

Calypso

Networks Association

GENERAL PRESENTATION

Calypso Core Definitions

The information contained in this document is public.

© Calypso Networks Association 2022. All rights reserved.

The authors of this document make no other representation or warranty regarding whether any particular physical implementation of any part of this specification does or does not violate, infringe, or otherwise use other patents, copyrights, trademarks, trade secrets, know-how, and/or other intellectual property of third parties, and thus any person who implements any part of this document should consult an intellectual property attorney before any such implementation.

Any party seeking to implement this document is solely responsible for determining whether their activities require another license to any technology. Calypso Networks Association shall not be liable for infringements of any third party's intellectual property right.

Author & Editor

Calypso
Networks Association

Link and Contact

Website	https://www.calypsonet.org
Technical Support	support@calypsonet.org

Revision List

Version	Date	Modifications
1	07-07-2022	First release

Table of Contents

LINK AND CONTACT	2
REVISION LIST	2
TABLE OF CONTENTS	3
1 OVERVIEW	5
1.1 CONTEXT	5
1.2 DOCUMENT PURPOSE	5
1.3 EXAMPLE, REMARK AND WARNING	6
2 CALYPSO BASE CONCEPTS	6
2.1 CNA AND THE CALYPSO STANDARD	6
2.2 THE CALYPSO SYSTEM	7
2.3 SECURITY IN THE CALYPSO SYSTEM	8
3 CALYPSO COMPONENTS: CARD	8
3.1 DEFINITION	8
3.2 CALYPSO RANGE OF PRODUCTS	9
3.3 CARD TECHNICAL CHARACTERISTICS	11
3.4 COMMUNICATION PROTOCOLS	12
3.4.1 CLA Byte	12
3.4.2 Command Cases	13
3.4.3 APDU Encapsulation	13
3.5 DATA ORGANIZATION	14
3.5.1 Introduction	14
3.5.2 Application Identifier (AID)	15
3.5.3 Calypso Serial Number	17
3.5.4 Calypso File Structures	18
3.5.5 File Identification	18
3.5.6 Current File, Current DF and Current Card Application	19
3.5.7 EF Types and Data Organization	19
3.5.8 Data Representation and Coding	24
3.6 INTEGRATION WITH OTHER CALYPSO CARD APPLICATIONS	24
3.7 STORED VALUE APPLICATION	24
4 CALYPSO COMPONENTS: SAM	26
5 CALYPSO COMPONENTS: TERMINAL	26
5.1 DEFINITION	26
5.2 READER LAYER	27

5.3	CALYPSO LAYER	28
5.4	TICKETING LAYER	28
6	CALYPSO COMPONENTS: CENTRAL SYSTEM/BACK OFFICE	29
7	SECURITY MECHANISMS	29
7.1	INTRODUCTION TO CALYPSO SECURITY	29
7.2	SYMMETRIC CRYPTOGRAPHY	30
7.2.1	Key types	30
7.2.2	Key Identifier	30
7.2.3	Key Index	31
7.2.4	Algorithms	31
7.3	PKI CRYPTOGRAPHY (ASYMMETRIC CRYPTOGRAPHY)	32
7.3.1	Principles	32
7.3.2	Public Key Infrastructure (PKI) Overview	32
7.3.3	Algorithms	33
7.4	ACCESS CONDITIONS	34
7.4.1	Principles	34
7.4.2	Groups of Commands	34
7.4.3	Access Modes	35
7.5	SECURE SESSION DESCRIPTION	35
7.5.1	Secure Session Security	35
7.5.2	Session MAC Authentication	37
7.6	PKI SECURITY MECHANISMS	38
7.6.1	Overview	38
7.6.2	PKI Session	39
7.7	RATIFICATION	39
7.8	TRANSACTION COUNTER	42
7.9	MEMORY MODIFICATION MANAGEMENT	42
8	ANNEX	43
8.1	CALYPSO REFERENCES	43
8.2	NORMATIVE REFERENCES	44
8.3	GLOSSARY AND ACRONYMS	46

1 OVERVIEW

1.1 Context

The passenger is at the core of all decisions and concerns of public transport operators. Making sure that the service meets the needs of the passengers, offering increased comfort and security and even personalized services; all this requires a detailed knowledge of the clients, their needs and habits. It is however not enough for the service offered to be of the highest possible quality it also needs to be balanced in terms of cost.

Public Transport is deeply tied with the city where it lives and operates. The flow of movement of people changes depending on multiple factors which means that the design of the city, and its evolution, together with its public transport infrastructure must be dynamic.

Ticketing systems are crucial for the decision-making process of transport operators and city officials. It is not enough to use these systems to facilitate the access of passengers to the transportation network. The information generated by these systems must be used to build a new relationship between transport operators and their passengers by providing a cohesive service offer across multiple operators and even an improved relationship between the citizen and the city.

For any entity interested by this global concern, Calypso provides an ever-evolving solution based on the most recent contactless technologies.

Calypso goes beyond the simple technical evolution of replacing a paper ticket by a contactless customer media. It is designed by transport operators for transport operators offering an open, modular, and complete solution.

1.2 Document Purpose

The purpose of this document is to serve as the entry point into the world of Calypso. It will introduce and present what constitutes a Calypso System, the core concepts of the technology to facilitate its understanding. The core concepts presented in this document are required for the clear understanding of all other Calypso technical documents (which will often refer to this document) it should thus be a constant companion in the exploration of the Calypso world. It is intended for anyone wishing to learn about the technical part of Calypso whether they are users (transport providers, PTAs, PTOs, etc) or solutions providers (consultants, integrators, system designers, software developers, etc).

The present document includes the following chapters:

- Calypso Base Concepts – Presents an overview of the Calypso Standard and its base concepts.
- Calypso Components: Card – Presents the Card and its role in the scope of a Calypso system.
- Calypso Components: SAM – Presents the SAM and its role in the scope of a Calypso system.
- Calypso Components: Terminal – Presents the Terminal and its role in the scope of a Calypso system.

- Calypso Components: Central System/Back Office – Provides a brief introduction of the Central System in the scope of a Calypso system.
- Security mechanisms – Describes the Calypso security mechanisms.

1.3 Example, Remark and Warning

Warnings, examples and remarks are indicated with a specific sign:



Example.



Remark.



Warning.

2 CALYPSO BASE CONCEPTS

2.1 CNA and the Calypso Standard

Calypso was born from the vision of transport operators to develop an open standard for contactless ticketing that would allow for fast and secure transactions answering the demanding needs of public transport throughput. The **Calypso** standard is unique in that it's owned and managed by its users being open to all suppliers in a fair and non-discriminatory manner thus ensuring the sustainability of the investments made by operators and/or authorities in their systems.

Through the *Icare* and *Calypso* European projects, associating *Brussels* in Belgium, *Lisbon* in Portugal, *Konstanz* in Germany, *Paris* in France and *Venice* in Italy, it was ensured that the standard was suited to any transport environment, and could be married to other services, such as electronic purse, loyalty, access control, etc.

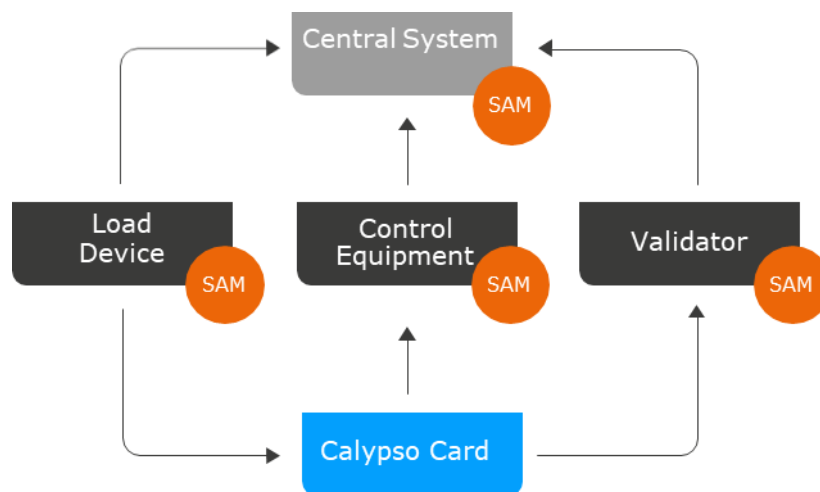
To ensure that the **Calypso** standard would remain open and that it would evolve according to the needs of the users the **Calypso Networks Association (CNA)** was created. **CNA** is a not-for-profit association, based in Belgium, open to all ticketing actors, with the mission to promote and manage the standard and its evolutions.

CNA brings together users and suppliers in a collaborative spirit to exchange ideas, experiences and requirements to advance open systems that support seamless, consumer ticketing needs through the participation in working groups that are open to all members.

The **Calypso** brand can only be used by vendors that have gone through a thorough certification process to ensure that all products behave consistently regardless of supplier or manufacturer.

More information may be found on the CNA web site: <http://www.calypsonet.org>.

2.2 The Calypso System



A Calypso system is a secure contactless system including different types of cards (or medias), SAMs, different types of terminals, and the central system:

- The Calypso card hosts one or several card applications complying with the Calypso card specifications, designed for payment or access to services such as public transport.
- The SAM ensures the authenticity of the cards and integrity of the card application data, protects the system keys, and prevents their theft. They can be installed inside terminals or made available remotely through a server.
- Terminals are devices located in the field (in stations, vehicles, etc) providing different services depending on the type of terminal, for example:
 - Reloading devices that load transport titles (one-way tickets, season tickets, etc.) and value into the cards.
 - Validators that are used to validate entrance to (and, in some cases, exit from) the network.
 - Hand-held control equipment that are used to verify that a card contains the required entitlements for travelling.
- The central system can be at the level of an operator or an authority (aggregating the data from all operators of the interoperable system). Some of its functionalities are keeping track of all transactions performed in the terminals, providing usage statistics, and verifying the overall system security and integrity.

All parts of the system must be designed with the main goals of a secure contactless system in mind, particularly in terms of the speed and security of transactions.

Smart card

In the present document “smart card” means both “card” and “SAM”, unless indicated otherwise.

2.3 Security in the Calypso System

When using a Calypso card, the users may wish to ensure:

- that the card data are genuine. It should not be possible for a defrauder to forge the data, or to modify it in the card.
- the integrity of the data written in the card, even if the power supply is unexpectedly cut during a single write operation, or during the synchronous update of related files in the card application (for example, adding a new event in the card application may be linked with a counter decrement).

The Calypso Standard solves these needs through a single mechanism called the **Secure Session**.

A secure session begins by a specific command sent to the card to open the secure session (*Open Secure Session*) and ends by a specific command to close the secure session (*Close Secure Session*).

During the secure session, it is possible to read and write data into the Calypso card application (the access may be restricted on some files by specific conditions, e.g. having presented a PIN code).

When the session closes, all data exchanged are authenticated by the card and by the SAM used by the terminal. This operation simultaneously proves:

- the authenticity of the terminal to the Calypso card application (authenticating the terminal),
- the authenticity of the Calypso card application to the terminal (authenticating the Calypso card application and the card),
- that the data exchanged are genuine and have not been tampered with by a defrauder (integrity of the data exchanged),
- that the modifications have been made in the card as requested by the terminal (non-repudiation).

The details of the secure session mechanism with symmetric keys, as well as a special feature called **Ratification**, are explained in detail in section 7.7 Ratification.

PKI Mode

When PKI mode is used, the secure session authenticates only one device: the card and its data. All commands which would modify the card application are rejected (only reading is allowed).

The PKI secure session mechanism is explained in detail in section 7.3 PKI Cryptography (Asymmetric Cryptography).

3 CALYPSO COMPONENTS: CARD

3.1 Definition

In the scope of Calypso documents, a **card** may be any portable device with an ISO/IEC 14443 interface. A few examples of so-called cards are:

- A contactless smart card.

- A mobile phone with contactless communication.
- A wristwatch with an embedded contactless component.

A **Calypso card**, within any card, is a secure element containing at least one application compliant with one of the Calypso card specifications and has received a certificate from CNA.

Important note: for composite cards, where Calypso card applications are stored in an independent and possibly removable secure component (the secure element), all requirements apply to this component except requirements dealing with contactless protocols, which apply to the card (e.g. a NFC phone).

The Calypso card is an integral part of an information system architecture, allowing transfer and verification of rights (e.g. transport rights). The Calypso card application may contain additional information other than rights, for example about the holder, the last uses, etc.

