ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation

ITU-T X.1410 (03/2023)

SERIES X: Data networks, open system communications and security

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

---

Security architecture of data sharing management based on the distributed ledger technology

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 (1)	X.1140–X.1149
Application Security (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.1350–X.1369
Intelligent transportation system (ITS) security	X.1370–X.1399
Distributed ledger technology (DLT) security	**X.1400–X.1429**
Application Security (2)	X.1450–X.1459
Web security (2)	X.1470–X.1489
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
Cyber Defence	X.1590–X.1599
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
Data protection	X.1770–X.1789
IMT-2020 SECURITY	X.1800–X.1819

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

Recommendation ITU-T X.1410

Security architecture of data sharing management based on the distributed ledger technology

Summary

Recommendation ITU-T X.1410 specifies a security architecture of data-sharing management based on distributed ledger technologies (DLTs). Based on the architecture, this Recommendation specifies the interfaces between the functional entities and the procedures of data-sharing management based on DLT. Distributed ledger technology is transforming the industries with innovative solutions and changing the way governments, institutions and businesses operate. It provides a solution for securely replicating, sharing and synchronizing data across a distributed computer network, considering its decentralization and tamper-proof features. Current approaches for sharing business data and personally identifiable information (PII) data with companies and digital platforms have led to privacy vulnerabilities from hacks or poor data management. Adopting DLT or blockchain in data-sharing management allows individuals or companies to maintain more direct control over their own confidential information. In the DLT-based solution, only non-PII data, e.g., hashed data values, are stored in the on-chain. PII data about a data owner are stored in the off-chain. A DLT-based solution provides a way that improves the traceability, verifiability and changeability of status of data.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1410	2023-03-03	17	11.1002/1000/15109

Keywords

Data sharing, DLT, security architecture.

* 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/software copyrights, 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 appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.1410

Security architecture of data sharing management based on the distributed ledger technology

1 Scope

This Recommendation specifies a security architecture for data sharing based on DLTs. This Recommendation includes:

– design of security architecture for data sharing based on distributed ledger technology (DLT);

– specification of the logical functions of the data-sharing security architecture;

– specification of the interfaces between the logical functions of the security architecture;

– specification of the procedures for data sharing based on DLT.

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.1408] Recommendation ITU-T X.1408 (2021), *Security threats and requirements for data access and sharing based on the 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 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 certification authority (CA) [b-ITU-T X.509]: An authority trusted by one or more entities to create and digitally sign public-key certificates. Optionally the certification authority may create the users' keys.

3.1.4 data provider [b-ISO/IEC/IEEE 15939]: Individual or organization that is a source of data.

3.1.5 identity [b-ISO/IEC 29100]: Set of attributes which make it possible to identify the personally identifiable information principal.

3.1.6 off-chain [b-ISO 22739]: Related to a blockchain system, but located, performed, or run outside that blockchain system.

3.1.7 on-chain [b-ISO 22739]: Located, performed, or run inside a blockchain system.

3.1.8 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.

3.1.9 public-key infrastructure (PKI) [b-ITU-T X.509]: The infrastructure able to support the management of public keys able to support authentication, encryption, integrity or non-repudiation services.

3.1.10 smart contract [b-ITU-T X.1400]: 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.

3.1.11 symmetric encryption system [b-ISO/IEC 18033-1]: Cryptographic technique used to protect the confidentiality of data, and which consists of three component processes: an encryption algorithm, a decryption algorithm, and a method for generating keys, where the encryption and decryption algorithms make use of the same key.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 data consumer: User that reads data and then processes it to the extent that lexical or coding boundaries are discovered.

NOTE – Adapted from [b-ISO/IEC 20944-1].

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

CA Certification Authority

DLT Distributed Ledger Technology

PII Personally Identifiable Information

PKI Public Key Infrastructure

5 Conventions

In this Recommendation:

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 "**can**" and "**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.

Italics are used in this Recommendation to indicate functions.

6 Architecture of DLT-based data sharing

Distributed ledger technology provides a solution for securely replicating, sharing and synchronizing data across a distributed computer network, considering its decentralization and tamper-proof features. Current approaches for sharing business data and personally identifiable information (PII) data with companies and digital platforms have led to privacy vulnerabilities from hacks or poor data management. Adopting DLT or blockchain in data-sharing management allows individuals or companies to maintain more direct control over their own confidential information. In the DLT-based solution, only non-PII data, e.g., hashed data values, are stored in the on-chain. PII data about a data owner are stored in the off-chain. A DLT-based solution provides a way that improves the traceability, verifiability and changeability of status of data. In this context, this Recommendation specifies the security architecture of data-sharing management based on DLTs. The data sharing management makes use of a symmetric encryption system where the encryption and decryption algorithms use the same key. The security threats are described in [ITU-T X.1408].

This Recommendation is developed based on [ITU-T X.1408]. While [ITU-T X.1408] is designed from a conceptual point of view this Recommendation is developed from the implementation point of view. The mapping of terminology between this Recommendation and [ITU-T X.1408] is shown in Table 1.

Table 1 – Mapping of terminologies between this Recommendation and [ITU-T X.1408]

This Recommendation	[ITU-T X.1408]
Data provider	Data-sharing agent (data owner)
Data consumer	Data consumer client (data processor)
Key storage server	Key management server
Data storage server	Data storage server (data storage service provider)
DLT-based data-sharing management	Data broker

6.1 Overview of functional architecture

Figure 1 illustrates the functional architecture of DLT-based data sharing management from the implementation point of view. This is consistent with the architecture of DLT-based data access and sharing shown in Figure B.1 of [ITU-T X.1408]. The latter one is designed from a conceptual point of view and is a high-level architecture. The functional architecture in Figure 1 consists of the five blue functional components and the three grey functional components.

