INSO
17114
1st. Edition
2014

جمهوری اسلامی ایران
Islamic Republic of Iran

سازمان ملی استاندارد ایران
Iranian National Standards Organization

استاندارد ملی ایران
۱۷۱۱۴
چاپ اول
۱۳۹۳

الزامات پروتکل درخواست صدور گواهی در زیرساخت کلید عمومی ایران

Requirements of Certification Request Protocol in Iranian Public Key Infrastructure

ICS:35.40

به نام خدا

آشنایی با سازمان ملی استاندارد ایران

مؤسسهٔ استاندارد و تحقیقات صنعتی ایران به موجب بند یک مادهٔ۳ قانون اصلاح قوانین و مقررات مؤسسهٔ استاندارد و تحقیقات صنعتی ایران، مصوب بهمن ماه ۱۳۷۱ تنها مرجع رسمی کشور است که وظیفه تعیین، تدوین و نشر استانداردهای ملی (رسمی) ایران را به عهده دارد.

نام موسسه استاندارد و تحقیقات صنعتی ایران به موجب یکصد و پنجاه و دومین جلسه شورای عالی اداری مورخ ۹۰/۶/۲۹ به سازمان ملی استاندارد ایران تغییر و طی نامه شماره ۲۰۶/۳۵۸۳۸ مورخ ۹۰/۷/۲۴ جهت اجرا ابلاغ شده است .

تدوین استاندارد در حوزه های مختلف در کمیسیون های فنی مرکب از کارشناسان سازمان ، صاحب نظران مراکز و مؤسسات علمی، پژوهشی، تولیدی و اقتصادی آگاه و مرتبط انجام می شود وکوششی 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 وارد کنندگان، مراکز علمی و تخصصی، نهادها، سازمان های دولتی و غیر دولتی حاصل می شود .پیش نویس استانداردهای ملی ایران برای نظرخواهی به مراجع ذی نفع و اعضای کمیسیون های فنی 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 می شود.

پیش نویس استانداردهایی که مؤسسات و سازمان های علاقه مند و ذی صلاح نیز با رعایت ضوابط تعیین شده تهیه می کنند درکمیتهٔ ملی طرح و بررسی و درصورت تصویب، به عنوان استاندارد ملی ایران چاپ و منتشر می شود .بدین ترتیب، استانداردهایی ملی تلقی می شوند که بر اساس مفاد نوشته شده در استاندارد ملی ایران شمارهٔ ۵ تدوین و در کمیتهٔ ملی استاندارد مربوط که سازمان ملی استاندارد ایران تشکیل می‌دهد به تصویب رسیده باشد.

سازمان ملی استاندارد ایران از اعضای اصلی سازمان بین المللی استاندارد (ISO)[1] ،کمیسیون بین المللی الکتروتکنیک (IEC)[2] و سازمان بین المللی اندازه شناسی قانونی (OIML)[3] است و به عنوان تنها رابط[4] کمیسیون کدکس غذایی (CAC)[5] در کشور فعالیت می کند. در تدوین استانداردهای ملی ایران ضمن توجه به شرایط کلی و نیازمندی های خاص کشور ، از آخرین پیشرفت های علمی، فنی و صنعتی جهان و استانداردهای بین‌المللی بهره‌گیری می‌شود .

سازمان ملی استاندارد ایران می تواند با رعایت موازین پیش بینی شده در قانون، برای حمایت از مصرف کنندگان، حفظ سلامت و ایمنی فردی و عمومی، حصول اطمینان از کیفیت محصولات و ملاحظات زیست محیطی و اقتصادی، اجرای بعضی از استانداردهای ملی ایران را برای محصولات تولیدی داخل کشور و/یا اقلام وارداتی، با تصویب شورای عالی استاندارد، اجباری نماید. سازمان می تواند به منظور حفظ بازارهای بین المللی برای محصولات کشور، اجرای استاندارد کالاهای صادراتی و درجه‌بندی آن را اجباری نماید. همچنین برای اطمینان بخشیدن به استفاده کنندگان از خدمات سازمان ها و مؤسسات فعال در زمینهٔ مشاوره، آموزش، بازرسی، ممیزی و صدورگواهی سیستم های مدیریت کیفیت و مدیریت زیست‌محیطی، آزمایشگاه ها و مراکز کالیبراسیون (واسنجی) وسایل سنجش، سازمان ملی استاندارد ایران این گونه سازمان ها و مؤسسات را بر اساس ضوابط نظام تأیید صلاحیت ایران ارزیابی می کند و در صورت احراز شرایط لازم، گواهینامهٔ تأیید صلاحیت به آن ها اعطا و بر عملکرد آن ها نظارت می کند. ترویج دستگاه بین المللی یکاها، کالیبراسیون (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 of Legal Metrology (Organisation Internationale de Metrologie Legale)

4 - Contact point

5 - Codex Alimentarius Commission

کمیسیون فنی تدوین استاندارد

« الزامات پروتکل درخواست صدور گواهی در زیرساخت کلید عمومی ایران »

رئیس :	**سمت و/ یا نمایندگی**
فیاضی، اسماعیل (فوق لیسانس نرم‌افزار و حقوق)	جانشین مدیرعامل شرکت ره‌آورد سامانه‌های امن
دبیر:	
فلاح چای، سیدمهدی (فوق لیسانس مخابرات رمز)	کارشناس مسؤول مرکز دولتی صدور گواهی الکترونیکی ریشه
اعضاء : (اسامی به ترتیب حروف الفبا)	
امین مقدم، عماد (فوق لیسانس مخابرات رمز)	کارشناس مرکز دولتی صدور گواهی الکترونیکی ریشه
امینیان، یوسف (لیسانس مهندسی کامپیوتر- نرم‌افزار)	کارشناس نرم‌افزار شرکت خدمات انفورماتیک راهبر
ایزدپناه، سحر (فوق لیسانس IT)	کارشناس مسؤول سازمان فناوری اطلاعات ایران
بهربگی، مهدی (فوق لیسانس مهندسی کامپیوتر)	مشاور PKI شرکت ره آورد سامانه های امن
پوربابایی، هادی (لیسانس مهندسی کامپیوتر - نرم‌افزار)	مدیر طرح و برنامه شرکت خدمات انفورماتیک راهبر
تیمورنژاد، علی (فوق لیسانس فناوری اطلاعات)	کارشناس PKI شرکت پیام پرداز
جامی، سارا (لیسانس علوم کامپیوتر)	کارشناس مرکز دولتی صدور گواهی الکترونیکی ریشه
جوادی نیا، رضا (لیسانس مهندسی کامپیوتر)	کارشناس PKI مرکز میانی عام
حاجی کریمیان، محسن (فوق لیسانس مهندسی کامپیوتر)	مدیر پروژه شرکت داده پردازی ایران

حسینی، ریحانه (لیسانس مهندسی کامپیوتر- نرم‌افزار)	کارشناس مرکز دولتی صدور گواهی الکترونیکی ریشه، دانشجوی فوق لیسانس مهندسی IT
خواجوی، هادی (لیسانس مهندسی کامپیوتر نرم‌افزار)	مدیرفروش شرکت پیام پرداز
راستی، رامبد (لیسانس مهندسی برق الکترونیک)	نماینده سازمان نظام صنفی کمیسیون افتا و مدیرگروه امنیت شرکت گام الکترونیک
شادمان، مهدی (لیسانس مهندسی کامپیوتر- نرم‌افزار)	رییس گروه شبکه و سخت افزار سازمان ثبت اسناد و املاک کشور
شاهی، فرید (لیسانس مهندسی کامپیوتر- نرم‌افزار)	سرپرست آزمایشگاه PKI ی مرکز تحقیقات صنایع انفورماتیک ایران
شیرازی، مریم (لیسانس مهندسی فناوری اطلاعات)	کارشناس پژوهشگاه استاندارد
طاهری، عباس (فوق لیسانس مهندسی IT- گرایش امنیت)	مدیر مرکز میانی پارس ساین شرکت امن افزار گستر شریف
قرنفلی، میثم (لیسانس مهندسی کامپیوتر)	کارشناس PKI مرکز میانی عام
عابدی، اسماعیل (لیسانس مهندسی کامپیوتر)	کارشناس مرکز دولتی صدور گواهی الکترونیکی ریشه، دانشجوی فوق لیسانس مدیریت تکنولوژی
کریمی، داود (فوق لیسانس IT)	مدیر امنیت سازمان امور مالیاتی کشور،
گوکی، رضا (لیسانس مهندسی کامپیوتر - نرم‌افزار)	کارشناس امنیت اطلاعات شرکت ره آورد سامانه های امن
هایراپطیان، کارین (فوق لیسانس مهندسی معماری کامپیوتر)	مدیر پروژه شرکت داده پردازی ایران

فهرست مندرجات

پیش‌گفتار

استاندارد « الزامات پروتکل درخواست صدور گواهی در زیرساخت کلید عمومی ایران » که پیش نویس آن در کمیسیون های مربوط توسط مرکز دولتی صدور گواهی الکترونیکی ریشه تهیه و تدوین شده است و در سیصد و چهل و سومین اجلاس کمیته ملی استاندارد رایانه و فرآوری‌داده‌ها مورخ ۱۳۹۳/۰۲/۳۰مورد تصویب قرار گرفته است ، اینک به استناد بند یک ماده ۳ قانون اصلاح قوانین و مقررات موسسه استاندارد و تحقیقات صنعتی ایران ، مصوب بهمن ماه ۱۳۷۱، به عنوان استاندارد ملی ایران منتشر می‌شود .

برای حفظ همگامی و هماهنگی با تحولات و پیشرفت های 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 ارائه شود ، هنگام تجدید نظر در کمیسیون فنی مربوط مورد توجه قرار خواهد گرفت . بنابراین ، باید همواره از آخرین تجدید نظر استانداردهای ملی استفاده کرد .

منابع و ماخذی که برای تهیه این استاندارد مورد استفاده قرار گرفته به شرح زیر است :

RFC 5272:2008, CMC: Certificate Management Over CMS

RFC 4211:2005, CRMF: Internet X.509 Certification Request Message Format

RFC 2986: 2000, PKCS #10: Certification Request Syntax Specification v1.7

RFC 2104: 1997, HMAC: Keyed-Hashing for Message Authentication

RFC 2119: 1997, Key words for use in RFCs to Indicate Requirement Levels

RFC 3279: 2002, Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

مقدمه

در این استاندارد، ساختار درخواست صدور گواهی الکترونیکی X.509 به منظور ارائه این درخواست به صادرکننده گواهی (CA)[1]، تعریف می‌گردد، ضمن این‌که نحوه ارائه درخواست صدور گواهی به CA به همراه الزامات لازم جهت ارائه آن، در قالب پروتکل درخواست صدور گواهی بیان می‌گردد. به طور کلی یک درخواست صدور گواهی شامل کلید عمومی هستار[2] درخواست‌کننده گواهی و اطلاعات لازم جهت ثبت گواهی برای این هستار می‌باشد. درخواست‌های صدور گواهی می‌توانند توسط یک مرجع ثبت (RA)[3] به نیابت از متقاضی گواهی یا به طور مستقیم توسط یک هستار نهایی (EE)[4] ارائه شوند.

ساختار درخواست صدور گواهی تعریف شده در این استاندارد، یک قالب مستقل نیست، بلکه توسط پروتکل درخواست صدور گواهی به‌کار گرفته می‌شود. ساختارهای اطلاعاتی، الزامات کنترلی و نیز سایر ملزومات درخواست صدور گواهی در این استاندارد تعریف و تبیین می‌شوند.

1 - Certificate Authority
2 - Entity
3 - Registration Authority
4 - End Entity

الزامات پروتکل درخواست صدور گواهی در زیرساخت کلید عمومی ایران

۱ هدف و دامنه کاربرد

هدف از تدوین این استاندارد، تعیین روش‌های ایجاد درخواست برای صدور گواهی الکترونیکی X.509 است. این استاندارد برای کلیه نرم‌افزارهایی که فرآیند ایجاد و پردازش درخواست صدور گواهی را انجام می‌دهند (مانند سامانه‌های صادرکننده و مدیریت گواهی و نرم‌افزارهای مجهز به زیرساخت کلید عمومی (PKE)[1]) کاربردپذیر است.

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:

۲-۱ سند جامع پروفایل‌های زیرساخت کلید عمومی کشور

۲-۲ استاندارد ملی "ساختار نحوی پیام‌های رمزنگاشتی در زیرساخت کلید عمومی ایران" به شماره ۱۷۱۱۳

2-3 RFC 2986: 2000, PKCS #10: Certification Request Syntax Specification v1.7

2-4 RFC 3852: 2004, Cryptographic Message Syntax (CMS)

2-5 RFC 3447: 2000, PKCS #1: RSA Cryptography Specification v2.1

۳ اصطلاحات و تعاریف

در این استاندارد اصطلاحات و تعاریف زیر به کار می‌رود:

1- Public Key Infrastructure

۱-۳
AlgorithmIdentifier[1]

از انواع داده تعریف شده در استاندارد X.509 می‌باشد که در حالت کلی، مشخص‌کننده یک الگوریتم (با یک شناسانه شیء(OID)[2] مشخص می‌شود) و پارامترهای آن است.

۲-۳
نشانه‌گذاری نحو انتزاعی.یک
ASN.1[3]

روش استاندارد تعریف شده در NIC X.690 جهت توصیف ساختارهای داده می‌باشد.

۳-۳
صفت[4]

داده‌ای مرکب از یک صفت (با یک شناسانه مشخص می‌شود) و یک یا چند مقدار برای آن صفت می‌باشد که در استاندارد X.501 تعریف شده است.

۴-۳
قواعد کدبندی پایه
BER[5]

قواعد کدبندی پایه که در استاندارد X.690 تعریف شده است.

۵-۳
گواهی

از انواع داده تعریف شده در استاندارد X.509 می‌باشد که در آن از یک امضای رقمی (دیجیتال) برای برقراری تناظر یک به یک بین شناسه منحصر به فرد[6] یک هستار و کلید عمومی آن استفاده شده است. شناسه منحصر به فرد صادرکننده گواهی، شماره ردیف خاص صادرکننده گواهی، شناسانه الگوریتم استفاده شده توسط صادر کننده گواهی به جهت امضاء، دوره اعتبار گواهی و الحاقیه‌های[7] مرتبط مختلف، بخش‌های دیگر این نوع داده هستند.

1 - AlgorithmIdentifier
2 - Object identifier
3 - Abstract Syntax Notation One
4 - Attribute
5 - Basic Encoding Rules
6 - Distinguished Name
7 - Extension

۶-۳

فهرست ابطال گواهی

CRL[1]

گواهی‌های الکترونیکی که باطل شده‌اند و دیگر نباید آن‌ها را معتبر به حساب آورد را، فهرست می‌کند. این ساختار داده که توسط صادرکننده CRL امضاء می‌گردد، دربرگیرنده اطلاعاتی مشتمل بر نام صادرکننده CRL، زمان انتشار CRL، زمانی که بر طبق برنامه[2] قرار است CRL بعدی منتشر گردد و فهرستی از شماره ردیف گواهی‌های باطل‌شده و زمان ابطال هر یک است. این ساختار داده، همان ساختار داده تعریف شده در RFC 1422 است.

۷-۳

قواعد کدبندی منحصر به فرد

DER[3]

قواعدی است که در X.690 تعریف شده و در آن، بر اساس X.509 محدودیت‌هایی بر روش کدبندی پایه اعمال شده است. این روش کدبندی که زیرمجموعه‌ای از روش کدبندی BER است، روشی یکتا برای تفسیر مقادیر ASN.1 فراهم می‌آورد.

۸-۳

X.509

استاندارد X.509 یکی از استانداردهای بخش استانداردسازی اتحادیه بین‌المللی مخابرات (ITU-T)[4] برای زیرساخت کلید عمومی است که ساختار گواهی الکترونیکی و فهرست ابطال گواهی را تعیین می‌کند.

۹-۳

صادرکننده گواهی(CA)

هستاری که مجاز به صدور و مدیریت گواهی‌های الکترونیکی است.

۱۰-۳

مرجع ثبت(RA)

هستاری که به عنوان واسط بین هستار نهایی و صادرکننده گواهی عمل می‌کند و وظیفه آن شناسایی کاربران و ثبت و ارسال درخواست‌های صدورگواهی برای CA می باشد.

۱۱-۳

هستار نهایی(EE)

هستاری که یک زوج کلید در اختیار دارد و گواهی برای او صادر می‌شود.

---

1 - Certificate Revocation List
2 - Scheduled
3 - Distinguished Encoding Rules
4 - International Telecommunication Union-Telecommunication

۱۲-۳
درخواست امضای گواهی
CSR[1]
پیامی که توسط یک متقاضی صدور گواهی یا نماینده او جهت صدور گواهی الکترونیکی X.509 ایجاد و به صادرکننده گواهی(CA) ارائه می‌گردد.

۱۳-۳
PKCS# 10
یک استاندارد رمزنگاری کلید عمومی است که دربردارنده یک قالب درخواست صدور گواهی (CSR) است.

۱۴-۳
قالب پیام درخواست صدور گواهی
CRMF[2]
یک قالب پیام درخواست صدور گواهی (CSR) که در RFC4211 توصیف شده است.

۱۵-۳
زیرساخت کلید عمومی
PKI[3]
مجموعه‌ای از خدمات، محصولات، خط‌مشی‌ها، فرایندها و سامانه‌های نرم‌افزاری و سخت‌افزاری گفته می‌شود که جهت مدیریت و به کارگیری گواهی‌های الکترونیکی X.509 و به منظور ارائه خدمات امنیتی مختلف مبتنی بر رمزنگاری کلید عمومی[4] مورد استفاده قرار می‌گیرد.

۱۶-۳
شناسانه شیء(OID)
یکی از انواع اولیه در ساختار ASN.1 است.

۱۷-۳
احراز هویت[5]
فرایند شناسایی هویتی که برای یک شخص یا برای یک هستار سامانه‌ای ادعا شده است.

۱۸-۳
شناسایی[6]
شناسایی و تشخیص یک هستار از هستارهای دیگر، از طریق بررسی مدارک شناسایی اشخاص و اطلاعات شناسایی دیگر از قبیل اسم‌رمزها، اطلاعات زیست‌سنجشی و غیره می‌باشد.

---

1 - Certificate Signing Request
2 - Certificate Request message Format
3 - Public Key Infrastructure
4 - Public Key Cryptography
5 - Authentication
6 - Identification

٤ مرور کلی

این استاندارد دربردارنده الزامات فنی پروتکل درخواست صدور گواهی الکترونیکی X.509 در زیرساخت کلید عمومی ایران می‌باشد. در این راستا دو نوع ساختار درخواست صدور گواهی الکترونیکی (CSR) شامل درخواست‌های مبتنی بر PKCS#10 و درخواست‌های مبتنی بر CRMF معرفی شده است.

استاندارد PKCS#10 یک استاندارد قالب درخواست صدور گواهی الکترونیکی است که درخواست‌های مبتنی بر این استاندارد به‌عنوان رایج‌ترین قالب درخواست صدور گواهی امضا در نرم‌افزارهای مختلف مورد استفاده قرار می‌گیرد و شامل سه بخش اصلی است: ۱- اطلاعات درخواست صدور گواهی ۲- شناسانه الگوریتم امضا و ۳-یک امضای الکترونیکی بر روی داده درخواست گواهی. در PKCS#10 درخواست حتما باید امضا شود تا امکان اثبات مالکیت کلید خصوصی توسط گیرنده درخواست (RA/CA) از طریق اعتبارسنجی امضا وجود داشته باشد. لازم به ذکر است که قبل از تولید یک CSR باید عملیات تولید زوج کلید صورت پذیرفته و کلید عمومی متناظر با این زوج کلید به همراه نام و مشخصات متقاضی گواهی الکترونیکی در قالب CSR به CA/RA جهت صدور گواهی، ارائه گردد.

درخواست مبتنی بر CRMF نوع دیگری از درخواست صدور گواهی الکترونیکی می‌باشد که جهت درخواست صدور گواهی‌های الکترونیکی با کاربردهای مختلف (شامل امضا و رمزبندی[1]) قابل استفاده می‌باشد. اثبات مالکیت کلیدهای امضا در درخواست مبتنی برCRMF مشابه درخواست‌های PKCS#10 می‌باشد؛ CRMF همچنین سه روش برای اثبات مالکیت کلید خصوصی با کاربرد رمزبندی ارائه می‌دهد که دو روش آن در زیرساخت کلید عمومی کشور پذیرفته شده است و در این استاندارد توصیف می‌گردد.

در این استاندارد علاوه بر معرفی قالب درخواست‌های مزبور و روش‌های اثبات مالکیت کلید خصوصی، به توصیف پروتکل درخواست صدور گواهی مبتنی بر CMC(CMS)[2] پرداخته شده که در آن درخواست‌های ساده و کامل PKI و الزامات کنترلی متناظر ارائه شده است. لازم به ذکر است که درخواست‌های گواهی در CMC می‌توانند حاوی یک درخواست صدور گواهی ساده در قالب CertificationRequest در PKCS#10، یا CertReqMsg در CRMF و یا هر دو باشند. در واقع در این بخش چگونگی هماهنگ‌سازی درخواست‌های صدور گواهی با ساختارهای CMS و چگونگی تولید درخواست‌های گروهی صدور گواهی به منظور به‌کارگیری در مراجع ثبت(RA) و صادرکنندگان گواهی(CA) توصیف می‌گردد.