Since the rights stored in the card represent a monetary value, their use is protected by cryptographic algorithms, preventing defrauders from forging a fake Calypso card application, or fraudulently reloading a Calypso card application.

The cryptographic algorithms use secret keys, secured in the cards and in the Calypso SAM (Secure Application Module).

After manufacturing, the card is initialized (by activating or loading the Calypso card application), pre-personalized by writing in it the secret keys and setting the AID, and finally personalized by filling in the card application data.

Calypso card applications can be used for many types of applications, including public transportation, access to city services, etc.

During its life, a Calypso card will typically:

- Be loaded with new information, transport rights, transport tokens.
- Be used for validation to access the transport network or by debiting the transport tokens.
- Be controlled to check its information, or to display it to the user.

A Calypso card may also be used for applications for purposes other than public transportation and may contain card applications other than Calypso. This, however, is outside the scope of the present document.

3.2 Calypso Range of Products

The Calypso standard has evolved to offer new products, to include advances in security and standardization, and to improve flexibility and performances within interoperable networks:

Product Name	Release Year	Main Characteristics & Evolutions
Basic	2019	<p>Low-end cards: Same main commands as Calypso Light (subset of Calypso Light commands). Same functional security as Calypso Prime and Light. One file structure allowed. Only TDES algorithm available. Only contactless: ISO/IEC 14443 (A or B) protocol.</p>
Prime Revision 3 PKI mode	2018	<p>Add an optional PKI mode: Revision 3 PKI mode for the secure session</p>
Light	2017	<p>Middle-end cards: Same main commands as Calypso Prime (subset of Calypso Prime commands). Same level of security as Calypso Prime Two file structures allowed. Only TDES algorithm available. Only contactless: ISO/IEC 14443 (A or B) protocol.</p>
HCE	2015	<p>Definition of a Calypso HCE application: Based on Calypso Prime Revision 3. Use NFC HCE interface provided by the Android OS. Security techniques to delay attacks on the software. Countermeasures to detect frauds at the central system level.</p>
Prime Revision 3 Extended mode	2013	<p>Definition of features optional and additional: AES cryptography. Session with encryption, prior authentication, extended signatures.</p>
Card Applet	2012	<p>Generic applet to be hosted in a Secure Element: GlobalPlatform mechanisms for installation and management. Full Calypso Prime Revision 3 specification. Same level of security as Calypso Prime. Provided free of charge to CNA members.</p>
Prime Revision 3 Regular mode	2009	<p>Improvement of compatibility between cards: Full ISO/IEC 14443 (A or B) protocol. TDES and DESX cryptography. Binary files, shared files, Stored Value application, PIN, improved key management, etc. Java Card compatibility improvements.</p>
Prime Revision 2	2001	<p>Standard specification abstracted from specific products: Full ISO/IEC 14443 type B protocol. DESX cryptography. Several functional improvements: application selection, key version management, etc.</p>
Prime Revision 1	1997	<p>World first open specification for public transport ticketing: ISO/IEC 7816 compliance. Secure session with ratification (invented and patented by Calypso). Innovatron protocol (used as the basis for ISO/IEC 14443 B protocol).</p>

For the purposes of this document the Calypso Range of Products is divided into three distinct families: Prime, Light and Basic. Comparison tables will be presented where they make sense to present any specificity of each product family.

3.3 Card Technical Characteristics

A Calypso card is based on a microprocessor smart card component (i.e. the secure element of the card). Its main characteristics are:

- Complies with ISO/IEC 14443 (A and/or B) and ISO/IEC TS 24192 (previous edition referenced under CEN/TS 16794).
- Complies with ISO/IEC 7816-4.
- Allows management of any data, for example EN 1545.
- Ensures a high security of transactions.
- Ensures fast contactless transactions.

For Calypso products featuring a contact interface, if the component is embedded in a smart card format with contacts, it also complies with ISO/IEC 7816-3.

ISO/IEC 14443 and ISO/IEC TS 24192 Compliance

For contactless operations, a Calypso card is compliant with the ISO/IEC 14443 (A and/or B), parts 1, 2, 3 and 4, and to ISO/IEC TS 24192 (previous edition referenced under CEN/TS 16794). Other low-level (contactless or contact) protocols may be managed by the card.

ISO/IEC 7816-3 Optional Compliance

A Calypso card package may comply with ISO/IEC 7816-1 and ISO/IEC 7816-2 standards if it's in a PVC card format, or may be packaged differently: embedded in a wristwatch, in a paper ticket, in a mobile phone, etc.

When packaged as an ISO/IEC 7816 card with contacts, it shall work in contact operations at 5V, according to the ISO/IEC 7816-3 T=0 standard. It is possible, but not required, to also allow other working voltages (e.g. 3V) or other protocols (e.g. T=1, I2C, USB, etc.).

ISO/IEC 7816-4 Compliance and EN 1545

The Calypso card application data are organized in files, according to the ISO/IEC 7816-4 standard. See the Calypso product specifications to have further information about examples of the possible file structures.

A Calypso card may be used for various applications whether by having multiple Calypso card applications in the same card or by coexisting, in the same card, with applications other than Calypso.

For public transport applications, its files may encode EN 1545 data structures. Such an encoding is not described in the Calypso product specifications.

Calypso products never analyze the data stored in their files. The data present in the files may thus be coded differently for each card application. It is entirely up to networks to define their data model and then for the

terminals using the Calypso cards (validation or reloading machines, etc.) to decide the meaning of the data written in the files of Calypso card applications (following the data model defined by the networks).

The actual file structure of a Calypso card application may be chosen from the *File Structure Registry* (ref. 060709-CalypsoFiles) or may be proprietary. File structures will keep being added to this registry, which is managed by CNA.

High Security

Different Calypso products and types of Calypso card applications may offer different levels of security.

Calypso allows the use of:

- PKI
- AES (algorithm with secret keys of 128 bits).
- Triple-DES (algorithms with secret keys of 112 bits).
- DESX (algorithms with secret keys of 128 bits) (deprecated).

Product Comparison

Product	Characteristics
Prime	Contactless mandatory: ISO/IEC 14443 (A or B) protocol. Contact allowed: ISO/IEC 7816 T=0. All algorithm (PKI, AES, TDES and DESX) supported.
Light	Only contactless: ISO/IEC 14443 (A or B) protocol. Only TDES algorithm supported.
Basic	Only contactless: ISO/IEC 14443 (A or B) protocol. Only TDES algorithm supported.

3.4 Communication Protocols

3.4.1 CLA Byte

A Calypso card application accept CLA=00h for all commands.

Within some contexts (e.g. mobile phones), Integrated Circuit Card (ICC) containing a Calypso card application may manage more than one logical channel. Such ICC types indicate to the terminal an available channel for the application selection and processing, as defined in ISO/IEC 7816-4. The logical channel is coded in the CLA byte.

3.4.2 Command Cases

By definition, depending on their data, a Calypso command and its response are in one of the following cases:

Case	Are there incoming data (data in the command)?	When the command is successful, are there outgoing data (data in the response)?
Case 1	No	No
Case 2	No	Yes
Case 3	Yes	No
Case 4	Yes	Yes

Notes:

- This definition supersedes ISO/IEC 7816-4, which defines the cases depending on the total number of bytes in the command and in its response, and on the presence or absence of Lc and Le fields.
- A Calypso command contains always at least 4 bytes (CLA, INS, P1 and P2), and a Calypso response contains always at least two bytes (SW1 and SW2).
- Depending on implementation, in ISO/IEC 7816-3 T=0 mode a command may fail before the actual transmission of the incoming data bytes. Regarding the table above, such command is nevertheless defined as having incoming data.


3.4.3 APDU Encapsulation

The following table illustrates the APDU encapsulations and the management of the ISO/IEC 7816 bytes Lc and Le (length of command data, expected length of response data) in contactless and contact modes, for the four Calypso cases, in the case of successful commands.


Case	ISO/IEC 14443	ISO/IEC 7816 T=0
Case 1	Cmd: CLA / INS / P1 / P2 Resp: SW1 / SW2	Cmd: CLA / INS / P1 / P2 / 00h Resp: SW1 / SW2
Case 2	Cmd: CLA / INS / P1 / P2 / Le Resp: Data (N bytes) / SW1 / SW2 XXh = number of data bytes available. - If Le = 00h N = XXh. - If Le <= XXh: N= Le bytes. - If Le > XXh: N=0, SW1SW2=6CXXh.	Cmd: CLA / INS / P1 / P2 / Le Resp: Data (N bytes) / SW1 / SW2 XXh = number of data bytes available. - If Le = 00h: N=0, SW1SW2=6CXXh. - If Le <= XXh: N= Le bytes. - If Le > XXh: N=0, SW1SW2=6CXXh. Exception for Get Response command: - If Le < XXh: N=0, SW1SW2=61XXh.
Case 3	Cmd: CLA / INS / P1 / P2 / Lc / Data Resp: SW1 / SW2 Lc = length of command data	Cmd: CLA / INS / P1 / P2 / Lc / Data Resp: SW1 / SW2 Lc = length of command data

Case 4	<p>Cmd: CLA / INS / P1 / P2 / Lc / Data / Le=00h Resp: Data (N bytes) / SW1 / SW2</p> <p>Lc = length of command data. N is always the number of data bytes available.</p>	<p>Cmd: CLA / INS / P1 / P2 / Lc / Data Resp: 61h / XXh Cmd: CLA / C0h / 00h / 00h / XXh Resp: Data (N bytes) / SW1 / SW2</p> <p>Lc = length of command data N=XXh is always the number of data bytes available. Get Response command is required to retrieve the response data and the command status.</p>
--------	---	--

Note: The above table applies only to successful commands (SW1-SW2=9000h, 62XXh and 63XXh). Commands in error return only the error status with the SW1-SW2 bytes and no data, whatever the case would be for the successful command.

 Commands managed by card applications other than Calypso card applications are not in the scope of the Calypso specifications (e.g. commands sent on a logical channel where the card application currently selected is not a Calypso card application).

Incorrect Framing

 Warning: Commands with incorrect framing are not in the scope of the Calypso specifications.

Product Comparison

Product	Characteristics
Prime	Multiple card applications allowed. Can coexist with non-Calypso card applications.
Light	Only one card application allowed.
Basic	Only one card application allowed.

3.5 Data Organization

3.5.1 Introduction

The data of a Calypso card application are organized in hierarchical files, as defined in ISO/IEC 7816-4.

There are two main types of files:

- Elementary File (EF): files containing user data, organized in linear or cyclic records, counters, etc., described in more details in next sections.
- Dedicated File (DF): directories which may contain Elementary Files and other Dedicated Files. If it exists, the root DF of a card is called the Master File (MF).

For the purposes of this document a DF complying with any Calypso product specification will be called a Calypso DF.

A Calypso DF may be any DF of a card: it may be contained in the MF or in a DF at any level below the MF, it may be the MF itself, or it may be contained in a card without MF.

Depending on the product type, a Calypso card may contain one or several Calypso card applications and may contain other card applications (e.g. banking electronic purse, loyalty). The distinct Calypso product specifications will detail the limits in terms of number of Calypso card applications present in the same Calypso card.

Product Comparison


Product	Characteristics
Prime	Multiple card applications allowed. Can coexist with non-Calypso card applications.
Light	Only one card application allowed.
Basic	Only one card application allowed.

3.5.2 Application Identifier (AID)

A Calypso card application is referred to by its Application Identifier, or AID, as defined in ISO/IEC 7816-4. A Calypso card application becomes selected when the card receives a Select Application command containing its AID.

When a card contains more than one card application, the AID is critical to enable the terminal to quickly find the relevant card application. The AID should depend only on the interoperability area (and not on the card technology or manufacturer).

It is important to avoid AID conflicts between different providers to allow the presence of different provider applications in the same card, and their successful selection by terminals.

 A Calypso card application is uniquely identified by the association of its AID and of its Calypso serial number, since:


- Card applications in the same card have different AIDs (ISO/IEC 7816-4).
- Calypso card applications in different cards have different serial numbers

 **Warning:**

- It is strongly recommended to avoid any situation where applications of a given card may have the same AID (for example with composite card such as NFC mobile phones).
- As defined by ISO/IEC 7816-5, for cards all bytes of the AID are significant including trailing zeros (if any).
- A card application AID should have an AID padded right with trailing zeros only for compatibility with terminals sending them when selecting an application.
- The size of the AID is indicated by Select Application and Get Data (with P1P2=004Fh and P1P2=006Fh).

