



SWIFTNet Messaging Services

SWIFTNet RMA Service

RMA Planning Guide

This planning guide explains how SWIFT intends to introduce the Relationship Management Application (RMA) during the SWIFTNet Phase 2 migration. All SWIFTNet FIN users must read this document. The document assumes that readers are familiar with the SWIFTNet Phase 2 project, and have either attended a roadshow, or have read and understood the *SWIFTNet Phase 2 Planning Guide*.

7 March 2008



Legal Notices

Copyright

Copyright © S.W.I.F.T. SCRL ("SWIFT"), Avenue Adèle 1, B-1310 La Hulpe, Belgium, or its licensors, 2008. All rights reserved.

You may copy this publication within your organisation. Any such copy must include these legal notices.

Confidentiality

This publication may contain SWIFT or third-party confidential information. Do not disclose this publication outside your organisation without the prior written consent of SWIFT.

Disclaimer

SWIFT supplies this publication for information purposes only. The information in this publication may change from time to time. You must always refer to the latest available version.

Translations

The English version of SWIFT documentation is the only official version.

Trademarks

SWIFT, S.W.I.F.T., the SWIFT logo, Sibos, SWIFTNet, SWIFTAlliance, SWIFTStandards, SWIFTReady, and Accord are trademarks of S.W.I.F.T. SCRL. Other SWIFT-derived service and product names, including SWIFTSolutions, SWIFTWatch, and SWIFTSupport are tradenames of S.W.I.F.T. SCRL. SWIFT is the trading name of S.W.I.F.T. SCRL. Other product or company names in this publication are tradenames, trademarks, or registered trademarks of their respective owners.

Preface

About this document

This planning guide explains how SWIFT intends to introduce the Relationship Management Application (RMA) during the SWIFTNet Phase 2 migration.

Scope

The key content areas of this document are as follows:

- BKE-to-RMA migration
- vendor-neutral management of relationships with RMA
- available RMA options and technical detail
- operational advice for RMA users
- requirements for testing before going live
- requirements for training before going live

Audience

SWIFT intends this document for the following audience:

- people within a customer's organisation that plan, design, and implement the migration from BKE to RMA
- people who operate RMA
- managers of RMA users

Assumptions

It is assumed that the reader understands the concepts of the SWIFTNet Phase 2 migration up to the C1 milestone, these are:

- the concept of (dual-) signing SWIFTNet FIN messages with PKI signatures and Message Authentication Code (MAC) or Proprietary Authentication Code (PAC) trailers
- the use of certificates on Hardware Security Module (HSM)
- how to log on with SWIFTNet FIN Phase 2 (protocol version 3)
- general SWIFTNet concepts such as Distinguished Names (DNs), Security Officers (SOs), and Role-Base Access Control (RBAC)

For more information about these concepts, see the *SWIFTNet Phase 2 Planning Guide* and the *SWIFTNet Phase 2 Detailed Overview*.

This *RMA Planning Guide* supersedes the following information in these documents:

- the sections in the *SWIFTNet Phase 2 Planning Guide* that explain Stage 5 (RMA testing and live use) and Stage 6 (RMA cutover)
- the sections in the *SWIFTNet Phase 2 Detailed Overview* that relate to RMA and BKE-to-RMA migration.

Changes since previous document version (20 August 2007)

This version of the RMA Planning Guide introduces changes in the section on Pricing.

Related documentation

- *SWIFTNet Phase 2 Planning Guide*
- *SWIFTNet Phase 2 Detailed Overview*
- *SWIFTNet Phase 2 Wallchart Planner*
- *SWIFTNet Service Description*
- *SWIFTNet Naming and Addressing Guide*

Note The SWIFTNet Phase 2-specific documentation is available at www.swift.com > Ordering & Support > SWIFTNet Phase 2.

Table of Contents

1	Relationship Management Application Overview	7
1.1	RMA Concepts	7
1.2	RMA for SWIFTNet FIN	13
1.3	Benefits of RMA	16
2	Managing Authorisations	17
2.1	Establishing a New Relationship	17
2.1.1	Issuing Authorisations	18
2.1.2	Accepting Authorisations	21
2.1.3	Rejecting Authorisations	21
2.2	RMA Queries and Answers	22
2.3	Terminating a Relationship	23
2.3.1	Revoking Authorisations	23
2.3.2	Deleting Authorisations	25
2.4	Changing an Existing Relationship	26
2.5	Importing and Exporting RMA Authorisations	29
2.6	Rules for Using RMA	31
2.7	Primary Operational Differences between RMA and BKE	31
3	RMA Technology	34
3.1	RMA Use of InterAct Store-and-Forward	34
3.2	Certificates, and RBAC Roles for RMA	38
3.3	Impact of New Message Standards	40
4	Testing RMA	42
4.1	SWIFTNet FIN Test and Training	42
4.2	RMA Sparring Partner	43
4.3	Synonym Testing	44
4.4	Testing with the Tankfile	44
4.5	Mandatory Test Scenarios	45
5	BKE-to-RMA Migration	46
5.1	Principles of the Migration	46
5.2	Timeline and Customer Milestones	47
5.3	Migration Activities	48
5.4	The RMA Recording and Bootstrapping Process	51
6	Pricing	58
	Appendix A RMA Distribution File Format	60
A.1	Overall File Format	60
A.2	RMAFileHeader	60
A.3	RMARecord	61

Appendix B RMA Reject Reasons63

 B.1 Reject Codes and Definitions 63

Appendix C Background Information for the Reader64

 C.1 Secure Login and Select 64

 C.2 Bilateral Key Exchange (BKE) 64

 C.3 SWIFTNet Naming and Addressing 65

 C.4 SWIFTNet FIN Addressing 66

 C.5 SWIFTNet PKI 66

1 Relationship Management Application Overview

BKE to manage relationships with SWIFTNet Phase 1

With SWIFTNet Phase 1, the Bilateral Key Exchange (BKE) security mechanism serves two purposes for SWIFTNet FIN users:

- BKE provides end-to-end authentication (proof of sender's identity).
- BKE allows the management of relationships with correspondents. This is because authenticated FIN messages can only be exchanged if a valid bilateral key is established with a correspondent.

Relationship Management with SWIFTNet Phase 2

With SWIFTNet Phase 2:

- SWIFTNet PKI digital signatures replace the end-to-end authentication aspect of BKE.
- The relationship management aspect of BKE is replaced by a new, more powerful way to manage business relationships, the Relationship Management Application (RMA).

When the migration to SWIFTNet Phase 2 is completed, all SWIFTNet FIN users must use RMA instead of BKE to manage relationships.

SWIFT has designed RMA to be easier to use than BKE. RMA offers more control than BKE, including:

- *who* can send the user messages
- *what* they can send (which Message Types [MTs])
- *when* they can send messages

The overall objective of RMA is to stop unwanted traffic at the sender. This has many benefits: by not receiving unwanted traffic, users save time and effort in treating this traffic, and are less exposed to the risks of wrongly processing such unwanted traffic. This helps protect the users against audit and regulatory compliance risks, and helps to avoid fines and damage to reputation.

In addition, SWIFT has designed RMA to extend, at a later date, to SWIFTNet services other than SWIFTNet FIN. FIN is the first service to use RMA.

The following sections provide an overview of how RMA works in general, and how it applies to SWIFTNet FIN.

1.1 RMA Concepts

The Relationship Management Application

For both FIN and SWIFTNet, it is the BIC8 that identifies an institution (for example, BANKBEBB). The BIC8 identifies the name of the institution and its legal country of residence. Traffic that an institution sends and receives over SWIFTNet FIN, SWIFTNet InterAct, and SWIFTNet FileAct carries the institution's BIC8.

The Relationship Management Application (RMA) enables the user to control which institutions (BIC8s) can send traffic to them. RMA also enables the user to control the type of traffic that these institutions can send. On FIN, this means that RMA enables the user to control which FIN Message Types (MTs) its correspondents can send them.

RMA enables the user to issue *authorisations* to correspondents that the user wants a business relationship with. Only correspondents that have accepted an authorisation from the user can send that user traffic. Correspondents may only send the type of traffic that the user has allowed in the authorisation. Similarly, the user can only send traffic to correspondents that have issued the user with an **authorisation-to-send**.

Users cannot grant themselves an **authorisation-to-send** to a correspondent. Therefore, RMA stops unwanted traffic at the sender. A sender must have an **authorisation-to-send**, issued by a correspondent, before they can send traffic to that correspondent. Therefore, receivers of traffic are in complete control of who can send them traffic, and what type of traffic.

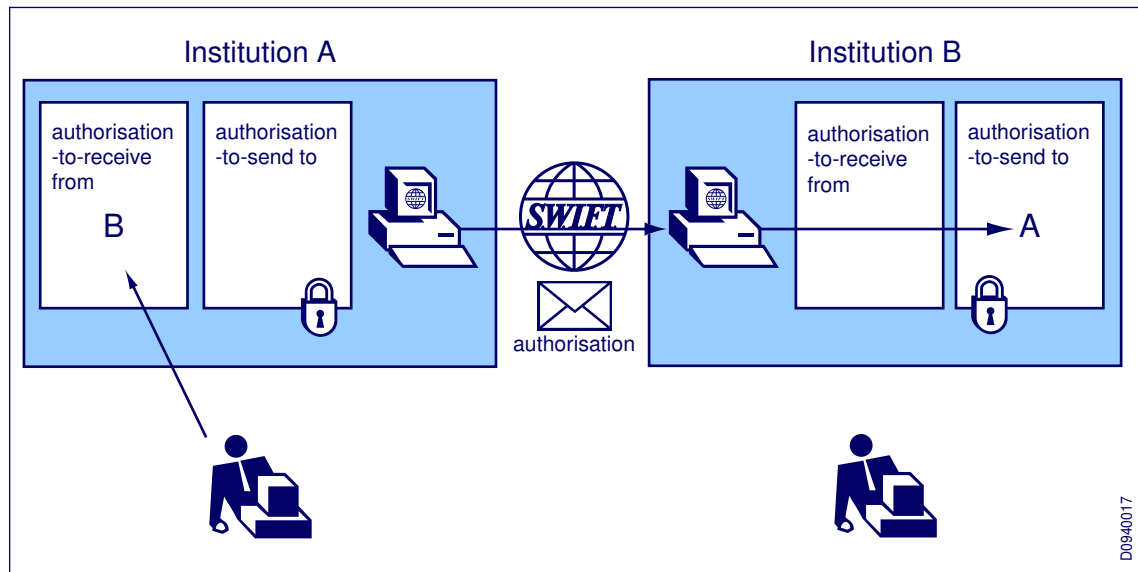
RMA authorisations

Each authorisation results in two *RMA records*, one record at the issuer, and one record at the correspondent. The RMA records are stored in the issuer's and the correspondent's local environment. This resembles the way in which Bilateral Key Exchange (BKE) keys are stored in the user's environment for direct management by the user.

When a user issues an authorisation to its correspondent, the correspondent stores the authorisation as an **authorisation-to-send** (to the issuing user). The user (that has issued the authorisation) stores an **authorisation-to-receive** (from that correspondent).

The following illustration shows this concept. When institution A issues an authorisation to institution B, it sends an authorisation message to B. At the same time it stores an RMA **authorisation-to-receive-from-B record**. When institution B receives and accepts the authorisation message, it then stores an RMA **authorisation-to-send-to-A record**.

Institution A issues an authorisation to institution B



RMA **authorisation-to-send** records enable the user to keep track of the correspondents that have authorised them to send traffic to. RMA **authorisation-to-receive** records enable the user to keep track of the correspondents to whom they have issued an authorisation: these are the correspondents that the user is willing to receive traffic from.

Authorisations are XML messages that customers exchange over SWIFTNet by means of InterAct Store-and-Forward messaging. Authorisations-to-send and authorisations-to-receive are RMA records that the user stores in its local environment. The padlocks on the **authorisation-to-send** in the illustration symbolise that an institution cannot grant itself an **authorisation-to-send**.

For more information about what users exchange in authorisations and store in RMA records, see "Managing Authorisations" on page 17.

Uni-directional and bi-directional relationships

In principle, RMA authorisations are always uni-directional (that is, one user authorises another to send traffic). To establish a bi-directional traffic flow, each of the parties to the traffic issues the other with an RMA **authorisation-to-send**.

Controlling what can be sent

An authorisation is always for a given type of traffic (a SWIFTNet service). For example, a user authorises a correspondent to send them SWIFTNet FIN messages. Once RMA is extended beyond SWIFTNet FIN, there will be different authorisations required for each SWIFTNet service for which a user authorises its correspondents to send traffic to them.

Within a service, an authorisation can more narrowly limit the type of traffic that can be sent. Specifically, on SWIFTNet FIN, an authorisation can limit the correspondent to send only certain Message Types (MTs). These are called *granular* authorisations, since they specify with additional granularity which traffic is allowed. For more information about granularity for SWIFTNet FIN, see "RMA for SWIFTNet FIN" on page 13.

It is always the issuer of the authorisation that specifies the type of traffic that is allowed. The correspondent cannot modify this. Therefore, the receiver of the traffic is in full control of what it lets its correspondents send to them.

Granular authorisations allow the fine-tuning of authorisations to better reflect the true nature of their business that is conducted with a correspondent. For example, if a user only does payments with a correspondent, then they can let this correspondent send them payments messages, and restrict this correspondent from sending other types of traffic, like securities, or Documentary Credits, for example. Therefore, unwanted traffic is stopped at the sender, and this has many benefits for compliance and risk control.

Operations on authorisations

The issuer can **issue** authorisations and can, at any time, **revoke** authorisations. The correspondent that receives an **authorisation-to-send** can **accept** or **reject** the authorisation. If a correspondent rejects an authorisation, then the issuer is notified. A correspondent that has accepted an authorisation can decide to **delete** the authorisation at a later stage. The issuer is notified if a correspondent deletes an authorisation.

A user can grant an **authorisation-to-send** to its correspondent, so that the correspondent can send them traffic. However, a user cannot grant themselves an **authorisation-to-send** that lets them send traffic to the correspondent. Only the correspondent can grant that user the right to send traffic. This is why a lock is shown on the **authorisation-to-send** records in the previous illustration.

To modify an authorisation (for example, to modify the type of allowed traffic within a service), the user must issue a new authorisation to its correspondent. The new authorisation replaces the previous one.

For more information about the operations on authorisations, see "Managing Authorisations" on page 17.

The two parts of RMA

The RMA consists of two parts:

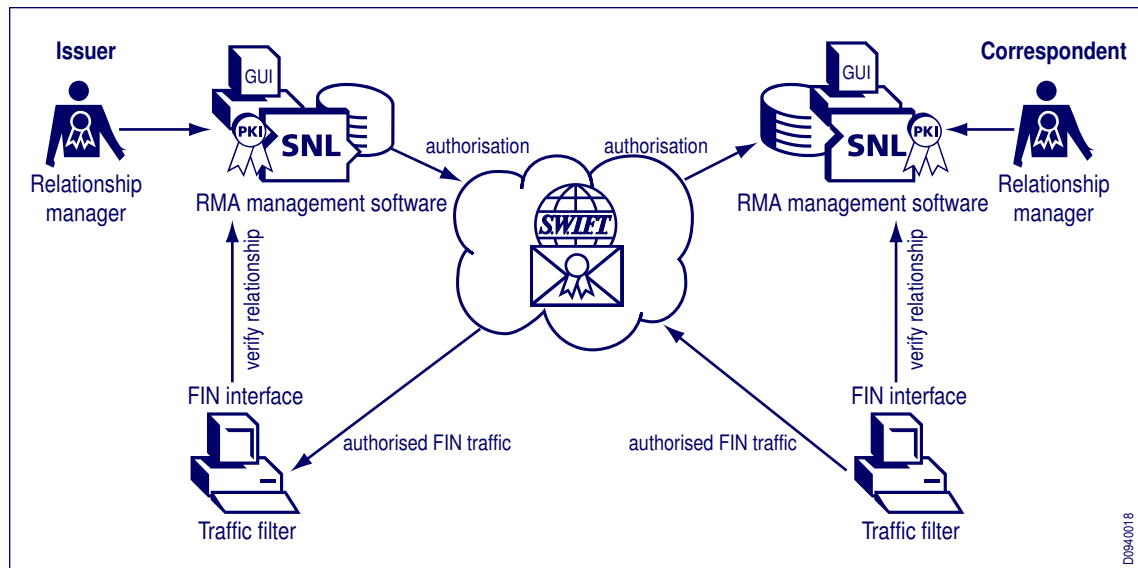
- **RMA management**

RMA management is the term for the way in which users exchange authorisations with correspondents over SWIFTNet, and securely store the resulting authorisations. *RMA management software* is the software that each user operates in its local environment, to perform the exchanges and store the resulting authorisation records in the user's local RMA data store. To perform the exchanges of authorisations over SWIFTNet InterAct Store-and-Forward, the RMA management software must have access to SWIFTNet, through a SWIFTNet Link (SNL). The *RMA manager* is the person that operates the RMA management software.

- **RMA traffic filtering**

Applications (for example, SWIFTNet FIN interfaces) can make decisions to authorise traffic, based on the authorisations that the RMA management software has stored. The function of decision making about traffic is called *RMA traffic filtering*. Applications that perform traffic filtering are sometimes called *RMA subscriber applications*. This guide, however, refers to traffic-filtering applications as *interfaces* (for example, a SWIFTNet FIN interface).

RMA management and RMA traffic filtering

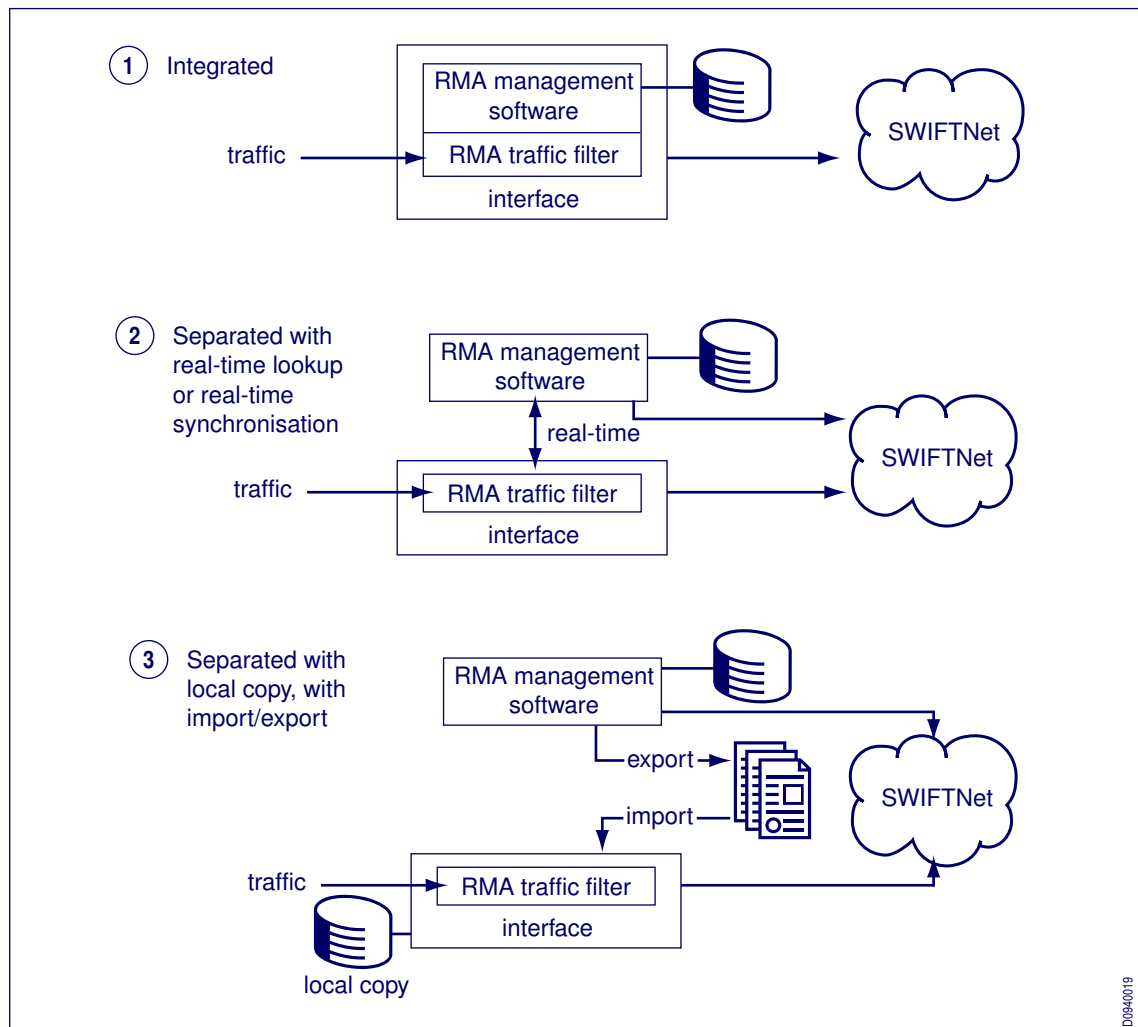


Traffic filtering occurs at both the sender's location and the receiver's location, as follows:

- When the sending interface sends traffic to a correspondent, the interface must check that the traffic is authorised (that is, that there is a matching **authorisation-to-send**). If there is no matching **authorisation-to-send**, then the interface does not send the traffic.
- When the receiving interface receives traffic from a correspondent, the interface must check that the traffic is authorised (that is, whether there is a matching **authorisation-to-receive**). If there is no **authorisation-to-receive**, then the interface flags the traffic as non-authorised (for example, puts it in an investigation queue). Filtering of received traffic offers a double assurance: normally, RMA stops unwanted traffic at the sender's location.

