INSO
11310-1

1st. Revision

2016

جمهوری اسلامی ایران
Islamic Republic of Iran

سازمان ملی استاندارد ایران
Iranian National Standards Organization

استاندارد ملی ایران

۱۱۳۱۰-۱
تجدیدنظر اول
۱۳۹۴

فناوری اطلاعات

- فنون امنیتی- خدمات مهر زمانی

قسمت ۱: چارچوب

Information technology - Security techniques - Time-stamping services - Part 1:Framework

ICS: 35.080

سازمان ملی استاندارد ایران

تهران، ضلع جنوب غربی میدان ونک، خیابان ولیعصر، پلاک ۲۵۹۲

صندوق‌پستی: ۶۱۳۹-۱۴۱۵۵ تهران- ایران

تلفن: ۵-۸۸۸۷۹۴۶۱

دورنگار: ۸۸۸۸۷۰۸۰ و ۸۸۸۸۷۱۰۳

کرج ، شهر صنعتی، میدان استاندارد

صندوق پستی: ۱۶۳-۳۱۵۸۵ کرج - ایران

تلفن: ۸- ۳۲۸۰۶۰۳۱ (۰۲۶)

دورنگار: ۳۲۸۰۸۱۱۴ (۰۲۶)

رایانامه: standard@isiri.org.ir

وبگاه: http://www.isiri.org

Iranian National Standardization Organization (INSO)

No.1294 Valiasr Ave., South western corner of Vanak Sq., Tehran, Iran

P. O. Box: 14155-6139, Tehran, Iran

Tel: + 98 (21) 88879461-5

Fax: + 98 (21) 88887080, 88887103

Standard Square, Karaj, Iran

P.O. Box: 31585-163, Karaj, Iran

Tel: + 98 (26) 32806031-8

Fax: + 98 (26) 32808114

Email: standard@isiri.org.ir

Website: http://www.isiri.org

به نام خدا

آشنایی با سازمان ملی استاندارد ایران

سازمان ملی استاندارد ایران به موجب بند یک مادهٔ ۳ قانون اصلاح قوانین و مقررات مؤسسهٔ استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱ تنها مرجع رسمی کشور است که وظیفهٔ تعیین، تدوین و نشر استانداردهای ملی (رسمی) ایران را به عهده دارد.

تدوین استاندارد در حوزه‌های مختلف در کمیسیون‌های فنی مرکب از کارشناسان سازمان، صاحب‌نظران مراکز و مؤسسات علمی، پژوهشی، تولیدی و اقتصادی آگاه و مرتبط انجام می‌شود و کوششی همگام با مصالح ملی و با توجه به شرایط تولیدی، فناوری و تجاری است که از مشارکت آگاهانه و منصفانهٔ صاحبان حق و نفع، شامل تولیدکنندگان، مصرف‌کنندگان، صادرکنندگان و واردکنندگان، مراکز علمی و تخصصی، نهادها، سازمان‌های دولتی و غیردولتی حاصل می‌شود. پیش‌نویس استانداردهای ملی ایران برای نظرخواهی به مراجع ذی‌نفع و اعضای کمیسیون‌های مربوط ارسال می‌شود و پس از دریافت نظرها و پیشنهادها در کمیتهٔ ملی مرتبط با آن رشته طرح و در صورت تصویب، به عنوان استاندارد ملی (رسمی) ایران چاپ و منتشر می‌شود.

پیش‌نویس استانداردهایی که مؤسسات و سازمان‌های علاقه‌مند و ذی‌صلاح نیز با رعایت ضوابط تعیین شده تهیه می‌کنند درکمیته ملی طرح، بررسی و درصورت تصویب، به عنوان استاندارد ملی ایران چاپ و منتشر می‌شود. بدین ترتیب، استانداردهایی ملی تلقی می‌شود که بر اساس مقررات استاندارد ملی ایران شمارهٔ ۵ تدوین و در کمیتهٔ ملی استاندارد مربوط که در سازمان ملی استاندارد ایران تشکیل می‌شود به تصویب رسیده باشد.

سازمان ملی استاندارد ایران از اعضای اصلی سازمان بین‌المللی استاندارد (ISO)[۱]، کمیسیون بین‌المللی الکتروتکنیک (IEC)[۲] و سازمان بین‌المللی اندازه‌شناسی قانونی (OIML)[۳] است و به عنوان تنها رابط[۴] کمیسیون کدکس غذایی (CAC)[۵] در کشور فعالیت می‌کند. در تدوین استانداردهای ملی ایران ضمن تـوجه به شرایط کلی و نیازمندی‌های خاص کشور، از آخرین پیشرفت‌های علمی، فنی و صنعتی جهان و استانداردهای بین‌المللی بهره‌گیری می‌شود.

سازمان ملی استاندارد ایران می‌تواند با رعایت موازین پیش‌بینی شده در قانون، برای حمایت از مصرف‌کنندگان، حفظ سلامت و ایمنی فردی و عمومی، حصول اطمینان از کیفیت محصولات و ملاحظات زیست‌محیطی و اقتصادی، اجرای بعضی از استانداردهای ملی ایران را برای محصولات تولیدی داخل کشور و/یا اقلام وارداتی، با تصویب شورای عالی استاندارد، اجباری کند. سازمان می‌تواند به منظور حفظ بازارهای بین‌المللی برای محصولات کشور، اجرای استاندارد کالاهای صادراتی و درجه‌بندی آن را اجباری کند. همچنین برای اطمینان بخشیدن به استفاده‌کنندگان از خدمات سازمان‌ها و مؤسسات فعال در زمینهٔ مشاوره، آموزش، بازرسی، ممیزی و صدورگواهی سیستم‌های مدیریت کیفیت و مدیریت زیست‌محیطی، آزمایشگاه‌ها و مراکز واسنجی (کالیبراسیون) وسایل سنجش، سازمان ملی استاندارد این‌گونه سازمان‌ها و مؤسسات را بر اساس ضوابط نظام تأیید صلاحیت ایران ارزیابی می‌کند و در صورت احراز شرایط لازم، گواهینامهٔ تأیید صلاحیت به آن‌ها اعطا و بر عملکرد آن‌ها نظارت می‌کند. ترویج دستگاه بین‌المللی یکاها، واسنجی وسایل سنجش، تعیین عیار فلزات گرانبها و انجام تحقیقات کاربردی برای ارتقای سطح استانداردهای ملی ایران از دیگر وظایف این سازمان است.

1- International Organization for Standardization
2- International Electrotechnical Commission
3- International Organization for Legal Metrology (Organisation Internationale de Metrologie Legals)
4- Contact point
5- Codex Alimentarius Commission

کمیسیون فنی تدوین استاندارد

«فناوری اطلاعات - فنون امنیتی - خدمات مهرزمانی - قسمت ۱: چارچوب»

«تجدید نظراول »

رئیس :	**سمت و/ یا محل اشتغال**
کمرخانی، حبیب (کارشناسی ارشد فن آوری اطلاعات و ارتباطات- امنیت)	سازمان بنادر و دریانوردی ایران
دبیر:	
بی مانند، هدی (کارشناسی ارشد مهندسی کامپیوتر- نرم افزار)	اداره کل استاندارد استان ایلام
اعضاء : (اسامی به ترتیب حروف الفبا)	
آذرکار، سید علی (کارشناسی ارشد مهندسی کامپیوتر- نرم افزار)	شرکت مهندسی پدیدپرداز
اکبری، علی (کارشناسی مهندسی برق—الکترونیک)	سازمان هلال احمر استان ایلام
بهادری، سندس (کارشناسی ارشد مهندسی کامپیوتر- نرم افزار)	عضو هیئت علمی دانشگاه آزاد اسلامی واحد ایلام
حیدری، نرگس (کارشناسی ارشد مهندسی کامپیوتر- نرم افزار)	عضو هیأت علمی دانشگاه آزاد اسلامی واحد ایلام
سالاری، معصومه (کارشناسی مهندسی کامپیوتر)	کارشناس جهاد دانشگاهی ایلام .
عبدی، اسرا (کارشناسی مترجمی زبان اتگلیسی)	کارشناس جهاد دانشگاهی ایلام

فهرست مندرجات

عنوان	صفحه
آشنایی با سازمان ملی استاندارد ایران	ج
کمیسیون فنی تدوین استاندارد	د
پیش‌گفتار	ز
۱ هدف و دامنه کاربرد	۱
۲ مراجع الزامی	۱
۳ اصطلاحات و تعاریف	۲
۴ نمادها و عبارات مخفف	۷
۵ کلیات	۷
۵-۱ پیش زمینه و خلاصه	۷
۵-۲ خدمات درگیر با مهر زمانی	۸
۵-۳ هستارهای فرایند مهر زمانی	۸
۵-۴ استفاده از مهر زمانی	۹
۵-۵ تولید نشان مهر زمانی	۹
۵-۶ تصدیق نشان مهر زمانی	۱۰
۵-۷ تجدید مهر زمانی	۱۰
۶ ارتباطات بین هستارهای درگیر	۱۱
۶-۱ تراکنش درخواست مهر زمانی	۱۱
۶-۲ تراکنش بررسی مهر زمانی	۱۲
۷ قالب‌های پیام	۱۲
۷-۱ درخواست مهر زمانی	۱۳
۷-۲ پاسخ مهر زمانی	۱۴
۷-۳ تصدیق مهر زمانی	۱۶
۷-۴ فیلدهای تعمیمی	۱۶
۷-۴-۱ توسعه ExtHash	۱۶
۷-۴-۲ توسعه ExtMethod	۱۷
۷-۴-۳ توسعه	۱۷
پیوست الف (الزامی) پودمان ASN.1 برای مهر زمانی	۱۸
پیوست ب (الزامی) برگزیدن قواعد پیام رمزنگاری	۲۳
کتاب‌نامه	۳۴

پیش گفتار

استاندارد «فناوری اطلاعات - فنون امنیتی - خدمات مهرزمانی - قسمت ۱: چارچوب» که نخستین بار در سال ۱۳۸۷ تدوین و منتشر شد، بر اساس پیشنهادهای دریافتی و بررسی و تأیید کمیسیون‌های مربوط برای اولین بار مورد تجدیدنظر قرار گرفت و در اجلاس سیصد و نود و سومین کمیته ملی استاندارد فناوری اطلاعات مورخ ۹۴/۱۲/۴ تصویب شد. اینک این استاندارد به استناد بند یک ماده ۳ قانون اصلاح قوانین و مقررات موسسه استاندارد و تحقیقات صنعتی ایران مصوب بهمن ماه ۱۳۷۱، به‌عنوان استاندارد ملی ایران منتشر می‌شود.

برای حفظ همگامی و هماهنگی با تحولات و پیشرفت‌های ملی و جهانی در زمینه صنایع، علوم و خدمات، استانداردهای ملی ایران در مواقع لزوم تجدید نظر خواهد شد و هر پیشنهادی که برای اصلاح و تکمیل این استانداردها ارایه شود، هنگام تجدید نظر در کمیسیون فنی مربوط مورد توجه قرار خواهد گرفت. بنابراین، باید همواره از آخرین تجدیدنظر استانداردهای ملی استفاده کرد.

این استاندارد جایگزین استاندارد ملی ایران شماره ۱-۱۱۳۱۰، سال ۱۳۸۷ می‌شود.