٥ مروری بر قالب‌های درخواست صدور گواهی

در این بند از استاندارد دو نوع ساختار درخواست صدور گواهی الکترونیکی (CSR) شامل درخواست‌های مبتنی بر PKCS#10 و درخواست‌های مبتنی بر CRMF معرفی می‌شود.

---

1 - Encryption

2 - Certificate Management over CMS

۵-۱ قالب درخواست صدور گواهی مبتنی بر PKCS#10

در این بند ساختار درخواست‌های ساده PKI [۱] طبق PKCS#10 توصیف می‌شود. یک درخواست PKI مبتنی بر PKCS#10 از سه بخش اطلاعات درخواست گواهی، شناسانه الگوریتم امضا و امضای رقمی بر روی اطلاعات درخواست صدور گواهی تشکیل شده است. اطلاعات درخواست صدور گواهی، مشتمل بر شناسه منحصر به فرد هستار، کلید عمومی هستار و مجموعه صفات اطلاعات دیگری در مورد هستار فراهم می‌آورند تشکیل شده است.

فرآیندی که یک درخواست صدور گواهی مبتنی بر PKCS#10 را تشکیل می‌دهد دارای مراحل زیر است:

۱- CertificationRequestInfo در برگیرنده شناسه منحصر به فرد مالک گواهی، کلید عمومی مالک گواهی و مجموعه صفات اختیاری که توسط یک هستار درخواست‌کننده گواهی ایجاد می‌گردد.

۲- CertificationRequestInfo با کلید خصوصی هستار مالک گواهی امضا می‌شود.

۳- CertificationRequestInfo، شناسانه الگوریتم امضا و امضای هستار در CertificationRequest تجمیع می‌گردد.

صادرکننده گواهی درخواست را با سنجیدن اصالت هستار درخواست‌کننده و ارزیابی امضای هستار تکمیل می‌کند. درصورتی که درخواست معتبر باشد، یک گواهی X.509 متشکل از نام و مشخصات مالک گواهی، کلید عمومی، نام صادرکننده، شماره ردیف انتخابی CA، دوره اعتبار و الگوریتم امضا ایجاد می‌کند.

اگر درخواست صدور گواهی دارای صفاتی از PKCS#9 باشد، CA ممکن است از آنها به همراه سایر اطلاعاتی که در خود دارد برای ایجاد الحاقیات گواهی X.509 استفاده کند.

یادآوری۱- یک هستار به‌طور معمول درخواست صدور گواهی را بعد از تولید یک زوج کلید عمومی/خصوصی ارسال می‌کند.

یادآوری۲- امضا روی درخواست گواهی، از درخواست صدور گواهی توسط یک هستار با کلید عمومی هستار دیگر جلوگیری می‌کند. چنین حمله‌ای به هستار این قابلیت را می‌دهد که وانمود کند مبدأ هر پیام امضا شده شخص دیگری است. تنها هنگامی‌که هستار، پیام امضا شده را نشناسد و بخش امضا شده پیام هویت امضا کننده را مشخص نکند، حمله معتبر خواهد بود. البته هستار هنوز هم قادر به رمزگشایی پیام‌هایی که برای دیگر اشخاص هستند نخواهد بود.

در زیربندهای بعدی ابتدا به توصیف و تشریح ساختار اطلاعات درخواست صدور گواهی CertificationRequestInfo پرداخته می‌شود و سپس ساختار سطح بالاتر CertificationRequest توصیف می‌گردد.

۵-۱-۱ ساختار CertificationRequestInfo

اطلاعات درخواست گواهی، ساختار ASN.1 زیر را دارا خواهد بود:

```
CertificationRequestInfo ::= SEQUENCE {
version                  INTEGER { v1(0) } (v1,...),
subject                  Name,
```

۱ درخواست ساده PKI در زیربند ۶-۲ توصیف شده است.

```
subjectPKInfo                SubjectPublicKeyInfo{{ PKInfoAlgorithms }},
attributes                   [0] Attributes{{ CRIAttributes }}
}
SubjectPublicKeyInfo { ALGORITHM : IOSet} ::= SEQUENCE {
algorithm          AlgorithmIdentifier {{IOSet}},
subjectPublicKey             BIT STRING
}
PKInfoAlgorithms ALGORITHM ::= {
... -- الگوریتم‌هایی که به صورت محلی تعریف شده‌اند به اینجا اضافه می‌شوند --  }
Attributes { ATTRIBUTE:IOSet } ::= SET OF Attribute{{ IOSet }}

CRIAttributes  ATTRIBUTE  ::= {
... -- الگوریتم‌هایی که به صورت محلی تعریف شده‌اند به اینجا اضافه می‌شوند --  }
Attribute { ATTRIBUTE:IOSet } ::= SEQUENCE {
type      ATTRIBUTE.&id({IOSet}),
values    SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}{@type})
}
```

مولفه‌های این ساختار دارای تعاریف زیر هستند:

- version – شماره نسخه را نشان می‌دهد و مقدارش معادل صفر است.
- subject – نام منحصر به فرد مالک گواهی (هستاری که کلید عمومی‌اش باید گواهی گردد).
- subjectPublicKeyInfo – دربرگیرنده اطلاعاتی در مورد کلید عمومی است که در گواهی قید می‌گردد. این اطلاعات، الگوریتم کلید عمومی هستار (و پارامترهای آن) را مشخص می‌کند؛ مثالی از الگوریتم کلید عمومی شناسانه rsaEncryption (در PKCS#1) است. این اطلاعات همچنین دربرگیرنده ساختار بیتی کلید عمومی هستار است. برای الگوریتم کلید عمومی که به آن اشاره شد، رشته بیتی حاوی شکل کدبندی DER ساختار RSAPublicKey (در PKCS#1) است. مقادیر ساختار {}SubjectPublicKeyInfo که برای subjectPKInfo مجاز هستند، به مقادیری که توسط PKIAlgorithms[1] مشخص شده‌اند، محدود می‌گردند.
- attributes – مجموعه‌ای از صفات است که اطلاعات بیشتری در مورد مالک گواهی فراهم می‌آورد (برخی از این صفات در PKCS#9 یافت می شود). نمونه‌هایی از این صفات عبارتند از چالش-اسم‌رمز؛ که یک اسم‌رمز برای درخواست ابطال گواهی توسط هستار، فراهم می‌آورد، یا اطلاعاتی که در الحاقیه‌های گواهی X.509 ظاهر می‌شوند (مانند صفت extensionRequest در PKCS#9). مقدارهای {}Attributes که برای این فیلد مجاز هستند به مقدارهایی که توسط CRLAttributes تعیین می‌گردند، محدود می‌شود.

۲-۱-۵ ساختار CertificationRequest

ساختار ASN.1 یک درخواست صدور گواهی در PKCS#10 به صورت زیر است:

---

۱ - این مقادیر با االحاقیه‌هاي ساختار PKInfoAlgorithms نشان داده شده است.

```
CertificationRequest ::= SEQUENCE {
        certificationRequestInfo  CertificationRequestInfo,
        signatureAlgorithm        AlgorithmIdentifier{{ SignatureAlgorithms }},
        signature                 BIT STRING
        }

AlgorithmIdentifier {ALGORITHM:IOSet } ::= SEQUENCE {
        algorithm       ALGORITHM.&id({IOSet}),
        parameters      ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL
        }

SignatureAlgorithms ALGORITHM ::= {... -- الگوریتم‌های تعریف شده محلی در این بخش اضافه می‌شود --
}
```

مؤلفه‌های این ساختار به صورت زیر هستند:

- CertificateRequestInfo – اطلاعات درخواست صدور گواهی است که در زیربند ۵-۱-۱ توضیح داده شد و مقدار آن امضاء می‌شود.
- SignatureAlgorithm – الگوریتم امضا (و پارامترهای آن) را مشخص می‌کند به‌طوری‌که تحت آن، اطلاعات درخواست صدور گواهی امضا می‌گردد. به طور مثال مشخصات می‌تواند شامل یک شی ALGORITHM برای sha1WithRSAEncryption (در PKCS#1) در مجموعه شناسانه الگوریتم‌ها یعنی SignatureAlgorithms باشد.
- signature – نتیجه امضای اطلاعات درخواست صدور گواهی با کلید خصوصی مالک گواهی است.

فرآیند امضا از دو مرحله زیر تشکیل می‌شود:

۱- مقدار مؤلفه certificationRequestInfo که به شکل کدبندی DER است، یک رشته بایتی را نتیجه می‌دهد.

۲- نتیجه مرحله ۱ با کلید خصوصی مالک گواهی مشخص شده در تقاضای گواهی، تحت الگوریتم امضا تعریف شده، امضا می‌شود و رشته بایتی امضا را نتیجه می‌دهد.

ساختار معادل CertifcationRequest می‌تواند به صورت زیر نوشته شود:

```
CertificationRequest ::= SIGNED { EncodedCertificationRequestInfo }
    (CONSTRAINED BY { -- Verify or sign encoded CertificationRequestInfo -- })

EncodedCertificationRequestInfo ::=  TYPE-
IDENTIFIER.&Type(CertificationRequestInfo)

SIGNED { ToBeSigned } ::= SEQUENCE {
    toBeSigned  ToBeSigned,
    algorithm       AlgorithmIdentifier { {SignatureAlgorithms} },
    signature   BIT STRING
```

۵-۲ درخواست صدور گواهی مبتنی بر CRMF

قالب پیام درخواست صدور گواهی CRMF جهت ارسال یک درخواست به صادرکننده گواهی از طریق یک مرجع ثبت یا به طور مستقیم توسط هستار نهایی کاربرد دارد که منجر به صدور گواهی می‌شود.

مراحل تشکیل یک درخواست صدور گواهی مبتنی بر CRMF به صورت زیر است:

الف- یک ساختار درخواست صدور گواهی یا CertRequest ایجاد می‌شود. این ساختار شامل کلید عمومی، کل اطلاعات مربوط به نام و مشخصات مالک گواهی یا بخشی از آن، سایر فیلدهای گواهی درخواست شده و اطلاعات کنترلی اضافه مربوط به فرآیند ثبت‌نام است. این اطلاعات می‌تواند توسط مالک گواهی مشخص شوند و RA نیز این قابلیت را دارد که آن‌ها را تغییر دهد یا آن‌ها را بر پایه مجموعه اطلاعاتی که از مالک گواهی دارد یا مستنداتی که توسط او ارائه شده است، مشخص کند.

ب- در صورت لزوم، مقدار اثبات مالکیت(POP)[1] کلید خصوصی متناظر با کلید عمومی که برای آن گواهی درخواست شده است، محاسبه می‌شود.

پ- اطلاعات افزوده ثبت[2] می‌تواند با مقدار اثبات مالکیت و ساختار CertRequest ترکیب شده و ساختار پیام درخواست صدور گواهی CertReqMessage را تشکیل دهد. این اطلاعات می‌تواند توسط مالک گواهی و یا RA اضافه گردد.

ت- پیام CertReqMessage با یک روش امن به CA ارسال می‌شود.

۵-۲-۱ ساختار کلی درخواست صدور گواهی CRMF

پیام درخواست صدور گواهی (CertReqMessage) از درخواست گواهی، فیلد اختیاری اثبات مالکیت و فیلد اختیاری اطلاعات ثبت‌نام تشکیل شده است.

```
CertReqMessages ::= SEQUENCE SIZE (1..MAX) OF CertReqMsg
  CertReqMsg ::= SEQUENCE {
    certReq   CertRequest,
    popo      ProofOfPossession  OPTIONAL,
    محتوای فیلد به نوع کلید بستگی دارد --
    regInfo   SEQUENCE SIZE(1..MAX) of AttributeTypeAndValue OPTIONAL
```

فیلدهای مختلف ساختار CertReqMsg به شرح زیر می‌باشد:

- certReq الگوی گواهی درخواست شده را در خود قرار می‌دهد. این الگو توسط (یا به نمایندگی از) مالک گواهی پر می‌شود. نیازی نیست که همه فیلدهای درون آن تکمیل گردند. جزییات این فیلد در زیربند ۵-۲-۳ آورده شده است.

1 - Proof of Possession
2 - Additional Registration Information

- popo بیانگر مالکیت کلید خصوصی متناظر با گواهی است که توسط هستار مشخص شده در فیلد مالک گواهی، درخواست شده است. این فیلد در ساختار و مفهوم بر اساس الگوریتم کلید عمومی و کاربردی که الگوریتم مورد استفاده قرار می‌گیرد(رمزبندی یا امضا)[1]می‌تواند متفاوت باشد. جزییات این فیلد در زیربند ۵-۲-۲ آورده شده است.
- regInfo درجایی که اطلاعاتی به منظور تکمیل درخواست لازم است، شامل اطلاعات مکمل مرتبط با زمینه درخواست صدور گواهی می‌باشد. این اطلاعات می‌تواند مشتمل بر اطلاعات تماس متقاضی گواهی، اطلاعات صدور صورتحساب، یا دیگر اطلاعات کمکی مناسب برای تکمیل درخواست باشد.

اطلاعاتی که به محتوای گواهی به طور مستقیم ارتباط دارند باید در محتوای CertReq قرار گیرند. هرچند که افزودن محتوای CertReq اضافی توسط RAها می‌تواند مقدار فیلد popo را (بسته به جزییات روش POP مورد استفاده نامعتبر کند). بنابراین، اطلاعات تهیه شده با هدف درج در محتوای گواهی ممکن است در regInfo قرار گیرد.

تعریف جزییاتی مانند اینکه چه چیزی در فیلد regInfo می‌تواند تعریف شود، در پروتکل درخواست صدور گواهی مورد استفاده قید می‌گردد.

۵-۲-۲ فرآیند اثبات مالکیت کلید خصوصی مبتنی بر CRMF

با هدف جلوگیری از برخی حملات و به منظور مجاز دانستن یک RA/CA برای بررسی تناظر بین نام و مشخصات ارائه شده برای مالک گواهی و یک زوج کلید، ساختار مدیریت PKI تعریف شده در این استاندارد، این امکان را فراهم می‌آورد که اثبات مالکیت (یعنی قابل استفاده بودن) کلید خصوصی متناظر با کلید عمومی که برای آن گواهی درخواست شده است، امکان‌پذیر باشد. در این بند روش‌های مختلف اثبات مالکیت کلید خصوصی مبتنی بر قالب درخواست CRMF ارائه شده است.

اثبات مالکیت کلید خصوصی بسته به نوع کلیدی که برای آن یک گواهی درخواست شده است، به روش‌های مختلفی انجام می‌شود. اگر یک کلید برای اهداف مختلفی استفاده شود (به طور مثال، کلید RSA برای رمزگشایی و امضا استفاده گردد)، در این صورت می‌توان از هر روش موجود در این استاندارد برای انجام POP استفاده نمود.

در این استاندارد اجرای فرآیند POP توسط RA، CA و یا هر دو امکان‌پذیر است. درصورتی‌که CA فرآیند POP را در حین صدور گواهی انجام دهد، RA باید فیلدهای CertRequest و ProofOfPossession مربوط به هستار نهایی را بدون تغییر برای CA ارسال کند. در این حالت، RA می‌تواند POP را بررسی و درخواست‌های گواهی دارای اشکال را به CA ارسال نکند. اگر بنا به ملاحظاتی که در خط مشی صادرکننده گواهی الکترونیکی بیان شده، CA الزامی نداشته باشد که فرآیند POP را بررسی کند، آن‌گاه توصیه می‌گردد

---

۱ - این کاربرد در بخش KeyUsage ازگواهی صادر شده، مشخص می‌شود

RA درخواست هستار نهایی و اثبات مالکیت را بدون تغییر، مانند فوق به CA ارسال کند و چنانچه انجام این کار امکان‌پذیر نباشدRA از فیلد raVerified جهت درستی‌سنجی انجام فرآیند اثبات مالکیت کلید خصوصی توسط خود و اعلام آن به CA، استفاده می‌کند.

```
ProofOfPossession ::= CHOICE {
    raVerified          [0]     NULL,
    signature           [1]     POPOSigningKey,
    keyEncipherment     [2]     POPOPrivKey,
    keyAgreement        [3]     POPOPrivKey }
```

فیلدهای ProofOfPossession معانی زیر را دارا هستند:

- فیلد raVerified اشاره بر این دارد که RA برای درخواست گواهی، عملیات POP را اجرا می‌کند. این فیلد هنگامی توسط RA استفاده می‌شود که: ۱- CA برای اجرای POP الزامی نداشته نباشد. ۲- RA بخواهد محتوای فیلد certReq را تغییر دهد. در پروتکل درخواست صدور گواهی باید روشی برای RA به منظور امضا کردن ProofOfPossession ارائه گردد. درخواست‌کننده صدور گواهی در این فیلد نباید مقداری قرار دهد و RA یا CA نباید فیلد ProofOfPossession که توسط درخواست‌کننده صدور گواهی مقداردهی شده است را مورد پذیرش قرار دهند.
- signature برای اجرای عملیات POP با کلیدهای مورد استفاده در کاربرد امضا به کار می‌رود. جزییات این فیلد در زیربند ۵-۲-۲-۱ آمده است.
- keyEncipherment برای اجرای عملیات POP با کلیدهای پوشیده‌سازی مورد استفاده در کاربرد پوشیده‌سازی کلید (با استفاده از الگوریتم RSA) به کار می‌رود. جزییات این فیلد در زیربند ۵-۲-۲-۲ آمده است.
- keyAgreement برای اجرای عملیات POP با کلیدهای رمزبندی مورد استفاده در توافق کلید (یعنی توافق کلید با استفاده از الگوریتم DH) به کار می‌رود؛ این روش در زیرساخت کلید عمومی کشور کاربردپذیر نمی‌باشد.

۵-۲-۲-۱اثبات مالکیت کلید امضا

به منظور اثبات مالکیت یک کلید امضا، قسمتی از داده شامل هویتی که گواهی برای آن درخواست شده، امضا می‌شود.

هنگام اجرای عملیات POP برای کلید امضا، سه حالت وجود دارد:

۱- حالتی که هویت مالک گواهی هنوز توسط CA/RA احراز هویت نشده است ولی اسم‌رمز و رشته شناسایی ارائه شده از طرف CA/RA در اختیار مالک گواهی می‌باشد؛ در این حالت، ساختار POPOSigningKeyInput با انتخاب publicKeyMAC برای authInfo تکمیل می‌گردد و برای محاسبه

مقدار publicKeyMAC نیز از اسم‌رمز و رشته شناسایی مذکور استفاده می‌شود. نحوه محاسبه کد اصالت‌سنجی پیام(MAC)[1] مبتنی بر اسم‌رمز در زیربند ۵-۲-۲-۳ آمده است. کلید عمومی متناظر با گواهی مورد درخواست، هم در ساختار POPOSigningKeyInput و هم در ساختار Certificate Template قرار می‌گیرد. فیلد امضا بر اساس ساختار POPOSigningKeyInput که با کدبندی DER مقداردهی شده است، محاسبه می‌گردد.

۲- حالتی که هویت مالک گواهی توسط CA/RA احراز هویت گردیده ولی درخواست‌کننده هنوز آن‌را در درخواست صدور گواهی قرار نداده است؛ در این حالت با انتخاب فیلد Sender به عنوان مقدار authInfo ساختار POPOSigningKeyInput مقداردهی می‌شود. کلید عمومی برای گواهی درخواست شده در ساختارهای POPOSigningKeyInput و Certificate Template قرار می‌گیرد. فیلد امضا روی ساختار POPOSigningKeyInput که از طریق روش DER کدبندی شده است، محاسبه می‌شود.