How to combine or segregate RMA management and RMA traffic filtering

There are three ways to combine or segregate RMA management and RMA traffic filtering.



- **Integrated**

The user can integrate RMA management and RMA traffic filtering interface into one application (as in SWIFTAlliance Access and SWIFTAlliance Entry). Integration is the simplest possible setup.

- **Separated with real-time lookup or synchronisation**

If the user does not integrate the RMA management and traffic-filtering functions, then the interface that performs the traffic filtering must consult the authorisations in the relationship management data store. To do this, the interface uses real-time lookup or real-time synchronisation. real-time lookup means that the interface consults the RMA management software's master RMA data store for every message that it sends or receives. real-time synchronisation means that the interface keeps its own local copy of (a subset of) the RMA data store, which is kept synchronised in real time with the master RMA data store of the RMA management software.

- **Separated with local copy, with import and export**

The interface can work on its own copy of (a subset of) the RMA records. To allow the interface to work on its own copy, the user must distribute the RMA authorisations. In this process, the

user exports the authorisations from the relationship management data store, and imports them into the interface. SWIFT has specified a standard file format for this purpose, that allows the user to distribute (export and import) authorisations between software developed by different vendors.

For more information about the standard RMA distribution file format, see "Overall File Format" on page 60. All SWIFTNet FIN interfaces and RMA management software support this file format.

No old, current, or future authorisations

For a given service (for example, SWIFTNet FIN), there can be only one valid **authorisation-to-send** and one valid **authorisation-to-receive** per correspondent at a given time. Unlike Bilateral Key Exchange (BKE), there is no notion of old, current, or future keys for authorisations.

If the user issues a new authorisation for a given correspondent and service, then it immediately replaces any existing authorisation. RMA thus immediately authorises traffic against the new authorisation.

No renewal required for authorisations

Unlike Bilateral Key Exchange (BKE) keys, authorisations do not carry a shared secret that is vulnerable to disclosure over time. For this reason, there are no technical or security reasons to renew authorisations periodically.

Authorisations reflect business decisions. Once an RMA user grants an authorisation, it remains valid until the business relationship ends or changes. It is good practice to review relationships on a regular basis, however, this is the user's own decision. There is no need to re-send an authorisation if there is no change to the relationship.

Temporary relationships

To plan the start or end (or both) of a business agreement, the user can specify a *start date* or an *end date* (or both) on an authorisation. By default, an authorisation does not have a start date or an end date. The user can specify a start date without an end date, and the reverse.

Warning If you specify a start date in an authorisation, then RMA does not authorise traffic until that start date.

The user can use end dates to establish a new temporary relationship, or to plan in advance the end of an existing relationship. The user can use start dates to plan the start of a new relationship. However, SWIFT does not advise the use of start dates to plan a change to an existing relationship. For more information about changing existing relationships, see "Changing an Existing Relationship" on page 26

Start dates and end dates are inclusive, which has the following interpretation:

- traffic that is sent on the start date or end date is authorised
 - traffic that is sent before the start date or after the end date is not authorised
 - traffic that is sent between the start and end dates is authorised
-

Important Because start dates and end dates must have the same meaning to the issuer and its correspondents regardless of time zone, these dates are expressed in Coordinated Universal Time (UTC). UTC is for this purpose equivalent to Greenwich Mean Time (GMT).

RMA traffic filtering compares the transmission date, of the traffic, with the start date or end date in the RMA record. Thus, even when the traffic stays queued for several days on SWIFTNet (for

example, over the weekend), the sender and receiver of the traffic apply the same rule. This is illustrated in the following example.

For example, an authorisation has an end date of Sunday 31st May 2009 (UTC). FIN messages sent on or before that day are authorised to be sent. Assuming that the receiver did not log on to FIN, and some of these messages are queued on SWIFTNet FIN. On Monday 1st June, after the end date of the authorisation, the receiver logs on and receives these queued messages. The receiver can see that these messages were sent before or on the end date of the authorisation, since all messages carry a *send date×tamp* (in UTC) in the SWIFTNet header. Messages that were sent before UTC midnight end-of-day Sunday 31st May will be authorised to receive.

Centralising RMA

As with BKE, you can exchange RMA authorisations for multiple BICs from one single system or location. You can perform the actual traffic filtering for these BICs in other (potentially remote) systems or locations.

For more information about how to centralise RMA, see "RMA Technology" on page 34.

RMA queries and answers

Institutions can use RMA to exchange short text messages. These messages are called RMA queries and answers. The query and answer mechanism routes the short messages to the RMA management software of the correspondent. Individuals that manage relationships at the receiving institution can then read these messages, and reply to them. The advantage of this mechanism over e-mails (or MT 999s) is that your message reaches the appropriate personnel within an institution that deal with correspondent relationships. Thus, talking about a relationship with a correspondent, even with correspondents with whom you have no relationship yet, is greatly simplified: there is no need to know a name, telephone number, or e-mail address of any person at the correspondent. Another advantage of the RMA query and answer mechanism is that a trace can be kept of the entire conversation, stored together with the authorisation.

1.2 RMA for SWIFTNet FIN

Overview

SWIFT, as the service administrator for SWIFTNet FIN, has decided to mandate RMA for SWIFTNet FIN. This means that, as of a certain date, all SWIFTNet FIN interfaces must apply RMA traffic filtering to all SWIFTNet FIN traffic. SWIFT has published the specifications of the RMA application to the SWIFTNet FIN interface vendors to ensure the compliance of SWIFTNet FIN interfaces. Compliance means that vendors must include RMA in the FIN interface qualification programme.

All SWIFTNet FIN users must migrate from BKE to RMA by a certain date. SWIFT foresees a transition period during which both mechanisms co-exist. For more information about the BKE-to-RMA migration, see "BKE-to-RMA Migration" on page 46.

Authenticated SWIFTNet FIN messages only

RMA filtering applies only to SWIFTNet FIN messages that require authentication. The list of all SWIFTNet FIN Message Types (MTs) that require authentication is available in the *SWIFT User Handbook, Standards General Information*. An example of an authenticated MT is the MT 103. If users want to exchange any unauthenticated SWIFTNet FIN messages with a counterparty, then they can do so without an authorisation. This was also possible without a BKE key. An example of an unauthenticated SWIFTNet FIN MT is the MT 999.

Granularity: control of which MTs can be sent

For authorisations related to SWIFTNet FIN, a user can optionally specify which authenticated MTs a correspondent can either send or not send to them. This list of MTs is called the *permission list*. If no permission list is specified (the default), then all FIN messages are authorised. More specifically, you can include (permit) or exclude (not permit) an entire SWIFTNet FIN message category, or include or exclude specific MTs within a message category.

All RMA management software must be able to receive and store such granular authorisations-to-send, though the ability to issue such granular authorisations is optional. All SWIFTNet FIN interfaces must be able to filter traffic based on granular authorisations. In other words, correspondents can always issue granular authorisations to the user, to stop the user from sending certain MTs to them. This is applicable even if the user decides to never issue granular authorisations to the correspondents.

Within a permission list, you can only specify SWIFTNet FIN messages that require authentication. SWIFTNet FIN messages that do not require authentication (for example, MT 999) are always authorised, therefore, will never appear in a permission list.

Message categories

SWIFTNet FIN has nine message categories:

- Category 1: Customer Payments and Cheques
- Category 2: Financial Institution Transfers
- Category 3: Foreign Exchange, Money Markets, and Derivatives
- Category 4: Collection and Cash Letters
- Category 5: Securities Markets
- Category 6: Precious Metals, Syndications
- Category 7: Documentary Credits and Guarantees
- Category 8: Travellers Cheques
- Category 9: Cash Management and Customer Status

All categories contain authenticated and non-authenticated MTs, except Category 9. Category 9 messages are all non-authenticated. Messages in Category 9 are thus always authorised, and Category 9 cannot be specified in the permission list of an authorisation.

Permission lists

If a permission list is present in an authorisation, then message categories that are not explicitly included (permitted) in the list, are by default excluded (not permitted).

Within a message category, the following three options exist:

- either all MTs of the category are included, or
- a list of specific 3-digit MTs within that category is included (and the other MTs in the category are excluded), or
- a list of specific 3-digit MTs within that category is excluded (and the other MTs in the category are included)

Example 1:

- include all messages in Category 1
- include all messages in Category 2
- include all messages in Category 3, but exclude MT 303 and MT 350

This permission list permits all messages of Category 1 (MT 102, MT 103, and so on), and all messages of Category 2 (MT 202, and so on), and all messages in Category 3, except MT 303 and MT 350. All other authenticated messages are excluded.

Example 2:

- exclude all messages in Category 7, but include MT 799
- exclude all messages in Category 1, but include MT 102
- include message Category 2, but exclude MT 202

This permission list permits MT 799, MT 102, and all MT 2xx messages except MT202. All other authenticated messages are excluded.

RMA and Closed User Groups

RMA serves a different need than Closed User Groups (CUGs) or Message User Groups (MUGs). RMA enables individual users inside a CUG or a MUG to determine the parties with whom they want to do business. CUGs remain important for access control to specific services (for example, who can use a particular SWIFTNet FInCopy service). MUGs are necessary to govern who has access to specific Message Types (MTs).

RMA and SWIFTNet FIN Test and Training

RMA filtering is optional for SWIFTNet FIN Test and Training (T&T) . A SWIFTNet FIN interface has a configuration setting that can turn the RMA filtering for T&T SWIFTNet FIN messages on or off. The sender and receiver BIC8 of a T&T SWIFTNet FIN message has a zero in the eighth character of the BIC8 (for example, BANKBEB0).

RMA and SWIFTNet FInCopy

A SWIFTNet FInCopy service administrator (the central institution) can decide to bypass RMA within the SWIFTNet FInCopy service that they administer.

When the SWIFTNet FIN interface sends or receives a message within such a SWIFTNet FInCopy service, the message bypasses the check for the existence of a valid RMA authorisation. However, the Hardware Security Module (HSM) still signs the message with the Public Key Infrastructure (PKI) key. In this way, SWIFTNet FIN delivers proof of sender identity for SWIFTNet FInCopy messages, both to the receiver and to the central institution. Members of such a SWIFTNet FInCopy service that exchange SWIFTNet FIN messages with each other outside the SWIFTNet FInCopy service must use RMA for those messages.

No wildcard authorisations

BKE keys apply to BIC8s, but users can group them at BIC6 or BIC4 level, which has led to criticism of BKE's complexity and lack of granularity. The ability to group BKEs at BIC6 or BIC4 levels opens the door to traffic exchange with many more institutions than initially intended. As a result, RMA authorisations do not support the concept of wildcards at BIC6 or BIC4 level. RMA authorisations for SWIFTNet FIN traffic will **only be exchanged between BIC8s**.

1.3 Benefits of RMA

RMA stops more unwanted traffic than BKE

Most current SWIFTNet FIN interfaces that use Bilateral Key Exchange (BKE) enable the user to enter a manual key. The ability to enter a manual key means that the receiver may receive unwanted messages. Users that receive an **authorisation-to-send**, cannot add, or modify the authorisation. If an institution cannot add or modify its own **authorisation-to-send**, then it cannot grant itself the right to send unwanted traffic to another institution. There is no possibility to bypass an **authorisation-to-send**. This system stops unwanted traffic at the sender.

RMA is extensible to other SWIFTNet services beyond SWIFTNet FIN

RMA is a generic SWIFTNet feature that SWIFT has designed to give users the necessary controls over other institutions that can send them traffic over SWIFTNet. SWIFT intends to apply RMA to SWIFTNet FIN and to other SWIFTNet services that require similar levels of control over correspondent traffic.

RMA offers more control over correspondent relationships

Users can determine which correspondents to receive traffic from, and the type of traffic to accept. SWIFTNet FIN users have the option to control correspondents up to the level of individual authenticated Message Types (MTs).

RMA is easier to use than BKE

At the outset, RMA users are not required to agree to a number of technical parameters with each correspondent (that is, there is no technical pre-agreement). Unlike the 4-message exchange in BKE (MT 960 to MT 963), a single SWIFTNet InterAct Store-and-Forward message is sufficient to establish an RMA authorisation. Thus, when exchanging an RMA authorisation it is not necessary to keep a *context* open for several days, until the exchange is completed. For each correspondent, at any given time, and for a given service, there can be only one valid **authorisation-to-send** and one valid **authorisation-to-receive**. RMA does not use the concept of old, current, and future authorisations. An authorisation without an end date does not expire. With no expiry, it is not necessary to exchange authorisations every 6 to 12 months with all correspondents.

RMA is more granular than BKE

Bilateral keys that users exchange with BKE, apply to all authenticated messages. However, RMA lets users define authorisations at the level of message category or message type. RMA enables the user to reflect the true nature of the business that it conducts with a correspondent.

RMA reduces risk and operational cost

RMA stops unwanted traffic at the sender. This shields the receiver from both the unwanted traffic and its potential negative consequences. Negative consequences may include investigations, audits, regulatory reporting, and risk to reputation.

2 Managing Authorisations

2.1 Establishing a New Relationship

Overview

To establish a new relationship, one party issues an authorisation to its correspondent. The correspondent can either accept or reject the authorisation. This section describes the actions that a user must do to establish a new relationship.

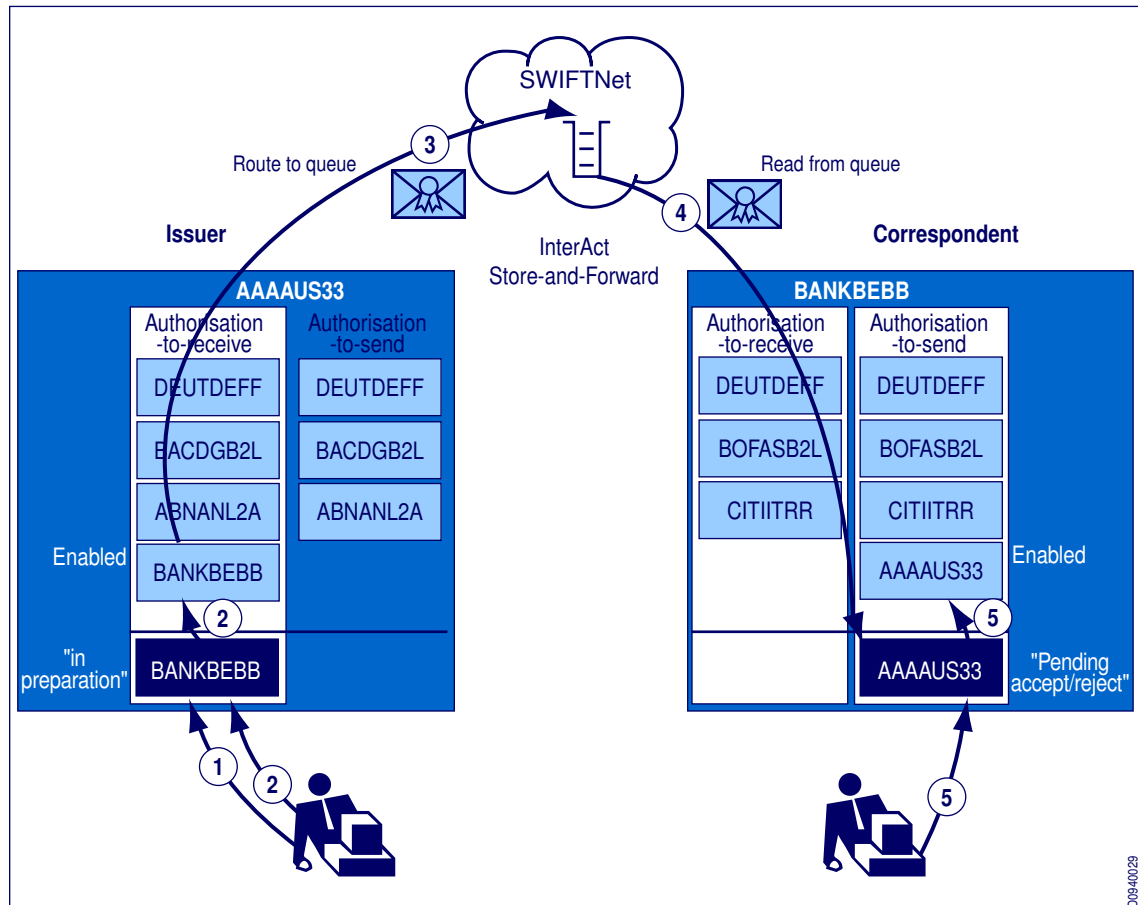
Note Users do not have to issue authorisations to convert existing Bilateral Key Exchange (BKE) key relationships to Relationship Management Application (RMA) authorisations. For more information about BKE-to-RMA migration, see "BKE-to-RMA Migration" on page 46.

2.1.1 Issuing Authorisations

Establishing a new relationship

When the relationship manager at an institution (the issuer) decides to issue an authorisation to another institution (the correspondent), the relationship manager instructs the RMA management software to issue an authorisation. RMA stores the authorisation immediately at the issuer's location as an enabled **authorisation-to-receive**. RMA sends the authorisation over SWIFTNet to the correspondent.

Authorisation issue and acceptance process



- **Step 1:** AAAAUS33 prepares an authorisation for BANKBEBB (draft). During this preparation stage, traffic is still unauthorised.
- **Step 2:** AAAAUS33 decides to issue the authorisation to BANKBEBB (this may require 4- eyes approval). RMA immediately stores the **authorisation-to-prepare** from BANKBEBB as **enabled** in AAAAUS33's RMA data store. RMA then sends the authorisation message to BANKBEBB.
- **Step 3:** RMA sends the authorisation message to BANKBEBB over SWIFTNet. More specifically, RMA routes the message to an InterAct Store-and-Forward queue. The benefit of the store-and-forward queue is that BANKBEBB does not have to be online when AAAAUS33 sends the authorisation.
- **Step 4:** BANKBEBB receives the authorisation message from SWIFTNet. More precisely, BANKBEBB reads the message from the InterAct Store-and-Forward queue. BANKBEBB

stores the **authorisation-to-send** until the appropriate personnel decide whether to accept or reject it. At this point, BANKBEBB does not have authorisation to send traffic to AAAAUS33. According to the RMA rules (see "Rules for Using RMA" on page 31), BANKBEBB must accept or reject the **authorisation-to-send** within six business days at the latest.