منبع و ماخذی که برای تهیه این استاندارد مورداستفاده قرار گرفته به شرح زیر است:

ISO 18014-1:2008, Information technology - Security techniques - Time-stamping services Part 1: Framework

فناوری اطلاعات - فنون امنیتی - خدمات مهر زمانی[1] - قسمت ۱: چارچوب

۱ هدف و دامنه کاربرد

هدف از تدوین این استاندارد ملی، تعیین موارد زیر است:

- شناسایی هدف یک مرجع مهر زمانی؛
- توصیف یک مدل عمومی که خدمات مهر زمانی بر اساس آن بنا شده است؛
- تعریف خدمات مهر زمانی؛
- تعیین پروتکل‌های اساسی بین هستارهای درگیرشونده

۲ مراجع الزامی

مدارک الزامی زیر حاوی مقرراتی است که در متن این استاندارد ملی ایران به آن‌ها ارجاع داده شده است. بدین ترتیب آن مقررات جزئی از این استاندارد محسوب می شوند.

در صورتی که به مدرکی با ذکر تاریخ انتشار ارجاع داده شده باشد، اصلاحیه‌ها و تجدید نظرهای بعدی آن مورد نظر این استاندارد ملی ایران نیست. در مورد مدارکی که بدون ذکر تاریخ انتشار به آن‌ها ارجاع داده شده است، همواره آخرین تجدید نظر و اصلاحیه‌های بعدی آن‌ها مورد نظر است.

استفاده از مراجع زیر برای این استاندارد الزامی است:

2-1 ISO 8601, Data elements and interchange formats - Information interchange - Representation of dates and times
2-2 ISO/IEC 10118 (all parts), Information technology - Security techniques - Hash-functions

1- Time-stamping

۳ اصطلاحات و تعاریف

در این استاندارد، اصطلاحات و تعاریف زیر به کار می‌رود.

۳-۱

مرجع صدور گواهی (CA)[1]

مرکز مورداعتماد برای ایجاد و اختصاص گواهی‌های کلید عمومی است.

یادآوری- مرجع صدور گواهی می‌تواند، به شکل اختیاری، کلیدها را ایجاد کند و به هستارها اختصاص دهد.

[منبع: ISO/IEC 11770-1:1996]

۳-۲

تابع چکیده‌ساز[2] مقاوم در برابر تصادم[3]

تابع چکیده‌سازی که ویژگی زیر را برآورده می‌سازد: به لحاظ محاسباتی یافتن دو ورودی متفاوت که به خروجی یکسانی نگاشت شوند، نشدنی است.

یادآوری- امکان‌سنجی محاسباتی وابسته به الزامات امنیتی و محیطی خاص است.

[منبع: ISO/IEC 10118-1:2000[4]]

۳-۳

بازنمود قلم‌های داده[5]

یک قلم داده یا برخی بازنمودهای آن از قبیل مقدار چکیده‌ساز رمزنگاشتی[6] است.

۳-۴

امضای رقمی[7](دیجیتال)

داده الحاقی به یک واحد داده یا تبدیل رمزنگاشتی از یک واحد داده که به گیرنده واحد داده اجازه می‌دهد تا مبدأ و یکپارچگی واحد داده را اثبات کند و از فرستنده و گیرنده واحد داده در برابر جعل از سوی طرف‌های سوم و از فرستنده در برابر جعل از سوی گیرنده محافظت کند.

[منبع: ISO/IEC 11770-3:1999]

1- Certification authority
2- Hash-function
3- Collision-resistant
۴- استاندارد ملی ایران شماره ۱-۱۱۳۱۰با منبع ISO/IEC 10118-1 در سال ۱۳۸۷ نشر شده است
5- Data items representation
6- Cryptographic
7 - Digital signature

۵-۳

اصالت‌سنجی هستار[1]

تأیید آنکه یک هستار همان چیزی است که ادعا می‌کند.

[منبع: ISO/IEC 9798-1:1997]

۶-۳

تابع چکیده‌ساز

تابعی که رشته‌ای از بیت‌ها را به رشته بیت‌هایی با طول ثابت نگاشت می‌کند و دو ویژگی زیر را محقق می‌سازد:

- برای خروجی داده شده، یافتن ورودی که به این خروجی نگاشت شود از لحاظ محاسباتی غیرممکن است.
- برای ورودی داده شده، یافتن ورودی ثانویه‌ای که به همان خروجی نگاشت شود از لحاظ محاسباتی غیرممکن است.

یادآوری- امکان‌سنجی محاسباتی وابسته به الزامات خاص امنیتی و محیطی است.

[منبع: ISO/IEC 10118-1:2000]

۷-۳

مقدار چکیده‌ساز[2]

رشته‌ای از بیت‌ها که خروجی یک تابع چکیده‌سازی است.

یادآوری- این تعریف مشابه تعریف کد چکیده‌ساز در استاندارد ISO/IEC 10118-1:2000 است.

۸-۳

کلید خصوصی[3]

کلیدی از جفت کلید نامتقارن یک هستار که بهتر است فقط توسط همان هستار استفاده شود.

[منبع: ISO/IEC 11770-1:1996]

1- Entity authentication
2- Hash value
3- Private key

۳-۹

کلید عمومی[1]

کلیدی از یک جفت کلید نامتقارن یک هستار که می‌تواند عمومی باشد.

[منبع: ISO/IEC 11770-1:1996]

۳-۱۰

گواهی[2] کلید عمومی

اطلاعات کلید عمومی هستاری که توسط مرجع صدور گواهی امضا شده و به موجب آن به‌صورتی غیرقابل جعل درآمده است.

[منبع: ISO/IEC 11770-1:1996]

۳-۱۱

شماره توالی[3]

پارامتر متغیر با زمان که مقدار آن از توالی مشخص گرفته‌شده و در بازه زمانی معین تکرارناپذیر است.

[منبع: ISO/IEC 11770-1:1996]

۳-۱۲

مهر زمانی

یک پارامتر متغیر با زمان که نقطه‌ای را در زمان نسبت به یک مرجع زمانی مشترک مشخص می‌کند.

[منبع: ISO/IEC 11770-1:1996]

۳-۱۳

تجدید[4] مهر زمانی

فرآیند صدور یک نمود[5] جدید مهر زمانی که دوره اعتبار نمود مهر زمانی اولیه را تمدید می‌کند.

1- Public key
2- Certificate
3- Sequence number
4- Renewal
5- Token

۱۴-۳

درخواست‌کننده[1] مهر زمانی

هستاری با داده‌ایی که می‌خواهد دارای مهر زمانی شود.

یادآوری- یک درخواست‌کننده همچنین می‌تواندطرف سوم مورد اعتمادی شامل مرجع مهر زمانی باشد.

۱۵-۳

نمود افزار مهر زمانی (TST)[2]

ساختار داده‌ای شامل انقیادی[3] قابل تصدیق بین بازنمود قلم‌های داده و یک مقدار زمانی است.

یادآوری- یک نمودافزار مهر زمانی همچنین می‌تواند شامل قلم‌های داده اضافی در زمان انقیاد باشد.

۱۶-۳

تصدیق‌کننده مهر زمانی

هستاری که دارای داده‌ایی است و می‌خواهد تصدیق کند که یک مهر زمانی معتبر مقید شده با آن را دارد.

یادآوری- فرآیند درستی‌سنجی[4] می‌تواند به‌وسیله خود تصدیق‌کننده یا طرف سوم مورد اعتماد انجام شود.

۱۷-۳

مرجع مهر زمانی (TSA)[5]

طرف سوم مورد اعتمادی که برای ارائه خدمت مهر زمانی، مورد اعتماد است.

۱۸-۳

خدمت مهر زمانی (TSS)[6]

خدمت ارائه‌کننده شواهدی مبنی بر اینکه یک قلم داده قبل از یک نقطه معین زمانی وجود داشته است

1- Requester
2- Time-stamp token
3 -Binding
4- Verification
5- Time-stamping authority
6- Time-stamping service

۳-۱۹

پارامتر متغیر با زمان

قلم داده استفاده‌شده توسط هستار به منظور بررسی اینکه پیامی، تکرار مجدد مانند عدد تصادفی، شماره توالی یا مهر زمانی نباشد.

[منبع: ISO/IEC 11770-1:1996]

۳-۲۰

طرف سوم مورد اطمینان (TTP)[1]

مرجع امنیتی یا نماینده آن که از طرف دیگر هستارها نسبت به فعالیت‌های مرتبط با امنیت مورد اعتماد است.

[منبع: ISO/IEC 11770-3:1999]

۳-۲۱

طرح‌واره مرجع زمانی[2]

مفاهیمی برای توصیف مشخصه‌های زمانی اطلاعات جغرافیایی درباره استفاده از یک ساعت اتمی، ساعت سیگنال GPS و غیره است.

یادآوری- به استاندارد ISO 19108:2002 رجوع شود.

۳-۲۲

گسیل سیگنال زمانی[3]

سیگنال‌های زمانی استاندارد که با ارجاع به UTC مطابق با طرح‌واره‌های استاندارد گسیل می‌شوند.

[منبع: ITU-R TF.460-6]

۳-۲۳

خط‌مشی مهر زمانی[4]

مجموعه قواعدی که کاربردپذیری نمود مهر زمانی را برای جامعه‌ای خاص و/یا رده‌ای از برنامه‌های کاربردی با الزامات امنیتی مشترک مشخص می‌کند.

1- Trusted third party
2- Time referencing scheme
3- Time-signal emission
4- Time-stamping policy

۴ نمادها و کوته‌نوشت‌ها

$TS(x_1, x_2, \ldots, x_n)$	تولید نمود مهر زمانی برای داده $x_1, x_2, \ldots, x_n$
D	داده‌ای که دارای مهر زمانی می‌شود
اطلاعات دیگر	اطلاعات مورداستفاده برای تولید نمود مهر زمانی که «TSTInfo» را کمتر از مقدار چکیده‌ساز داده موجود در مهر زمانی قرار می‌دهد.
$T_0, T_1, \ldots T_n$	نقطه زمانی که دارای مهر زمانی می‌شود.
$t_0, t_1, t_2, \ldots t_n$	نقطه زمانی که دارای مهر زمانی می‌شود.
S	نقطه زمانی که در آن پایان امضای رقمی هستار تولید می‌شود.

۵ کلیات

۵-۱ پیش‌زمینه و خلاصه

استفاده از داده‌های رقومی که مجاز است در رسانه‌ای که به‌راحتی قابل اصلاح است تولید شود، موضوع چگونگی تأیید زمانی را که این داده‌ها ایجادشده یا برای آخرین بار تغییر کرده است، مطرح می‌کند. مهر زمانی رقومی باید شواهدی را از به‌هنگام بودن ارائه دهد. مهر زمانی رقومی باید الزامات زیر را برآورده سازد:

- پارامتر متغیر زمانی باید به روشی غیرقابل جعل، به داده مقید شده تا شواهد وجود داده پیش از نقطه معینی از زمان را ارائه کند.
- داده باید به روشی که افشا نشود، ارائه شود.

روش‌های مهر زمانی مشخص‌شده در این استاندارد ملی، این الزامات را از طریق مهر زمانی مقدار چکیده‌ساز داده که کنترل یکپارچگی و محرمانگی را ممکن می‌سازد، برآورده می‌کند. در این حالت خود داده‌ها، افشا نمی‌شوند. چکیده‌سازی داده‌ها با مقدار زمان فعلی توسط TSA مقید می‌شود. این انقیاد یکپارچگی و اعتبار مهر زمانی را نشان می‌دهد. نمود مهر زمانی که ارائه کننده این عناصر است به درخواست‌کننده مهر زمانی ارسال می‌شود.