۳- حالتی که مالک گواهی نام و کلید عمومی خود را در ساختار Certificate Template قرار می‌دهد؛ در این حالت فیلد poposkInput از ساختار POPOSigningKey حذف می‌شود. فیلد امضا روی ساختار POPOSigningKeyInput که از طریق روش DER کدبندی شده است، محاسبه می‌شود.

```
POPOSigningKey ::= SEQUENCE {
    poposkInput         [0] POPOSigningKeyInput OPTIONAL,
    algorithmIdentifier     AlgorithmIdentifier,
    signature               BIT STRING }
```

فیلدهای ساختار POPOSingingKey به شرح زیر هستند:

- poposkInput در صورت وجود، حاوی داده‌ای است که باید امضا گردد. در صورتی که ساختار Certificate Template، مقادیر کلید عمومی و نام مالک گواهی را شامل نشود، وجود این فیلد الزامی است.
- algorithmIdentifier الگوریتم امضا و پارامترهای آن که برای تولید مقدار POP کاربرد دارد را تعیین می‌کند.
- signature دربردارنده مقدار محاسبه شده POP است. در صورت وجود فیلد poposkInput ، امضا روی مقدار کد شده این فیلد به روش DER محاسبه می‌شود و در غیر این صورت، امضا با استفاده از مقدار فیلد certReq که به شکل کدبندی DER است، محاسبه می‌شود.

```
POPOSigningKeyInput ::= SEQUENCE {
   authInfo                    CHOICE {
    sender          [0]        GeneralName,
```

1 - Message Authentication Code

-- فقط زمانی استفاده می‌شود که درخواست‌کننده گواهی(فرستنده) شناسایی شده باشد و اطلاعات شناسایی او موجود باشد. (به طور مثال، یک DN از گواهی که در قبل صادر شده و اکنون معتبر است)

publicKeyMAC PKMACValue },

-- در صورتی که برای فرستنده، GeneralName احراز اصالت شده‌ای تا آن زمان وجود نداشته باشد استفاده می‌شود؛

publicKeyMAC شامل یک MAC مبتنی بر کلمه‌عبور روی مقدار publicKey که با DER کدبندی شده، است. —

publicKey SubjectPublicKeyInfo } CertTemplate از --

فیلدهای ساختار POPOSigningKeyInput به صورت زیر هستند:

- sender دربردارنده شناسانه هویت احراز هویت شده‌ای است که از قبل برای مالک گواهی احراز شده است.
- publicKeyMAC دربردارنده مقداری است که با استفاده از راز به اشتراک‌گذاشته شده[1] موجود بین CA/RA و درخواست‌کننده گواهی محاسبه می‌شود.[2]
- publicKey دربردارنده نسخه رونوشت کلید عمومی از الگوی گواهی است. این فیلد باید به طور دقیق همان مقداری را دارا باشد که در الگوی گواهی وجود دارد.

```
PKMACValue ::= SEQUENCE {
    algid   AlgorithmIdentifier,
    value               BIT STRING }
```

فیلدهای ساختار PKMACValue به صورت زیر هستند:

- algid الگوریتمی را که مقدار MAC را محاسبه می‌کند، مشخص می‌کند. همه پیاده‌سازی‌ها باید از id-PasswordBasedMAC پشتیبانی کنند. جزییات این الگوریتم در زیربند ۵-۲-۲-۳ بیان شده است.
- value دربرگیرنده مقدار محاسبه شده MAC است. مقدار MAC با کدبندی DER کلید عمومی مالک گواهی محاسبه می‌شود.

CA/RA استفاده از راز به اشتراک گذاشته شده را با استفاده از ۱-فیلد[3] نام کلی در درخواست صدور گواهی یا ۲-کنترل‌هایregToken یا authToken تعیین می‌کند. کنترل regToken حاوی اطلاعاتی است که

---

1 - Shared Secret

۲ -راز به اشتراک گذاشته شده در واقع همان رشته شناسايي مي‌باشد كه مي‌تواند به صورت يك اسم‌رمز در اولين مراجعه يك متقاضي صدور گواهي به RA يا CA در اختيار او قرار گيرد.

3 - Field

توسط CA برای وارسی هویت مالک گواهی قبل از صدور گواهی به کار می‌رود. کنترل authToken حاوی اطلاعاتی جهت وارسی غیر رمزنگاشتی هویتی است که در ارتباط با CA است که می‌تواند شامل ۴ رقم آخر کد ملی یا نام خانوادگی یا سایر اطلاعات مرتبط یا چکیده این اطلاعات باشد.

۵-۲-۲-۲ کلیدهای رمزبندی کلید

اجرای عملیات POP روی کلیدهای رمزبندی کلید، به واسطه یکی از روش‌هایی که در ادامه می‌آید، انجام می‌شود. ممکن است یک چالش رمزبندی‌شده ارسالی از طرف CA/RA رمزگشایی شود (حالت مستقیم) و یا اینکه گواهی صادر شده به صورت رمزبندی شده بازگردد و به عنوان پاسخ چالش استفاده شود (حالت غیرمستقیم).

```
POPOPrivKey ::= CHOICE {
    thisMessage                 [0] BIT STRING,   -- deprecated
    subsequentMessage       [1] SubsequentMessage,
    dhMAC                       [2] BIT STRING,   -- deprecated
    agreeMAC                    [3] PKMACValue,
    encryptedKey                [4] EnvelopedData }

  SubsequentMessage ::= INTEGER {
    encrCert (0),
    challengeResp (1) }
```

فیلدهای ساختار POPOPrivKey به صورت زیر هستند:

- thisMessage دربردارنده کلید خصوصی رمزبندی شده‌ای است که باید برای آن، گواهی صادر شود. مالکیت کلید خصوصی با عرضه داشتن آن به CA/RA اثبات می‌شود. وقتی که محتوای رمزبندی شده این فیلد کلید خصوصی است، روش صحیح استفاده از این فیلد، ایجاد یک ساختار EncryptedValue و سپس تعریف آن ساختار با نوع BIT STRING است. بنابراین در حال حاضر از این فیلد با توجه به وجود فیلد encryptedKey استفاده نمی‌شود.
- subsequentMessage، برای آن‌که نشان دهد عملیات POP به‌واسطه رمزگشایی یک پیام از CA/RA و بازگشت پاسخ، کامل خواهد شد به کار می‌رود. نوع پیامی که باید رمزگشایی شود، توسط مقادیر زیر نشان داده می‌شود.
  - encrCert نشان می‌دهد که گواهی صادر شده به صورت رمزبندی شده بازگشت می‌یابد. درخواست‌کننده باید گواهی را رمزگشایی و موفقیت خود را برای CA/RA اثبات کند.
  - challengeResp نشان می‌دهد که پیام چالش از CA/RA به درخواست‌کننده ارسال شده است.

- dhMAC برای کلیدهای توافق کلید دیفی-هلمن[1] استفاده می‌شود. این فیلد دربرگیرنده MAC محاسبه شده‌ای است که با استفاده از کلید خصوصی درخواست‌کننده و کلید عمومی CA/RA به دست می‌آید. از این فیلد با توجه به وجود فیلد agreeMAC استفاده نمی‌شود.
- agreeMAC برای کلیدهای توافق کلید به کار می‌رود. این فیلد دربرگیرنده MAC محاسبه شده‌ای است که با استفاده از کلید خصوصی درخواست‌کننده و کلید عمومی CA/RA به دست می‌آید.
  - macAlg دربرگیرنده الگوریتمی است که روش مورد استفاده برای محاسبه مقدار MAC را مشخص می‌کند.
  - macValue دربرگیرنده مقدار محاسبه شده MAC است.

  فیلد agreeMAC در زیرساخت کلید عمومی کشور کاربردپذیر نمی‌باشد.
- encryptedKey شامل کلید خصوصی رمزبندی شده‌ای است که با کلید عمومی که برای آن گواهی صادر شده است منطبق می‌باشد. همچنین این فیلد دربرگیرنده یک مقدار شناسایی است تا نشان دهد توسط درخواست‌کننده گواهی ایجاد شده است. استفاده از این فیلد در زیرساخت کلید عمومی کشور مجاز نمی‌باشد.

۵-۲-۲-۱-۲ ملاحظات مرتبط با چالش و پاسخ

در ادامه ملاحظاتی در مورد چگونگی عملیات اثبات مالکیت غیرمستقیم و مسایلی که لازم است در به کارگیری پیام‌ها برای پیاده‌سازی آن مورد توجه قرار بگیرد، ارائه می‌شود.

۱- درخواست اصلی[2] شامل اثبات شناسانه یک نوع داده و بخش عمومی کلید رمزگذاری است. توجه شود که اثبات شناسانه نوع داده باید بخش عمومی کلید رمزبندی را بپوشاند، تا از وقوع حملات جایگزینی (که در آن مهاجم کلید عمومی درخواست‌کننده را با کلید عمومی خودش جایگزین می‌کند) جلوگیری شود.

۲- پیام پاسخی که از کارساز می‌آید شامل یک داده رمزبندی‌شده از یک نوع داده‌ای است. این مقدار لازم است هنگامی‌که از سمت کارساز می‌آید به طریقی احراز هویت شود. برای کلیدهای RSA، این مقدار می‌تواند با رمزنگاری مستقیم توسط کلید عمومی RSA مشخص شود؛

۳- پیام درخواست دوم شامل چکیده مقدار رمزگشایی شده است.. قرار دادن اطلاعاتی مانند رشته شناسایی در فرآیند درهم‌سازی مطلوب است چرا که به روشن بودن پیام کمک می‌کند. این مقدار بازگشت شده باید در یک مقدار اثبات شناسانه دیگر قرار گیرد.

از شناسانه تراکنش‌ها و مقادیر تصادفی nonce هنگام اجرای عملیات POP به طور غیرمستقیم، باید استفاده شود تا این موارد مجاز باشند: ۱- اتصال پیام‌های مختلف در فرآیند به یکدیگر، ۲- مجاز بودن هر هستار برای وارد کردن مقداری داده تصادفی به عملیات اثبات شناسانه.

---

1 - Diffie-Hellman

۲ - درخواستی که توسط هستار نهایی ارسال می‌شود.

۵-۲-۲-۳ استفاده از MAC مبتنی بر اسم‌رمز

همان‌طور که در زیربند ۵-۲-۲-۱ اشاره شد، ساختار POPOSigningKeyInput که برای اثبات مالکیت کلید خصوصی به کار می‌رود، شامل یک MAC مبتنی بر اسم‌رمز روی مقدار publicKey که با DER کدبندی شده، است. الگوریتم MAC مبتنی بر اسم‌رمز، یک راز به اشتراک گذاشته شده دریافت و از آن به منظور محاسبه مقدار وارسی[1] روی بخشی از اطلاعات استفاده می‌کند. فرض بر آن است که بدون اسم‌رمز، مقدار وارسی صحیح، قابل محاسبه نیست. این الگوریتم برای ایجاد تأخیر در هر نوع حمله مبتنی بر واژه‌نامه به اسم‌رمز، یک تابع یکطرفه را چندین بار محاسبه می‌کند.
شناسانه الگوریتم و ساختار پارامتر استفاده شده برای MAC مبتنی بر کلمه‌عبور به صورت زیر است:

```
id-PasswordBasedMAC OBJECT IDENTIFIER ::=
                                    { 1 2 840 113533 7 66 13}

PBMParameter ::= SEQUENCE {
  salt               OCTET STRING,
  owf                AlgorithmIdentifier,
  iterationCount     INTEGER,
  mac                AlgorithmIdentifier
  )
```

فیلدهای ساختار PBMParameter به صورت زیر هستند:

- salt دربردارنده یک مقدار است که به صورت تصادفی تولید شده است. این مقدار در محاسبه کلید فرآیند MAC کاربرد دارد. توصیه می‌شود که طول salt حداقل ۸ بایت (۶۴ بیت) باشد.
- owf الگوریتم و پارامترهای مرتبط با محاسبه کلیدی که در فرآیند MAC کاربرد دارد را مشخص می‌کند. همه پیاده‌سازی‌ها باید از SHA-1 پشتیبانی کنند.
- iterationCount تعداد دفعاتی که درهم‌سازی درحین فرآیند محاسبه کلید به کار می‌رود را مشخص می‌کند. مقدار کمینه این فیلد ۱۰۰ است. بسیاری از افراد به طور کمینه، مقدار ۱۰۰۰ تعداد تکرار را درنظر می‌گیرند. بین حفاظت از اسم‌رمز در برابر حملات و زمان صرف شده توسط کارساز جهت پردازش تکرارهای مختلف برای استخراج اسم‌رمز، یک موازنه برقرار است. درهم‌سازی به طور کلی به عنوان یک عملیات کم‌هزینه شناخته می‌شود ولی این موضوع در مورد همه توابع درهم‌سازی که در آینده ممکن است پیشنهاد شوند، درست نخواهد بود.
- mac، الگوریتم و پارامترهای آن را برای کاربرد در تابع MAC تعیین می‌کند. در این استاندارد الگوریتم-های مورد تایید جهت پیاده‌سازی MAC شامل HMAC-SHA1 ، HMAC-SHA2، Triple-DES-MAC و AES-MAC می‌باشد. .

1 - check value

در ادامه شبه کد[1] الگوریتم آورده شده است:

ورودی‌ها:

- pw –رشته یک بایتی که دربردارنده اسم‌رمز کاربر است
- data –رشته یک بایتی که دربردارنده مقداری است که باید MAC به آن اعمال شود
- Iter – دفعات تکرار

خروجی:

- MAC – یک رشته بایتی که مقدار MAC محاسبه شده را در خود قرار می‌دهد.

۱- تولید مقدار salt تصادفی S

۲-افزودن salt به pw : K = pw || salt .

۳- درهم‌سازی مقدار K : K=HASH (K)

۴-اگر Iter بیشتر از صفر باشد، Iter = Iter -1 . برو به گام ۳ .

۵- یک HMAC مطابق استاندارد [HMAC][2] محاسبه شود.

MAC = HASH (K XOR opad, HASH (K XOR ipad, data))

opad و ipad در استاندارد مرتبط با HMAC تعریف شده‌اند.

۳-۲-۵ ساختار CertRequest در CRMF

ساختار CertRequest از شناسانه درخواست، الگوی محتوای گواهی، و مجموعه‌ای از اطلاعات کنترلی اختیاری تشکیل شده است.

```
CertRequest ::= SEQUENCE {
    certReqId     INTEGER,        -- ID for matching request and reply
    certTemplate  CertTemplate, --Selected fields of cert to be issued

  CertTemplate ::= SEQUENCE {
    version          [0] Version          OPTIONAL,
    serialNumber [1] INTEGER          OPTIONAL,
```

1 - pseudo-code

2 - The Keyed-Hash Message Authentication Code, FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION (FIPS) 198

```
    signingAlg      [2] AlgorithmIdentifier   OPTIONAL,
    issuer          [3] Name                  OPTIONAL,
    validity        [4] OptionalValidity      OPTIONAL,
    subject         [5] Name                  OPTIONAL,
    publicKey       [6] SubjectPublicKeyInfo  OPTIONAL,
    issuerUID       [7] UniqueIdentifier      OPTIONAL,
    subjectUID      [8] UniqueIdentifier      OPTIONAL,
    extensions      [9] Extensions            OPTIONAL }

  OptionalValidity ::= SEQUENCE {
    notBefore       [0] Time OPTIONAL,
    notAfter        [1] Time OPTIONAL } --at least one must be present

  Time ::= CHOICE {
    utcTime       UTCTime,
    generalTime   GeneralizedTime }
```

فیلدهای ساختار CertRequest در زیر تعریف شده‌اند:

- certReqId دربردارنده یک مقدار صحیح است. درخواست‌کننده صدور گواهی، از این مقدار برای ارتباط یک درخواست صدور گواهی مشخص به یک پاسخ گواهی استفاده می‌کند.
- certTemplate دربردارنده یک الگوی گواهی X.509 است. درخواست‌کننده آن فیلدهایی که مقدارشان مطلوب است را پر می‌کند. جزییات بیشتر در مورد این فیلدها در ادامه تعریف شده است.

توصیف فیلدهای ساختار CertTemplate به شرح زیر می‌باشد:

- version درصورت پشتیبانی باید برابر با مقدار ۲ باشد. بهتر است این فیلد حذف شود.
- serialNumber باید حذف شود. این فیلد در هنگام تولید گواهی مقدار داده می‌شود.
- signingAlg باید حذف شود. این فیلد در هنگام تولید گواهی مقدار داده می‌شود.
- issuer به طور معمول حذف می‌شود. این فیلد توسط مرکز صدورگواهی که درخواست‌کننده قصد دارد از آن گواهی دریافت کند، در حالتی پر می‌شود که یک RA به چند CA خدمت می‌دهد. در این صورت نام CA در این فیلد درج می‌شود.
- validity به طور معمول حذف می‌شود. این فیلد می‌تواند برای درخواست اینکه گواهی‌ها در زمان خاصی در آینده آغاز شوند یا در زمان مشخصی به انقضا برسند، استفاده شود. به عنوان مثال حالتی که این فیلد به طور معمول استفاده می‌شود، صدور یک گواهی متقابل[1] برای یک CA است. در این حالت، اعتبارسنجی گواهی موجود در این فیلد قرار می‌گیرد و گواهی جدید دوره اعتبار یکسانی با گواهی موجود خواهد داشت. اگر اعتبار حذف نشود، به طور کمینه یکی از زیرفیلدها باید مشخص شود. در ادامه این زیر فیلدها آمده‌اند:

1 - Cross certificate

- notBefore دربردارنده زمان شروع درخواست‌شده برای گواهی است. زمان درج شده از قواعد زمان در فیلد notBefore مورد اشاره در مستند جامع پروفایل های زیرساخت کلید عمومی کشور که توسط مرکز دولتی صدور گواهی الکترونیکی ریشه منتشر شده پیروی می‌کند.
- notAfter دربردارنده زمان انقضای درخواست‌شده برای گواهی است. زمان درج شده از قواعد زمان در فیلد notAfter مورد اشاره در سند جامع پروفایل های زیرساخت کلید عمومی کشور پیروی می‌کند.
- subject نام پیشنهادی برای درخواست‌کننده را در خود نگهداری می‌کند. این فیلد به‌طور معمول با نامی که در قبل برای درخواست‌کننده توسط CA صادرشده، پر می‌شود.
- publicKey دربرگیرنده کلید عمومی که برای آن گواهی صادر شده است، می‌باشد. اگر درخواست‌کننده کلید خود را تولید کرده باشد، این فیلد باید پر شود. اگر کلید توسط RA/CA تولید شده بود، این فیلد حذف می‌شود.
- issuerUID باید حذف شود.
- subjectUID باید حذف شود.
- extentions دربردارنده الحاقیاتی است که درخواست‌کننده می‌خواهد در گواهی قرار دهد. این الحاقیات به‌طور کلی به مواردی مانند تنظیم کاربرد کلید در keyEncipherment رسیدگی می‌کنند.

به استثنای فیلد publicKey، CA/RA اجازه تغییر هر فیلد درخواستی را دارد. درخواست‌کننده باید گواهی بازگشت داده شده را بررسی کند تا یقین یابد فیلدها به صورت قابل قبولی تنظیم شده‌اند. بهتر است در صورت امکان CA/RA از فیلدهای الگو استفاده کند.

مواردی وجود داردکه فیلدهای الگو می‌توانند حذف شوند. اگر تولید کلید در CA/RA انجام شود و اثبات شناسانه در مکان دیگری باشد (مثل id-regCtrl-regToken) آن‌گاه فیلدی وجود ندارد تا لازم باشد که درخواست‌کننده آن‌را مشخص کند.

٦ الزامات پروتکل درخواست صدور گواهی مبتنی بر CMC

در این بند، الزامات مرتبط با مدیریت درخواست صدور گواهی الکترونیکی در صادرکنندگان گواهی و مراجع ثبت، در قالب پروتکل درخواست صدور گواهی مبتنی بر CMC ارائه شده است. در واقع CMC ارائه دهنده یک **پروتکل مدیریت گواهی الکترونیکی** بر اساس ساختار نحوی پیام‌های رمزنگاشتی(CMS)[1] است.[2] در این استاندارد نحوه مدیریت درخواست‌های صدور گواهی الکترونیکی از پروتکل مزبور تبعیت می‌کند و در این بند به توصیف آن پرداخته شده است.

1 - Cryptographic Message Syntax

۲- الزامات مرتبط با نحو پیام‌های رمزنگاشتی در استاندارد ملی ایران به شماره ۱۷۱۱۳: سال ۱۳۹۲، منتشر شده است.

پروتکل درخواست صدور گواهی مبتنی بر CMC سه نیاز ضروری در زیرساخت کلید عمومی کشور را برآورده می‌کند:

۱- هماهنگ‌سازی درخواست صدور گواهی با ساختارهای نحوی CMS جهت استفاده در RA به منظور ارائه درخواست‌های درستی‌سنجی شده به CA؛

۲- تولید و ارائه درخواست‌های گروهی صدور گواهی الکترونیکی[1]؛

۳- تعامل‌پذیری و آزمون‌پذیری سامانه مدیریت درخواست صدور گواهی الکترونیکی در مراجع ثبت و صادرکنندگان گواهی.

۶-۱ملاحظات مرتبط با پروتکل درخواست صدور گواهي مبتني بر CMC

در پروتکل درخواست صدور گواهی مبتنی بر CMC ملاحظات ذیل باید در نظر گرفته شود.

- پروتکل باید هر چقدر که امکان دارد با ساختار پیام رمزنگاشتی(CMS) تعریف شده در استاندارد ملی ایران به شماره ۱۷۱۱۳: سال ۱۳۹۲[2]، PKCS#10 و قالب پیام درخواست صدور گواهی (CRMF) که در زیربند ۵-۲ این استاندارد تعریف گردید، سازگار باشد.
- پروتکل باید همچنان از رویه موجود که طی آن برای یک درخواست صدور گواهی PKCS#10 ، یک پاسخ "cert-only" مبتنی بر PKCS#7 ارسال می‌شود پشتیبانی کند.

- پروتکل باید قادر به پشتیبانی آسان از پروتکل‌های ثبت‌نام با چند کلید که مورد نیاز S/MIME[3] یا دیگر گروه‌ها هستند، باشد.
- پروتکل باید به‌گونه‌ای طراحی شود که امکان انجام عملیات تولید کلید در سمت کاربر فراهم شود.
- پروتکل باید دربرگیرنده روش‌های عملیات اثبات مالکیت (POP) منطبق با این استاندارد باشد.
- پروتکل باید از پاسخ به درخواست‌های ثبت‌نام معلق یا عقب افتاده در مواردی که رویه‌های خارجی برای صدور یک گواهی لازم است، پشتیبانی کند.
- پروتکل باید از مجموعه‌ای از مراجع ثبت (RA) به عنوان رابط بین درخواست‌کننده‌های صدور گواهی و مراکز صدور گواهی (CA) پشتیبانی کند.