- **Step 5:** BANKBEBB accepts the authorisation (this may require 4-eyes approval). RMA stores the **authorisation-to-send** to AAAAUS33 in BANKBEBB's data store, as enabled. From this moment, BANKBEBB has authorisation to send traffic to AAAAUS33.

Note BANKBEBB does not return an acknowledgement of acceptance to AAAAUS33.

Issuer and Correspondent

All RMA authorisations that a user issues must identify the following BIC8s:

- **The issuer BIC8**

The issuer BIC8 issues the authorisation (RMA does not allow wildcards in the issuer BIC8).

- **The correspondent BIC8**

The issuer grants the **authorisation-to-send** to the correspondent BIC8 (RMA does not allow wildcards in the correspondent BIC8).

Note: If RMA management software allows a user to specify multiple BIC8s in one go when issuing an authorisation, then this means that it will send multiple authorisation messages.

Live RMA

An RMA authorisation message that grants the permission to use a specific SWIFTNet service to send traffic must quote that service name in the message. For example, each RMA message that grants the permission to send live FIN traffic must quote the service name for live SWIFTNet FIN (that is, `swift.fin`).

Test and Training RMA

A Test and Training (T&T) RMA message is sent on the T&T RMA SWIFTNet service (`swift.rma!p`). A live RMA message is sent on the live RMA SWIFTNet service (`swift.rma`). A T&T RMA message uses a T&T BIC8 for both the issuer and the correspondent. A T&T BIC8 has a zero as the eighth character (for example, BANKBEB0).

An RMA authorisation that grants permission for T&T SWIFTNet FIN must identify `swift.fin!p` in the authorisation message as the service name for T&T SWIFTNet FIN.

Note The `swift.fin!p` service does not exist on SWIFTNet. However, RMA uses the `swift.fin!p` service name by convention, inside the authorisation message, to identify T&T FIN.

Immediately effective

An authorisation is immediately *enabled* at the issuer's side at the moment of transmission. RMA does not wait for an acknowledgement or acceptance message from the correspondent. For more information, see the topic on **Error handling** in this section.

Start date and end date

An RMA authorisation can have a start date, an end date, or both. For more information, see "RMA Concepts" on page 7.

Message Type permissions

An RMA authorisation for SWIFTNet FIN can have a list of Message Type (MT) permissions. By default, in absence of a permission list, an RMA authorisation allows all MTs. For more information, see "RMA for SWIFTNet FIN" on page 13.

DateTimeIssued

An RMA message contains the date and time (in Coordinated Universal Time [UTC]) at which the issuer created or changed the authorisation. The RMA management software automatically enters this information, which is close to the actual date and time of transmission of the authorisation message. The correspondent's RMA management software compares the DateTimeIssued information with the DateTimeIssued of previous authorisations from the same issuer. RMA keeps only the most recent authorisation.

If the issuer sends an authorisation, while at the correspondent side an earlier authorisation is still pending acceptance or rejection, then RMA at the correspondent side automatically discards the older authorisation.

No KMA

In the Bilateral Key Management (BKE) process, issuers send BKE messages (MT 96n) to the correspondent's Key Management Authority (KMA) BIC. In that process, the KMA BIC can be different from the correspondent BIC8 that used the key.

The concept of a separate KMA BIC does not exist in RMA. RMA messages that grant permissions to a live BIC8 are always sent to that BIC8. The benefit is that the parties do not have to agree upon a KMA BIC at the start of the relationship.

No features are lost, because RMA users can still receive and process authorisations on a system other than that which uses the authorisations for traffic filtering. With RMA, issuers do not have to know the location at which the receiver receives and processes authorisations.

For more information about how to combine or segregate RMA management from RMA traffic filtering see, page 11. For more information about how to set up RMA management on a system other than the FIN interface, see "RMA Technology" on page 34.

Error handling

If the transmission of the authorisation message to SWIFTNet fails, then the subsequent error handling is vendor-specific. The most likely scenario is that RMA flags the record as having an error. RMA then logs an event and the authorisation returns to its former state (for example, draft or preparation).

If the transmission of the authorisation message to SWIFTNet succeeds, then RMA routes the message to SWIFTNet store-and-forward, which queues it. The SWIFTNet system may return an error if it cannot deliver the message to the correspondent. For example, the system may return an error if the message times out after remaining queued for more than 14 days, or after 10 unsuccessful delivery attempts.

Such errors typically indicate a problem on the correspondent side. In such cases, the authorisation remains in the enabled state at the issuer side because it is uncertain whether the correspondent has processed the authorisation. The RMA management software at the issuer flags such errors to the user.

SWIFT recommends that users that receive error notification contact the correspondent (for example, by means of the RMA query and answer mechanism). Depending on the correspondent's response, the user may decide to re-issue the authorisation.

2.1.2 Accepting Authorisations

Overview

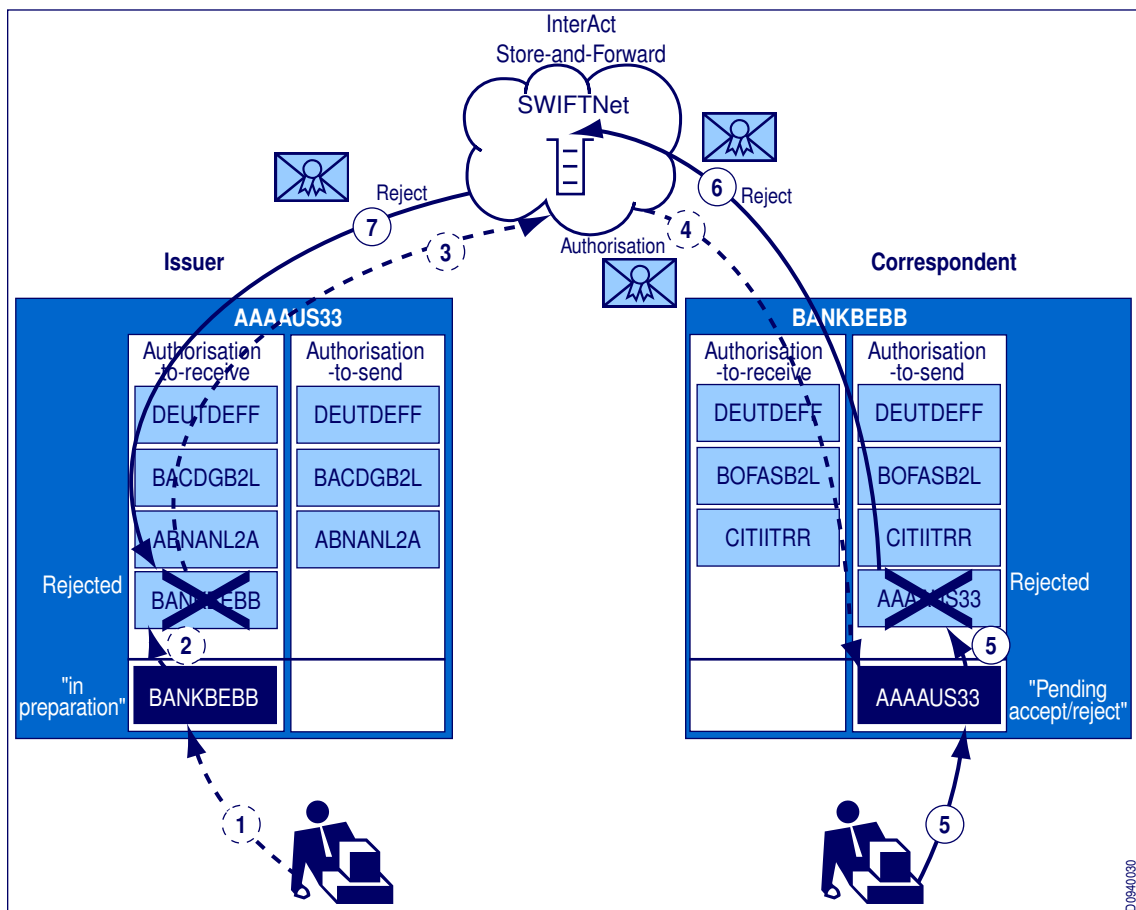
The correspondent that receives the **authorisation-to-send** can either *accept* or *reject* the authorisation. Typically, a human being makes this decision. If the correspondent accepts the authorisation, then RMA stores the authorisation with *enabled* status. From that moment, the correspondent can send traffic, as the authorisation permits, to the issuer.

2.1.3 Rejecting Authorisations

Overview

A correspondent can reject an authorisation message that it does not agree with.

Authorisation rejection process



Note The first four steps (shown as broken lines) are the same as for the issuing of authorisations before acceptance, see "Issuing Authorisations" on page 18.

- **Step 5:** BANKBEBB rejects the authorisation (this may require 4-eyes approval). RMA stores the **authorisation-to-send** in the RMA data store of BANKBEBB as *rejected*. RMA then sends

a reject message to AAAAUS33. From this moment, BANKBEBB is not authorised to send traffic to AAAAUS33.

- **Step 6:** RMA sends the reject message to AAAAUS33 over SWIFTNet store-and-forward. More precisely, RMA routes the reject message to an AAAAUS33 SWIFTNet InterAct Store-and-Forward queue.
- **Step 7:** AAAAUS33 receives the reject message from SWIFTNet. RMA immediately changes the status of the **authorisation-to-receive** from BANKBEBB, which is in the AAAAUS33 RMA data store, to *rejected*.

Important A reject always covers the whole authorisation, including all Message Types (MTs) that the issuer has granted for the service. The correspondent cannot pick and choose, or reject one or more specific MTs within the list of granted MTs.

If the correspondent rejects an authorisation, then RMA creates a record, with status rejected, in the RMA management data store. After rejection, the correspondent's RMA traffic filter does not allow outgoing messages (on the rejected service) to the issuer. Upon rejection, RMA returns an explicit reject message to the issuer. The reject message indicates the reason for the rejection.

Reasons for the rejection of an authorisation includes the following:

- the correspondent does not want to accept **authorisation-to-send** messages
- the authorisation does not match the agreement between the parties

The issuer has no choice but to accept the rejection. After receipt of the rejection, the issuer's traffic filter no longer authorises traffic from the correspondent (on the granted service, and for the granted MTs).

To avoid misinterpretations, the RMA protocol provides standard reject reason codes with standard meanings, see "Reject Codes and Definitions" on page 63. The correspondent can select one of the standard reasons, and can also provide limited free text (maximum 105 characters) as additional explanation in the reject message.

Important A user must always be careful when rejecting an incoming authorisation. Before rejecting it, a user must check whether there is already an existing **authorisation-to-send** to this correspondent. If there is already an **authorisation-to-send**, then this is a change to an existing relationship, see "Changing an Existing Relationship" on page 26.

2.2 RMA Queries and Answers

Free format (query) messages

A user may request RMA-related information from its correspondents by means of a free-format text (query) message (maximum 350 characters). The queried party can answer with another free-format text message (maximum 2000 characters).

The following are possible uses for the RMA query and answer mechanism:

- to seek contact with a new correspondent, to discuss the possibility of establishing a relationship
- to request the other party to issue or re-issue an authorisation
- to enquire why the other party rejected or revoked an authorisation
- to request clarification before the acceptance or the rejection of an authorisation

- to exchange contact details (for example, name, e-mail address, and telephone numbers) for further discussion of RMA matters

2.3 Terminating a Relationship

Overview

At any time, either party can terminate an authorisation (for example, in the case of the business relationship ending), as follows:

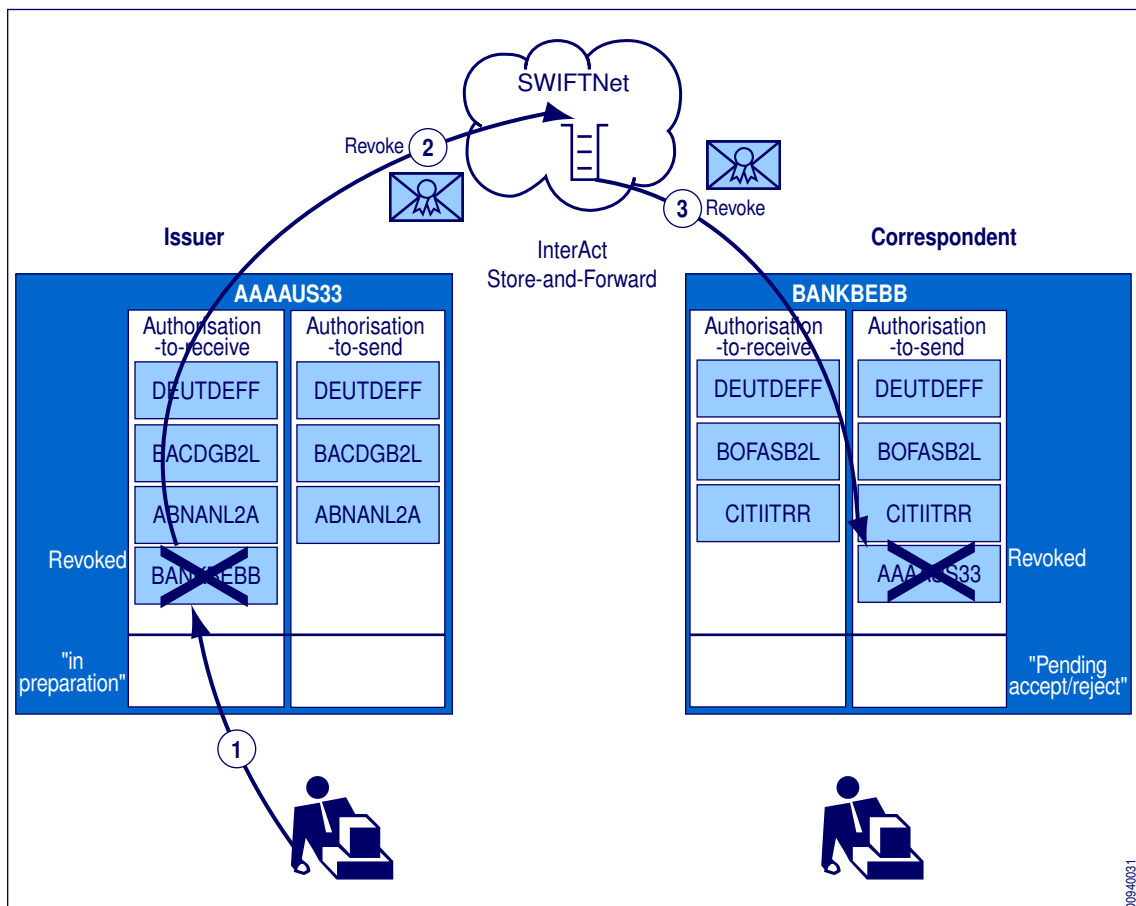
- The party that granted (issued) the authorisation can **revoke** the authorisation.
- The party that accepted the **authorisation-to-send** can **delete** it. A *deletion* resembles a *rejection*, except that it occurs after the party has accepted the authorisation.

2.3.1 Revoking Authorisations

Overview

The revoking party instructs the RMA management software to revoke a previous authorisation. As a result, RMA immediately changes the status of this party's RMA **authorisation-to-receive** record to *revoked*. RMA then sends a revocation message to the correspondent. A revocation applies to a specific service (`swift.fin` or `swift.fin!p`), and all authenticated MTs within that service. Non-authenticated FIN messages (such as MT 999) remain possible.

Authorisation revocation process



Note The situation at the start of the process is that BANKBEBB has accepted a previous authorisation from AAAAUS33. RMA has stored the authorisation on both sides of the relationship with the status *enabled*.

- **Step 1:** AAAAUS33 decides to revoke the authorisation that it previously issued to BANKBEBB. RMA immediately puts the **authorisation-to-receive** into the *revoked* state in AAAAUS33's RMA data store. RMA sends a revoke message to BANKBEBB. From this moment, RMA flags traffic that AAAAUS33 receives from BANKBEBB as unauthorised.
- **Step 2:** RMA sends the revocation message over SWIFTNet to BANKBEBB, which gets routed and stored in a SWIFTNet InterAct Store-and-Forward queue.
- **Step 3:** BANKBEBB receives the revocation message. RMA immediately changes the status of the **authorisation-to-send** to AAAAUS33 that is in BANKBEBB's RMA data store to *revoked*.

The correspondent cannot refuse a revocation message. On receipt of the revocation, RMA immediately changes the status of the correspondent's RMA record for the **authorisation-to-send** to *revoked*. RMA does not return an explicit acknowledgement message to the revoking party. According to the rules, see "Rules for Using RMA" on page 31, the correspondent must act as soon as possible on the revocation. At most, the correspondent must act within one business day. This includes the time to distribute (if applicable) the *revoked* RMA record to the correspondent's traffic-filtering applications. After revocation, the correspondent's RMA traffic filter does not allow the correspondent to send messages to the issuer on the *revoked* service.

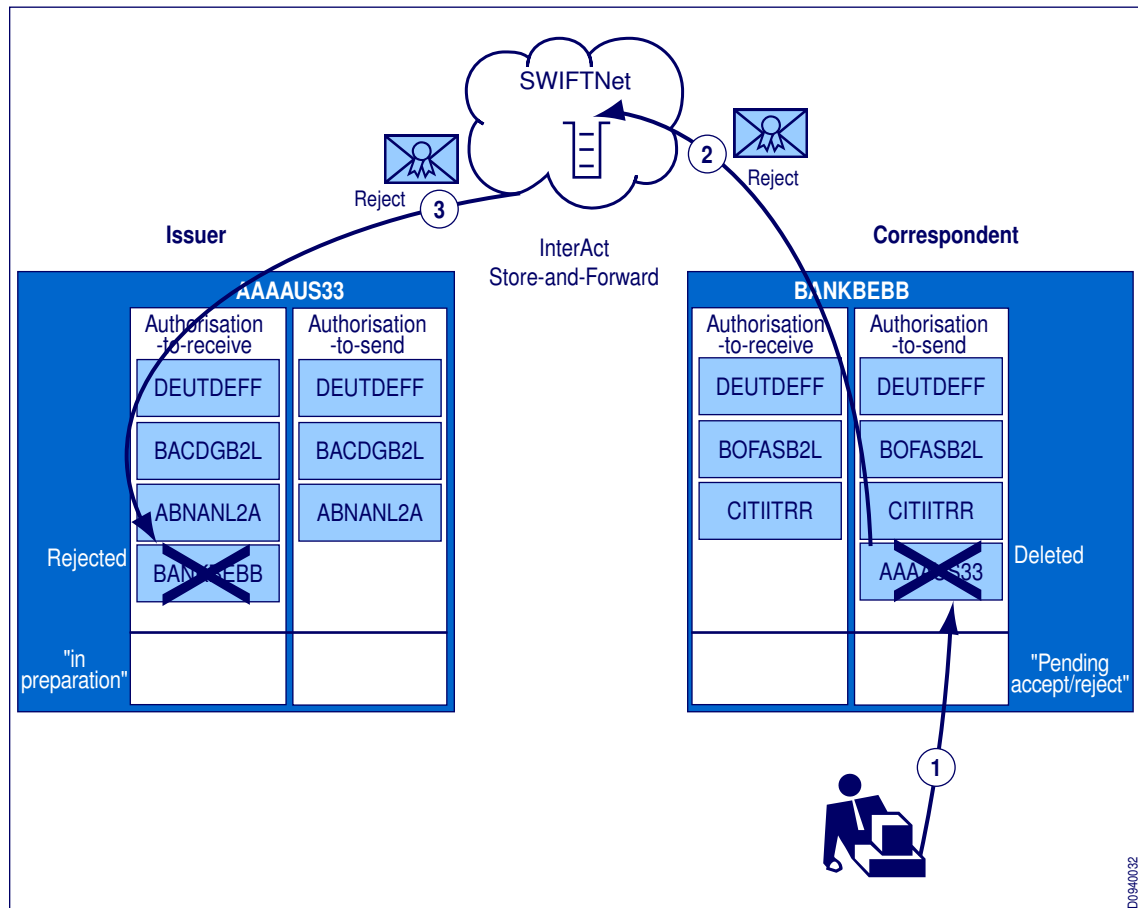
The revoking party's RMA traffic filter no longer authorises any traffic that it receives from the *revoked* correspondent on the *revoked* service. Refusal to authorise traffic occurs as soon as the status of the RMA authorisation record changes to *revoked*, even if the correspondent decides to ignore the revocation. In some circumstances, the revoking party may send traffic from the *revoked* correspondent to an investigation queue.

