
– the security of data protection, which mainly refers to using secured data storage to protect data.

8.2 Network security

Routing, Sybil, eclipse and DDoS attacks are threats to peer-to-peer (P2P) networks. Network security is the basis of a secure and healthy DLT system, because P2P networks form the infrastructure of DLT systems. Security mechanisms should be provided to reduce the possibility of all four types of attack.

8.3 Consensus security

A consensus mechanism is a protocol that ensures all nodes are synchronized with each other and agree on which transactions are legitimate and are added to the DLT system. Without a good consensus algorithm at the centre of the consensus mechanism, the DLT system is at risk of various attacks. However, a consensus algorithm alone is not enough. Attack strategies such as 51%, selfish mining and double-spending make use of the vulnerabilities of the consensus mechanism. Specific security mechanisms should be provided to resist 51%, double-spending and selfish mining attacks, in addition to the DLT consensus algorithm.

8.4 Application security

There are many different DLT application security scenarios. Different applications have various user authentication and authorization requirements. Additional application security mechanisms should be provided to resist smart contract attacks.

9 Security capabilities

9.1 Security capabilities diagram

This Recommendation identifies security capabilities against identified DLT threats. Some capabilities can be supplied by more than one part of a DLT independently or co-ordinately. Figure 1 shows an overall description of security capabilities that DLT should have.

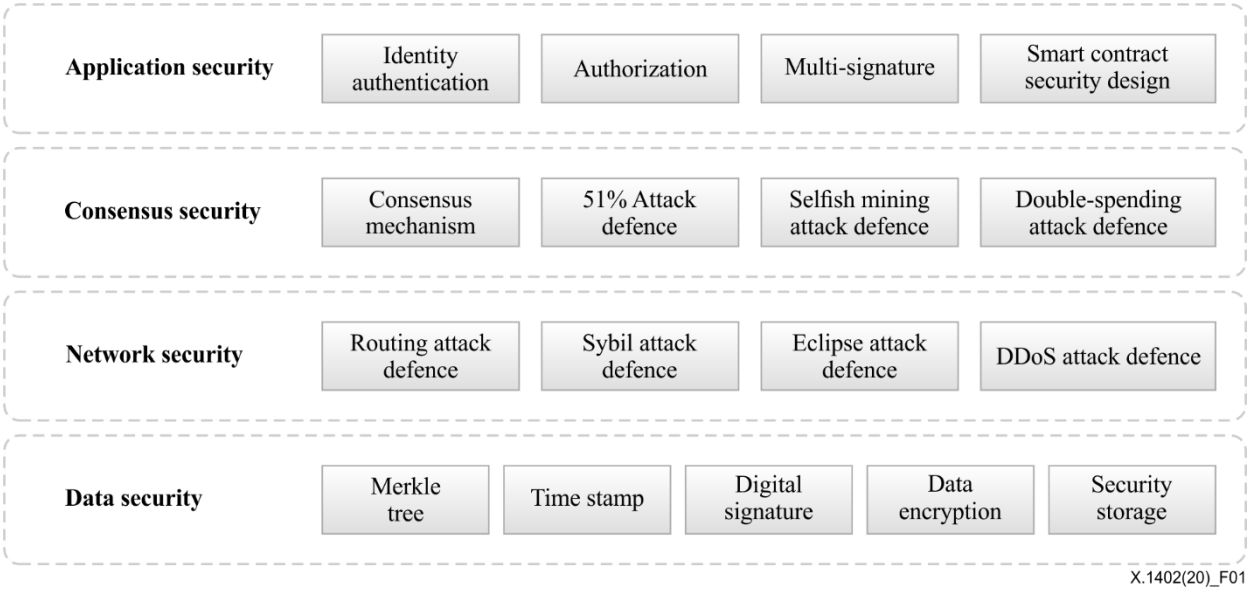

Figure 1 – Security capabilities for DLT

9.2 Data security

9.2.1 Merkle tree

A Merkle tree [b-Merkle] is used to generate a hash structure in a hierarchical manner for all transaction data, which can ensure data integrity. The root of the Merkle tree is saved in each block, and any change in the data will result in the change of the root. To ensure integrity of a transaction data or a set of transaction data, a hash value (or simply hash), also called a message digest, is computed by a cryptographic hash algorithm that maps a string of arbitrary length to a bit string of a fixed length.

9.2.2 Time stamp

Data in each block of a DLT database can be given a time stamp, which is a kind of encrypted voucher document that can indicate the data that has already existed at a particular time.

9.2.3 Digital signature

A digital signature is a cryptographic transformation of data that allows a node receiving it to prove the origin and integrity of the data, protect the sender node and the recipient node of the data against forgery by third parties and protect the sender node against forgery of the recipient node.

