ICS 35.040
L 80

中华人民共和国国家标准

GB/T 20520—2006

信息安全技术　公钥基础设施 时间戳规范

Information security technology—Public key infrastructure—Time stamp specification

2006-08-30 发布　　2007-02-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　　次

前　言

本标准由全国信息安全标准化技术委员会提出并归口。

本标准起草单位：中国科学院信息安全国家重点实验室。

本标准主要起草人：冯登国、张凡、荆继武、庄湧、张立武、路晓明。

引　言

本标准主要对时间戳协议的请求响应消息格式做出了规定，并在此基础上增加了对时间戳的产生和颁发方式、时间戳系统组成、时间戳管理、时间戳系统安全的要求。

本标准参考了国内外的相关时间戳规范，最大程度地保证标准的互操作性，保证了 TSA 的互操作性、TSA 和时间戳的安全性以及 TSA 的时间精确性，为开发时间戳产品提供了可依据的标准。

本标准凡涉及密码算法相关内容，按国家密码管理部门相关规定执行。

本标准例子中提及的密码算法均为举例性说明，具体使用时均须采用国家密码管理部门批准的相应算法。

信息安全技术　公钥基础设施
时间戳规范

1　范围

本标准规定了时间戳系统部件组成、时间戳的管理、时间戳的格式和时间戳系统安全管理等方面的要求。

本标准适用于时间戳系统的设计和实现，时间戳系统的测试和产品采购亦可参照使用。

2　规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

GB 17859—1999　计算机信息系统安全保护等级划分准则

GB/T 20518—2006　信息安全技术　公钥基础设施　数字证书格式

GB/T 20273—2006　信息安全技术　数据库管理系统安全技术要求

GB/T 20271—2006　信息安全技术　信息系统通用安全技术要求

RFC 2630　加密消息语法

3　术语和定义

下列术语和定义适用于本标准。

3.1

时间戳　time stamp

使用数字签名技术产生的数据，签名的对象包括了原始文件信息、签名参数、签名时间等信息。TSA 对此对象进行数字签名产生时间戳，以证明原始文件在签名时间之前已经存在。

3.2

可信时间　trusted time

准确的、值得信赖的当前时间值，这个时间值的来源应是高度权威的。

3.3

时间戳机构　time stamp authority

用来产生和管理时间戳的权威机构。

3.4

时间戳协议　time stamp protocol

由本标准规定的一系列规范，包括时间戳的格式、各部件交流的消息格式、时间戳的颁发方式等内容。

3.5

时间戳服务　time stamp service

时间戳机构给用户提供的颁发时间戳服务，由用户提供文件，时间戳机构给此文件签发时间戳。

3.6

请求方　requester

向 TSA 系统发出申请时间戳请求的人、硬件或者软件。

4 缩略语

下列缩略语适用于本标准：

CA　认证机构（Certification Authority）

CRL　证书撤销列表(Certificate Revocation List)

FTP　文件传输协议(File Transfer Protocol)

OCSP　在线证书状态协议(Online Certificate Status Protocol)

OID　对象标识符(Object Identifier)

PKI　公开密钥基础设施(Public Key Infrastructure)

TSA　时间戳机构(Time Stamp Authority)

TSP　时间戳协议(Time Stamp Protocol)

UTC　协调世界时(Universal Time Coordinated)

HTTP　超文本传输协议(Hyper Text Transfer Protocol)

MIME　多用途网络邮件扩充协议(Multipurpose Internet Mail Extension)

5 时间戳系统的组成

时间戳系统至少要包括三个部分：

a) 可信时间源：即 TSA 的时间来源，TSA 系统中的所有部件的时间都应以这个可信时间源为标准，尤其在颁发的时间戳中填写的时间应严格按照可信时间源填写。而作为可信时间源自身，其或者就是国家权威时间部门发布的时间，或者是用国家权威时间部门认可的硬件和方法获得的时间。此外，TSA 应估算从可信时间源到 TSA 的时间传递过程中可能出现的误差，TSA 应公布最大可能误差作为其可信程度的一个标志。

b) 签名系统：负责接收时间戳申请、验证申请合法性以及产生和颁发时间戳，最后将时间戳存储到数据库中。这个过程中，申请消息和颁发时间戳格式都要符合第 8 章所规定的格式，而时间戳的产生和颁发应符合第 6 章给出的要求。

