

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1401

(11/2019)

SERIES X: DATA NETWORKS, OPEN SYSTEM
COMMUNICATIONS AND SECURITY

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

Security threats to distributed ledger technology

Recommendation ITU-T X.1401



International
Telecommunication
Union

Documents
Document Sharing Hub

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	X.1700–X.1729

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

Recommendation ITU-T X.1401

Security threats to distributed ledger technology

Summary

Distributed ledger technology (DLT) provides a technical mechanism for achieving verifiable trust through consensus and collective decision-making. DLT also involves maintaining ledgers in a decentralized way and using crypto-mechanisms that can deliver some intrinsic security features. However, a distributed ledger system still has security limitations, for example, confidentiality must be added depending on use case, data sensitivity and applicable data privacy regulations.

Recommendation ITU-T X.1401 provides a structured and systematic threat analysis method to design, implement and operate a distributed ledger technology system and to evaluate its security.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1401	2019-11-29	17	11.1002/1000/14092

Keywords

Distributed ledger technology, DLT.

* 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/11.1002/1000/11.1002/1000/14092>.

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/>.

© ITU 2020

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope.....	1
2 References.....	1
3 Definitions	1
3.1 Terms defined elsewhere	1
3.2 Terms defined in this Recommendation.....	2
4 Abbreviations and acronyms	2
5 Conventions	5
6 Threats to distributed ledger systems	5
6.1 Threats to protocols	5
6.2 Threats to networks	12
6.3 Threats to data	15
7 How stakeholders can use this Recommendation.....	18
Annex A – Security features of DLT	20
Appendix I – DLT threat assessment.....	21
Appendix II – Impact of quantum computing on common cryptographic algorithms	22
Appendix III – Risk DLT system Protection security expression model.....	23
III.1 Abbreviations and acronyms	23
III.2 Risk DLT system Protection security expression model	23
Bibliography.....	34

Introduction

Distributed ledger technology (DLT) achieves trust through collective decision-making and maintaining a single global ledger in a decentralized way.

This Recommendation identifies possible threats to various functional components of a distributed ledger system, such as protocol, network and data. This Recommendation can be considered in the design or implementation of a DLT system as a reference baseline.

Recommendation ITU-T X.1401

Security threats to distributed ledger technology

1 Scope

This Recommendation provides guidance and categorization on security threats to distributed ledger technology (DLT) components. Each of the threats is described in four dimensions:

- targeted component;
- attacks;
- attack impact; and
- attack likelihood.

It also assigns an index to security threats with attack methods or vulnerabilities so that each threat can be referenced by other parts of this Recommendation and other Recommendations.

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.

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 account [b-ITU-T FG DLT D1.1]: Representation of an entity whose data is recorded on a distributed ledger.

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

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

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

3.1.6 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.

3.1.7 fork [b-ITU-T FG DLT D1.1]: Creation of two or more different versions of a distributed ledger.

3.1.8 guideline [b-ISO/IEC 27000]: Description that clarifies what should be done and how, to achieve the objectives set out in policies.

3.1.9 hashing [b-NISTIR 8202]: A method of calculating a relatively unique output (called a *hash digest*) for an input of nearly any size (a file, text, image, etc.). The smallest change of input, even a single bit, will result in a completely different output digest.



Documents

Document Sharing Hub

Rec. ITU-T X.1401 (11/2019)

3.1.10 ledger [b-ITU-T FG DLT D1.1]: Information store that keeps final and definitive (immutable) records of transactions.

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

3.1.12 peer-to-peer [b-ISO 22739]: Relating to, using, or being a network of peers that directly share information and resources with each other without relying on a central entity.

3.1.13 permissioned distributed ledger system [b-ITU-T FG DLT D1.1]: Distributed ledger system in which permissions are required to maintain and operate a node.

3.1.14 permissionless distributed ledger system [b-ITU-T FG DLT D1.1]: Distributed ledger system where permissions are not required to maintain and operate a node.

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

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

3.1.17 proof of work [b-ITU-T FG DLT D1.1]: Consensus process to solve a difficult (costly, time-consuming) problem that produces a result that is easy for others to correctly verify.

3.1.18 risk [b-ITU-T X.1521]: The relative impact that an exploited vulnerability would have to a user's environment.

3.1.19 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.