Data sent by a node should be hashed and then signed by the private key of the node. The signature should be verified before the data is written into the blockchain to ensure data integrity.

9.2.4 Data encryption

Data encryption is a cryptographic transformation of data that is useful for defending against data leakage risk.

A data encryption and decryption scheme can be designed based on the public and private key of the node, which can ensure data confidentiality.

DLT uses asymmetric encryption to generate data signatures and generate each user address to ensure transaction security. However, the transaction data is completely transparent and not encrypted. In order to adapt to applications that require high data security, it is necessary to opt for a data encryption operation.

9.2.5 Security storage

A private key needs a set of secure storage methods.

A private key is used to sign the information exchanged in a DLT system. It can be stored in the hardware security module (HSM). The security of HSMs is very high and can prevent the private key from unauthorized access.

A private key can be stored in a keystore file that is encrypted with a password and then stored in the local file system. When users need to use it, they should enter a password to decrypt the keystore and get the private key.

Secret sharing refers to methods for distributing a secret among a group of participants, each of whom is allocated a share of the secret. Secret sharing in a DLT wallet can protect a private key from malware because attackers need to compromise more participants than the threshold to steal the private key.

Cold wallets are not, and have never been, connected to the Internet (offline) or have been created using a computer that has never been connected to the network If the computer is not connected to the Internet, attackers cannot steal the private key without physically touching the computer. This method uses two computers (the second of which has to be disconnected from the Internet) and a new private key is generated using the wallet software. A part of the data is sent to this new wallet using

the private key of a user. If the computer is not connected to the Internet, attackers cannot steal the private key without physically touching the computer.

9.3 Network security

9.3.1 Routing attack defence

In a private distributed ledger system, it is recommended that the nodes in the DLT be authenticated by a trusted third party, which guarantees the credibility of the nodes and the routing information sent by the nodes.

9.3.2 Sybil attack defence

A trusted certification authority (CA) can be used to issue a certificate for every node when it joins the DLT system for its identity authentication. The certificate contains the identifier (ID), IP address and public key of the node and the certificate is signed with the private key of the CA. To prevent an attacker from having a large number of certificates, there are two solutions: One is to charge for a certificate to increase the cost of the attack; the other is to bind the node ID to the identity of the node owner in the real world.

9.3.3 Eclipse attack defence

To prevent an eclipse attack, nodes can store some trustworthy IP addresses and deploy a mechanism to check the misbehaving nodes in the network. The IP addresses that misbehave in the network could be banned from connection. In addition, nodes should check on incoming and outgoing connections to reduce the effect of an eclipse attack.

9.3.4 DDoS attack defence

An anti-DDoS system can be implemented to defend against attack. An anti-DDoS system can filter data packets, for example, according to IP address. It can also detect the content of data packets to judge whether the data stream is normal and then prohibit the abnormal data stream.

Sufficient bandwidth should be reserved to resist DDoS attacks.

9.4 Consensus security

9.4.1 Consensus mechanism

A consensus mechanism is an algorithm for reaching consensus on the sequence of transactions within the same time window.

A consensus mechanism is used to select which nodes construct the block and to make data records of each node consistent in the DLT. A consensus mechanism is used to compile a correct record of data and blocks. The correct data is recorded on the DLT, as long as non-malicious computing resources or encrypted token resources (depending on the consensus mechanism adopted) of the whole network are the majority. At present, there are many consensus mechanisms: proof of work (PoW), proof of stake (PoS), delegate proof of stake (DPoS), practical Byzantine fault tolerance (PBFT), etc.

9.4.2 51% Attack defence

In order to defend against 51% attack, it is recommended that computing power or other stake resources be expanded as much as possible. For example, for a PoW consensus mechanism, the greater the computing power of the whole DLT system, the higher the cost for the attacker to implement the 51% attack.

9.4.3 Selfish mining attack defence

To implement a selfish mining attack needs control over a large amount of computing power. Therefore one of the solutions to defend against selfish mining is to expand the computing power as

much as possible. The greater the computing power of the whole DLT network, the less the attacker is likely to implement a selfish mining attack.

There are other methods to mitigate selfish mining attacks. For example:

Freshness preferred: The whole DLT network chooses blocks with recent timestamps to extend the blockchain. This will make selfish mining attackers lose block races against newly mined blocks.

Randomly choose: If there is a blockchain with multiple forks and each fork has the same length, the fork to be extended should be chosen randomly. This will decrease the probability that other miners extend the fork of the selfish pool.

ZeroBlock: If a selfish miner keeps a mined block privately, exceeding a specified interval, when this block is published later on the network, it will be rejected by honest miners. This will prevent the attacker from withholding blocks.