c) 时间戳数据库：负责保存 TSA 系统颁发的时间戳，而且应定期备份。对时间戳数据库的存储、备份和检索应符合第 7 章给出的规定。

时间戳系统的结构如图 1 所示：

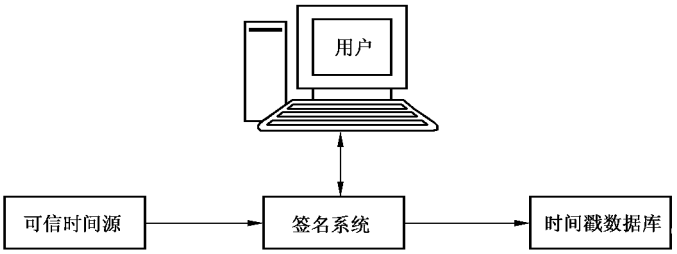

图 1　时间戳系统结构

6 时间戳的产生和颁发

6.1 申请和颁发方式

TSA 可以通过不同的方式接收时间戳申请和颁发时间戳，但应至少支持下列四种方式的其中一种：

a) 通过电子邮件申请。用户使用电子邮件向一个 TSA 指定的电子邮件地址发送时间戳申请，而 TSA 也将颁发的时间戳通过电子邮件返回给用户。申请与颁发使用的 MIME 对象在第 8 章中说明。

b) 通过文件传输申请。用户将申请消息的编码存储在一个文件中,文件被传送给 TSA 后,TSA 同样将产生的时间戳保存在一个文件中传给用户。文件的传送可以使用任意可信赖的方法,例如使用 FTP 协议。

c) 通过套接字(Socket)申请。TSA 在计算机的某个端口监听用户发来的申请,而用户在与 TSA 计算机的这个端口建立一个安全的套接字连接后,把申请消息发给 TSA。最后 TSA 同样把产生的时间戳在这个连接上发给用户。

d) 通过 HTTP 申请。用户连上 TSA 的申请网页后,使用网页产生申请消息,然后通过 HTTP 协议将申请消息传送给 TSA,而 TSA 也随后通过 HTTP 协议发回时间戳。

6.2 可信时间的产生方法

可信时间的最初源头应来源于国家权威时间部门(如国家授时中心),或者使用国家权威时间部门认可的硬件和方法获得的时间。

可以使用以下的一种或多种方法获得时间:

a) 使用某种无线接收装置,通过无线手段获得国家权威时间部门的时间发布,如长波信号、卫星信号等。

b) 使用某种时间同步协议从一个指定网络地址获得时间。该网络地址发布的时间和使用的时间同步协议都应是可信的,且通过了国家权威时间部门认可。

c) 使用某种通过国家权威时间部门认证的硬件获得时间,如使用原子钟等。

考虑到任何一种方法都可能产生误差,TSA 可以使用多种方法产生可信时间,以保证时间的精确度。通过多种方法产生最终可信时间应该是产生的多个可信时间的折衷。TSA 应该给出一个可靠方案,该方案要考虑每种方法的可能误差和可信程度,对它们的结果做出一个加权平均,获得最终结果。

6.3 时间的同步

TSA 系统各部件根据可信时间同步自身的时间,这一行为应满足下列条件:

a) 在获得可信时间后,TSA 应迅速根据可信时间对所有部件的时间进行调整(尤其是最关键的签名系统),TSA 应该保证这一过程尽可能的快,并且尽可能保证其不被打断。

b) 为了保证 TSA 各个部件的时间精确性,TSA 应该定期从可信时间源获得可信时间,再根据可信时间检查自身时间,与可信时间源保持时间同步。

c) 每次同步的间隔时间应该是一个比较短的时间。具体时间视 TSA 的运行策略以及运行 TSA 的硬件或软件时钟的可靠性而定,但应取得尽可能大的安全性,并且不应长于 30 min。

d) 每次同步的间隔时间应该是可配置的,在 TSA 运行的策略和环境变化后,可以调整这一时间间隔以取得尽可能大的安全性。

e) TSA 各个部件应该采取统一行动检查并同步时间,不允许出现一个部件检查同步了而其他部件却不行动的情况。

f) 在启动 TSA 系统的过程中,可信时间源应是第一个启动的部件。而且在 TSA 开始工作之前,时间同步应该先进行一段时间,以保证 TSA 开始颁发时间戳时,各部件的时间已经同可信时间源同步过了。

g) 在定期同步时间的过程中,如果获得可信时间失败或者发现收到的时间信息被篡改,TSA 系统应该立即停止接受时间戳申请和时间同步,同时向管理者发出警报并写入审计日志。