A Calypso card application may use any AID value, however it is recommended to use an AID registered according to ISO/IEC 7816-5 or with CNA as described below.

To facilitate interoperability of Calypso card applications, CNA manages a list of *registered* Calypso AID values.

 To request a registered Calypso AID, please contact the Calypso technical support: support@calypsonet.org

There are two kinds of registered Calypso AIDs:

- *Standard Calypso AID*: in compliance with ISO/IEC 7816 5, it is made of 6 to 16 bytes, beginning with the Calypso RID, and followed by a Calypso PIX.
- *Legacy Calypso AID*: made of 8 to 16 bytes, beginning with a header of 8 characters (in ISO/IEC 8859-1 coding), and followed by a Calypso PIX.

Standard Calypso RID

Value	Description
A0 00 00 02 91h	Calypso RID, worldwide registered according to ISO/IEC 7816 5.

Registered Calypso PIX

Offset in PIX	Size	Value	Description
00h	4 bytes	YY xx xx xxh	Registered Calypso Extension. Assigned by CNA (YY different from 00h, 3xh, FFh).
04h	2 bytes	xx xxh	Optional sub-extension. Free coding proprietary to the owner of the Registered Calypso Extension. Possibly recorded in Calypso AID registry.
06h	0 to 5 bytes	00 ... 00h	Optional bytes, free coding proprietary to the owner of the Registered Calypso Extension.

Legacy Calypso 8-Byte Header

Value	Description
31 54 49 43 2E 49 43 41h	Default legacy header, used by legacy ticketing applications. As defined in Calypso Revision 2. "1TIC.ICA" in ISO/IEC 8859-1 coding.
30 45 54 50 2E 49 43 41h	Legacy header for the Store Value application. "0ETP.ICA" in ISO/IEC 8859-1 coding.
32 4D 50 50 2E 49 43 41h	Legacy header for card applications not dedicated to public transport. "2MPP.ICA" in ISO/IEC 8859-1 coding.
33 4D 54 52 2E 49 43 41h	Legacy header dedicated to the Master File. "3MTR.ICA" in ISO/IEC 8859-1 coding.

A Calypso PIX may also begin with FFh, indicating *proprietary and not registered* PIX, which size may be of 1 to 11 bytes (see below). CNA provides no uniqueness guarantee for such PIX.



For example, the owner of the Registered Calypso Extension A0 00 12 34h could decide of a sub-extension B0 01h for the first version of its card application (B0 02h for the second version, etc.). With the standard Calypso RID, the full AID would be: A0 00 00 02 91 A0 00 12 34 B0 01h.

A terminal might select a Calypso card application with the following command, omitting the second byte of the sub-extension:

Select Application with incoming data = 31 54 49 43 2E 49 43 41 A0 00 12 34 B0h

The terminal would then receive the full AID in the answer to the command, including second byte of the sub-extension, which indicates the version of this card application.

Product Comparison

Product	Characteristics
Prime	Any AID allowed.
Light	Any AID allowed.
Basic	Any AID allowed.

3.5.3 Calypso Serial Number

A Calypso serial number is an 8-byte value identifying a Calypso card, or a Calypso card application, e.g. each Calypso card application is associated to a serial number of 8 bytes.

The Calypso serial number allows a unique identification of the card or card application, among the whole range of Calypso products. It is for example used for the key diversification in the SAM, and for identifying and blacklisting card applications or cards, etc.

Two cards may not contain the same Calypso serial number.

Different Calypso card applications in the same card share the same Calypso serial number.

The two most significant bits of the Calypso serial number's 4th byte allow to identify the card product type in the following way:

Value	Product type	Assigned Range ¹
%00	Calypso Prime or Calypso Light	00 00 XX <u>0</u> X XX XX XX XXh to 00 00 XX <u>3</u> X XX XX XX XXh
%01	Calypso Basic	00 00 XX <u>4</u> X XX XX XX XXh to 00 00 XX <u>7</u> X XX XX XX XXh


¹ For Calypso HCE, the 2 first bytes of the serial number returned from the mobile phone are different from 0000h and contains the date and time of end of validity of the Debit Key.

%1x	Calypso HCE	00 00 XX <u>8</u> X XX XX XX XXh to 00 00 XX <u>F</u> X XX XX XX XXh
-----	-------------	---

For example:

- Calypso Prime or Light serial number: 0000000012345678h or 00001234F56789ABh
- Calypso Basic serial number: 00000040F0123456h or 00005678F9ABCDEF
- Calypso HCE serial number: 00000080FEDCBA98h or 000012FEF3456789A

The Calypso card manufacturers and issuers of the Calypso card or Calypso card application ensure that its Calypso serial numbers follow the rules defined in the *Calypso Technical Note #014* (ref. 070610-TN-014-SerialNumbers), including for prototypes, samples, test cards, etc. It is critical for the good working of the Calypso system that the serial number be unique to a card. The rules described in the *Calypso Technical Note #014* ensure this uniqueness.

 To request a serial number range, please contact the Calypso technical resource: support@calypsonet.org.

3.5.4 Calypso File Structures

The allowed Calypso file structures depends on the type of Calypso product.

Each Calypso card application owner (transit networks, regional authorities, etc.) may choose the best file structure for its needs among the registry or may define a new file structure (if the product allows for it). The file structure, as well as the model of the data stored in a Calypso card application, is an essential part of the interoperability of a Calypso system and should be specified by the service providers within the area of interoperability.

A Calypso card application and its DF are selected by the application name (AID). The selected card application indicates its file structure to terminals in the Application Subtype byte of the Startup Information in the answer to the Select Application command, by a reference defined in the *File Structure Registry* (ref. 060709-CalypsoFiles), or by a proprietary reference for unregistered file structures.

Product Comparison

Product	Characteristics
Prime	File structure can be freely defined by the network.
Light	Two pre-defined file structures are available.
Basic	A single pre-defined file structure is available.

3.5.5 File Identification

A file in a Calypso card application is identified by its *Long File Identifier* (LID), on two bytes. All files have an LID, unique at least within a DF. If the MF exists, it has an LID of 3F00h. The *path* of a file is the list of the LID of the DFs that contain it, followed by the file LID.

An EF may also be identified by its *Short File Identifier* (SFI), from 1 to 30, unique within a DF.

3.5.6 Current File, Current DF and Current Card Application

At any moment, one file (EF or DF) is the currently selected file, called the *Current File*. This is the default file in which operations normally occur.

If the Current File is an EF, the *Current DF* is the DF containing this file (the parent DF).

If the Current File is a DF, it is also the *Current DF*.

The *Current Card Application* is the Calypso card application currently selected on the considered logical channel. Its DF is the *Current DF*.

The selection of a file may be done by the following means:

- For a Calypso DF, by the *Select Application* command, using its AID.
- For any type of file, by the *Select File* command, using its LID (or its path, but such operating mode is not in the scope of the Calypso card product specifications).
- For an EF, by any command referencing a file with its SFI in the Current DF only. For example, the *Read Record(s)* command may specify the SFI of the file to read.

When a card has more than one DF, DFs of a card are ordered and may be selected successively in order with the *Select Application* and *Select File* commands. The order depends on the file structure, except for the MF which is always the first file in the selection order (when it is present).

3.5.7 EF Types and Data Organization

Inside an EF, data are organized in blocks called **records** or in a continuous **sequence of bytes** (only for Binary files, see below).

Access to the data, for reading or modifying, is subject to access rights described in section 7.4 *Access Conditions*.

Records

An EF organized in records may have 1 to many records organized in sequence, from record #1 to record #N (the number of records in the file).

All the records of Calypso card application files exist after initialization of the Calypso card application. A record that has never been written contains only zeroes (00h).

All the records of an EF have the same size. Different EFs may have different record sizes.

The file size (record number and size) of each EF depends on the EF and on the file structure. It is defined during initialization and cannot be modified once set.

Each record contains a single continuous sequence of data bytes from byte #0 (first byte) to byte #S-1 (last byte, for a record containing S bytes).

Record data are accessed from the first byte (#0), except for Read Record Multiple and Search Record Multiple commands when available in the Calypso product.

Linear Files

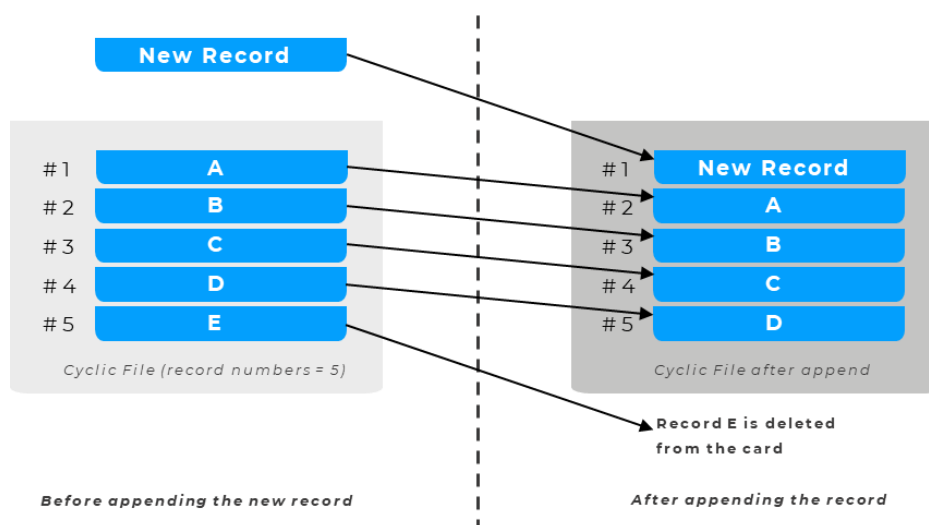
A Linear EF is made of 1 to several records.

Any record of a Linear EF may be accessed directly for reading or for modification.

Cyclic Files

A Cyclic EF is made of 1 to several records organized in a cycle, from the most recent (record #1) to the oldest (record #N).

Appending a record to the file makes it the number 1, while the others are renumbered, and the oldest one is removed.



In a Cyclic file, it is possible to read any record directly, but the only modifications possible are appending a new record and modifying the most recent record.

Counters Files

A Counters EF is made of a single record containing an ordered sequence of K counters of three bytes each, from counter #1 (bytes at offsets 0, 1 and 2 of the record) to counter #K.

In addition to usual record access operations, counter-specific operations may be done on a Counters EF:

- Increase / Increase Multiple: to add value to one, or several, counters.
- Decrease / Decrease Multiple: to subtract value from one, or several, counters.

The value of each counter is a 24-bit unsigned number, managed with big-endian convention (see section 3.5.8 *Data Representation and Coding*).

The number of counters in a Counters EF is equal to the record size, divided by 3 (e.g. 9 counters for 27, 28 or 29 bytes, 18 counters for 54 bytes). The record size need not be a multiple of 3 (bytes in excess are not accessible through counter-specific operations).

Binary Files

A Binary EF contains a single continuous sequence of data bytes from byte #0 (first byte) to byte #N-1 (last byte, for a binary file of N bytes).

Any sequence of bytes may be accessed directly for reading or for modification.

The binary files are available only in few Calypso product.

Shared EF

The Shared EF mechanism allows access the same data from different EFs of the same card application, or of different card applications.

The access to the data is controlled by the access rights of each EF: for example, both EFs might have the same access rights, or one EF could be limited to reading only.

When files share their data, all the data of the file is shared, and the files have the same number of records and record size.

The Shared EF mechanism may be used for any type of EF except the Stored Value files and the Simulated Counter Files.

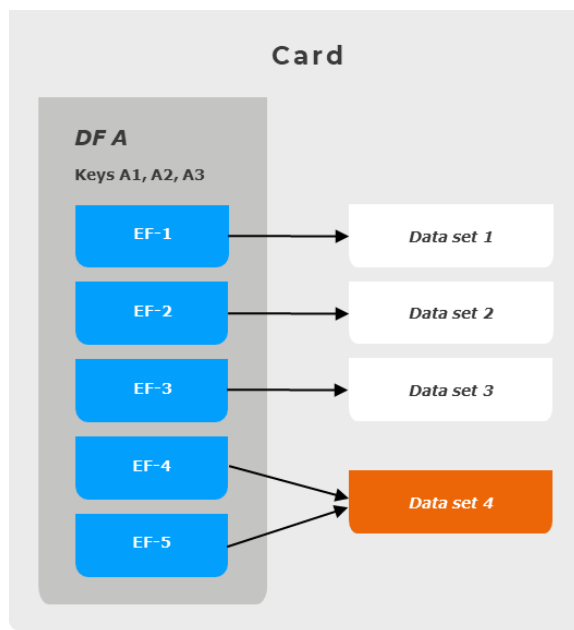
Files which share the same data have a *Data Ref* value, unique to the data shared, which allows detecting the Shared Files.