2.3.2 Deleting Authorisations

Overview

The deleting party instructs the RMA management software to delete an **authorisation-to-send** that it has accepted previously. RMA immediately changes the status of the RMA **authorisation-to-send** record to *deleted*. RMA then sends a reject message to the issuer of the authorisation. The rest of the process is the same as for rejecting an authorisation, see "Rejecting Authorisations" on page 21.

Authorisation deletion process



Note The situation at the start of the process is that BANKBEBB has accepted a previous authorisation from AAAAUS33. RMA has stored the authorisation on both sides of the relationship with the status *enabled*.

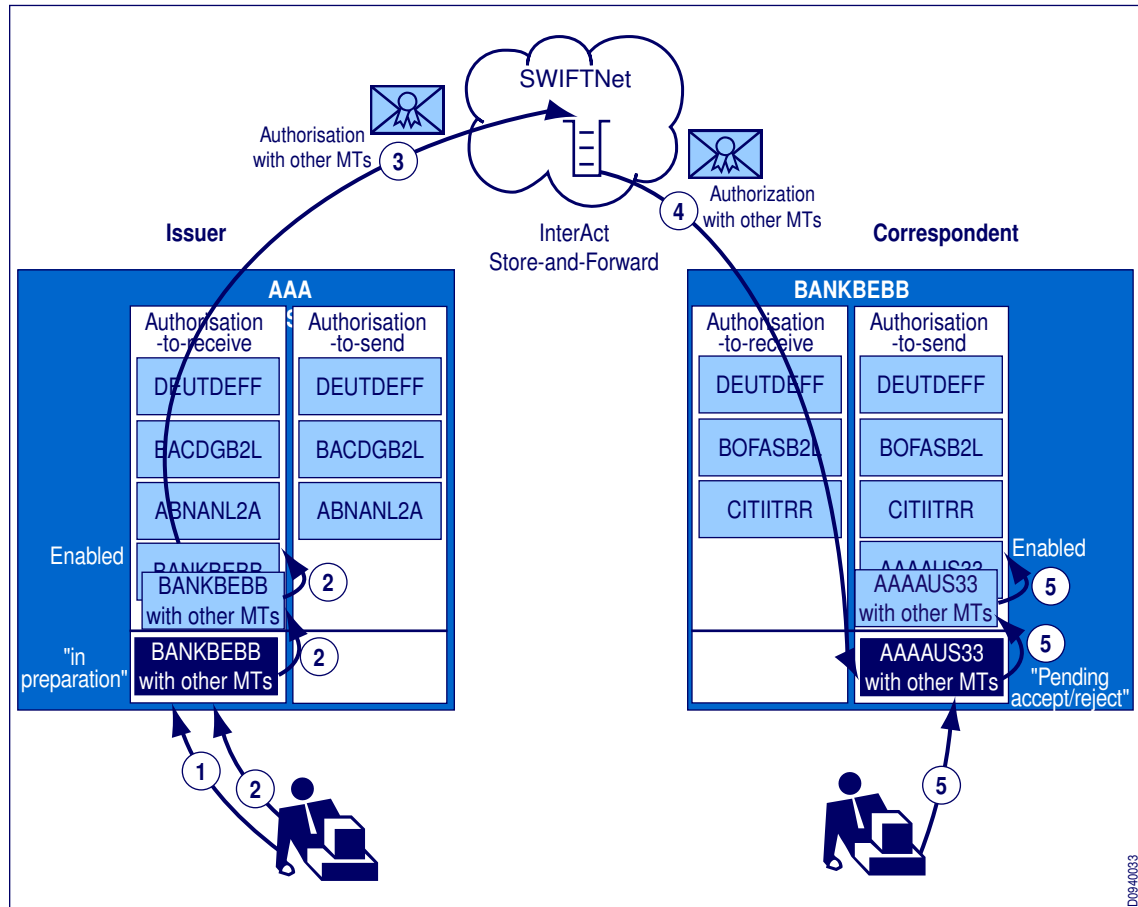
- **Step 1:** BANKBEBB decides to delete the **authorisation-to-send** that it has on file for AAAAUS33. RMA immediately puts the **authorisation-to-send** in a *deleted* state, in BANKBEBB's RMA data store. RMA then sends a reject message to AAAAUS33. From this moment, BANKBEBB is not authorised to send to AAAAUS33.
- **Step 2:** RMA sends the reject message over SWIFTNet to AAAAUS33. The reject message carries a *reject reason* that the correspondent deleted the authorisation.
- **Step 3:** AAAAUS33 receives the reject message. RMA immediately changes the status of the **authorisation-to-receive** from BANKBEBB, which is in AAAAUS33's data store, to *rejected*. From this moment, AAAAUS33 does not authorise traffic that it receives from BANKBEBB.

2.4 Changing an Existing Relationship

How to change existing authorisations

A user may want to change an existing relationship (for example, to change the types of messages that it has authorised a correspondent to send). Only the issuer of the authorisation can change the authorisation. The other party cannot change it, but only delete it, or ask the issuer (through the query mechanism) to change it. To change the relationship, the issuer must create a new authorisation and send it to the correspondent. The new authorisation overwrites the existing authorisation records at both ends.

Authorisation change process



Note The situation at the start of the process is that BANKBEBB has accepted a previous authorisation from AAAAUS33. RMA has stored the authorisation on both sides of the relationship with the status *enabled*.

- **Step 1:** AAAAUS33 decides to modify the list of Message Types (MTs) that it allows BANKBEBB to send. AAAAUS33 prepares a new authorisation for BANKBEBB in draft mode.
- **Step 2:** AAAAUS33 issues the authorisation to BANKBEBB (this may require 4-eyes approval). RMA stores the new **authorisation-to-receive** from BANKBEBB, with the status *enabled*, in AAAAUS33's RMA data store. The new authorisation immediately supersedes (overwrites) the previous **authorisation-to-receive** from BANKBEBB. From this moment, AAAAUS33 receives from BANKBEBB only those messages that appear in the new list of permitted MTs, and will

not authorise MTs that were excluded in the modified permission list (even if BANKBEBB has not accepted the modified authorisation yet).

- **Step 3:** RMA sends the authorisation message with the modified list of MTs, over SWIFTNet store-and-forward, to BANKBEBB.
- **Step 4:** BANKBEBB receives the new authorisation message and stores the message until the appropriate personnel decide whether to accept or reject it.
- **Step 5:** BANKBEBB accepts the authorisation with the modified list of permitted MTs. RMA stores the **authorisation-to-send** to AAAAUS33 in BANKBEBB's RMA data store with the status *enabled*. The new authorisation immediately supersedes (overwrites) the previous **authorisation-to-send** to AAAAUS33.

BANKBEBB does not return an acceptance acknowledgement to AAAAUS33.

Until BANKBEBB completes Step 5 (accepts or rejects), the old **authorisation-to-send** to AAAAUS33 is still in effect at BANKBEBB. Before Step 5, BANKBEBB can still send old Message Types (MTs) that are no longer permitted by AAAAUS33's new authorisation. However, AAAAUS33's receive filter would not authorise such MTs when received by AAAAUS33.

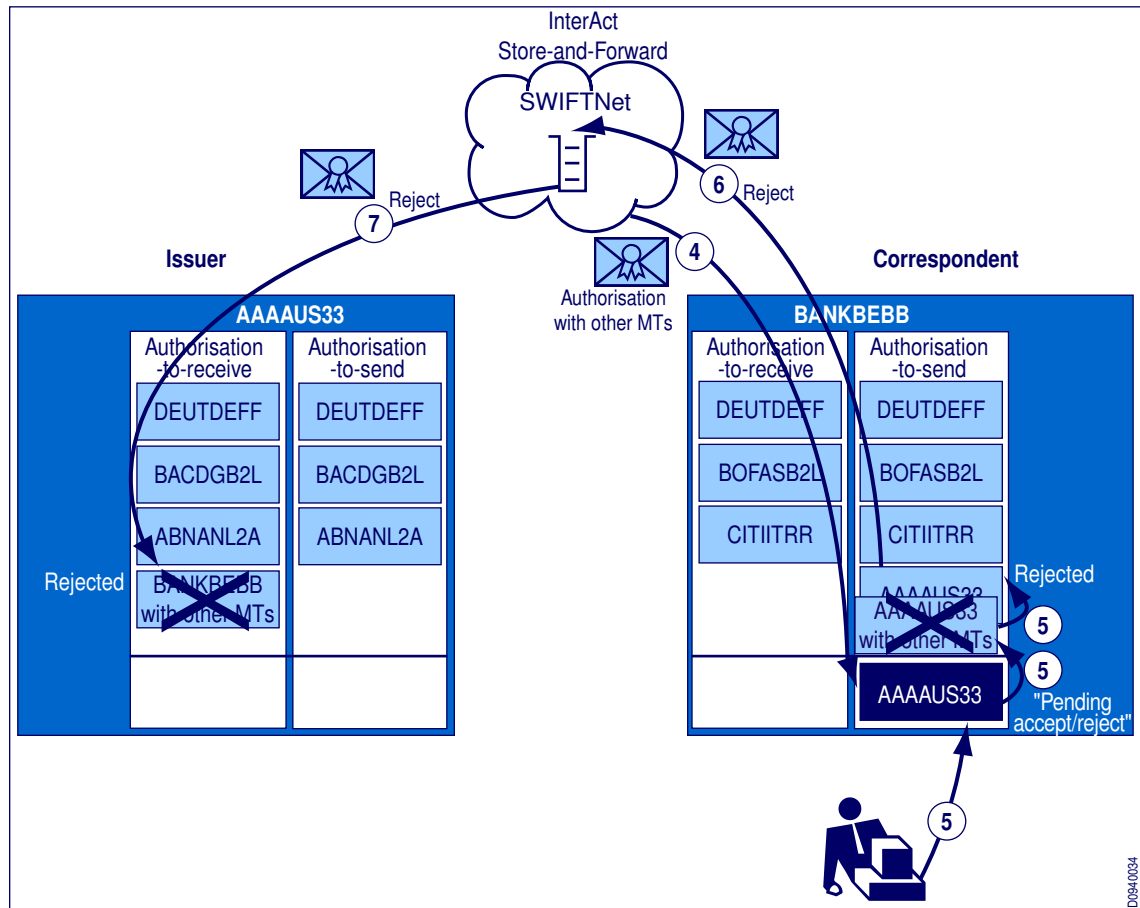
Important Issuers **must not use a start date to change an existing authorisation**. A start date is not necessary, because the new authorisation becomes effective immediately and overwrites the previous authorisation. If the new authorisation contains a start-date, then there is no authorisation in effect until that start-date. Meaning, the old authorisation does not remain in effect till the start date, since it is already overwritten by the new one.

The following section explains in more detail what happens when the incoming new authorisation is rejected instead of accepted.

How to reject a change to an authorisation

When an incoming change to an existing authorisation is rejected, the rejected authorisation will overwrite, at both ends, the previously existing authorisation.

Rejection of a change to the authorisation process



Note The first three steps in this process are the same as those of the previous process. AAAAUS33 has issued a modified authorization to BANKBEBB, but BANKBEBB has not yet received it.

- **Step 4:** BANKBEBB receives the authorization message from SWIFTNet and stores the message until the appropriate personnel decide whether to accept or reject it. Until BANKBEBB decides, the old **authorization-to-send** to AAAAUS33 is still in effect at BANKBEBB. At this moment, BANKBEBB can still send old Message Types (MTs) that are no longer permitted by AAAAUS33's new authorization. AAAAUS33's receive filter, however, would not authorize these MTs.
- **Step 5:** BANKBEBB rejects the modified authorization. RMA stores the modified **authorization-to-send** to AAAAUS33 in BANKBEBB's RMA data store in a rejected state. The rejected authorization immediately supersedes (overwrites) the previously stored **authorization-to-send** to AAAAUS33. RMA sends a reject message to AAAAUS33. From this moment, BANKBEBB can no longer send any authenticated MTs to AAAAUS33 for the rejected service (for example, SWIFTNet FIN).
- **Step 6:** RMA sends the reject message to AAAAUS33 over SWIFTNet store-and-forward.

- **Step 7:** AAAAUS33 receives the reject message from SWIFTNet. RMA immediately changes the status of the modified **authorisation-to-receive** from BANKBEBB in AAAAUS33's RMA data store to *reject*. From this moment, AAAAUS33 does not authorise any messages that it receives from BANKBEBB for the rejected service (for example, SWIFTNet FIN).

Important A user must always be careful before rejecting a change to an existing authorisation. Rejecting a change to an existing relationship ends the entire relationship. A rejection always covers the whole authorisation, including all Message Types (MTs) that the issuer has granted for the service. The receiver cannot pick and choose one or more specific MTs within the list of granted MTs. And, the RMA record in a rejected state supersedes (overwrites) the previous authorisation. After rejection, no authenticated traffic is possible anymore between the correspondent and the issuer of the rejected authorisation.

How to request a revised authorisation

If the receiver disagrees with the changed terms of an authorisation (for example, the permitted MTs), but does not want to break off the entire relationship, then this party can take the following action:

- Send an RMA query to the issuer, to request an authorisation to which both parties agree. The issuer might answer, and a conversation can take place, using RMA query/answer. In the text of the query, explain what is requested.
- If the issuer agrees to issue a new authorisation, then accept the new authorisation when it arrives. This automatically discards the previous authorisation that was never accepted or rejected.

However, if the issuer insists on the original terms of the authorisation, then the receiver must either accept or reject the proposed authorisation, within a maximum of six business days.

2.5 Importing and Exporting RMA Authorisations

File format

Users can export RMA records (authorisations) from the RMA management software into a text file with an XML structure. This text file is called an **RMA distribution file**. The main purpose of the RMA distribution file is to allow users to import authorisations into one or more traffic-filtering applications (for example, SWIFTNet FIN interfaces). Users can then manage the authorisations elsewhere. Bootstrap files use the same file format as RMA distribution files, see "BKE-to-RMA Migration" on page 46. SWIFT standardises the file format, and all vendors must be able to treat this.

For more information, see "Overall File Format" on page 60.

RMA queries and answers are not part of the distribution file. RMA management software may offer other methods to export RMA queries and answers. Similarly, the history or audit trail of how and when changes are made to authorisation records is not part of the distribution file. RMA management software may offer other methods to display or export such history or audit trails.

RMA records in the distribution file

The RMA distribution files contain the authorisations-to-send and authorisations-to-receive with their status (that is, *enabled*, *revoked*, *rejected*, or *deleted*). The distribution files also contain the parameters of the authorisation as described in "Issuing Authorisations" on page 18.

Subset of BIC8s

In the case where a customer operates more than one BIC8, the RMA distribution file can contain the records for all the customer's BIC8s, or for a subset of its BIC8s. For example, only the subset that is relevant to the branch or the traffic filtering application to which the authorisations will be distributed. The header of the RMA file indicates the list of own BIC8s that are present in the file's RMA records.

Test and Training, or Live

The RMA distribution file can contain the RMA records for multiple services (for example, SWIFTNet FIN, and in the future, other services). And, it can contain live authorisations (`swift.fin`) or Test and Training (T&T) authorisations (`swift.fin!p`), or both.

The importer of the file can decide for which BICs and which services they import the authorisations.

Authenticity and integrity protection with LAU

An RMA distribution file must have a Local Authentication User (LAU) signature. RMA calculates this signature over the entire file, with an LAU key that the customer has selected. An LAU key is a 128-bit key, which appears as a string of 32 hexadecimal characters. The LAU key is a secret that is known to both the exporter of the RMA file, and the importer of the RMA file. The LAU key does not encrypt the file, but signs it (authentication). The importer of the file (for example, a FIN interface) must verify the LAU signature before it allows the import to occur. If the LAU signature does not match, then none of the records in the file are imported.

Complete or partial

The RMA distribution file can be complete or partial.

If the distribution file is partial, then it only contains a subset of the records for a particular own BIC8 (for example, the records that were added or changed since a previous moment). When importing, the records are added or changed as appropriated, but no records are deleted.

If an RMA distribution file is complete, then the import process replaces all RMA records at the importing side with the records in the distribution file. Records are replaced only for the own BIC8s present in the distribution file. The import process deletes any records on the importing side that are not in the distribution file.

Note It is the exporting side that decides whether a file is complete or partial. The importing side cannot overrule this decision. The exporting side (for example, the head office) has full control over the authorisations for BIC8s that it manages for the importing side (for example, the branch).

Note Bootstrap files are always partial, see "The RMA Recording and Bootstrapping Process" on page 51.

Regular import and export

Important: When RMA file distribution is used, users must distribute RMA records to their FIN interfaces at least once per day, and preferably several times per day. This is required to comply with the RMA rules. To facilitate the regular distribution of RMA records, many vendors offer a facility to export automatically RMA authorisations from the RMA management software, and automatically import them in the FIN interfaces. If the SWIFTNet FIN interfaces are geographically dispersed, then the user can use SWIFTNet FileAct to transport the RMA distribution files. SWIFTNet FileAct transports the files reliably, securely, and automatically from the RMA management software to the (SWIFTNet FIN) interfaces. A customer can also use its own internal network for file transportation. In this case, however, the customer must be able to guarantee reliability and confidentiality.

2.6 Rules for Using RMA

User-to-user obligations similar to BKE policy

- Users must process (accept or reject) and distribute (if required) RMA messages (authorisations, queries) on time, and in all cases within six business days.
- If a user receives an **authorisation-to-send** for which the other party requires an equivalent authorisation (to allow SWIFTNet FIN traffic in both directions), then this user must send this equivalent authorisation within six business days.
- Users must process RMA revocations as soon as possible and, at a maximum, within one business day. This includes the time to distribute the revoked RMA records to traffic filtering applications (SWIFTNet FIN interfaces).
- Users must read and empty the RMA queues every business day.
- Users are not allowed to circumvent the protections offered by RMA, to grant themselves an **authorisation-to-send**.

2.7 Primary Operational Differences between RMA and BKE

No need to renew authorisations

Bilateral Key Exchange (BKE) is an ongoing process, in which users renew keys every 6 to 12 months. Users send a Relationship Management Application (RMA) authorisation only once. Users do not have to renew the authorisation. Because users do not have to renew authorisations, RMA generates much fewer messages than the current BKE traffic.