9.4.4 Double spending attack defence

The most effective way to prevent double spending in a PoW consensus system is to wait for multiple numbers of confirmations before delivering goods or services to the payee. In particular, the possibility of a successful double spending attack decreases with increases in the number of confirmations received. This limits attackers from possible revisions of the history of transactions in the blockchain.

Techniques such as listening period, inserting observers, and forwarding double spending attempts can also be used to detect double spending attacks. In the listening period technique, the vendor monitors all transactions received during a listening period and only delivers the product if there is no attempt at double spending. In the inserting observers technique, the vendor inserts a set of observers that will directly relay all transactions to the vendor that they receive from the network. In this way, the vendor is able to see greater numbers of transactions in the network during its listening period. Thus, the chances of detecting a double spending attack are increased. The forward double spending attempts technique requires each DLT node to forward, instead of discarding, all transactions that attempt a double spending attack so that the vendor can be notified when one occurs. This will increase the chances of detecting a double spending attack. However, while all the existing solutions make double spending attacks harder to carry out, they cannot eliminate them.

9.5 Application security

9.5.1 Identity authentication

DLT uses a private key to authenticate a user, which avoids binding user personally identifying information.

In some important scenarios, such as banks, "know your customer" (KYC) validation is required, allowing participants to create and manage their own identity, and authorize other participants to access this identity.

Multi-factor authentication (MFA) [ITU-T X.1158] is a method of confirming a user's claimed identity, to whom access is granted only after successfully presenting two or more credential factors to an authentication mechanism.

In order to minimize exposure of sensitive information, DLT can make use of privacy-enhancing cryptographic schemes such as zero-knowledge proof (ZKP) protocols [b-ISO/IEC 20008-2]. A ZKP protocol allows users to prove their identity without exposure of the identity information.

9.5.2 Authorization

Authorization is used to achieve authorization management for different users. Public and private distributed ledger systems have different scopes of authorization. Each node of the DLT is connected

through a P2P network, and each new added node will synchronize all the data on the DLT. The DLT data is fully open for each node, and the node can freely view any transaction information in any block. Therefore, in some scenarios involving data privacy protection, it is necessary to implement authorization management.

9.5.3 Multi-signature

Multi-signature requires more than one key to authorize a DLT transaction. Multi-signature allows the creation of 2-of-3 escrow services. For example: when Alice wants to pay Bob, she sends a transaction to a multi-signature address, which requires at least two signatures from the group "Alice, Bob and Trent" to redeem the money. Although an attacker can compromise one of the three, the attacker cannot authorize a DLT transaction with just one signature.

9.5.4 Smart contract security design

To resist smart contract attacks against virtual machine vulnerabilities, there is a need to customize highly controlled or simplified virtual machines to run smart contracts. Some related functions, such as accessing system resources, accessing memory directly and interacting with file systems should be forbidden in the customized virtual machine.

For example, in order to achieve high-level security for a virtual machine, some projects build new instead of using existing virtual machines such as Java. The operation codes related to accessing system resources, accessing memory directly and interacting with the file system do not exist in the new virtual machine.

10 Security framework methodology

A security framework for DLT is an approach developed to mitigate DLT security threats. There are different security threats to various DLT applications. It is necessary to understand which security threats exist to develop a security framework for a chosen specific DLT application.

The first step is to analyse security threats as introduced from a technical perspective in clause 7. The second step is based on security threat analysis results; other business and regulatory requirements should be taken into consideration to identify security requirements as described in clause 8. The third step is to use security threats and security requirements as inputs to identify the security capabilities as described in clause 9.

It is not possible to provide one common framework to fit all DLT applications. The following steps can be used to develop a security framework for a specific DLT application:

– step 1: identify security threats to the specific DLT application according to clause 7;
– step 2: identify security requirements according to clause 8, based on security threats identified by step 1, and take other specific business and regulatory requirements into consideration;
– step 3: identify security capabilities as described in clause 9, using security threats identified by step 1 and requirements identified by step 2 as inputs.

The four kinds of security requirement listed in clause 8 correspond to the four kinds of security capabilities that are outlined in clause 9.1 and described in detail in clauses 9.2 to clause 9.5, respectively.

Once security capabilities have been identified, security controls, policies and procedures can be determined accordingly. The determination and implementation of the relevant security controls, policies and procedures lie outside the scope of this Recommendation.

Besides the security capabilities described in clause 9, a DLT application needs other common security capabilities, e.g., physical security, operational security and incident management, to

guarantee its security. These common security capabilities lie outside the scope of this Recommendation.

Appendix I provides an example of a security framework analysis for commodity tracing as a service on a private distributed ledger system.

Appendix I

Example of security framework analysis for commodity tracing as a service on a private distributed ledger system

(This appendix does not form an integral part of this Recommendation.)

