ICS 35.030
CCS L 80

中华人民共和国国家标准

GB/T 17903.2—2021
代替 GB/T 17903.2—2008

信息技术　安全技术　抗抵赖
第2部分：采用对称技术的机制

Information technology—Security techniques—Non-repudiation—Part 2：Mechanisms using symmetric techniques

（ISO/IEC 13888-2：2010 MOD）

2021-10-11 发布　　2022-05-01 实施

国家市场监督管理总局
国家标准化管理委员会　发布

目　　次

前言 …… Ⅰ
引言 …… Ⅱ
1　范围 …… 1
2　规范性引用文件 …… 1
3　术语和定义 …… 1
4　缩略语 …… 2
5　符号 …… 2
6　通用要求 …… 3
7　安全信封 …… 3
8　抗抵赖令牌的生成与验证 …… 3
8.1　TTP 创建令牌 …… 3
8.2　抗抵赖机制使用的数据项 …… 4
8.3　抗抵赖令牌 …… 4
8.4　TTP 进行的令牌验证 …… 5
9　特定抗抵赖机制 …… 6
9.1　抗抵赖机制 …… 6
9.2　原发抗抵赖机制 …… 6
9.3　交付抗抵赖机制 …… 8
9.4　获取时间戳令牌的机制 …… 9
附录 A（资料性）　抗抵赖机制实例 …… 10
A.1　原发抗抵赖与交付抗抵赖机制实例 …… 10
A.2　机制 M1：必选 NRO，可选 NRD …… 10
A.3　机制 M2：必选 NRO，必选 NRD …… 12
A.4　机制 M3：带有中介 TTP 的必选 NRO 和必选 NRD …… 13
参考文献 …… 15

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是GB/T 17903《信息技术　安全技术　抗抵赖》的第2部分。GB/T 17903已经发布了以下部分：

——第1部分：概述；

——第2部分：采用对称技术的机制；

——第3部分：采用非对称技术的机制。

本文件代替GB/T 17903.2—2008《信息技术　安全技术　抗抵赖　第2部分：采用对称技术的机制》。与GB/T 17903.2—2008相比，除结构调整和编辑性改动外，主要技术变化如下：

a) 增加了第5章 符号；

b) 删除了原第6章 本文件各章的组织；

c) 增加了9.1 抗抵赖机制；

d) 增加了图1和图2；

e) 删除了原9.3和9.4中的技术内容；

f) 将原第10章的技术内容移至附录A中。

本文件使用重新起草法修改采用ISO/IEC 13888-2:2010《信息技术　安全技术　抗抵赖　第2部分：采用对称技术的机制》。

本文件与ISO/IEC 13888-2:2010的技术性差异及其原因如下：

——关于规范性引用文件，本文件做了具有技术性差异的调整，以适应我国的技术条件，调整的情况集中反映在第2章“规范性引用文件”中，具体调整如下：

- 用等同采用国际标准的GB/T 17903.1代替了ISO/IEC 13888-1。
- 删除了规范性引用文件ISO/IEC 9798-1:1997和ISO/IEC 10118(所有部分)，这些规范性引用文件仅在术语中作为来源引用。
- 增加了规范性引用文件GB/T 15852，GB/T 15852规定了消息鉴别码算法，是本文件中应用的重要密码算法。
- 增加了规范性引用文件GB/T 20520和GB/T 25069。

——增加了图1和图2，以帮助理解第9章的技术内容。

——删除了部分与GB/T 25069重复的通用术语，改为在第3章引用GB/T 25069的形式。

本文件做了下列编辑性修改：

——纳入了ISO/IEC 13888-2:2010/Cor.1:2012的技术勘误，所涉及的条款的外侧页边空白位置用垂直双线(‖)进行了标示。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本文件起草单位：中国科学院软件研究所、网神信息技术(北京)股份有限公司、中国科学院数据与通信保护研究教育中心、联想(北京)有限公司、上海格尔软件股份有限公司。

本文件主要起草人：张严、张振峰、张立武、黄亮、李敏、李汝鑫、郑强、李俊、蔡冉。

本文件及其所代替文件的历次版本发布情况为：

——2008年首次发布为GB/T 17903.2—2008；

——本次为第一次修订。

引　言

GB/T 17903 旨在对抗抵赖服务的通用模型以及特定的抗抵赖机制进行规范，抗抵赖服务通过生成、收集、维护、利用和验证有关已声称事件或动作的证据，以解决有关该事件或动作已发生或未发生的争议，由三个部分构成。

——第 1 部分：概述。目的在于规定抗抵赖服务的通用模型。

——第 2 部分：采用对称技术的机制。目的在于规定若干特定的、采用对称技术的抗抵赖机制。

——第 3 部分：采用非对称技术的机制。目的在于规定若干特定的、采用非对称技术的抗抵赖机制。

信息技术　安全技术　抗抵赖
第2部分：采用对称技术的机制

1　范围

本文件确立了抗抵赖服务的通用结构，以及若干特定的抗抵赖机制，用于提供原发抗抵赖（NRO）与交付抗抵赖（NRD）。

本文件适用于采用对称技术实现的消息抗抵赖相关应用的设计、实现与测试。

2　规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 15852（所有部分）　信息技术　安全技术　消息鉴别码[ISO/IEC 9797（所有部分）]