همچنین مجاز است نمودهای مهر زمانی شامل اطلاعات مرتبط با نمودهای از پیش تولید شده باشد. در اینجا پارامترهای ورودی برای فرآیند مهر زمانی، نمایش داده‌ها و اطلاعات اضافی داده‌های دارای مهر زمانی پیش از درخواست مهر زمانی است. علاوه براین TSA مجاز است قلم‌های داده متعددی را در رابطه با فرآیند مهر زمانی منتشر کند تا شواهدی را مبنی بر در دسترس بودن داده با روشی به‌هنگام بعد از سایر چکیده‌سازی داده‌های دربرگرفته‌شده، ارائه دهد. انتشار پی‌درپی چکیده‌سازی‌ها شواهدی به دست می‌دهد که داده مربوط، پیش از چکیده‌سازی منتشرشده ثانویه وجود داشته است. این رویکرد به تصدیق‌کننده اجازه بررسی یک مهر زمانی بدون درگیر کردن مرجع دیگر را می‌دهد.

۵-۲ خدمات درگیر با مهر زمانی

در مهر زمانی دو عملیات اصلی درگیر هستند:

- فرآیند مهر زمانی که مقادیر زمان را به مقادیر داده مقید می‌کند.
- فرآیند درستی‌سنجی مهر زمانی که صحت انقیادهای رمزنگاشتی را ارزیابی می‌کند.

مرجع مهر زمانی (TSA)، خدمات مهر زمانی را با در نظر گرفتن اینکه فرآیند درستی‌سنجی مهر زمانی می‌تواند درگیر سایر مراجع مورد اعتماد باشد، ارائه می‌دهد.

زمان ارائه‌شده باید الزام کلی دقیق بودن را برآورده سازد؛ خدمتی که زمان را برای TSA ارائه می‌کند، خارج از دامنه کاربرد این استاندارد ملی است.

یادآوری- منابع زمانی معمولاً به گسیل سیگنال زمانی استاندارد بر اساس طرح‌واره‌های مرجع زمان استاندارد وابسته است.

۵-۳ هستارهای فرآیند مهر زمانی

ممکن است هستارهای زیر هنگام درخواست یک مهر زمانی، درگیر شوند:

یک *هستار* داده‌ای دارد که می‌خواهد دارای مهر زمانی شود؛ مثلاً برای داشتن شواهدی مبنی بر وجود داده در یک نقطه معین از زمان. در این حالت هستار به‌عنوان درخواست‌کننده‌ی مهر زمانی عمل می‌کند. هستار همچنین ممکن است شواهدی را درخواست کند که داده‌ی دارای مهر زمانی دریافتی، مهر زمانی معتبری داشته و ممکن است به‌عنوان *تصدیق‌کننده* مهر زمانی عمل کند.

مرجع مهر زمانی (TSA) خدمت دارای مهر زمانی را پیشنهاد می‌کند. ماهیت این خدمت به دلیل کمک به تشخیص اعتبار داده‌ها و خصوصاً اعتبار اجزای رمزنگاشتی مرتبط با این داده‌ها، بسیار حساس است. مرجع مهر زمانی شواهدی را مبنی بر اینکه داده‌ها در یک نقطه معین از زمان وجود داشته‌اند، ارائه می‌دهد و درستی پارامتر زمانی را تضمین می‌کند.

همه هستارهای معرفی‌شده، با پروتکل دست دهی دوطرفه[1] ارتباط برقرار می‌کنند. بدین معنی که هستار، درخواستی را برای TSA ارسال و در برگشت، یک مهر زمانی دریافت می‌کند (جزئیات در بندهای ۵ -۱ و ۵ -۲ مشاهده شود). نمودافزار، اطلاعات کافی را برای دادن اجازه به هستار به‌منظور تصدیق نمودافزار در نقطه بعدی از زمان دارد.

خدمت مهر زمانی ممکن است به‌صورت برخط[2] و برون‌خط[3] عمل کند (به‌عنوان مثال، استفاده از پروتکل ذخیره و ارسال[4]). اختلاف در سطح انتقال پروتکل‌های ارتباطی بین هستارهای درگیر ایجاد شده است.

1- Two-way handshake
2- online
3- offline
4- Store-and-forward protocol

۵-۴ استفاده از مهر زمانی

مهر زمانی، زمان دقیقی را که سند الکترونیکی تولید شده، تغییریافته یا حتی امضاشده نشان نمی‌دهد. هستار ارائه‌کننده یک سند برای مهر زمانی، ممکن است سند را مستقل از TSA امضا کند در حالی که TSA مقدار زمانی را به چکیده‌ساز سند امضاشده مقید می‌کند.

تنها شاهد در دسترس این است که یک سند پیش از مهر زمانی ضمیمه شده، وجود داشته است.

همچنین مهرهای زمانی نقش مهمی را برای اعتبار اسناد امضاشده بازی می‌کنند. سه امکان مختلف برای وقتی که مهر زمانی و امضای داده‌ها ممکن است رخ هد، وجود دارد. داده ممکن است پیش از آنکه درخواست‌کننده‌ی مهر زمانی آن را امضا کند دارای مهر زمانی شود، بعد از تمهید برای امضای فرستنده سند و قبل و بعد از امضا. این امر منجر به نتایج مختلفی در هنگام بررسی به‌موقع اعتبار امضا می‌شود. جدول ۱ این سه امکان را توصیف می‌کند.

جدول ۱ - ترتیب زمانی امضاها و مهرهای زمانی

حالت ۱	t_1	مرجع مهر زمان (TSA) مهر زمانی تولید می‌کند.
	S	درخواست‌کننده، داده‌ها را به همراه مهر زمانی ارائه‌شده، امضا می‌کند.
حالت ۲	S	درخواست‌کننده داده‌ها را امضا می‌کند.
	t_2	مهرهای زمانی TSA، داده را امضا می‌کند.
حالت ۳	t_1	مرجع مهر زمان (TSA) مهر زمانی تولید می‌کند.
	S	درخواست‌کننده، داده‌ها را به همراه مهر زمانی ارائه‌شده، امضا می‌کند.
	t_2	مهرهای زمانی TSA، داده را امضا می‌کند.

از لحاظ فنی:

حالت ۱: در حالتی که امضا، مهر زمانی را دربر می‌گیرد نقطه‌ای از زمان را که داده امضا شده، به‌طور دقیق تعریف نمی‌کند. این حالت بیان می‌کند که امضا پس از آنکه داده دارای مهر زمانی شده، ارائه شده است.

حالت ۲: تصریح می‌کند که داده پیش از نقطه زمانی بیان‌شده، امضا شده است.

حالت ۳: بازه‌ای را که طی آن سند امضاشده، تعریف می‌کند.

۵-۵ تولید نمودافزار مهر زمانی

هنگام تولید نمودافزار مهر زمانی، ابتدا درخواست‌کننده مقدار چکیده‌ساز را برای داده‌هایی که باید دارای مهر زمانی شوند، محاسبه و آن را به TSA موجود درون پیام درخواستی مهر زمانی ارسال می‌کند. مرجع مهر

زمانی مقدار چکیده‌ساز و پیام درخواست مهر زمانی را به مقدار زمان فعلی به‌عنوان نمودافزار مهر زمانی متصل و آن را به درخواست‌کننده ارسال می‌کند.

۵-۶ درستی‌سنجی نمودافزار مهر زمانی

هنگام درستی‌سنجی نمودافزار مهر زمانی، اعتبار نمودافزار مهر زمانی که دربرگیرنده پارامتر زمانی است، تصدیق می‌شود. به شکلی دیگر، ارزیابی درستی نمودافزار مهر زمانی ممکن است به طرف سوم مورد اعتمادی (TTP) محول شود.

۵-۷ تجدید مهر زمانی

داده‌ی دارای مهر زمانی ممکن است دوباره در زمان بعدی دارای مهر زمانی شود. این فرآیند، تجدید مهر زمانی است و ممکن است به‌طور اختیاری توسط TSA پیاده‌سازی شود. برای مثال این فرآیند ممکن است به دلایل زیر نیاز داشته باشد:

- سازوکار استفاده‌شده برای انقیاد مقدار زمان به داده‌ها، نزدیک به پایان چرخه عمر عملیاتی آن است (به‌عنوان مثال: هنگام استفاده از امضای رقمی و گواهی کلید عمومی که در آستانه انقضا است).
- تابع رمزنگاشتی استفاده‌شده برای انقیاد مقدار زمان به داده هنوز قابل‌اعتماد است؛ با این حال، شواهد محکمی مبنی بر آسیب‌پذیری آن در آینده نزدیک وجود دارد (به‌عنوان مثال زمانی که تابعی چکیده‌ساز با حملات جدید و یا قدرت محاسباتی قابل‌دسترس، در آستانه شکستن باشد).
- صادرکننده TSA در حال پایان عملیات به‌عنوان یک ارائه‌دهنده خدمت باشد.

تجدید، نوعی فرآیند پایه‌ای مهر زمانی است که در آن مهر زمانی موجود روی داده‌های رقمیِ قبلی به‌طور مشخصی با داده‌ی محدود شده به مهر زمانی جدید بر روی همان داده‌ی رقمی با مقدار زمانی جدید (فعلی) ترکیب شده است. با ترکیب مهر زمانی از پیش موجود در تولید مهر زمانی جدید و با فرض اینکه ملاحظات امنیتی مناسبی محقق شده باشد، بازه‌ی زمانیِ اعتبارِ مهر زمانیِ اولیه نسبت به داده‌ی رقمی دارای مهر زمانی با پوشش مهر زمانی جدید بسط می‌یابد.

فرض می‌شود داده D در زمان T_0 دارای مهر زمانی شود:

$$TS\ (D, (\text{other info}), T_0)$$

در زمان T_1، در حالی که مهر زمانی قابل‌اعتماد است، تجدید به شکل زیر است:

$$TS\ (D, (TS\ (D, (\text{other info}), T_0), \text{other info}), T_1)$$

هنوز هم وجود D در T_0 ثابت می‌کند که اولین مهر زمانی داده شده در T_1 معتبر است.

تجدید باید قبل از آنکه هر یک از شرایط اشاره شده در بالا باعث بی‌اعتباری مهر زمان شود، انجام گیرد.

در زمان تصدیق یک نمودافزارِ مهر زمانیِ تجدید شده، موارد زیر واقع می‌شوند:

- دورترین نقطه‌ی نمودافزارِ مهر زمانی صادرشده در T_1، در زمان فعلی بررسی شده است.
- نمودافزارِ مهر زمانی ضمیمه‌ی صادرشده در T_0 در همان زمان صدور T_1 مربوط به نمودافزارِ مهر زمانی ضمیمه‌شده، تصدیق شده است.

در رویداد تجدیدهای تو در توی چندگانه، هر نمودافزارِ زمانیِ تو در تو، به جای تصدیق درونی‌ترین نمودافزارِ مهر زمانی، در زمان صدور تجدید بعدی تصدیق شده است.

۶ ارتباطات بین هستارهای درگیرشونده

هستارهای درگیرشونده در فرآیند مهر زمانی، یا هستارهایی هستند که درخواست مهر زمانی می‌کنند یا هستارهایی هستند که مهر زمانی را تصدیق می‌کنند و یک یا چند TSA در طرف دیگر هستند. تراکنش‌های بین این هستارها در بندهای زیر معرفی خواهند شد.