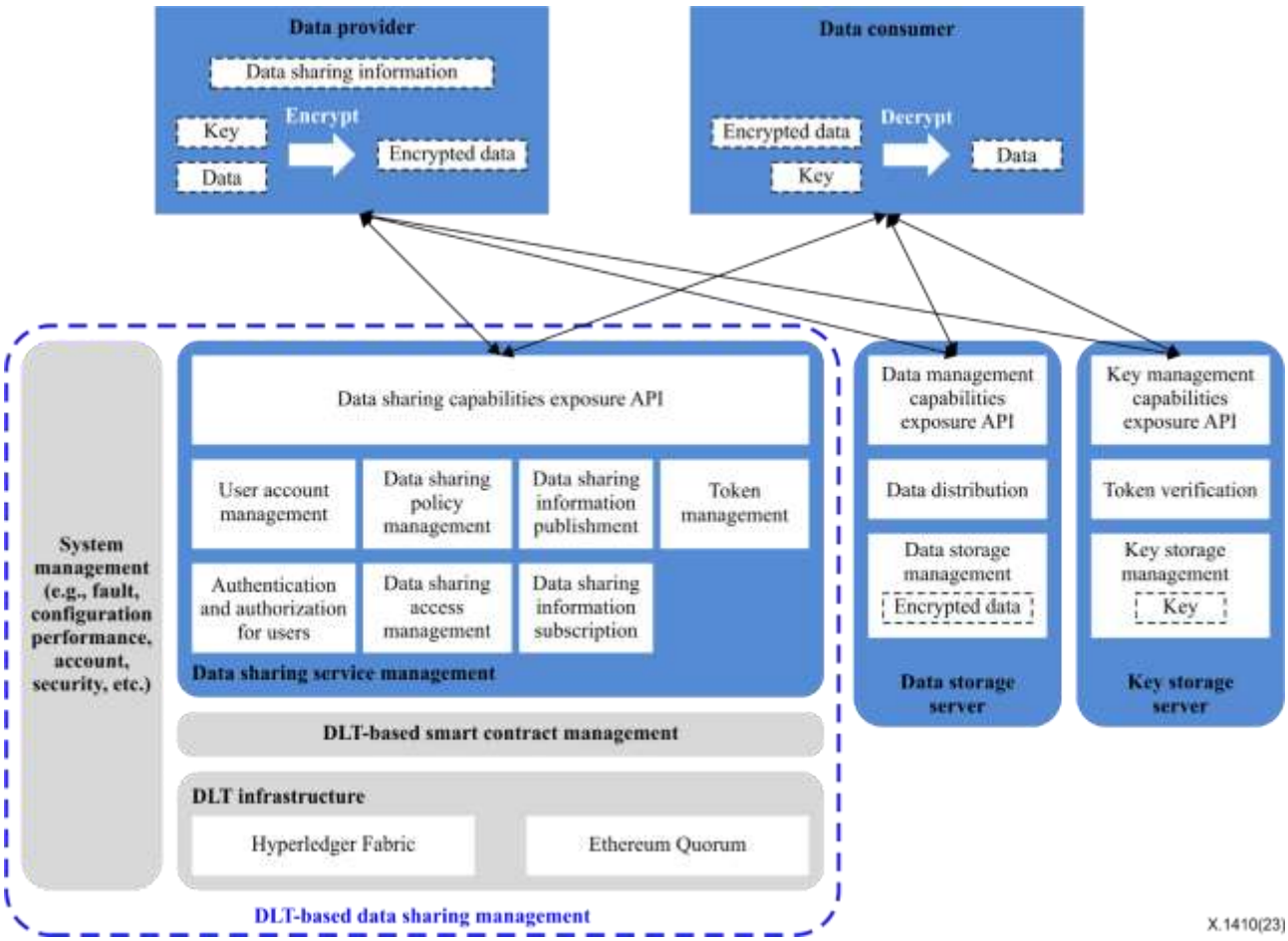

Figure 1 – Functional architecture of DLT-based data-sharing management

These components are described as follows:

– The five blue functional components are *data provider*, *data consumer*, *data-sharing service management*, *key storage server* and *data storage server*, which are defined and described in detail in clause 6.2.

– The three grey functional components are *DLT infrastructure*, *DLT-based smart contract management* and *system management (e.g., fault, configuration, performance, account, security)*, which reuse the existing open source DLT platforms (e.g., Hyperledger Fabric) and lie outside the scope of this Recommendation.

The procedures of DLT-based data-sharing management are described in Annex A.

6.2 Functional components

6.2.1 Data provider and data consumer

This Recommendation introduces two types of user: data provider (i.e., users to share their data) and data consumer (i.e., users to consume the shared data by other users). Both types have the same capabilities:

1) to obtain digital certificates (including public keys) and the corresponding private keys from the certification authority (CA);

2) to store the obtained digital certificates and the corresponding private keys securely;

3) to communicate with the DLT-based data-sharing management server;

4) to get and store the digital certificates, network parameters and network configurations of the DLT network nodes;

5) to receive notifications from DLT-based data-sharing management;

6) to make use of a symmetric encryption system.

Data provider and data consumer have their own specific capabilities as follows.

– A *data provider* has the following capabilities:

  1) to collect raw data and desensitize it (e.g., making data anonymous, removing sensitive information such as name, mobile phone number, credit card data) without having a negative impact on the quality of data to be shared;
  2) to generate a key of a symmetric encryption system;
  3) to encrypt data with the key using symmetric encryption system; ;
  4) to store the ciphertext and the corresponding encryption key securely on the local or remote server;
  5) to ensure mutual authentication with the DLT-based data-sharing management;
  6) to provide to the DLT-based data-sharing management, data-sharing information, which includes data provider identifier, data provider public key, ciphertext identifier and its storage address, the encryption key identifier and its storage address, the industries to be allowed to access, the users to be allowed to access, and other data attributes (e.g., data category, data introduction, usage and price);
  7) to combine data-sharing information and other information to form a data-sharing policy, and then to sign the data-sharing policy with the private key;
  8) to send the data-sharing policy with the corresponding digital signature to the DLT-based data-sharing management;
  9) to receive the notification from the DLT-based data-sharing management, which indicates that the smart contract for data sharing has been created;
  10) to manage the data-sharing policy, such as query and update.

– A *data consumer* has the following capabilities:

  1) to subscribe to published data-sharing messages;
  2) to ensure mutual authentication with the DLT-based data-sharing management;
  3) to obtain data-sharing information;
  4) to send data consumer information (e.g., data consumer identity, data consumer public key) and the shared data information (e.g., data provider identity, data provider public key, ciphertext identifier and encryption key identifier) together with the digital signature (done by data consumer private key) to DLT-based data-sharing management;
  5) from DLT-based data-sharing management, to receive notifications which include shared data information, key storage address, ciphertext storage address and access token;
  6) to send the access token to the key storage server in order to obtain the data encryption key according to the key storage address;
  7) to get the ciphertext according to the data storage address;
  8) to decrypt the ciphertext with the obtained key of the symmetric encryption system to get the data in plaintext;
  9) to manage the records for accessing the shared data.

6.2.2 Data sharing service management

The capabilities of each component of the data-sharing service management are described as follows.

– ***User account management*** has the capabilities to manage user accounts, such as creation, update, query and deletion.

– ***Authentication and authorization for users*** have the following capabilities:

1) to perform mutual authentication with users (i.e., data provider, data consumer);
2) to authorize users to share the data and access the shared data.

– ***Data-sharing policy management*** has the following capabilities:

1) to receive data-sharing information and data-sharing policy together with the corresponding digital signature from data provider;
2) to create a DLT transaction according to the received information from data provider;
3) to submit the DLT transaction to the underlying components (i.e., *DLT infrastructure* and *Smart contract management*) to create a data-sharing smart contract, which will be distributed to the DLT network nodes;
4) to notify the *data provider* and *data-sharing information publication* that the data-sharing smart contract has been created;
5) to enable data providers to manage their shared data.

– ***Data-sharing access management*** has the following capabilities:

1) to receive data access request (including data consumer information, shared data information, and the corresponding digital signature) from data consumer and check whether the data consumer is allowed to access the data according to the data-sharing policy (e.g., only the data consumer from the specified industry or country could access the shared data);
2) to create a DLT transaction according to the received information from data consumer;
3) to submit the DLT transaction to the underlying components (e.g., *DLT infrastructure*) to record the data access, which will be distributed to the DLT network nodes;
4) to notify the *token management* that data access transaction has been created and an access token is to be created;
5) to receive the access token from *token management*;
6) to send data-sharing access information (e.g., access token, key storage address, data storage address) to data consumer;
7) to enable data consumer to manage their data-sharing access records.

– ***Data-sharing information publication*** has the following capabilities:

1) to receive notifications from *data-sharing policy management*, which contains new data-sharing information;
2) to publish new data-sharing information;
3) to notify a *data-sharing information subscription* that new data-sharing information is published.

– ***Data-sharing information subscription*** has the following capabilities:

1) to receive notifications from *data-sharing information management*, which contains new data-sharing information;
2) to send new data-sharing information to the subscribers.