GB/T 17903.1　信息技术　安全技术　抗抵赖　第1部分：概述（GB/T 17903.1—2008，ISO/IEC 13888-1：2004，IDT）

GB/T 20520　信息安全技术　公钥基础设施　时间戳规范

GB/T 25069　信息安全技术　术语

3　术语和定义

GB/T 17903.1 和 GB/T 25069 界定的以及下列术语和定义适用于本文件。

3.1

密码校验函数　cryptographic check function

以秘密密钥和任意字符串作为输入，并以密码校验值作为输出的密码变换。不知道秘密密钥就不可能正确计算校验值。

3.2

数据完整性　data integrity

数据没有遭受以未授权方式所作的更改或破坏的特性。

3.3

证据生成者　evidence generator

产生抗抵赖证据的实体。

3.4

杂凑函数　hash-function

将任意长比特串映射为定长比特串的函数，满足如下属性：

——给定一个输出比特串，寻找一个输入比特串来产生这个输出比特串，在计算上是不可行的；

——给定一个输入比特串，寻找另一个不同的输入比特串来产生相同的输出比特串，在计算上是不可行的。

注1：计算上的可行性取决于特定安全要求和环境。

注2：在本文件中，杂凑函数的输出的比特串称为杂凑码。

3.5

消息鉴别码 message authentication code；MAC

消息鉴别码算法的输出的比特串。

3.6

消息鉴别码算法 MAC algorithm

一种带密钥的密码算法，用于将比特串和秘密密钥映射为定长比特串的函数，并满足以下两种性质：

——对任意密钥和任意输入串，该函数都能有效进行计算；

——对任一固定的密钥，该密钥在未知情况下，即便已知输入串集合中的第 i 个输入串和对应的函数值，且串集合中的第 i 个输入串值在观测前面的第 $i-1$ 个函数值之后可能已经选定，要算出该函数对任意新输入串的值在计算上是不可行的。

3.7

时间戳 time-stamp

对时间和其他待签名数据进行签名得到的数据，用于表明数据的时间属性。

3.8

时间戳机构 time-stamp authority

用来产生和管理时间戳的可信服务机构。

4 缩略语

下列缩略语适用于本文件。

DA：交付机构(Delivery Authority)

GNRT：通用抗抵赖令牌(Generic Non-Repudiation Token)

NRT：抗抵赖令牌(Non-Repudiation Token)

NRD：交付抗抵赖(Non-Repudiation of Delivery)

NRDT：交付抗抵赖令牌(Non-Repudiation of Delivery Token)

NRO：原发抗抵赖(Non-Repudiation of Origin)

NROT：原发抗抵赖令牌(Non-Repudiation of Origin Token)

Pol：抗抵赖策略(Non-Repudiation policy)

PON：肯定或否定，验证过程的结果(Positive Or Negative)

SENV：安全信封(Secure ENVelope)

TSA：可信时间戳机构(trusted Time Stamp Authority)

TST：时间戳令牌(Time-Stamping Token)

TTP：可信第三方(Trusted Third Party)

5 符号

GB/T 17903.1 界定的以及下列符号适用于本文件。

a：仅为实体 A 和可信第三方(TTP)所知的密钥

b：仅为实体 B 和可信第三方(TTP)所知的密钥

da：交付机构(DA)的密钥

Imp(m)：消息 m 的印记，即 1)m 的杂凑码，或 2)m 本身

$MAC_X(y)$：使用实体 X 的秘密密钥对数据 y 计算得到的消息鉴别码

Pol：应用于证据的抗抵赖策略的可区分标识符

SENV(.)：用来计算安全信封的函数

TSA：TSA 的可区分标识符

ttp：仅为 TTP 所知的密钥，用于生成抗抵赖令牌

TTP：TTP 的可区分标识符

(y,z)：y 和 z 按顺序的连接

z_1：由提供 NRO 令牌的有关数据字段组成的数据字段

z_2：由提供 NRD 令牌的有关数据字段组成的数据字段

6 通用要求

本文件中，凡涉及密码算法的相关内容，按国家有关法规实施；凡涉及采用密码技术保护保密性、完整性、真实性、不可否认性的，应遵循密码国家标准和行业标准。

本文件中规定的抗抵赖机制均应满足以下通用要求：

——使用抗抵赖机制的各相关实体能够独立与 DA、TSA 或 TTP 进行通信；

——如果两个实体使用本文件中规定的某个抗抵赖机制，双方信任同一个第三方；

——在使用本文件规定的抗抵赖机制前，每个实体均与 DA、TSA 或 TTP 建立一个共享的对称密钥，每个 DA、TSA 或 TTP 实体均持有一个仅为自己所知的密钥；

注 1：密钥管理、密钥生成及密钥建立机制见 GB/T 17901.1[1] 等相关的国家标准或密码行业标准及 ISO/IEC 11770[3,1]。

——抗抵赖服务中的所有实体共享一个公共函数 Imp；

——为创建安全信封而选取的 MAC 函数为抗抵赖服务的所有参与者所持有；

——生成抗抵赖令牌的 TTP 能够访问时间和日期。

本文件规定的抗抵赖机制的强度依赖于所使用的密码学机制和参数的安全级别和强度。

注 2：本文件不规定抗抵赖机制中数据项的具体传输机制，抗抵赖机制的使用者可根据业务的安全需求来选择适当的机制，以确保使用抗抵赖机制的各实体间可以对数据项进行一致的解析。

7 安全信封

安全信封是一种数据完整性校验方法，用于在共享一个秘密密钥的两个实体（该密钥仅为这两个实体所知）间来相互传递消息。SENV 可以用于保护输入数据项，此时 SENV 使用实体的秘密密钥来产生；此外，SENV 也可以用于生成和验证证据，此时 SENV 由 TTP 使用仅为 TTP 持有的秘密密钥来产生。

对于要保护的数据 y，实体 X 使用如下的基于对称密码技术的完整性技术来创建安全信封。其中，实体 X 的秘密密钥 x 用于计算密码校验值 $MAC_X(y)$，并将该值附加在数据的后面，即：

$$SENV_X(y)=[y, MAC_X(y)]$$

在计算 $MAC_X(y)$ 时，应使用满足 GB/T 15852（所有部分）要求的消息鉴别码算法。

注：本文件规定的安全信封的强度依赖于所使用的密码学机制和参数的安全级别和强度，抗抵赖机制的使用者可根据实际业务的安全需求来选择适当的机制，GB/T 15852（所有部分）中给出了各种消息鉴别码机制安全性的说明。