۶-۱ تراکنش درخواست مهر زمانی

ارتباط بین یک هستار (درخواست‌کننده) و TSA هنگام درخواست مهر زمانی، شامل گام‌های زیر است:

درخواست‌کننده مقدار چکیده‌ساز را برای داده‌ای که باید دارای مهر زمانی شود، تولید می‌کند. سازوکار تولید چکیده‌ساز باید یکی از توابع چکیده‌ساز مشخص‌شده در استاندارد ISO/IEC 10118 باشد.

الف) پیام درخواست مهر زمانی به TSA، همراه با داده‌های زیر ارسال می‌شود:

- مقدار چکیده‌ساز،
- الگویتم چکیده‌سازی استفاده‌شده و
- (به‌صورت اختیاری) تک‌بار[1]

ب) مرجع مهر زمانی جامعیت درخواست دریافتی را وارسی می‌کند.

پ) مرجع مهر زمانی مهر زمانی (نمودافزار مهر زمانی) تولید می‌کند. خود مهر زمانی، ساختار داده‌ای شامل موارد زیر است:

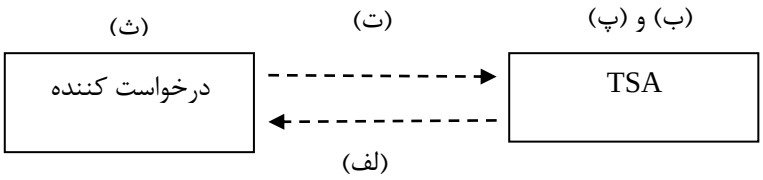
- پارامتر زمانی تولیدشده یا دریافت‌شده از یک منبع قابل‌اعتماد
- داده تحویل‌شده به درخواست‌کننده و
- داده تولید شده توسط TSA برای انقیاد مقدار زمانی با مقدار چکیده‌ساز، الگوریتم چکیده‌سازی و به‌طور اختیاری، تک بار

1- Nonce

ت)- اگر انقیاد رمزنگاشتی از امضاهای رقمی استفاده می‌کند، ممکن است TSA از الگوریتم‌های رمزنگاشتی، آن‌گونه که در استاندارد ISO/IEC 14888 ارائه شده است، استفاده کند. TSA نمودافزار مهر زمانی را به هستار درخواست‌کننده برمی‌گرداند.

ث) هستار ممکن است فوراً جامعیت و صحت نمودافزارِ مهر زمانیِ دریافتی را بررسی کند یا اجازه دهد به‌طور مشروط، طرف مورد اعتماد آن را انجام دهد.

شکل ۱ ارتباطات بین درخواست‌کننده و TSA را نشان می‌دهد. حرف‌گذاری به متن ارجاع داده می‌شود.



شکل ۱ - ارتباطات بین درخواست‌کننده وTSA

۶-۲ تراکنش درستی‌سنجی مهر زمانی

درستی‌سنجی نمودافزارهای تولیدی با استفاده از سازوکارهای نمودافزار مستقل، استفاده از اطلاعات گنجانده‌شده درنمودافزار مهر زمانی تکی را ممکن می‌سازد. تصدیق‌کننده ممکن است نیازمند دریافت اطلاعات اضافی موردنیاز توسط سازوکار به‌منظور تکمیل عملیات درستی‌سنجی باشد؛ این کار ممکن است از طریق هستار درخواست‌کننده یا از طرف هستار توسط TTP دیگری انجام شود.

درستی‌سنجی نمودافزارهای تولیدی با استفاده از سازوکارهای نمودافزار پیوندشده، استفاده از اطلاعات گنجانده‌شده در درنمودافزار مهر زمانی تکی و احتمالاً سایر نمودافزارهای تولید شده توسط TSA را ممکن می‌سازد. تصدیق‌کننده ممکن است نیازمند دریافت اطلاعات اضافی موردنیاز توسط سازوکار به‌منظور تکمیل عملیات درستی‌سنجی باشد؛ این کار ممکن است از طریق هستار درخواست‌کننده یا از طرف هستار توسط TTPدیگری انجام شود.

اطلاعات اضافی در استانداردهای ISO/IEC 18014-2 و ISO/IEC 18014-3 ارائه شده است.

۷ قالب‌های پیام

دو نوع پیام وجود دارد که برای تولید تراکنش‌های معرفی‌شده در بند ۵ موردنیاز است: پیام بین درخواست‌کننده/تصدیق‌کننده مهر زمانی و TSA و پیام بین TSA و درخواست‌کننده/تصدیق‌کننده. همه پیام‌ها در ASN.1 توصیف خواهد شد. یک پودمان کامل ASN.1 در پیوست الف ارائه شده است. پیام‌ها مطابق خدمتی که بازنمود می‌کنند، متمایز خواهند شد.

۷-۱ درخواست مهر زمانی

پیام‌های TimeStampReq توسط هستارها برای درخواست خدمات مهر زمانی از مراجع مهر زمانی اســتفاده می‌شوند. پیام TimeStampReq مانند قالب زیر شکل می‌گیرد:

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

جدول متغیرها و مقادیر آن‌ها را توضیح می‌دهد.

فیلد داده	شرح
version	شماره ویرایش نحو
messageImprint	آن MessageImprint که ارائه‌دهنده خدمت باید آن را به یک مقدار زمانی مقید کند.
reqPolicy	خط‌مشی خدمت درخواست‌شده از سوی TSA صادرکننده نمودافزار مهر زمانی
nonce	درخواست ویژه را شناسایی می‌کند؛ هدف این مقدار ارتباط یک درخواست ویژه با پاسخ متناظر است.
certReq	به TSA سیگنال می‌دهد که اطلاعات گواهی را در صورت وجود ارائه کند.
extensions	دربرگیرنده بسط‌های موردنیاز برای تحقق مناسب عملیات درخواستی مهر زمانی است.

نوع MessageImprint برای کپسوله کردن داده نقش پیام، همراه با یک نشانگر الگوریتم استفاده‌شده در تولید نقش پیام استفاده می‌شود.

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

فیلد داده	شرح
hashAlgorithm	شناسه الگوریتم چکیده‌ساز و مقدار پارامتر
hashedMessage	مقدار چکیده‌سازی متناظر با پیامی که باید دارای مهر زمانی شود، به‌طوری که با تابع چکیده‌سازی مشخص در فیلد داده hashAlgorithm محاسبه شده است.

تابع چکیده‌ساز باید تابع چکیده‌ساز پایداری در برابر تصادم باشد.

TSAPolicyId به‌صورت زیر تعریف می‌شود:

```
TSAPolicyId::= POLICY.&id({TSAPolicies})
```

۷-۲ پاسخ مهر زمانی

پاسخ به درخواست مهر زمانی، ساختار داده‌ای TimeStampResp است که قالب زیر را دارد:

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

ساختار TimeStampToken به‌صورت زیر تعریف می‌شود:

```
TimeStampToken::= SEQUENCE {
contentType CONTENT.&id({Contents})
content [0]
EXPLICIT CONTENT.&Type ({Contents}{@contentType})}
```

این ساختار برای کپسوله کردن ساختار TSTInfo استفاده می‌شود. ساختار TSTInfo به‌صورت زیر تعریف شده است:

```
TSTInfo::= SEQUENCE {
version Version,
policy TSAPolicyId,
messageImprint MessageImprint,
serialNumber SerialNumber,
genTime GeneralizedTime,
accuracy Accuracy OPTIONAL,
ordering BOOLEAN DEFAULT FALSE,
nonce Nonce OPTIONAL,
tsa [0] EXPLICIT GeneralName OPTIONAL,
extensions [1] Extensions OPTIONAL}
```

جدول زیر فیلدهای داده‌هایی را که هنوز تعریف نشده‌اند، توضیح می‌دهد:

فیلد داده	شرح
accuracy	درستی فیلد genTime که با ساعت هماهنگ جهانی (UTC)[a] مقایسه شده است. مرجع مهر زمانی تضمین می‌کند که تفاوت زمان بین UTC و ساعت داخلی خود، با دقت محدود شده است.
genTime	زمانی که TSA در نمودافزار مهر زمانی لحاظ شده است.
serialNumber	serialNumber ممکن است با مقدار صفر تنظیم شود. اگر serialNumber صفر نیست، پس این فیلد یک رقم اختصاص داده‌شده توسط TSA به هر TimeStampToken است که باید برای هر TimeStampTokenصادرشده از TSA داده‌شده، منحصربه‌فرد باشد.

[a] Coordinated Universal Time

ساختار TSTInfo در ساختار TimeStampToken با استفاده از روش کپسوله کردن ContentInfo که مناسب پیاده‌سازی TSA است، کپسوله می‌شود. هنگامی که کپسوله شدن در ساختار ContentInfo که از

ساختار EncapsulatedContentInfo استفاده می‌کند، اتفاق می‌افتد، فیلد eContentType شامل شناسه-های شیء زیر است:

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

به‌علاوه، فیلد eContent شامل ساختار TSTInfo و DER کدگذاری شده به‌عنوان یک رشته هشت‌تایی است.

فیلد GeneralizedTime ترکیبی از قالب پایه برای تاریخ‌های تقویم به شکل کامل و قالب پایه برای ساعت هماهنگ جهانی، مطابق با استاندارد ISO 8601 (با موضوع: اجزای داده‌ها و قالب‌های تبادل – تبادل اطلاعات – تاریخ‌ها و زمان‌ها) است.

این قالب به شکل زیر است:

Y Y Y Y M M D D [T] h h m m s s [, f f f]Z

که هر یک از نویسه‌ها، به‌جز آخری، جایگزین یک رقم تکی است:

YYYY سال را نشان می‌دهد (۰۰۰۰-۹۹۹۹)

MM ماه را نشان می‌دهد (۰۱-۱۲)

DD روز ماه را نشان می‌دهد (۰۱-۳۱)

[T] شاخص زمانی برای مشخص کردن مؤلفه‌ی زمان روز را نشان می‌دهد یا این فیلد ممکن است نادیده گرفته شود.

hh ساعت روز را نشان می‌دهد (۰۰-۲۳)

mm دقیقه‌ی ساعت را نشان می‌دهد (۰۰-۵۹) و

ss ثانیه‌های دقیقه را نشان می‌دهد (۰۰-۵۹)

fff اختصاری برای کسری از ثانیه بدون در نظرگرفتن صفرهای بی‌ارزش است.

نویسه Z (Zulu Time) بیانگر ساعت هماهنگ جهانی (UTC) است.

```
Accuracy::= SEQUENCE{
seconds INTEGER OPTIONAL,
millis [0] INTEGER (1..999) OPTIONAL,
micros [1] INTEGER (1..999) OPTIONAL
}
(ALL EXCEPT ({- none; at least one component shall be present-}))
```

۷-۳ درستی‌سنجی مهر زمانی

پروتکل درستی‌سنجی شبیه پروتکل درخواست مهر زمانی است و شامل پیام درخواست (VerifyReq) و پاسخ مربوطه (VerifyResp) است. ساختار داده با قالب زیر به‌کار می‌رود:

```
VerifyReq::= SEQUENCE {
version Version,
tst TimeStampToken,
requestID [0] OCTET STRING OPTIONAL
}
```

و

```
VerifyResp::= SEQUENCE{
version Version,
status PKIStatusInfo,
tst TimeStampToken,
requestID [0] OCTET STRING OPTIONAL
}
```

فیلد requested درخواست را با پاسخ متناظر متصل می‌کند.