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

3.1.21 token [b-ITU-T FG DLT D1.1]: A digital representation of value on a shared distributed ledger that is owned and secured using cryptography to ensure its authenticity and prevent modification or tampering without the owner's consent.

3.1.22 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 mining: A reward-seeking activity in some consensus mechanisms, which operates through a demonstration of proof of work.

3.2.2 consensus mechanism (CM): Rules and procedures by which 'consensus', meaning an agreement that a set of transactions is valid, is reached.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AA	Analysis Attack
ACA	Asymmetric Cryptographic Algorithm
ACAT	Asymmetric Cryptographic Algorithm Threats
AN	Authentic Node
ATD	Account Data and Transaction Data
ATDT	Account Data & Transaction Data Threats

BA	Backdoor Attack
BFCA	Brute Force Cracking Attack
BGP	Border Gateway Protocol
CA	Collision Attack
CAS	Component Attack Surface
CC	Component Classes
CHA	Cryptographic Hash Algorithm
CHAT	Cryptographic Hash Algorithm Threats
CM	Consensus Mechanism
CM-51A	51% Attack
CM-BA	Bribing Attack
CM-BWA	Block Withholding Attack
CM-CHA	Chain Hopping Attack
CM-DSA	Double-Spending Attack
CM-SMA	Selfish Mining Attack
CN	Compromised Node
CPA	Cryptographic Protocol Attack
CVSS	Common Vulnerability Scoring System
DA	DDoS Attack
DCA	Data Component Attacks
DCC	Data Component Class
DDoS	Distributed Denial of Service
DDOST	Network DDOS Threat
DLT	Distributed Ledger Technology
DLTN	DLT Network
DLTS	DLT System
DLTS-CC	DLTS Component Classes
DLTS-P	DLTS protections from the threat
DLTS-R	DLTS risks resulting from threats
DSA	Digital Signature Attack
DSEM	DLTS Security Expression Model
EA	Eclipse Attack
ECDSA	Elliptic Curve Digital Signature Algorithm
FNIA	Fraudulent Node Identity Attack
FUD	Forget Unlocking Data
ISP	Internet Service Provider
IP	Internet Protocol

MA	Malware Attack
MCCA	Mathematical Cryptanalysis Cracking Attack
MEA	Mishandled Exceptions Attack
MHR	Merkle Hash Root
MN	Malicious Node
NB	Node Buffer
NCA	Network Components Threats
NCC	Network Component Class
NI	Node Identity
NIM	Node Identity Management
NIT	Node Identity Threats
NNI	Network Node Identity
NR-DA	Network Routing Delayed Attack
NR-PA	Network Routing Partition Attack
NRT	Network Routing Threats
NRTT	Node Routing Table Threat
OIA	Integer Overflow Attack
P2P	Peer-to-Peer
PA	Physical Attack
PCC	Protocol Component Class
PCL	Paper Private Key Code Loss
PIN	Personal Identification Number
PKLeT	Private Key Leakage Threats
PKLoT	Private Key Loss Threats
PMMA	Protocol Message Manipulation Attack
PNC	P2P Network Connection
PoW	Proof of Work
PQC	Practical Quantum Computers
PQCT	Practical Quantum Computers Threats
PrA	Preimage Attack
PrK	Private Key
PRNA	Predictable Random Number Attack
PSD	Public Sensitive Data
PSDA	Public Sensitive Data Attack
SA	Sybil Attack
SBA	Spam Block Attack
SC	Smart Contract

SCA	Software Client Attack
SCT	Smart Contract Threats
SPA	Second Preimage Attack
SPV	Simple Payment Verification
TD	Transaction Data
TDA	Timestamp Dependence Attack
TMA	Timestamp Manipulation Attack
TN	Targeted Node
TT	Transactions Threat
UAA	Unauthorized Access Attack
UL	Unlocking Loss
UT	Unconfirmed Transaction
VM	Virtual Machine
VM-EA	Virtual Machine Escape Attack
VM-FHA	Fault Handling Attack
VM-MCA	Memory Corruption Attack
VMT	Virtual Machine Threats
WKMA	Weak Key Material Attack
ZKP	Zero-Knowledge Proof

5 Conventions

None.