6.4 申请和颁发过程

无论 TSA 的运行方式使用 6.1 中说明的哪一种方法,整个申请和颁发时间戳的过程至少应该包括以下基本过程:

a) 用户通过 6.1 中说明的一种方法,向 TSA 提交申请请求,请求消息的格式应该符合第 8 章的规定。

b) TSA 的签名系统接收到申请请求后,根据第 8 章中对时间戳格式的说明,对请求消息的合法性进行检查。

c) 如果请求消息不合法或者由于某种内部原因 TSA 无法颁发这个时间戳，TSA 应该产生一个时间戳的失败响应，其格式也应该遵循第 8 章的规定。在其中 TSA 应该详细填写申请被拒绝的原因。

d) 如果请求消息合法，且系统也正常运转，TSA 的签名系统就应该根据第 8 章中说明的时间戳格式，填写正常的时间戳并签名。

e) TSA 签名系统通过可信通道把新生成的时间戳发送给时间戳数据库，由时间戳数据库将其归档保存。而由于申请被拒绝而产生的时间戳失败响应，应该由 TSA 本身的策略决定是否将其保存，本标准不作硬性规定。

f) TSA 通过与用户申请方式对应的颁发方式，将新生成的时间戳发给用户。

g) 用户在收到时间戳后，应该使用 TSA 的证书验证时间戳的合法性，并检查时间戳内容是否有错误。如果时间戳不合法或者有错误，用户应该立即向 TSA 管理者报告异常情况，TSA 机构应该提供一个用户反馈渠道，在用户发现异常情况时可以通过此渠道通知管理员。如果时间戳一切正常，用户可以自行保存此时间戳，以备后用。

h) 如果管理员收到用户的异常报告，应该立即检查审计日志和时间戳数据库，找出错误原因所在。TSA 应该对这种情况准备有完备的处理预案。

7 时间戳的管理

7.1 时间戳的保存

7.1.1 在 TSA 方的保存

在 TSA 系统中，时间戳数据库应该负责保存由此 TSA 产生的所有时间戳。该时间戳数据库应符合 9.2.4 对数据库的要求。

考虑到空间限制，时间戳数据库可以在一定时间后或者数据库中的数据达到一定的量后，将数据库中的所有数据转移到它处，而将其自身清空。这一过程只能由管理员执行，并且转移后的保存要符合 7.2 对时间戳备份的要求。

对于每一个时间戳，时间戳数据库在保存它们时至少要保存以下相关信息：

a) 时间戳入库的时间；

b) 时间戳的序列号；

c) 时间戳的完整编码。

除此之外，对于时间戳的保存，应考虑今后可能的用途，如用户查询、取证，存储的必要信息应便于今后使用。

7.1.2 在用户方的保存

在用户收到时间戳后，用户自行将时间戳保存在一个文件中。由用户自行保证时间戳的安全性，如果保存的时间戳发生了问题，用户可以向 TSA 机构申请取回自己的时间戳。

7.2 时间戳的备份

时间戳应按如下要求备份：

a) 每隔一定时间，管理员应该备份时间戳数据库的所有数据；

b) 备份所使用的介质应符合 9.1 的要求；

c) 备份应该使用异地备份的方式；

d) 备份数据应该是以方便检索的方式存放；

e) 备份数据的访问应在有管理员在场的时候进行；

f) 备份数据不一定需要加密或签名，但如果使用，所选择的算法应符合国家密码管理部门的相关规定。

7.3 时间戳的检索

TSA 应该给用户提供一个能够方便的检索时间戳的环境，使用户可以通过网络或者面对面的方式检索和获得时间戳。

TSA 提供给用户检索的时间戳应该不仅仅是时间戳数据库中保存的时间戳，还应该包括以前备份的时间戳。

TSA 应该至少支持通过以下三种信息检索时间戳：

a) 根据时间戳入库的时间检索，允许检索出多个结果，再由用户自行选择；

b) 根据时间戳的序列号检索，由于序列号唯一，这种检索应该只有唯一一个结果；

c) 根据时间戳的完整编码检索，这种检索也应该只有唯一一个结果。

时间戳的检索结果可以通过 6.1 的颁发方式发给用户，也可以通过另外的可靠方式，如使用 IC 卡、光盘等，让用户带回。

7.4 时间戳的删除和销毁

7.4.1 时间戳的删除

当 TSA 系统由于内部错误或者外部攻击导致产生错误的时间戳时，应该允许删除时间戳数据库中的错误数据。

所有从时间戳数据库中删除的时间戳应先进行备份，以备以后审计查询。这种备份数据应与正常备份数据区分开来，单独存放，但同样也需要符合 7.2 中对备份的要求。

所有遭到删除的时间戳应在第一时间在公开渠道公布，如互联网等。公布的时间戳信息应该尽可能的详细。

时间戳的删除只能由 TSA 管理员操作进行。

7.4.2 时间戳的销毁

在确定时间戳已经丧失其价值后，TSA 可以完全销毁时间戳，即不仅从时间戳数据库中删除，也从所有时间戳备份中删除。

在时间戳生成后，应经过足够长的时间，时间戳才允许被销毁。这一保存时间应足够长，长到可以确定该时间戳已丧失使用价值。此外，保存时间也可以由 TSA 的策略决定，但 TSA 应在用户申请时间戳时向用户详细说明。但是无论采用什么方法决定保存时间，时间戳都只能在 TSA 证书失效后销毁。