۷-۴ فیلدهای بسط‌یافته

۷-۴-۱ بسط ExtHash

یک درخواست‌کننده خدمات مهر زمانی ممکن است خواستار عرضه مهر زمانی برای بیشتر از مقدار چکیده‌ساز به‌دست‌آمده از قلم داده‌ی واحد باشد.

عرضه مقادیر چکیده‌ساز متعدد حاصل از یک داده‌ی واحد با استفاده از توابع چکیده‌ساز مختلف به درخواست‌کننده این امکان را می‌دهد که نمودافزار مهر زمانی حاصل را از خرابی رمزنگاری هر تابع چکیده‌ساز واحد مجزا کند.

برای فعال‌سازی عرضه مقادیر چکیده‌ساز متعدد، بسط زیر تعریف می‌شود:

```
ExtHash::= SEQUENCE SIZE (1..MAX) OF MessageImprint
tsp-ext-hash::= OBJECT IDENTIFIER { tsp-ext 1{
extHash EXTENSION} =::
SYNTAX ExtHash IDENTIFIED BY tsp-ext-hash
}
```

این بسط هم در فیلد «extensions» از پیام TimeStampReq ارسالی برای درخواست‌کننده به TSA و هم در فیلد «extensions» از نتیجه‌ی ساختار TimeStampToken که توسط TSA شکل‌گرفته، انجام و به درخواست‌کننده برگردانده می‌شود.

اگر این بسط وجود داشته و TSA قادر به پردازش آن باشد، آنگاه TSA باید هم مقدار چکیده‌ساز را در پیام درخواست مهر زمانی که در فیلد messageImprints مشخص شده، مقید کند و هم آن‌هایی را که در این بسط به مقدار زمانی گنجانده شده در نتیجه‌ی نمودافزار مهر زمانی اختصاص دارد، مقید کند.

۷-۴-۲ بسط ExtMethod

درخواست‌کننده خدمات مهر زمانی، ممکن است بخواهد به TSA خاصی نشان دهد که کدام روش مهر زمانی هنگام شکل‌دهی نمودافزار مهر زمانی نهایی استفاده شود. برای فعال‌سازی درخواست‌کننده برای نشان دادن به یک TSA خاص که کدام روش مهر زمانی را در شکل‌دهی نمودافزار مهر زمانیِ نتیجه استفاده کند، بسط زیر تعریف می‌شود:

```
Method::= METHOD.&id ({Methods({
ExtMethod::= SEQUENCE SIZE (1..MAX) OF Method
tsp-ext-meth::= OBJECT IDENTIFIER { tsp-ext 2 }
extMethod EXTENSION =::{
SYNTAX ExtMethod IDENTIFIED BY tsp-ext-meth
}
```

این بسط هم در فیلد «extensions» از پیام TimeStampReq ارسالی برای درخواست‌کننده به TSA و هم در فیلد «extensions» از نتیجه‌ی ساختار TimeStampToken که توسط TSA شکل‌گرفته، انجام و به درخواست‌کننده برگردانده می‌شود.

اگر این بسط وجود داشته باشد آنگاه TSA باید تلاش کند تا درخواست را برای روش خاص در دسترس تکمیل کند و در غیر این‌صورت خطا برگرداند. اگر درخواست‌کننده بیش از یک روش ممکن را مشخص سازد، بهتر است TSA یکی از روش‌های پیشنهادی برای استفاده در شکل‌دهی نمودافزار مهر زمانی را انتخاب کند. اگر این بسط وجود نداشته باشد، TSA از سازوکار مهر زمانی پیش‌فرض خود استفاده می‌کند.

۷-۴-۳ بسط ExtRenewal

درخواست‌کننده از خدمات مهر زمانی ممکن است بخواهد به TSA نشان دهد که درخواست مهر زمانی فعلی یک تجدید مهر زمانی روی‌داده‌ای است که در گذشته دارای مهر زمانی شده است، به‌طوری که مدت اعتبار مهر زمانی قدیمی به‌طور مؤثری افزایش یافته است. برای این منظور، بسط زیر تعریف شده است:

```
extRenewal EXTENSION::= { SYNTAX ExtRenewal IDENTIFIED BY tsp-ext-renewal }
ExtRenewal::= TimeStampToken
tsp-ext-renewal OBJECT IDENTIFIER::= {tsp-ext renewal(3)}
```

این بسط هم در فیلد «extensions» از پیام TimeStampReq ارسالی برای درخواست‌کننده به TSA و هم در فیلد «extensions» از نتیجه‌ی ساختار TimeStampToken که توسط TSA شکل‌گرفته، انجام و به درخواست‌کننده برگردانده می‌شود. TSA محتویات این بسط تغییرنیافته را بازتولید می‌کند.

پیوست الف

(الزامی)

پودمان ASN.1 برای مهر زمانی

این پودمان شامل ASN.1 صحیح مبتنی بر استانداردهای فعلی ASN.1 است که مراحل وارسی را توسط وارسی‌کننده‌ی[1] نحوی قابل‌اطمینانی که در پروژه‌ی ITU-T ASN.1 استفاده می‌شود با موفقیت گذرانده است.

```
TimeStampProtocol {
iso(1) standard(0) time-stamp(18014) modules(0) tsp(1)}
DEFINITIONS IMPLICIT TAGS::= BEGIN
-- EXPORTS All ; --
IMPORTS
-- ISO/IEC 9594-8 | ITU-T Rec. X.509 AuthenticationFramework --
EXTENSION, Extensions
FROM AuthenticationFramework {
joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 4 }
-- ISO/IEC 9594-8 | ITU-T Rec. X.509 CertificateExtensions --
GeneralName
FROM CertificateExtensions {
joint-iso-itu-t ds(5) module(1) certificateExtensions(26) 4 }
AuthenticatedData, SignedData
FROM CryptographicMessageSyntax {
iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16)
modules(0) cms(1) };
time-stamping-services OBJECT IDENTIFIER::=
{ iso(1) standard(0) time-stamp(18014) }
modules OBJECT IDENTIFIER::= { time-stamping-services modules(0) }
tsp-ext OBJECT IDENTIFIER::= { time-stamping-services ext(1) }
TimeStampReq::= SEQUENCE {
version Version,
messageImprint MessageImprint,
reqPolicy TSAPolicyId OPTIONAL,
nonce Nonce OPTIONAL,
certReq BOOLEAN DEFAULT FALSE,
extensions[0] Extensions OPTIONAL
}
MessageImprint::= SEQUENCE {
hashAlgorithm DigestAlgorithmIdentifier,
hashedMessage OCTET STRING
}
DigestAlgorithmIdentifier::= AlgorithmIdentifier {{ DigestAlgorithms }}
```

1- checker

```
DigestAlgorithms ALGORITHM:: = {
{ OID sha1 PARMS NULL },
--
... -- Expect additional digest algorithms --
}
TSA PolicyId:: = POLICY.&id({TSAPolicies})
TSAPolicies POLICY::= {
--
... -- Any supported TSA policy --
}
TimeStampResp::= SEQUENCE {
status PKIStatusInfo,
timeStampToken TimeStampToken OPTIONAL
}
PKIStatusInfo::= SEQUENCE {
status PKIStatus,
statusString PKIFreeText OPTIONAL,
failInfo PKIFailureInfo OPTIONAL
}
PKIStatus::= INTEGER {
granted (0), -- the request is completely granted
grantWithMods (1), -- modifications were necessary, the requester is responsi
ble for asserting the differences
rejection (2), -- the request could not be fulfilled, the failure code
delivers additional information
waiting (3), -- the request is not processed
revocationWarning (4), -- a revocation is imminent
revocationNotification (5) -- notification that a revocation has been occurred
}
PKIFreeText::= SEQUENCE SIZE(1..MAX) OF UTF8String
PKIFailureInfo::= BIT STRING {
badAlg (0), -- unrecognized or unsupported Algorithm Identifier
badRequest (2), -- transaction not permitted or supported
badDataFormat (5), -- data submitted has the wrong format
timeNotAvailable (14), -- the TSAs service is not available
unacceptedPolicy (15), -- the requested TSA policy is not supported
unacceptedExtension (16), -- the requested TSA extension is not supported,
addInfoNotAvailable (17), -- the requested additional information is not
available,
systemNotAvailable (24),-- system is not available
systemFailure (25), -- System Failure
verificationFailure (27) -- verification of time stamp has failed
}
TimeStampToken::= SEQUENCE {
contentType CONTENT.&id({Contents}),
content[0] EXPLICIT CONTENT.&Type({Contents}{@contentType})
}
Contents CONTENT::= {
{ SignedData IDENTIFIED BY id-signedData } |
```

```
{ AuthenticatedData IDENTIFIED BY id-ct-authData } |
{ ETSTInfo IDENTIFIED BY id-data } |
{ DigestedData IDENTIFIED BY id-digestedData },
--
... -- Expect additional time-stamp mechanisms --
}
TSTInfo::= SEQUENCE {
version Version,
policy TSAPolicyId,
messageImprint MessageImprint,
serialNumber SerialNumber OPTIONAL,
genTime GeneralizedTime,
accuracy Accuracy OPTIONAL,
ordering BOOLEAN DEFAULT FALSE,
nonce Nonce OPTIONAL,
tsa [0] EXPLICIT GeneralName OPTIONAL,
extensions [1] Extensions OPTIONAL
}
Version::= INTEGER { v1(1) }
SerialNumber::= INTEGER -- Expect large values
Accuracy::= SEQUENCE {
seconds INTEGER OPTIONAL,
millis[0] INTEGER(1..999) OPTIONAL,
micros [1] INTEGER(1..999) OPTIONAL
}
(ALL EXCEPT({ -- no components present -- }))
Ordering::= BOOLEAN
Nonce::= INTEGER -- Expect large values
-- Time-stamping extensions --
TSExtensions EXTENSION::= {
extHash |
extMethod |
extRenewal,
--
... -- Expect additional extensions --
}
extHash EXTENSION::= {SYNTAX ExtHash IDENTIFIED BY tsp-ext-hash}
ExtHash::= SEQUENCE SIZE(1..MAX) OF MessageImprint
extMethod EXTENSION::= {SYNTAX ExtMethod IDENTIFIED BY tsp-ext-meth}
ExtMethod::= SEQUENCE SIZE(1..MAX) OF Method
Method::= METHOD.&id({Methods})
Methods METHOD::= {
--
... -- Any time-stamping method --
}
extRenewal EXTENSION::= { SYNTAX ExtRenewal IDENTIFIED BY tsp-ext-renewal }
ExtRenewal::= TimeStampToken
EncapsulatedContentInfo::= SEQUENCE {
eContentType CONTENT.&id({EContents}),
```

```
eContent [0] EXPLICIT
CONTENT.&Type({EContents}
{@eContentType})
}
EContents CONTENT::= {
{ ETSTInfo IDENTIFIED BY id-ct-TSTInfo },
--
... -- Expect additional content types --
}
-- Supporting definitions
AlgorithmIdentifier { ALGORITHM:IOSet }::= SEQUENCE {
algorithm ALGORITHM.&id({IOSet}),
parameters ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL
}
ALGORITHM::= CLASS {
&id OBJECT IDENTIFIER UNIQUE,
&Type OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }
CONTENT::= TYPE-IDENTIFIER-- ISO/IEC 8824-2, Annex A
OIDS::= CLASS {
&id OBJECT IDENTIFIER UNIQUE
}
WITH SYNTAX { OID &id }
POLICY::= OIDS -- Supported TSA policies
METHOD::= OIDS -- TSA Methods
-- Information object identifiers
--
tsp-ext-hash OBJECT IDENTIFIER::= { tsp-ext hash(1) }
tsp-ext-meth OBJECT IDENTIFIER::= { tsp-ext meth(2) }
tsp-ext-renewal OBJECT IDENTIFIER::= { tsp-ext renewal(3) }
der OBJECT IDENTIFIER::= {
joint-iso-itu-t asn1(1) ber-derived(2) distinguished-encoding(1) }
sha1 OBJECT IDENTIFIER::= {
iso(1) identified-organization(3) oiw(14) secsig(3) 2 26 }
pkcs7 OBJECT IDENTIFIER::= {
iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7)
}
id-data OBJECT IDENTIFIER::= {
pkcs7 data(1) }
id-signedData OBJECT IDENTIFIER::= {
pkcs7 signedData(2) }
id-ct-authData OBJECT IDENTIFIER::= {
iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
pkcs-9(9) smime(16) ct(1) 2 }
id-ct-TSTInfo OBJECT IDENTIFIER::= {
iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
pkcs-9(9) smime(16) ct(1) 4 }
-- verification of a timestamp token
```

```
VerifyReq::= SEQUENCE {
version Version,
tst TimeStampToken,
requestID [0] OCTET STRING OPTIONAL
}
VerifyResp::= SEQUENCE {
version Version,
status PKIStatusInfo,
tst TimeStampToken,
requestID [0] OCTET STRING OPTIONAL
}
END -- TimeStampProtocol --
```