6 Threats to distributed ledger systems

Threats to distributed ledger systems are firstly categorized by the protocol, network or data layer which they affect, then further decomposed according to components of each layer. Each threat is specified by identifying the targeted component, acronyms of attacks specific on the component, impact and likelihood of the threat. Other Recommendations can refer to an attack with the acronym as an index.

6.1 Threats to protocols

Protocols are the configurations that drive network behaviour. Consequently, any unauthorized modification of protocols can have a significant impact on distributed ledger system behaviour.

The threats to protocols can be further decomposed into following six groups:

- Consensus mechanism;
- Smart contract;
- Virtual machine;
- Cryptographic hash algorithm;
- Asymmetric cryptographic algorithm; and
- Practical quantum computers.



6.1.1 Consensus mechanism threats

Targeted component: Consensus mechanism (CM)

The target under consideration is the consensus mechanism which is the rules and procedures by which consensus is reached.

Consensus mechanism attacks:

There are many different consensus mechanisms each focused on reaching different forms of consensus such as proof of work, stake, elapsed time, activity, importance, weight, capacity, etc. Weaknesses of consensus mechanism can be exploited by following attacks:

- **51% attack (CM-51A):** In the distributed ledger system using competition to achieve consensus, if an attacker controls enough DLT nodes and therefore proof of competition, it may have the ability to revoke or rewrite the distributed ledger system Ledger by creating a new fork in the chain and replacing the authentic fork.
- **Timestamp manipulation attack (CM-TMA):** The attacker chooses a malicious timestamp in a block producing instead of the system time to spoof the consensus mechanism to cause the block with the correct timestamp to be refused by other nodes, thereby gaining the advantage.
- **Bribing attack (CM-BA):** A briber owning enough resource entices other participants to take special actions in order to attack the blockchain by providing extra profit such as purchasing computing power or tokens.
- **Selfish mining attack (CM-SMA):** In the use case of proof of work (PoW) consensus mechanism, a malicious node attempts to waste other node's computing resources in following ways: When the malicious node finds a new effective block, the malicious node does not broadcast the block to other nodes immediately, but continues to mine proof of work blocks before a new block is produced by other nodes. When other authentic nodes mine a new block, the malicious node will broadcast the block immediately to the network causing the block mined by other authentic nodes to be considered invalid [b-ARXIV2013].
- **Chain hopping attack (CM-CHA):** In a chain hopping attack, the attackers (miners) switch between various blockchains to maximize their profits by taking advantage of the difficult adjustment algorithms of the chain. This will lead to the loss of honest miners' awards and make the block generating time become very unstable.
- **Block withholding attack (CM-BWA):** A malicious node withholds a newly found block from other miners' nodes. The malicious node attempts to mine in secret and let the rest of the other mining nodes work on a block that will end up rejected and orphaned. Another block withholding attack (BWA) is that mining nodes in a mining pool submit only partial PoW solutions (he/she drops the newly found valid block rather than report it to the mining pool). This will waste the resources of other authentic mining nodes.
- **Double-spending attack (CM-DSA):** A group of malicious nodes in a distributed ledger system may be able to change the Ledger transaction history by creating a new fraudulent fork and rolling back the transaction history such that an asset may be spent twice in two transactions, the first one of which being authentic.

Consensus mechanism attack impact:

Successful CM attacks can allow malicious nodes to control the distributed ledger system with the goal to control the propagation of the chain or weaken or destroy the decentralized consensus mechanism operating the chain. An un-authentic fork will allow the attacker to add fraudulent blocks in the fraudulent fork and make legitimate transaction data in the authentic fork invalid.

Consensus mechanism attack likelihood:

Most consensus mechanism attacks need a large amount of resources such as computing power, votes or tokens. The likelihood of launching attacks depends on the resource cost of a potential attack.

6.1.2 Smart contract threats

Targeted component: Smart contract (SC)

Any distributed ledger system which uses smart contract could be threatened. A smart contract (SC) 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. The smart contract is stored in the DLT, encoded as part of a "creation" transaction that introduces a contract to the DLT.

