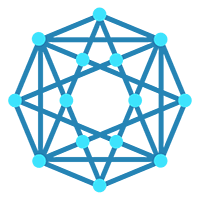
# 可信时间戳

什么是可信时间戳? 有什么用呢？



时间戳是自1970年1月1日（00:00:00 GMT）至当前时间的总秒数。它也被称为Unix时间戳（Unix Timestamp）。

Unix时间戳(Unix timestamp)，或称Unix时间(Unix time)、POSIX时间(POSIX time)，是一种时间表示方式，定义为从格林威治时间1970年01月01日00时00分00秒起至现在的总秒数。

Unix时间戳不仅被使用在Unix系统、类Unix系统中(比如Linux系统)，也在许多其他操作系统中被广泛采用。是从1970年1月1日（UTC/GMT的午夜）开始所经过的秒数，不考虑闰秒。

UNIX时间戳的0按照ISO 8601规范为 ：1970-01-01T00:00:00Z.

一个小时表示为UNIX时间戳格式为：3600秒；一天表示为UNIX时间戳为86400秒，闰秒不计算。

在大多数的UNIX系统中UNIX时间戳存储为32位，这样会引发2038年问题或Y2038。

时间戳（time-stamp）是一个经加密后形成的凭证文档，它包括三个部分：

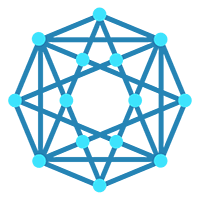
（1）需加时间戳的文件的摘要（digest）；

（2）DTS收到文件的日期和时间；

（3）DTS的数字签名。

一般来说，时间戳产生的过程为：用户首先将需要加时间戳的文件用Hash编码加密形成摘要，然后将该摘要发送到DTS，DTS在加入了收到文件摘要的日期和时间信息后再对该文件加密（数字签名），然后送回用户。

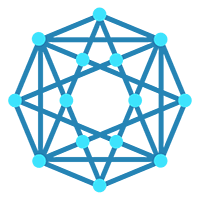
书面签署文件的时间是由签署人自己写上的，而数字时间戳则不然，它是由认证单位DTS来加的，以DTS收到文件的时间为依据。



## 时间戳分类（引用百度百科）

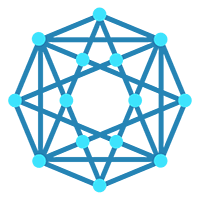
1.自建时间戳：此类时间戳是通过时间接收设备（如GPS，CDMA，北斗卫星）来获取时间到时间戳服务器上，并通过时间戳服务器签发时间戳证书。此种时间戳可用来企业内部责任认定，在法庭认证时并不具备法律效力。因其在通过时间接收设备接收时间时存在被篡改的可能，故此不能做为法律依据。

2.具有法律的效力的时间戳：它是由我国中科院国家授时中心与北京联合信任技术服务有限公司负责建设的我国第三方可信时间戳认证服务。由国家授时中心负责时间的授时与守时监测。因其守时监测功能而保障时间戳证书中的时间的准确性和不被篡改。获取时间戳平台有“大众版权保护平台”，可与我国中科院国家授时中心时间同步。



## 时间戳的用途（引用百度百科）

数字时间戳技术是数字签名技术一种变种的应用。在电子商务交易文件中，时间是十分重要的信息。在书面合同中，文件签署的日期和签名一样均是十分重要的防止文件被伪造和篡改的关键性内容。数字时间戳服务（DTS：digital time stamp service）是网上电子商务安全服务项目之一，能提供电子文件的日期和时间信息的安全保护。



## 法律效力（引用百度百科）

自建时间戳不具备法律效力。

可信时间戳即由国家法定时间源来负责保障时间的授时和守时监测，任何机构包括时间戳中心自己不能对时间进行修改以保障时间的权威，只有这样产生的时间戳才具有法律效力。

可信 时间戳司法判例：

首例时间戳司法应用案例判决书发生法律效力附该案件裁判文书

2008年11月25日，深圳市龙岗区法院依据最高法院“知识产权司法保护活动月”的要求公开宣判知识产权纠纷案，其中“利龙湖”一案系国内首例时间戳技术司法应用案例，宣判后双方当事人均未提起上诉，该案判决书已经发生法律效力。

时间戳在医疗领域进程：

2010年11月14日下午，卫生部在国2招第五会议室组织部有关部门和专家召开了"可信时间戳与电子病历法律效力研讨论证会"，医政司、政策法规司、 医院管理研究所、卫生部统计信息中心、协和医院、北大人民医院、北医三院、北京市西城法院法官、中国医院协会自律维权部、国家法官学院、国家授时中心、联合信任时间戳服务中心的有关领导和专家等参加了论证会。

会议听取了国家授时中心副主任窦忠和联合信任时间戳服务中心董事长张昌利的汇报。与会专家对可信时间戳在解决我国电子病历法律效力问题进行了深入的探讨和论证，认为TSA+CA的模式是一种有效解决电子病历法律效力的方法，建议在医院广泛采用，将选择部分试点医院开展试点应用，并作相关技术标准和法规的研究。