– ***Token management*** has the following capabilities:

1) to receive notifications from *data-sharing access management* to create an access token;

2) to create the access token;

3) to send the access token created to *data-sharing access management*.

– ***Data-sharing capabilities exposure application programming interface (API)*** enables users to manage and access data-sharing services as above.

6.2.3 Key storage server

The capabilities of each component in key storage server are described as follows.

– ***Key storage management*** has the following capabilities:

1) to receive requests from the *data provider* to store the key;

2) to authenticate the *data provider*;

3) to store the key securely, for example, to store the key in cipher text and/or in isolated storage;

4) to notify the *data provider* that the key has been stored;

5) to receive notifications from *token verification*, which indicates the verification result of the access token;

6) to send the key to *data consumer*.

– ***Token verification*** has the following capabilities:

1) to receive the access token from the data consumer;

2) to verify the received access token;

3) to send the verification result to the *key storage management*.

– *A k****ey management capabilities exposure API*** enables users to manage and access a data encryption key.

6.2.4 Data storage server

The capabilities of each component in data storage server are described as follows.

– ***Data storage management*** has the following capabilities:

1) to receive the request from the *data provider* store the ciphertext;

2) to authenticate the *data provider*;

3) to store the ciphertext;

4) to notify the *data provider* that the ciphertext has been stored.

– ***Data distribution*** has the following capabilities:

1) to receive the request from a *data consumer*;

2) to authenticate the *data consumer*;

3) to send the ciphertext to the *data consumer*.

– *A **data management capabilities exposure API*** enables users to manage and access shared data.

7 Security architecture of DLT-based data-sharing management

According to the functional architecture in Figure 1, data consumers request and receive a data encryption key, with which they then decrypt the data in ciphertext and finally obtain the shared data in plaintext. After that, shared data in plaintext cannot be controlled by DLT-based data-sharing management. Data consumers may forward the shared data in plaintext to other users who do not have permission to access it.

In order to support data providers sharing data with others securely and to prevent data consumers from forwarding the shared data in plaintext to other users, the functional architecture in Figure 1 requires enhanced security features, which are shown in Figure 2.

7.1 Overview of security architecture

Figure 2 describes the security architecture of DLT-based data-sharing management, which ensures that:

– the communications between the functional components in Figure 1 are secure;

– data providers encrypt the data and also set the access permission for it before sharing it with others;

– data consumers decrypt the shared data in ciphertext according to the access permission before accessing it;

– data consumers encrypt the shared data in the same way as data providers do after finishing data access, in this way the shared data in ciphertext is stored by data consumers;

– data consumers only forward the shared data in ciphertext to others.

In this way, even if users receive the shared data forwarded by others, they have to request access permission for it from the DLT-based data-sharing management.

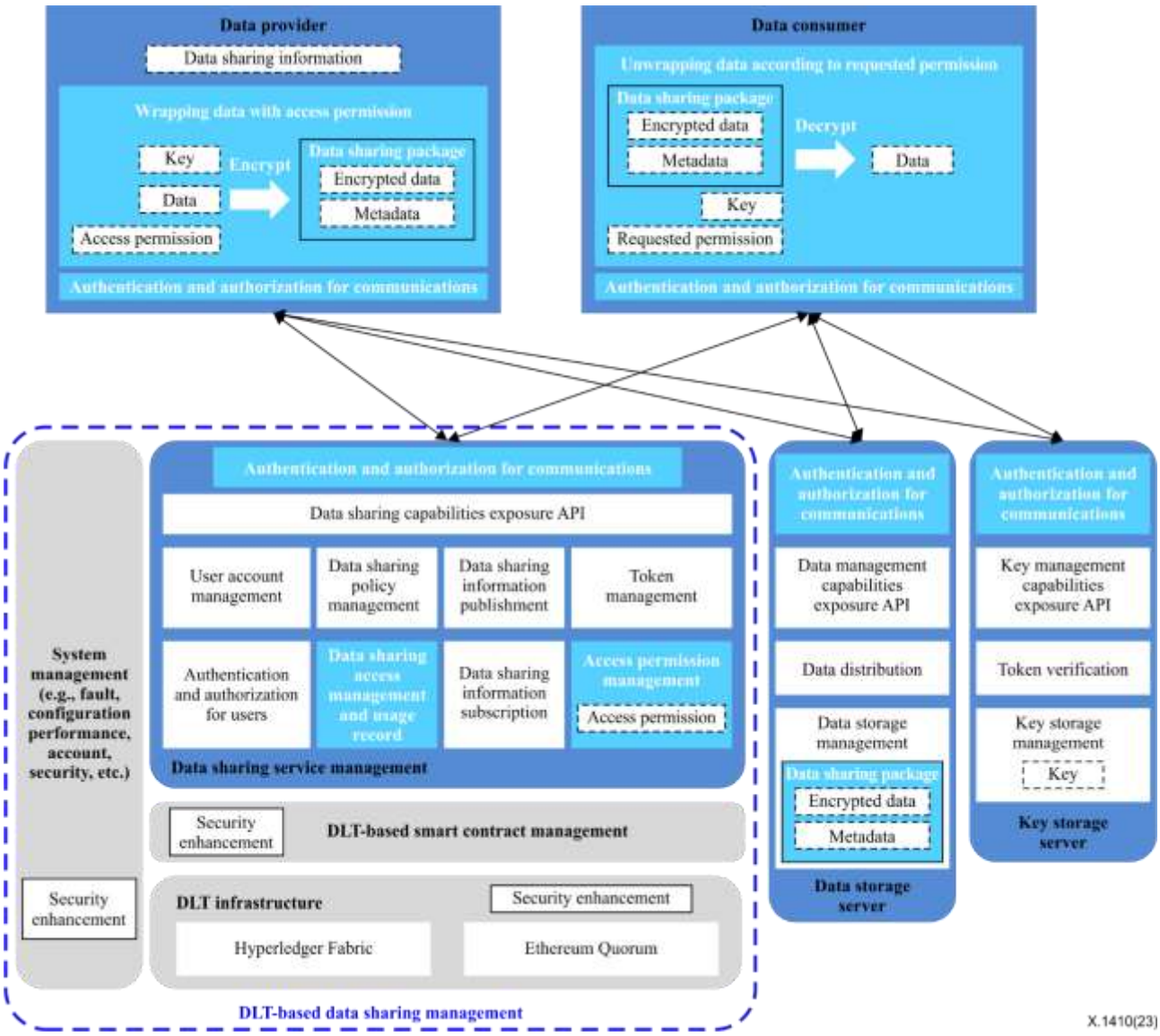

Figure2 – Security architecture of DLT-based data-sharing management

Unlike Figure 1, Figure 2 contains functional components with a light-blue background that are logical security functions, which are described in detail in clause 7.2.

This Recommendation does not describe the security enhancement for the three grey functional components in Figure 2; see [b-ITU-T X.1402].

7.2 Security functional components

7.2.1 Wrapping data with access permission

In order to support users setting access permission for the data to be shared, the *data provider* in Figure 1 needs to be enhanced by introducing a new logical security function, *wrapping data with access permission*, which has the following capabilities:

– generating a key of a symmetric encryption system that is used to encrypt the data to be shared;

– communicating with the *key storage server* to register for a key identifier and the address for getting it;

– communicating with the *data storage server* to register for a data-sharing package identifier and the address for getting it;

– setting access permission for the data to be shared, which includes access times, expiration time, read only, the key identifier and its storage address, the data-sharing package identifier and its storage address;
– generating metadata, which includes a data-sharing package identifier, data provider identifier, data provider's public key and the signature of the preceding information;
– generating a data-sharing package, which includes the encrypted data and the metadata;
– uploading the key to the *key storage server* through a secure transmission channel;
– uploading the data-sharing package to the *data storage server* through a secure transmission channel;
– uploading access permission to the *data-sharing service management* through a secure transmission channel.