时间戳的销毁只能由 TSA 管理员操作进行。

7.5 时间戳的查看和验证

7.5.1 时间戳的查看

TSA 应该给用户提供一个方便安全的方法查看其颁发的时间戳，例如提供一个查看软件等。用户可以凭借此方法查看时间戳中所有可查看的内容。

7.5.2 时间戳的验证

TSA 应该给用户提供一个方便安全的方法对其颁发的时间戳进行验证，例如提供一个验证软件或者通过互联网验证等。

验证前，用户应该可以通过 CRL 或 OCSP 协议先验证 TSA 证书的有效性。

TSA 提供的验证服务应该包括以下两种：

a) 使用 TSA 证书验证用户给定的时间戳是否是由该 TSA 签发；

b) 用户提供时间戳和源文件，验证该时间戳是否是该文件的时间戳。

8 时间戳的格式

8.1 对 TSA 的要求

完成一个完整的时间戳，TSA 系统应满足下列条件：

a) 拥有一个可信时间源，应满足 9.2.2 的要求，可信时间的产生要满足 6.2 的要求；

b) 在每一个时间戳里都要包含一个可信时间值；

c) 在每一个新生成的时间戳里都要包含一个一次性随机整数(nonce 域)；

d) 只要可能，当从一个请求者处收到一个合法请求时，就要根据这个请求生成一个时间戳；

e) 当时间戳生成时，要在其内包含一个唯一的标识符，这个标识符表明了时间戳生成时的安全策略；

f) 只在数据的散列(Hash)值上盖时间戳，散列函数拥有一个唯一的对象标识符(OID)；

g) 要能够检查单向散列函数的标识符，并且验证数据的散列值长度是否符合该散列函数的结果长度；

h) 除了 g)中要求的对散列值长度的检查外，对于输入的散列值数据不做任何其它检查；

i) 在时间戳内不包含任何请求方的标识；

j) 用专门的密钥对时间戳签名，在密钥相应的证书里应该说明该密钥的这个用途；

k) 如果请求方在申请消息的扩展域内提出了一些额外要求并且 TSA 支持这些扩展，TSA 就应该在时间戳内包含相应的额外信息。反之，如果 TSA 不支持这些扩展，就应该返回一个出错信息。

8.2 密钥标识

TSA 应有专门的密钥对时间戳消息签名。但一个 TSA 可以有许多不同的私钥以适应不同的要求，例如适应不同的策略、不同的算法、不同的密钥长度和不同的性能要求。相应的证书也应包含唯一一个 Key Usage 的扩展域，该域在 GB/T 20518—2006 的 5.2.3.2.4 中定义。其中的 KeyPurposeID 应为 Id-kp-timestamping：

```
id-kp-timeStamping OBJECT IDENTIFIER ::= {iso(1)
                    identified-organization(3) dod(6)
                    internet(1) security(5) mechanisms(5) pkix(7)
                    kp (3) timestamping (8)}
```

8.3 时间的表示格式

时间戳中使用的时间应是 UTC 时间。其精度应至少精确到秒。

语法结构为：YYYYMMDDhhmmss[.s...]Z。

例如：20031101001326.34352Z。

各个位解释如下：

a) YYYY 为年份，应是 4 位数年份如 2003；

b) MM 为月份，如果月份只有一位数，要加上一个前导 0，如 01、11；

c) DD 为日，如果只有一位数，要加上前导 0，如 01、20；

d) hh 为小时，如果只有一位数，要加上前导 0，如 00、23；

e) mm 为分钟，如果只有一位数，要加上前导 0，如 05、59；

f) ss 为秒，如果只有一位数，要加上前导 0，如 01、59；

g) [.s...]是可选的，表示秒的小数部分。小数点如果出现应是“.”，秒的小数部分如果出现，应把后面跟的 0 都省略掉，如果秒的小数部分等于 0，则应全部都省略掉，小数点也应省略掉；

h) 最后的 Z 表示这是一个 UTC 时间。

午夜(格林威治时间)应该表示成：“YYYYMMDD000000Z”，其中的“YYYYMMDD”表示午夜之后的这一天。

8.4 时间戳申请和响应消息格式

8.4.1 申请消息格式

时间戳服务申请消息格式如下：

```
TimeStampReq ::= SEQUENCE  {
        version                 INTEGER  { v1(1) },
        messageImprint          MessageImprint,
        reqPolicy               TSAPolicyId              OPTIONAL,
        nonce                   INTEGER                  OPTIONAL,
        certReq                 BOOLEAN                  DEFAULT FALSE,
        extensions              [0] IMPLICIT Extensions  OPTIONAL  }
```