This mechanism allows for example the following file structure characteristics:

- Access to a few common files from all DF.
- Access to the same EFs, from two different card applications, with two different sets of keys.
- Access to the same file with two LID/SFI: for example, accessing the same file with LID 2050h and 2030h and with SFI 1Eh and 06h.

The Shared EF mechanism is available only in few Calypso product.

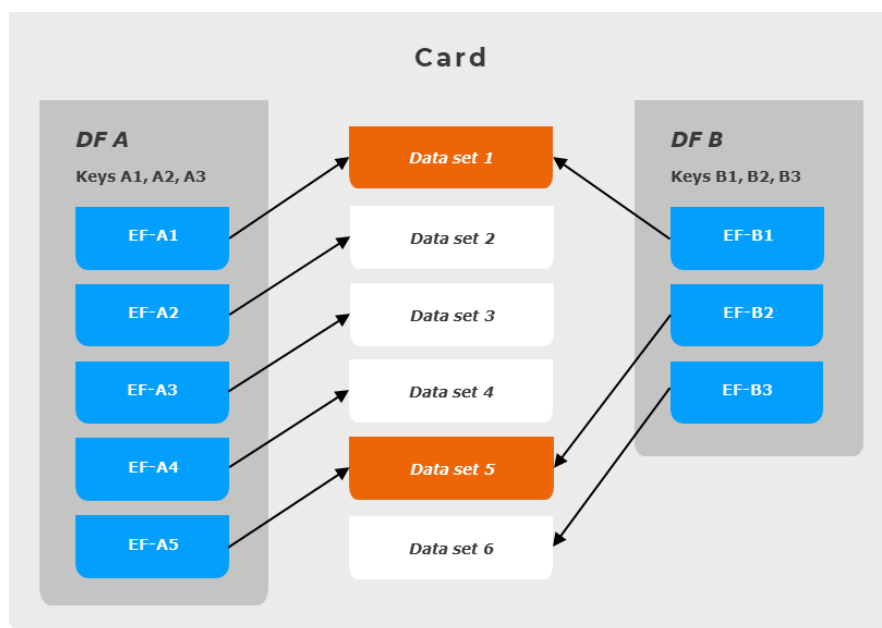
Example 1, Access to the same EF data through two EFs of the same card application:



In this example, the Calypso card application contains two files (EF-4 and EF-5) with different SFI and LID, but with the same data.

The *Data Ref* value of files EF-1, EF-2 and EF-3 is 0 (no shared data). The *Data Ref* value of EF-4 and EF-5 is identical and not 0 (for example "1").

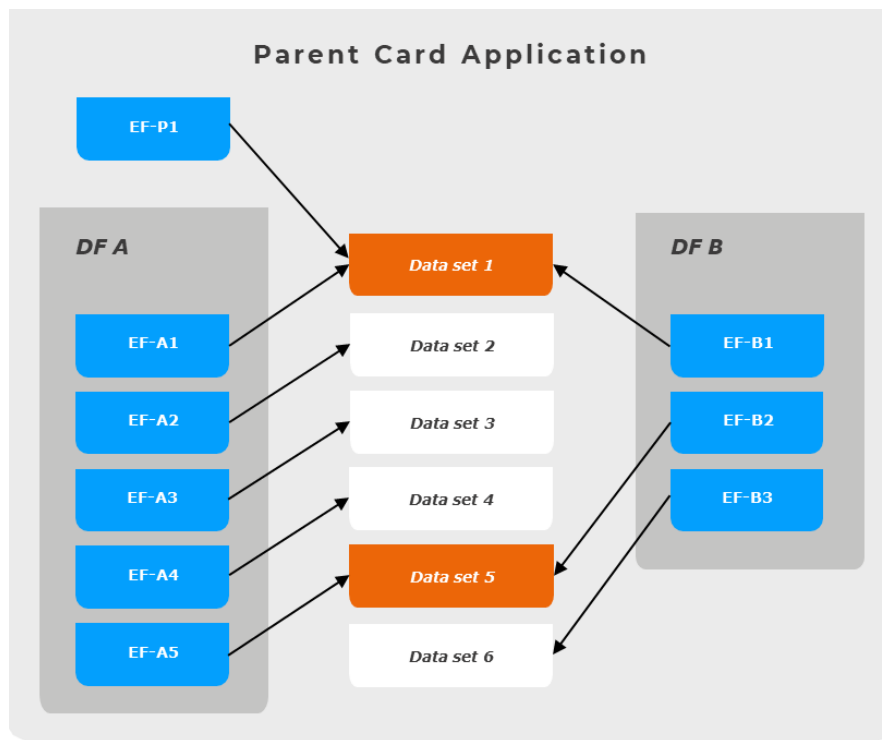
Example 2, Access to the same DF with two AIDs and two sets of keys:



In this example, the two Calypso card applications have the same file structure, and all the files are shared. Each Calypso card application is selected by its AID. This organization allows to access the same files with two sets of keys.

The *Data Ref* values of files EF-A1 and EF-B1 are not 0 and have the same value, different from the other files. The *Data Ref* values of files EF-A2 and EF-B2 are not 0 and have the same value, different from the other files, etc.

Example 3, *Access to subset of files*:



It is possible to share the file data between more than two files, and in more than two DFs.

The *Data Ref* values of files EF-P1, EF-A1 and EF-B1 are not 0 and have the same value, different from the other files. The *Data Ref* values of files EF-A5 and EF-B2 are not 0 and have the same value, different from the other files, etc.

Access Control

Access to the data, for reading or modifying, is subject to access rights described in section 7.4 *Access Conditions*.

Product Comparison

Product	Characteristics
Prime	Linear, cyclic, counters, binary files and shared EF supported.
Light	Linear, cyclic and counters files supported.
Basic	Linear, cyclic and counters files supported.

3.5.8 Data Representation and Coding

Bytes Ordering

Unless indicated otherwise in the Calypso specifications, when stored and when sent or received, the bytes are ordered with the most significant byte (MSByte) first (leftmost, lowest address, also called *Big-Endian*).

RFU Values

For a given data field, the Calypso specifications may indicate a list of possible values. All other values are reserved for future use (*RFU values*).

Unless indicated otherwise, the commands reject a parameter or command data field having an RFU value.

RFU Fields

For a given set of data fields, the Calypso specifications may indicate some fields as reserved for future use (*RFU fields*).

3.6 Integration with other Calypso Card Applications

The Calypso specifications allow for multiple applications to exist in a same card and define the links that can be implemented between two distinct Calypso card applications. Calypso allows the following type of links:

- **Stored Value application:** to allow a purse debit linked to a public transport network entrance, Calypso defines a specific card application, the Stored Value, with which other Calypso card applications may be linked. This link is presented in section 3.7 *Stored Value Application*.
- **File sharing:** several Calypso card applications may share data, seen from each card application as a regular file. This mechanism is presented in section 3.5.7 EF Types and Data Organization.

Product Comparison

Product	Characteristics
Prime	Integration with another Calypso card application allowed.
Light	Integration with another Calypso card application not allowed.
Basic	Integration with another Calypso card application not allowed.

3.7 Stored Value Application

The Calypso Stored Value application manages a Stored Value, with a specific security access.

When available, the Stored Value commands may be used directly from another Calypso card application, without an explicit selection of the Stored Value application. It may be used within a secure session, or independently.

The Stored Value may range from $-8,388,608$ to $8,388,607$. If expressed in euro cents, the value may therefore range from approximately $-83,886$ euros to $+83,886$ euros.

The possible Stored Value transactions are:

- Loading the Stored Value.
- Debiting the Stored Value.
- “Undebiting” the Stored Value (for a partial or total refund of the last debit).

Every transaction increases the Stored Value Transaction Number (SV TNum), which allows for a maximum of 65,535 operations.

The last Stored Value transactions are recorded in the Stored Value log files (Load Log and Purchase Log).

The Stored Value operations are managed by the following commands:

SV Debit	Debits the Stored Value
SV Get	Initializes a Stored Value operation (reload, debit, undebit)
SV Reload	Reloads the Stored Value
SV Undebit	Reloads the Stored Value by the amount of the last debit

Every Stored Value transaction begins with SV Get, followed by the operation to do (SV Debit, SV Reload or SV Undebit).

SV Get may also be used alone to read the Stored Value data.

[Link to other Calypso Card Applications](#)

Each Calypso Prime application of a given card may be linked to a Calypso Stored Value application of the same card, allowing synchronization of the modifications made in both applications.

A Calypso card application cannot be linked to more than one Stored Value.

Several Calypso card applications may be linked to the same Stored Value.

A Stored Value application cannot be linked to another Stored Value application.

[Product Comparison](#)

Product	Characteristics
Prime	Stored Value supported. Presence depends on the product configuration.
Light	Stored Value not supported.
Basic	Stored Value not supported.

4 CALYPSO COMPONENTS: SAM

A Calypso SAM (Secure Application Module) is a secure microprocessor smart card permanently connected with the equipment interacting with the cards. A Calypso HSM (Hardware Security Module) is a secure electronic equipment located in a remote server.

In the scope of this document, and Calypso specifications in general, any reference to a SAM also applies to an HSM, unless indicated otherwise.

In Calypso systems, the symmetric keys are always securely stored in secure devices used by the terminal or other information system. A SAM protects the keys and prevents their theft. Without a SAM, it would be possible to steal the key values and duplicate them, allowing to manufacture false cards and/or to load forged value into cards.

The SAM can authenticate a Calypso card and the data received from it and can prove to the card the authenticity of the terminal using symmetric keys.

With symmetric keys:

- The SAM performs all cryptographic computations needed to manage the Calypso commands (key derivation, MAC or signature computation, etc.).
- Modifying data in a Calypso card application requires a Calypso SAM with the secret keys corresponding to the file being modified and to the operation being done.
- Authenticating a Calypso card application and its data (read only) requires a Calypso SAM with any key of the card application.

A SAM should contain only the keys necessary for the terminal functions (personalization, loading or debiting). There are typically different kinds of SAM corresponding to the different types of terminals.

5 CALYPSO COMPONENTS: TERMINAL

5.1 Definition

A terminal or, in the terms of the ISO 24014 (IFM) standard, the "medium access device", is any device, system or piece of equipment that interacts with smart cards at any point of their life cycle. For example, validators, control terminals, point of sales terminals, personalization systems. Although not intended for them, it may also apply to manufacturing and audit equipment.

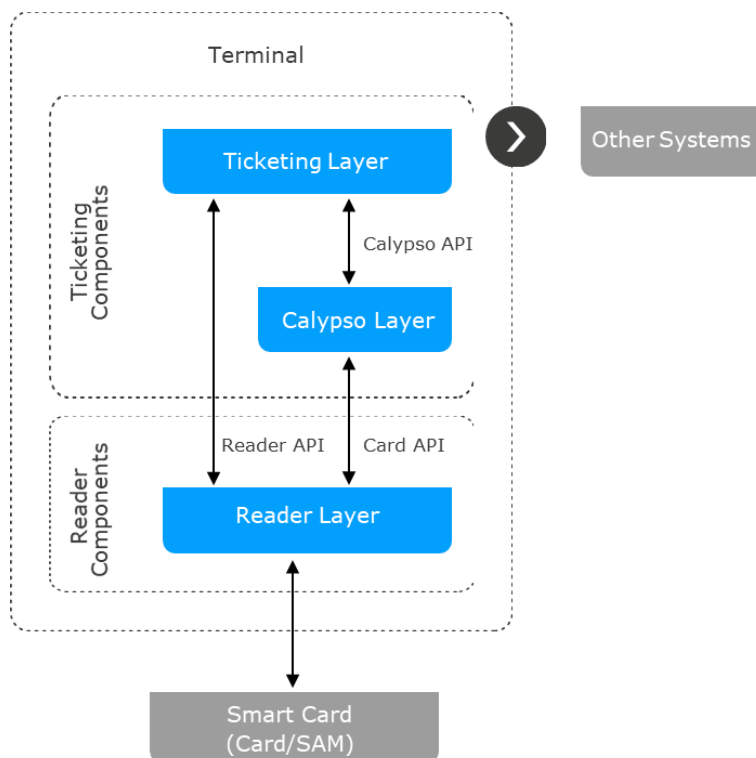
The terminal architecture is based on the following principles:

1. The assurance of the independence of the application software from the equipment on which it runs.
2. The separation of the parts that are common to all solutions from those that are specific.
3. The possibility of de-localizing some functions.
4. The use of generic technology.