7.2.2 Unwrapping data according to the requested permission

In order to support users accessing the shared data according to the requested permission, the *data consumer* in Figure 1 needs to be enhanced by introducing a new logical security function, *unwrapping data according to the requested permission*, which has the following capabilities:

– getting the address for executing the data-sharing smart contract;
– communicating with the *data-sharing service management*, executing the data-sharing smart contract and obtaining the requested permission, which includes access times, expiration time, read only, the key identifier and its storage address, the data-sharing package identifier and its storage address;
– communicating with the *key storage server* to get the encryption key;
– communicating with the *data storage server* to get the data-sharing package, which includes encrypted data and metadata;
– validating the received data-sharing package based on the signature in the metadata;
– decrypting the encrypted data based on the encryption key;
– presenting the shared data in plaintext to the users according to the requested permission;
– encrypting the shared data in the same way as data providers do after finishing data access.

7.2.3 Access permission management

In order to support users setting access permission for the shared data or to support users accessing the shared data according to the requested permission, the *data-sharing service management* in Figure 1 needs to be enhanced by introducing a new logical security function, *access permission management*, which has the following capabilities:

– supporting users setting access permission for the data to be shared:
  • receiving access permission set by the *data provider* from *data-sharing policy management*;
  • storing the access permission;
  • sending the response to *data-sharing policy management*;
– supporting users accessing the shared data according to the requested permission:
  • receiving the permission requested by the *data consumer* from the *data-sharing access management and usage record*;
  • generating the requested permission;
  • sending the requested permission to the *data-sharing access management and usage record*.

7.2.4 Data-sharing access management and usage record

In order to support users tracking the usage of their shared data, the *data-sharing access management and usage record* in Figure 2 needs to be enhanced by the following capabilities:

– communicating with *access permission management* in order to get the requested permission;

– recording the usage of the shared data based on the requested permission.

7.2.5 Authentication and authorization for communications

In order to make communications secure, the five functional components in Figure 1 (i.e., *data provider*, *data consumer*, *data-sharing service management*, *key storage server* and *data storage server)* need to be enhanced by introducing a new logical security function, *authentication and authorization for communications*.

When the *data provider* or *data consumer* sends the request message to *data-sharing service management*, or the *key storage server* or *data storage server*, the *authentication and authorization for communications* can support:

– the *data-sharing service management/key storage server*/*data storage server* authenticating the *data provider/data consumer* based on the certificate [b-IETF RFC 4306] [b-IETF RFC 5246] or pre-shared key [b-IETF RFC 4279] [b-IETF RFC 4306];

– the *data provider/data consumer* authenticating the *data-sharing service management/key storage server*/*data storage server* based on the certificate [b-IETF RFC 4306] [b-IETF RFC 5246];

– the data-*sharing service management/key storage server*/*data storage server* authorizing the *data provider*/*data consumer* based on whitelist/blacklist [b-IETF RFC 5782] [b-IETF RFC 5851] or access control list [b-IETF RFC 4314] [b-IETF RFC 4949];

– the generation of the session key, which will be used to protect the communications between *data provider/data consumer* and the *data-sharing service management/key storage server*/*data storage server*.

7.3 Procedures of sharing data securely

7.3.1 The procedure of data providers to share the data with setting access permission

The procedure for data providers to share data by setting an access permission is shown in Figure 3.

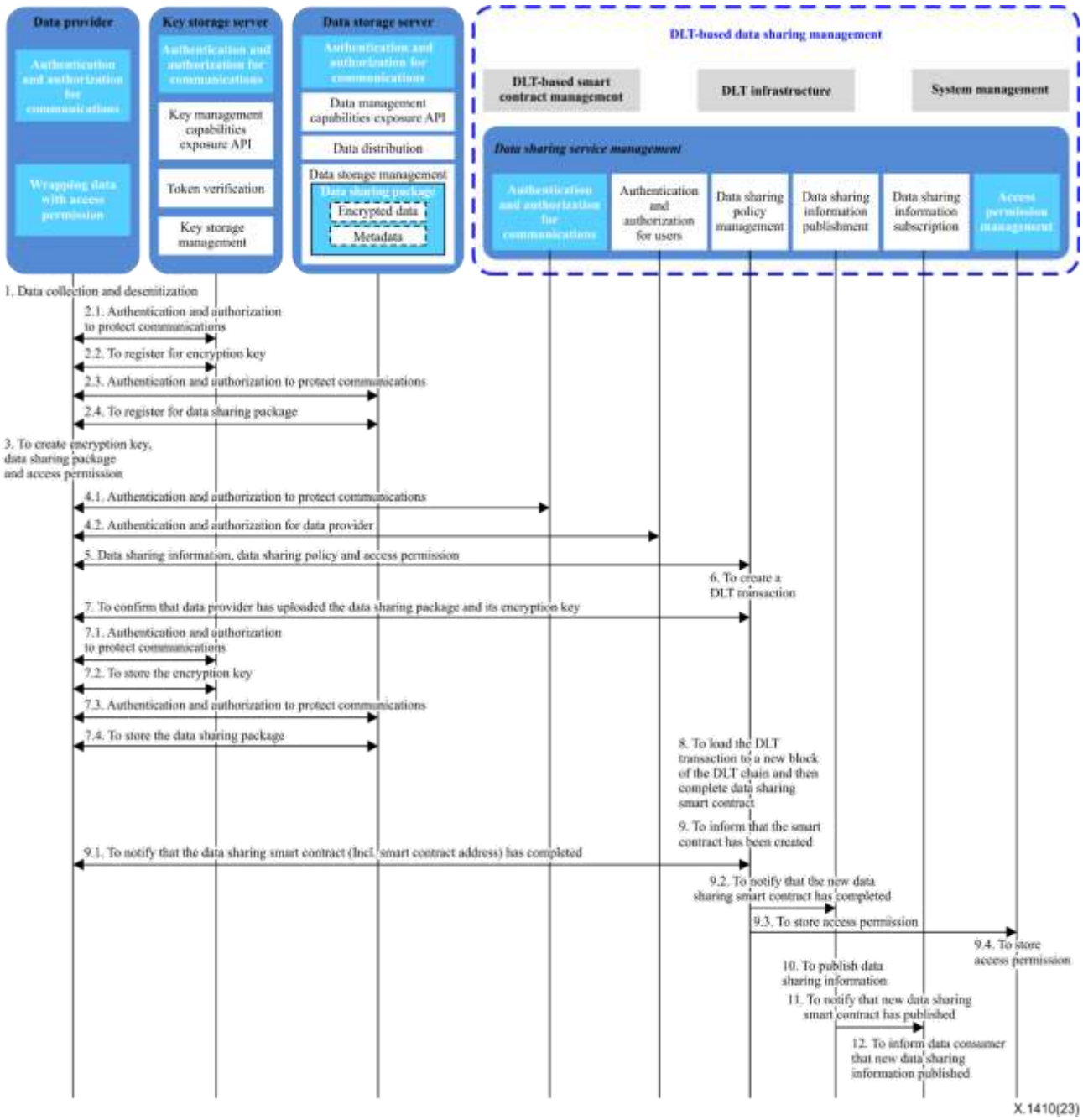

Figure 3 – Procedure for data providers to share data by setting an access permission

As shown in Figure 3, the procedure is described as follows.