پیوست ب

(الزامی)

گزیده‌ی نحو پیام رمزنگاشتی

این پیوست، نحو پیام رمزنگاشتی (CMS)[1] را که شامل انواع محتوای موردنیاز برای مهر زمانی است مشخص می‌کند. نحو ASN.1، وادار به استفاده از استانداردهای جاری که در کتاب‌نامه این استاندارد ملی فهرست شده‌اند، می‌کند.

یادآوری- تعاریف CMS و انواع محتوا در RFC 3852 [2] ارائه شده است.

ب-۱ معرفی

نحو پیام رمزنگاشتی (CMS) برای امضای رقمی، خلاصه‌سازی یا تصدیق پیام‌های اختیاری استفاده می‌شود.

نحو پیام رمزنگاری، نحوی برای کپسوله کردن به‌منظور حفاظت از داده‌ها است. این نحو، امضای رقمی و رمزگذاری را پشتیبانی می‌کند و اجازه‌ی کپسوله کردن‌های متعدد را برای پوشش کپسوله‌ای که می‌تواند درون دیگری جای گیرد، می‌دهد. به‌طور مشابه، یک طرف می‌تواند به‌صورت رقمی برخی از داده‌هایی را که پیش‌تر کپسوله شده‌اند به شکل رقمی امضا کند. این نحو همچنین به ویژگی‌های اختیاری مانند زمان امضا کردن، اجازه می‌دهد همراه با محتوای پیام امضا شوند و برای سایر ویژگی‌ها مانند تصدیق امضای دوم[2] این امکان را فراهم می‌کند که با امضایی مرتبط شود.

مقادیر نحو پیام رمزنگاشتی با استفاده از ASN.1 و کدگذاری BER تولید می‌شوند. مقادیر نوعاً به‌صورت رشته‌های هشت‌تایی نمایش داده می‌شوند. در حالی که بسیاری از سامانه‌ها قادر به انتقال رشته‌های هشت‌تایی اختیاری به شکل قابل‌اطمینانی هستند امّا روشن است که بسیاری از سامانه‌های پست الکترونیکی این توانایی را ندارند. این استاندارد سازوکارهای کدگذاری رشته‌های هشت‌تایی را برای انتقال قابل‌اطمینان در چنین محیط‌هایی پوشش نمی‌دهد.

ب-۲ مرور کلی

نحو پیام رمزنگاری (CMS) برای پشتیبانی انواع مختلف محتوا کافی است. این استاندارد محتوای حفاظتی ContentInfo را تعریف می‌کند. این محتوای حفاظتی نوع محتوای شناسایی‌شده‌ی تکی را کپسوله می‌کند و نوع شناسایی‌شده می‌تواند کپسوله‌سازی بیشتری را مهیا سازد.

---

1 - Cryptographic Message Syntax
2 - Countersignature

به‌عنوان یک فلسفه کلی طراحی، هر نوع محتوایی، اجازه پردازش گذر تکی[1] را با کدگذاری با قواعد کدگذاری پایه (BER)[2] با طول نامحدود می‌دهد. عملیات گذر تکی به‌خصوص اگر محتوا بزرگ باشد، روی نوار ذخیره شده باشد یا از فرآیندهای دیگری وارد شده باشد، مفید است. عملیات گذر تکی اشکال مهمی دارد: انجام عملیات کدگذاری با استفاده از کدگذاریِ قواعد کدگذاری ممتاز (DER)[3] در یک گذر تکی مشکل است، زیرا طول مؤلفه‌های مختلف ممکن است از قبل معلوم نباشد. به هر حال ویژگی‌های امضا شده در نوع محتوای داده‌ی امضا شده، نیازمند انتقال در قالب DER است تا اطمینان حاصل شود که گیرندگان می‌توانند محتوایی شامل یک ویژگی ناشناخته یا بیشتر را تصدیق کنند. ویژگی‌های امضا شده، تنها انواع داده استفاده‌شده در CMS هستند که نیازمند کدگذاری DER است.

ب-۳ نحو کلی

شناسه شیء زیر نوع اطلاعات محتوی را شناسایی می‌کند:

```
id-ct-contentInfo OBJECT IDENTIFIER::= { iso(1) member-body(2)(
us(840) rsadsi(113549) pkcs(1) pkcs9(9) smime(16) ct(1) 6}{
```

نحو پیام رمزنگاشتی شناسه نوع محتوا را با محتوا مرتبط می‌سازد. نحو باید نوع ContentInfo از ASN.1 را داشته باشد:

```
ContentInfo::= SEQUENCE {
ContentType ContentType,
content [0] EXPLICIT ANY DEFINED BY contentType }
ContentType::= OBJECT IDENTIFIER}
```

فیلدهای ContentInfo معانی زیر را دارند:

فیلد contentType نوع محتوای مرتبط را نشان می‌دهد. این فیلد شناسه شیء است و یک رشته منحصربه‌فرد از اعداد صحیح اختصاص داده شده توسط مرجعی است که نوع محتوا را تعریف می‌کند.

فیلد content محتوای مرتبط است. نوع محتوا به‌طور منحصربه‌فرد می‌تواند توسط contentType تعیین شود. انواع محتوا برای داده و داده‌های امضاشده در این بخش تعریف شده است.

ب-۴ نوع محتوای داده

شناسه شیء زیر، نوع محتوای داده را شناسایی می‌کند:

```
id-data OBJECT IDENTIFIER::= { iso(1) member-body(2) us(840)
 rsadsi(113549) pkcs(1) pkcs7(7) 1}
```

1 - Single pass processing
2 - Basic Encoding Rules
3 - Distinguished Encoding Rules

نوع محتوای داده برای اشاره به رشته‌های هشت‌تایی اختیاری درنظر گرفته شده است، مانند فایل‌های متنی ASCII و تفسیر آن به برنامه کاربردی واگذار می‌شود. چنین رشته‌هایی نیازی به داشتن ساختار داخلی ندارند (اگرچه آن‌ها می‌توانند تعریف ASN.1 خودشان یا دیگر ساختارها را داشته باشند).

نوع محتوای داده به‌طور کلی در نوع محتوای داده امضاشده کپسوله می‌شود.

ب-۵ نوع محتوای داده امضا شده

نوع محتوای داده امضاشده شامل محتوای هر نوع و مقدار صفر یا مقادیر بیشتر امضاء است. هر تعداد از امضاکنندگان به‌صورت موازی می‌توانند هر نوع محتوا را امضا کنند.

برنامه کاربردی نوعی مربوط به نوع محتوای داده امضاشده، امضای رقمی امضاکننده‌ی محتوای نوع محتوای داده را ارائه می‌دهد. نوع دیگر برنامه کاربردی، گواهی‌ها و فهرست لغو گواهی‌ها (CRL)[1] را منتشر می‌کند.

فرآیندی که با آن داده امضاشده ساخته شده، شامل مراحل زیر است:

- برای هر امضاکننده خلاصه پیام یا مقدار چکیده‌ساز برای محتوا با یک الگوریتم خاص امضاکننده و خلاصه پیام محاسبه می‌شود. اگر امضاکننده هر اطلاعاتی غیر از محتوا را امضا کند، خلاصه پیام محتوا و اطلاعات دیگر با الگوریتم خلاصه پیام امضاکننده (به بند ب-۵-۴ مراجعه شود) خلاصه می‌شوند و نتیجه «خلاصه پیام» خواهد بود.
- برای هر امضاکننده، خلاصه پیام با استفاده از کلید خصوصی امضاکننده به‌صورت رقمی امضا می‌شود.
- همان‌طور که در بند ب-۵-۳ تعریف شده است، برای هر امضاکننده، مقدار امضا و سایر اطلاعات خاص امضاکننده در مقدارSignerInfo جمع‌آوری می‌شوند. گواهی‌ها و فهرست لغو گواهی‌ها و موارد غیر مرتیط با هر امضاکننده، در این مرحله جمع‌آوری می‌شوند.
- همان‌طور که در بند ب-۵-۱ تعریف شده است، الگوریتم‌های خلاصه پیام و مقادیر SignerInfo برای همه امضاکنندگان، همراه با محتوا در یک مقدارSignedData جمع‌آوری شده‌اند.

یک گیرنده به‌طور مستقل خلاصه پیام را محاسبه می‌کند. این خلاصه پیام و کلید عمومی امضاکننده برای تصدیق مقدار امضا استفاده می‌شود. کلید عمومی امضاکننده هم توسط نام متمایز صادرکننده همراه با شماره سریال خاص صادرکننده یا توسط شناسه‌ی کلید موضوعی که به‌طور منحصربه‌فرد، گواهی کلید عمومی را شناسایی می‌کند، ارجاع داده می‌شود. گواهی امضاکننده ممکن است در فیلد گواهی‌های SignedData گنجانده شود.

این بند به ۶ قسمت تقسیم شده است. قسمت اول نوع سطح بالای SignedData را توضیح می‌دهد. قسمت دوم EncapsulatedContentInfo، قسمت سوم نوع اطلاعات SignerInfo هر امضاکننده را توضیح می‌دهد و

1 - Certificate revocation lists

قسمت‌های چهارم، پنجم و ششم به ترتیب محاسبه خلاصه پیام، تولید امضا و فرآیندهای درستی‌سنجی امضا را توضیح می‌دهند.

ب-۵-۱ نوع SignedData

شناسه موضوع زیر نوع محتوای داده امضاشده را شناسایی می‌کند:

```
id-signedData OBJECT IDENTIFIER::= { iso(1) member-body(2)
 us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2 }
```

نوع محتوای داده امضاشده باید نوع SignedData از ASN.1 را داشته باشد:

```
SignedData::= SEQUENCE}
version CMSVersion,
digestAlgorithms DigestAlgorithmIdentifiers,
encapContentInfo EncapsulatedContentInfo,
certificates [0] IMPLICIT CertificateSet OPTIONAL,
crls [1] IMPLICITCertificateRevocationLists
OPTIONAL,
signerInfos SignerInfos{
with
DigestAlgorithmIdentifiers::= SET OF
DigestAlgorithmIdentifier
and
SignerInfos::= SET OF SignerInfo
```

فیلدهای نوع SignedData معانی زیر را دارند:

فیلد version شماره نسخه نحو است. اگر هیچ گواهی ویژگی در فیلد گواهی‌ها موجود نباشد، نوع محتوای کپسوله‌شده شناسه داده است و همه اجزای SignerInfos نسخه ۱ هستند؛ بنابراین مقدار version باید ۱ باشد. به بیانی دیگر، اگر گواهی‌های ویژگی موجود باشند، نوع محتوای کپسوله‌شده غیر از شناسه داده است یا هر جزیی از SignerInfos نسخه ۳ است پس مقدار version باید ۳ باشد.