- The use of the experience of industrials who have already built in, all or part of these principles.

A modular architecture is the natural result of these principles. The terminal is composed of a limited number of software modules.

For a Calypso System, the terminal can be represented by the diagram below:



The three layers provide the basic building blocks for interoperability since they allow multiple actors with distinct devices (from different suppliers) to share the use of the same set of cards in a consistent way.

5.2 Reader Layer

The Reader Layer is an abstraction layer between the software modules of a terminal that need to access smart cards and the hardware and software environment of the reader component (virtual machine -Java for example- , libraries, etc.).

This layer does not contain any specific elements for Calypso and all settings are generic, regardless of the card application (Calypso, CIPURSE, MIFARE, etc.).

The Calypso specifications define the requirements for the Reader Layer, see the *Reader Layer Requirements* document (ref. 200422-ReaderLayerRequirements).

The Calypso specifications define consistent and uniform interfaces, regardless of the reader provider (and even type of smart card), ensuring that all readers provide a minimal common set of functionalities. The two interfaces are:

- The Reader API: used by the ticketing layer to implement reader management. It defines the interfaces and classes needed to manage readers, reader events and selection mechanisms.

- The Card API: used by the layer dedicated to the functional processing of cards, in a Calypso system the Calypso Layer. It defines the interfaces and classes needed to communicate with the smart cards and specify the card selection data.

5.3 Calypso Layer

The Calypso Layer is the software module, or library, inside a terminal that is responsible for all of the Calypso related processing. It is located between the Ticketing Layer and the Reader Layer.

The Calypso Layer ensures full compliance with Calypso specifications managing all access to the card, SAM and the interaction between both. It also manages all cryptographic operations, even if they are outside the card interaction with the SAM, as long as the SAM is used for their computation (e.g. transaction signature generation, recelling operations, etc).

The Calypso Layer supports all types of Calypso card Products: Prime (native, applet or HCE), Light and Basic.

The Calypso specifications define the requirements for the Calypso Layer, see the *Calypso Layer Requirements* document (ref. 200423-CalypsoLayerRequirements).

The Calypso specifications define a high-level API, named Calypso API, to interact with a Calypso product (card, SAM, etc.). The Calypso API defines the interfaces and classes needed to operate a Calypso transaction and recover a Calypso card image.

The Calypso Layer depends on and interacts with the Reader Layer via the Card API.

5.4 Ticketing Layer

The Ticketing Layer is the software module, or library, located on the top of the Calypso Layer and the Reader Layer.

The Ticketing Layer is where we move from the universal (both the Calypso and Reader layer can and should be the same for everyone) and step into the specific. First with the Data Model processing that can be specific for the interoperability area (whether it's a network, region or country) and then with the Business Rules management that can be specific for a single operator.

The Ticketing Layer depends on and interacts with the Calypso Layer via the Calypso API and the Reader Layer via the Reader API.

The Calypso specifications defines the requirements to be followed by the Ticketing Layer, see the *Ticketing Layer Requirements* document (ref. 200430-TicketingLayerRequirements).

6 CALYPSO COMPONENTS: CENTRAL SYSTEM/BACK OFFICE

The central system, also called Back Office, keeps track of the transactions, making statistics and verifying the system security and integrity.

The central system is not in the scope of the Calypso specifications which deal with the security of the contactless interface subsystem and only provide some best practices to follow in a Calypso ticketing system, see the Ticketing Layer Requirements document (ref. 200221-TicketingLayerRequirements). The Calypso specifications do not address either the management of data exchanges between independent terminals and the central system.

CNA acts as a facilitator to implement the ticketing system but it is the operator's responsibility to integrate Calypso system within their existing IT infra-structure or use an existing integrator's AFC system.

7 SECURITY MECHANISMS

7.1 Introduction to Calypso Security

This chapter describes the security mechanisms of the Calypso card application specifications. The specific security mechanisms that are available in each Calypso product can be consulted in the dedicated product specification.

The access to data in a Calypso card application is submitted to a number of rules which may require that specific access rights be granted. These rules depend upon file access conditions specific to each file, and upon cryptographic computations using secret keys stored in the card.

Specific security mechanisms are also used to change the value of these keys, and to manage the optional PIN.

Furthermore, to handle the specific ergonomics of the contactless link, two special security features called "secure session" and "ratification" are used (and described in this chapter).

Cryptographic Algorithms

Cryptography is called **symmetric** when the sender of the protected data and its recipient use the same secret key. Typical symmetric algorithms are AES, TDES and DESX. With symmetric cryptography, it is first necessary to share a secret key, exchanged by secure means, before being able to exchange protected data.

Cryptography is called **asymmetric** when the sender of the protected data and its recipient do not use the same key but only one of a pair: a **private** key and a **public** key. The public key is characteristic of the private key: there is only one public key for a given private key, and reciprocally.

Support of symmetric cryptography is mandatory for all Calypso card applications, and support of asymmetric cryptography is mandatory only for Calypso Prime applications supporting PKI mode.

The cryptographic algorithms can be used are:

- AES, TDES and DESX based on secret keys (symmetric cryptography).

- RSA and ECDSA, based on public/private key pairs (asymmetric cryptography).

Symmetric cryptography mechanisms are described in section 7.2 *Symmetric Cryptography*, while asymmetric cryptography mechanisms are described in section 7.3 *PKI Cryptography (Asymmetric Cryptography)*. The detailed specification of these algorithms is not in the scope of this document².

7.2 Symmetric Cryptography

7.2.1 Key types

A Calypso card application contains the following symmetric cryptographic secret keys, and no other:

Key Index	Key Type	Description
Key #1	Issuer Key	This key is typically used to modify data relating to the card application itself.
Key #2	Load Key	This key is typically used as a reloading key.
Key #3	Debit Key	This key is typically used as a debit, validation or control key.

Key Hierarchy

The Issuer Key, the Load Key and the Debit Key have hierarchical rights:

- All actions allowed with the Debit Key are also allowed with the Load Key and with the Issuer Key.
- All actions allowed with the Load Key are also allowed with the Issuer Key.

Key Derivation (Key Diversification)

For the security of the system, the symmetric key values written in a Calypso card application are different in all cards. Thus, if the keys of one card become known, the keys of other cards remain secret.

To simplify the key value management, the keys of each Calypso card application are derived from *master keys*. Each derived key is computed by a cryptographic operation using the Calypso serial number. This way, keys are different in all cards and terminals have a simple way to manage all of them.

The master keys are never present outside a SAM. The derived keys are never present outside of a SAM or Calypso card.

7.2.2 Key Identifier

The keys written in a Calypso card application are identified by a public parameter: the Key Identifier, containing the key function, *KIF*, and the key version, *KVC*.

A key identifier should be unique within one interoperable system. Keys derived from the same master key share the same KIF and KVC as their master key.

² For more information, please contact the Calypso technical support: support@calypsonet.org.

In order for terminals to choose which secret key to use, they need to know the key identifier made of the KIF and KVC (to choose the correct master key), and the Calypso serial number of the card application (for the key derivation).

Terminals get the Calypso serial number with the Select Application command. The Open Secure Session command return the key identifier of the key selected for the secure session (the Select File command also allows reading all the key identifiers of a Calypso card application).

Key Function (KIF)

The KIF is a value on one byte, used to identify the purpose of the key (e.g. ticketing Issuer Key, Stored Value Debit Key, etc.).

Key Version (KVC)

The KVC is a value on one byte. It allows identification of different keys for the same key type (debit Keys for two interoperable networks, two versions of the Issuer Key, etc.).


7.2.3 Key Index

Each symmetric key of a Calypso card application is referenced by its index: #1, #2 or #3.

The Issuer Key is index 1, the Load Key index 2 and the Debit Key index 3.

7.2.4 Algorithms

For symmetric keys, a Calypso card application supports at least TDES, and may support one or several other Calypso cryptographic algorithms.


 Remark: The type of cryptographic algorithm cannot be read from the card.

AES, TDES, DESX

AES and TDES are described in ISO/IEC 18033-3:2010.

AES (Advanced Encryption Standard) is a recent encryption algorithm, more secure than DES based algorithms for transferring large amounts of data.

TDES (also called Triple-DES) is made of three successive DES operations, and uses a double DES key. It is a much stronger algorithm than simple DES, and of equivalent strength to DESX.

 DESX was introduced in international papers to prevent the brute force attack on simple DES of enumerating all the possible keys. It increases the key size by 8 bytes with a simple (and fast) operation. DESX also has the advantage that by setting the first 8 bytes of the key to zero, the DESX operation is changed to a simple DES operation.

Warning: The simple DES algorithm is obsolete and the DESX algorithm is deprecated. It is strongly recommended to use them only for compatibility with existing systems, and to update such systems to TDES or AES keys.

Signing and Ciphering Algorithms

The AES, TDES and DESX algorithms are the elementary functions of the cryptographic signing and ciphering algorithms used for the management of the secure session, and for the management of the keys.

7.3 PKI Cryptography (Asymmetric Cryptography)

7.3.1 Principles

In an asymmetric cryptography, the process is asymmetrical, depending on the kind of protection provided:

- **Authenticity:** with his private key the sender generates a **signature** of the data, sent with the data to the recipient, who uses the **public** key of the sender to verify that the signature matches the data. The sender must use a secure environment, but not the recipient.
- **Confidentiality:** the sender encrypts the data with the **public** key of the recipient, sends the result to the recipient, who uses his **private** key to decrypt the data. The recipient must use a secure environment, but not the sender.
- **Combined:** the data are signed with the **private** key of the sender and also encrypted with the **public** key of the recipient. Both the sender and the recipient must use a secure environment.

Product Comparison

Product	Characteristics
Prime	PKI Cryptography supported.
Light	PKI Cryptography not supported.
Basic	PKI Cryptography not supported.

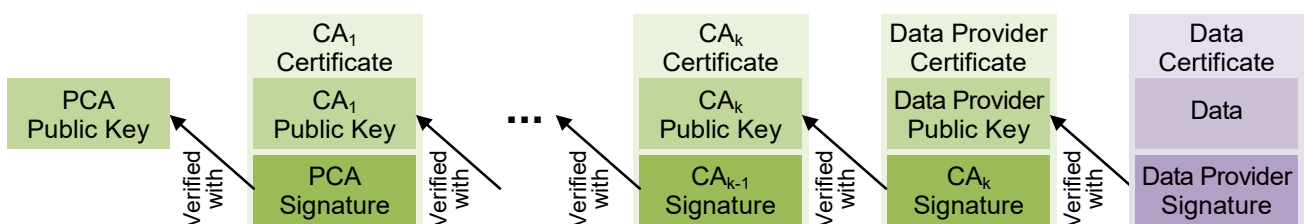
7.3.2 Public Key Infrastructure (PKI) Overview

With asymmetric cryptography, exchanging protected data only requires exchanging public keys. Such secure information system is called a **Public Key Infrastructure** (“PKI”).

To ensure ownership, **public key certificates** are used. A public key certificate is a set of data containing the value of a public key, usually an identifier as well as some other data (e.g. end of validity date), and a signature. The signature is generated by a third party called a **certification authority** (or “CA”), trusted by both the owner of the key pair and the recipients of the public key certificate.

It is then possible to build a secure information system based on a chain of certification authorities, from the implicitly trusted (here called the **primary certification authority** or “PCA”) to the provider of authentic data.

Certification Authorities Trust Chain



The recipient gets from the sender the signed data and all public key certificates of the trusted chain. Knowing only the public key of the PCA, the recipient is able to verify the authenticity of the data (e.g. the public key of a Calypso Prime PKI application).