1) The *data provider* collects the original data and desensitizes it without having a negative impact on the quality of data to be shared.

2) The *data provider* registers the data-sharing package and its key on the local or remote key storage server and data storage server as follows:

   2.1) *data provider* and *authentication and authorization for communications* of the *key storage server* conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them;

   2.2) *data provider* connects to the *key storage server* and registers for a key identifier and the address from which to get it;

2.3) *data provider* and *authentication and authorization for communications* of the *data storage server* conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them;

2.4) *data provider* connects to the *data storage server* and registers for a data-sharing package identifier and the address for getting it.

3) The *data provider* creates data-sharing information, which includes data provider identifier, data provider public key, data-sharing package identifier and its storage address, key identifier and its storage address, the industries to be allowed to access, the users to be allowed to access, and other data attributes (e.g., data category, data introduction, usage and price).

The *data provider* generates a key that is used to encrypt the data to be shared.

The *data provider* sets access permission for the data, which includes access times, expiration time, read only, the key identifier and its storage address, the data-sharing package identifier and its storage address to be shared;

The *data provider* generates metadata, which includes a data-sharing package identifier, data provider identifier, data provider public key and the signature of the preceding information;

The *data provider* generates a data-sharing package, which includes the encrypted data and the metadata.

4) Before publishing the data to be shared, it is necessary for:

4.1) *data provider* and *authentication and authorization for communications* of the *data-sharing service management* to conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them.

4.2) *data provider* and the *authentication and authorization for users* of the *data-sharing service management* to conduct mutual authentication. After successful mutual authentication, the component *authentication and authorization for users* checks whether the data provider has the right to publish data to be shared.

5) The *data provider* provides the data-sharing information according to the requirements from the *data-sharing policy management* of the *data-sharing service management*. The *data provider* creates a data-sharing policy with data-sharing and other information. The *data provider* sends the data-sharing information, data-sharing policy, access permission and its digital signature to the *data-sharing policy management*.

6) After receiving the data-sharing information, data-sharing policy, access permission and the corresponding digital signature from the *data provider*, the *data-sharing policy management* verifies the digital signature and then creates a DLT transaction.

7) The *data-sharing policy management* confirms with the *data provider* that the key and the data-sharing package has been stored on the *key storage server* and the *data storage server*, respectively. If not, the following steps need to be taken:

7.1) *data provider* and *authentication and authorization for communications* of the *key storage server* conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them;

7.2) *data provider* connects to the *key storage server* and stores the key on it;

7.3) *data provider* and *authentication and authorization for communications* of the *data storage server* conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them;

7.4) *data provider connects* to the *data storage server* and stores the data-sharing package on it.

8) The *data-sharing policy management* sends one or more DLT transactions together with their digital signatures to the underlying components *DLT infrastructure* and *DLT-based smart contract management* to form a new block that contains one or more smart contracts for data sharing. The newly generated block is distributed to the associated DLT nodes. Loading of DLT transactions on to the DLT chain depends on the underlying specific implementation technology (e.g., Hyperledger Fabric, Ethereum Quorum), which lies outside the scope of this Recommendation.

9) The *data-sharing policy management* informs relevant components that the smart contract for data sharing has been created.

   9.1) The *data-sharing policy management* notifies the *data provider* that the data-sharing smart contract has been completed. The notification includes the address for executing the smart contract.

   9.2) The *data-sharing policy management* notifies *data-sharing information publication* that the data-sharing smart contract has been completed. The notification includes the address for executing the smart contract and the data-sharing information.

   9.3) The *data-sharing policy management* notifies *Access permission management* that the data-sharing smart contract has been completed. The notification includes the address for executing the smart contract and access permission.

   9.4) The *access permission management* stores the access permission.

10) After receiving the notification from *data-sharing policy management*, the *data-sharing information publication* publishes the new data-sharing information received.

11) The *data-sharing information publication* informs *data-sharing information subscription* that the smart contract for data sharing has been completed. The notification includes the address for executing the smart contract and new data-sharing information.

12) After receiving the notification from *data-sharing information publication*, the *data-sharing information subscription* sends the address for executing the smart contract and new data-sharing information to subscribers.

7.3.2 The procedure of data consumers to access the shared data according to the requested permission

The procedure for data consumers to access shared data according to the requested permission is shown in Figure 4.