8 抗抵赖令牌的生成与验证

8.1 TTP 创建令牌

在本章描述的抗抵赖机制中，TTP 担当证据生成和证据验证机构的角色。TTP 可信赖地维护特

定记录的完整性并直接参与解决争议。

对于要应用抗抵赖机制的消息 m，TTP 颁发与 m 相对应的“令牌”。令牌包括一个由 TTP 使用其秘密密钥 ttp 作用于该消息所确定的数据而形成的安全信封。因为其他实体都不知道 TTP 的秘密密钥 ttp，所以 TTP 是唯一可以创建和验证令牌的实体。根据 GB/T 17903.1 中对通用抗抵赖令牌的定义和本文件描述的场景，本节中 GNRT 的定义如下：

$GNRT=[text, SENV_{TTP}(y)]=[text, y, MAC_{TTP}(y)]$

y 表示安全信封中数据项，其具体内容见 8.2，其中包含与 m 相关的信息。此外，在发布令牌之前，TTP 应检查证据请求中的数据项。

注 1：消息 m 可以是明文或密文。

注 2：text 为文本域，包括一些不需要密码学保护但在计算完整性校验值 MAC 和安全信封 SENV 时要用到的，或用于标识消息和密钥的附加数据（如消息标识符或密钥标识符），text 的内容可以与消息 m 无关。本信息的具体内容依赖于所使用的技术。

8.2 抗抵赖机制使用的数据项

8.2.1 安全信封使用的数据项

在本文件描述的抗抵赖机制中，将对安全信封 $SENV_X(z_i)=[z_i, MAC_X(z_i)]$ 进行交换，下列数据字段构成了该安全信封的内容：

$z_i=[Pol, f_i, A, B, C, D, E, TG, T_i, Q, Imp(m)]$

数据域 z 中包含的数据项定义如下：

Pol ——适用于证据的抗抵赖策略的可区分标识符；

f_i ——提供的抗抵赖服务的类型；

A ——原发实体的可区分标识符；

B ——与原发实体进行交互的实体的可区分标识符；

C ——证据生成者的可区分标识符；

D ——证据请求者的可区分标识符，如果证据请求者与原发实体不同；

E ——动作涉及的其他实体的可区分标识符（的集合）；

TG ——证据生成的日期与时间；

T_i ——事件或动作发生的日期与时间；

Q ——需要保护的可选数据；

Imp(m)——与动作有关的消息 m 的印记，即 1）m 的杂凑码，或 2）m 本身。

注：在本文件中，根据抗抵赖服务类型的不同，i 的值为 1（原发抗抵赖）或 2（交付抗抵赖）。

8.2.2 抗抵赖令牌使用的数据项

在本文件描述的抗抵赖机制中，抗抵赖令牌 NRT 包括一个文本域 text 与一个安全信封，定义如下：

$NRT=[text, SENV_{TTP}(y)]$

注：文本域包括一些不需要密码学保护但在计算完整性校验值 MAC 和安全信封 SENV 时要用到的，或用于标识消息和密钥的附加数据（如消息标识符或密钥标识符）。本信息的具体内容依赖于所使用的技术。

8.3 抗抵赖令牌

8.3.1 证据提供

证据通常由抗抵赖令牌提供，如果策略要求，也可以由附加令牌提供，例如：时间戳令牌（TST）、或由另一个可信的第四方（如公证人）提供的对事件和动作以及消息的存在性给予附加保证的令牌等。

如果可信第三方可以独自生成可信时间戳，则不需要增加 TST 作为证据。

注 1：抗抵赖令牌(NROT、NRDT)中包含的时间可认为是安全可靠的，因为它是由可信机构提供的。

注 2：如果可信第三方(TTP、DA)不能提供可信时间戳，那么抗抵赖集合中就需要增加由可信时间戳机构(TSA)提供的 TST 以完成证据。

8.3.2 原发抗抵赖令牌

原发抗抵赖令牌(NROT)由 TTP 应原发者的请求而创建，其格式如下：

$NROT=[text, z_1, MAC_{TTP}(z_1)]$，其中

$z_1=[Pol, f_1, A, B, C, D, TG, T_1, Q, Imp(m)]$

NROT 所需信息 z_1 中包含的数据项定义如下：

Pol ——适用于证据的抗抵赖策略的可区分标识符；

f_1 ——原发抗抵赖服务的标记；

A ——原发者的可区分标识符；

B ——预定接收者的可区分标识符；

C ——生成证据的 TTP 的可区分标识符；

D ——观察者的可区分标识符，如果存在独立观察者；

TG ——证据生成的日期与时间；

T_1 ——消息原发的日期与时间；

Q ——需要保护的可选数据；

Imp(m)——与动作有关的消息 m 的印记，即 1)m 的杂凑码，或 2)m 本身。

8.3.3 交付抗抵赖令牌

交付抗抵赖令牌(NRDT)由 TTP 应接收者的请求而创建，其格式如下：

$NRDT=[text, z_2, MAC_{TTP}(z_2)]$，其中

$z_2=[Pol, f_2, A, B, C, D, TG, T_2, Q, Imp(m)]$

NRDT 所需信息 z_2 中包含的数据项定义如下：

Pol ——适用于证据的抗抵赖策略的可区分标识符；

f_2 ——交付抗抵赖服务的标记；

A ——原发者的可区分标识符；

B ——接收者的可区分标识符；

C ——TTP 的可区分标识符；

D ——观察者的可区分标识符，如果存在独立观察者；

TG ——证据生成的日期与时间；

T_2 ——消息交付的日期与时间；

Q ——需要保护的可选数据；

Imp(m)——与动作有关的消息 m 的印记，即 1)m 的杂凑码，或 2)m 本身。

8.3.4 时间戳令牌

时间戳令牌 TST 可由可信时间戳机构使用 GB/T 20520 中的方法生成。

8.4 TTP 进行的令牌验证

8.4.1 验证过程

在抗抵赖交换过程的某个环节上，可能需要 TTP 对实体的令牌(如上定义所示)进行验证。在交换

完成以后的某个时刻，也可能需要再次验证令牌，或者向第四方提供证据以证明其真实性。

验证过程不仅要检验令牌是否由 TTP 创建，而且要检验令牌是否与消息的数据字段确切相关。为了验证令牌是否为给定的消息而创建，实体把由消息计算而得的 Imp(m)与数据字段 z 中包括的 Imp(m)进行比较，然后要求 TTP 对令牌及其数据字段进行验证。