No need to activate on the third Sunday of the month

In the past, users had to renew BKE keys regularly. To make this process more convenient, most institutions applied a rule to activate new keys on the third Sunday of each month. Because RMA does not require users to renew authorisations regularly, this rule no longer applies. Users can activate authorisations on a date that meets business requirements. Typically, an RMA authorisation does not carry a start date, which means it is effective immediately.

No open contexts

With BKE, four messages are needed (MT 960 to MT 963) to exchange a BKE key. These four messages were potentially exchanged over the course of several days. Thus, BKE required a *context* to be kept open for several days. Establishing a relationship with RMA only requires one message. Thus, no contexts need to be kept open. This simplifies RMA operations compared to BKE.

Acquire store-and-forward queue

BKE is part of the SWIFTNet FIN message flow (MT 96n messages). Users must perform a SWIFTNet FIN Login and Select to send and receive BKE messages. SWIFT has implemented RMA as a SWIFTNet InterAct Store-and-Forward service, which means that users do not perform a SWIFTNet FIN Login and Select to send or receive RMA messages. To receive RMA messages, a user must log on to (acquire) the appropriate SWIFTNet store-and-forward queues. With no requirement to log on to SWIFTNet FIN, users can send and receive RMA messages from a system that does not act as a SWIFTNet FIN interface. Users must acquire and empty their SWIFTNet store-and-forward queues for RMA at least once per business day.

RMA is uni-directional

Typically, BKE is a bilateral protocol. At the end of the exchange, users have a bilateral key to use for both send and receive functions. Users can also exchange uni-directional keys with BKE, but this is less common. RMA is only uni-directional, which means that both parties must issue authorisations if 2-way traffic is needed. When a user receives an authorisation from a correspondent, then the user must decide whether they want to issue an authorisation back to that correspondent, and if so, issue it within six business days.

No pre-agreements

Unlike BKE, RMA does not need pre-agreements (preliminary technical agreements) with counterparts. Users were unable to use BKE without prior agreement by the parties on several technical parameters (for example, renewal frequency, and Key Management Authority [KMA]). These technical parameters do not exist in RMA.

No KMA

With BKE, the BKE messages could be sent to a different BIC than the BIC for which the keys were. This other BIC was called the Key Management Authority (KMA). With RMA, this concept of KMA does not exist: RMA messages are sent to the same BIC (except in Test and Training, see "SWIFTNet FIN Test and Training" on page 42). This simplifies things, since this eliminates the need for two correspondents to agree first on the KMA. Yet, no functionality is lost, since a party can still decide to process all the RMA authorisation messages in a different place (for example, centrally), and the other party does not have to know this. For information about how to centralise RMA, see "RMA Technology" on page 34.

No previous, current, future authorisations

BKE uses a system of previous, current, and future bilateral keys. RMA has only one active/valid authorisation. A new RMA authorisation automatically replaces the current authorisation.

Faster revocation

Discontinuation of a BKE key can take days. With RMA, a revocation only takes one message, and the revoked party cannot refuse that message. The revoking party immediately stops accepting traffic from the revoked party, and the revoked party has a maximum of one business day to process the revocation.

Faster distribution

Users that currently perform BKE key distribution are candidates for future RMA distribution. Because RMA updates become active/valid immediately, users must distribute updates faster than for BKE. BKE users typically exchange keys that only become active in the future, which gives the user time to distribute the keys. To cater for the immediacy of RMA updates, and to comply with the rules for using RMA (see, "Rules for Using RMA" on page 31), RMA users must aim for at least daily, and preferably several times per day, or even real-time distribution if possible.

Updating of back-office systems

Some users, working with their back-office systems, keep track of the correspondents to whom they can send authenticated messages (with whom they have a BKE key), to help the back-office applications or operators. In this case, the user may want to transfer RMA authorisations to these back-office systems. The RMA distribution file can be used for that purpose.

3 RMA Technology

Introduction

To configure the RMA management software correctly, the user requires some knowledge of how RMA uses SWIFTNet's technical features. This chapter explains this.

3.1 RMA Use of InterAct Store-and-Forward

RMA messages

RMA messages are XML messages that are carried inside SWIFTNet InterAct Store-and-Forward messages. Thus, each RMA message (authorisation, revocation, reject, query, or answer) is always contained in an InterAct envelope. This section describes how this envelope is expected to be filled in for an RMA message, and how the SWIFTNet Central System will process the message.

For general information about InterAct and store-and-forward, please refer to the *SWIFTNet Service Description*.

Requestor and responder DNs

When sending an RMA message, the user sends it from a BIC8 to another BIC8. The RMA management software typically uses this BIC8 to fill in a requestor DN (sender) and responder DN (receiver) in the InterAct envelope. For an RMA message, the requestor and responder DNs always follow a particular convention.

The requestor Distinguished Name (DN) of an RMA message always has two levels. The second level contains the live BIC8 (usually in lower case) of the institution that sends the RMA message.

Example:

- `o=aaaaaus33, o=swift`

The responder DN of an RMA message always has three levels. The second level contains the live BIC8 (usually in lower case) of the institution that receives the RMA message. The third level contains an abbreviation of the service name for which the issuer has granted the authorisation.

Example:

- `cn=swiftfin, o=bankbebb, o=swift`

(for an RMA message that relates to either live or Test and Training [T&T] SWIFTNet FIN)

Note

As for all InterAct messages, the requestor and responder DNs of RMA messages cannot contain T&T BICs, meaning BIC8s with a zero in the eighth character. Users must always use a live BIC8 on the second level of valid DNs in the envelope of SWIFTNet InterAct messages. Therefore, before a T&T RMA message can be sent, the user must know a live BIC8 where the correspondent wants to receive RMA messages for a given T&T BIC8.

Service name

Users send live RMA messages over the `swift.rma` SWIFTNet service. Users send T&T RMA messages over the `swift.rma!p` SWIFTNet service.

Request types

An RMA message request type indicates the type of RMA message (authorisation, revocation, reject, query, or answer). Typically, the RMA management software inserts the correct RequestType value when the issuer sends an RMA message. The following table shows the possible values for these RequestTypes as currently known (early 2007). SWIFT can change the RequestType value for authorisation requests when new FIN message standards become available. For more information, see "Impact of New Message Standards" on page 40.

These RequestTypes follow the convention for XML messages: the first part (*xrma*) indicates the service, the second part (*001, 002, ..., 005*) indicates the message type, the third part (*001 or 002*) indicates the *variant* of the message type (live or test). Only authorisation, revocation, and reject have a live or test variant. Query and answers have no variant (have the same message structure for live or test). The last digit (*01, 02 ...*) indicates the version of this message. This version can change when new message standards are introduced.

Request	RequestType for Live RMA swift.rma	RequestType for T&T RMA swift.rma!p
Authorisation	"xrma.001.001.01" (for FIN Message Standards 2006) "xrma.001.001.02" (for FIN Message Standards 2007)	"xrma.001.002.01" (for FIN Message Standards 2006) "xrma.001.002.02" (for FIN Message Standards 2007)
Revocation	"xrma.002.001.01"	"xrma.002.002.01"
Reject	"xrma.003.001.01"	"xrma.003.002.01"
RmaQuery	"xrma.004.001.01"	"xrma.004.001.01"
RmaAnswer	"xrma.005.001.01"	"xrma.005.001.01"

Non-repudiation

RMA messages always use the non-repudiation feature of SWIFTNet InterAct, meaning, the non-repudiation flag must be turned on in the InterAct envelope of all RMA messages. Non-repudiation means that the sender cannot deny having sent the message, and the receiver cannot deny having received it. For more information about non-repudiation, see the *SWIFTNet Service Description*. See the *SWIFTNet Messaging Operations Guide* for the procedure that requests SWIFT to verify the transmission or receipt (or both) of a message that is in dispute.

MVAL

SWIFT applies message validation on all RMA messages. Message VALidation (MVAL) is the SWIFTNet Central System that validates XML message syntax against rules specified in an XML schema. MVAL guarantees that receivers only receive syntactically correct RMA messages. The validation rules depend on the Service and the Request Type. For example, an authorisation for live SWIFTNet FIN can carry a permission list of permitted SWIFTNet FIN Message Types (MTs). The validation rules check that all MTs in the permission list are existing authenticated SWIFTNet FIN MTs, according to the active live FIN Message Standards. For more information, see "Impact of New Message Standards" on page 40.

Store-and-forward queue names

For each customer that subscribes to the RMA service, SWIFT automatically creates default SWIFTNet store-and-forward queues. RMA messages for a customer are delivered to those queues. The queue name contains the live BIC8 of the customer (in lower case). The following table shows the queue names for a customer that has the live BIC8 AAAAUS33 and the Test and Training (T&T) BIC8 AAAAUS30.

	Live RMA	T&T RMA
Store-and-forward queue name	aaaaus33_rma	aaaaus33_rma!p

Note The T&T queue name ends with an !p and contains the live BIC8, rather than the T&T BIC8.

Message Routing Rules

For each customer that subscribes to the RMA service, SWIFT automatically creates default Message Routing Rules (MRRs). MRRs route RMA messages for a customer to that customer's store-and-forward queues.

The MRRs for live and T&T RMA are as follows:

- **MRR rule for live RMA**

RMA messages (SWIFTNet InterAct requests) sent to responder *,o=<BIC8>,o=swift on service swift.rma are routed to store-and-forward queue <BIC8>_rma (for example aaaaus33_rma)

- **MRR rule for T&T RMA**

RMA messages (SWIFTNet InterAct requests) sent to responder *,o=<BIC8>,o=swift on service swift.rma!p are routed to store-and-forward queue <BIC8>_rma!p (for example aaaaus33_rma!p)

PKI signatures

All RMA messages carry a PKI digital signature, which protects the message from tampering, and provides proof of the sender's identity. The signature is calculated by the sender, using the certificate associated to the sender's *signer DN*. This is further explained in the section "Certificates, and RBAC Roles for RMA" on page 38.

Signature storage

If an issuer successfully sends an authorisation message to SWIFTNet (that is, the message is store-and-forward ACKed), then the sending RMA management software receives from the sending SWIFTNet Link (SNL) the value of the Public Key Infrastructure (PKI) digital signature that was calculated on the RMA message. RMA stores this signature in the RMA record.

When RMA receives an authorisation message from SWIFTNet, the RMA management software extracts the same signature. Once again, RMA stores this signature in the RMA record. In this way, both the issuer's **authorisation-to-receive** and the correspondent's **authorisation-to-send** contain the same signature. Issuer and correspondent can compare the signatures to validate that the **authorisation-to-receive** corresponds exactly to the **authorisation-to-send**.

Closed User Group

The `swift.rma` and the `swift.rma!p` service use the Closed User Group (CUG) feature of SWIFTNet. SWIFT automatically subscribes new SWIFTNet FIN customers to these services, and therefore new FIN users are automatically members of these CUGs. For more information about the transition from BKE-to-RMA for existing (not new) SWIFTNet FIN customers, see "BKE-to-RMA Migration" on page 46.

RBAC

Only Distinguished Names (DNs) that have the appropriate Role-Based Access Control (RBAC) role can send (sign) RMA messages. The role is called `RMA` on the `swift.rma` and `swift.rma!p` services. For more information, see "Certificates, and RBAC Roles for RMA" on page 38.

Similarly, only DN's that have the RBAC role that gives the right-to-read from store-and-forward queues can receive (read) the RBAC messages from those queues.

Error handling

- **Failure to send**

If the RMA management software encounters an unrecoverable error when it attempts to send an RMA message, then this is flagged to the RMA manager. The flagging mechanism is vendor-specific. For more information, see your RMA management software documentation.

- **Retries**

If the sending RMA management software encounters a recoverable error when it attempts to send an RMA message, then it can try to re-send the message automatically. The number of times that RMA can re-send is vendor-specific. For more information, see your RMA management software documentation.

- **Delivery notifications**

Users can optionally send RMA messages with the store-and-forward delivery notification option. If the user sets this option, then the SWIFTNet system returns a delivery notification message to the original sender of the RMA message on successful delivery. The delivery notification message is a SWIFTNet InterAct Store-and-Forward message. SWIFTNet delivers the notification message to a delivery notification queue that the sender specified when it sent the message. The delivery notification queue is normally the same queue as that for incoming RMA messages. The way in which a customer can track which RMA messages have or have not yet received a delivery notification is vendor-specific. For more information, see your RMA management software documentation.

- **Failed delivery notification**

It is possible that the SWIFTNet system is unable to deliver the RMA message. Failure to deliver usually indicates a problem at the receiver's side.

The following examples describe typical problems that result in failure to deliver:

- The receiver does not read its store-and-forward queues. In this case, SWIFTNet waits 14 days. If the receiver does not read the message from the queue in that period, then SWIFTNet aborts the message (removes it from the queue). SWIFTNet sends a failed delivery notification to the sender.
- The receiver fails to read the message properly from the queue (for example, does not return a proper store-and-forward ACK). In this case, SWIFTNet tries to deliver the message up to 10 times, after which it aborts the message (removes the message from the queue). SWIFTNet sends a failed delivery notification to the sender.

Failed delivery notification is a standard, compulsory feature of SWIFTNet store-and-forward. Failed delivery notifications are SWIFTNet InterAct Store-and-Forward messages, which SWIFTNet returns to the notification queue that the sender indicated in the original message. Typically, the queue indicator is a configuration parameter at the sender's end. The sender usually selects the same queue that receives its other RMA messages.

- **Duplicate message detection**

As the previous error scenarios indicate, a user may send the same message more than once, or SWIFTNet may attempt to deliver a message more than once. This is not a problem for RMA. RMA authorisations always have a unique date and time stamp (DateTime Issued). If SWIFTNet delivers the same message twice to the receiving RMA management software, then this software ignores duplicates which it identifies by the date and time stamp.

- **Out-of-sequence delivery**

Although rare, it is possible that the store-and-forward system may deliver messages out of sequence. Delivery of out-of-sequence messages is not a problem for RMA messages, because they always carry a unique date and time stamp. RMA management software automatically discards older RMA messages if newer RMA records are already on file.

3.2 Certificates, and RBAC Roles for RMA

Live RMA

Live RMA messages must be signed with a Public Key Infrastructure (PKI) certificate that is of the *business* type. These certificates must be stored on a Hardware Security Module (HSM) and must have policy ID 1.3.21.6.2.

The signing PKI certificate's Distinguished Name (the signer DN) must contain, at level 2 of the DN, the same live BIC8 as the issuer BIC8 of the authorisation. The issuer BIC8 must also be the same live BIC8 as in level 2 of the requestor DN. The signer DN must have the Role-Based Access Control (RBAC) role *RMA* for the `swift.rma` service. A Security Officer (SO) within the institution grants the RBAC role to a DN. By granting the DN the correct type of certificate and the appropriate RBAC role, the SO decides which DNs can be used for live RMA.

Note All RMA messages (that is, authorisation, revocation, reject, query, and answer) are signed. For each message, the signer DN must fulfil the conditions in the previous paragraph.

To receive live RMA messages, the user must read the store-and-forward queue `<liveBIC8>_rma`, for example, `aaaaus33_rma`. To be permitted to read the queue, the user must acquire the queue, presenting a signer DN that has the appropriate RBAC role. More precisely, the SO must grant the signer DN this queue name as qualifier to the RBAC role `SnFRequestor` under the `swift.snf.control` service.

Test and Training RMA

Users can sign a Test and Training (T&T) RMA message with either a business certificate on an HSM (policy ID 1.3.21.6.2) or a lite certificate (no policy ID). Lite certificates can be on a disk or on an HSM. The signing certificate's DN must contain, at level 2 of the DN, a live BIC8 (for example, `BANKBEBB`), preferably the BIC8 that owns the issuer's T&T BIC8 (for example, `BANKBEB0`), though any other live BIC8 is allowed. In this way, the certificate (signer DN) that signs T&T RMA messages can be the same certificate that signs live RMA messages for a given BIC8. RMA does, however, permit different certificates to sign T&T and live RMA messages. An SO must have granted the signer DN for T&T the RBAC role *RMA* for the `swift.rma!p` service.

Note The exclamation mark and the 'p' at the end is a convention used on SWIFTNet to distinguish pilot and test services from live services.

To receive T&T RMA messages, the customer must use a signer DN that has the appropriate RBAC role to read the store-and-forward queue `<liveBIC8>_rma!p`, for example, `aaaaus33_rma!p`. More precisely, an SO must grant the signer DN this queue name as qualifier to the RBAC role `SnFRequestor` under the `swift.snf.control` service.

Note Users cannot use a DN that only has live RMA roles for T&T RMA. Conversely, users cannot use a DN that only has T&T roles for live RMA

Managing the authorisations for multiple BIC8s from the same RMA management software

For each BIC8 that issues authorisations, an SO must grant a DN the correct type of certificate and the correct RBAC roles. Therefore, if the RMA management software must manage authorisations for multiple BIC8s, then it requires multiple DNs and multiple certificates. The RMA management software requires one DN per live BIC8, each with its own certificate and appropriate RBAC roles.

Using the same certificate for live RMA and Test & Training RMA

If the user wants to use the same signer DN and certificate for both live and T&T RMA, then the following conditions apply:

- The certificate of that DN must be a *business* type (policy ID 1.3.21.6.2), and must be on an HSM.
- The BIC8 at level 2 of that DN must be a live BIC8. Preferably, this live BIC8 should be the owner of the T&T BIC8, but this is not mandatory. The user must communicate this live BIC8 to its correspondents, as the live BIC8 on which they want to receive RMA for the T&T BIC8. A user can use the same live BIC8 for all its T&T BIC8s, (for example, to economise on the number of certificates).
- The DN must have the RMA RBAC role under both the live `swift.rma` service and the T&T `swift.rma!p` service.
- The DN must have the RBAC role that permits to read from the live RMA store-and-forward queue `<BIC8>_rma`, and the RBAC role that permits to read from the test RMA store-and-forward queue `<BIC8>_rma!p`.
- The RMA management software must send the live and T&T RMA messages through the same SWIFTNet Link (SNL). The reason for this condition is that two different SNLs cannot simultaneously access a single certificate.

Note: If the RMA management software manages authorisations for multiple T&T BICs, then it is possible to use one single live BIC8's certificate for all these T&T BICs.

Using a different certificate for live RMA and Test and Training RMA

If the user wants to prevent live RMA messages being sent from the system that does T&T RMA, then one possible technique is to use a different certificate for live and test RMA.

In that case, the following conditions and statements apply:

- The signer DN that has the live `swift.rma` RBAC role must be different from the signer DN with the T&T `swift.rma!p` RBAC role. For example, `cn=live, o=aaaaus33, o=swift` and `cn=test, o=aaaaus33, o=swift`.

- Both DN's can have a different live BIC8 in the level 2 of the DN. The live BIC8 in the DN used for T&T RMA should preferably be the owner of the T&T BIC8, but this is not mandatory.
- The DN's must have different certificates (two different DN's can never share the same certificate). The certificate used for live RMA must be a business certificate (policy ID 1.3.21.6.2) and must be on HSM, while the certificate used for T&T can be a Lite certificate on disk or HSM.
- The DN for live RMA must have the RBAC role that permits to read from the live RMA store-and-forward queue `<BIC8>_rma`, and the DN for T&T RMA must have the RBAC role that permits to read from the test RMA store-and-forward queue `<BIC8>_rma !p`.
- The RMA management software for live and for test can be on the same or on different systems (using same or different SNLs).

Using the same certificate for RMA and SWIFTNet FIN

The RMA management software must use signer DN's and certificates that comply with the rules for RMA in the previous topic. By contrast, the RMA traffic filtering software (for example, a SWIFTNet FIN interface) must use signer DN's that comply with the rules for the traffic on that specific service (for example, the rules for FIN). In particular, for both live FIN and T&T FIN messages, the signer DN must contain the same live BIC8 as in the FIN message header, and must also have the RBAC role `FIN` on the `swift.fin` service.