对于各项的具体解释如下：

a) 版本(version)域表示时间戳申请消息格式的版本号，依据本标准写成的申请消息版本为 1。

b) messageImprint 域应该包含需要加盖时间戳的数据的散列值。该散列值的类型是 Octet String，它的长度应是相应散列算法的结果长度(例如：SHA-1 算法结果是 160 位，MD5 算法结果是 128 字节)。具体格式：

```
MessageImprint ::= SEQUENCE  {
        hashAlgorithm       AlgorithmIdentifier,
        hashedMessage       OCTET STRING  }
```

在 hashAlgorithm 域中表示的散列算法应该是一个已知的散列算法(一个单向函数)。TSA 应该检查给出的散列算法是否符合国家密码管理部门的相关规定。如果 TSA 不认识给出的散列算法或这个散列算法不符合国家密码管理部门的相关规定，那么 TSA 应该拒绝提供时间戳服务，并在返回消息中设置'bad _ald'的 pkiStatusInfo 结构。

c) 如果用户需要指明时间戳应该在什么样的安全策略下生成，用户可以设置 reqPolicy 域说明需要的安全策略。

d) 如果请求消息中有 nonce 域，它是为了在没有可靠的本地时钟的情况下检验响应消息的合法性并防止重放攻击。nonce 是一个很大的随机数，而且以很高的概率不被重复(例如：一个 64 比特的整数)。在这种情况下，nonce 应被包含在响应消息中，否则响应消息应该被拒绝接受。

e) 如果请求消息中有 certReq 域且被设为 true，则 TSA 应在其响应消息中给出它的公钥证书，该证书由响应消息中 SigningCertificate 属性的 ESSCertID 指出，证书本身则存放在响应消息中 SighedData 结构的 Certificates 域。这个域也可以包含其他证书。如果请求消息中没给出 certReq 域或者 certReq 域被设为 false，则响应消息中就不必要给出上述证书。

f) 扩展(extensions)域是未来给申请消息添加额外信息的一种方法。扩展都在 GB/T 20518—2006 中定义。对于一个扩展，无论它是否是关键扩展，只要它在请求消息中出现，且又无法被 TSA 识别，则 TSA 应该不生成时间戳且返回一个失败信息(unacceptedExtension)。

时间戳请求消息不需要给出请求方的身份标识，因为 TSA 并不验证这个信息(见 8.1)。如果在某些情况下 TSA 需要验证请求方的身份，那么就应进行双向的身份验证。

8.4.2 响应消息格式

TSA 在收到申请消息后，无论申请成功还是失败，都要给请求方发回一个响应消息。该响应消息或者是正确的时间戳，或者是包含了失败信息的时间戳。时间戳响应消息的具体格式如下：

```
TimeStampResp ::= SEQUENCE  {
        status                  PKIStatusInfo,
        timeStampToken          TimeStampToken          OPTIONAL  }
```

响应消息中的 status 具体定义如下：

```
PKIStatusInfo ::= SEQUENCE {
        status          PKIStatus,
        statusString    PKIFreeText     OPTIONAL,
```

```
                failInfo          PKIFailureInfo      OPTIONAL   }
```

其中 PKIStauts 有如下定义：

```
PKIStatus ::= INTEGER {
                granted                    (0),
                grantedWithMods            (1),
                rejection                  (2),
                waiting                    (3),
                revocationWarning          (4),
                revocationNotification     (5)   }
```

如果 PKIStatusInfo 中的 status 值为 0 或者 1 时，响应消息中的 TimeStampToken 就应出现，否则 TimeStampToken 就不能出现。

status 不能有除 PKIStatus 外的其他值，如果请求方收到一个不认识的值，查看时间戳时应该报告错误。

如果申请失败，则用 statusString 给出一个说明失败原因的字符串。而 failInfo 也用来说明时间戳请求被拒绝的具体原因，具体值如下

```
PKIFailureInfo ::= BIT STRING {
        badAlg                (0),        --申请使用了不支持的算法
        badRequest            (2),        --非法的申请
        badDataFormat         (5),        --数据格式错误
        timeNotAvailable      (14),       --TSA 的可信时间源出现问题
        unacceptedPolicy      (15),       --不支持申请消息中声明的策略
        unacceptedExtension   (16),       --申请消息中包括了不支持的扩展
        addInfoNotAvailable   (17),       --有不理解或不可用的附加信息
        systemFailure         (25)        --系统内部错误 }
```