فیلد digestAlgorithms مجموعه‌ای از شناسه‌های الگوریتم خلاصه پیام است. ممکن است هر تعداد از عناصر در مجموعه که شامل صفر هم است، وجود داشته باشد. هر عنصر، الگوریتم خلاصه پیام را همراه با همه پارامترهای مرتبط که توسط یک امضاکننده یا بیشتر استفاده‌شده است، شناسایی می‌کند. در نظر است تا مجموعه الگوریتم‌های خلاصه پیام به‌کار گرفته‌شده توسط همه امضاکنندگان را با هر ترتیبی فهرست کند تا درستی‌سنجی امضای یک‌گذر[1] را تسهیل کند. فرآیند خلاصه‌سازی پیام در بند ب-۵-۴ توضیح داده شده است.

فیلد encapContentInfo محتوای امضا شده شامل شناسه نوع محتوا و خود محتوا است. جزئیات نوع EncapsulatedContentInfo در بند ب-۵-۲ بحث شده است.

1 - One-pass

فیلد certificates مجموعه‌ای از گواهی‌ها است. سعی شده است تا مجموعه گواهی‌ها برای دربرگرفتن مسیرهای گواهی از یک «ریشه» شناخته شده یا «مرجع صدور گواهی سطح بالا» برای همه امضاکنندگان در فیلد signerInfos کافی باشد. ممکن است گواهی‌هایی بیشتر از نیاز وجود داشته باشند و ممکن است گواهی‌های کافی برای دربرگرفتن مسیرهای گواهی از دو مرجع صدور گواهی سطح بالای مستقل یا بیشتر وجود داشته باشد. همچنین در حالتی که انتظار می‌رود گیرندگان، ابزاری متفاوت برای به دست آوردن گواهی‌های لازم داشته باشند (به‌طور مثال از مجموعه گواهی‌های پیشین)، ممکن است گواهی‌هایی کمتر از تعداد مورد نیاز وجود داشته باشد. گواهی امضاکنندگان ممکن است ضمیمه شده باشد. استفاده از نسخه ۱ گواهی‌های ویژگی به‌شدت مأیوس‌کننده است.

فیلد crls مجموعه‌ای از فهرست‌های لغو گواهی است. سعی می‌شود تا مجموعه‌ای شامل اطلاعات کافی برای تعیین اینکه آیا گواهی‌ها در فیلد گواهی‌ها معتبر هستند موجود باشد، اما چنین تناظری لازم نیست. فهرست‌های لغو گواهی منبع اصلی اطلاعات وضعیت لغو هستند. ممکن است فهرست‌های لغو گواهی بیشتر از نیاز و همچنین ممکن است کمتر از نیاز موجود باشد.

فیلد signerInfos مجموعه‌ای از اطلاعات به ازای هر امضاکننده است. ممکن است هر تعداد از عناصر در مجموعه که شامل صفر هم است موجود باشد. جزئیات نوع SignerInfo در بند ب-۵-۳ بحث شده است.

ب-۵-۲ نوع EncapsulatedContentInfo

محتوی در نوع EncapsulatedContentInfo نشان داده می‌شود:

```
EncapsulatedContentInfo::= SEQUENCE {
eContentType ContentType,
eContent [0] EXPLICIT OCTET STRING OPTIONAL }
ContentType::= OBJECT IDENTIFIER
```

فیلدهایی از نوع EncapsulatedContentInfo معانی زیر را دارد:

فیلد eContentType شناسه شیء است. شناسه شیء به‌صورت منحصربه‌فرد نوع محتوا را تعیین می‌کند. فیلد eContent خودش محتوایی است که به‌عنوان یک رشته هشت‌تایی نگهداری می‌شود. فیلد eContent نیازی به DER کدشده ندارد.

حذف اختیاری eContent در فیلد EncapsulatedContentInfo ایجاد «امضاهای خارجی» را ممکن می‌سازد. در مورد امضاهای خارجی، محتوایی که امضا می‌شود در مقدار EncapsulatedContentInfo گنجانده شده در نوع محتوای داده امضا شده، وجود ندارد. اگر مقدار eContent در EncapsulatedContentInfo وجود نداشته باشد، آنگاه signatureValue محاسبه و به eContentType اختصاص داده می‌شود، چنان‌که مقدار eContent وجود دارد.

در حالت بدی که هیچ امضاکننده‌ای وجود ندارد، مقدارEncapsulatedContentInfo که «امضاشده» است بی‌ربط است. در این حالت بهتر است نوع محتوایی که با مقدار EncapsulatedContentInfo «امضاشده»،

داده شناسه باشد (همان‌طور که در بند ب-۴ تعریف شد) و بهتر است فیلد محتوای مقدار EncapsulatedContentInfo حذف شود.

ب-۵-۳ نوع SignerInfo

اطلاعات برای هر امضاکننده در نوع SignerInfo نمایش داده می‌شود:

```
SignerInfo::= SEQUENCE {
version CMSVersion,
sid SignerIdentifier,
digestAlgorithm DigestAlgorithmIdentifier,
signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL,
signatureAlgorithm SignatureAlgorithmIdentifier,
signature Signature  Value,
unsignedAttrs [1] IMPLICIT UnsignedAttributes OPTIONAL{
```

با

```
SignerIdentifier::= CHOICE {
issuerAndSerialNumber IssuerAndSerialNumber,
subjectKeyIdentifier [0] SubjectKeyIdentifier{
SignedAttributes::= SET SIZE (1..MAX) OF Attribute
UnsignedAttributes::= SET SIZE (1..MAX) OF Attribute
Attribute::= SEQUENCE {
attrType OBJECT IDENTIFIER,
attrValues SET OF AttributeValue{
AttributeValue::= ANY
SignatureValue::= OCTET STRING
```

فیلدهای از نوع SignerInfo معانی زیر را دارند:

فیلد version شماره نسخه نحو است. اگر فیلد SignerIdentifier انتخاب issuerAndSerialNumber باشد آنگاه مقدار version باید ۱ باشد. اگر SignerIdentifier همان subjectKeyIdentifier باشد، آنگاه version باید ۳ باشد.

فیلد sid گواهی امضاکننده را مشخص می‌کند (و کلید عمومی امضاکننده). کلید عمومی امضاکننده برای تصدیق امضا توسط گیرنده موردنیاز است. فیلد SignerIdentifier دو جایگزین را برای مشخص کردن کلید عمومی امضاکننده مهیا می‌کند. مورد اول issuerAndSerialNumber، گواهی امضاکننده را توسط نام متمایز منتشرکننده و شماره‌سریال گواهی شناسایی می‌کند. subjectKeyIdentifier گواهی امضاکننده را توسط شناسه کلید شناسایی می‌کند. وقتی که به گواهی X.509 ارجاع می‌شود، شناسه کلید مقدار بسط X.509 subjectKeyIdentifier را تطبیق می‌دهد.

فیلد digestAlgorithm الگوریتم خلاصه پیام و هر پارامتر مرتبط استفاده‌شده توسط امضاکننده را شناسایی می‌کند. خلاصه پیام یا برای محتوایی که امضا می‌شود محاسبه شده یا برای محتوایی با ویژگی‌های امضا

شـده در فرآینـدهـای بنـد ب-۵-۴. الگـوریتم خلاصـه پیـام بایـد بـین مـوارد فهرسـت شـده در فیلـد digestAlgorithms از SignerData مربوط باشد.

فیلد signedAttrs مجموعه‌ای از ویژگی‌های امضا شده است. این فیلد اختیاری است، اما اگـر نـوع محتـوای مقـدار EncapsulatedContentInfo کـه امضـا شـده، شناسـه داده نیسـت، ایـن فیلـد بایـد موجـود باشـد. SignedAttribute باید کدگذاری DER باشد حتی اگر بقیه‌ی ساختار با BER کدگـذاری شـده باشـد. اگـر فیلد موجود باشد، باید دست کم شامل دو ویژگی زیر باشد:

ویژگی content-type مقدار نوع محتوای خودش در مقـدار EncapsulatedContentInfo را کـه امضـا شـده است، دارد. بند ب-۶-۱ ویژگی content-type را تعریف می‌کند. ویژگی content-type همان‌طور که در بنـد ب-۶ -۳ تعریف شده است نباید به‌عنوان بخشی از ویژگی امضانشده‌ی امضای دوم، استفاده شود.

ویژگی message-digest همان مقدار خلاصه پیام محتوا را دارد. بند ب-۶-۲ ویژگی خلاصه پیـام را تعریـف می‌کند.

فیلد signatureAlgorithm الگوریتم امضا و هر پارامتر مرتبط مورداستفاده توسط امضاکننده را بـرای تولیـد امضای رقمی شناسایی می‌کند.

فیلد signature نتیجه تولید امضای رقمی با استفاده از خلاصـه پیـام و کلیـد خصوصـی امضـاکننده اسـت. جزییات امضا بستگی به الگوریتم امضای به‌کار گرفته‌شده دارد.

فیلد unsignedAttributes مجموعه‌ای از ویژگی‌های امضا نشده است. این فیلد اختیاری است. انواع ویژگی-های مفید مانند امضاهای دوم در بند ب-۶ تعریف شده‌اند.

فیلدهای نوع SignedAttributes و UnsignedAttributes معانی زیر را دارند:

attrType نوع ویژگی رانشان می‌دهد و یک شناسه شیء است.

فیلد attrValues مجموعه مقادیری است که ویژگی را در بردارد. نوع هر مقدار در مجموعه می‌تواند به‌صورت منحصربه‌فرد توسط attrType تعیین شود. فیلد attrValues می‌توانـد محـدودیت‌هـایی را روی تعـداد اقـلام مجموعه تحمیل کند.

ب-۵-۴ فرایند محاسبه خلاصه پیام

فرایند محاسبه خلاصه پیام، خلاصه پیام را برای محتوای امضا شده یا محتوای همراه بـا ویژگـی امضـا شـده محاسبه می‌کند. در هر دو حالت، ورودی اولیه برای فرآیند محاسبه خلاصه پیام، «مقدار» محتوای کپسـوله-شده‌ی است که امضا مـی‌شـود. بـه‌طـور مشـخص، ورودی اولیـه encapContentInfo eContent OCTET STRING است که به آن فرآیند امضا اعمال می‌شود. تنها هشت‌تایی‌های شامل مقدار eContent OCTET STRING، ورودی الگوریتم خلاصه پیام هستند نه برچسب یا طول رشته‌ی هشت‌تایی.