۶-۲ مرور اجمالی بر پروتکل درخواست صدور گواهي مبتني بر CMC

به‌طور کلی در این استاندارد یک تراکنش ثبت‌نام PKI یک رفت و برگشت ساده پیام‌ها را در بر می‌گیرد. در ساده‌ترین حالت، یک درخواست ثبت‌نام PKI، که از این پس درخواست PKI نامیده می‌شود، از سمت کاربر

---

1 -Batch CSR

2 - PKCS #7

۳- Secure/Multipurpose Internet Mail Extensions، استانداردی برای رمزنگاری کلید عمومی و امضای اطلاعات MIME است.

به کارساز ارسال می‌شود و یک پاسخ ثبت‌نام PKI، که از این پس پاسخ PKI نامیده می‌شود، از سوی کارساز به کارخواه باز می‌گردد. در حالت‌های پیچیده‌تر مانند صدور گواهی با وقفه بیش از یک رفت و برگشت لازم است.

در این استاندارد دو نوع درخواست و دو نوع پاسخ PKI تعریف می‌شود. درخواست‌های PKI با استفاده از PKCS#10 یا ساختار CRMF شکل می‌گیرند. دو نوع درخواست PKI عبارتند از:

- درخواست PKI ساده: فقط با PKCS#10 آماده می‌شود. (هنگامی‌که به خدمت‌های دیگری نیاز نباشد، این نوع درخواست استفاده می‌شود)

- درخواست PKI کامل: یک یا چند PKCS#10، CRMF یا دیگر ساختارهای پیام درخواست (دسته‌ای از درخواست‌ها)، که در قالب یک ساختار CMS به عنوان بخشی از PKIData قرار می گیرد.

پاسخ PKI براساس SignedData یا AuthenticatedData (که در استاندارد ملی ایران به شماره ۱۷۱۱۳: سال ۱۳۹۲، الزامات ساختارنحوی پیام‌های رمزنگاشتی در زیرساخت کلید عمومی ایران ، تعریف شده است) می-باشد. دو نوع پاسخ PKI عبارتند از:

- پاسخ PKI ساده: ساختار داده SignedData و به شکل " cert-only" است (درصورتی که به خدمت‌های دیگری نیاز نباشد).

- پاسخ PKI کامل: ساختار داده PKIResponse که در SignedData پنهان می‌شود.

در این استاندارد خدمت‌های ویژه‌ای برای تمدید گواهی[1] (یعنی یک گواهی جدید با همان کلیدی که گواهی قبلی برای آن صادر شده بود) یا کلیدگذاری مجدد گواهی[2] (یعنی یک گواهی جدید با کلید جدید صادر شود) برای گواهی کاربر فراهم نشده است. درخواست‌های کلیدگذاری مجدد یا تمدید گواهی همانند درخواست صدور گواهی است به‌جز اینکه، عملیات اثبات هویت می‌تواند با استفاده از گواهی موجود و از طریق یک CA مورد اعتماد انجام می‌شود.

برای تمایز بین یک درخواست کلیدگذاری مجدد و درخواست صدور گواهی جدید (به طور عمومی با هدف جدید) خدمت‌های ویژه‌ای فراهم نشده است. به‌طور معمول یک کنترل برای عدم انتشار درخواست به-روزرسانی کلید، در درخواست کلیدگذاری مجدد قرار می‌گیرد و در درخواست صدور گواهی جدید حذف می‌شود. از مراکز CA یا دیگر عامل‌های انتشار نیز انتظار می‌رود که برای حذف گواهی‌ها از فهرست انتشار، براساس تولید گواهی‌های جدید، انقضاء یا ابطال گواهی‌های قبلی، خط‌مشی‌هایی داشته باشند.

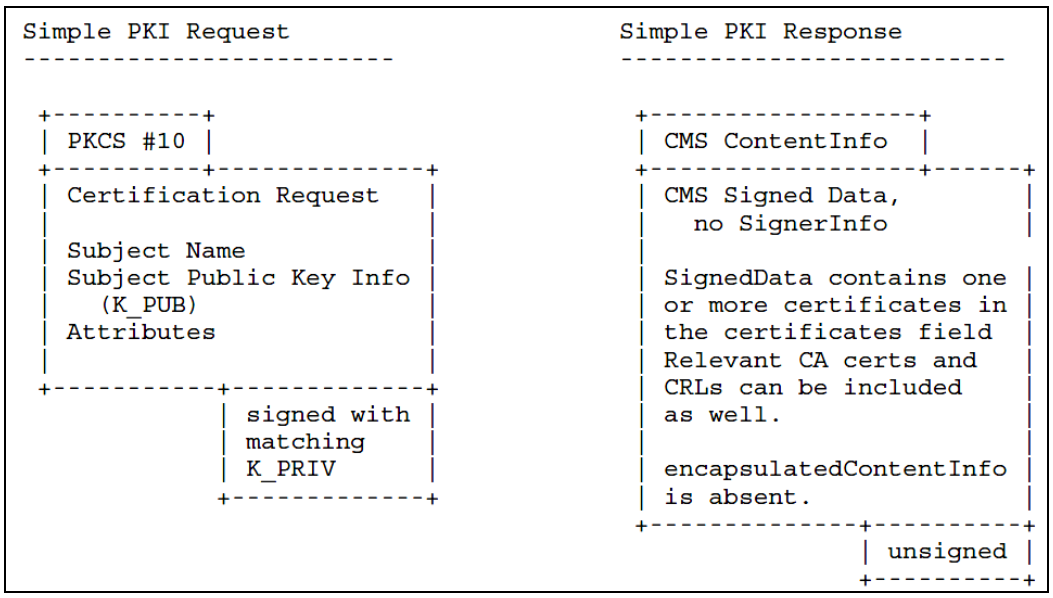
1 - renewal
2 - rekey

همچنین مراکز RA می‌توانند درخواست‌های PKI را دریافت، آنها را در لایه دوم درخواست PKI بـه همـراه الزامات یا ملاحظاتی (از RA) پوشانده و آنگاه درخواست PKI جدید و توسعه‌یافته را به CA ارسال کنند.

این استاندارد به انواع روش‌های انتقال پیـام‌هـای CMC نمـی پـردازد. اسـتفاده از CMS بـه معنـی انتقـال درخواست‌ها با استفاده از خـدمت رایانامـه (MIME یـا S/MIME) نیسـت. روش‌هـای انتقـال در rfc5273 تبیین شده است.

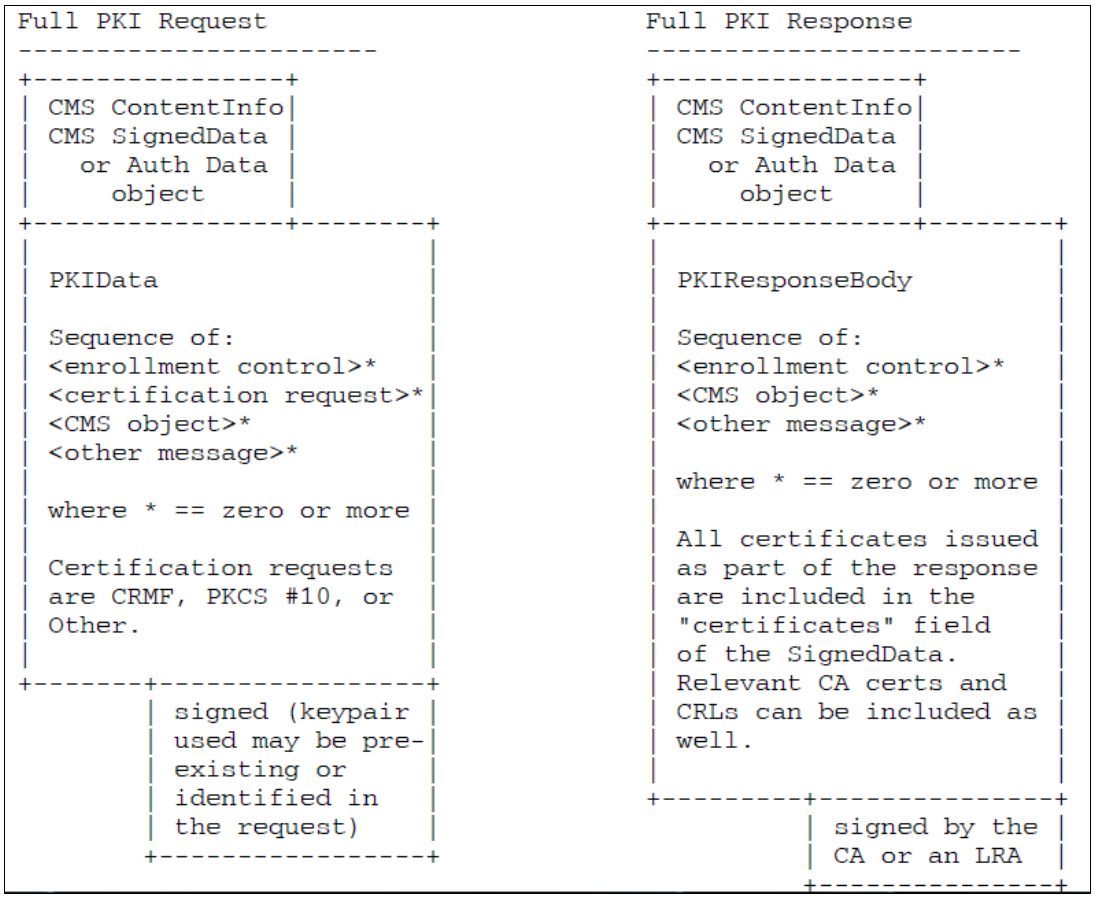
۶-۳ پیام‌های درخواست/پاسخ

شکل ۱ درخواست و پاسخ PKI ساده را نشان می‌دهد. در بندهای آتی به جزییات بیشتری از آن‌ها پرداختـه می‌شود.



شکل ۱ – درخواست و پاسخ ساده

شکل ۲ درخواست و پاسخ PKI کامل را نشان می‌دهد. در بندهای آتی به جزییات بیشتری از آن‌ها پرداخته می‌شود.



شکل ۲ – درخواست و پاسخ کامل

۷ درخواست‌های PKI

دراین بند به بیان جزییات دونوع درخواست PKI پرداخته می‌شود.

۷-۱ درخواست ساده

این درخواست از ساختار CertificationRequest که در PKCS#10 تعریف شده است، پیروی می‌کند. هنگامی‌که کارساز درخواست ساده PKI را پردازش می‌کند، در پاسخ اگر آن درخواست موفقیت‌آمیز بود، یک پاسخ ساده PKI برگشت داده می‌شود و درصورت عدم موفقیت، با پاسخ کامل PKI جواب می‌دهد. البته کارساز ممکن است در صورت عدم موفقیت، درخواست ساده ارسال شده، آن‌را بدون پاسخ بگذارد.

از درخواست ساده درصورتی که به اثبات هویت نیاز باشد، نباید استفاده کرد. همچنین درصورتی که کلید خصوصی قابلیت تولید برخی از انواع امضا را نداشته باشد، نمی‌تواند مورد استفاده قرار بگیرد. همچنین درخواست ساده برای خدمات ویژه‌ای که دراین استاندارد تبیین می‌شوند، نمی‌تواند مورد استفاده قرار بگیرد.

کارخواه ممکن است به عنوان یک صفت ExtensionReq یک یا چند الحاقیه X.509v3 را در درخواست صدور گواهی برپایه PKCS#10 قرار دهد. این صفت به صورت زیر تعریف می‌شود:

ExtensionReq ::= SEQUENCE SIZE (1..MAX) OF Extension

در این ساختار Extension از مستند جامع پروفایل‌های زیرساخت کلید عمومی کشور وارد می‌شود و ExtensionReq نیز به صورت زیر تعیین می‌گردد:

id-ExtensionReq OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840)

rsadsi(113549) pkcs(1) pkcs-9(9) 14}

کارساز باید قادر به پردازش همه الحاقیات تعریف شده(و نه منع شده) در مستند جامع پروفایل‌های زیرساخت کلید عمومی کشور بسته به کاربرد باشد. کارسازان لازم نیست قابلیت پردازش الحاقیاتی غیر از X.509v3 که با استفاده از این پروتکل، انتقال می‌یابند و نیز الحاقیات خصوصی را داشته باشند. لازم نیست که همه الحاقیات درخواست شده کاربر در یک گواهی قرار گیرد. کارسازان مجاز به اصلاح الحاقیات مورد درخواست کاربر هستند ولی نباید آن‌ها را به گونه‌ای تغییر دهند (به طور مثال تغییر کاربرد کلید از KeyEncipherment به digitalSignature) که منجر به انحراف از هدف الحاقیه مورد درخواست کارخواه گردند. اگر درخواستی به دلیل عدم امکان استفاده از یک الحاقیه درخواست شده، رد شود و یک پاسخ PKI بازگردد، کارساز باید در آن پاسخ، در فیلد CMCFailInfo مقدار unsupportedExt را قرار دهد.

یک درخواست ساده PKI از شناسه منحصر به فرد مالک گواهی، کلید عمومی، مجموعه‌ای از صفات اختیاری که همگی توسط هستار درخواست‌کننده گواهی امضا شده‌اند، تشکیل شده است. درخواست‌های گواهی به CA ارسال شده و به شکل یک گواهی کلید عمومی X.509 تبدیل می‌گردد.

مجموعه صفات اختیاری[1] دو هدف را برآورده می‌کنند: تهیه اطلاعات بیشتر در مورد یک هستار معلوم (به طور مثال تعیین یک چالش-اسم‌رمز برای اینکه در آینده هستار بتواند ابطال یک گواهی را درخواست کند)، یا فراهم آوردن اطلاعات دیگر به منظور درج در گواهی‌های X.509 (مانند آدرس پستی درخواست‌کننده گواهی جهت تحویل گواهی امضا شده برای حالاتی که رایانامه وجود ندارد).

مراکز صدور گواهی ممکن است درخواست‌های گواهی را به شکل‌های غیرالکترونیکی دریافت و با همین روش نیز پاسخ دهند. انتظار می‌رود اطلاعاتی که برای این نوع درخواست و پاسخ لازم است، توسط مراکز صدور گواهی فراهم گردد.

۷-۱-۱ ساختار درخواست‌های ساده گواهی PKI

ساختار درخواست‌های ساده PKI طبق PKCS#10 توصیف می‌شود که در بخش ۵-۱ به طور کامل شرح داده شد.

---

۱ - فهرستی از صفات اختیاری در PKCS#9 موجود است.

۲-۷ درخواست کامل

این نوع درخواست کارایی و انعطاف‌پذیری بیشتری نسبت به درخواست ساده PKI فراهم می‌کند. درخواست کامل یا در SignedData یا در AuthenticatedData به همراه یک ساختار کپسوله شده id-cct-PKIData پوشانده می‌شود.

وقتی‌که یک کارساز درخواست کامل PKI را پردازش می‌کند، یک پاسخ PKI باید بازگشت داده شود. این پاسخ می‌تواند به صورت زیر باشد:

- پاسخ ساده PKI - اگر ثبت‌نام موفقیت‌آمیز باشد، در این صورت فقط گواهی‌ها (بدون کنترل) بازگشت داده می‌شود.
- پاسخ کامل PKI – اگر ثبت‌نام موفقیت‌آمیز باشد و اطلاعات بیشتری در تکمیل گواهی بازگشت داده شود. همچنین درصورتی که ثبت‌نام معلق یا مردود گردد این پاسخ بازگشت داده می‌شود.

درصورت استفاده از SignedData، می‌توان با استفاده از کلید خصوصی یک درخواست صدور گواهی امضای ادغام شده (یعنی در فیلدهای tcr (PKCS#10) یا crm (CRMF) از TaggedRequest) یا با استفاده از کلید امضایی که از قبل گواهی شده است، امضا را تولید کرد. اگر کلید خصوصی یک درخواست صدور گواهی امضا استفاده شود، آن‌گاه موارد زیر باید اجرا شوند:

۱- درخواست صدور گواهی که حاوی کلید عمومی متناظر است باید الحاقیه شناسانه کلید مالک گواهی را دارا باشد.

۲- باید فرم subjectKeyIdentifier ساختار signerIdentifier در SignerInfo به کار رود.

۳- مقدار subjectKeyIdentifier ساختار SignerInfo باید معادل با شناسانه کلید مالک گواهی در درخواست صدور گواهی متناظر باشد (دلیل استفاده از شکل subjectKeyIdentifier از ساختار SignerInfo این است که برای کلید امضا تا کنون گواهی صادر نشده است). اگر کلید درخواست برای امضا کردن استفاده می‌شود، باید فقط یک SignerInfo در SignedData وجود داشته باشد.

درصورت استفاده از AuthenticatedData موارد زیر باید مد نظر قرار گیرد:

۱- باید از گزینه اطلاعات گیرنده اسم‌رمز[1] در RecipientInfo استفاده شود.

۲- از کلیدی که به صورت تصادفی تولید شده برای محاسبه مقدار کد اصالت‌سنجی پیام (MAC) بر روی محتوای کپسوله شده، استفاده می‌شود.

۳- ورودی الگوریتم استخراج کلید، الحاق شناسانه (کدبندی شده به صورت UTF8) و راز به اشتراک گذاشته شده (اسم‌رمز) است.

1 - Password Recipient Info

در هنگام ایجاد یک درخواست PKI برای کلیدگذاری مجدد گواهی یا تمدید گواهی موارد زیر درنظر گرفته می‌شود:

۱- کنترل‌های اثبات هویت و احراز هویت وجود ندارند. این اطلاعات با استفاده از گواهی موجود از CA در هنگام امضای درخواست PKI فراهم می‌شود. به طور معمول در این حالت، مرکز صدور گواهی که گواهی اصلی (اولیه) را صادر کرده است و صادرکننده گواهی که درخواست برای آن شکل یافته است، یکی هستند، اما می‌توانند یک متصدی مشترک داشته باشند.

۲- مراکز CA و RA می‌توانند محدودیت‌های بیشتری به گواهی امضا تحمیل کنند. به طور مثال ممکن است این مراکز آخرین گواهی امضا را برای کارخواه به کار ببرند.

۳- برخی از مراکز CA ممکن است از عملیات تمدید گواهی (یعنی استفاده مجدد از کلیدهای قبلی) جلوگیری کنند. در این صورت CA باید یک پاسخ PKI با مقدار noKeyReuse به عنوان کد خطای فیلد CMCFailInfo بازگشت دهد.

۷-۲-۱ ساختار داده PKIData

برای درخواست کامل PKI ساختار داده PKIData استفاده می‌شود. این ساختار به صورت زیر مشخص می‌شود:

```
id-cct-PKIData ::= {id-pkix id-cct(12) 2 }
```

ساختار ASN.1 آن به صورت زیر است:

```
PKIData ::= SEQUENCE {
        controlSequence         SEQUENCE SIZE(0..MAX) OF
    TaggedAttribute,
        reqSequence             SEQUENCE SIZE(0..MAX) OF TaggedRequest,
        cmsSequence             SEQUENCE SIZE(0..MAX) OF
    TaggedContentInfo,
        otherMsgSequence        SEQUENCE SIZE(0..MAX) OF OtherMsg
        }
```

تعابیر فیلدهای PKIData به صورت زیر است:

- controlSequence – دنباله‌ای از کنترل‌ها است. کنترل‌هایی که در این استاندارد از آن‌ها استفاده می‌شود، در پیوست الف معرفی می‌شوند. کنترل‌ها می‌توانند توسط اشخاص دیگری نیز تعریف گردند. جزییات ساختار TaggedAttribute در زیربند ۷-۲-۱-۱ تعریف می‌گردد.
- reqSequence – دسته یا دنباله‌ای از درخواست‌های گواهی[1] است. درخواست‌های گواهی می‌تواند یک درخواست صدور گواهی ساده (در قالب CertificationRequest در PKCS#10، یا CertReqMsg در

1 -CSR Batch

CRMF کـه در ایـن اسـتاندارد تعریـف شـده اسـت، باشـد. جزییـات بیشـتر در ایـن خصـوص در زیربند ۷-۲-۱-۲ آمده است. درصورتی که یک درخواست PKI خارج از این استاندارد وجود داشته باشد، ولی کارساز آن را درک نکند (یا آن‌را پردازش نکند)، فیلـد CMCStatus بـا مقـدار noSupport بایـد در پاسخ به درخواست صدور گواهی ارسال شود و دیگر درخواست‌های گواهی پردازش نگردند.

- cmsSequence – دنباله‌ای از ساختارهای پیام CMS است. زیربند ۷-۲-۱-۳ به جزییـات بیشـتر اشـاره می‌کند.
- otherMsgSequence – دنباله‌ای از داده های اختیاری است. داده‌هایی که در این فیلد قـرار مـی‌گیرنـد توسط یک یا چند کنترل مورد ارجاع واقع می‌شوند که موجب می‌شود تا بدون ایـن کـه داده در کنتـرل ادغام شود، کنترل‌ها قادر باشند از مقادیر زیادی از داده استفاده کنند. به زیربند ۷-۲-۱-۴ برای جزییات بیشتر مراجعه شود.