A commodity tracing service based on DLT ensures that commodity-related information is not tampered with during commodity circulation, so as to achieve commodity information traceability. The related information includes: product barcode, logistics information and quality inspection information. To identify which security threats are relevant to the commodity tracing service, each one should be reviewed. One approach could be as simple as a table showing a 'Y' (for Yes) next to the threat. Tables I.1, I.2 and I.3 show steps 1, 2 and 3, respectively, of a security framework analysis for commodity tracing as a service on a private distributed ledger system.

Table I.1 – Step 1 of a security framework analysis for commodity tracing as a service on a private distributed ledger system

Security requirement	Security threat	Is this threat applicable to this service?
Data (clause 8.1)	Private key leakage (clause 7 a))	Y
	Data leakage (clause 7 b))	Y
Network (clause 8.2)	DDoS attack (clause 7 c))	
	Sybil attack on network (clause 7 g))	Y
	Routing attack (clause 7 h))	Y
	Eclipse attack (clause 7 j))	
Consensus (clause 8.3)	51% Attack (clause 7 d))	
	Double-spending attack (clause 7 e))	
	Selfish mining attack (clause 7 f))	
Application (clause 8.4)	Smart contract attack (clause 7 i))	Y

Table I.2 – Step 2 of a security framework analysis for commodity tracing as a service on a private distributed ledger system

Specific business and regulatory requirements	Security requirement	Is this requirement applicable to this service?
1. Mutual authentication and authorization should be supported between nodes. 2. Nodes should be able to store the ledger records uninterrupted.	Data (clause 8.1)	Y
	Network (clause 8.2)	
	Consensus (clause 8.3)	Y
	Application (clause 8.4)	Y

Table I.3 – Step 3 of a security framework analysis for commodity tracing as a service on a private distributed ledger system

		Security capabilities																
Security requirement	Security threat	Merkle tree (clause 9.2.1)	Time stamp (clause 9.2.2)	Digital signature (clause 9.2.3)	Data encryption (clause 9.2.4)	Security storage (clause 9.2.5)	Routing attack defence (clause 9.3.1)	Sybil attack defence (clause 9.3.2)	Eclipse attack defence (clause 9.3.3)	DDoS attack defence (clause 9.3.4)	Consensus mechanism (clause 9.4.1)	51% Attack defence (clause 9.4.2)	Selfish mining attack defence (clause 9.4.3)	Double-spending attack defence (clause 9.4.4)	Identity Authentication (clause 9.5.1)	Authorization (clause 9.5.2)	Multi-signature (clause 9.5.3)	Smart contract security design (clause 9.5.4)
Data (clause 8.1)	Private key leakage (clause 7 a))				Y	Y												
	Data leakage (clause 7 b))	Y	Y	Y	Y	Y												
Network (clause 8.2)	DDoS attack (clause 7 c))																	
	Sybil attack on network (clause 7 g))							Y										
	Routing attack (clause 7 h))						Y											
	Eclipse attack (clause 7 j))																	
Consensus (clause 8.3)	51% Attack (clause 7 d))										Y							
	Double-spending attack (clause 7 e))										Y							
	Selfish mining attack (clause 7 f))										Y							
Application (clause 8.4)	Smart contract attack (clause 7 i))														Y	Y		Y

Appendix II

A risk, target and protection relationship model for DLT

(This appendix does not form an integral part of this Recommendation.)

In Appendix III of [ITU-T X.1401], a DLT system (DLTS) is composed of component domains (CDs) – application, service, protocol, network and data.

The risk to target and protection relationship model at the system level is illustrated in Figure II.1 and Table II.1. The model represents the risks to the target on the left side as DLTS risk (DLTS-R) and protection from security countermeasures from those threats on the right side as DLTS protection (DLTS-P). The model indicates an intrinsic link between a threat as an attack exploit to a component weakness as a vulnerability and one or more security countermeasures that either detect the attack (detective control) or mitigate it (preventative control).

Appendix III of [ITU-T X.1401] covers DLTS and DLTS-R, the centre and left side of Figure II.1 and Table II.1. This appendix covers the corresponding DLTS-P, the right side of Figure II.1.

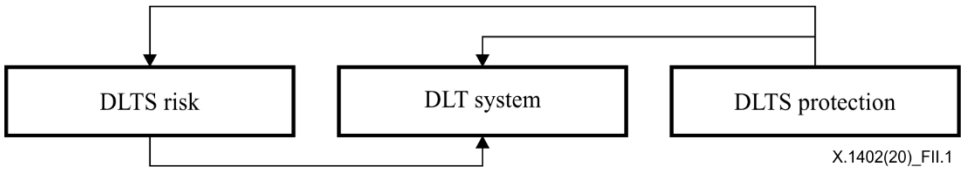

Figure II.1 – DLTS risk to DLT system and DLT protection

Each domain decomposes further until a single target component can be defined at the vulnerability level, where the attack occurs, and countermeasures are discussed.