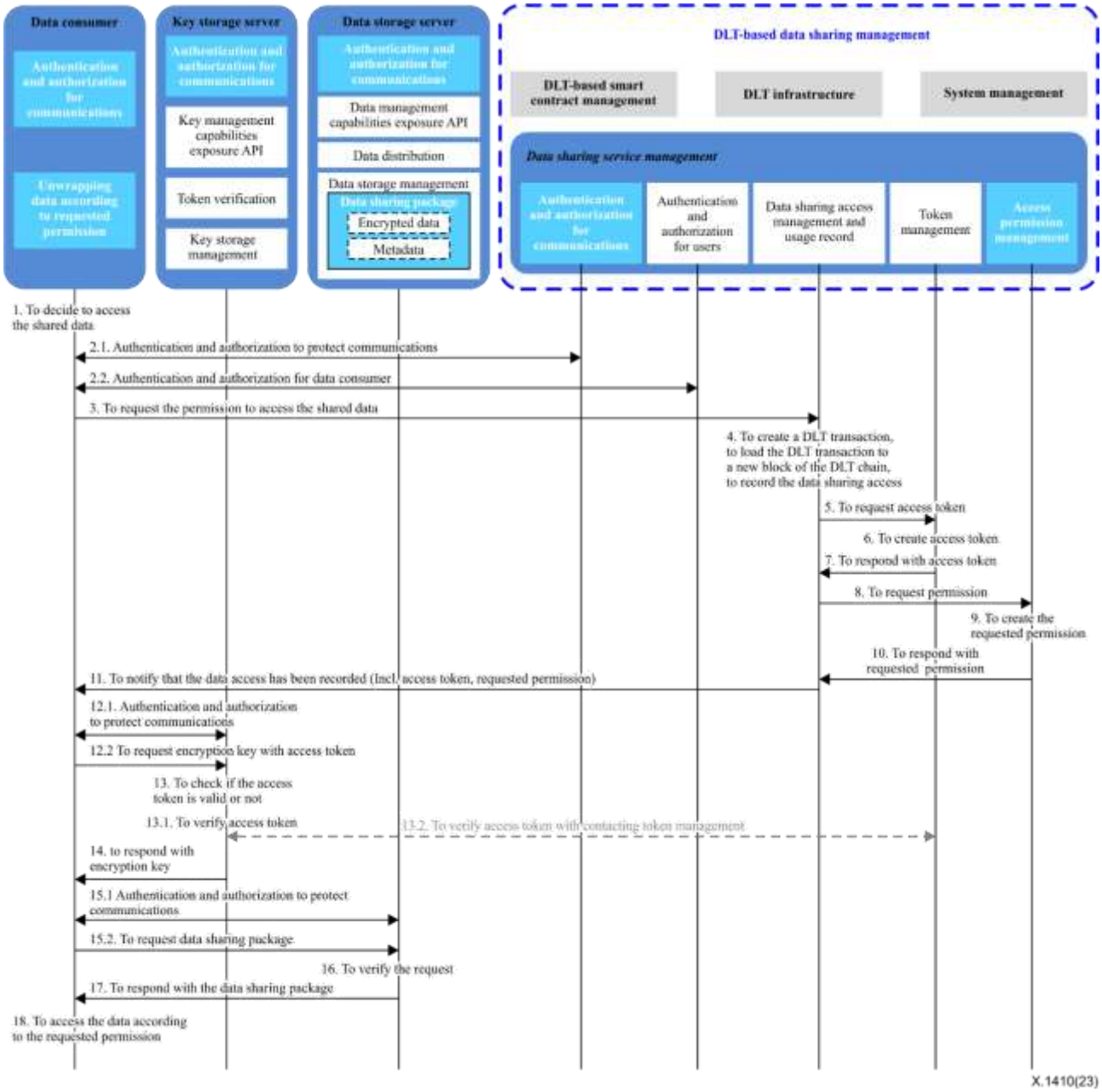

Figure 4 – Procedure of data consumers to access the shared data according to the requested permission

As shown in Figure 4, the procedure of data consumers to access the shared data according to the requested permission is described as follows:

1) The *data consumer* gets data-sharing information by searching on the *data-sharing information publication*, or subscribing to the published information or being forwarded by others. The data consumer decides to access the shared data and executes the data-sharing smart contract.

2) Before executing the data-sharing smart contract to access the shared data, the following operations are performed.

   2.1) The *data consumer* and *authentication and authorization for communications* of the *data-sharing service management* conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them.

2.2) The *data consumer* ensures mutual authentication with the component *authentication and authorization for users* of the *data-sharing service management*. After successful mutual authentication, the component *authentication and authorization for users* checks whether the data consumer has the right to access the shared data.

3) The *data consumer* sends its identity information (e.g., identifier, public key) and the data-sharing information together with its digital signature to the *data-sharing access management and usage record*.

4) After receiving the *data consumer* identity information, data-sharing information and its digital signature, the *data-sharing access management and usage record* verifies the digital signature and then checks whether the data consumer is allowed to access the data according to the data-sharing policy (e.g., only a data consumer from a specified industry or country can access the shared data). If all requirements for accessing the shared data are satisfied, the *data-sharing access management and usage record* creates a DLT transaction according to the received information from data consumer. The *data-sharing access management and usage record* sends one or more DLT transactions together with its digital signature to the underlying components *DLT infrastructure* and *DLT-based smart contract management* to form a new block that contains one or more data access records. The newly generated block is distributed to the associated DLT network nodes. Loading of DLT transactions on to the DLT chain depends on the underlying specific implementation technology (e.g., Hyperledger Fabric, Ethereum Quorum), which lies outside the scope of this Recommendation.

5) The *data-sharing access management and usage record* informs *token management* to create an access token.

6) After receiving notification from the *data-sharing access management and usage record*, *token management* creates an access token.

7) *Token management* sends the created access token to the *data-sharing access management and usage record.*

8) The *data-sharing access management and usage record* informs the *access permission management* to create requested permission.

9) After receiving notification from the *data-sharing access management and usage record*, *access permission management* creates requested permission.

10) The *access permission management* sends the requested permission to the *data-sharing access management and usage record.*

11) After receiving the access token and requested permission, the *data-sharing access management and usage record* records the usage of the shared data and sends the access token, requested permission, key information (e.g., identifier and its storage address), data-sharing package information (e.g., identifier and its storage address), other information (e.g., digital certificates of *key storage server* and *data storage server*) to the *data consumer*.

12) After receiving the access token, requested permission, key information and data-sharing package information from the *data-sharing access management and usage record*, the *data consumer* performs the following operations:

12.1) the *data consumer* and *authentication and authorization for communications* of the *key storage server* conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them;

12.2) the *data consumer* sends the access token to the *token verification* of the *key storage server* according to the key storage address to obtain the encryption key.

13) The *token verification* of the *key storage server* checks whether the access token is valid.

13.1) The *token verification* of the *key storage server* receives the access token from the *data consumer* and then verifies the access token.

13.2) Optionally, the *token verification* may have to communicate with the *token management* of the *DLT-based data-sharing management* when verifying the access token.

14) The *token verification* sends the verification result to the *key storage management* of the *key storage server*. After receiving the verification result from the *token verification*, the *key storage management* of the *key storage server* sends the key to the *data consumer*.

15) After receiving the key from the *key storage management* of the *key storage server*, the *data consumer* does the following operations.

15.1) The *data consumer* and *authentication and authorization for communications* of the *data storage server* conduct mutual authentication and obtain a session key that will be used to protect subsequent communications between them.

15.2) The *data consumer* sends the request to the *data storage server* in order to obtain the data-sharing package.

16) After receiving the request from the *data consumer*, the *data distribution* of the *data storage server* authenticates the *data consumer*.

17) The *data distribution* of the *data storage server* sends the data-sharing package to the *data consumer*.

18) The *data consumer* accesses the data according to the requested permission. The *data consumer* encrypts the shared data in the same way as the *data provider* does after finishing data access.

Annex A

Procedures for DLT-based data-sharing management

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

This annex describes two main procedures: 1) for a data provider to publish the data to be shared based on DLT; and 2) for a data consumer to access the shared data based on DLT.

Before describing the DLT-based data-sharing management procedures, it is assumed that the following conditions are in place:

– each user (e.g., *data provider*, *data consumer*) has obtained a digital certificate and its corresponding private key, generated by itself or from a CA;

– each user has performed the corresponding correct configurations (such as the digital certificate of the DLT node, DLT network connection parameters, network provisioning);

– symmetric encryption systems are available for users (e.g., Data provider, Data consumer). They provide sufficient cryptographic strength for confidentiality of data;

– the *DLT infrastructure* and *DLT-based smart contract management* work properly;

– the DLT network works properly;

– the *key storage server* and *data storage server* work properly.

A.1 The procedure for a data provider to publish the data to be shared based on DLT

The procedure for a data provider to publish data to be shared based on DLT is shown in Figure A.1.