بهتر است کلیه درخواست‌های گواهی که در یک PKIData کدبندی می‌شوند به مقصد یک هسـتار باشـند. مراکز RA که قابلیت پردازش دسته‌ای دارند باید درخواست‌های PKI دریافتی را درون فیلد cmsSequence ساختار PKIData قرار داده و به صورت امضا شده، ارسال کنند.

پردازش PKIData توسط گیرنده به صورت زیر است:

۱- کلیه کنترل‌ها باید بررسی شده و به یک روش مناسب پردازش گردند. روش مناسب پردازش بـه صـورت اتمام پردازش در لحظه ورود، نادیده گرفتن کنترل‌ها یا قراردادن کنترل‌ها در فهرست کارهـای در دسـت اقدام برای پردازش‌های آتی است. کنترل‌ها به هر ترتیبی می‌توانند پردازش شوند؛ بنابراین ترتیـب آن‌هـا در دنباله مهم نیست.

۲- مواردی که در reqSequence هستند، به عنوان کنترل قلمداد نمی‌شوند. این موارد که درخواست‌های گواهی هستند نیز باید پردازش شوند. همانند کنترل‌ها، پردازش مناسب به صورت فوری و در لحظه یا به صورت افزودن در فهرست کارهای در دست اقدام برای پردازش‌های آتی می‌تواند انجام شود.

۳- در نهایت، فهرست کارهای دردست اقدام پردازش می‌شود. در بسیاری از حالات این فهرست به صورت قرار دادن وظایف مشخص در یک گروه، مرتب می‌شود.

درصورتی که فیلدهای cmsSequence یا otherMsgSequence از ساختار PKIData وجود داشـته باشـند و کنترل خاصی آنها را مورد ارجاع قرار نداده باشد، نیاز به پردازش ندارنـد. یعنـی از ایـن دو فیلـد صـرف‌نظـر می‌شود.

۷-۲-۱-۱ ساختار کنترل

فعالیت‌هایی که روی درخواست/پاسخ PKI انجام می‌شود، براساس کنترل‌هایی است که در ساختار آن پیام‌ها قرار می‌گیرد. هر کنترل متشکل از یک شناسانه و مقداری براساس آن است. ساختار یک کنترل بـه صـورت زیر است:

```
TaggedAttribute ::= SEQUENCE {
                    bodyPartID      BodyPartID,
                    attrType        OBJECT IDENTIFIER,
                    attrValues      SET OF AttributeValue
                    }

AttributeValue ::= ANY
```

تعابیر فیلدهای TaggedAttribute به شرح زیر است:

- bodyPartID – یک مقدار صحیح منحصر به فرد است که کنترل را مشخص می‌کند.
- attrType – OID مربوط به کنترل است.
- attrValues – مقادیری از نوع داده است که در پردازش کنترل کاربرد دارد. ساختار این داده بسته به نوع کنترل است.

درصورتی که کنترلی درون PKIData تشخیص داده نشود یا این که آن کنترل در قبل توسط کنترل Control Processed پردازش نشده باشد (به زیربند الف-۱۴ مراجعه شود) و خطای دیگری تولید نشده باشد، کارساز نهایی باید پردازش کل ساختار داده را متوقف کند. پاسخ PKI باید شامل CMCFailInfo با مقدار badRequest باشد و فیلد bodyList نیز bodyPartID کنترل(های) نادرست یا نامشخص را شامل گردد. یک کارساز، کارساز نهایی خواهد بود اگر و تنها اگر درخواست PKI به کارساز دیگری ارسال نشود. اگر کارساز قصد انتقال درخواست PKI را داشته باشد ولی در عوض خطای پردازش درخواست را بازگشت دهد، به عنوان کارساز نهایی درنظر گرفته نمی‌شود.

کنترل‌های تعریف شده در این استاندارد در پیوست الف آورده شده است.

۷-۲-۱-۲ قالب‌های درخواست گواهی

درخواست‌های گواهی می‌توانند در قالب PKCS#10 یا CRMF باشند. زیربند ۵-۱ الزاماتی برای استفاده از PKCS#10 بیان می‌کند و در زیربند ۵-۲ الزاماتی برای CRMF آورده شده است.

```
TaggedRequest ::= CHOICE {
            tcr [0] TaggedCertificationRequest,
            crm [1] CertReqMsg,
            orm [2] SEQUENCE {
                bodyPartID              BodyPartID,
                requestMessageType      OBJECT IDENTIFIER,
                requestMessageValue     ANY DEFINED BY requestMessageType
                                }
                }
```

فیلدهای TaggedRequest دارای تعابیر زیر هستند:

- tcr – درخواست‌های گواهی است که از ساختار PKCS#10 تبعیت می‌کند.
- crm – درخواست‌های گواهی است که از ساختار CRMF تبعیت می‌کند.
- orm – درخواست‌های گواهی که در خارج از این استاندارد تعریف می‌شوند. مثالی از این نوع درخواست، ساختار درخواست صدور گواهی صفت است. این ساختار در زیرساخت کلید عمومی کشور کاربردپذیر نمی‌باشد. فیلدهای این ساختار به صورت زیر است:
  - bodyPartID – شناسانه این نوع درخواست صدور گواهی است.
  - requestMessageType – نوع سایر درخواست‌ها را مشخص می‌کند. تعریف این فیلد در حوزه استاندارد حاضر نیست.
  - requestMessageValue – داده همراه با سایر انواع درخواست‌ها است.

۷-۲-۱-۲-۱ سایر درخواست‌های گواهی

این استاندارد تعریف و استفاده از قالب‌های دیگری برای درخواست‌های گواهی را مجاز نمی‌داند.

۷-۲-۱-۳اشیاء موجود در اطلاعات محتوا[1]

فیلد cmsSequence در پیام‌های PKIData و PKIResponse شامل اشیاء اطلاعات محتوای برچسب‌دار یا بدون آن است. این ساختار به صورت زیر است:

```
TaggedContentInfo ::= SEQUENCE {
        bodyPartID BodyPartID,
        contentInfo ContentInfo
}
```

تعابیر فیلدهای این ساختار در زیر آمده است:

- bodyPartID – یک مقدار صحیح یکتا است که این شی اطلاعات محتوا را مشخص می‌کند.
- contentInfo یک شی ContentInfo (که در استاندارد ملی ایران به شماره۱۷۱۱۳: سال ۱۳۹۲، الزامات ساختارنحوی پیام‌های رمزنگاشتی در زیرساخت کلید عمومی ایران تعریف شده است) می‌باشد.

چهار نوع ساختار داده که در cmsSequence استفاده می‌شوند، عبارتند از: AuthenticatedData، Data، EnvelopedData و SignedData. این ساختار داده‌ها در استاندارد ملی ایران به شماره۱۷۱۱۳: سال ۱۳۹۲، الزامات ساختارنحوی پیام‌های رمزنگاشتی در زیرساخت کلید عمومی ایران تعریف شده‌اند.

۷-۲-۱-۳-۱ داده احراز هویت شده

1 Content Info Objects

ساختار داده AuthenticatedData روشی برای انجام اعتبارسنجی مبتنی بر راز از پیش اشتراک گذاشته شده برای داده‌ای که بین دو طرف ارسال می‌شود، فراهم می‌کند. این روش برخلاف SignedData مشخص نمی‌کند که کدام طرف در اصل داده را تولید کرده است.

در مواردی که راز به اشتراک گذاشته شده وجود داشته باشد ولی اعتماد مبتنی بر PKI استقرار نیافته باشد، AuthenticatedData امکان احراز هویت مبدأء را فراهم می‌کند. ممکن است که در نرم‌افزار کارخواه، گواهی مرجع اعتماد[1] نصب نشده باشد یا گواهی برای یک کلید امضا موجود نباشد و لذا اعتماد مبتنی بر PKI استقرار نیابد.

در این استاندارد از نوع AuthenticatedData برای موارد زیر استفاده می‌شود:

- کنترل id-cmc-authData (طبق زیربند الف-۱۲)

- در محیط‌هایی که یک گواهی برای کلید مختص رمزگذاری ایجاد می‌گردد از آن به عنوان پوشاننده[2] سطح بالای داده استفاده می‌شود.

این ساختار داده می‌تواند انواع PKIData و PKIResponse را در خود جای دهد. این ساختار داده‌های ادغام شده می‌توانند کنترل‌های بیشتری برای پردازش دارا باشند.

۷-۲-۱-۳-۲ داده

ساختار داده Data، انتقال داده‌های بدون ساختار را امکان‌پذیر می‌سازد. ساختار داده Data در این استاندارد در مورد زیر کاربرد دارد:

- نگهداری مقدار تصادفی رمزشده y برای اثبات POP در کنترل رمزگذاری شده POP (طبق پیوست).

۷-۲-۱-۳-۳ داده پوشیده شده

این ساختار داده امکان پوشاندن داده را فراهم می‌کند. در این استاندارد ساختار داده EnvelopedData ، یک روش اولیه محرمانگی برای اطلاعات حساس است. EnvelopedData می‌تواند برای کل درخواست PKI امکان رمزگذاری را به وجود بیاورد (کل درخواست PKI را رمزبندی کند). اگر رمزگشایی EnvelopedData ناموفق باشد، یک پاسخ کامل PKI با مقدار badMessageCheck برای فیلد CMCFailInfo و مقدار صفر برای فیلد bodyPartID باید بازگشت داده شود.

۷-۲-۱-۳-۴ داده امضا شده

ساختار داده SignedData برای احرازهویت و اطمینان از یکپارچگی داده کاربرد دارد. این ساختار داده در این استاندارد برای موارد زیر استفاده می‌شود:

---

1 - Trust anchor
2 - Wrapper

- پوشاننده خارجی درخواست PKI
- پوشاننده خارجی پاسخ PKI

به عنوان بخشی از پردازش درخواست/پاسخ PKI ، امضا(ها) باید اعتبارسنجی شده باشند. اگر امضایی بررسی نگردد و درخواست/پاسخ حاوی چیزی غیر از یک کنترل اطلاعات وضعیت CMC باشد، باید یک پاسخ کامل PKI که دربردارنده کنترل اطلاعات وضعیت CMC است، بازگشت داده شود. در این پاسخ فیلد CMCFailInfo با مقدار badMessageCheck و مقدار صفر برای فیلد bodyPartID باید وجود داشته باشد.
در پاسخ PKI، SignedData این امکان را فراهم می‌کند که کارساز داده بازگشتی را امضا کند (درصورتی که وجود داشته باشد) و گواهی‌ها و CRLهای متناظر با درخواست PKI را ارسال کند. در حالتی که هیچ داده بازگشتی وجود ندارد و فقط گواهی و CRL ارسال می‌شود، وجود فیلدهای EncapsulatedInfo و SignedInfo نیز لزومی نخواهد داشت.

۷-۲-۱-۴ دیگر بدنه‌های پیام

فیلد otherMsgSequence موجود در درخواست/پاسخ PKI به اشیاء داده اختیاری اجازه می‌دهد که به عنوان بخشی از درخواست/پاسخ PKI ارسال گردند. این موضوع برای دربرگرفتن یک شی داده است که در قبل در یک فیلد cmsSequence پوشانده نشده است (طبق زیربند ۷-۲-۱-۳). شی داده به جز زمانی که یک کنترل، آن را با bodyPartID مورد ارجاع قرار دهد، نادیده گرفته می‌شود.

```
OtherMsg ::= SEQUENCE {
        bodyPartID BodyPartID,
        otherMsgType OBJECT IDENTIFIER,
        otherMsgValue ANY DEFINED BY otherMsgType
        }
```

فیلدهای OtherMsg به صورت زیر تعبیر می‌شود:

- bodyPartID – شناسانه یکتایی است که به این شی داده اشاره می‌کند.
- otherMsgType – OID معرف نوع بدنه پیام است.
- otherMsgValue – داده در این ساختار است.

۷-۲-۲ شناسایی بدنه پیام

هر عنصر از PKIData یا PKIResponse دارای یک شناسانه بدنه پیام[1] است. شناسانه بدنه پیام یک مقدار صحیح ۴ بایتی است و ساختار ASN.1 آن به صورت زیر می‌باشد:

```
bodyIdMax INTEGER ::=  4294967295
BodyPartID ::= INTEGER (0..bodyIdMax)
```

1 - Body part ID

شناسانه‌های بدنه پیام در فیلد certReqIds ساختار CertReqMsg (در TaggedRequest) یا در فیلد bodyPartID دیگر اشیاء کدبندی می‌شوند. در یک ساختار PKIData یا PKIResponse شناسانه بدنه پیام باید منحصر به فرد باشد. این شناسانه می‌تواند در لایه‌های مختلف قرار داده شود (به طور مثال یک PKIData درون ساختار PKIData دیگر).

مقدار صفر bodyPartID برای استفاده به عنوان ارجاع به PKIData فعلی ذخیره شده است.

برخی کنترل‌ها مانند کنترل افزودن الحاقیات(طبق زیربند الف-۵-۲) از شناسانه بدنه پیام در فیلد pkiDataReference برای ارجاع به یک درخواست PKI در ساختار PKIData فعلی استفاده می‌کنند. برخی کنترل‌ها نیز مانند کنترل اطلاعات وضعیت CMC توسعه یافته[1] (طبق زیربند الف-۱-۱) از شناسانه بدنه برای ارجاع به عناصری در درخواست/پاسخ PKI قبلی بهره می‌گیرند. این امر بیانگر یک خطای کنترلی و یا یک خطا در یک درخواست PKI می‌باشد.

فهرستی از اجزای بدنه پیام در یک درخواست/پاسخ PKI در BodyPartList قرار می‌گیرد. نگارش ASN.1 این فیلد به صورت زیر است:

```
BodyPartList ::= SEQUENCE SIZE (1..MAX) OF BodyPartID
```

مسیری از شناسانه‌های بدنه پیام که به صورت تودرتو قرار می‌گیرند در BodyPartPath واقع می‌شود (یعنی، کنترل Modify Certification Request طبق زیربند الف-۵-۱). تعریف ASN.1 این فیلد به صورت زیر است:

```
BodyPartPath ::= SEQUENCE SIZE (1..MAX) OF BodyPartID
```

۸ پاسخ PKI

در این بند به جزییات بیشتری از دو نوع پاسخ PKI پرداخته می‌شود.

۸-۱ پاسخ ساده PKI

کارخواهان باید قادر باشند که پاسخ ساده PKI را پردازش کنند. پاسخ ساده PKI از یک SignedData بدون ساختارهای EncapsulatedContentInfo و SignerInfo تشکیل می‌شود. گواهی‌های درخواست شده در پاسخ PKI در فیلد گواهی SignedData بازگشت داده می‌شوند.

کارخواهان نباید تصور کنند که گواهی‌ها به ترتیب خاصی ارسال شده‌اند. بهتر است کارسازان همه گواهی‌های میانی لازم برای تشکیل یک مسیر کامل گواهی به یک یا چند مرجع اعتماد را قرار دهند. علاوه بر این ممکن است کارسازان CRLها را نیز به کارخواهان ارسال کنند. ممکن است کارسازان گواهی‌های خود امضا را نیز ارائه کنند. کارخواهان نباید به گواهی‌های خود امضای همراه، فقط به این دلیل که در فهرست

1 -Extended CMC Status Info

گواهی وجود دارند، به طور ضمنی اعتماد کنند. در زمانی که کارخواهان گواهی‌های خود امضا را از کارساز دریافت می‌کنند، بهتر است که برای کاربران امکانی برای استفاده از گواهی به عنوان یک مرجع اعتماد به وجود بیاورند

۸-۲ پاسخ کامل PKI

کارخواه باید قادر باشد که پاسخ کامل PKI را پردازش کند. پاسخ کامل PKI از یک SignedData یا AuthenticatedData که ساختار داده PKIResponse را در خود جای می‌دهد، تشکیل شده است. گواهی‌های صادر شده در یک پاسخ PKI در فیلد گواهی‌های SignedData که به صورت آنی درون یک ساختار دیگر جای می‌گیرد، قرار داده می‌شود.

کارخواهان نباید تصور کنند که گواهی‌ها به ترتیب خاصی ارسال شده‌اند. بهتر است کارسازان همه گواهی‌های میانی لازم برای تشکیل یک مسیر کامل گواهی به یک یا چند مرجع اعتماد را قرار دهند، تا اینکه فقط گواهی‌های جدید صادر شده را دربرداشته باشند. علاوه بر این ممکن است کارسازان CRLها را نیز به کارخواهان ارسال کنند. ممکن است کارسازان گواهی‌های خود امضا را نیز ارائه کنند. کارخواهان نباید به گواهی‌های خود امضای همراه، فقط به این دلیل که در فهرست گواهی وجود دارند، به طور ضمنی اعتماد کنند. در زمانی که کارخواهان گواهی‌های خود امضا را از کارساز دریافت می‌کنند، بهتر است که برای کاربران امکانی برای استفاده از گواهی به عنوان یک مرجع اعتماد به وجود بیاورند.

۸-۲-۱ ساختار داده PKIResponse

این ساختار داده در پاسخ کامل PKI استفاده می‌شودو به صورت زیر تعریف می‌شود:

```
id-cct-PKIResponse ::= {id-pkix id-cct(12) 3 }
```

ساختار ASN.1 این ساختار داده به صورت زیر می‌باشد:

```
PKIResponse ::= SEQUENCE {
        controlSequence                 SEQUENCE SIZE(0..MAX) OF
    TaggedAttribute,
        cmsSequence             SEQUENCE SIZE(0..MAX) OF
    TaggedContentInfo,
        otherMsgSequence        SEQUENCE SIZE(0..MAX) OF OtherMsg
        }
ReponseBody ::= PKIResponse
```

یادآوری- در RFC 2797 نوع ASN.1 فوق ResponseBody نامیده می‌شود. در این استاندارد به دلایل خوانایی و حفظ معنا، این نوع با PKIResponse مشخص می‌گردد.

فیلدهای PKIResponse معانی زیر را دارا هستند:

- ControlSequence – دنباله‌ای از کنترل‌ها است. در پیوست الف کنترل‌های تعریف شده در این استاندارد وجود دارند. همچنین کنترل‌ها می‌توانند توسط طرف‌های دیگر نیز تعریف شوند. جزییات ساختار TaggedAttribute در زیربند ۷-۲-۱-۱ آورده شده است.
- cmsSequence – یک دنباله از اشیاء پیام CMS است. به زیربند ۷-۲-۱-۳ برای جزییات بیشتر مراجعه شود.
- otherMsgSequence - دنباله‌ای از اشیاء اختیاری داده است. اشیاء داده‌ای که در این‌جا قرار می‌گیرد توسط یک یا چند کنترل مورد ارجاع قرار می‌گیرند. این امر منجر می‌شود که کنترل‌ها از حجم داده زیادی بدون اینکه داده در کنترل‌ها ادغام شود، استفاده کنند. به زیربند ۷-۲-۱-۴ برای جزییات بیشتر مراجعه شود.

پردازش PKIResponse باید همراه با بررسی و پردازش همه کنترل‌ها باشد. روش مناسب پردازش به صورت اتمام پردازش در لحظه ورود، نادیده گرفتن کنترل‌ها یا قراردادن کنترل‌ها در فهرست کارهای دردست اقدام برای پردازش‌های آتی است. پردازش‌های بیشتر برای cmsSequence یا otherMsgSequence اعضای PKIResponse درصورتی که همه قسمت‌ها وجود داشته باشند و کنترلی به آن‌ها ارجاع نداده باشد، لازم نیست. در این حالت این فیلدها در نظر گرفته نمی‌شوند.

CMC

در CMC کنترل‌ها قسمتی از پیام‌های درخواست و پاسخ کامل PKI هستند. هر کنترل به صورت یک OID یکتا و یک بخش داده، کدبندی می‌شود (به ساختار کنترل‌ها زیربند ۷-۲-۱-۱ مراجعه شود). نوع کدبندی بخش داده هر کنترل منوط به خود آن کنترل است. سامانه‌های پردازشگر، در ابتدا OID (TaggedAttribute attrType) را تشخیص داده و سپس کنترل متناظر با آن (TaggedAttribute attrValues) را قبل از بدنه پیام پردازش می‌کنند.