2011年5月4日健康报发表《可信时间戳为电子病历法律效力提供保证》的文章。

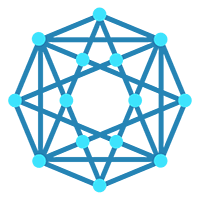
首次颁发档案管理系统接入可信时间戳核准证书

为规范和保障可信时间戳在档案领域的正确使用，日前，国家档案局档案科学技术研究所按照国家档案局技术部审核通过的可信时间戳档案应用方案要求，组织专家对首批3家公司开发的档案管理软件中的可信时间戳接入功能进行了检测，并向通过检测的软件开发商颁发可信时间戳接入核准书。

国家授时中心负责授时和守时保障的联合信任时间戳服务中心将依据接入核准书，向相关档案软件用户提供可信时间戳的服务。

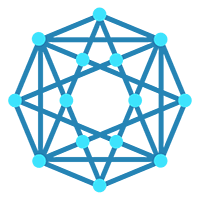
可信时间戳是由联合信任时间戳服务中心签发的一个电子凭证，用于证明电子数据文件自申请可信时间戳后内容保持完整、未被更改。可信时间戳接入核准书的颁发，标志着可信时间戳在档案领域规范化应用已经开始，并将起到电子档案和档案数字化副本内容防篡改、保障档案的法律凭证的作用。

根据《电子签名法》有关数据电文原件形式的要求，申请了可信时间戳认证的电子文件、电子档案或纸质档案的数字化副本等可视为法规规定的原件形式。



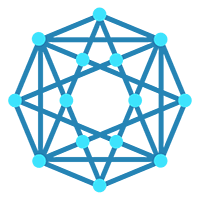
## 时间戳不同语言环境的获取代码

|  |  |
| --- | --- |
| Java | time |
| Java | Math.round(new Date().getTime()/1000)getTime()返回数值的单位是毫秒 |
| Microsoft .NET / C# | epoch = (DateTime.Now.ToUniversalTime().Ticks - 621355968000000000) / 10000000 |
| MySQL | SELECT unix\_timestamp(now()) |
| Perl | time |
| PHP | time() |
| PostgreSQL | SELECT extract(epoch FROM now()) |
| Python | 先 import time 然后 time.time() |
| Ruby | 获取Unix时间戳：Time.now 或 Time.new显示Unix时间戳：Time.now.to\_i |
| SQL Server | SELECT DATEDIFF(s, '1970-01-01 00:00:00', GETUTCDATE()) |
| Unix / Linux | date +%s |
| VB / ASP | DateDiff("s", "01/01/1970 00:00:00", Now()) |
| 其他操作系统(如果Perl被安装在系统中) | 命令行状态：perl -e "print time" |



## 时间戳到普通时间的转换

|  |  |
| --- | --- |
| Java | String date = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm:ss").format(new java.util.Date(Unix timestamp \* 1000)) |
| Java | 先 var unixTimestamp = new Date(Unix timestamp \* 1000) 然后 commonTime = unixTimestamp.toLocaleString() |
| Linux | date -d @Unix timestamp |
| MySQL | from\_unixtime(Unix timestamp) |
| Perl | 先 my $time = Unix timestamp 然后 my ($sec, $min, $hour, $day, $month, $year) = (localtime($time))[0,1,2,3,4,5,6] |
| PHP | date('r', Unix timestamp) |
| PostgreSQL | SELECT TIMESTAMP WITH TIME ZONE 'epoch' + Unix timestamp) \* INTERVAL '1 second'; |
| Python | 先 import time 然后 time.gmtime(Unix timestamp) |
| Ruby | Time.at(Unix timestamp) |
| SQL Server | DATEADD(s, Unix timestamp, '1970-01-01 00:00:00') |
| VB / ASP | DateAdd("s", Unix timestamp, "01/01/1970 00:00:00") |
| 其他操作系统(如果Perl被安装在系统中) | 命令行状态：perl -e "print scalar(localtime(Unix timestamp))" |



## 普通时间到时间戳的转换

|  |  |
| --- | --- |
| Java | long epoch = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm:ss").parse("01/01/1970 01:00:00"); |
| Java | var commonTime = new Date(Date.UTC(year, month - 1, day, hour, minute, second)) |
| MySQL | SELECT unix\_timestamp(time)时间格式: YYYY-MM-DD HH:MM:SS 或 YYMMDD 或 YYYYMMDD |
| Perl | 先 use Time::Local 然后 my $time = timelocal($sec, $min, $hour, $day, $month, $year); |
| PHP | mktime(hour, minute, second, day, month, year) |
| PostgreSQL | SELECT extract(epoch FROM date('YYYY-MM-DD HH:MM:SS')); |
| Python | 先 import time 然后 int(time.mktime(time.strptime('YYYY-MM-DD HH:MM:SS', '%Y-%m-%d %H:%M:%S'))) |
| Ruby | Time.local(year, month, day, hour, minute, second) |
| SQL Server | SELECT DATEDIFF(s, '1970-01-01 00:00:00', time) |
| Unix / Linux | date +%s -d"Jan 1, 1970 00:00:01" |
| VB / ASP | DateDiff("s", "01/01/1970 00:00:00", time) |