为了验证由对称完整性技术生成的安全信封，应进行如下操作：使用实体 X 的相应秘密密钥 x 对安全信封中包含的数据 y 重新计算密码校验值 $MAC_x(y)$，然后把结果与所提供的密码校验值进行比较。

TTP 应使用 8.4.2 与 8.4.3 中定义的两种验证方法之一进行验证。

8.4.2 在线令牌验证

当使用在线令牌验证方法时，TTP 使用包含秘密密钥 ttp 的安全模块来验证令牌。安全模块将该令牌与使用数据项 z_i 和秘密密钥 ttp 在其内部生成的值进行比较，并返回比较结果，该结果决定了令牌是否有效。由于密钥 ttp 不为 TTP 之外的任何人所知，如果安全模块返回的结果表明该令牌是有效的，那么所验证的令牌也可以认为是真实可信的。

8.4.3 令牌表

当使用令牌表方法时，TTP 将发布的所有令牌储存在一张表中。对每个已创建的令牌，TTP 记录下令牌和相关的数据字段(z_i)以及秘密密钥 ttp 的密钥标识符。要验证一个令牌，TTP 把该令牌作为索引在表中进行查找。如果在表中能够找到要验证的令牌，而且该令牌所带的数据字段(即令牌的一部分)与表中对应的数据字段相符，则认为该令牌是真实可信的。

9 特定抗抵赖机制

9.1 抗抵赖机制

本章规定的抗抵赖机制支持生成下列抗抵赖证据：原发抗抵赖、交付抗抵赖。另外，本章定义了时间戳的生成机制。当实体 A 想要向实体 B 发送消息，则实体 A 称为抗抵赖传输的原发者，实体 B 称为接收者，在本章中，称实体 A 为原发者 A，实体 B 为接收者 B。本章规定的抗抵赖机制的实例见附录 A。

注：当 Imp(m)即消息 m 本身时，不必将 m 与令牌一起发送，并且验证 Imp(m)的步骤也可省略。

9.2 原发抗抵赖机制

9.2.1 步骤与机制

原发者 A 创建了一条消息并发送给特定的接收者 B。接收者 B 使用 TTP 来验证与之相关的原发抗抵赖令牌，从而检验该消息来源于其声称的发送者。

本机制如图 1 所示，包含三个步骤：第一步，原发者 A 构造数据并封装入 SENV 发送给 TTP。TTP 生成原发抗抵赖令牌(NROT)并返回给原发者 A；第二步，原发者 A 将 NROT 与消息 m 发送给接收者 B；第三步，接收者 B 把安全信封中封装的 NROT 发送给 TTP 进行验证。原发抗抵赖在第三步建立。

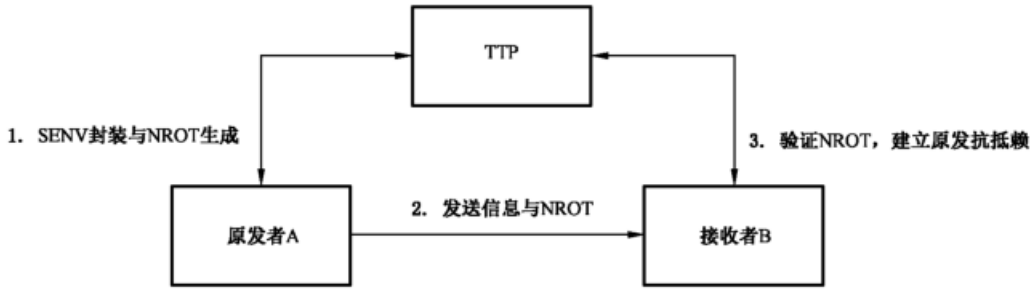

图 1　原发抗抵赖机制步骤

9.2.2　令牌生成

9.2.2.1　步骤 1——原发者 A 与 TTP 间

步骤 1 中，原发者 A 与 TTP 间执行如下操作。

a) 原发者 A 使用密钥 a 生成安全信封 $SENV_A(z_1')$，其中 z_1'同 8.3.2 规定的 z_1，但数据项 TG 为空。原发者 A 把安全信封发送给 TTP 以请求 NROT。

b) TTP 验证安全信封来自原发者 A。如果验证通过，TTP 插入数据项 TG 以完成 z_1，并使用密钥 ttp 计算：

$NROT=[text, z_1, MAC_A(z_1)]$

然后将 $SENV_A(NROT)$返回给原发者 A。

c) 原发者 A 验证 $SENV_A(NROT)$来自 TTP，且其中的 z_1 内容与 z_1'相一致。

9.2.2.2　步骤 2——原发者 A 到接收者 B

步骤 2 中，原发者 A 向接收者 B 发送：(m，NROT)。

9.2.2.3　步骤 3——接收者 B 与 TTP 间

步骤 3 中，接收者 B 与 TTP 间执行如下操作。

a) 接收者 B 执行以下验证操作：校验 z_1 中的策略 Pol 满足其安全要求；校验 z_1 中的 f_1 标示了原发抗抵赖令牌；校验标识符 A，B，C 有效；校验标识符 D 为实际存在的独立观察者；校验时间 TG 与 T_1 的正确性；校验 z_1 中 Imp(m)值的正确性。