Smart contract attacks:

There are vulnerabilities in smart contract that can be exploited by malicious users. The vulnerabilities include but are not limited to:

- **Timestamp dependence attack (SC-TDA):** A smart contract may use the timestamp as a triggering condition to execute some critical operations, e.g., transferring money between accounts. Attackers can select a specific time stamp which triggers a condition in their favour. This may cause severe economic losses.
- **Mishandled exceptions attack (SC-MEA):** This is a possible vulnerability when a smart contract calls another as a premise. If there is an exception raised (e.g., exceeding call stack limit) in the callee contract but the exception is not handled properly so that the callee does not perform the premise condition and an exception in the callee contract may also not get propagated to the caller. Or the caller contract does not check the execution status (e.g., explicitly check the return value) to verify if the call has been executed properly, the caller may continue to execute the next steps such as sending money.
- **Integer overflow attack (SC-IOA):** In computer programming, an integer overflow occurs when an arithmetic operation attempts to create a numeric value that is outside of the range that can be represented with a given number of bits – either larger than the maximum or lower than the minimum representable value. In smart contracts, integer overflow can result in a lot of tokens being created and owned by the attackers.
- **Predictable random number attack (SC-PRNA):** Smart contracts such as gambling games always use random numbers to determine the outcome of the game. Developers often use block header information to generate "random numbers". However, this "random number" can be expected by a malicious miner, which allows a malicious miner to control the outcome of the game.

Smart contract attack impact:

A successful attack on a smart contract could have the following impact:

Vulnerabilities in smart contracts may be exploited by malicious users to gain profit without following the agreement of related parties. It is difficult to update the smart contract which makes the consequence more durable.

Smart contract attack likelihood:

Vulnerabilities in smart contracts prevail in distributed ledger systems. For example, several design bugs were found as reported by [b-ACM2016]. Lack of formal verification makes it difficult to find the vulnerabilities. The attack difficulty also depends on the condition of vulnerabilities exploitation.

6.1.3 Virtual machine threats

Targeted component: Virtual machine (VM)

Any distributed ledger system which uses smart contract VM could be threatened. The virtual machine (VM) is the platform running different kinds of smart contracts and provides an execution environment such as a sandbox environment.

VM attacks:

The smart contract VMs running on nodes of blockchain receive and deploy the code of smart contracts. There are vulnerabilities in the VMs that can be exploited by malicious users. Vulnerabilities in smart contract VMs exist because of inadequate design and implemented mechanisms. The vulnerabilities include but are not limited to:

- **Escape attack (VM-EA):** The smart contract VM provides a sandbox environment for running byte codes. Users should only execute corresponding code within the sandbox. Attackers can exploit the VM escape vulnerability and seek more execution permissions outside the sandbox which may result in attackers executing other malicious code or impacting contracts in other VMs.
- **Fault handling attack (VM-FHA):** The VM can implement some fault-tolerant handling when aware of malicious data or code. Vulnerabilities in handling may cause some logical problems like a "short address attack".
- **Memory corruption attack (VM-MCA):** Memory corruption occurs in a computer program when the contents of a memory location are modified due to programmatic behaviour that exceeds the intention of the original programmer or program/language constructs; this is termed violating memory security. Although there are many mitigation techniques to memory corruption attacks to make a memory corruption bug harder to be exploited, memory corruption bugs are still very harmful to the distributed ledger system. A memory crash bug in block validation may result in a crash of the entire distributed ledger system. Attackers exploiting memory out of bound write bugs in smart contract VMs may control thousands of nodes across the network in seconds.

VM attack impact:

Vulnerabilities in smart contract VM may be exploited by malicious users to gain profit without following the agreement of related parties, which may cause resource consuming, denial of service, information disclosure or remote code execution and so on.

VM attack likelihood:

Vulnerabilities exploitability of the smart contract VMs varies according to the vulnerabilities. Attackers need to have knowledge of the mechanism and then invoke some specific contract with some specific transaction to exploit the vulnerabilities.

6.1.4 Cryptographic hash algorithm threats

Targeted component: Cryptographic hash algorithm (CHA)

A cryptographic hash algorithm $H: \{0,1\}^* \rightarrow \{0,1\}^L$ is a mathematical algorithm that takes a string of arbitrary length as input and produce a bit string of a fixed length L where L is a fixed non-negative integer. An output of the cryptographic hash algorithm is called the "hash value" (simply "hash") or "message digest".

Cryptographic hash algorithms are used in:

- the generation of the account address from account owner's public key,
- the construction of Merkle tree,