For live SWIFTNet FIN (in Phase 2) the signer DN must have a business certificate that has policy ID 1.3.21.6.2 on a Hardware Security Module (HSM). For T&T SWIFTNet FIN, the signer DN can have a lite certificate. Therefore, the user can use the same signer DN, and the same certificate, for both RMA management and for SWIFTNet FIN. Users can apply segregation by not assigning the appropriate RBAC roles to those signer DN's.

3.3 Impact of New Message Standards

Overview

When SWIFT introduces new SWIFTStandards FIN messages, this typically changes the syntax of existing FIN message types. Such changes have no impact on RMA. However, sometimes, entirely new FIN Message Types (MTs) may become available or old MTs may be removed (obsoleted). If these new or removed MTs also require authentication, then there is a minor impact on RMA, as described in this section.

In any case, all existing authorisations that are already stored, remain valid. There is no need to re-exchange authorisations with all correspondents when a new FIN message standard is introduced.

Well in advance of the SWIFTStandards FIN message activation, SWIFT will allow the exchange of authorisations that contain any new MTs in the permission lists. Therefore, a customer can grant or deny its correspondents permission to send these MTs, before the live availability of these MTs on SWIFTNet FIN.

An authorisation by default permits all MTs. And when not all MTs are permitted, then often an entire category of MTs (for example, the entire category 1) is included or excluded, instead of enumerating all permitted (or excluded) MT numbers. With such authorisations, message standard changes that introduce new MTs, have no impact.

Impact on RMA management software:

- SWIFT will communicate such changes to RMA management software vendors well in advance. In this way, a vendor can ensure that its RMA software is able to issue authorisations with the new MTs, well in advance of the availability of this MT in live SWIFTNet FIN.
- A customer must update its RMA management software for this, in advance of the live availability of the FIN message standards.

4 Testing RMA

4.1 SWIFTNet FIN Test and Training

Sending and receiving Test and Training (T&T) authorisations

To send T&T authorisations, a user must use a PKI certificate (signer DN) set up according to the rules described in "Certificates, and RBAC Roles for RMA" on page 38.

To issue a T&T RMA authorisation to a correspondent's T&T BIC8, a user must know the live BIC8 that the correspondent wants to use to receive T&T RMA messages. That live BIC8 must be present in the responder DN of the T&T RMA messages that the user issues to that correspondent T&T BIC8.

Therefore, before sending T&T authorisations to a correspondent, a user must ask the correspondent which live BIC8 they want to use for T&T RMA for its T&T BIC8. In the same way, to receive T&T authorisations, a user must tell its correspondents which live BIC8 they want to use to receive RMA for its T&T BIC8.

Example

AAAAUS30 wants to exchange T&T authorisations with BANKBEB0:

1. AAAAUS30 asks BANKBEB0 which live BIC8 they want to use for T&T RMA.
(This live BIC8 can be BANKBEBB, BANKBEB2, or an entirely different live BIC8 [for example, BBBBGB2L]).
 2. AAAAUS30 uses the RMA software to configure this live BIC8 as the responder BIC8 (sometimes called correspondent's signing BIC8) when sending T&T RMA messages to BANKBEB0.
 3. AAAAUS30 informs BANKBEB0 which live BIC8 they want to use for receiving RMA messages for T&T BIC8.
(BANKBEB0 must use this live BIC8 as the responder DN when they send T&T RMA messages to AAAAUS30. For example, assuming that the live BIC8 is AABBBDEFF, the T&T authorisations that AAAAUS30's correspondent issues to them arrives on the store-and-forward queue, `aabbdef_f_rma!p`).
- AAAAUS30 can then read the authorisations from that queue, provided they have a PKI-certificate for AABBBDEFF with the appropriate RBAC role. For more information, see "RMA Technology" on page 34.

Using or bypassing RMA for Test and Training

RMA is optional for SWIFTNet FIN T&T. A parameter on the SWIFTNet FIN interface enables users to bypass the RMA traffic filtering for SWIFTNet FIN T&T traffic. This *authorisation required for T&T* parameter offers users the most flexibility for working in FIN T&T mode. For example, if this *authorisation required for T&T* parameter is set to *No* users do not have to set up authorisations with themselves before sending themselves FIN T&T traffic. This option also enables users to test a new business flow without an exchange of authorisations with a correspondent.

This option only affects the traffic filtering, not the exchange of authorisations. Regardless of the setting of this option, it remains possible to exchange authorisations for FIN T&T. Such T&T authorisations are then ignored for traffic filtering, on FIN interfaces where *authorisation required for T&T* is turned off.

This option can also be used to let incoming tankfile messages pass authorisation. For more information about testing with the tankfile, see "Testing with the Tankfile" on page 44. This option is system-wide on a FIN interface. It is sometimes called the *RMA bypass for T&T*. *RMA bypass for T&T 'on'* is equivalent to *authorisation required for T&T 'off'*.

4.2 RMA Sparring Partner

Overview

SWIFT offers an RMA Sparring Partner that resembles the Bilateral Key Exchange (BKE) Sparring Partner. The RMA Sparring Partner enables users to become familiar with the RMA application in the context of a test environment. The SWIFT Customer Security Management (CSM) department operates the RMA Sparring Partner.

Note SWIFT recommends that users exchange RMA messages with the RMA Sparring Partner on T&T for the SWIFTNet FIN service. A successful exchange of messages with the RMA Sparring Partner is a useful step towards RMA readiness.

BIC8 for the RMA Sparring Partner

The T&T BIC8 for the RMA Sparring Partner is SWHQBE90.

Note The RMA Sparring Partner's live BIC8 is SWHQBE9H. This BIC8 must be used to issue T&T authorisations to the Sparring Partner (in the responder DN).

When replying to you, the Sparring Partner automatically uses the same live BIC8 (in the responder DN), as you have used in the signer DN of the RMA message to which the Sparring Partner replies.

RMA Sparring Partner functionality

The RMA Sparring Partner performs the following tasks:

- **Accepts or rejects the authorisations it receives.**

The system always accepts authorisations that it receives and automatically returns an authorisation that has analogue permissions. If the RMA Sparring Partner receives a reject message, then the system returns a reject message that has the same Reason Code.

- **Sends and revokes authorisations issued.**

The RMA Sparring Partner always accepts revocation messages that it receives. The system immediately replies with the reverse revocation message.

- **Sends and receives RMA query and answer messages.**

If the RMA Sparring Partner receives a query message, then the system replies with an answer message that has predefined text. The system also replies with an analogue query message.

- **Compatible with any RMA.**

The RMA Sparring Partner manages incoming requests from any RMA manager software that complies with the SWIFTNet Relationship Management Application Vendor Specification.

4.3 Synonym Testing

Synonyms and RMA

In Phase 1, a synonym BIC would require its own bilateral keys. Similarly, in Phase 2, a synonym BIC requires its own RMA authorisations. An authorisation for the live master BIC of a synonym does not authorise the synonym BICs to send traffic. A synonym BIC8 requires its own authorisation.

Synonyms and Test and Training

Synonyms are usable only on the live SWIFTNet FIN service. Synonym testing is not available in Test and Training (T&T). This is independent of RMA, or of SWIFTNet Phase 2. For RMA testing, this means that users cannot test traffic filtering for synonyms in FIN T&T. Instead, a user must check with its SWIFTNet FIN interface vendor to ensure that the synonym functionality is part of the vendor's qualification cycle. This is not applicable for RMA management software, used to exchange authorisations. For RMA management software, there is no difference between synonym BICs and other BICs. Thus, the user does not have to perform special tests dedicated to synonyms on its RMA management software.

4.4 Testing with the Tankfile

Overview

A SWIFTNet FIN user can send an MT 073 Message Sample Request to ask the SWIFTNet FIN system to send sample Test and Training (T&T) SWIFTNet FIN messages to the user's SWIFTNet FIN interface. For more information about this feature, see the *SWIFT User Handbook*, *SWIFTNet FIN Operations Guide* and *SWIFTNet FIN System Messages*.

In response to an MT 073, SWIFT systems send sample SWIFTNet FIN messages (tankfile messages) that originate from a Pseudo Logical Terminal (PLT) (for example, LRLRXXXX, also called a Pseudo-BIC8) to the T&T BIC8 that requested the samples (for example, to BANKBEB0).

Users that receive tankfile messages during SWIFTNet Phase 1 can authenticate these messages by means of a manual bilateral key, which the user enters on the SWIFTNet FIN interface. For more information about this, see the *SWIFTNet FIN Operations Guide*.

In SWIFTNet Phase 2, the user has the choice between the following two scenarios for the receipt of tankfile messages that require authentication:

- The user can configure its SWIFTNet FIN interface so that it does not require RMA authorisation for T&T. Tankfile messages (and all other T&T messages) then pass the authorisation check, when the user is logged on in Phase 2.
- The user can configure the SWIFTNet FIN interface to require authorisation for T&T. Then FIN flags an incoming tankfile message (of a message type that requires authentication, for example, MT 103), in the SWIFTNet FIN interface as having failed authorisation, when the user is logged on in Phase 2. Depending on the capabilities of the SWIFTNet FIN interface, the user may still be able to do something meaningful with these messages (for example, the user can manually bypass the authorisation check).

Users cannot use RMA messages to issue an RMA authorisation from a T&T BIC to a Pseudo-BIC (Pseudo-LT). This is because a Pseudo-BIC8 such as LRLRXXXX appears to be a live BIC8 (no zero in the 8th character), and RMA does not allow the exchange of authorisations between test and live BICs.

However, it is possible that the process of RMA bootstrapping converts the manual bilateral key between the user's BIC and a Pseudo-BIC, to an **authorisation-to-receive** from the Pseudo-BIC.

For more information about the RMA traffic recording and RMA bootstrap process, see "The RMA Recording and Bootstrapping Process" on page 51.

4.5 Mandatory Test Scenarios

Overview

The objective of RMA is to stop unwanted traffic, and thus RMA provides the user with considerable power. Mistakes should be avoided, when using RMA in a live environment.

For example:

- If a user accidentally issues a revocation to one of its correspondents, then this correspondent can no longer send authenticated FIN traffic to the user.
- If a user receives an authorisation, and accidentally rejects it (or accepts it first, and then deletes it), then the user can no longer send authenticated FIN traffic to its correspondent.

In both cases, the party that cannot send traffic any more, requires the other party to help them correct the situation by re-issuing a correct authorisation.

This shows how important it is that all SWIFT users have properly trained staff, when dealing with live RMA.

To emphasise this importance, and to protect the community, SWIFT mandates that a user performs a minimum set of RMA test scenarios in T&T, and that a user confirms explicitly to SWIFT that they are ready for live RMA operations, before SWIFT technically enables the user to perform live RMA.

The minimum test scenarios are available on www.swift.com/swiftnetphase2. The explicit confirmation of readiness is through a form that must be submitted on www.swift.com > Ordering & Support, as of 2008.

For more information about the rules governing the BKE-to-RMA migration, see "BKE-to-RMA Migration" on page 46.

5 BKE-to-RMA Migration

Overview

During 2008, all SWIFTNet FIN users will switch from Bilateral Key Exchange (BKE) to RMA to manage relationships.

5.1 Principles of the Migration

No "big bang"

At no point during the migration will all customers be required to perform an action simultaneously.

Customers can migrate independently from each other

A customer's migration is not in any way linked to its correspondents' migration. This means that all customers can perform the technical steps to migrate from BKE to RMA, independently from their correspondents.

Of course, if after those technical steps are completed, a customer wants to exchange RMA authorisations with another customer, both parties must be ready for that. Thus, migration to RMA requires customers, at some point, to wait for other customers to be ready. That is why there is a specific moment, which SWIFT calls T2, when all customers are ready for RMA. In this respect, customers do not migrate completely independently from each other.

Automatic conversion of BKE keys to RMA authorisations

The method that SWIFT designed to convert BKE keys into RMA authorisations avoids the need for all users to exchange authorisations with each other.

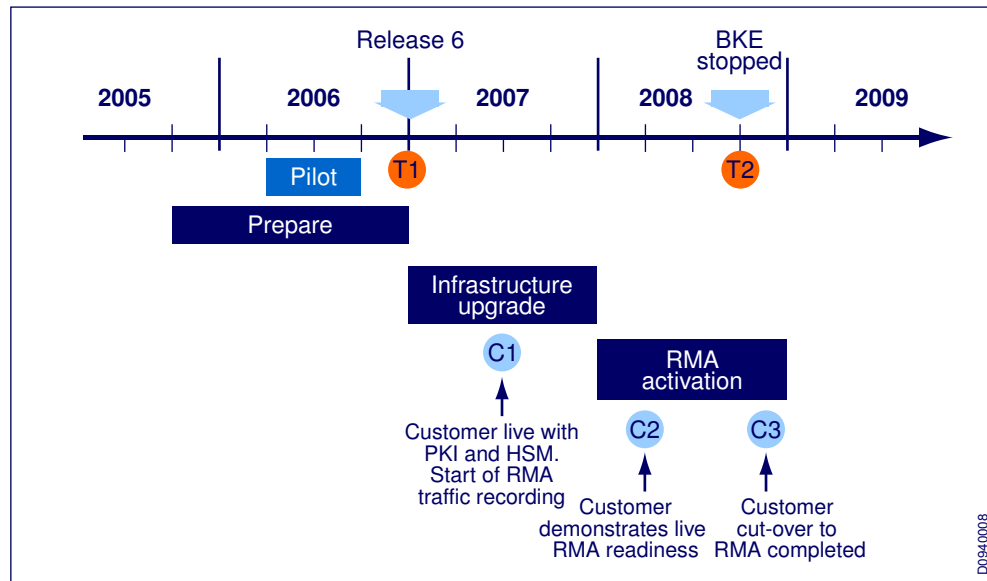
The conversion method consists of two steps:

- All FIN interfaces will perform *RMA traffic recording*. This means that the FIN interface keeps track of the BIC8s with which it has exchanged authenticated FIN traffic.
- Each RMA management software application will *bootstrap* the RMA records. RMA bootstrapping only generates RMA records for correspondents with whom the user has a valid BKE key, and with whom the user has actually exchanged authenticated SWIFTNet FIN traffic in the period preceding the bootstrapping (typically more than one year). Therefore, the recording and bootstrap mechanism performs a cleanup of the existing bilateral key file. For more information about the recording and the bootstrap mechanism, see "The RMA Recording and Bootstrapping Process" on page 51.

5.2 Timeline and Customer Milestones

Overview

This section briefly shows the overall timeline and intermediate milestones. The activities leading up to these milestones are explained in the next section, "Migration Activities" on page 48.



T1 milestone (2 January 2007)

All SWIFTNet Phase 2 components are available from SWIFT.

C1 milestone (in 2007)

The customer has completed the infrastructure upgrade. The customer has installed SNL Release 6 and related SWIFTNet FIN interface software. The certificates that the customer uses for SWIFTNet FIN are on Hardware Security Modules (HSMs). Users can log on to live SWIFTNet FIN with protocol version 3, and send live, dual-signed SWIFTNet FIN messages.

Sent messages are dual-signed as follows:

- once with a Message Authentication Code (MAC) signature calculated with a BKE key
- once with a Public Key Infrastructure (PKI) signature calculated on an HSM

The SWIFTNet FIN interface automatically starts the RMA traffic recording, see "The RMA Recording and Bootstrapping Process" on page 51.

Live RMA service start (January 2008)

This is the earliest date when customers can be activated on the live RMA service.

Before the C2 deadline (in the first and second quarter of 2008)

The customer demonstrates its readiness to perform live RMA.

Note Customers can find the C2 deadlines that SWIFT has assigned to individual countries at www.swift.com/swiftnetphase2.

T2 milestone (planned for end September 2008)

Customers can no longer use Bilateral Key Exchange (BKE) to maintain relationships. Customers use live RMA to manage relationships, and can fully benefit from the new features offered by RMA.

C3 milestone (quarter 4 2008)

The customer completes the cutover to RMA by bootstrapping its RMA records. The customer stops RMA recording, stops using BKE keys, and stops dual-signing. Once all customers have reached C3, the SWIFTNet Phase 2 migration is complete.

T3 milestone (planned in 2009)

SWIFT refuses dual-signed messages.

5.3 Migration Activities

RMA preparation and testing: between C1 and C2

1. The customer must test the management of authorisations with the new RMA messages on Test and Training (T&T) RMA. Customers can use the RMA Sparring Partner for these tests. SWIFT monitors customer tests to ensure that customers perform a minimum set of test scenarios. The test scenarios are available on www.swift.com/swiftnetphase2.

Note All RMA messages are free of charge until the end of the migration (end of 2008, 3 months after T2).

2. If applicable, the customer must test the export and import distribution of authorisations from the RMA management interface to the SWIFTNet FIN interface.
3. The customer must test the backup procedures, and resilience and disaster recovery procedures that protect the RMA datastore. Resilience and disaster recovery tests ensure that customers do not lose RMA authorisations, and can fulfil the obligation to process incoming RMA messages every business day.
4. The customer must adapt relationship management business processes to RMA. Evaluate the possibilities offered by the new features of RMA that did not exist in Bilateral Key Exchange (BKE) (for example, granularity). Ensure that compliance departments, and correspondent banking teams know of the new features.
5. The customer must ensure RMA operators receive adequate training.
6. The customer can monitor the progress of RMA traffic recording on its SWIFTNet FIN interfaces, with the RMA migration status report. This report shows the BKE keys that have been used since C1, and the BKE keys that have not been used yet.

As of January 2008, and at the latest six weeks before the assigned C2 deadline

As of 2008, the customer completes the form on www.swift.com > Ordering & Support to confirm its readiness to perform live RMA:

1. The customer confirms that they have completed the RMA testing (including at least the minimum test scenarios as documented on www.swift.com/swiftnetphase2).
2. The customer confirms that they have operators trained to perform RMA.
3. The customer confirms that they understand the rules of RMA before and after T2, and are ready to meet their obligations.

Processing of RMA readiness form

Once the customer has submitted the RMA readiness form, SWIFT will activate the customer for live RMA.

At latest, at the assigned C2 deadline

The customer must demonstrate its ability to perform live RMA. For each of its live BIC8s, the customer must send a live RMA authorisation to itself and receive a live RMA message from itself. SWIFT will monitor this. To ensure that the whole community is ready for RMA before T2, SWIFT will apply penalties to customers who have not demonstrated on time their readiness for live RMA. For more information, see "Pricing" on page 58.

Between the assigned C2 deadline and T2

1. The customer must maintain live operational readiness. To demonstrate this, the customer must send and receive a live RMA authorisation message to and from itself (once for each live BIC8) at least once per month. To encourage users to repeat this demonstration of readiness, SWIFT will accord a financial incentive. For more information, see "Pricing" on page 58.
2. The customer can (if wanted) repeat tests in Test and Training (T&T) of relationship management operational procedures.
3. All customers continue to use BKE as the primary method to maintain correspondent relationships BKE renewal continues, until T2.

Warning Before T2, customers must **not** send live RMA messages to correspondents with whom they have a BKE key. Before T2, no party can expect another party to act on live RMA messages.

4. As an exception to the above, the customer can use RMA to establish relationships with *new* correspondents, with which they do not have a BKE key. The customer can do this only if both parties agree and if both are sure they will not fall back to Phase 1 mode. In the case of SWIFTNet FINCopy, all three parties must agree. If one of the parties still wants to use BKE, then the other party cannot refuse to do so.

Warning Customers that establish a new relationship with RMA only, cannot fall back to BKE for this specific relationship. In particular, if one of the parties falls back to Phase 1 mode, then this party can no longer send authenticated FIN messages to the other party (for lack of a BKE key). This party will also not be able to receive authenticated messages from the other party, as these will be aborted, and the sender receives an MT 019 Abort Notification.