Calypso Prime PKI Application Trust Chain

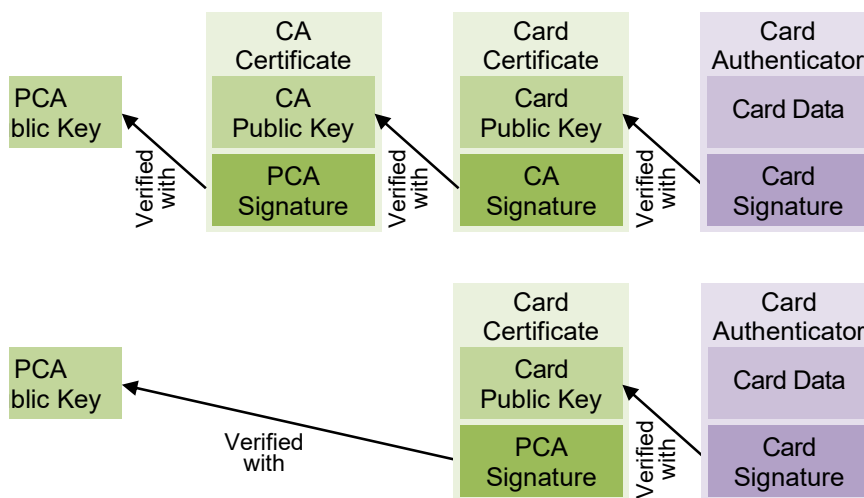
In Calypso Prime PKI mode scheme, a trust chain ends with the public key of the Calypso Prime PKI application (it is not a certification authority), and may contain at most one certification authority other than the primary certification authority.

Therefore, an autonomous PKI application contains:

- Its own key pair.
- The corresponding public key certificate.
- Optionally (depending on the trust chain of the project), the public key certificate of a certification authority.

Since a Calypso Prime PKI application does not verify the certificates that it contains, it does not contain the public key of the primary certification authority.

Calypso Application Trust Chains



The primary certification authority of a PKI application may be part of a larger trust chain, out of scope for the PKI application (and this specification).

7.3.3 Algorithms

Two asymmetric algorithms are used for PKI mode:

- RSA: authentication of public key certificates (the authenticated public key may be an RSA key or an ECC key).
- ECDSA: authentication the PKI application, using ECC keys.

7.4 Access Conditions

7.4.1 Principles

Each file is associated to its *access conditions*: access mode and key index for each group of commands.

Calypso access modes and groups of commands defined are detailed in following sections.

Access control mechanisms apply identically to symmetric cryptography (regardless of the symmetric algorithm) and to asymmetric cryptography, with the following restrictions:

- Access to files in Confidential mode or in Confidential & PIN mode requires using Extended mode with an AES key.
- With PKI mode (asymmetric cryptography), no modification of the Calypso card application is allowed unless the access mode is Always.

7.4.2 Groups of Commands

The Calypso commands subject to access conditions are listed below:

- Invalidate.
- Rehabilitate.
- Append: Append Record.
- Read: Read Binary, Read Record(s), Read Record Multiple, Search Record Multiple.
- Update: Update Binary, Update Record.
- Write: Write Binary, Write Record.
- Decrease: Decrease, Decrease Multiple.
- Increase: Increase, Increase Multiple.
- Put Data.

The commands subject to access conditions are divided into four groups, as follows:

File Type	Description			
	0	1	2	3
DF	Rehabilitate	Invalidate	–	–
Linear EF Binary EF	Read	Update	Write	–
Cyclic EF	Read	Update	Write	Append
Counters EF Simulated Counter EF	Read	Update	Decrease	Increase

The Put Data command is not included in a group because its access rights are implicit.

The Stored Value commands (SV Get, SV Debit, SV Reload and SV Undebit) are not subject to access conditions (see section 3.7 *Stored Value Application*).

7.4.3 Access Modes

The possible access modes for each group of commands are:

Access Mode	Code	Description
Always	1Fh	Free access: access rights are always granted.
Never	00h	Access forbidden: access rights are never granted.
Session	10h	Access for modification is possible only if inside a secure session, using the corresponding key or a key of lower index.
Confidential	14h	Access for reading and access for modification are possible only inside a secure session with encryption active using the corresponding key or a key of lower index (see Extended mode). <i>This access mode is available only for EFs.</i>
PIN	01h	Access for reading is granted only if the PIN code has been previously successfully verified by the card application. <i>This access mode is available only for Read (group 0 of an EF), and only in Calypso card applications supporting the PIN.</i>
Confidential & PIN	15h	Access for reading is only possible only if the following two conditions are fulfilled: - Access inside a secure session with encryption active using the corresponding key or a key of lower index (see Extended mode). - The PIN code has been previously successfully verified by the Calypso card application. <i>This access mode is available only for EFs, only for group 0, and only in Calypso card applications supporting the PIN.</i>

Product Comparison

Product	Characteristics
Prime	All modes supported.
Light	Always, Never and Session supported.
Basic	Always, Never and Session supported.

7.5 Secure Session Description

This section describes secure sessions when done with symmetric cryptography. See section 7.6 *PKI Security Mechanisms* for secure sessions with asymmetric cryptography.

7.5.1 Secure Session Security

Using a symmetric key, the secure session performs simultaneously:

- the authentication of the Calypso card application,
- the authentication of the terminal,

- the authentication of all data exchanged during the secure session,
- the proof that the requested modifications have been done in the card,
- optionally, intermediary mutual authentications before the secure session closing,
- optionally, encryption of all card application data exchanged during the secure session.

These operations are done with an optimized algorithm to allow a very quick transaction. This is particularly important when using the card with a contactless terminal used for validation.

A secure session is started by the Open Secure Session command and closed by the Close Secure Session command. The Open Secure Session command indicates which key is used during the secure session. The Close Secure Session command does the unique and global authentication by mutual exchange of a cryptogram called a *MAC (Message Authentication Code)*.

Many commands may be given during the secure session. All the commands and data exchanged during the secure session are included in the MAC (as described in the next section). The data exchanged include a challenge (random number) sent by the terminal, and another one generated by the card.

At the end of the exchange, the high-order bytes of the MAC are sent by the terminal to the card, proving its authenticity to the card. The card then validates the data modifications received, and validates the transaction.

After this internal recording, the card sends the remaining lower bytes of the MAC to the terminal, proving its authenticity and the actual recording of the transaction.

All data modification commands processed during the secure session are automatically canceled if the final authentication fails, or is not done. The data modification commands are:

- Append / Update / Write Record,
- Update / Write Binary,
- Decrease / Increase (and Multiple),
- Invalidate and Rehabilitate,
- Put Data,
- SV Debit, SV Reload and SV Undebit.

Thus, the secure session mechanism ensures that the modifications made during the secure session are *all completely and correctly done*, or else that *none are done*. If the secure session is aborted or not successfully closed (because of an incorrect MAC, a card error, an unexpected shut down, etc.), then all modifications done during the secure session are cancelled.

Depending upon the secure session opening parameters, it is also possible to:

- Perform mutual authentications during a secure session, allowing for example to authenticate the card application before accessing its data.
- Protect the card application data against eavesdropping, with on-the-fly encryption.

Furthermore, a feature named the “*ratification*”, allows the terminal to handle gracefully a possible communication link problem (see section 7.7 *Ratification*).

These rules apply in exactly the same way in contactless and in contact modes.

Warning:

- During one secure session, the maximum amount of data modifications that may be done is limited. A Calypso card application mandates a minimum of possible modifications per secure session.
- If the DF changes during the secure session, the secure session is automatically aborted by the card.

7.5.2 Session MAC Authentication

The command *Open Secure Session* opens a secure session with the key indicated.

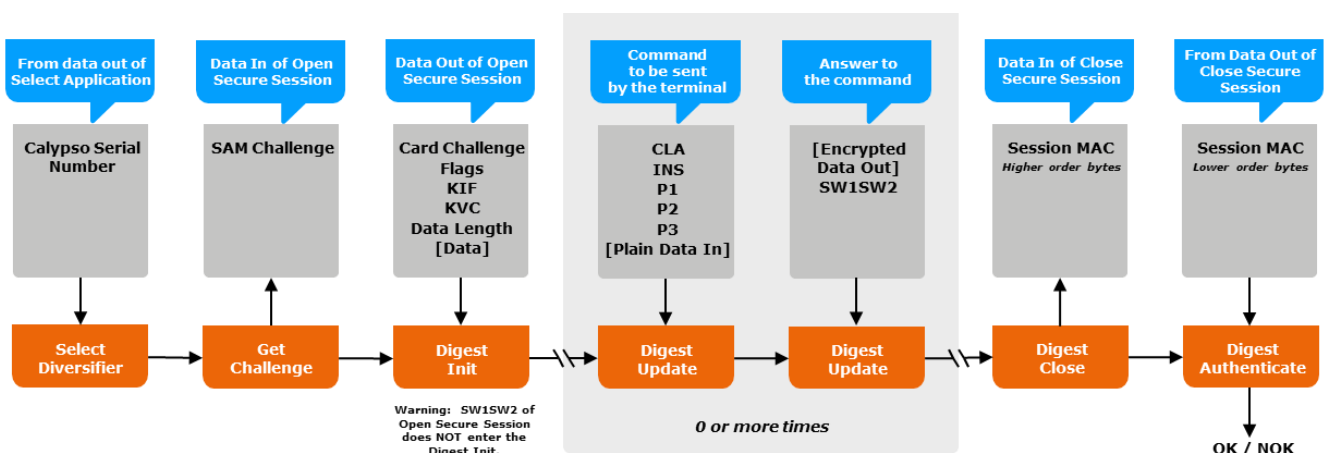
When Open Secure Session is used with a symmetric key (indicated by its index: 1, 2 or 3), the secure session is authenticated with a MAC (Message Authentication Code) computed using the indicated key and the data exchanged during the secure session. It is called the **Session MAC**.

During the secure session, all data received and sent by the card are processed in a MAC computation algorithm. When the secure session closes, the computation ends and provides the Session MAC.

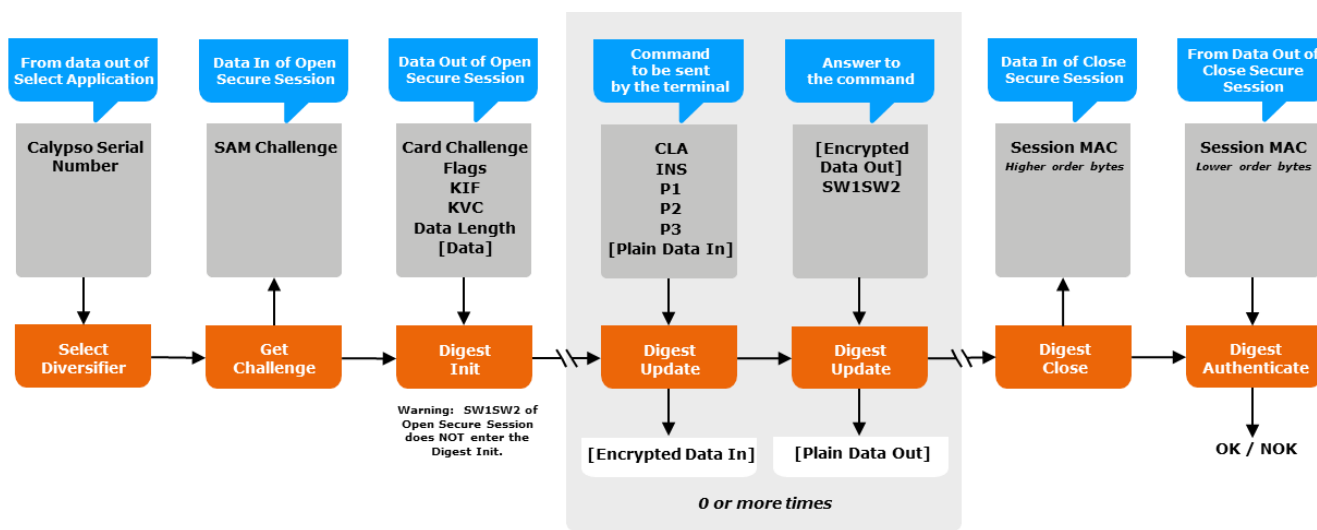
The Session MAC is computed within the card and SAM. The computation algorithm is based on an ISO/IEC 9797-1 algorithm for AES and TDES keys (Algorithm 1 with Padding method 2, see also section 7.2.4 *Algorithms*) and on a Calypso hash algorithm for DESX and simple DES keys.