For example, CDs represented at level 3 decompose into component classes (CCs) at level 4, which decomposes into a singular component (CO) at level 5, which has one or more component vulnerabilities (CVs) at level 6. This is illustrated in Table II.1.

Table II.1 – Different levels of DLTS risk, target and protection

LEVEL	RISK Nomenclature	TARGET Nomenclature	Protection Nomenclature
1	**DLT ECOSYSTEM RISK** (DLTE-R)	**DLT ECOSYSTEM** (DLTE)	**DLT ECOSYSTEM PROTECTION** (DLTE-P)
2	**DLT SYSTEM RISK** (DLTS-R)	**DLT SYSTEM** (DLTS)	**DLT SYSTEM PROTECTION** (DLTS-P)
3	**THREAT DOMAIN** (DLTS-R-TD)	**COMPONENT DOMAIN** (DLTS-R-CD)	**SECURITY DOMAIN** (DLTS-P-SD)
4	**THREAT CLASS** (DLTS-R-TC)	**COMPONENT CLASS** (DLTS-R-CC)	**SECURITY CLASS** (DLTS-P-SC)
5	**COMPONENT THREATS** (DLTS-R-CT)	**COMPONENT** (DLTS-R-CO)	**COMPONENT SECURITY** (DLTS-P-CS)
6	**COMPONENT VULNERABILITY ATTACK** (DLTS-R-CVA):	**COMPONENT VULNERABILITY** (DLTS-R-CV):	**COMPONENT VULNERABILITY ATTACK COUNTERMEASURE** (DLTS-P-CVAC):

Level 6 is where an attack exploit occurs on a CV and it is also where the countermeasure is applied. Using the nomenclature of Table II.1 row 6, a CV is threatened by its CV attack and responds with a CV attack countermeasure.

The one or more level 6 vulnerabilities of a component defines its component attack surface (CAS). Different threat vectors exploit the various vulnerabilities of a component. The CAS will be completely addressed by sequencing through each vulnerability and looking at the threat and countermeasure relationship.

Table II.1 can be expanded to incorporate all DLTS CDs of application, service, protocol, network and data, as shown in Table II.2. This nomenclature allows for the creation of precise relationships.

Table II.2 – Expansion of Table II.1 to incorporate application, service, protocol, network and data CDs