نتیجه فرآیند محاسبه خلاصه پیام بستگی به آن دارد که آیا فیلد signedAttrs موجود است یا خیر. هنگامی‌که فیلد موجود نباشد نتیجه فقط خلاصه پیام محتواست که در بالا توضیح داده شد. هنگامی‌که فیلد موجود است نتیجه، خلاصه پیام کدگذاری DER کامل مقدار SignedAttributes گنجانده شده در فیلد signedAttrs است. از آنجا که مقدار SignedAttributes در صورت وجود باید شامل مقدار محتوا و ویژگی‌های خلاصه پیام باشد آن مقادیر به‌طور غیرمستقیم در نتیجه گنجانده شده است. آن‌طور که در بند ب-۶ -۳ تعریف شد، ویژگی نوع محتوا هنگامی‌که به‌عنوان بخشی از یک ویژگی امضای دوم امضانشده استفاده می‌شود موردنیاز نیست. کدگذاری جداگانه‌ی فیلد signedAttrs برای محاسبه خلاصه پیام انجام می‌شود.

نشان[1] [0] IMPLICIT در فیلد signedAttrs برای کدگذاری DER استفاده نشده، بلکه نشان EXPLICIT SET OF استفاده می‌شود. کدگذاری DER از نشان EXPLICIT SET OF به‌جای نشان [0] IMPLICIT که در محاسبه خلاصه پیام با طول و هشت‌تایی‌های محتوای مقدار SignedAttributes گنجانده شده، استفاده می‌شود.

هنگامی‌که فیلد signedAttrs وجود نداشته باشد، تنها هشت‌تایی‌های شامل مقدار signedData encapContentInfo eContent OCTET STRING (به‌طور مثال محتویات یک فایل)، به عنوان ورودی برای محاسبه خلاصه پیام است. مزیتش این است که نیازی به دانستن طول محتوای امضا شده در فرایند تولید امضا نیست.

اگرچه برچسب encapContentInfo eContent OCTET STRING و هشت‌تایی‌های طول در محاسبه خلاصه پیام گنجانده نشده‌اند اما هنوز توسط سایر ابزار محافظت می‌شوند. هشت‌تایی‌های طول توسط ماهیت الگوریتم خلاصه پیام حفاظت شده‌اند، زیرا از نظر محاسباتی یافتن دو نوع محتوای پیام مشخص با هر طولی که خلاصه پیام یکسان داشته باشند، امکان‌پذیر نیست.

ب-۵-۵ فرآیند تولید امضا

ورودی فرایند تولید امضا، شامل نتیجه فرآیند محاسبه خلاصه پیام و کلید خصوصی امضاکننده است. جزئیات تولید امضا بستگی به الگوریتم امضای به کار گرفته‌شده دارد. شناسه موضوع همراه با هر پارامتری که الگوریتم امضای به کار گرفته‌شده توسط امضاکننده را تعیین می‌کند، در فیلد signatureAlgorithm نگه‌داری می‌شود. مقدارِ امضای تولید شده توسط امضاکننده، به‌عنوان OCTET STRING کدگذاری و در فیلد امضا نگه‌داری می‌شود.

ب-۵-۶ فرآیند درستی‌سنجی امضا

ورودی فرایند درستی‌سنجی امضا شامل نتیجه فرایند محاسبه خلاصه پیام و کلید عمومی امضاکننده است. گیرنده ممکن است کلید عمومی صحیح برای امضاکننده را از هر طریقی به دست آورد، اما روش ارجح،

1 - Tag

استفاده از تأییدیه‌ی حاصل از فیلد تأییدیه‌های SignedData است. انتخاب و اعتبار کلید عمومی امضاکننده ممکن است مبتنی بر اعتبارسنجی مسیر گواهی (به کتاب‌نامه [۱] مراجعه شود) به‌علاوه سایر زمینه‌های خارجی باشد اما خارج از دامنه کاربرد این استاندارد است. جزئیات درستی‌سنجی امضا بستگی به الگوریتم امضای به‌کاررفته دارد.

گیرنده ممکن است به مقادیر خلاصه پیام محاسبه شده توسط صادرکننده پیام، اعتماد نکند. اگر signedData signerInfo شامل signedAttributes باشد آنگاه خلاصه پیام محتوا باید همان‌طور که در بند ب-۵-۴ توضیح داده شد، محاسبه شود. برای آنکه امضا معتبر باشد مقدار خلاصه پیام محاسبه شده توسط گیرنده باید همان مقدار ویژگی messageDigest گنجانده شده در signedAttributes از signedData signerInfo باشد.

ب-۶ ویژگی‌های مفید

این بند ویژگی‌های مفیدی را که ممکن است با داده‌ی امضا شده استفاده شود، تعریف می‌کند. ویژگی‌ها به ترتیب خاصی فهرست نشده‌اند.

ب -۶-۱ نوع محتوا

نوع ویژگی مرتبط با نوع محتوا نوع محتوای ContentInfo را با داده‌ی امضاشده مشخص می‌کند، البته اگر ویژگی‌های تأییدشده موجود باشد، نوع ویژگی مرتبط با نوع محتوا موردنیاز است.

ویژگی نوع محتوا باید ویژگی امضا شده یا ویژگی تأییدشده‌ای باشد و نمی‌تواند ویژگی امضا نشده، ویژگی تأیید نشده یا ویژگی محافظت‌نشده‌ای باشد.

شناسه شیء زیر ویژگی نوع محتوا را شناسایی می‌کند:

```
id-contentType OBJECT IDENTIFIER::= { iso(1) member-body(2)
 us(840) rsadsi(113549) pkcs(1) pkcs9(9) 3}
```

مقادیر ویژگی نوع محتوا، نوع ContentType از ASN.1 را دارد:

```
ContentType::= OBJECT IDENTIFIER
```

ویژگی نوع محتوا باید مقدار ویژگی واحدی داشته باشد، حتی اگر نحو آن به‌عنوان SET OF AttributeValue تعریف شده باشد. مقدار صفر(۰) یا نمونه‌های چندگانه‌ی AttributeValue مجاز نیستند.

قواعد SignedAttributes و AuthAttributes هریک به‌عنوان SET OF Attributes تعریف شده‌اند. SignedAttributes در signerInfo نباید شامل نمونه‌های چندگانه‌ی ویژگی نوع محتوا باشد.

به‌طور مشابه AuthAttributes در AuthenticatedData نباید شامل نمونه‌های چندگانه‌ی ویژگی نوع محتوا باشد.

ب-۶-۲ خلاصه پیام

نوع ویژگی خلاصه پیام، خلاصه پیام encapContentInfo eContent OCTET STRING را که در داده امضاشده، امضا می‌شود مشخص می‌کند (به بند ب-۵-۴ رجوع شود). برای داده‌ی امضاشده، خلاصه پیام با استفاده از الگوریتم خلاصه پیام امضاکننده محاسبه می‌شود.

در داده‌ی امضاشده، نوع ویژگی امضا شده‌ی خلاصه پیام باید وقتی که هر یک از ویژگی‌ها موجود هستند، وجود داشته باشد.

ویژگی خلاصه پیام باید ویژگی امضا شده باشد و نمی‌تواند ویژگی امضا نشده باشد.

شناسه موضوع زیر ویژگی خلاصه پیام را شناسایی می‌کند:

id-messageDigest OBJECT IDENTIFIER::=
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9) 4 }

مقادیر ویژگی خلاصه پیام، نوع MessageDigest از ASN.1 را دارد.

MessageDigest::= OCTET STRING

ویژگی نوع محتوا باید مقدار ویژگی واحدی داشته باشد، حتی اگر نحو آن به‌عنوان SET OF AttributeValue تعریف شده باشد. نباید مقادیر صفر یا نمونه‌های چندگانه‌ی AttributeValue وجود داشته باشند..

قواعد SignedAttributes به‌عنوان یک SET OF Attributes تعریف می‌شود. SignedAttributes در signerInfo باید فقط یک نمونه از ویژگی خلاصه پیام باشد.

ب-۶-۳ امضای دوم

نوع ویژگی امضای دوم، یک امضا یا بیشتر را برای هشت‌تایی‌های محتوای امضای OCTET STRING از یک مقدار SignerInfo را در داده امضا شده تعیین می‌کند. بنابراین، نوع ویژگی امضای دوم، امضای دیگر را امضا می‌کند (به‌صورت سری امضا می‌کند).

ویژگی امضای دوم باید ویژگی امضا نشده باشد و نمی‌تواند ویژگی امضا شده‌ای باشد.

شناسه شیء زیر ویژگی امضای دوم را شناسایی می‌کند:

id-count ersignature OBJECT IDENTIFIER:: = { iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) pkcs9(9) 6}

مقادیر ویژگی امضای دوم دارای نوع ASN.1 از Countersignature است:

Countersignature::= SignerInfo

مقادیر امضای دوم دارای همان مقادیر SignerInfo برای امضاهای معمولی است، به‌جز اینکه:

۱- فیلد signedAttributes نباید شامل ویژگی نوع محتوا باشد در حالی که نوع محتوا برای امضاهای دوم وجود ندارد.

۲- فیلد signedAttributes در صورتی که شامل هر ویژگی دیگری باشد، باید ویژگی خلاصه پیام را داشته باشد.

۳- ورودی فرآیند خلاصه کردن پیام، هشت‌تایی‌های محتوای کدگذاری DER فیلد signatureValue مربوط به مقدار SignerInfo است که ویژگی با آن مرتبط شده است.

ویژگی امضای دوم می‌تواند چندین مقدار ویژگی داشته باشد. قواعد به‌عنوان SET OF AttributeValue تعریف شده‌اند و باید یک نمونه AttributeValue یا بیشتر از آن موجود باشد.

قواعد UnsignedAttributes به‌عنوان یک SET OF Attributes تعریف شده است. UnsignedAttributes در signerInfo ممکن است شامل چند نمونه از ویژگی امضای دوم باشد. امضای دوم، از آنجا که دارای نوع SignerInfo است می‌تواند خودش شامل ویژگی امضای دوم باشد. بنابراین ساخت یک سری طولانی اختیاری از امضاهای دوم ممکن است.

کتاب‌نامه

[۱] استاندارد ملی ایران به شماره ۱۰۸۲۵-۱: سال ۱۳۹۱، فناوری اطلاعات - فنون امنیتی - احراز هویت هستار قسمت ۱- کلیات

[۲] استاندارد ملی ایران به شماره ۱۰۸۲۲-۱: سال ۱۳۹۲، فناوری اطلاعات - فنون امنیتی - مدیریت کلید - قسمت ۱- چارچوب

[۳] استاندارد ملی ایران به شماره ۱۰۸۲۲-۳: سال ۱۳۸۷، فناوری اطلاعات - فنون امنیتی- مدیریت کلید - قسمت ۳- ساز و کارهای مبتنی بر فنون نامتقارن

[4] ISO/IEC TR 14516:2002, Information technology — Security techniques — Guidelines for the use and management of Trusted Third Party services

[5] ISO/IEC 8824-1:2002 | X.680: ITU-T Recommendation X.680 (2002), Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation

[6] ISO/IEC 8824-2:2002 | X.681: ITU-T Recommendation X.681 (2002), Information technology — Abstract Syntax Notation One (ASN.1): Information object specification

[7] ISO/IEC 9796 (all parts), Information technology — Security techniques — Digital signature schemes giving message recovery

[8] ISO/IEC 14888-3:2006, Information technology — Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms

[9] ISO/IEC 15946-1:2002, Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General

[10] ISO 19108:2002, Geographic information — Temporal schema

[11] IETF RFC 3280, Internet X.509 Public Key Infrastructure *Certificate and Certificate Revocation List (CRL) Profile*, April 2002

[12] IETF RFC 3852, *Cryptographic Message Syntax (CMS)*, July 2004

[13] ITU-R TF.460-6 (02/02), *Standard-frequency and time-signal emissions*