At T2 (planned for 27 September 2008)

The customer must stop the automatic BKE process, at the latest, before the last quit and log off from SWIFTNet FIN just before T2. Stopping the BKE process avoids the unnecessary queuing of BKE messages (for transmission) by the SWIFTNet FIN interface. These queued BKE messages would then be rejected (NAKed) by the SWIFTNet FIN systems, when the customer performs a Login and Select to SWIFTNet FIN after T2.

After T2, SWIFT will reject (NAK) BKE messages (MT96n messages). Customers can no longer use BKE to establish or maintain relationships. Customers cannot renew BKE keys, although existing BKE keys are still valid. SWIFTNet FIN interfaces can continue to send dual-signed messages (with MAC or PAC signatures, plus a PKI signature).

Important At T2, all existing BKE keys remain valid. Future BKE keys become active at their foreseen date. No new (future) keys can be established after T2, but since BKE keys do not have an expiration date, this has no impact.

After T2, decommission USE equipment

After T2, the customer must remove and destroy the Secure Card Reader (SCR) and Integrated Circuit Cards (ICC). SWIFT does not require the customer to return the card readers as they cannot be reused. In addition to checking internal requirements for auditing, the customer must also check the local legal requirements for the destruction of equipment, which may require the customer to document the process.

After T2, comply with RMA rules

After T2 all customers must comply to the RMA rules, see "Rules for Using RMA" on page 31:

1. After T2, the customer must read and empty its RMA store-and-forward queues every business day.
2. After T2, the customer must process incoming RMA messages (that is, authorisations, revocations, rejections, and queries) according to the RMA rules. This means that queries must be answered, and authorisations must be processed (accepted or rejected, and imported to the FIN interfaces) within six business days, and revocations must be processed within one business day. This includes the time to import the (revoked) authorisation to FIN interfaces.
3. In case the customer has central RMA management software, and distributes RMA records to FIN interfaces with an RMA distribution file, then the regular (at least daily) distribution to the FIN interfaces, must be started after T2.

Note All customers must comply with these obligations immediately after T2, regardless of whether they have already performed their RMA bootstrapping.

After T2 before C3, perform RMA bootstrapping

After T2, the process of RMA recording continues to record BIC8 pairs that send authenticated SWIFTNet FIN traffic to each other, until RMA bootstrapping is finalised:

1. The customer must perform the RMA bootstrap on its SWIFTNet FIN interfaces and its RMA management software.

This bootstrap process converts the BKE keys that have been used since C1 (for which the SWIFTNet FIN interface has recorded traffic), into RMA authorisations. For more information, see "The RMA Recording and Bootstrapping Process" on page 51.

2. The customer can use the bootstrap report to verify that the bootstrapped RMA records correspond to the BKE keys for which the customer expected an RMA authorisation.

Using the RMA management software, the customer can then treat exceptions, for example:

- Revoke unwanted authorisations-to-receive (which sends a revoke message to the correspondent).
- Delete unwanted authorisations-to-send (which sends a reject message to the correspondent).
- Issue authorisations for BKE relationships that are not bootstrapped. A BKE relationship may not be bootstrapped, because there was no record of incoming authenticated traffic for the relationship.

- Ask correspondents to issue authorisations-to-send that are missing after the bootstrap. Such authorisations may not be bootstrapped because the customer has never sent authenticated traffic to that correspondent. This can be asked with the RMA query/answer mechanism. Once the correspondent issues the authorisations-to-send, the customer can accept these authorisations on its RMA management software.

Note After T2, sending an RMA revocation or rejection to the correspondent guarantees that RMA removes the authorisation at the correspondent side, even if that correspondent has not bootstrapped yet and still has a BKE key. This is because the correspondent is obliged to treat incoming RMA messages, and an RMA record always supersedes the corresponding BKE key. Sending a BKE discontinuation message is not possible at this stage, since after T2, MT 96n messages can no longer be sent over SWIFTNet.

3. If the customer has multiple SWIFTNet FIN interfaces, and central RMA management software, then the customer must export a bootstrap file from each of its SWIFTNet FIN interfaces, and import these files into its central RMA management software.

Note Bootstrap RMA records never overwrite RMA records that a customer has created through an exchange of RMA messages with correspondents.

4. Normally, the customer performs the bootstrap only once. However, after bootstrapping, BKE keys for which no corresponding RMA record exists, can still be used, and the recording process continues. Thus, customers can continue to record for a while, and then repeat the bootstrap process.

At C3

After bootstrapping, a customer must instruct its SWIFTNet FIN interfaces to stop using the BKE keys, and to stop dual-signing. This also stops the RMA traffic recording. The BKE-to-RMA migration is then *finalised*.

From this point onwards, the customer only uses RMA to manage relationships. This completes the customer's live RMA cut-over.

T3

SWIFT declares T3, when all customers have reached C3. After T3, SWIFT rejects (NAKs) dual-signed FIN messages. SWIFT only accepts single-signed messages with PKI signatures. At this point, customers cannot fall back to BKE keys and the migration is complete.

5.4 The RMA Recording and Bootstrapping Process

Overview

SWIFT has designed a process that enables users to convert Bilateral Key Exchange (BKE) keys into RMA authorisations.

The conversion process has the following benefits:

- users do not have to exchange authorisations with correspondents with whom they already have a BKE key
- users can convert independently of any correspondents

- if users migrate BIC4 or BIC6 wildcarded BKE keys, then the process only creates authorisations for correspondent BIC8s with whom the customer has exchanged actual authenticated traffic
- the process cleans up the BKE file (that is, it does not convert unused BKE keys to authorisations)

The conversion method works in two stages: a FIN traffic recording stage, and an RMA bootstrap stage

FIN traffic recording (pre RMA-bootstrap)

For a limited period, each SWIFTNet FIN interface records the relationships that it uses for authenticated FIN traffic. All FIN interfaces do this recording independently of each other.

When does recording happen? Recording starts for a BIC8, when the FIN interface logs on with this BIC8 in Phase 2 mode (with FIN protocol version 3 and certificate on HSM), at or before C1. The recording only stops, when the customer finalises its migration to RMA, and the FIN interface stops using BKE keys, at C3 (see "Migration Activities" on page 48). This means that for most BIC8s, this recording lasts for more than one year. If the BIC8 temporarily falls back to using Phase 1 during this period, then recording temporarily stops, and resumes after the BIC8s log on again in Phase 2 mode. This does not delete the data recorded before the fallback.

What is recorded? The FIN interface records the BIC8 pairs (own BIC8 and correspondent BIC8) for which it sent or received authenticated SWIFTNet FIN messages. The direction of traffic (sent or received) is also recorded. The FIN interface does not create authorisations at this point, and the BKE keys remain in use to authenticate the messages. As the SWIFTNet FIN interface exchanges authenticated FIN messages with more and more correspondents, the list of BIC8 pairs that the SWIFTNet FIN interface has recorded grows. The user can use the migration status report on the interface to monitor the progress of this recording.

Note The SWIFTNet FIN interface records *authenticated* FIN traffic only (for example, MT 103s). The SWIFTNet FIN interface does not record the transmission or receipt of non-authenticated FIN traffic (for example, MT 999s), because non-authenticated FIN traffic does not require authorisations.

RMA bootstrap

After T2, the user performs the RMA bootstrap process, which converts the BKE keys into RMA authorisations, as follows:

1. RMA Bootstrap export

After T2, the user instructs each of its FIN interfaces to perform the RMA bootstrap export. This creates an RMA bootstrap file at each FIN interface. A bootstrap file, contains a bootstrapped authorisation for each of the BIC8 pairs for which a valid BKE key is present at the FIN interface bootstrap time, and for which the FIN interface has recorded authenticated traffic. The bootstrap file is in XML format (see Appendix A, "RMA Distribution File Format" on page 60). The bootstrap export also creates a bootstrap report, which lists, in human-readable format, the content of the bootstrap file.

2. RMA Bootstrap import

After the RMA bootstrap export, the user transfers these RMA bootstrap files from its FIN interfaces to the RMA management software (this step is not needed if the RMA management is on the same system as the FIN interface). The user then instructs the RMA management software to import these bootstrap files. The bootstrap import creates the RMA records: one for each of the authorisations in the bootstrap file. If the RMA management software already has an RMA record for the same BIC8 pair, for example from a previous bootstrap import, or

because an authorisation was exchanged with that correspondent (or revoked, rejected or deleted), then the bootstrap never overwrites such a prior record.

Result of the conversion process

This conversion process creates a bootstrapped authorisation only if the following conditions are both satisfied:

- a valid BKE key exists for the BIC8 pair *at bootstrap time*
- the FIN interface has recorded authenticated traffic for the BIC8 pair before bootstrapping

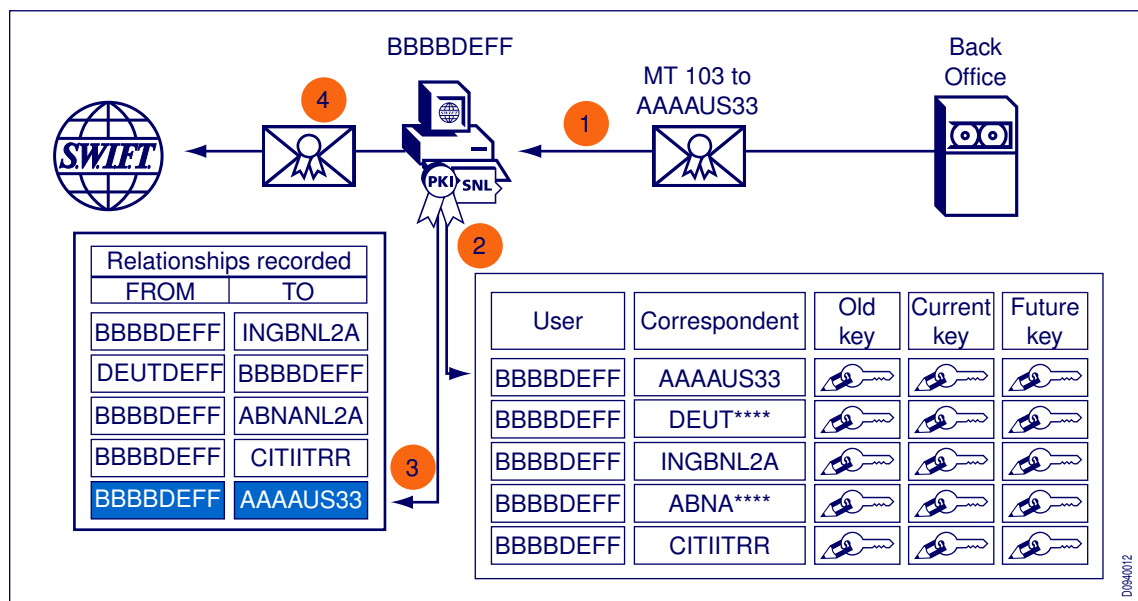
In other words, only a subset of the BKE keys and only a subset of BIC8 pairs for which SWIFTNet FIN has recorded traffic are bootstrapped into authorisations:

- For wildcarded (shared, BIC4 or BIC6) BKE keys, RMA bootstraps into authorisations only matching BIC8 pairs for which SWIFTNet FIN has observed actual authenticated traffic.
- If BKE keys are deleted, discontinued, or excluded after the traffic was recorded, then no bootstrapped authorisations are created.

FIN traffic recording when sending

This section provides details about how traffic recording works, when SWIFTNet FIN sends an authenticated FIN message.

Data recorded for a sent message



1. The user instructs the SWIFTNet FIN interface to send an authenticated message to AAAAUS33.
2. The SWIFTNet FIN interface checks for a valid Bilateral Key Exchange (BKE) send key for AAAAUS33.
3. The SWIFTNet FIN interface records that there was authenticated traffic for that BIC8 pair in that direction, but does not yet, create an authorisation.
4. The SWIFTNet FIN interface sends the signed message and attaches a Message Authentication Code (MAC) trailer.

During sending, a recording only occurs in the following circumstances:

- if the sending Logical Terminal (LT) is logged on with FIN protocol version 3 (that is, it can sign FIN messages with certificates on HSM)
- if the user has not yet finalised bootstrapping on this FIN interface
- if the FIN interface can successfully authenticate the message

In other words, the SWIFTNet FIN interface makes a recording only if there is a valid and current BKE send key present on the SWIFTNet FIN interface for this BIC8 pair.

When sending, the SWIFTNet FIN interface does not record in the following circumstances:

- there are no BKE keys for this BIC8 pair
- there is only a previous key or future key, or the current key is discontinued
- the user has flagged the bilateral key relationship as *suspended* or *excluded*

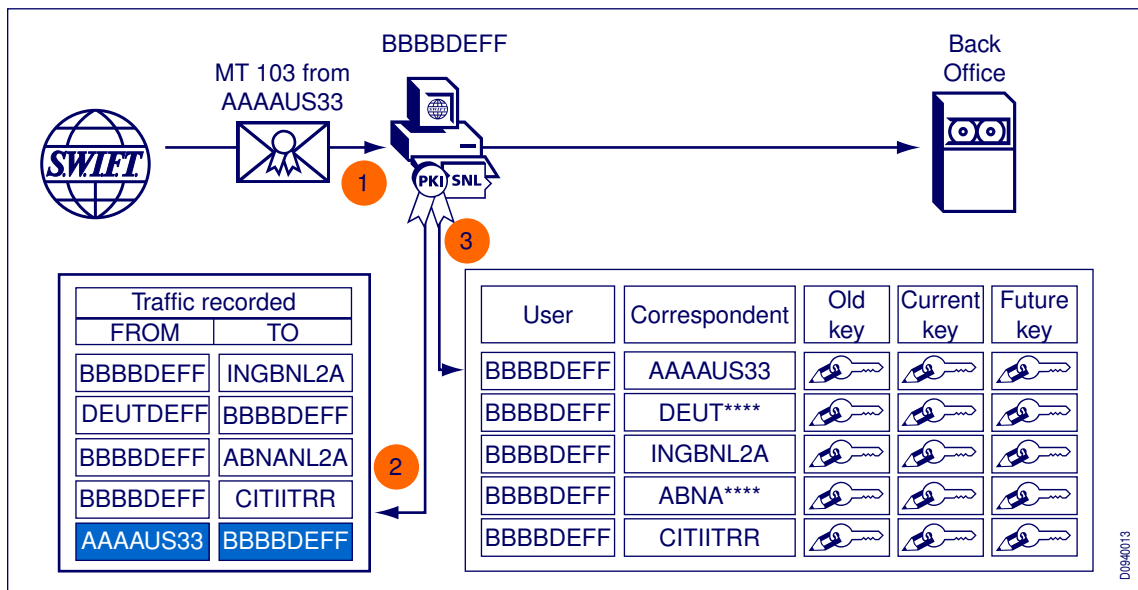
The reason for this is that the SWIFTNet FIN interface cannot authenticate the message. Therefore the user cannot send the message.

If the SWIFTNet FIN interface makes a recording, then the recording remains in place, regardless what happens to the message afterwards. For example, the recording remains on file, even if the message is subsequently negatively acknowledged (NAKed) by SWIFTNet FIN, or is aborted later (MT 019), or receives a non-delivery warning (MT 010).

Traffic recording when receiving

This section provides details about how traffic recording works at the FIN interface, when it receives an authenticated FIN message.

Data recorded for a received message



1. The SWIFTNet FIN interface receives an authenticated message from AAAAUS33.
2. The SWIFTNet FIN interface records that there was authenticated traffic for that BIC8 pair in that direction, but does not yet create an authorisation.

3. The SWIFTNet FIN interface checks the message's authentication against the BKE key. If authentication fails, then the SWIFTNet FIN interface flags the message, and may place it in an investigation queue.

During receipt, the SWIFTNet FIN interface makes a recording only when the receiving LT is logged on with FIN protocol version 3 (that is, it can sign FIN messages with certificates on HSM) and if the user has not finalised bootstrapping on this FIN interface (before C3, see section "Migration Activities" on page 48). Under these conditions, SWIFTNet FIN makes a recording for a received, authenticated FIN message regardless of whether the message succeeds or fails the authentication check at reception (that is, regardless of whether there is a valid, current BKE key). However, only recordings for which there is a valid BKE key at RMA bootstrap time, lead to bootstrapped authorisations. If there is (still) no valid BKE key at RMA bootstrap time, then RMA does not create a bootstrapped authorisation from the recording.

The rationale behind not using the BKE key as a condition for recording at receipt, is that the receipt of an authenticated FIN message is proof that at least the sender of the message has a valid, current BKE key to send to the user's BIC8. If the message fails authentication on receipt, then the receiving SWIFTNet FIN interface cannot know whether this indicates a problem at the sender or at the receiver. For this reason, the message requires flagging (recording) and manual investigation. The user knows of such recordings by looking at the migration status report of its FIN interface, which reports the recordings that took place, including if they have a valid BKE key or not.

This manual investigation has two possible outcomes, there can either be a relationship (and a key), or not, as follows:

- If a receive recording does not have a corresponding bilateral key, and both user and correspondent believe there should be a relationship, then the parties must establish a valid BKE key to avoid further BKE authentication failures. Once the valid BKE key is present, then RMA can create a bootstrap authorisation because there is a valid BKE key and a recording. If the user detects the lack of a key after T2, then the parties must exchange an authorisation rather than a BKE key (either before or after bootstrapping). Users cannot exchange BKE keys after T2.

Note: Before T2, users must not exchange authorisations to resolve the problem of a missing BKE key.

- If a receive recording does not have a corresponding bilateral key, and the receiver does not want a relationship, then the best course of action before T2 is for the receiver to ask the correspondent to delete or discontinue its BKE key. Otherwise, the correspondent can continue to send authenticated SWIFTNet FIN traffic that will fail authentication at the receiver. Deleting or discontinuing the BKE key before bootstrapping avoids RMA creating a bootstrapped **authorisation-to-send** at the sender's side. In any case, RMA does not create a bootstrapped authorisation at the receiver's side because no valid BKE key exists at the receiver.

In the previous scenario, there is no guarantee that the correspondent will delete or discontinue the BKE send key (this is a known shortcoming of BKE). Therefore, after bootstrapping, the correspondent may have an **authorisation-to-send** to the receiver, while the receiver does not have a corresponding **authorisation-to-receive**. For this reason, the BIC8 pairs for which authenticated traffic was received without a valid BKE receive key, shows up in the migration status report of the FIN interface. After T2, the receiver should check the migration status report, and send an RMA revocation to such correspondents, to ensure the revocation of any unwanted bootstrapped authorisations-to-send.

Note: Users can always send an RMA revocation to a correspondent, even if the receiver has never issued an authorisation to that correspondent, and has no (bootstrapped) **authorisation-to-receive** for that correspondent.

To conclude, FIN makes a recording on reception, even in the following circumstances:

- there are no BKE keys for a BIC8 pair
- there is only a previous key or only a future key
- the keys are discontinued
- the user has flagged the bilateral key relationship as Suspended or Excluded

At bootstrap time, RMA does not convert such recordings into authorisations, because no valid BKE key exists.

RMA bootstrap export

The bootstrap file has the format of an RMA distribution file. This is an XML format (see Appendix A, "RMA Distribution File Format" on page 60).

A bootstrap file is always a partial file, not a complete file, and contains only bootstrapped authorisations. Authorisations that users exchanged over SWIFTNet with RMA messages (for example, with new correspondents) are not present in the bootstrap file.

Bootstrapped authorisations never have a validity period (the From and To dates are not present).

A distinguishing characteristic of bootstrapped authorisations is that they do not have either a DateTimelssued field or a Signature field. By contrast, authorisations that users exchange over SWIFTNet always have a DateTimelssued field and a Signature field that corresponds to the PKI signature of the authorisation message.