1.0	DLT SYSTEM ECOSYSTEM						
ID	Domain	DLTS RISK Nomenclature		DLTS TARGET Nomenclature		DLTS Protection Nomenclature	
2.0		DLTS-X Risk	DLTS-R	DLT X System	DLTS	DLTS-X-Protection	DLTS-P
3.0		THREAT DOMAIN (TD)		COMPONENT DOMAIN (CD)		Security Domain (SD)	
3.1	Application	Application TD	DLTS-R-ATD	Application CD	DLTS-X-ACD	Application SD	DLTS-P-ASD
3.2	Service	Service TD	DLTS-R-STD	Service CD	DLTS-X-SCD	Service SD	DLTS-P-SSD
3.3	Protocol	Protocol TD	DLTS-R-PTD	Protocol CD	DLTS-X-PCD	Protocol SD	DLTS-P-PSD
3.4	Network	Network TD	DLTS-R-NTD	Network CD	DLTS-X-NCD	Network SD	DLTS-P-NSD
3.5	Data	Data TD	DLTS-R-DTD	Data CD	DLTS-X-DCD	Data SD	DLTS-P-DSD
3.6	Physical	Physical TD	DLTS-R-PhTD	Physical CD	DLTS-X-PhCD	Physical SD	DLTS-P-PhSD
4.0		THREAT CLASS (TC)		COMPONENT CLASS (CC)		Security Class (SC)	
4.1	Application	Application TC	DLTS-R-ATD-ATC	Application CC	DLTS-X-ACD-ACC	Application SC	DLTS-P-ASD-ASC
4.2	Service	Service TC	DLTS-R-STD-STC	Service CC	DLTS-X-SCD-SCC	Service SC	DLTS-P-SSD-SSC
4.3	Protocol	Protocol TC	DLTS-R-PTD-PTC	Protocol CC	DLTS-X-PCD-PCC	Protocol SC	DLTS-P-PSD-PSC
4.4	Network	Network TC	DLTS-R-NTD-NTC	Network CC	DLTS-X-NCD-NCC	Network SC	DLTS-P-NSD-NSC
4.5	Data	Data TC	DLTS-R-DTD-DTC	Data CC	DLTS-X-DCD-DCC	Data SC	DLTS-P-DSD-DSC
4.6	Physical	Physical TC	DLTS-R-PhTD-PhTC	Physical CC	DLTS-X-PhCD-PhCC	Physical SC	DLTS-P-PhSD-PhSC
5.0		COMPONENT THREAT (CT)		COMPONENT (CO)		COMPONENT SECURITY (CS)	
5.1	Application	Application CT	DLTS-R-ATD-ATC-ACT	Application Component	DLTS-X-ACD-ACC-AC	Application ST	DLTS-P-ASD-ASC-AST
5.2	Service	Service CT	DLTS-R-STD-STC-SCT	Service Component	DLTS-X-SCD-ACC-SC	Service ST	DLTS-P-SSD-SSC-SST
5.3	Protocol	Protocol CT	DLTS-R-PTD-PTC-PCT	Protocol Component	DLTS-X-PCD-ACC-PC	Protocol ST	DLTS-P-PSD-PSC-PST
5.4	Network	Network CT	DLTS-R-NTD-NTC-NCT	Network Component	DLTS-X-NCD-ACC-NC	Network ST	DLTS-P-NSD-NSC-NST
5.5	Data	Data CT	DLTS-R-DTD-DTC-DCT	Data Component	DLTS-X-DCD-ACC-DC	Data ST	DLTS-P-DSD-DSC-DST
5.6	Physical	Physical CT	DLTS-R-PhTD-PhTC-PhCT	Physical Component	DLTS-X-PhCD-ACC-PhC	Physical ST	DLTS-P-PhSD-PhSC-PhST
6.0		COMPONENT VULNERABILITY ATTACK (CVA)		COMPONENT VULNERABILITY (CV)		CVA COUNTERMEASURE (CVAC)	
6.1	Application	Application CVA	DLTS-R-ATD-ATC-ACT-ACVA	Application CV	DLTS-X-ACD-ACC-AC-ACV	Application CVAC	DLTS-P-ASD-ASC-AST-ACVAC
6.2	Service	Service CVA	DLTS-R-STD-STC-SCT-SCVA	Service CV	DLTS-X-SCD-ACC-SC-SCV	Service CVAC	DLTS-P-SSD-SSC-SST-SCVAC
6.3	Protocol	Protocol CVA	DLTS-R-PTD-PTC-PCT-PCVA	Protocol CV	DLTS-X-PCD-ACC-PC-PCV	Protocol CVAC	DLTS-P-PSD-PSC-PST-PCVAC
6.4	Network	Network CVA	DLTS-R-NTD-NTC-NCT-NCVA	Network CV	DLTS-X-NCD-ACC-NC-NCV	Network CVAC	DLTS-P-NSD-NSC-NST-NCVAC
6.5	Data	Data CVA	DLTS-R-DTD-DTC-DCT-DCVA	Data CV	DLTS-X-DCD-ACC-DC-DCV	Data CVAC	DLTS-P-DSD-DSC-DST-DCVAC
6.6	Physical	Physical CVA	DLTS-R-DTD-DTC-DCT-DCVA	Physical CV	DLTS-X-PhCD-ACC-PhC-PhCV	Physical CVAC	DLTS-P-PhSD-PhSC-PhST-PhCVAC

An example of the nomenclature for a protocol domain is illustrated in Table II.3, which is extracted from Table II.2.

Table II.3 – Protocol domain extracted from Table II.2

1.0	DLT SYSTEM ECOSYSTEM						
ID	Domain	DLTS RISK Nomenclature		DLTS TARGET Nomenclature		DLTS Protection Nomenclature	
2.0		DLTS-X Risk	DLTS-R	DLT X System	DLTS	DLTS-X-Protection	DLTS-P
3.0		THREAT DOMAIN (TD)		COMPONENT DOMAIN (CD)		Security Domain (SD)	
3.3	Protocol	Protocol TD	DLTS-R-PTD	Protocol CD	DLTS-X-PCD	Protocol SD	DLTS-P-PSD
4.0		THREAT CLASS (TC)		COMPONENT CLASS (CC)		Security Class (SC)	
4.3	Protocol	Protocol TC	DLTS-R-PTD-PTC	Protocol CC	DLTS-X-PCD-PCC	Protocol SC	DLTS-P-PSD-PSC
5.0		COMPONENT THREAT (CT)		COMPONENT (CO)		COMPONENT SECURITY (CS)	
5.3	Protocol	Protocol CT	DLTS-R-PTD-PTC-PCT	Protocol Component	DLTS-X-PCD-ACC-PC	Protocol ST	DLTS-P-PSD-PSC-PST
6.0		COMPONENT VULNERABILITY ATTACK (CVA)		COMPONENT VULNERABILITY (CV)		CVA COUNTERMEASURE (CVAC)	
6.3	Protocol	Protocol CVA	DLTS-R-PTD-PTC-PCT-PCVA	Protocol CV	DLTS-X-PCD-ACC-PC-PCV	Protocol CVAC	DLTS-P-PSD-PSC-PST-PCVAC

Based on the specific selection of a domain (protocol) and attack level (6), Table II.4 illustrates the one to one alignment between all threats targets and countermeasures (risks, targets and protections) in the domain involving both the nomenclature and unique tagging.