failInfo 不能有除 PKIFailureInfo 外的其他值，如果请求方收到一个不认识的值，查看时间戳时应该报告错误。

TimeStampToken 实际上应该是一个 ContentInfo 结构，该结构在 RFC2630 中定义。并且其 content type 应该是一个 signed data content type。

```
TimeStampToken ::= ContentInfo
        --contentType 为 RFC2630 所定义的 id-signedData
        --content 为 RFC2630 所定义的 SignedData
```

在 ContentInfo 定义的 SignedData 结构中，EncapsulatedContentInfo 类中的域有如下意思：

a) eContentType 是一个对象标识符唯一指定内容的类型。对于一个时间戳，它定义为：

```
id-ct-TSTInfo   OBJECT IDENTIFIER ::=
        { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) ct(1) 4};
```

b) eContent 就是时间戳内容本身，它是一个 octet string。内容应该是下面说明的 TSTInfo 的 DER 编码。

TimeStampToken 一定不能含有除 TSA 签名以外的任何签名。而 TSA 证书的证书标识符(ESSCertID)应作为 SignerInfo 的属性包含在 SigningCertficate 属性里。

关于 TSTInfo 的具体定义如下：

```
TSTInfo ::= SEQUENCE  {
                version                 INTEGER  { v1(1) },
                policy                  TSAPolicyId,
```

```
messageImprint      MessageImprint,
serialNumber        INTEGER,
genTime             GeneralizedTime,
accuracy            Accuracy                    OPTIONAL,
ordering            BOOLEAN                     DEFAULT FALSE,
nonce               INTEGER                     OPTIONAL,
tsa                 [0] GeneralName             OPTIONAL,
extensions          [1] IMPLICIT Extensions     OPTIONAL  }
```

对于 TSTInfo 的具体各项解释如下：

a) version 域说明了时间戳的版本号，依据本标准写成的时间戳版本号为 1。

b) policy 域应指明响应消息是根据 TSA 的哪个策略生成的。如果类似的域出现在 TimeStampReq 中，这里应有相同的值，否则应返回错误(unacceptedPolicy)。这个 policy 可以包含下列类型的信息(但下列并不全面)：

——这个时间戳在什么条件下使用；

——时间戳日志的有效性，以便以后能够证实时间戳是可信的。

c) messageImprint 应同 TimeStampReq 中类似的域有相同的值，前提是散列值的长度与 hashAlgorithm 标记的算法预期的长度相同。

d) serialNumber 域是 TSA 分配给每个时间戳的一个整数。对一个给定的 TSA 发出的每一个时间戳它都应是唯一的(即 TSA 的名字和序列号可以确定一个时间戳标志)。应该注意的是，即使经历一个可能的服务中断(例如崩溃)后，这个特性也应保留。

e) genTime 是 TSA 创建时间戳的时间。用 UTC 时间表示，以减少使用本地时区用法造成的混乱。时间的具体格式应遵守 8.3 的规定。

f) accuracy 表示时间可能出现的最大误差，genTime 加上 accuracy 的值，就可以求得 TSA 创建这个时间戳的时间上限，同理，减去 accuracy 的值就是 TSA 创建时间戳的时间下限。具体定义如下：

```
Accuracy ::= SEQUENCE {
                seconds         INTEGER                 OPTIONAL,--s
                millis          INTEGER (1..999)        OPTIONAL,--ms
                micros          [1] INTEGER (1..999)    OPTIONAL--μs }
```

如果 seconds、millis 或者 micros 没出现，则不出现的这些域的值应被赋为 0。

当 accuracy 这个可选项不出现时，精确度可以从别的途径得到，例如 TSAPolicyId。

g) ordering 表示时间戳排序条件。如果 ordering 域不出现，或者 ordering 域出现但被置为 false，那么 genTime 域只表示 TSA 创建时间戳的时间。在这种情况下，只有两个时间戳中第一个的 genTime 与第二个的 genTime 之差大于这两个 genTime 的精确度的和，同一个 TSA 或者不同的 TSA 签发的时间戳标志才有可能排序。如果 ordering 域出现并被置为 true，同一个 TSA 发的每一个时间戳都可以依据 genTime 排序，而不必考虑 genTime 精确度。

h) nonce 域如果在 TimeStampReq 中出现，在这里也应出现，值也应等于 TimeStampReq 中的值。

i) tsa 域的目的是为鉴别 TSA 的名字提供一个线索。如果出现，应与验证时间戳的证书里的 subject names 中的一个相同。

j) 扩展(extensions)域是为将来增加额外的信息而采用的一种通常的做法。由 GB/T 20518—2006 定义。特殊的扩展类型可以由组织或者团体自行定义并声明注册。