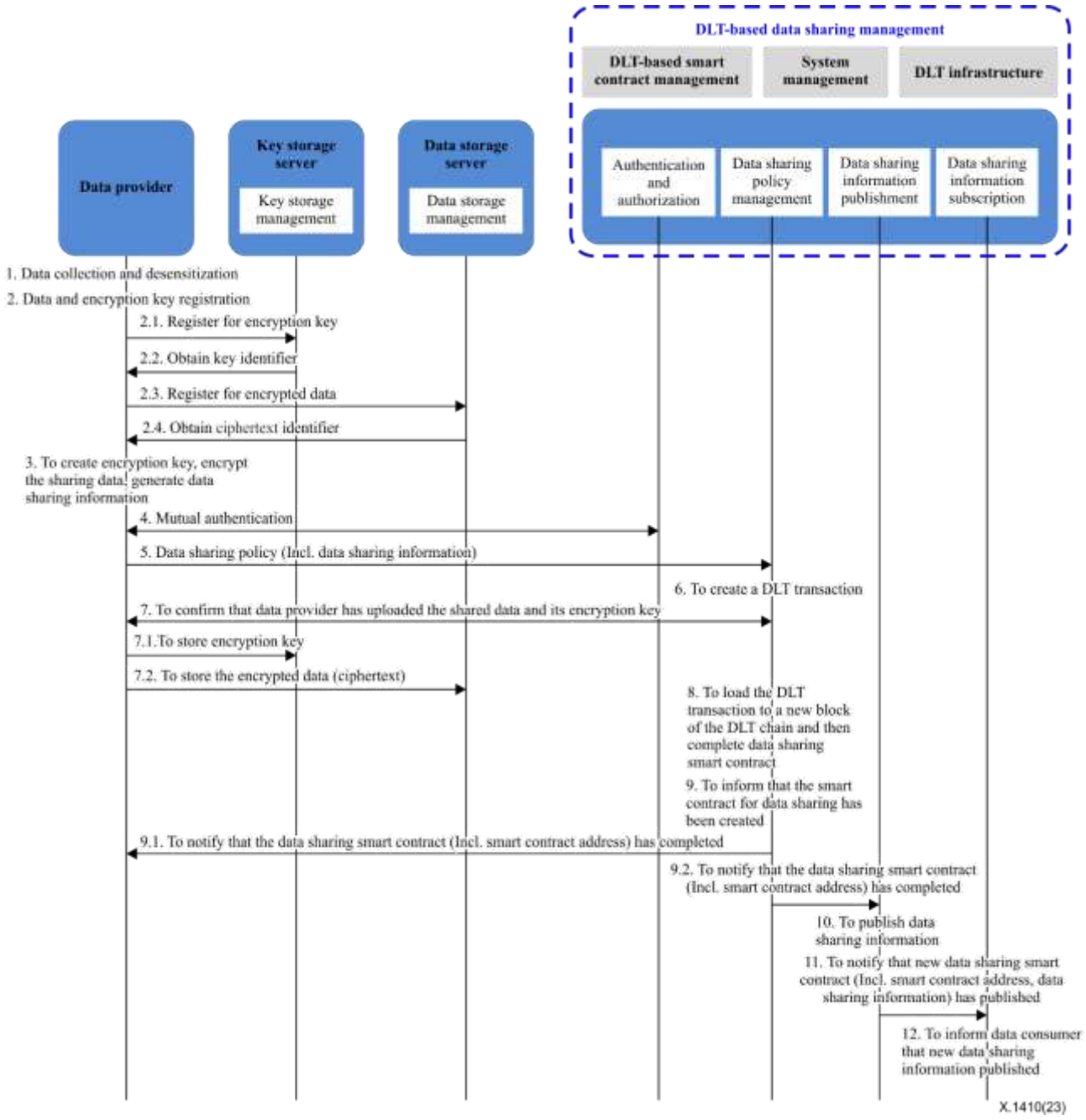

Figure A.1 – Procedure of data provider to publish the data to be shared based on DLT

As shown in Figure A.1, the procedure is described as follows.

1) The *data provider* collects the original data and desensitizes it without having a negative impact on the quality of data to be shared.

2) The *data provider* registers the data to be shared and a key of a symmetric encryption system on the local or remote key storage server and data storage server as follows:

   2.1) to generate a data encryption key used for a symmetric encryption system;

   2.2) from the *key storage server*, the *data provider* obtains the key identifier and its storage address;

   2.3) to encrypt the data with the key using the symmetric encryption system, and register the encrypted data, i.e., ciphertext.

   2.4) from the *data storage server*, the *data provider* obtains the ciphertext identifier and its storage address.

3) The *data provider* generates a data encryption key and encrypts the data to be shared with the generated encryption key. The encrypted data is the ciphertext. The *data provider* creates data-sharing information, which includes data provider identifier, data provider public key, ciphertext identifier and its storage address, data encryption key identifier and its storage address, the industries to be allowed to access, the users to be allowed to access, and other data attributes (e.g., data category, data introduction, usage, price).

4) Before publishing the data to be shared, the *data provider* ensures mutual authentication with the component *authentication and authorization for users* of the *DLT-based data-sharing management*. A credential digital certificate could be used for mutual authentication. After successful mutual authentication, the component *authentication and authorization for users* checks whether the data provider has the right to publish data to be shared.

5) After successful authentication and authorization, the *data provider* provides the data-sharing information according to the requirements from the *data-sharing policy management* of the *DLT-based data-sharing management*. The *data provider* creates a data-sharing policy with data-sharing and other information. The *data provider* sends the data-sharing policy and its digital signature to the *data-sharing policy management*.

6) After receiving the data-sharing policy and the corresponding digital signature from the *data provider*, the *data-sharing policy management* verifies the digital signature and then creates a DLT transaction.

7) The *data-sharing policy management* confirms with the *data provider* that the key and the ciphertext has been stored on the *key storage server* and the *data storage server* respectively. If not, the following steps need to be taken:

   7.1) the *data provider* stores the key on the *key storage server*;

   7.2) the *data provider* stores the ciphertext on the *data storage server*.

8) The *data-sharing policy management* sends one or more DLT transactions together with its digital signature to the underlying components *DLT infrastructure* and *DLT-based smart contract management* to form a new block that contains one or more smart contracts for data sharing. The newly generated block is distributed to the associated DLT nodes. Loading of DLT transactions on to the DLT chain depends on the underlying specific implementation technology (e.g., Hyperledger Fabric, Ethereum Quorum), which lies outside the scope of this Recommendation.

9) *Data-sharing policy management* informs relevant components that the smart contract for data sharing has been created.

   9.1) *Data-sharing policy management* notifies the *data provider* that the data-sharing smart contract has been completed. The notification includes the address for executing the smart contract.

   9.2) *Data-sharing policy management* notifies *data-sharing information publication* that the data-sharing smart contract has been completed. The notification includes the address for executing the smart contract and the data-sharing information.

10) After receiving the notification from *data-sharing policy management*, the *data-sharing information publication* publishes the received new data-sharing information.

11) *Data-sharing information publication* informs *data-sharing information subscription* that the smart contract for data sharing has been completed. The notification includes the address for executing the smart contract and new data-sharing information.

12) After receiving the notification from *data-sharing information publication*, the *data-sharing information subscription* sends the address for executing the smart contract and new data-sharing information to the subscriber.

A.2 The procedure of data consumer to access the shared data based on DLT

The procedure for a data consumer to access shared data based on DLT is shown in Figure A.2.