Table II.4 – Protocol domain attack layer expanded from Table II.3

COMPONENT THREATS	COMPONENT VULNERABILITY ATTACKS			COMPONENT		COMPONENT VULNERABILITY ATTACK COUNTERMEASURE	
Protocol Component Threats	Protocol Component Vulnerability Attacks	Protocol Component Vulnerability Attack ID:	Attack ID	Protocol Component	Protocol Component ID	Protocol Component Vulnerability Attack Countermeasures	Protocol Component Vulnerability Attack Countermeasure ID
Consensus Mechansim Threats		**DLTS-R-PTD-PTC-PCT:**	CMT	**Consensus Mechansim (CM)**	DLTS-X-PCD-ACC-PC: CM		**DLTS-P-PSD-PSC-PST**
	51% Attack	DLTS-R-PTD-PTC-PCT-PCVA:CMT-	51A	**Consensus Mechansim Protocol**	DLTS-X-PCD-ACC-PC-PCV: CM	51% Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Timestamp Manipulation Attack	DLTS-R-PTD-PTC-PCT-PCVA:CMT-	TMA	**Consensus Mechansim Protocol**	DLTS-X-PCD-ACC-PC-PCV: CM	Timestamp Manipulation Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Bribing Attack	DLTS-R-PTD-PTC-PCT-PCVA:CMT-	BA	**Consensus Mechansim Protocol**	DLTS-X-PCD-ACC-PC-PCV: CM	Bribing Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Selfish Mining Attack	DLTS-R-PTD-PTC-PCT-PCVA:CMT-	SMA	**Consensus Mechansim Protocol**	DLTS-X-PCD-ACC-PC-PCV: CM	Selfish Mining Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Chain Hopping Attack	DLTS-R-PTD-PTC-PCT-PCVA:CMT-	CHA	**Consensus Mechansim Protocol**	DLTS-X-PCD-ACC-PC-PCV: CM	Chain Hopping Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Block Withholding Attack	DLTS-R-PTD-PTC-PCT-PCVA:CMT-	BWA	**Consensus Mechansim Protocol**	DLTS-X-PCD-ACC-PC-PCV: CM	Block Withholding Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Double-Spending Attack	DLTS-R-PTD-PTC-PCT-PCVA:CMT-	DSA	**Consensus Mechansim Protocol**	DLTS-X-PCD-ACC-PC-PCV: CM	Double-Spending Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
Smart Contract Threats		**DLTS-R-PTD-PTC-PCT:**	SCT	**Smart Contract (SC)**	DLTS-X-PCD-ACC-PC: SC		**DLTS-P-PSD-PSC-PST**
	Timestamp Dependence Attack	DLTS-R-PTD-PTC-PCT-PCVA:SCT-	TDA	**Smart Control Protocol**	DLTS-X-PCD-ACC-PC-PCV: SC	Timestamp Dependence Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Mishandled Exceptions Attack	DLTS-R-PTD-PTC-PCT-PCVA:SCT-	MEA	**Smart Control Protocol**	DLTS-X-PCD-ACC-PC-PCV: SC	Mishandled Exceptions Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Integer Overflow Attack	DLTS-R-PTD-PTC-PCT-PCVA:SCT-	OIA	**Smart Control Protocol**	DLTS-X-PCD-ACC-PC-PCV: SC	Integer Overflow Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Predictable Random Number Attack	DLTS-R-PTD-PTC-PCT-PCVA:SCT-	PRNA	**Smart Control Protocol**	DLTS-X-PCD-ACC-PC-PCV: SC	Predictable Random Number Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
Virtual Machine Threats		**DLTS-R-PTD-PTC-PCT:**	VMT	**Virtual Machine (VM)**	DLTS-X-PCD-ACC-PC: VM		**DLTS-P-PSD-PSC-PST**
	Escape Attack	DLTS-R-PTD-PTC-PCT-PCVA:VMT-	EA	**Virtual Machine Protocol**	DLTS-X-PCD-ACC-PC-PCV: VM	Escape Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Fault Handling Attack	DLTS-R-PTD-PTC-PCT-PCVA:VMT-	FHA	**Virtual Machine Protocol**	DLTS-X-PCD-ACC-PC-PCV: VM	Fault Handling Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Memory Corruption Attack	DLTS-R-PTD-PTC-PCT-PCVA:VMT-	MCA	**Virtual Machine Protocol**	DLTS-X-PCD-ACC-PC-PCV: VM	Memory Corruption Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
Cryptographic Hash Algorithm Threats		**DLTS-R-PTD-PTC-PCT:**	CHAT	**Cryptographic Hash Algorithm (CHA)**	DLTS-X-PCD-ACC-PC: CHA		**DLTS-P-PSD-PSC-PST**
	Collision Attack	DLTS-R-PTD-PTC-PCT-PCVA:CHAT-	CA	**Cryptographic Hash Algorithm Protocol**	DLTS-X-PCD-ACC-PC-PCV: CHA	Collision Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Second Preimage Attack	DLTS-R-PTD-PTC-PCT-PCVA:CHAT-	SPA	**Cryptographic Hash Algorithm Protocol**	DLTS-X-PCD-ACC-PC-PCV: CHA	Second Preimage Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Preimage Attack	DLTS-R-PTD-PTC-PCT-PCVA:CHAT-	PA	**Cryptographic Hash Algorithm Protocol**	DLTS-X-PCD-ACC-PC-PCV: CHA	Preimage Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
Asymmetric Cryptographic Algorithm Threats		**DLTS-R-PTD-PTC-PCT:**	ACAT	**Asymmetric Cryptographic Algorithm (ACA)**	DLTS-X-PCD-ACC-PC: ASA		**DLTS-P-PSD-PSC-PST**
	Weak Key Material Attack	DLTS-R-PTD-PTC-PCT-PCVA:ACAT-	WKMA	**Asymmetric Cryptographic Algorithm Protocol**	DLTS-X-PCD-ACC-PC-PCV: ASA	Weak Key Material Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Backdoor Attack	DLTS-R-PTD-PTC-PCT-PCVA:ACAT-	BA	**Asymmetric Cryptographic Algorithm Protocol**	DLTS-X-PCD-ACC-PC-PCV: ASA	Backdoor Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Cracking Attack	DLTS-R-PTD-PTC-PCT-PCVA:ACAT-	CA	**Asymmetric Cryptographic Algorithm Protocol**	DLTS-X-PCD-ACC-PC-PCV: ASA	Cracking Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC
	Protocol Message Manipulation Attack	DLTS-R-PTD-PTC-PCT-PCVA:ACAT-	PMMA	**Asymmetric Cryptographic Algorithm Protocol**	DLTS-X-PCD-ACC-PC-PCV: ASA	Protocol Message Manipulation Attack Countermeasure	DLTS-P-PSD-PSC-PST-PCVAC