b) 接收者 B 使用密钥 b 生成 $SENV_B(NROT)$并发送给 TTP，要求验证来自原发者 A 的 NROT。

c) TTP 验证 $SENV_B(NROT)$来自接收者 B，并验证 NROT 是真实可信的。如果 $SENV_B(NROT)$是有效的，TTP 向接收者 B 发送 $SENV_B(PON, NROT)$，其中：如果 NROT 是真实可信的，PON 为肯定，如果 NROT 不可信，则 PON 为否定。

d) 接收者 B 检验 $SENV_B(PON, NROT)$来自 TTP；若检验通过，并且验证结果 PON 为肯定，则建立了原发抗抵赖(即，消息来自原发者 A)。

e) 储存 NROT 以供将来原发抗抵赖使用。

9.2.3　令牌验证

若原发抗抵赖机制的证据使用者(接收者 B)在未来的某时刻需要再次验证 NROT 的真实可信性，则其可以单独执行 9.2.2.3 中规定的步骤 3 来完成验证。

9.3 交付抗抵赖机制

9.3.1 步骤与机制

本机制如图 2 所示，包含三个步骤：第一步，接收者 B 在接收到消息 m 后，向 TTP 发送请求以要求生产交付抗抵赖令牌，该请求封装在安全信封中，TTP 生成交付抗抵赖令牌（NRDT），并返回给接收者 B；第二步，接收者 B 将 NRDT 发送给原发者 A；第三步，原发者将 NRDT 封装在安全信封发送给 TTP 进行验证。交付抗抵赖在第三步建立。

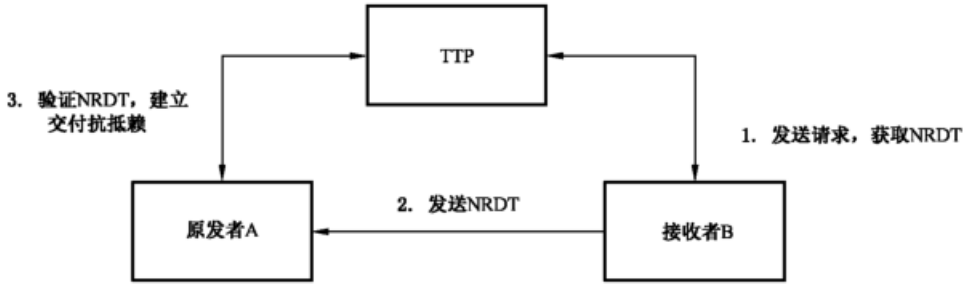

图 2 交付抗抵赖机制步骤

9.3.2 令牌生成

9.3.2.1 步骤 1——接收者 B 与 TTP 间

步骤 1 中，接收者 B 与 TTP 间执行如下操作。

a) 接收者 B 使用密钥 b 生成安全信封 $SENV_B(z_2')$，其中 z_2' 同 8.3.3 规定的 z_2，但数据项 TG 为空。接收者 B 把安全信封发送给 TTP 以请求 NRDT。

b) TTP 验证安全信封来自接收者 B。如果验证通过，TTP 插入数据项 TG 以完成 z_2，并使用密钥 ttp 计算：

$$NRDT=[text, z_2, MAC_{TTP}(z_2)]$$

然后将 $SENV_B(NRDT)$ 返回给接收者 B。

c) 接收者 B 验证 $SENV_B(NRDT)$ 来自 TTP，且其中的 z_2 内容与 z_2' 相一致。

9.3.2.2 步骤 2——接收者 B 到原发者 A

步骤 2 中，接收者 B 向原发者 A 发送：NRDT。

9.3.2.3 步骤 3——原发者 A 与 TTP 间

步骤 3 中，原发者 A 与 TTP 间执行如下操作。

a) 原发者 A 执行以下验证操作：校验 z_2 中的策略 Pol 满足其安全要求；校验 z_2 中的 f_2 标示了交付抗抵赖令牌；校验标识符 A，B，C 有效；校验标识符 D 为实际存在的独立观察者；校验时间 TG 与 T_2 的正确性；校验 z_2 中 Imp(m)值的正确性。

b) 原发者 A 使用密钥 a 生成 $SENV_A(NRDT)$ 并发送给 TTP，要求验证来自接收者 B 的 NRDT。

c) TTP 验证 $SENV_A(NRDT)$ 来自原发者 A，并验证 NRDT 是真实可信的。如果 $SENV_A(NRDT)$ 是有效的，TTP 向原发者 A 发送 $SENV_A(PON, NRDT)$，其中：如果 NRDT 是真实可信的，PON 为肯定，如果 NROT 不可信，则 PON 为否定。

d) 原发者 A 检验 $SENV_A(PON, NRDT)$ 来自 TTP；若检验通过，并且验证结果 PON 为肯定，则建立了交付抗抵赖。

e） 储存 NRDT 以供将来交付抗抵赖使用。

9.3.3 令牌验证

若交付抗抵赖机制的证据使用者(原发者 A)在未来的某时刻需要再次验证 NRDT 的真实可信性，则其可以单独执行 9.3.2.3 中规定的步骤 3 来完成验证。

9.4 获取时间戳令牌的机制

当可信时间源被请求且令牌生成方的时钟无法被信任时，需要依赖于可信第三方的 TSA 来提供可信时间戳。

实体 X(请求者)与 TSA 之间获取时间戳的通信可见 GB/T 20520。

附 录 A
（资料性）
抗抵赖机制实例

A.1 原发抗抵赖与交付抗抵赖机制实例

本附录所示的抗抵赖机制可在两个实体 A 和 B 之间提供原发抗抵赖和交付抗抵赖。实体 A 欲向实体 B 发送消息，于是成为抗抵赖交换的原发者。作为消息的接收方，实体 B 就是接收者。在本附录中，称实体 A 为原发者 A，实体 B 为接收者 B。在使用下列机制之前，假设实体 A 和实体 B 分别持有密钥 a 和 b，TTP 除了拥有自己的密钥 ttp 以外，还持有密钥 a 和 b。