The bootstrap file is not encrypted, but the SWIFTNet FIN interface does sign it with a Local AUthentication (LAU) signature. The LAU algorithm calculates this signature over the whole file, using a user-defined LAU secret key. The LAU key is a 128-bit key, which is (typically) written as 32 hexadecimal characters. The LAU key is a shared secret between the SWIFTNet FIN interface and the RMA management software. The user must enter the LAU key, both at the FIN interface and at the RMA software.

RMA bootstrap import

Before it imports the bootstrap file, the RMA management software, verifies the LAU signature. If there is a mismatch, then RMA does not import the file. In this way, RMA provides a protection against corruption of, or tampering with, the file when in transit from the SWIFTNet FIN interface to the RMA management software. RMA also provides a guarantee of authenticity of the file, since only the SWIFTNet FIN interface can calculate the LAU signature with the secret LAU key.

For each of the bootstrapped authorisations in the file, RMA creates a record in the RMA data store. The exception is if there is already an RMA record, which RMA created either through a previous bootstrap import or through an exchange of RMA messages over SWIFTNet. Therefore, authorisations that RMA creates through an exchange of RMA messages over SWIFTNet always take precedence over bootstrapped authorisations. This principle also applies to the revoked, rejected, or deleted authorisations.

Users must repeat the import for all the bootstrap files, from each of the SWIFTNet FIN interfaces that performed traffic recording. Also, if a user exchanges RMA authorisations for some of its own BIC8s on other systems (for example, on SWIFTNet FIN interfaces that are, or were, capable of an exchange of RMA authorisations), then the user must take the following actions:

- export the RMA authorisations from these systems in a *partial* RMA distribution file
- import the RMA authorisations into the RMA data store, into which RMA imports the bootstrapped authorisations for these BICs

Note Exchanged authorisations always supersede an equivalent bootstrapped authorisation.

Once the user has exchanged authorisations, and has built a complete RMA data store, then the user must treat exceptions as follows:

- revoke unwanted authorisations-to-receive
- delete unwanted authorisations-to-send
- issue authorisations for the missing authorisations-to-receive
- ask correspondents to issue missing authorisations-to-send

When the user has treated the exceptions, it can export the complete RMA distribution file, and import it into the SWIFTNet FIN interfaces that require the authorisations. Such export and import of authorisations is only necessary if the user's RMA management software is separate from its SWIFTNet FIN interface, without a real-time link. From this moment, the RMA authorisations supersede the matching BKE keys on the SWIFTNet FIN interface.

Note The import step is not reversible: once the user imports RMA authorisations, it cannot un-import them.

6 Pricing

Yearly RMA service fee

A yearly RMA service fee of EUR 400 for each live BIC8 becomes applicable when RMA becomes mandatory (as of 1 October 2008). The service fee covers the use of SWIFTNet RMA across all SWIFTNet services (that is, SWIFTNet FIN and other SWIFTNet services that adopt SWIFTNet RMA in the future). The service fee reflects the value of the service to a given BIC8. The value of the service is its ability to protect and manage correspondent relationships and to ensure a secure environment.

In 2008, the yearly RMA service fee is only applicable in the last quarter (after 1 October 2008), and will thus be charged pro-rata (that is, one fourth of the year). Thus, in 2008 the yearly RMA service fee will amount to EUR 100 per BIC8. However, see also the section on Migration Incentives at the end of this chapter.

RMA message price

RMA messages are only necessary to establish, change, or revoke a relationship. Bilateral Key Exchange (BKE) requires key renewals every 6 to 12 months. Migration to RMA automatically converts existing BKE records into RMA records. Because migration does not involve an exchange of RMA messages, SWIFT can offer the conversion free of charge. Due to the automatic conversion of BKE records, SWIFT expects a very low quantity of RMA messages in view of the migration. The RMA message price reflects the value of the underlying action and aims to prevent misuse.

RMA messaging fees

Message	Price per message	Charged to
Authorisation	EUR 5	Sender
Revocation	EUR 5	Sender
Reject	EUR 5	Receiver ⁽¹⁾
Query	EUR 0.5	Sender
Answer	EUR 0.5	Sender

(1) For an **authorisation-to-send**, the issuer pays for all of the RMA messages. For example, the authorisation itself, any reject messages related to that authorisation (even if the reject comes later as part of a delete on the initiative of the correspondent), and the revoke message.

RMA cost compared to BKE

The fee structure design correlates closely with current spending on BKE per institution. In most circumstances, a customer will pay the same for RMA as it currently pays for BKE, on a yearly basis.

Financial penalties

As in previous migrations, penalties encourage the community to adhere to migration timing.

Every customer knows its C2 deadline. This C2 deadline is the latest date by when the customer must demonstrate its readiness for live RMA. For more information about C2, see "BKE-to-RMA Migration" on page 46. All C2 deadlines are in the second quarter of 2008. The third quarter of 2008 is a contingency period, where penalties apply. The following penalties apply to users (BIC8s) that have not demonstrated their ability to use live RMA by sending and receiving a live RMA authorisation message.

Penalty

July 2008	EUR 5,000 per BIC8
August 2008	EUR 10,000 per BIC8
September 2008	EUR 20,000 per BIC8

Migration incentives

To encourage users to adopt SWIFTNet RMA, including granular authorisations, SWIFT will waive the RMA message charges in 2008. After T2, SWIFT will continue to waive the RMA message charges for an additional three months. Thus, the RMA messages will only be charged as of 2009.

To encourage users to repeatedly send an RMA authorisation to themselves and receive an RMA authorisation from themselves, SWIFT will also waive the yearly RMA service fee for the last quarter of 2008, on condition that the user has repeatedly demonstrated its live RMA readiness before T2. More precisely, this fee is waived for the last quarter 2008, if all of the following criteria are met:

1. the user (BIC8) has successfully demonstrated its ability to use the Live RMA service by sending an RMA authorisation to itself and receiving an RMA authorisation from itself before 30 June 2008.
2. the user (BIC8) repeats this Live RMA authorisation to itself and from itself in the course of July 2008.

Appendix A

RMA Distribution File Format

A.1 Overall File Format

Introduction

The distribution file contains a set of RMA data store records and a header record that describes the distribution file.

The Distribution file is one XML document with the following structure:

```
<Sw:RMAFile>
  <Sw:RMAFileHdr>
    (<Sw:RMARecrd>)+
</Sw:RMAFile>
```

The root element contains the following namespace attributes:

```
xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:Doc="urn:swift:snl:ns.Doc"
xmlns:SwSec="urn:swift:snl:ns.SwSec"
```

A.2 RMAFileHeader

Structure

The structure of the RMAFileHeader is as follows (where "?" denotes zero or one occurrence, and "+" denotes one or more occurrences):

```
<Sw:RMAFileHdr>
  <Sw:Bic8Lst>
    (<Doc:Bic8>)+
  </Sw:Bic8Lst>
  <Sw:SvcLst>
    (<Doc:SvcNm>)+
  </Sw:SvcLst>
  <Sw:FileMaintncSts>
  <Sw:FileDesc>
  <Sw:CrDtTm>
  <Sw:TltRecrd>
  <Sw:LAU>
    <Sw:LAUVal>
      (<Sw:LAUAlgo>)?
    </Sw:LAU>
</Sw:RMAFileHdr>
```

The following table explains the elements of the RMAFileHeader:

Element	Value
Bic8Lst	Contains a list of BIC8s.
Bic8	This is the BIC8 of the institution that is either the issuer for authorisations issued or the correspondent for authorisations received.
SvcLst	Contains a list of services.
SvcNm	The name of the SWIFTNet business service (live, pilot, or integration testbed [ITB]) for which the authorisation applies (for example, <i>swift.fin</i>).

Element	Value
FileMaintncSts	Contains the word <i>complete</i> in a complete distribution file or <i>partial</i> if the distribution file is partial.
FileDesc	Free-format description of the distribution file. Typically, the description gives more information about the content of the distribution file.
CrDtTm	YYYY-MM-DDTHH:MM:SSZ Date and time of creation of the distribution file.
TltRecrd	Total number of records on the file. This is the number of Sw:RMARecord within the Sw:RMAFile.
LAU	Contains the elements that relate to local authentication.
LAUVal	The result of the local authentication.
LAUAlgo	Only present if the default algorithm is not used. This element is currently not used.

Note Because the LAU key is a shared key, users must enter it. The key is currently 128 bits long. SWIFT recommends that users communicate and enter the key as 32-hex digits (so that users can easily share keys between heterogeneous systems).

A.3 RMARecord

Structure

The structure of the RMARecord is as follows:

```

<Sw:RMARecrd>
  <Sw:Tp>
  <Sw:RMASts>
  <Doc:Issr>
  <Doc:Crspdt>
  <Doc:SvcNm>
  <Doc:IssdDtTm>?
  (<Doc:VldtyPrd>
    <Doc:FrDt>
    <Doc:ToDt>
  </Doc:VldtyPrd>)?
  (<Doc:Permsn>)?
  (<Doc:FINSvcPermsn>)?
  (<SwSec:Signature>)?
</Sw:RMARecrd>

```

The following table explains the elements of an RMARecord:

Element	Value
Tp	Issued or received.
RMASts	The status of the authorisation record, which can be enabled, rejected, revoked, or deleted.
Issr	The BIC-8 of the Issuer of the authorisation.
Crspdt	The BIC-8 of the Correspondent of the authorisation.
SvcNm	The SWIFTNet business service (live, pilot, or integrated testbed [ITB]) for which the authorisation applies (for example, <i>swift.fin</i>).
IssdDtTm	YYYY-MM-DDTHH:MM:SSZ Date and time of creation of the authorisation by the Issuer.

Element	Value
VldtyPrd	Contains the elements of the ValidityPeriod.
FrDt	YYYY-MM-DD Start date of the validity period of the authorisation.
ToDt	YYYY-MM-DD End date of the validity period of the authorisation.
Permsn	Permissions on the RequestType. This element is optional and is never present if the FINSvcPermsn element is present.
FINSvcPermsn	The element FINSvcPermsn is for <i>swift.fin</i> . This element is not present if the Permsn element is present.
Signature	The Signature of the Request.

Appendix B

RMA Reject Reasons

B.1 Reject Codes and Definitions

Examples

xrma.003.1	<p>No business relation exists.</p> <p>This reject reason indicates that an authorisation is rejected because the business relation does not exist.</p>
xrma.003.2	<p>DateTimeIssued too far in the future.</p> <p>This reject reason indicates that when the issuer created the authorisation, the issuer's clock was set to an unacceptable time.</p>
xrma.003.3	<p>Unexpected permissions.</p> <p>The permissions are not in line with the business relationship. SWIFT recommends using the free text to indicate clearly the expectations.</p>
xrma.003.4	<p>Record deleted.</p> <p>This reject reason indicates that the authorisation was deleted after having been accepted.</p>
xrma.003.5	<p>Insufficient.</p> <p>This reject reason indicates that the authorisation was acceptable except for the validity period.</p>
xrma.003.6	<p>No longer valid authorisation digest.</p> <p>This reject reason is used when RMA detects an error in the digest. A digest error may indicate a security violation at either the issuer or correspondent side.</p>
xrma.003.7	<p>Duplicate value in permissions.</p> <p>Duplicate value found in Excl or Incl list.</p>
xrma.003.99	<p>Undefined.</p> <p>This reject reason covers all other cases. In this case, the correspondent must provide more information in the free text.</p>

Appendix C

Background Information for the Reader

C.1 Secure Login and Select

SWIFT makes the Secure Login and Select (SLS) financial messaging service (collectively known as SWIFTNet FIN) available to users through the following applications:

- The General Purpose Application (GPA) provides access to various system-related messages and functions, and controls access to the FIN application.
- The financial messaging application (FIN) provides user-to-user message processing facilities, and some system-related messages.

Access to GPA is by means of the Login command. Access to FIN is by means of the Select command. SWIFT authenticates the Login and Select commands for the following reasons:

- to ensure that only valid Logical Terminals (LTs) access the system
- to ensure that, from among the LTs that access the GPA, only LTs that are authorised to use FIN can successfully select FIN

The Login and Select messages that users send to SWIFT contain an authentication code. The authentication code derives from a session key that is unique for every GPA and FIN session. The secret information that enables the interface to generate these session keys is stored on the user Integrated Circuit Cards (ICCs). Users can use a card reader to read ICCs. The user can use either the Basic Card Reader (BCR) or the Secure Card Reader (SCR). The SCR offers more processing capabilities (for example, it allows Bilateral Key Exchange [BKE]).

SWIFT has the same secret information, and verifies that it is connected to an authorised LT when it receives the Login and Select command. SWIFT returns acknowledgements to the customer. The acknowledgements contain authentication information that allow the receiving LT to verify that it is connected to SWIFT. SWIFT performs such checks at each Login and Select command.

SLS is the only way to connect to SWIFTNet FIN. To use SLS to access the SWIFTNet FIN service, each user must have activated an available ICC set. The user must also have configured the user ICCs of that set for live usage.

C.2 Bilateral Key Exchange (BKE)

Before sending authenticated messages to SWIFTNet FIN counterparts, users must exchange bilateral keys. SWIFT protects the exchange with an asymmetric algorithm that uses a private key that never leaves a hardware module at the customer site. SWIFT renews the private key at predefined intervals.

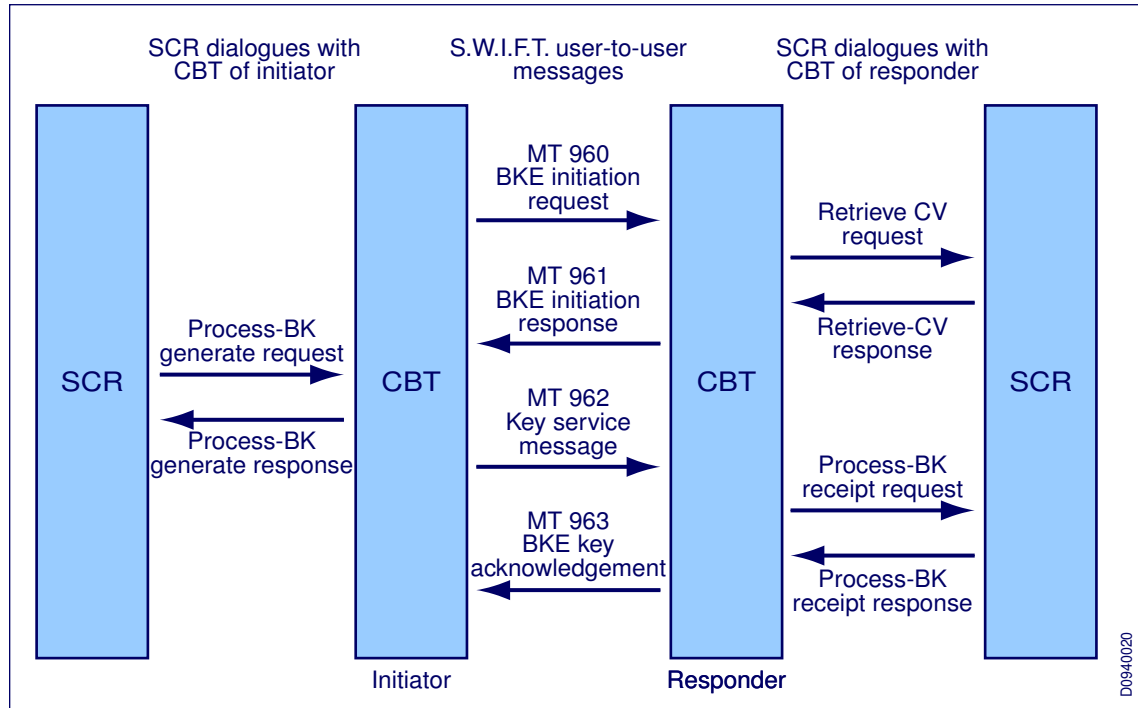
The users agree on the following parameters of the bilateral key exchange:

- the identity of the Key Management Authority (KMA) (for example, the KMA that actually exchanges the bilateral keys)
- the identity (or identities) of one or more key users, based on BIC8, BIC6, or BIC4 (depending on sharing arrangements)
- the key type (uni-directional or bi-directional)

- the party that initiates the key exchange
- the timing of exchanges

The parties then use the MT 96x series of messages, as the following illustration shows, to exchange bilateral keys.

The BKE 4-message exchange



For messages that require authentication, the sender uses the text of the message, and the relevant symmetric bilateral key, to calculate a Message Authentication Code (MAC). The user appends the MAC to the message and forwards it to the receiver. The receiver uses the same symmetric bilateral key to verify the MAC.

C.3 SWIFTNet Naming and Addressing

The SWIFTNet Naming and Addressing scheme provides identification rules for the following items:

- The addresses of senders and receivers of SWIFTNet InterAct and SWIFTNet FileAct messages. Typically, addresses include institution (BICs) departments, lines-of-business, and commercial services.
- The PKI certificates used for signing the messages. PKI certificates include the identity of the certificate's owners (for example, institutions [BICs], interfaces, applications, or individuals). In SWIFTNet FIN the PKI certificates identify the Sending Institution (BIC) and the sending SWIFTNet FIN CBT.

The scheme format follows the X.500 standard, which uses Distinguished Names (DNs) to identify both the addresses and PKI certificates. DN's are segmented, and have a hierarchical structure. DN's can therefore identify both high-level and granular entities.

The two standard levels for SWIFTNet DNs are as follows:

- the root level: `o=swift`
- the organisation level: `o=bbbbcc11`, which SWIFT assigns. The organisation level contains the SWIFT user's BIC. The user's BIC provides continuity with current SWIFTNet FIN naming and enables re-use of existing registration and publication infrastructures.

The subsequent sub-levels of SWIFTNet DNs are as follows:

- the organisational unit level (optional) as follows:
 - `ou=london`, which reflects a geographical location or region
 - `ou=fin`, which reflects a service
- the user or application level:
 - `cn=fincbt1`, which reflects a SWIFTNet FIN CBT or application
 - `cn=john-smith`, which reflects a person

For more information about naming and addressing, see the *SWIFTNet Naming and Addressing Guide*.

C.4 SWIFTNet FIN Addressing

SWIFTNet FIN addresses identify the FIN-bridges, because that is where all users send SWIFTNet FIN messages inside SWIFTNet InterAct envelopes.

The address DN to which users send SWIFTNet FIN traffic is as follows:

- `cn=finb01,o=swift,o=swift`

The address DN is automatically configured in the Computer-Based Terminal (CBT). Configuration occurs after the address DN has used Secure Login and Select (SLS) to establish the session with SWIFT. Security Officers (SOs) do configure the CBT.

C.5 SWIFTNet PKI

SWIFT offers a Public Key Infrastructure (PKI) service to enable SOs to issue the digital certificates that institutions use in the messaging layer. The user bases its trust in the identity of the correspondent, and in the integrity of the message, on the digital signatures that SWIFTNet PKI provides.

SWIFT users appoint SOs to administer certificates. SWIFT provides the SOs with an online utility to create the certificate DNs in the SWIFTNet directory.

For SWIFTNet FIN, the SO must create the PKI certificates for its FIN CBTs. The certificate DNs are as follows:

- `cn=fincbt1,o=bbbbcc11,o=swift` for a small institution
- `cn=fincbt1,ou=fin,o=bbbbcc11,o=swift` for a large institution that has many certificates that it must group into sub-hierarchies for easy management

Note Within the PKI certificates, the DN sub-levels beneath the BIC only serve to make local management of certificates easier. In principle, these sub-levels do not have any meaning for the correspondent.

The following illustration shows the hierarchy of PKI certificate DNs as it applies to a large user.

Hierarchy of PKI certificates