8.5 保存文件

当需要保存时间戳申请和响应消息在文件中时，文件里应只包含消息的 DER 编码，且不能有额外的消息头和消息尾。

保存在文件中时，时间戳申请消息应该存放在一个扩展名为 .tsq 的文件中，时间戳响应消息应该存放在扩展名为 .tsr 的文件中。

8.6 所用 MIME 对象定义

8.6.1 电子邮件传输

如果使用电子邮件传输时间戳申请和响应消息，可以使用以下 MIME 对象：

a) 申请消息

Content-Type：application/timestamp-query

Content-Transfer-Encoding：base64

<<申请消息的 DER 编码，再用 base64 编码>>

b) 响应消息

Content-Type：application/timestamp-reply

Content-Transfer-Encoding：base64

<<响应消息的 DER 编码，再用 base64 编码>>

8.6.2 HTTP 传输

如果使用 HTTP 协议传输时间戳申请和响应消息且需要用到 MIME 对象，可以使用如下对象：

a) 申请消息

Content-Type：application/timestamp-query

<<申请消息的 DER 编码>>

b) 响应消息

Content-Type：application/timestamp-reply

<<响应消息的 DER 编码>>

8.7 时间戳格式的安全考虑

时间戳格式的使用有如下安全考虑：

a) 请求方在产生 nonce 时应只用一次性随机数，并且不应采用局部时钟来考虑它等待响应的时间。由于中间人攻击会引入延迟，这样任何超过可接收时间的时间戳响应消息都应该被认为是可疑的。因为每种传输方式都有不同的延迟特征，响应时间的可接受程度将依赖于采用的传输方式，以及其他一些具体环境因素。

b) 如果不同实体用同样的数据和同样的散列算法申请时间戳，或者单个实体对同一对象多次申请时间戳，那么它们申请时间戳时将有同样的散列值，结果可能导致观测者存取这些时间戳时产生混乱。TSA 系统和客户端都应该仔细处理这些情况。

c) 无意或故意重放相同消息的情况是可能发生的。当在网络出现问题时，一个或多个副本请求就可能被发送到 TSA，无意重放就可能发生；当中间人重放合法的时间戳响应时，故意重放就会发生。TSA 应该采取有效的技术手段处理这些情况，例如：

1) 采用一次性随机数检查重放，因此 TSA 应采用 nonce 域申请；

2) 请求方采用局部时钟和一个移动的时间窗口，在这个时间窗内记住所有的散列值，当接收到一个响应时，请求方保证响应时间在时间窗内，并且散列值在时间窗内也只产生一次。如果在同一时间窗内，同一个散列值出现两次以上，则应要求请求者或者用一个一次性随机数，或者等到时间窗口移动到散列值只出现一次的时候。

9 时间戳系统的安全

9.1 物理安全

时间戳系统的物理安全要求应遵循 GB/T 20271—2006 中 4.1 的相关要求，保障环境、设备和记录介质的安全。

9.2 软件安全

9.2.1 运行环境

TSA 的所有部件运行的计算机环境，即计算机信息系统，其安全等级应达到或高于 GB 17859—1999 中规定的第二级“系统审计保护级”的要求。

TSA 所有部件应有完善的反毒、防火墙解决方案，系统中不应运行与 TSA 运行无关的程序和服务，可以操作 TSA 部件计算机的人员应受到严格控制，访问 TSA 部件计算机的口令密码也应受到严格控制，确保只有授权人员知道。

9.2.2 可信时间源

TSA 的时间源应是国家标准时间，其或者就是国家权威时间部门发布的时间，或者是用国家权威时间部门认可的硬件和方法获得的时间。

无论采用什么方法获得可信时间，从可信时间源到签名系统的时间传递过程中，应采用严格措施保证时间信息的完整性，保证其没有被任何人篡改。即使有入侵者中途修改了时间信息，签名系统应有能力发现时间信息已被修改，同时向 TSA 管理者发出警报。