下面给出了使用在线 TTP 的三种不同的抗抵赖机制（M1、M2 和 M3）。

注 1：通过在 SENV 消息中包含时间戳或序列号，可以防止未授权延迟或消息重放。通过在 NROT 和 NRDT 中包含时间戳，可进一步验证消息传输时的时间戳。

注 2：当 Imp(m)即消息 m 本身时，不必将 m 与令牌一起发送，并且验证 Imp(m)的步骤也可省略。

A.2 机制 M1：必选 NRO，可选 NRD

A.2.1 机制 M1 的步骤

机制 M1 见图 A.1，共包含 5 个步骤，在两个实体与 TTP 之间通过 3 个步骤建立原发抗抵赖，如果继续可选的 NRD 步骤（根据接收者的决定），那么可通过再执行 2 个步骤建立交付抗抵赖。

注 1：尽管是否继续进行交付抗抵赖的步骤取决于接收者，但要注意，一旦建立了交付抗抵赖，这一可选的交付抗抵赖就完全绑定了。

注 2：本机制可以提供原发抗抵赖并可选地提供交付抗抵赖。该协议的用法（仅提供原发抗抵赖或同时提供原发抗抵赖和交付抗抵赖）由发送者 A、接收者 B 和 TTP 在具体协议执行前商定。

A.2.2 步骤 1——原发者 A 与 TTP 间

步骤 1 中，原发者 A 与 TTP 间执行如下操作。

a) 原发者 A 使用密钥 a 生成安全信封 $SENV_A(z_1')$，其中 z_1' 同 8.3.2 规定的 z_1，但数据项 TG 为空。原发者 A 把安全信封发送给 TTP 以请求 NROT。

b) TTP 验证安全信封来自原发者 A。如果验证通过，TTP 插入数据项 TG 以完成 z_1，并使用密钥 ttp 计算：

$NROT=[text, z_1, MAC_{TTP}(z_1)]$

然后将 $SENV_A(NROT)$ 返回给原发者 A。

c) 原发者 A 验证 $SENV_A(NROT)$ 来自 TTP。

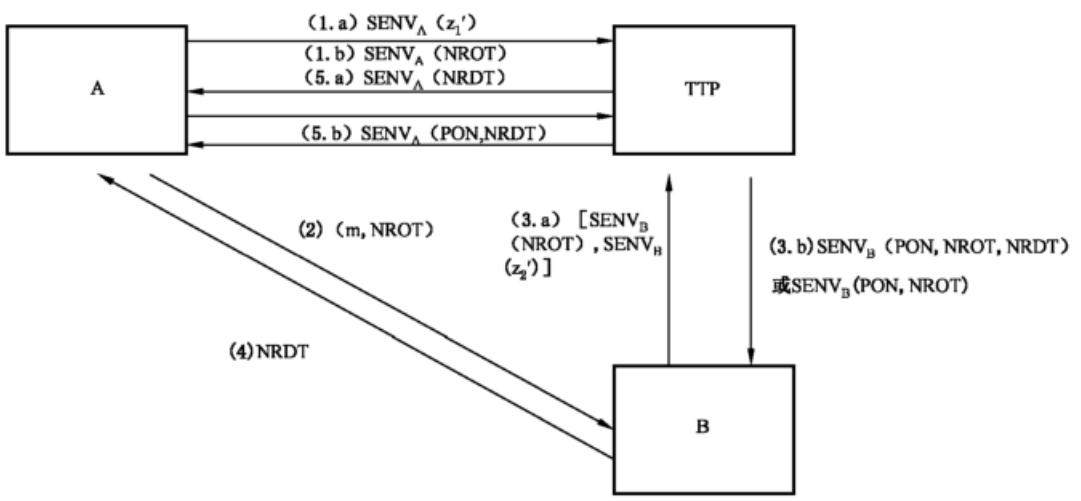

图 A.1　机制 M1

A.2.3　步骤 2——原发者 A 到接收者 B

步骤 2 中，原发者 A 向接收者 B 发送：(m，NROT)。

A.2.4　步骤 3——接收者 B 与 TTP 间

步骤 3 中，B 与 TTP 间执行如下操作。

a) 实体 B 验证 z_1 中 Imp(m)值的正确性。然后使用密钥 b 生成安全信封 SENV$_B$(NROT)和 SENV$_B$(z_2')，其中 z_2'同 8.3.3 规定的 z_2，但数据项 TG 为空。接收者 B 把上述安全信封发送给 TTP 以要求验证来自原发者 A 的 NROT 并请求生成 NRDT。

b) TTP 验证 SENV$_B$(NROT)与 NROT。如果两者均有效，则 TTP 验证 SENV$_B$(z_2')来自接收者 B。如果验证通过，TTP 插入数据项 TG 以完成 z_2，并计算：

NRDT=[text，z_2，MAC$_{TTP}$(z_2)]

如果 SENV$_B$(NROT)与 NROT 均是真实可信的，则 TTP 使用密钥 ttp 计算并向接收者 B 发送 SENV$_B$(PON，NROT，NRDT)，其中 PON 为肯定。如果 SENV$_B$(NROT)有效但 NROT 不可信，TTP 使用密钥 ttp 计算并向接收者 B 发送 SENV$_B$(PON，NROT)，其中 PON 为否定。

c) 接收者 B 检验 SENV$_B$(PON，NROT，NRDT)来自 TTP；若检验通过，并且验证结果 PON 为肯定，则建立了原发抗抵赖(即，消息来自原发者 A)。

d) 储存 NROT 以供将来原发抗抵赖使用。

A.2.5　步骤 4——接收者 B 到原发者 A

步骤 4 中，接收者 B 向原发者 A 发送：NRDT。

A.2.6　步骤 5——原发者 A 与 TTP 间

步骤 5 中，原发者 A 与 TTP 间执行如下操作。

a) 原发者 A 校验 z_2 中 Imp(m)值的正确性。然后使用密钥 a 生成 SENV$_A$(NRDT)并发送给 TTP，要求验证来自接收者 B 的 NRDT。

b) TTP 验证 SENV$_A$(NRDT)来自原发者 A，并验证 NRDT 是真实可信的。如果 SENV$_A$(NRDT)是有效的，TTP 向原发者 A 发送 SENV$_A$(PON，NRDT)，其中：如果 NRDT 是真实