The terminal sends the higher order bytes of the Session MAC to the card. If these higher order bytes are correct, the card validates the changes made to the card application data during the secure session, and sends back the lower order bytes of the Session MAC to the terminal.

In contactless mode, when the secure session is closed, the changes made to the card are not *ratified* until the next Calypso command. When a secure session is opened, the Calypso card application informs the terminal whether the last secure session was ratified or not.



Secure session operations with SAM commands indicated, symmetric cryptography, without encryption



Secure session operations with SAM commands indicated, symmetric cryptography, with encryption

i See also:

- For usual problems in terminal software resulting in an incorrect Session MAC: *Calypso Layer Requirements* (ref.200423-CalypsoLayerRequirements).
- For a secure session example, with the complete and detailed commands exchanged with a Calypso card application and with a Calypso SAM: *Calypso Technical Note #313* (ref.081010-TN-313-SecureSessionExamples).

Session Events

The secure session management events are:

- *Open*: successful Open Secure Session command.
- *Close*: successful Close Secure Session command having a Session MAC or a Session Signature. The closed secure session is no longer running.
- *Abort*: any event that ends a secure session without closing it. The modifications done during the secure session are canceled.
- *Ratify*: the ratification state is set to “ratified”.

A secure session is said to be *running* after being opened and until being closed or aborted.

7.6 PKI Security Mechanisms

7.6.1 Overview

The Calypso security using asymmetric keys is based on the usual secure session mechanism. Such secure session is called a **PKI Session**.

Using an asymmetric key, the secure session performs simultaneously:

- the authentication of the PKI application of the card, and
- the authentication of all data exchanged during the secure session.

It can be processed by terminals *without* any SAM, since it requires no secret data.



Warning: A PKI Session allows authenticating the PKI application and its data, but not their modification.

The PKI application does only the following asymmetric computations: ECC key pair generation or verification, Session Signature generation. Any other asymmetric computations (public key certificate generation or verification, Session Signature Verification) are done by other devices or systems: terminal, server, SAM, HSM, etc.

The PKI Session is very close to secure sessions with symmetric keys (see section 7.5 *Secure Session Description*), replacing the Session MAC by the Session Signature.

7.6.2 PKI Session

The secure session unfolds as described in section 7.5.1 *Secure Session Security*, except that at the end of the secure session the terminal provides no cryptogram. Only the PKI application provides its Session Signature proving its authenticity and the authenticity of any data exchanged.

No SAM or HSM is necessary since the key used to verify the Session Signature is a public key.

Since the terminal is not authenticated, data of the PKI application can be modified only in “Always” access mode, and a PKI Session is always closed in “*ratified*” state.

Unlike Session MAC computation, successful Open Secure Session commands fully enter the Session Signature, as any other command during the secure session.

There is no encryption mode for PKI Sessions.

7.7 Ratification

During any communication, it may happen that the link is broken unexpectedly. This is particularly true during contactless communication, where the card may be taken out of the terminal contactless field during normal use, and before the completion of the transaction.

The secure session is a very efficient mechanism to solve this problem, as an interruption before the secure session closing cancels all the modifications done to the card, leaving it in the same state as it was before the secure session.

However, after the end of the secure session, and the validation of the changes by the card, the acknowledgement (including the card Session MAC) still has to reach the terminal. If the communication link is broken between the secure session closing, and the good reception of this acknowledgement, the terminal has no proof that the card is legitimate and that the transaction succeeded. In this case, the user might have paid, or have its transport rights decreased, and not be allowed entrance to the network.

The odds of occurrence are small (a few milliseconds during a transaction of 100 to 200 ms). However, on several hundred thousand transactions every day, this problem will sometime occur.

For example:

The holder of a Calypso card, containing a contract with 10 trips, tries to enter the transport network. The card is presented to a terminal, but with a gesture too fast (or too far from the antenna) so that the terminal only has the time to process the transaction up to sending the Close Secure Session command. The card is taken out of the field in the middle of the contactless frame carrying the response of the command.

As the Close Secure Session command has been fully processed by the card, the contract has been decreased to 9 trips. But as the response of the command did not reach the terminal, the terminal did not allow the access.

Believing the terminal is out of order, the holder tries to use another terminal. This terminal may allow the access, but it should analyze the content of the Calypso card application to determine whether to perform the debit or not, taking the risk of decreasing the contract a second time, and of refusing access for a few minutes before this second decrease.

This kind of problem occurs in any transactional system, and particularly in all contactless transactions.

Usual Solution

For a single terminal, the usual solution to this problem is for the terminal to remember the cards that might fall in this case, and to handle them properly if they are presented again soon after.

The problem is even more complex in transport networks, where many terminals may control the same network entrance, and where the user will often try another terminal if the previous one failed to open the gate.

In this case, the second terminal does not know about the previous attempt and applies a standard algorithm. For example:

- If the previous transaction was at another entrance or is too old, the terminal assumes that it is a new transaction: it debits the card again and grants the access.
- If the previous transaction was at the same entrance and is recent, the terminal forbids the access to the transport network because it assumes that the user tried a second entry with a card already used for entrance.

The terminal then takes the risk to reject a legitimate user who has just made a transaction too fast on another terminal.

To allow the user to enter the network without paying twice, while avoiding this very complex management in the terminals of a network at entrance or exit, the **ratification** mechanism was designed, as described hereafter.

Ratification Solution

The ratification works as follows:

- Step 1 On session closing by the Close Secure Session command, the card records the secure session as “not ratified”. The card then sends the secure session closing acknowledgement to the terminal (which contains the card Session MAC).

Step 2 As soon as possible after the receipt of the card secure session closing acknowledgement, the terminal sends a new command to the card acting as the terminal acknowledgement³. Without waiting for the card answer⁴ the terminal asks the SAM to verify the card Session MAC. According to the SAM answer, the terminal decides whether to grant the access to the transport network (if granted: presents a “green light”, open the gate, etc.).

Step 3 Two possibilities:

- If the card is removed from the terminal contactless field before reception of the new command, the state of the recorded secure session remains “*not ratified*”, meaning that the card has *not* received the terminal acknowledgement to the secure session closing, until modified by a new Close Secure Session command. The odds of occurrence are small, providing that the time between steps 1 and 2 is made as small as possible by the terminal.
- Otherwise, on receipt of the new command, the card changes the state of the recorded secure session to “*ratified*”, meaning that the card has received the terminal acknowledgement to the session closing. The card is then removed from the terminal contactless field.

When a secure session is opened, the card returns the state of the previous secure session.

The secure session remains in the “*not ratified*” state only if the communication is broken after card records the secure session as “*not ratified*” and before it changes the state of the recorded secure session to “*ratified*”. The probability of occurrence is small, because the corresponding duration is very short.

The ratification mechanism allows a second terminal to take the following actions:

If the previous transaction was at another entrance or is too old, the terminal assumes that it is a new transaction regardless of the ratification state: it debits the Calypso card application and grants the access.

If the previous transaction was a granted access at the same entrance and is recent then, according to the ratification state:

Ratified: The terminal forbids the access⁵, **without any risk to reject a legitimate user**, as it knows that the previous terminal completely processed the transaction.

Not ratified: The terminal grants the access without debiting the card again, letting **all legitimate users enter the network**.

³ Using the ISO/IEC 14443 deselection frame S(DESELECT) works but it is now deprecated. Moreover, this ratification procedure is not supported by Calypso Prime applications not able to ratify on deselection, by Calypso Light products and by Calypso Basic products.

⁴ Since the terminal must verify the card session MAC (then grant or deny access accordingly) even if no response is received from the card, there is no need to wait for this response. Furthermore, waiting for this response would increase the gate opening delay (by as much as the card removal detection delay, which may be of more than 0.5s, depending on the card and on the terminal).

⁵ Or debits the card again, depending on the network policy regarding multiple presentation of the same card.

Notes:

- Terminals should always use the card application data to grant access (being “not ratified” is not enough).
- When the terminal rejects a card and closes a secure session, it should close the secure session in “ratified” mode (P1=80h) instead of “not ratified” mode (P1=00h).
- In contact mode and in PKI mode the secure session is always closed in “ratified” state.
- The recommended command to use to change the ratification state is a Read Records command with an incorrect P2.

Security Issue


When a terminal grants the access without debiting the card, it risks allowing someone to enter twice if the previous transaction was interrupted after reception by the terminal of the close secure session acknowledgement, but before reception by the card of the new command sent by the terminal.

This risk is extremely small because it is very difficult to prevent the ratification while opening the gate (the communication must be broken during a very short time interval, which cannot be replicated consistently manually). However, attacks based on this feature could theoretically be improved by manufacturing a specific electronic equipment cutting the communication on purpose at the right moment.

Furthermore, such fraud attempts would be easily detectable by the central system.

7.8 Transaction Counter

To prevent unlimited use of the secret keys, and to ensure uniqueness of secure session cryptograms, a counter is modified at each command involving a secret key.

-  The association of the AID, the serial number and the Transaction Counter identifies uniquely all secure operations done with any given Calypso card application.

7.9 Memory Modification Management

In order to ensure that the writing and erasing in non-volatile memory cannot be corrupted by an unexpected shutdown, Calypso card applications implement an automatic recovery mechanism. All data written during a secure session are either all completely and correctly written in the Calypso card, or not written at all.

8 ANNEX

8.1 Calypso References

The latest edition of the referenced document (including any related amendments) applies.

<i>010209-MU-CalypsoCardSpec</i> <i>060908-MU-CalypsoR2Amd1</i>	Calypso Specification – Card Application Revision 2/Amendment 1 – “Calypso Prime Revision 2”
<i>060708-SP-CalypsoPrime</i>	Calypso Specification – Calypso Prime
<i>060709-CalypsoFiles</i>	Calypso Specification – File Structure Registry
<i>101010-SamCalypso</i>	Calypso Specification – SAM-C1 User Manual
<i>101101-HoplinkAppli</i>	Calypso Specification – Hoplink Specifications
<i>130507-UM-CalypsoCardApplet</i>	User Manual – Calypso Card Applet
<i>141113-CalypsoHCEApplication</i> <i>180314-CalypsoHCEGenSpecs</i>	Calypso Specification – Host Card Emulation Application – “Calypso HCE Application”
<i>150422-CalypsoHCEGuidelines</i>	Calypso Specification – HCE Guidelines
<i>170101-CalypsoLightApplication</i>	Calypso Specification – Calypso Light
<i>170202-SystemArchitecture</i>	Calypso Specification – Security Architecture and Key Ceremony
<i>191011-CalypsoBasic</i>	Calypso Specification – Calypso Basic
<i>200422-ReaderLayerRequirements</i>	Calypso Specification – Reader Layer Requirements
<i>200423-CalypsoLayerRequirements</i>	Calypso Specification – Calypso Layer Requirements
<i>200430-TicketingLayerRequirements</i>	Calypso Specification – Ticketing Layer Requirements
<i>000907-TN-001-StartupInfo</i>	Technical Note #001 – Calypso Startup Information – Specification and Management
<i>CalypsoTN007-OffLineSamLoad</i>	Technical Note #007 – SAM – Off-Line Update of Keys and Ceilings
<i>070610-TN-014-SerialNumbers</i>	Technical Note #014 – Calypso Serial Numbers – Management Rules
<i>110404-TN-016-CardDataSign</i>	Technical Note #016 – Card Data Signature – Management
<i>CalypsoTN017-SamSupply</i>	Technical Note #017 – SAM Power Supply
<i>171208-TN-021-CnaPublicKeys</i>	Technical Note #021 – Calypso PKI – CNA Public Keys
<i>081010-TN-313-SecureSessionExamples</i>	Technical Note #313 – Secure Session Examples
<i>CalypsoTN315-StoredValue</i>	Technical Note #315 – Stored Value Guidelines
<i>110530-TN-318-PrePerso</i>	Technical Note #318 – Pre-Personalization – Symmetric Key Loading Process
<i>CalypsoTN319-SvSupervision</i>	Technical Note #319 – Stored Value Supervision

<i>190606-TN-323-Initialization</i>	Technical Note #323 – Initialization– Symmetric Key Loading Process
<i>190820-TN-325-PkiModeExample</i>	Technical Note #325 – PKI Mode Example – Certificates and Session Signature Verification

8.2 Normative References

For references with a date or a version, only the edition cited applies. For other references the latest edition of the referenced document (including any related amendments) applies.