تعریف OID کنترل‌ها مطابق زیر در بین دو کمان قرار می‌گیرد:

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

در جدول زیر نام، OID، و ساختار کنترل‌هایی که در این استاندارد به آنها اشاره شده است، فهرست می‌گردد:

جدول ۱ – شناسانه کنترل های CMC

شرح شناسانه	شناسانه	ساختار ASN.1
id-cmc-statusInfo	id-cmc 1	CMCStatusInfo
id-cmc-identification	id-cmc 2	UTF8String
id-cmc-identityProof	id-cmc 3	OCTET STRING
id-cmc-dataReturn	id-cmc 4	OCTET STRING
id-cmc-transactionId	id-cmc 5	INTEGER
id-cmc-senderNonce	id-cmc 6	OCTET STRING
id-cmc-recipientNonce	id-cmc 7	OCTET STRING
id-cmc-addExtensions	id-cmc 8	AddExtensions
id-cmc-encryptedPOP	id-cmc 9	EncryptedPOP
id-cmc-decryptedPOP	id-cmc 10	DecryptedPOP
id-cmc-lraPOPWitness	id-cmc 11	LraPOPWitness
id-cmc-getCert	id-cmc 15	GetCert
id-cmc-getCRL	id-cmc 16	GetCRL
id-cmc-revokeRequest	id-cmc 17	RevokeRequest
id-cmc-regInfo	id-cmc 18	OCTET STRING
id-cmc-responseInfo	id-cmc 19	OCTET STRING
id-cmc-queryPending	id-cmc 21	OCTET STRING
id-cmc-popLinkRandom	id-cmc 22	OCTET STRING
id-cmc-popLinkWitness	id-cmc 23	OCTET STRING
id-cmc-popLinkWitnessV2	id-cmc 33	OCTET STRING
id-cmc-confirmCertAcceptance	id-cmc 24	CMCCertId
id-cmc-statusInfoV2	id-cmc 25	CMCStatusInfoV2
id-cmc-trustedAnchors	id-cmc 26	PublishTrustAnchors
id-cmc-authData	id-cmc 27	AuthPublish
id-cmc-batchRequests	id-cmc 28	BodyPartList
id-cmc-batchResponses	id-cmc 29	BodyPartList
id-cmc-publishCert	id-cmc 30	CMCPublicationInfo
id-cmc-modCertTemplate	id-cmc 31	ModCertTemplate
id-cmc-controlProcessed	id-cmc 32	ControlsProcessed
id-cmc-identityProofV2	id-cmc 34	IdentityProofV2

الف-۱ کنترل‌های CMC Status Info

این کنترل‌ها اطلاعاتی در مورد وضعیت درخواست/پاسخ کارخواه و کارساز در اختیار قرار می‌دهند. دو نوع از این کنترل‌ها در این بخش بیان می‌شود.

ممکن است که کارسازان چند کنترل CMC Status Info را در یک بخش بدنه پیام قرار دهند. کارخواهان باید قادر باشند که به چندین کنترل از این دست در یک پاسخ PKI پاسخگو باشند.کارخواهان باید از کنترل CMC Status Info توسعه یافته (CMCStatusInfoV2) استفاده کنند ولی ممکن است که علاوه بر آن از CMC Status Info نیز استفاده کنند. کارخواهان نیز باید قادر به پشتیبانی از کنترل CMC Status Info توسعه یافته باشند.

الف-۱-۱ کنترل CMC Status Info توسعه یافته

این نوع کنترل با OID زیر شناخته می‌شود:

```
id-cmc-statusInfoV2 ::= { id-cmc 25 }
```

تعریف ساختار ASN.1 آن نیز به صورت زیر است:

```
CMCStatusInfoV2 ::= SEQUENCE {
        cMCStatus               CMCStatus,
        bodyList                SEQUENCE SIZE (1..MAX) OF
    BodyPartReference,
        statusString            UTF8String          OPTIONAL,
        otherInfo               OtherStatusInfo     OPTIONAL
        }

OtherStatusInfo ::= CHOICE {
        failInfo                    CMCFailInfo,
        pendInfo                PendInfo,
        extendedFailInfo            ExtendedFailInfo
        }

PendInfo ::= SEQUENCE {
        pendToken       OCTET STRING,
        pendTime        GeneralizedTime
        }

ExtendedFailInfo ::= SEQUENCE {
        failInfoOID     OBJECT IDENTIFIER,
        failInfoValue   ANY DEFINED BY failInfoOID
        }

BodyPartReference ::= CHOICE {
        bodyPartID      BodyPartID,
        bodyPartPath    BodyPartPath
        }
```

تعریف فیلدهای کنترل CMCStatusInfoV2 به صورت زیر است:

- cMCStatus – دربردارنده مقدار وضعیت بازگشت داده شده است. جزییات مربوط به آن در بخش پیوست الف-۱-۳ تعریف می‌شود.
- bodyList – فهرست کنترل‌ها یا دیگر عناصری که در آنها مقدار وضعیت کاربرد دارد را مشخص می‌کند. اگر خطایی برای یک درخواست ساده PKI بازگشت داده شود، این فیلد مقدار صحیح ۱ را در bodyPartID از ساختار BodyPartReference خواهد داشت.
- statusString – دربرگیرنده اطلاعات توصیفی اضافی است. مقادیر این فیلد برای انسان قابل خواندن است.
- otherInfo – دربرگیرنده اطلاعات اضافی در مورد کد وضعیت CMC است که توسط فیلد cMCStatus بازگشت داده می‌شود.

تعریف فیلدهای OtherStatusInfo به صورت زیر است:

- failInfo – در زیربند الف-۱-۴ این فیلد شرح داده خواهد شد. در این فیلد کد خطایی که جزییات وقوع یک خطا را شرح می‌دهد قرار می‌گیرد. تنها درصورتی که cMCStatus دربردارنده مقدار خطا باشد، این گزینه وجود خواهد داشت.
- pendInfo – دربردارنده اطلاعاتی است در مورد اینکه چه هنگام و چطور کارخواهان باید نتایج درخواست را تقاضا کند. درصورتی که cMCStatus دارای مقادیر pending ۱ یا partial ۲ باشد از این فیلد استفاده می‌شود. فیلد pendInfo از ساختار PendInfo استفاده می‌کند که دارای فیلدهای زیر است:
  - pendToken – توکن[3] مورد استفاده در کنترل Query Pending است (طبق زیربند الف-۱۰).
  - pendTime – مدت زمان پیشنهادی است که کارساز لازم دارد تا بتواند به سوال درباره وضعیت درخواست صدور گواهی پاسخ دهد.
- extendedFailInfo – دربردارنده اطلاعات جزیی خطای برنامه‌های کاربردی است. فقط اگر cMCStatus شامل مقدار خطا باشد این قلم وجود خواهد داشت. در هنگام تعریف مقادیر جدید باید دقت شود، زیرا ممکن است این مقادیر توسط همه کارسازان و کارخواهان به صورت درستی قابل تشخیص نباشد. درصورت عدم توانایی در تشخیص خطا، مقدار CMCFailInfo برابر با internalCAError خواهد بود. این فیلد از ساختار ExtendedFailInfo استفاده می‌کند که دارای فیلدهای زیر است:
  - failInfoOID – دربردارنده OID است که به مجموعه‌ای از مقادیر خطاهای توسعه یافته تخصیص داده شده است.
  - failInfoValue- دربردارنده یک کد خطا از مجموعه کدهای خطای توسعه یافته است.

---

۱ - pending به حالتی گفته می‌شود که درخواست گواهی به صورت معوق در آمده باشد.
۲ - partial به حالتی گفته می‌شود که درخواست گواهی در حال پردازش‌های تکمیلی است و هنوز ناتمام است.
3 - Token

اگر فیلد cMCStatus دارای مقدار success[1] باشد، کنترل CMC Status Info توسعه یافته، ممکن است حذف گردد، مگر این که این کنترل تنها مورد موجود در پاسخ باشد.

الف-۱-۲ کنترل CMC Status Info

این کنترل دارای OID به صورت زیر است:

```
id-cmc-statusInfo ::= { id-cmc 1 }
```

ساختار ASN.1 این کنترل نیز به فرم زیر تعریف شده است:

```
CMCStatusInfo ::= SEQUENCE {
        cMCStatus   CMCStatus,
        bodyList    BodyPartList,
        statusString  UTF8String OPTIONAL,
        otherInfo   CHOICE {
                failInfo                CMCFailInfo,
                pendInfo        PendInfo } OPTIONAL
}
```

فیلدهای ساختار CMCStatusInfo به صورت زیر است:

- cMCStatus - دربردارنده مقدار وضعیت بازگشت داده شده است. جزییات مربوط به آن در بخش پیوست الف-۱-۳ تعریف می‌شود.
- bodyList - فهرست کنترل‌ها یا دیگر عناصری که در آنها مقدار وضعیت کاربرد دارد را مشخص می‌کند. اگر خطایی برای یک درخواست ساده PKI بازگشت داده شود، این فیلد مقدار صحیح ۱ را می‌گیرد.
- statusString - دربرگیرنده اطلاعات توصیفی اضافی است. مقادیر این فیلد برای انسان قابل خواندن است.
- otherInfo - دربرگیرنده اطلاعات اضافی در مورد کد وضعیت CMC است که توسط فیلد cMCStatus بازگشت می‌شود.
  - failInfo - در زیربند الف -۱-۴ این فیلد شرح داده خواهد شد. در این فیلد کد خطایی که جزییات وقوع یک خطا را شرح می‌دهد، قرار می‌گیرد. تنها درصورتی که cMCStatus دربردارنده مقدار خطا باشد، این گزینه وجود خواهد داشت.
  - pendInfo – از ساختار ASN.1 فیلد PendInfo در زیربند الف-۱-۱ استفاده می‌کند. این فیلد دربردارنده اطلاعاتی در مورد زمان و نحوه تقاضای کارخواه به منظور دریافت نتایج درخواست خود است. این فیلد در cMCStatus باید برابر با مقادیر pending یا partial باشد. برای جزییات بیشتر

---

۱ - success وضعیتی است که درخواست گواهی با موفقیت پردازش شده است.

به زیربند الف-۱-۱ (کنترل CMC Status Info توسعه یافته) و زیربند الف-۱۰ (کنترل Query Pending) مراجعه شود.

اگر فیلد cMCStatus دارای مقدار success باشد، کنترل CMC Status Info می‌تواند حذف گردد مگر اینکه تنها آیتم در پاسخ باشد. اگر وضعیتی برای درخواست کامل یا ساده PKI وجود نداشته باشد، فرض بر مقدار success است.

الف-۱-۳ مقدارهای CMCStatus

فیلد CMCStatus در کنترل‌های Extended CMC Status Info و CMC Status Info وجود دارد. این فیلد در بردارنده کدی است که بیانگر موفقیت یا عدم موفقیت یک عملیات خاص است. ساختار ASN.1 آن در زیر آمده است:

```
CMCStatus ::= INTEGER {
        success                 (0),
        -- reserved    (1),
        failed                  (2),
        pending                 (3),
        noSupport    (4),
        confirmRequired         (5),
        popRequired             (6),
        partial                 (7)
}
```

تعریف مقدارهای CMCStatus به صورت زیر است:

- success – بیان می‌کند که درخواست تولید شده یا فعالیت کامل شده است.
- failed – بیان می‌کند که درخواست تولید نشده یا فعالیت کامل نشده است. اطلاعات بیشتر در این خصوص در جایی دیگر در پاسخ قرار می‌گیرد.
- pending – بیان می‌کند که درخواست PKI هنوز در مرحله پردازش است. درخواست کننده مسؤول پیگیری[۱] در مورد درخواست کامل PKI است. این مقدار تنها می‌تواند برای عملیات درخواست صدور گواهی بازگشت داده شود.
- noSupport – بیانگر این است که از عملیات مورد تقاضا پشتیبانی نمی‌شود.
- confirmRequired – بیان می‌کند که قبل از آن که بتوان از گواهی استفاده کرد، باید کنترل Confirm Certificate Acceptance (طبق زیربند الف-۱۱) بازگشت داده شود.
- popRequired – بیان می‌کند که عملیات POP مستقیم لازم است. (طبق زیربند الف-۳-۱-۳)

---

۱ - این پیگیری به روش polling است، یعنی طبق یک بازه زمانی مشخص، درخواست‌کننده درخواست خود را پیگیری می‌کند.

- partial – بیان می‌کند که یک پاسخ PKI ناتمام بازگشت داده شده است. درخواست‌کننده مسؤول پیگیری بخش‌های تکمیل نشده درخواست کامل PKI است.

الف-۱-۴ CMCFailInfo

این فیلد در کنترل‌های CMC Status Info توسعه یافته و CMC Status Info موجود است. این فیلد اطلاعات بیشتری در مورد تفسیر شرایط مردود شدن درخواست ارائه می‌کند. ساختار ASN.1 آن به صورت زیر است:

```
CMCFailInfo ::= INTEGER {
        badAlg              (0),
        badMessageCheck     (1),
        badRequest          (2),
        badTime             (3),
        badCertId           (4),
        unsupportedExt              (5),
        mustArchiveKeys     (6),
        badIdentity         (7),
        popRequired                 (8),
        popFailed           (9),
        noKeyReuse                  (10),
        internalCAError             (11),
        tryLater                    (12),
        authDataFail                (13)
```

مقدارهای CMCFailInfo به صورت زیر تعریف شده‌اند:

- badAlg – بیانگر یک الگوریتم ناشناخته یا پشتیبانی نشده است.
- badMessageCheck – بیانگر عدم موفقیت در بررسی صحت است.
- badRequest – بیان می‌کند که تراکنش غیرمجاز یا پشتیبانی نشده است.
- badTime – بیان می‌کند که فیلد زمان پیام به اندازه کافی به زمان سامانه نزدیک نیست.
- badCertId – بیان می‌کند که هیچ گواهی با معیارهای فراهم شده، سازگار نمی‌باشد.
- unsupportedExt – بیان می‌کند که یک الحاقیه X.509 درخواست شده، توسط CA گیرنده پشتیبانی نمی‌شود.
- mustArchiveKey – بیان می‌کند که عناصر کلید خصوصی باید تهیه گردند.
- badIdentity – بیان می‌کند که کنترل تشخیص هویت در ارزیابی مردود شده است.
- popRequired – بیان می‌کند که کارساز به یک عملیات POP قبل از صدور گواهی نیاز دارد.
- popFailed – بیان می‌کند که پردازش عملیات POP موفقیت‌آمیز نبوده است.

- noKeyReuse – بیان می‌کند که خط‌مشی کارساز استفاده مجدد از کلید را مجاز نمی‌داند.
- internalCAError – بیان می‌کند که CA دارای خطای داخلی ناشناخته است.
- tryLater – بیان می‌کند که کارساز نمی‌تواند درخواست را در این لحظه دریافت کند و کارخواه باید در زمان دیگری سعی به ارسال درخواست کند.
- authDataFail – بیان می‌کند که درهنگام پردازش اطلاعات احرازهویت، اشکالی به وجود آمده است.

درصورتی که به دلایل بیشتری برای رخداد خطا نیاز باشد، بهتر است از بخش ExtendedFailureInfo در کنترل CMC Status Infoتوسعه یافته، استفاده شود. محدوده کدهای خطایی علاوه بر آنچه در بالا اشاره شد، باید بین ۱۰۰۰ تا ۱۹۹۹ درنظر گرفته شود.

الف-۲ کنترل‌های Identity Proof و Identification

برخی از مراکز CA و RA الزام دارند تا عملیات اثبات هویت در یک درخواست صدور گواهی قرار داشته باشد. روش‌های زیادی برای انجام این امر با درجه‌های مختلفی از امنیت و قابلیت اطمینان وجود دارد. روشی برای اثبات هویت کارخواه بر اساس راز به اشتراک گذاشته شده بین کارساز/ کارخواه توسط CMC فراهم می‌شود. اگر کارخواهان از درخواست کامل PKI پشتیبانی کنند، آنگاه کارخواهان باید این روش اثبات هویت را به کار ببرند (طبق زیربند الف-۲-۲). کارسازان باید این روش را فراهم کنند، اما ممکن است علاوه بر آن از روش‌های دوطرفه نیز، استفاده کنند.

این استاندارد کنترل Identification را نیز فراهم می‌کند (طبق زیربند الف-۲-۳). این کنترل روش ساده‌ای است که به کارخواه اجازه می‌دهد تا هویت خود را به کارساز اعلام کند. به طور کلی، راز به اشتراک گذاشته شده و شناسانه‌ای از آن، از کارساز به کارخواه انتقال داده می‌شود. شناسانه در کنترل Identification قرار می‌گیرد و راز به اشتراک گذاشته شده برای محاسبه کنترل Identity Proof استفاده می‌شود.

الف-۲-۱ کنترل Identity Proof Version 2

OID این کنترل به صورت زیر است:

```
id-cmc-identityProofV2 ::= { id-cmc 34 }
```

ساختار ASN.1 آن نیز به صورت زیر تعریف شده است:

```
IdentifyProofV2 ::= SEQUENCE {
        hashAlgID               AlgorithmIdentifier,
        macAlgID                AlgorithmIdentifier,
        witness                         OCTET STRING
```

تعبیر فیلدهای IdentifyProofV2 به شرح زیر است:

- hashAlgID – شناسانه و پارامترهای الگوریتم درهم‌ساز مورد استفاده برای تبدیل راز به اشتراک گذاشته شده به یک کلید برای الگوریتم MAC است.
- macAlgID – شناسانه و پارامترهایی برای الگوریتم کد احراز اصالت‌سنجی پیام (MAC) است که برای محاسبه مقدار فیلد witness به کار می‌رود.
- witness – مقدار اثبات هویت را شامل می‌شود.

روش لازم با انتقال توکن (راز مشترک) از خارج از سامانه آغاز می‌شود. راز به اشتراک گذاشته شده باید به صورت تصادفی تولید شود. توزیع توکن از حوزه این استاندارد خارج است. آنگاه کارخواه از این توکن برای اثبات هویت به صورت زیر استفاده می‌کند:

۱- فیلد reqSequence از ساختار PKIData (به طور دقیق همانند هنگامی که در درخواست کامل PKI شامل طول و نوع دنباله قرار می‌گیرد، کدبندی می‌شود)، مقداری است که باید اعتبارسنجی شود.

۲- با استفاده از hashAlgID ، از راز مشترکی که با کدبندی UTF-8 است، یک hash محاسبه می‌شود.

۳- یک MAC نیز با استفاده از مقدار تولید شده در گام اول به عنوان پیام و مقدار حاصل از گام دوم به عنوان کلید محاسبه می‌شود.

۴- سپس نتیجه گام سوم به عنوان مقدار فیلد witness در کنترل Identity Proof Version 2 کدبندی می‌شود.

وقتی که کارساز کنترل Identity Proof Version 2 را ارزیابی می‌کند، مقدار MAC را با همین روش محاسبه و با مقدار witness در درخواست PKI مقایسه می‌کند.

اگر کارسازی ارزیابی کنترل Identity Proof Version 2 را مردود گرداند، مقدار CMCFailInfo باید در پاسخ کامل PKI موجود بوده و برابر با مقدار badIdentity باشد. استفاده مجدد از راز به اشتراک گذاشته شده در درخواست‌های گواهی تکراری به کارخواه و کارساز این امکان را می‌دهد که تصمیم مشابهی برای مقدارهای اثبات هویت قابل قبول بگیرند. هرچند استفاده مجدد از راز به اشتراک گذاشته شده می‌تواند به طور بالقوه موجب وقوع برخی از انواع حملات گردد.

پیاده‌سازی‌ها باید قادر به پشتیبانی از توکن‌هایی با طول حداقل ۱۶ نویسه باشند. در پیوست الف از NIST SP 800-63 راهنماهایی در مورد میزان آنتروپی که در واقع از یک توکن با طول معلوم براساس مجموعه نویسه حاصل می‌شود، آورده شده است.

الف-۲-۲ کنترل Identity Proof

OID این کنترل به صورت زیر است:

id-cmc-identityProof ::= { id-cmc 3 }

ساختار ASN.1 آن نیز در زیر چنین تعریف شده است:

IdentifyProof ::= OCTET STRING

این کنترل نیز به همان روشی پردازش می‌شود که کنترل Identity Proof Version 2 پردازش می‌گردد. در این مورد، الگوریتم درهم‌سازی، SHA-1 و الگوریتم MAC ، HMAC-SHA1 درنظر گرفته شده‌اند.

الف-۲-۳ کنترل Identification

کارسازان می‌توانند به انتخاب خود کنترل Identification حفاظت نشده را در یک کنترل Identification Proof قرار دهند. هدف از کنترل Identification قرار دادن یک رشته متنی است که به کارساز در یافتن مکان راز به اشتراک گذاشته شده، مورد نیاز برای ارزیابی محتوای کنترل Identity Proof کمک می‌کند. درصورتی که کنترل Identification در درخواست کامل PKI قرار داده شده باشد، استخراج کلید در گام دوم (طبق زیربند الف-۲-۱) به این صورت است که به جای تنها راز مشترک، مقدار hash حاصل از الحاق راز به اشتراک گذاشته شده و مقدار هویت به شکل UTF8 (بدون بایت‌های طول و نوع) درهم‌سازی می‌گردند.

OID کنترل Identification به صورت زیر تعریف می‌شود:

id-cmc-identification ::= { id-cmc 2 }

ساختار ASN.1 این کنترل در زیر تعریف شده است:

Identification ::= UTF8String

الف-۲-۴ تولید توکن راز به اشتراک گذاشته شده به صورت سخت‌افزاری

برخی اوقات راز به اشتراک گذاشته شده بین EE و کارساز با استفاده از یک افزاره سخت‌افزاری که مجموعه‌ای از توکن‌ها را تولید می‌کند، محاسبه می‌گردد. بنابراین، EE می‌تواند هویت خود را با انتقال این توکن به صورت متن آشکار به همراه یک رشته نام[1] اثبات کند. این پروتکل می‌تواند با یک افزاره تولید توکن راز به اشتراک گذاشته شده سخت‌افزاری با تغییرات زیر استفاده گردد:

۱- کنترل Identification باید در این فرآیند قرار گیرد و باید حاوی توکن‌های تولید شده به صورت سخت‌افزاری باشد.

۲- راز به اشتراک گذاشته شده مورد استفاده در فوق همان مقدار توکن تولید شده به صورت سخت‌افزاری را دارد.