可信的，PON 为肯定，如果 NROT 不可信，则 PON 为否定。

c) 原发者 A 检验 $SENV_A$(PON，NRDT)来自 TTP；若检验通过，并且验证结果 PON 为肯定，则建立了交付抗抵赖。

d) 原发者 A 储存 NRDT 以供将来交付抗抵赖使用。

A.3 机制 M2：必选 NRO，必选 NRD

A.3.1 机制 M2 的步骤

机制 M2 见图 A.2，在两个实体与 TTP 之间通过 4 个步骤建立原发抗抵赖和交付抗抵赖。在本机制中，TTP 在向接收者 B 发送消息收据的同时，直接通过 SENV 把它发送给原发者 A。

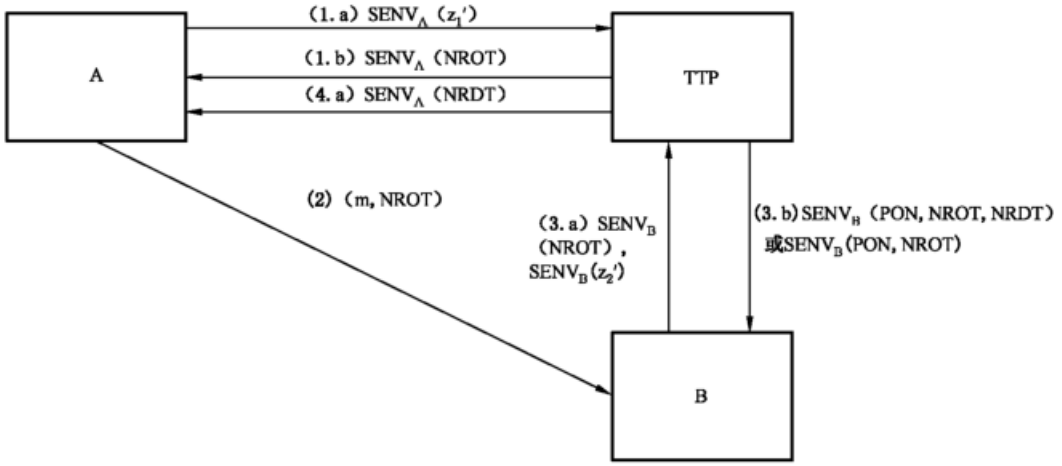

图 A.2 机制 M2

A.3.2 步骤 1——原发者 A 与 TTP 间

步骤 1 中，原发者 A 与 TTP 间执行如下操作。

a) 原发者 A 使用密钥 a 生成安全信封 $SENV_A(z_1')$，其中 z_1'同 8.3.2 规定的 z_1，但数据项 TG 为空。原发者 A 把安全信封发送给 TTP 以请求 NROT。

b) TTP 验证安全信封来自原发者 A。如果验证通过，TTP 插入数据项 TG 以完成 z_1，并使用密钥 ttp 计算：

$NROT=[text, z_1, MAC_{TTP}(z_1)]$

然后将 $SENV_A$(NROT)返回给原发者 A。

c) 原发者 A 验证 $SENV_A$(NROT)来自 TTP。

A.3.3 步骤 2——原发者 A 到接收者 B

步骤 2 中，原发者 A 向接收者 B 发送：(m，NROT)。

A.3.4 步骤 3——接收者 B 与 TTP 间

步骤 3 中，B 与 TTP 间执行如下操作。

a) 接收者 B 验证 z_1 中 Imp(m)值的正确性。然后使用密钥 b 生成安全信封 $SENV_B$(NROT)和 $SENV_B(z_2')$，其中 z_2'同 8.3.3 规定的 z_2，但数据项 TG 为空。接收者 B 把上述安全信封发送给 TTP 以要求验证来自 A 的 NROT 并请求生成 NRDT。

b) TTP 验证 $SENV_B$(NROT)与 NROT。如果两者均有效，则 TTP 验证 $SENV_B(z_2')$来自接收

者 B。如果验证通过，TTP 插入数据项 TG 以完成 z_2，并计算：

$NRDT=[text, z_2, MAC_{TTP}(z_2)]$

如果 $SENV_B(NROT)$与 NROT 均是真实可信的，则 TTP 使用密钥 ttp 计算并向接收者 B 发送 $SENV_B(PON, NROT, NRDT)$，其中 PON 为肯定。如果 $SENV_B(NROT)$有效但 NROT 不可信，TTP 使用密钥 ttp 计算并向接收者 B 发送 $SENV_B(PON, NROT)$，其中 PON 为否定。

c) 接收者 B 检验 $SENV_B(PON, NROT, NRDT)$来自 TTP；若检验通过，并且验证结果 PON 为肯定，则建立了原发抗抵赖。

d) 储存 NROT 以供将来原发抗抵赖使用。

A.3.5 步骤 4——原发者 A 与 TTP 间

步骤 4 中，原发者 A 与 TTP 间执行如下操作。

a) 在步骤 3 中，如果 $SENV_B(NROT)$与 NROT 均是真实可信的，则 TTP 向接收者 B 发送 NRDT 之后，TTP 立即向原发者 A 发送 $SENV_A(NRDT)$。

b) 原发者 A 检验 $SENV_A(NRDT)$与 NRDT；若检验通过，则建立了交付抗抵赖（即，消息被接收者 B 接收）。

c) 原发者 A 储存 NRDT 以供将来交付抗抵赖使用。

A.4 机制 M3：带有中介 TTP 的必选 NRO 和必选 NRD

A.4.1 机制 M3 的步骤

机制 M3 见图 A.3，在两个实体与 TTP 之间通过 4 个步骤建立原发抗抵赖和交付抗抵赖。在本机制中，TTP 在原发者 A 和接收者 B 之间充当了中间人的角色，两个实体不再直接通信。为此，原发者 A 发送消息给 TTP 作为步骤 1 中的一部分，TTP 将其传递给接收者 B 作为步骤 2 的一部分。