For example, taking a single threat-to-target-countermeasure use case: 51% attack on a consensus mechanism described in clause 6.1.1 of [ITU-T X.1401], and 51% attack countermeasures described in clause 9.4.2.

Bibliography

[b-ITU-T X.509] Recommendation ITU-T X.509 (2019) | ISO/IEC 9594-8:2017, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

[b-ITU-T X.1255] Recommendation ITU-T X.1255 (2013), *Framework for discovery of identity management information*.

[b-ITU-T X.1257] Recommendation ITU-T X.1257 (2016), *Identity and access management taxonomy*.

[b-ITU-T Y.2091] Recommendation ITU-T Y.2091 (2011), *Terms and definitions for next generation networks*.

[b-ITU-T FG DLT D1.1] Technical Specification ITU-T FG DLT D1.1 (2019), *Distributed ledger technology terms and definitions*.

[b-ITU-R M.1224] Recommendation ITU-R M.1224 (2011), *Vocabulary of terms for International Mobile Telecommunications (IMT).*

[b-ISO/IEC 20008-2] ISO/IEC 20008-2:2013, *Information technology – Security techniques – Anonymous digital signatures – Part 2: Mechanisms using a group public key*.

[b-ISO 22739] ISO 22739:2020, *Blockchain and distributed ledger technologies – Vocabulary*.

[b-ISO/IEC 27000] ISO/IEC 27000:2018, *Information technology – Security techniques – Information security management systems – Overview and vocabulary*.

[b-Merkle] Merkle, R.C. (1988). A digital signature based on a conventional encryption function. In: Pomerance, C., editor. *Advances in Cryptology – Crypto'87, A Conference on the Theory and Applications of Cryptographic Techniques*, Santa Barbara, CA, 1987-08-16/20, vol. 293 of *Lecture Notes in Computer Science*, pp. 369–378. Berlin: Springer.

[b-NISTIR 8202] Yaga, D., Mell, P., Roby, N., Scarfone, K. (2018). *NISTIR 8202: Blockchain technology overview*, NIST Internal Report 8202. Gaithersburg, MD: National Institute of Standards and Technology. 68 pp. Available [viewed 2020-09-17] at: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

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, 2020