۳- کلیه درخواست‌های گواهی باید دارای یک نام مالک گواهی باشد. نام مالک گواهی باید دارای فیلدهای مورد نیاز برای شناسایی دارنده افزاره توکن سخت‌افزاری باشد.

1 - Name string

۴- درخواست صدور گواهی باید برای جلوگیری از استراق‌سمع به طور کامل پوشیده شود. هر چند که توکن حساس به زمان است، اما استراق‌سمع کننده فعال، نباید مجاز به استخراج توکن و ثبت یک درخواست صدور گواهی دیگر با همان توکن باشد.

الف-۳ پیوند اطلاعات هویتی و POP

در یک درخواست کامل PKI، اطلاعات هویتی کارخواه در امضای SignedData که حاوی کلیه درخواست‌های گواهی است، انتقال می‌یابد. اما اطلاعات اثبات مالکیت زوج کلیدها به طور مستقل از هر درخواست صدور گواهی CRMF یا PKCS #10 حمل می‌گردد. (برای کلیدهایی که قادر به تولید امضای رقمی هستند، POP به واسطه امضا روی درخواست PKCS #10 یا CRMF فراهم می‌شود. در مورد کلیدهایی که فقط در رمزبندی کاربرد دارند، کنترل‌های زیربند الف-۷ به کار می‌روند.) در نتیجه برای جلوگیری از حملاتی به روش جانشینی[1]، پروتکل باید تضمین کند که یک هستار اطلاعات POP و اثبات هویت را تولید کرده است.

در این زیربند دو ساز و کار برای پیوند اطلاعات هویتی و POP توصیف می‌گردد: مقدارهای فیلد witness (گواه) که به روش رمزنگاشتی از راز به اشتراک گذاشته شده استخراج می‌گردند (طبق زیربند الف-۳-۱-۳) و انطباق شناسه منحصر به فرد (DN) مالک گواهی / راز به اشتراک گذاشته شده (طبق زیربند الف-۳-۲) . کارخواهان و کارسازان باید از روش مقدار گواه[2] پشتیبانی کنند. کارخواهان و کارسازان می‌توانند از انطباق راز به اشتراک گذاشته شده / شناسه منحصر به فرد مالک گواهی یا روش‌های دوجانبه پشتیبانی کنند. راهکار پیشنهادی در این دو روش، مجبور کردن کارخواهان به امضای برخی از داده‌های درون هر درخواست صدور گواهی است که می‌تواند به طور مستقیم به راز به اشتراک گذاشته شده، اختصاص یابد؛ این موضوع کوشش‌هایی که برای پروتکل درخواست‌های گواهی از هستارهای مختلف در یک درخواست کامل PKI انجام می‌شود را نافرجام می‌گذارد.

الف-۳-۱ پیوند رمزنگاشتی

پیوند رمزنگاشتی[3]، اولین روشی است که اطلاعات هویتی و POP را پیوند می‌زند و کارخواهان را مجبور می‌کند تا بخشی از اطلاعات را که به صورت رمزنگاشتی از راز به اشتراک گذاشته شده و استخراج شده، به عنوان یک الحاقیه امضا شده در هر درخواست صدور گواهی (PKCS #10 یا CRMF) قرار دهند.

الف-۳-۱-۱ کنترل‌های POP Link Witness Version 2

این کنترل دارای OID به صورت زیر است:

1 - Substitution-style attacks
2 - witness
3 - Cryptographic Linkage

id-cmc-popLinkWitnessV2 ::= { id-cmc 33 }

ساختار ASN.1 این کنترل به صورت زیر تعریف شده است:

```
PopLinkWitnessV2 ::= SEQUENCE {
                keyGenAlgorithm AlgorithmIdentifier,
                macAlgorithm            AlgorithmIdentifier,
                witness                 OCTET STRING
```

فیلدهای PopLinkWitnessV2 به صورت زیر تعریف می‌شوند:

- keyGenAlgorithm – در بردارنده الگوریتم مورد استفاده در تولید کلید الگوریتم MAC است. به طور کلی این فیلد شامل یک الگوریتم درهم‌ساز است، ولی می‌تواند الگوریتم پیچیده‌تری نیز باشد.
- macAlgorithm – دربردارنده الگوریتم مورد استفاده برای ایجاد مقدار گواه است.
- witness – مقدار گواه محاسبه شده را شامل می‌شود.

این روش درصورتی که DN های دارنده گواهی با مقدار null مورد استفاده قرار بگیرد مفید است (به این دلیل که برای مثال، کارساز می‌تواند DN دارنده گواهی را برای گواهی فقط بر پایه راز به اشتراک گذاشته شده تولید کند). هنگامی‌که کارخواه راز به اشتراک گذاشته شده را به صورت خارج از رویه از کارساز دریافت کرد، پردازش شروع می‌شود. آنگاه کارخواه مقادیر زیر را محاسبه می‌نماید:

۱- کارخواه یک رشته بایتی تصادفی R که بهتر است حداقل دارای طول ۵۱۲ بایت باشد را تولید می‌کند.

۲- با استفاده از الگوریتم موجود در keyGenAlgorithm از راز به اشتراک گذاشته شده، یک کلید محاسبه می‌کند.

۳- براساس مقدار تصادفی تولید شده در گام اول و با استفاده از کلید تولیدی در گام دوم، مقدار MAC محاسبه می‌گردد.

۴- مقدار تصادفی تولید شده در گام اول، به عنوان مقدار کنترل POP Link Random کدبندی می‌شود. این کنترل باید در درخواست کامل PKI وجود داشته باشد.

۵- مقدار MAC تولید شده در گام سوم در کنترل POP Link Witness یا فیلد witness کنترل POP Link Witness V2 قرار می‌گیرد.

- در CRMF، کنترل POP Link Witness/POP Link Witness V2 در فیلد control ساختار CertRequest قرار می‌گیرد.
- در PKCS #10، کنترل POP Link Witness/POP Link Witness V2 در فیلد attributes ساختار CertificationRequestInfo قرار می‌گیرد.

به محض دریافت درخواست، کارسازان باید بررسی کنند که هر درخواست صدور گواهی حاوی یک رونوشت از کنترل POP Link Witness/POP Link Witness V2 است و اینکه مقدار آن با استفاده از روش فوق از راز به اشتراک گذاشته شده و رشته تصادفی درون کنترل POP Link Random به دست آمده است.

کنترل Identification (طبق زیربند الف-۲-۳) یا DN دارنده گواهی یک درخواست صدور گواهی می‌تواند برای کمک به شناسایی اینکه چه راز به اشتراک گذاشته‌ای، استفاده شده است، به کار می‌رود.

الف-۳-۱-۲ کنترل POP Link Witness

این کنترل دارای OID به صورت زیر است:

id-cmc-popLinkWitness ::= { id-cmc 23 }

ساختار ASN.1 آن نیز به صورت زیر است:

PopLinkWitness ::= OCTET STRING

برای این کنترل، SHA-1 به عنوان الگوریتم تولید کلید استفاده می‌شود. الگوریتم HMAC-SHA1 نیز به عنوان الگوریتم MAC به کار می‌رود.

الف-۳-۱-۳ کنترل POP Link Random

OID این کنترل به صورت زیر تعریف شده است:

id-cmc-popLinkRandom ::= { id-cmc 22 }

ساختار ASN.1 آن نیز به صورت زیر تعریف می‌شود:

PopLinkRandom ::= OCTET STRING

الف-۳-۲ پیوند راز به اشتراک گذاشته شده / DN دارنده گواهی

دومین روش اتصال اطلاعات هویتی وPOP، پیوند دادن یک نام ترکیبی دارنده گواهی مشخص به راز مشترکی است که به صورت خارج از رویه توزیع شده است و نیاز دارد کارخواهانی که از راز به اشتراک گذاشته شده برای اثبات هویت استفاده می‌کنند، عنوان دقیق DN دارنده گواهی را در هر درخواست صدور گواهی قرار دهند. انتظار می‌رود که بسیاری از پیوندهای کارخواه-کارسازی که از اثبات هویت مبتنی بر راز به اشتراک گذاشته شده استفاده می‌کنند، از این سازوکار استفاده کنند (معمول نیست که اطلاعات DN دارنده گواهی از درخواست صدور گواهی حذف گردد).

هنگامی که راز به اشتراک گذاشته شده تولید و به صورت خارج از رویه برای آغاز فرآیند ثبت‌نام، انتقال یافت (طبق زیربند الف-۲) یک DN دارنده گواهی مشخص به راز به اشتراک گذاشته شده، اختصاص یافته و با کارخواه ارتباط برقرار می‌شود. (DN دارنده گواهی تولیدی باید در انطباق با خط‌مشی CA به ازای هر هستار منحصربه‌فرد باشد؛ DN دارنده گواهی null نمی‌تواند استفاده شود. یک تجربه رایج می‌تواند این باشد که مقدار شناسایی به عنوان بخشی از DN دارنده گواهی قرار داده شود). هنگامی که کارخواه درخواست کامل PKI را تولید کرد، باید از این دو بخش اطلاعاتی به صورت زیر استفاده کند:

۱- کارخواه باید DN دارنده گواهی خاصی که همراه با راز به اشتراک گذاشته شده، دریافت کرده است را به عنوان نام دارنده گواهی در هر درخواست صدور گواهی (CRMF/PKCS #10) ، درون درخواست کامل PKI قرار دهد.

۲- کارخواه باید دارای یک کنترل Identity Proof (طبق زیربند الف -۲-۲) یا کنترل Identity Proof Version 2 (طبق زیربند الف-۲-۱) باشد، که از راز به اشتراک گذاشته شده موجود در درخواست کامل PKI به دست آمده است.

کارسازی که این پیام را دریافت می‌کند باید الف- کنترل Identity Proof را بررسی کند، ب- درستی‌سنجی کند که DN دارنده گواهی که در هر درخواست صدور گواهی قرار دارد، با DN همراه راز به اشتراک گذاشته شده انطباق دارد. اگر هریک از این درستی‌سنجی‌ها ناموفق باشد، درخواست صدور گواهی باید مردود گردد.

الف-۳-۳ پیام‌های کلیدگذاری مجدد یا به‌روزرسانی کلید

هنگام درخواست کلیدگذاری مجدد یا تمدید گواهی، پیوند اطلاعات هویتی و POP ساده است. کارخواه، فیلد DN دارنده گواهی برای گواهی امضای حاضر را در فیلد نام دارنده گواهی هر درخواست صدور گواهی که ایجاد می‌شود، رونویسی می‌کند. حال POP برای هر درخواست صدور گواهی این اطلاعات را نیز پوشش می‌دهد. بیرونی‌ترین لایه امضا با استفاده از گواهی امضای فعلی تولید می‌شود که اجازه می‌دهد هویت اصلی به درخواست صدور گواهی تخصیص یابد. از آن جایی که نام در گواهی امضای فعلی و نام در درخواست صدور گواهی انطباق دارد، پیوند لازم حاصل می‌شود.

الف-۴ کنترل Data Return

این کنترل به کارخواهان اجازه می‌دهد که داده‌های اختیاری (به طور معمول نوعی اطلاعات وضعیت داخلی) را به کارساز ارسال کنند و داده‌های بازگشتی را به عنوان بخشی از پاسخ کامل PKI دریافت دارند. داده‌ای که در کنترل Data Return قرار می‌گیرد، برای کارساز واضح نمی‌باشد. همین کنترل برای درخواست و پاسخ کامل PKI استفاده می‌شود. اگر کنترل Data Return در درخواست کامل PKI ظاهر شود، کارساز باید آن را به عنوان بخشی از پاسخ PKI بازگرداند.

درهنگامی‌که اطلاعات درون کنترل Data Return لازم است محرمانه باشد، انتظار می‌رود که کارخواه نوعی رمزبندی را به محتوای آن اعمال کند. جزییات این موضوع خارج از حوزه این استاندارد است.

OID کنترل Data Return به صورت زیر تعریف می‌شود:

PopLinkRandom ::= OCTET STRING

تعریف ASN.1 این کنترل نیز به صورت زیر است:

DataReturn ::= OCTET STRING

یک کارخواه از این کنترل برای قرار دادن یک شناسانه جهت علامت‌گذاری محل اصلی قرارگیری کلید خصوصی استفاده می‌کند. این شناسانه ممکن است شناسانه افزاره سخت‌افزاری حاوی کلید خصوصی باشد.

الف-۵ کنترل‌های تغییر گواهی RA

این کنترل‌ها برای RAها ایجاد شده‌اند تا قادر به تغییر محتوای یک درخواست صدور گواهی باشند. بنا به دلایل مختلف ممکن است تغییرات لازم باشد. این دلایل می‌توانند شامل افزودن الحاقیه‌های گواهی یا تغییر نام مالک گواهی و/یا اسامی جایگزین مالک گواهی باشند.

دو کنترل برای این مقصود وجود دارد. اولین کنترل Modify Certification Request (طبق زیربند الف-۵-۱)، به RA اجازه می‌دهد که هر فیلدی درون گواهی را تغییر دهد یا حذف کند. دومین کنترل، Add Extensions (طبق زیربند الف-۵-۲) فقط اجازه افزودن الحاقیه‌ها را می‌دهد.

الف-۵-۱ کنترل Modify Certification Request

این کنترل توسط RAها برای تغییر فیلدهای در یک گواهی درخواست شده استفاده می‌شود.
OID این کنترل به صورت زیر است:

id-cmc-modCertTemplate ::= { id-cmc 31 }

ساختار ASN.1 این کنترل نیز به صورت زیر تعریف شده است:

```
ModCertTemplate ::= SEQUENCE {
        pkiDataReference        BodyPartPath,
        certReferences                  BodyPartList,
        replace                         BOOLEAN DEFAULT TRUE,
        certTemplate                    CertTemplate
```

تعریف فیلدهای ModCertTemplate در زیر آمده است:

- pkiDataReference – مسیری از درخواست PKI است که شامل درخواست(های) گواهی می‌باشد که قرار است تغییر یابند.
- certReferences – مربوط است به یک یا چند درخواست صدور گواهی در درخواست PKI که توسط pkiDataReference برای تغییر، مشخص می‌گردند. هر BodyPartID دنباله certReferences باید

معادل با bodyPartID ساختار TaggedCertificationRequest (در PKCS #10) یا certReqId ساختار CertRequest موجود در یک CertReqMsg (در CRMF) باشد. الحاقیه‌های درون فیلد certTemplate در هر درخواست صدور گواهی ارجاع داده شده در دنباله certReferences به کار رفته‌اند. اگر یک درخواست مربوط به bodyPartID یافت نشود، CMCFailInfo با مقدار badRequest که به این کنترل ارجاع دارد، بازگشت داده می‌شود.

- replace – تغییر درخواست صدور گواهی هدف، به واسطه جایگزینی یا حذف فیلدها را مشخص می‌کند. اگر مقدار آن TRUE باشد، داده درون این کنترل با داده درون درخواست صدور گواهی هدف جایگزین می‌شود. اگر مقدار این فیلد FALSE باشد، فیلد درون درخواست صدور گواهی هدف، حذف می‌شود. برای فیلدهای الحاقی certTemplate عملکرد کمی متفاوت است. با هر الحاقیه به صورت اختصاصی رفتار می‌شود.
- certTemplate - یک الگوی گواهی است. اگر این فیلد وجود داشته باشد و replace دارای مقدار TRUE باشد این فیلد جایگزین آن در درخواست صدور گواهی می‌شود. اگر این فیلد وجود داشته باشد و replace دارای مقدار FALSE باشد فیلد موجود در درخواست صدور گواهی حذف می‌شود. اگر این فیلد وجود نداشته باشد، هیچ عملیاتی انجام نمی‌شود. هر الحاقیه به عنوان یک فیلد مجزا پردازش می‌شود.

کارسازان باید قادر به پردازش کلیه الحاقیه‌های تعریف شده باشند. البته این موضوع درخصوص مواردی که در مستند جامع پروفایل‌های زیرساخت کلید عمومی کشور منع شده‌اند، صادق نیست. نیازی نیست که کارسازان قادر به پردازش هر الحاقیه X.509v3 انتقال یافته، با استفاده از این پروتکل باشند، همچنین پردازش الحاقیه‌های خصوصی لازم نیست. ضرورتی وجود ندارد که کارسازان همه الحاقیه‌های درخواست شده توسط RA را در یک گواهی قرار دهند. کارسازان مجاز به تغییر الحاقیه‌های درخواست شده توسط RA هستند. کارسازان نباید یک الحاقیه را به گونه‌ای تغییر دهند که مفهوم الحاقیه درخواست شده توسط کارخواهان، معکوس گردد. اگر یک درخواست صدور گواهی به موجب ناتوانی در مدیریت یک الحاقیه درخواست شده، مردود گردد و یک پاسخ کامل PKI بازگشت داده شود، کارساز باید فیلد CMCFailInfo را با مقدار unsupportedExt باز گرداند.

اگر یک درخواست صدور گواهی، هدف چند کنترل Modify Certification Request باشد، اقدامات زیر انجام می‌شود:

- اگر کنترل A در لایه‌ای که لایه کنترل B را دربردارد، وجود داشته باشد، کنترل A باید بر کنترل B ارجحیت داشته باشد. به عبارت دیگر کنترل‌ها از لایه‌های داخلی‌تر به سمت لایه‌های بیرونی پردازش می‌شوند.
- اگر کنترل A و کنترل B در یک PKIData (یعنی یک لایه) باشند، ترتیب کاربرد آن‌ها تعریف نشده است.

ترتیب یکسان از کاربرد کنترل‌ها زمانی استفاده می‌شود که یک درخواست صدور گواهی هدف دو کنترل Modify Certification Request و Add Extensions باشد.

الف-۵-۲ کنترل Add Extensions

کنترل Add Extensions به دلیل وجود کنترل Modify Certification Request استفاده نمی‌شود. این کنترل به نحوی جایگزین شده است که فیلدهای موجود در درخواست صدور گواهی علاوه بر الحاقیه‌ها قابل تغییر هستند.

کنترل Add Extensions توسط RA ها برای تعیین الحاقیه‌های اضافه‌ای که در گواهی قرار می‌گیرند، استفاده می‌شود.
OID این کنترل به صورت زیر است:

id-cmc-addExtensions ::= { id-cmc 8 }

ساختار ASN.1 آن نیز در زیر تعریف شده است:

```
AddExtensions ::= SEQUENCE {
        pkiDataReference        BodyPartID,
        certReferences          SEQUENCE OF BodyPartID,
        extensions              SEQUENCE OF Extension
```

فیلدهای AddExtensions به صورت زیر تعریف شده‌اند:

- pkiDataReference – شامل شناسانه بخش بدنه درخواست‌گواهی جاسازی شده[1] است.
- certReferences – فهرستی از ارجاع‌ها به یک یا چند درخواست صدور گواهی درون یک PKIData است. هر شناسانه بدنه دنباله certReferences باید معادل با bodyPartID یک TaggedCertificationRequest (در PKCS#10) یا certReqId ساختار CertRequest درون CertReqMsg (در CRMF) باشد. بنا به تعریف الحاقیه‌های فهرست شده به هر درخواست صدور گواهی که در دنباله certReferences مورد ارجاع قرار می‌گیرد، اعمال می‌شود. اگر یک درخواست صدور گواهی معادل bodyPartID یافت نگردد، CMCFailInfo به همراه یک مقدار badRequest بازگشت داده می‌شود که به این کنترل اشاره دارد.
- extensions – دنباله‌ای از الحاقیه‌ها است که باید به درخواست‌های گواهی مورد ارجاع اعمال شود.

کارسازان باید قادر به پردازش کلیه الحاقیه‌های تعریف شده باشند. البته این موضوع درخصوص مواردی که در سند جامع پروفایل‌های زیرساخت کلید عمومی کشور[2] منع شده‌اند، صادق نیست. نیازی نیست که کارسازان قادر به پردازش هر الحاقیه X.509v3 انتقال یافته با استفاده از این پروتکل باشند، همچنین

1 - Embedded

۲ این سند از طریق تارنماي مرکز دولتي صدور گواهي الکترونيکي ريشه به آدرس www.rca.gov.ir قابل دريافت است.

پردازش الحاقیه‌های خصوصی لازم نیست. ضرورتی وجود ندارد که کارسازان همه الحاقیه‌های درخواست شده توسط RA را در یک گواهی قرار دهند. کارسازان مجاز به تغییر الحاقیه‌های درخواست شده توسط RA هستند. کارسازان نباید یک الحاقیه را به گونه‌ای تغییر دهند که مفهوم الحاقیه درخواست شده توسط کارخواه، معکوس گردد. اگر یک درخواست صدور گواهی به موجب ناتوانی در مدیریت یک الحاقیه درخواست شده مردود گردد و یک پاسخ کامل PKI بازگشت داده شود، کارساز باید فیلد CMCFailInfo را با مقدار unsupportedExt باز گرداند.

اگر چندین کنترل Add Extensions در یک درخواست کامل PKI وجود داشته باشد، رفتار دقیق به خط‌مشی CA واگذار می‌گردد. هرچند توصیه می‌شود که خط‌مشی زیر به کار رود.

۱- درصورتی که تعارضی در یک PKIData وجود دارد، درخواست صدور گواهی با قرار گرفتن مقدار badRequest در CMCFailInfo مردود می‌گردد.