<i>CEN/TS 16794-1</i>	Public transport - Communication between contactless readers and fare media - Part 1: Implementation requirements for ISO/IEC 14443
<i>CEN/TS 16794-2</i>	Public transport - Communication between contactless readers and fare media - Part 2: Test plan for ISO/IEC 14443
<i>EMV 4.3 - Book 1</i>	EMV Specifications Book 1 - Application Independent ICC to Terminal Interface Requirements
<i>EMV Contactless v2.2</i>	EMV Contactless Specifications for Payment Systems
<i>EN 1545-1</i>	Identification card systems - Surface transport applications - Part 1: Elementary data types, general code lists and general data elements
<i>EN 1545-2</i>	Identification card systems - Surface transport applications - Part 2: Transport and travel payment related data elements and code lists
<i>FIPS PUB 180-4</i>	Specification of Secure Hash Standard (SHS)
<i>FIPS PUB 186-4</i>	Specification of the Digital Signature Standard (DSS)
<i>ISBN 3-540-61512-1</i>	DESX algorithm definition. Kilian & Rogaway, "How to Protect DES Against Exhaustive Key Search", from CRYPTO'96. Advances in cryptology 16th annual international conference, Springer Verlag
<i>ISO/IEC 3166-1</i>	Codes for the representation of names of countries and their subdivisions - Part 1: Country codes
<i>ISO/IEC 7816-1</i>	Identification cards - Integrated circuit(s) cards with contacts - Part 1: Physical characteristics
<i>ISO/IEC 7816-2</i>	Identification cards - Integrated circuit(s) cards with contacts - Part 2: Dimensions and location of contacts
<i>ISO/IEC 7816-3</i>	Identification cards - Integrated circuit cards - Part 3: Cards with contacts – Electrical interface and transmission protocols
<i>ISO/IEC 7816-4</i>	Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange

<i>ISO/IEC 7816-5</i>	Identification cards - Integrated circuit cards - Part 5: Registration of application providers
<i>ISO/IEC 8859-1</i>	Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1
<i>ISO/IEC 9797-1:2011</i>	Message authentication techniques - Part 1: Mechanisms using a block cipher
<i>ISO/IEC 10118-3:2018</i>	IT Security techniques -- Hash-functions -- Part 3: Dedicated hash-functions
<i>ISO/IEC 14443-1</i>	Cards and security devices for personal identification- Contactless proximity objects - Part 1: Physical characteristics
<i>ISO/IEC 14443-2</i>	Cards and security devices for personal identification- Contactless proximity objects - Part 2: Radio frequency power and signal interface
<i>ISO/IEC 14443-3</i>	Cards and security devices for personal identification- Contactless proximity objects - Part 3: Initialization and anticollision
<i>ISO/IEC 14443-4</i>	Cards and security devices for personal identification- Contactless proximity objects - Part 4: Transmission protocol
<i>ISO/IEC 14888-2:2008</i>	IT Security techniques -- Digital signatures with appendix -- Part 2: Integer factorization based mechanisms
<i>ISO/IEC 14888-3:2018</i>	IT Security techniques -- Digital signatures with appendix -- Part 3: Discrete logarithm based mechanisms
<i>ISO/IEC 18033-3:2010</i>	Information technology -- Security techniques -- Encryption algorithms -- Part 3: Block ciphers
<i>ISO/IEC TS 24192 1</i>	Cards and security devices for personal identification – Communication between contactless readers and fare media used in public transport – Part 1: Implementation requirements for ISO/IEC 14443 (all parts)
<i>ISO/IEC TS 24192 2</i>	Cards and security devices for personal identification – Communication between contactless readers and fare media used in public transport – Part 2: Test plan for ISO/IEC 14443 (all parts)
<i>SEC 2: Recommended Elliptic Curve Domain Parameters</i>	Standards for Efficient Cryptography – SEC 2: Recommended Elliptic Curve Domain Parameters – Certicom Research (http://www.secg.org/sec2-v2.pdf)

8.3 Glossary and Acronyms

AES	<i>Advanced Encryption Standard</i> (as defined in ISO/IEC 18033-3): symmetrical cryptographic algorithm using 128-bit data and key.
AFC	<i>Automated Fare Collection</i> : automated ticketing system of a public transportation network.
AFI	<i>Application Family Identifier</i> (as defined in ISO/IEC 14443-3).
AID	<i>Application Identifier</i> : value unique in a card, allowing to unambiguously identify a card application, as defined in ISO/IEC 7816-4 and ISO/IEC 7816-5.
AKS	<i>Android Key Store</i> : custom Java Security Provider in the KeyStore facility which allows generating and saving private keys that may be seen and used only by a specific app.
AM	<i>Identification Authorization Module</i> : HAM delivered to the Application Provider which includes activation rights and a range of Calypso serial numbers
APDU	<i>Application Protocol Data Unit</i> (as defined in ISO/IEC 7816-4).
Applet	Application which may be loaded into a card (usually associated with the Java environment).
APSD	<i>Application Provider Security Domain</i> .
ATR	<i>Answer To Reset</i> : data returned by the card during startup.
C-MAC	<i>Command MAC</i> .
CA	<i>Certification Authority</i> : entity allowed to provide public key certificates.
CAAD	Abbreviation for <i>Card Access Authorization Descriptors</i> , a feature of the SAM which allows the secure sharing of the same card application with a unique set of key.
Calypso Card	Within any card, a secure element containing at least one application compliant with one of the Calypso card specifications
Card	Any portable device having an ISO/IEC 14443 interface. In the present document “card” always means “Calypso card” (unless indicated otherwise). Formerly called <i>portable object</i> .
CardCert	Public key certificate of a PKI application.
CD97	Ticketing card upon which this specification is based.
CNA	<i>Calypso Networks Association</i> .
CMS	<i>Card Management System</i> .
CSM	<i>Calypso Secure Module</i> .

CSN	<i>Calypso Serial Number</i> : unique serial number given to a Calypso card.
Cryptogram	Message in code, resulting from a cryptographic operation.
Dedicated File	(DF) Equivalent of a directory. A DF contains other files.
DES	Ciphering algorithm producing 8 bytes of data from 8 input bytes, using a 7 bytes key (as defined in <i>ANSI X3.92-1981</i>). Also called “simple DES”, now deprecated.
DESX	Ciphering algorithm producing 8 bytes of data from 8 input bytes, using a 15 bytes key (as defined in <i>How to Protect DES Against Exhaustive Key Search</i> by Kilian & Rogaway).
DF	<i>Dedicated File</i> (as defined in ISO/IEC 7816-4).
DGI	<i>Data Grouping Identifier</i> .
ECC	<i>Elliptic Curve Cryptography</i> : asymmetric cryptography using public-private key pairs based on the algebraic structure of elliptic curves over finite fields, as defined in ISO/IEC 14888-3:2018 standard.
ECDSA	<i>Elliptic Curve Digital Signature Algorithm</i> (also “EC-DSA”): cryptographic algorithm for generation and verification of digital signatures based on ECC, as defined in ISO/IEC 14888-3:2018 standard.
EF	<i>Elementary File</i> (as defined in ISO/IEC 7816-4).
Elementary File	(EF) File containing data. The types of EF defined by Calypso are: Linear, Cyclic, Counters, Simulated Counter and Binary files.
EMV	<i>Europay MasterCard Visa</i> .
Error counter	Counter of incorrect PIN presentations.
eSE	<i>Embedded Secure Element</i> .
FCI	<i>File Control Information</i> .
FCP	<i>File Control Parameters</i>
GP	<i>GlobalPlatform</i> .
GP-CMS	<i>GlobalPlatform Card Management System</i> .
GTML	Contactless ticketing card upon which Calypso specification were originally based.
HAM	Generic name for an HCE Authorization Module (AM and KM).
HAP	<i>HCE Application Provider</i> .
HCE	<i>Host Card Emulation</i> .

HCS	<i>Hardware Credential Storage</i> : Android feature providing more security by making cryptographic private keys unavailable for extraction, also called hardware-backed key store (HBK).
HSM	<i>Hardware Security Module</i> : multi-channel SAM used in central systems.
ICS	<i>Implementation Conformance Statement</i> : structured document which lists all information needed for a product identification and for the progress of an evaluation process (implementation options, configuration details ...).
ICV	<i>Initial Chaining Vector</i> .
ISD	<i>Issuer Security Domain</i> .
ISO/IEC	<i>International Organization for Standardization / International Electrotechnical Commission</i> .
KIF	<i>Key Identifier</i> : value identifying the type of key.
KVC	<i>Key Version and Category</i> . Arbitrary value identifying a key among several of the same type.
KM	<i>Key Authorization Module</i> . HAM delivered to the service provider which includes the Calypso master debit key of the network for CSN and debit key renewal.
LA	<i>Local Application</i> .
Long File Identifier (LID)	External unique number identifying a file (0000h to FFEh, without 3FFh). All files have an LID. The MF has an LID of 3F00h.
LID	<i>Long File Identifier</i> .
LSB	<i>Least Significant Byte</i> .
M/O	<i>Mandatory/Optional</i> .
MAC	<i>Message Authentication Code</i> . Cryptogram computed from a symmetric key (secret key) and some data, which authenticates these data.
MIDP	<i>Mobile Information Device Profile</i> .
MSB	<i>Most Significant Byte</i> .
MNO	<i>Mobile Network Operator</i> .
N/A	<i>Not Applicable</i>
NFC	<i>Near-field Communication</i> .
NID	<i>Network Identifier</i> : allows the validator to select the eligible contracts in the Hoplink application.

OTA	<i>Over-the-Air.</i>
OTI	<i>Over-the-Internet.</i>
PCA	<i>Primary Certification Authority:</i> first certification authority of a trust chain. Its public key is implicitly trusted. If it provides a certificate for its public key, this certificate is verified with the public key itself (i.e. the certificate is auto-signed).
PKI	<i>Public Key Infrastructure:</i> system ensuring information security based on asymmetric cryptography, which allows protecting data by sharing only public keys.
PTA	<i>Public Transit Authorities.</i>
PTO	<i>Public Transport Operators.</i>
R-MAC	<i>Response MAC.</i>
RAM	<i>Remote Application Management.</i>
Record	The data in the files are organized in records of equal size.
Record Number	A file may contain more than one record. The record number identifies one record in the file. Files have record number from 1 to the maximum number of records of the file. For Cyclic files, record number 1 is the most recent record appended to the file.
RSA	Prevalent asymmetric cryptographic algorithm, based on factorization of large prime numbers, as defined in ISO/IEC 14888-2:2008 standard; also known as “PKCS #1” (IETF RFC 3447 v2.2).
RFU	<i>Reserved for Future Use.</i>
ROM	<i>Read-Only Memory.</i>
SAM	<i>Secure Application Module:</i> secure microprocessor smart card.
SCP	<i>Secure Channel Protocol.</i>
SD	<i>Security Domain.</i>
SE	<i>Secure Element:</i> secure microprocessor able to store and operate software, especially ISO/IEC 7816-4 applications.
SFI	<i>Short File Identifier.</i>
Signature	Result of a cryptographic computation added to some digital information in order to authenticate it.
SIM	<i>Subscriber Identity Module.</i>
SHA-256	Standard hash algorithm (FIPS PUB 180-4) providing a 256-byte hash.

Short File Identifier	(SFI) External unique number identifying an EF (1 to 30). An EF may have no SFI (indicated by the value 0).
SP	<i>Service provider: the transport network.</i>
SSD	<i>Supplementary Security Domain.</i>
SW	<i>Status Word.</i>
SWP	<i>Single-Wire Protocol.</i>
T2	Hoplink is the interoperable ticketing application developed by CNA. Some figures and names of files and fields may use this acronym of Triangle 2, the former denomination of Hoplink.
TDES	Symmetric cryptographic algorithm is made of three successive DES operations (as defined in ISO/IEC 18033-3), also called "Triple-DES", or "3DES". In Calypso a TDES key is made of two DES keys, making it a much stronger algorithm than simple DES, and of equivalent strength to DESX.
TID	Identifier of the Hoplink Partner. This unique number identifies the Hoplink partner. It is delivered by CNA.
TLV	<i>Tag Length Value.</i>
TTP	<i>Trusted Third-Party.</i>
TPDU	<i>Transmission Protocol Data Unit (as defined in ISO/IEC 7816-3).</i>