用于从时间源接收时间的软件也应检查时间的连续性、完整性，确保其真实有效。

9.2.3 签名系统

签名系统对密钥的所有使用应遵照国家密码管理局有关规范执行。

对签名系统的访问，应有严格的访问控制，包括对密钥的访问控制也应遵照国家密码管理局的有关规范。

签名系统要有完整的审计系统，符合 9.2.5 对审计的要求。

9.2.4 时间戳数据库

时间戳数据库的安全等级应达到或高于 GB/T 20273—2006 中规定的第二级“系统审计保护级”的要求。

9.2.5 审计

9.2.5.1 审计数据产生

TSA 的签名系统应该有完善的审计功能部件，它应对下列事件产生审计记录：

a) 审计功能的启动和结束；

b) 表 1 中的事件

表 1 审计事件

TSA 功能	事　件	附 加 信 息
安全审计	所有对审计变量(如：时间间隔，审计事件的类型)的改变	
	所有删除审计记录的企图	
	对审计日志签名	数字签名，散列结果或认证码应该保存在审计日志之中
本地数据输入	所有的安全相关数据输入系统	若输入的数据与其他数据相关则须验证用户访问相关数据的权限
远程数据输入	所有被系统所接受的安全相关信息	

表 1（续）

TSA 功能	事　　件	附　加　信　息
数据输出	所有对关键的或安全相关的信息进行输出的请求	
私钥载入	部件私钥的载入	
私钥的存储	对为密钥恢复而保存的证书主体私钥的读取	
可信公钥的输入，删除和存储	所有对于可信公钥的改变（如：添加、删除）	包括公钥和与公钥相关的信息
私钥和对称密钥的输出	私钥和对称密钥（包括一次性会话密钥）的输出	
时间戳申请	所有的时间戳申请请求	若申请成功，在日志中保存申请请求和产生的时间戳的拷贝； 若申请失败，在日志中保存失败原因和产生的时间戳失败响应的拷贝
部件的配置	所有的与安全相关的配置	
可信时间的获取和同步	根据可信时间源同步时间	包括如果可信时间和本地时间不匹配时，根据可信时间改变本地时间，以及同步过程中发生的所有错误

对于表 1 中的每一个事件，其审计记录应包括：事件的日期和时间、用户、事件类型、事件是否成功，以及表中附加信息栏中要求的内容。

日志记录中不得出现明文形式的私钥、非对称密钥和其他安全相关的参数。

审计功能部件应能将可审计事件与发起该事件的系统用户身份相关联。

9.2.5.2　审计查阅

审计功能部件应该为审计员提供查看日志所有信息的能力。

审计功能部件应以适于阅读和解释的方式向阅读者提供日志信息。

9.2.5.3　审计事件存储

审计功能部件应具有以下能力：

a）　受保护的审计踪迹存储，要求审计踪迹的存储受到应有的保护，能防止对审计记录的非授权修改，并可以检测对审计记录的修改；

b）　防止审计数据丢失，要求当审计踪迹存储已满时，审计功能部件应能够阻止所有审计事件的发生，除非该事件是由审计员发起的。

9.2.5.4　可信的时间

审计功能部件应该在每一条审计记录上都加上正确的时间，时间同样也应该来源于可信时间源。

9.2.5.5　审计日志签名

对审计日志的签名应遵循如下规范：

a）　审计功能部件应定期让 TSA 给审计日志加盖时间戳；

b）　时间戳签名的对象是从上次生成时间戳后加入的所有审计日志条目以及上次签名的时间戳的值；

c）　审计功能部件给审计日志加盖时间戳的时间周期应该是可以配置的；

d）　对审计日志做时间戳的事件应写入日志中，时间戳应包含在其中。

参 考 文 献

[1] Housley, R., Ford, W., Polk, W. and D. Solo, "Internet X.509 Public Key Infrastructure, Certificate and CRL Profile", RFC 2459, January 1999.

[2] C. Adams, BBN, D. Pinkas, Integris, R. Zuccherato, Entrust, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001

[3] Digital Signature Standard. FIPS Pub 186. National Institute of Standards and Technology. 19 May 1994.

[4] Hoffman, P., "Enhanced Security Services for S/MIME", RFC 2634, June 1999.

[5] ISO/IEC 10181-5: Security Frameworks in Open Systems. Non-Repudiation Framework. April 1997.

[6] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[7] Secure Hash Standard. FIPS Pub 180-1. National Institute of Standards and Technology. 17 April 1995.

[8] "Information-Information technology -Securitytechniques-Time-stampings Service". ISO/IEC 18014, 1 October 2002.

中 华 人 民 共 和 国
国 家 标 准
信息安全技术 公钥基础设施
时间戳规范
GB/T 20520—2006
*
中 国 标 准 出 版 社 出 版 发 行
北京西城区复兴门外三里河北街 16 号
邮政编码:100045
http://www.spc.net.cn
电话:(010)51299090、68522006
2007 年 1 月第一版
*
书号:155066·1-28704

GB/T 20520-2006