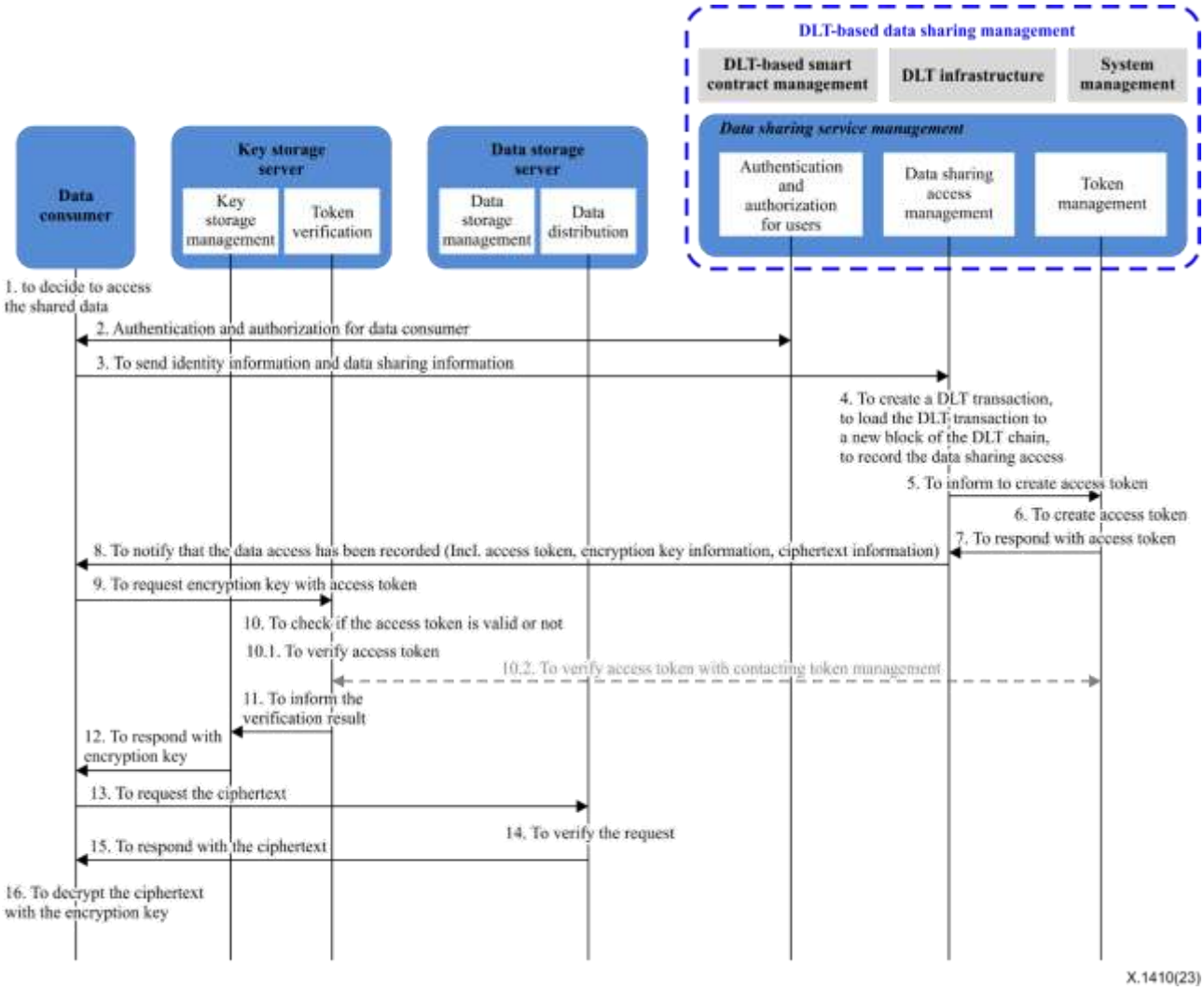

Figure A.2 – Procedure of data consumer to access the shared data based on DLT

As shown in Figure A.2, the procedure is described as follows:

1) The *data consumer* gets the data-sharing information by searching the *data-sharing information publication* subscribing to the published information or being forwarded by others. The *data consumer* decides to access the shared data and executes the data-sharing smart contract.

2) Before executing the data-sharing smart contract to access the shared data, the *data consumer* ensures mutual authentication with the component *authentication and authorization for users* of the *DLT-based data-sharing management*. A credential digital certificate is recommended for mutual authentication. After successful mutual authentication, the component *Authentication and authorization for users* checks whether the data consumer has the right to access the shared data

3) After successful authentication and authorization, the *data consumer* sends its identity information (e.g., identifier, public key) and the data-sharing information together with its digital signature to *data-sharing access management*.

4) After receiving the data consumer's identity information, data-sharing information and its digital signature, *data-sharing access management* verifies the digital signature and then checks whether the data consumer is allowed to access the data according to the data-sharing policy (e.g., only a data consumer from a specified industry or country can access the shared data). If all requirements for accessing the shared data are satisfied, *data-sharing access management* creates a DLT transaction according to the information received from data consumer. *Data-sharing access management* sends one or more DLT transactions together with their digital signatures to the underlying components *DLT infrastructure* and *DLT-based smart contract management* to form a new block that contains one or more data access records. The newly generated block is distributed to the associated DLT network nodes. Loading of DLT transactions on to the DLT chain depends on the underlying specific implementation technology (e.g., Hyperledger Fabric, Ethereum Quorum), which lies outside the scope of this Recommendation.

5) The *data-sharing access management* informs *token management* to create an access token.

6) After receiving notification from the *data-sharing access management*, *token management* creates an access token.

7) *Token management* sends the created access token to the *data-sharing access management.*

8) After receiving the access token, the *data-sharing access management* sends the access token, key information (e.g., identifier and its storage address), ciphertext information (e.g., identifier and its storage address), other information (e.g., digital certificates of *key storage server* and *data storage server*) to the *data consumer*.

9) After receiving the access token, key information and ciphertext information from *data-sharing access management*, the *data consumer* sends the access token to the *token verification* of the *key storage server* according to the key storage address to obtain the encryption, which is used to decrypt the ciphertext.

10) *Token verification* checks whether the access token is valid.

    10.1) *Token verification* of the *key storage server* receives the access token from the *data consumer* and then verifies the access token.

    10.2) Optionally, *token verification* may have to communicate with *token management* of the *DLT-based data-sharing management* when verifying the access token.

11) *Token verification* sends the verification result to *key storage management*.

12) After receiving the verification result from *token verification*, *key storage management* sends the encryption key to the *data consumer*.

13) After receiving the encryption key from *key storage management*, the *data consumer* sends a request to the *data storage server* in order to obtain the ciphertext.

14) After receiving the request from the *data consumer*, the *data distribution* of the *data storage server* authenticates the *data consumer*.

15) *Data distribution* of the *data storage server* sends the ciphertext to the *data consumer*.

16) The *data consumer* decrypts the *ciphertext* with the encryption key and then accesses the shared data in plaintext.

Bibliography

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

[b-ITU-T X.1400] Recommendation ITU-T X.1400 (2020), *Terms and definitions for distributed ledger technology*.

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

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

[b-ISO/IEC 18033-1] ISO/IEC 18033-1:2021, *Information security – Encryption algorithms – Part 1: General*.

[b-ISO/IEC 20944-1] ISO/IEC 20944-1:2013, *Information technology – Metadata registries interoperability and bindings (MDR-IB) – Part 1: Framework, common vocabulary, and common provisions for conformance*.

[b-ISO/IEC 29100] ISO/IEC 29100:2011, *Information technology — Security techniques — Privacy framework*.

[b-ISO/IEC/IEEE 15939] ISO/IEC/IEEE 15939:2017, Syst*ems and software engineering – Measurement process.*

[b-IETF RFC 4279] IETF RFC 4279 (2005), *Pre-shared key ciphersuites for transport layer security (TLS)*.

[b-IETF RFC 4306] IETF RFC 4306 (2005), *Internet key exchange (IKEv2) protocol*.

[b-IETF RFC 4314] IETF RFC 4314 (2005), *IMAP4 access control list (ACL) extension*.

[b-IETF RFC 4949] IETF RFC 4949 (2007), *Internet security glossary, version 2*.

[b-IETF RFC 5246] IETF RFC 5246 (2008), *The transport layer security (TLS) protocol: Version 1.2*.

[b-IETF RFC 5782] IETF RFC 5782 (2010), *DNS blacklists and whitelists*.

[b-IETF RFC 5851] IETF RFC 5851 (2010), *Framework and requirements for an access node control mechanism in broadband multi-service networks*.

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, 2023