在本机制中，TTP 可选地生成并向原发实体发送提交抗抵赖与传输抗抵赖令牌。

A.4.2 步骤 1——原发者 A 与 TTP 间

步骤 1 中，原发者 A 与 TTP 间执行如下操作。

a) 原发者 A 使用密钥 a 生成安全信封 $SENV_A(z_1')$，其中 z_1'同 8.3.2 规定的 z_1，但数据项 TG 为空。原发者 A 把安全信封发送给 TTP 以请求 NROT。

b) TTP 验证安全信封来自原发者 A。如果验证通过，TTP 插入数据项 TG 以完成 z_1，并使用密钥 ttp 计算：

$NROT=[text, z_1, MAC_{TTP}(z_1)]$

然后将 $SENV_A(NROT)$返回给 A。

c) 原发者 A 验证 $SENV_A(NROT)$来自 TTP。

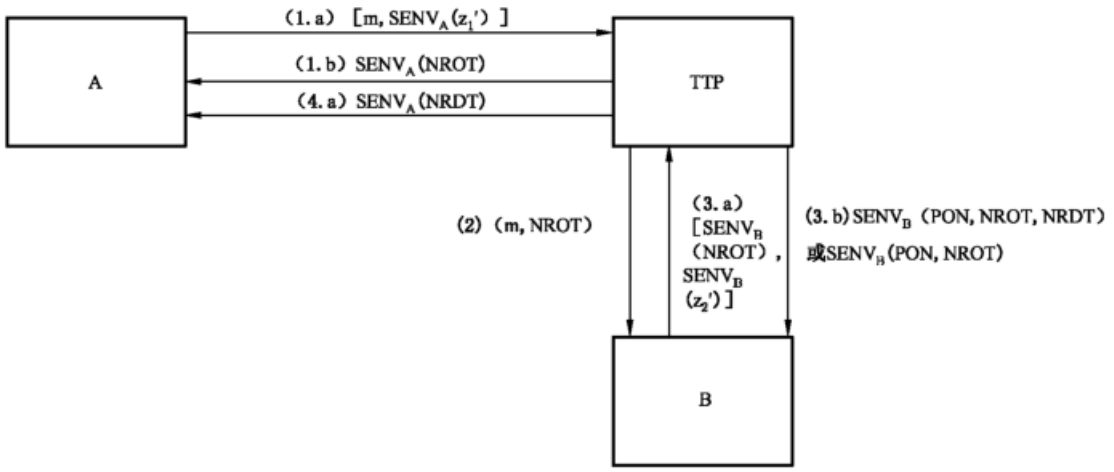

图 A.3　机制 M3

A.4.3　步骤 2——TTP 到接收者 B

步骤 2 中，TTP 向接收者 B 发送：(m，NROT)。

A.4.4　步骤 3——接收者 B 与 TTP 间

步骤 3 中，接收者 B 与 TTP 间执行如下操作。

a) 由于 NROT 不是以安全信封的方式收到的，因此接收者 B 需要与 TTP 一起来验证 NROT，所以 B 在验证 z_1 中 Imp(m)值的正确性之后，使用密钥 b 生成安全信封 $SENV_B$(NROT)和 $SENV_B(z_2')$，其中 z_2'同 8.3.3 规定的 z_2，但数据项 TG 为空。接收者 B 把上述安全信封发送给 TTP 以要求验证来自原发者 A 的 NROT 并请求生成 NRDT。

b) TTP 验证 $SENV_B$(NROT)与 NROT。如果两者均有效，则 TTP 验证 $SENV_B(z_2')$来自接收者 B。如果验证通过，TTP 插入数据项 TG 以完成 z_2，并计算：

$NRDT=[text, z_2, MAC_{TTP}(z_2)]$

如果 $SENV_B$(NROT)与 NROT 均是真实可信的，则 TTP 使用密钥 ttp 计算并向接收者 B 发送 $SENV_B$(PON，NROT，NRDT)，其中 PON 为肯定。如果 $SENV_B$(NROT)有效但 NROT 不可信，TTP 使用密钥 ttp 计算并向接收者 B 发送 $SENV_B$(PON，NROT)，其中 PON 为否定。

c) 接收者 B 检验 $SENV_B$(PON，NROT，NRDT)来自 TTP；若检验通过，并且验证结果 PON 为肯定，则建立了原发抗抵赖。

d) 储存 NROT 以供将来原发抗抵赖使用。

A.4.5　步骤 4——原发者 A 与 TTP 间

步骤 4 中，原发者 A 与 TTP 间执行如下操作。

a) 在步骤 3 中向接收者 B 发送 NRDT 之后，TTP 立即向原发者 A 发送 $SENV_A$(NRDT)。

b) 原发者 A 检验 $SENV_A$(NRDT)与 NRDT；若检验通过，则建立了交付抗抵赖。

c) 原发者 A 储存 NRDT 以供将来交付抗抵赖使用。

参 考 文 献

[1] GB/T 17901.1—2020 信息技术 安全技术 密钥管理 第1部分:框架

[2] GB/T 18794.4—2003 信息技术 开放系统互连 开放系统安全框架 第4部分:抗抵赖框架

[3] ISO/IEC 11770-2:2008 Information technology—Security techniques—Key management—Part 2: Mechanisms using symmetric techniques

[4] ISO/IEC 11770-3:2008 Information technology—Security techniques—Key management—Part 3: Mechanisms using asymmetric techniques