۲- اگر تعارض بین چند PKIData مختلف باشد، بیرونی‌ترین نسخه الحاقیه استفاده می‌شود (RA مجاز به ابطال الحاقیه درخواست شده است).

الف-۶ کنترل Transaction Indentifier و کنترل‌های Sender and Recipient Nonce

تراکنش‌ها به وسیله یک شناسانه تراکنش شناخته و ردگیری می‌گردند. درصورت استفاده از آن‌ها، کارخواهان شناسانه‌های تراکنش را تولید و مقدار آنها را تا زمانی که کارساز با پیام پاسخ کامل PKI پاسخ دهد که تراکنش را تکمیل می‌کند، نگهداری می‌کنند. کارسازان نیز شناسانه‌های تراکنش دریافتی را در پاسخ کامل PKI قرار می دهند.

OID کنترل Transaction Identifier به صورت زیر است:

id-cmc-transactionId ::= { id-cmc 5 }

این کنترل دارای تعریف ASN.1 به صورت زیر می‌باشد:

TransactionId ::= INTEGER

کنترل Transaction Identifier به یک تراکنش مشخص اشاره می‌کند. این کنترل توسط کارخواه و کارساز برای مدیریت وضعیت یک عملیات به کار می‌رود. کارخواهان ممکن است که این کنترل را در یک درخواست قرار دهند. اگر درخواست اصلی یک کنترل Transaction Identifier را دربرداشت، کلیه درخواست‌های و پاسخ‌های بعدی باید شامل همان کنترل باشند.

با استفاده از کنترل‌های Sender & Recipient Nonce از حمله تکرارجلوگیری می‌شود. اگر از تک شمارها استفاده شود، در اولین پیام تراکنش کنترل Recipient Nonce ارسال نمی‌شود، یک کنترل Sender Nonce توسط مبدأ تراکنش در پیام قرار می‌گیرد و برای ارجاعات بعدی نگهداری می‌شود. گیرنده یک

کنترل Sender Nonce این مقدار را به مبدأ بـه عنـوان کنتـرل Recipient Nonce برمـی‌گردانـد و کنتـرل Sender Nonce خود را نیز ارسال می‌کند. به محض دریافت این پاسخ توسط مبدأ، آغازکننده تراکنش مقدار کنترل Recipient Nonce را با مقدار نگهداری شده توسط خودش مقایسه می‌کند. اگر این دو با هم تطـابق داشته باشند، پیام برای پردازش امنیتی بیشتر پذیرفته می‌شود. مقدار دریافتی کنتـرل Sender Nonce نیـز برای قرار گرفتن در پیام بعدی همراه با همان تراکنش حفظ می‌شود.

OID های زیر برای این کنترل‌ها تعریف شده‌اند:

```
id-cmc-senderNonce ::= { id-cmc 6 }
id-cmc-recipientNonce ::= { id-cmc 7 }
```

تعریف ASN.1 کنترل Sender Nonce به صورت زیر است:

```
SenderNonce ::= OCTET STRING
```

تعریف ASN.1 کنترل Recipient Nonce به صورت زیر است:

```
RecipientNonce ::= OCTET STRING
```

کارخواهان می‌توانند یک کنترل Sender Nonce را در درخواست PKI آغازین قـرار دهنـد. اگـر یـک پیـام شامل کنترل Sender Nonce باشد، پاسخ باید شامل مقدار انتقال یافته کنترل دریافت شده قبلـی Sender Nonce ، به عنوان یک کنترل Recipient Nonce و یک مقدار جدید، به عنوان کنترل Sender Nonce خود باشد.

الف-۷ کنترل‌های Encrypted and Decrypted POP

درصورتی که روش دیگری برای اجرای فرآیند POP نباشد، ممکن است لازم باشد، کارسـازان از ایـن روش برای اجرای POP استفاده کنند. اگـر POP لازم بـرای هـر عنصـر درون PKIData وجـود نداشـته باشـد، کارسازان بهتر است همه درخواست‌های گواهی‌های درون یک PKIData را رد کنند.

کارسازان باید اطمینان یابند که هستار درخواست‌کننده گواهی، در واقع مالک مولفه خصوصی متناظر بـا آن زوج کلید باشد. برای کلیدهایی که می‌توانند به عنوان کلیدهای امضـا اسـتفاده گردنـد، امضـای درخواسـت صدور گواهی با کلید خصوصی به عنوان اجرای عملیات POP روی آن زوج کلید به کار گرفته می‌شود. برای کلیدهایی که فقط می‌توانند به منظور عملیات رمزبندی استفاده گردند، POP باید به وسیله اجبـار کـارخواه به رمزگشایی یک مقدار اجرا گردد.

بنا به ضرورت، عملیات POP برای کلیدهایی که فقط برای رمزبندی به کار می‌روند، در یـک مرحلـه انجـام نمی‌شود. لذا چهار گام مشخص به صورت زیر بیان می‌شود:

۱- کارخواه به کارساز در مورد مولفه عمومی یک زوج کلید رمز گذاری جدید اطلاع می‌دهد.

۲- کارساز برای کارخواه یک چالش عملیات POP که با کلید عمومی رمزبندی فعلی رمزبندی شده است، ارسال می‌کند.

۳- کارخواه چالش عملیات POP را با استفاده از کلید خصوصی که متناظر با کلید عمومی موجود است، رمزگشایی کرده و متن اصلی را به کارساز باز می‌گرداند.

۴- کارساز چالش POP رمزگشایی شده را ارزیابی کرده و پردازش درخواست صدور گواهی را ادامه می‌دهد.

این استاندارد دو کنترل متفاوت را تعریف می‌کند. اولین کنترل مربوط به چالش رمزبندی شده ارسالی از کارساز به کاربر در گام دوم است. دومین کنترل نیز در خصوص چالش رمزگشایی شده ارسالی از کارخواه به کارساز در گام سوم می‌باشد.

کنترل Encrypted POP برای ارسال چالش رمزبندی شده از کارساز به کارخواه به عنوان بخشی از PKIResponse استفاده می‌شود. (فرض می‌گردد پیام ارسال شده در گام اولی که در بالا اشاره شد، درخواست کامل PKI است و پاسخ در گام دوم، پاسخ کامل PKI است که دربردارنده یک CMCFailInfo بوده و به وضوح مشخص می‌کند عملیات POP الزام شده است و چالش POP را در کنترل encryptedPOP ارائه می‌کند).

OID کنترل Encrypted POP به صورت زیر تعریف شده است:

```
id-cmc-encryptedPOP ::= { id-cmc 9 }
```

ساختار ASN.1 این کنترل نیز به صورت زیر تعریف می‌گردد:

```
EncryptedPOP ::= SEQUENCE {
        request             TaggedRequest,
        cms                 ContentInfo,
        thePOPAlgID         AlgorithmIdentifier,
        witnessAlgID        AlgorithmIdentifier,
        witness             OCTET STRING
```

OID کنترل Decrypted POP به صورت زیر تعریف شده است:

```
id-cmc-decryptedPOP ::= { id-cmc 10 }
```

ساختار ASN.1 این کنترل نیز به صورت زیر تعریف می‌گردد:

```
DecryptedPOP ::= SEQUENCE {
        bodyPartID      BodyPartID,
        thePOPAlgID     AlgorithmIdentifier,
```

thePOP OCTET STRING

الگوریتم POP رمزشده به صورت زیر عمل می‌کند:

۱- کارساز به صورت تصادفی مقدار اثبات POP را تولید و آن را پیوست درخواست می‌کند.

۲- کارساز کنترل Encrypted POP را با فیلدهای زیر باز می‌گرداند:

- request – درخواست اصلی گواهی است (به این دلیل در این جا قرار می‌گیرد، زیراکه کارخواه نیازی به ذخیره نسخه کپی از آن ندارد)
- cms – یک EnvelopedData، که دارای ساختار داده کپسوله شده id-data و محتوای آن نیز POP Proof Value می‌باشد؛ این مقدار لازم است تا به اندازه کافی طولانی باشد تا کسی نتواند مقدار آن را از طریق hash گواه، معکوس کند. اگر درخواست صدور گواهی یک الحاقیه شناسانه کلید مالک گواهی (SKI) را شامل بود، آن گاه شناسانه گیرنده نیز باید SKI باشد. اگر شکل issuerAndSerialNumber به‌کار رفته بود، IssuerName باید به صورت Null کدبندی شده و SerialNumber نیز در bodyPartID درخواست صدور گواهی بیان گردد.
- thePOPAlgID - الگوریتم مورد استفاده در محاسبه مقدار POP بازگشتی را معلوم می‌کند.
- witnessAlgID – الگوریتم درهم‌سازی مورد استفاده در POP Proof Value برای ایجاد فیلد witness را مشخص می‌کند.
- witness – مقدار درهم‌سازی شده POP Proof Value است.

۳- کارخواه فیلد cms را رمزگشایی کرده تا مقدار POP Proof Value را به دست آورد. کارخواه مقدار درهم‌ساز (POP Proof Value) را با استفاده از witnessAlgID محاسبه و با مقدار فیلد witness مقایسه می‌کند. اگر این دو مقدار برابر نبود یا رمزگشایی موفقیت‌آمیز نبود، کارخواه باید فرآیند ثبت‌نام را متوقف کند. کارخواه فرآیند را با ارسال یک درخواست شامل کنترل CMC Status Info و CMCFailInfo با مقدار POPFailed قطع می‌کند.

۴- کارخواه کنترل Decrypted POP را به عنوان بخشی از یک PKIData جدید ایجاد می‌کند. فیلدهای DecryptedPOP عبارتند از:

- bodyPartID – به درخواست صدور گواهی در PKI Request جدید ارجاع دارد.
- thePOPAlgID – از encrptedPOP رونویسی می‌شود.
- thePOP – حاوی اثبات مالکیت است. این مقدار به واسطه الگوریتم thePOPAlgID با استفاده از POP Proof Value و درخواست، محاسبه می‌شود.

۵- کارساز مقدار thePOP را از مقدار ذخیره‌شده[1] و درخواست، مجدد محاسبه کرده و آن را با مقدار thePOP مقایسه می‌کند. اگر این دو دارای ارزش یکسانی نبودند، کارساز نباید گواهی را صادر کند. کارساز می‌تواند یک چالش جدید را مجدد صادر کند یا به کلی درخواست را مردود کند.

هنگام تعریف الگوریتم‌هایی برای thePOPAlgID و witnessAlgID، در خصوص اطمینان از اینکه نتیجه witnessAlgID یک مقدار مناسب برای ایجاد میانبر در محاسبه با thePOPAlgID نیست، باید دقت به عمل آید. POP Proof Value به عنوان یک مقدار راز در الگوریتم HMAC به کار برده می‌شود و درخواست به عنوان داده استفاده می‌گردد. اگر POP Proof Value بیش از مقدار ۶۴ بایت باشد، تنها اولین ۶۴ بایت POP Proof Value به عنوان مقدار راز استفاده می‌شود.

مسأله اصلی که در مورد الگوریتم فوق وجود دارد، میزان اندازه حالاتی است که یک CA برای ارزیابی مقدار POP بازگشتی باید نگه دارد. موارد زیر یکی از بی‌شمار راه‌های ممکنی است که این مسأله را با کاهش مقدار حالت‌هایی که در CA نگه‌داری شده به یک مقدار (یا یک مجموعه) کوچک از مقادیر حل می‌کند.

۱- کارساز مقدار x را با عنوان seed به صورت تصادفی تولید می‌کند، این مقدار برای کلیه درخواست‌ها ثابت است. (ارزش x به طور عادی براساس یک اصل معمول تغییر می‌یابد و برای مدت کوتاهی تا پس از آن نگهداری می‌شود)

۲- برای درخواست صدور گواهی R، کارساز مقدار y=F(x , R) را محاسبه می‌کند. F می‌تواند برای مثال HMAC-SHA1 (x,R) باشد. به منظور حفظ وضعیت بدون حالت بودن، مهم است که y به طور ثابت تنها با مقدار حالت معلوم x و تابع F قابل محاسبه باشد. سایر ورودی‌ها از ساختار درخواست صدور گواهی وارد می‌گردد. مقدار y نباید با اطلاع از R قابل پیش‌بینی باشد، در نتیجه از یک تابع یک طرفه مانند HMAC-SHA1 استفاده می‌شود.

الف-۸ کنترل RA POP Witness

در فرانامه‌ی درخواست صدور گواهی که RA را درگیر می‌سازد، ممکن است CA اجازه دهد (لازم داشته باشد) که RA پروتکل POP را با هستاری که درخواست صدور گواهی را تولید کرده، اجرا کند. در این صورت، RA نیاز به روشی دارد تا به CA اطلاع دهد که عملیات POP را انجام داده است. کنترل RA POP Witness این موضوع را مشخص می‌کند.
OID این کنترل به صورت زیر است:

id-cmc-lraPOPWitness ::= { id-cmc 11 }

تعریف ASN.1 این کنترل نیز به صورت زیر است:

1 -Cached

```
LraPopWitness ::= SEQUENCE {
        pkiDataBodyid BodyPartID,
        bodyIds         SEQUENCE of BodyPartID
```

فیلدهای LraPopWitness در زیر تعریف شده‌اند:

- pkiDataBodyid – دربرگیرنده شناسانه بدنه TaggedContentInfo درونی است که شامل درخواست کامل PKI کارخواه می‌باشد. اگر درخواست در PKIData فعلی باشد، مقدار این فیلد صفر خواهد بود.
- bodyIds – فهرستی از درخواست‌های گواهی است که آن‌ها را RA به صورت خارج از رویه احراز هویت کرده است.

درصورتی که یک کارساز گواهی RA را برای انجام عملیات ارزیابی POP مجاز نداند، CMCFailInfo با مقدار popFailed را باز می‌گرداند. خود CA نباید برای ارزیابی مجدد POP یک چالش-پاسخ را شروع کند.

الف-۹ کنترل‌های Registration Information و Response Information

کنترل Registration Information به کارخواهان اجازه ارسال اطلاعات اضافه به عنوان بخشی از درخواست کامل PKI را می‌دهد.
OID این کنترل به صورت زیر است:

```
id-cmc-regInfo ::= { id-cmc 18 }
```

ساختار ASN.1 این کنترل به صورت زیر تعریف می‌گردد:

```
RegInfo ::= OCTET STRING
```

اطلاعات RegInfo براساس توافق دوجانبه بین کارخواه و کارساز خواهد بود.

کنترل Response Information به یک کارساز اجازه می‌دهد که اطلاعات بیشتری را به عنوان بخشی از پاسخ کامل PKI بازگرداند.
OID این کنترل به صورت زیر تعریف شده است:

```
id-cmc-responseInfo ::= { id-cmc 19 }
```

تعریف ساختار ASN.1 این کنترل نیز به صورت زیر است:

```
ResponseInfo ::= OCTET STRING
```

اطلاعات فیلد ResponseInfo براساس توافق دوجانبه بین کارخواه و کارساز خواهد بود.

الف-۱۰ کنترل Query Pending

در برخی از محیط‌ها الزامات پردازشی برای اقدامات شخصی یا دیگر بررسی‌های هویتی می‌تواند در بازگرداندن گواهی تاخیر ایجاد کند. کنترل Query Pending به کارخواهان اجازه پرس‌وجو از کارساز در مورد وضعیت درخواست صدور گواهی معلق را می‌دهد. کارساز یک pendToken به عنوان بخشی از کنترل‌های CMC Status Info توسعه یافته و CMC Status Info (در فیلد otherInfo) باز می‌گرداند. کارخواه نسخه‌ای از pendToken را در کنترل Query Pending برای شناساندن درخواست صدور گواهی صحیح به کارساز قرار می‌دهد. کارساز زمانی را برای کارخواه برای پرس‌وجو از وضعیت درخواست صدور گواهی به تعویق افتاده در نظر می‌گیرد.

OID این کنترل در زیر آمده است:

id-cmc-queryPending ::= { id-cmc 21 }

این کنترل دارای ساختای ASN.1 به صورت زیر است:

QueryPending ::= OCTET STRING

درصورتی که کارساز یک CMCStatuInfo با وضعیت‌های ناتمام یا معلق را بازگرداند (یعنی تراکنش هنوز به تعویق افتاده است) فیلد otherInfo ممکن است حذف شود. درصورتی که این فیلد حذف نشده بود، مقدار pendInfo باید مشابه مقدر pendInfo اصلی باشد.

الف-۱۱ کنترل Confirm Certificate Acceptance

مراکز CA لازم دارند تا کارخواهان تاییدی مبتنی بر اینکه گواهی‌های صادر شده برای EE قابل قبول هستند را ارائه کنند. کنترل Confirm Certificate Acceptance برای این منظور استفاده می‌شود. اگر در پاسخ PKI وضعیت CMC Status Info برابر با مقدار confirmRequired باشد، آنگاه کارخواه باید کنترل Confirm Certificate Acceptance را در درخواست کامل PKI ارسال کند.

کارخواهان باید قبل از استفاده از گواهی به هر منظوری، منتظر پاسخ PKI مبتنی بر دریافت تایید از طرف کارساز باشند.

OID کنترل Confirm Certificate Acceptance به صورت زیر است:

id-cmc-confirmCertAcceptance ::= { id-cmc 24 }

ساختار ASN.1 این کنترل نیز به صورت زیر تعریف شده است:

CMCCertId ::= IssuerAndSerialNumber

CMCCertId شامل صادرکننده و شماره ردیف گواهی پذیرفته شده است.

کارسازان باید یک پاسخ کامل PKI برای کنترل Confirm Certificate Acceptance بازگردانند.

درصورتی که CA این کنترل را قرار دهد، دو رفت و برگشت کامل برای پیام وجود دارد:
۱- کارخواه درخواست صدور گواهی را به CA ارسال کند.

۲- CA یک پاسخ کامل PKI با گواهی و این کنترل باز می‌گرداند.

۳- کارخواه یک درخواست کامل PKI به CA با کنترل CMC Status Info توسعه یافته برای وضعیت پذیرفته شده گواهی و کنترل Confirm Certificate Acceptance را ارسال می‌کند یا این که کنترل Extended CMC Status Info را برای وضعیت رد گواهی ارسال می‌کند.

۴- CA به کارخواه پاسخ کامل PKI را با CMC Status Infoتوسعه یافته، مبنی بر موفقیت‌آمیز بودن ارسال می‌کند.

الف-۱۲ کنترل Authenticated Data

این کنترل به کارساز اجازه می‌دهد که با یک روش احراز هویت شده، داده را به کارخواه ارسال کند. این کنترل از ساختار داده AuthenticatedData برای ارزیابی داده استفاده می‌کند. این کنترل هنگامی که کارخواه یک راز به اشتراک گذاشته شده و یک شناسانه راز با کارساز داشته باشد ولی یک نقطه اعتماد در کارخواه بارگذاری نشده باشد، استفاده می‌شود، زیرا در این حالت گواهی امضای کارساز نمی‌تواند ارزیابی گردد. حالت خاصی که این کنترل برای آن ایجاد شده است کنترل Publish Trust Anchors است (که بررسی آن در حوزه این استاندارد قرار نمی‌گیرد) ولی می‌تواند با حالت‌های خاص دیگری نیز به کار رود.
OID این کنترل به صورت زیر است:

```
id-cmc-authData ::= { id-cmc 27 }
```

ساختار ASN.1 این کنترل نیز در زیر آمده است:

```
AuthPublish ::= BodyPartID
```

AuthPublish شناسانه بدنه‌ای است که به عضوی از cmcSequence در ساختار پاسخ PKI یا داده PKI جاری اشاره دارد. عنصر cmcSequence فیلد AuthenticatedData است. محتوای کپسوله شده یک id-cct-PKIData است. درصورت موفق بودن احراز هویت کنترل های موجود در Control Sequence لازم است، اجرا گردند.
اگر عملیات احراز اصالت موفق نباشد، CMCFailInfo با مقدار authDataFail بازگشت داده می‌شود.

الف-۱۳ کنترل‌های Batch Request و Batch Response

این کنترل‌ها به RA اجازه می‌دهند که چندین درخواست را با هم دریک درخواست کامل PKI جمع کرده و آنرا برای CA ارسال کند. آنگاه کارساز درخواست‌ها را پردازش کرده و نتایج آن را در یک پاسخ کامل PKI ارجاع می‌دهد.
OID کنترل Batch Request به صورت زیر شناخته می‌شود:

id-cmc-batchRequests ::= {id-cmc 28}

OID کنترل Batch Response به صورت زیر شناخته می‌شود:

id-cmc-batchResponses ::= {id-cmc 29}

ساختار ASN.1 هردوی این کنترل‌ها به صورت زیر است:

BodyPartList ::= SEQUENCE of BodyPartID

داده‌ای که با این کنترل‌ها همراه می‌شود، مجموعه‌ای از شناسانه‌های بدنه است. هر درخواست/پاسخ به صورت یک ورودی مستقل در cmcSequence یک PKIData/PKI Response جدید قرار می‌گیرد. سپس شناسانه‌های این ورودی‌ها در فهرست بخش بدنه وابسته به این کنترل قرار می‌گیرد.

هنگامی که کارساز کنترل Batch Request را پردازش می‌کند، ممکن است که پاسخ آن را در یک یا چند پاسخ PKI بدهد. مقدار Partial در CMCStatus برای همه به غیراز آخرین پاسخ PKI بازگشت داده می‌شود. اگر کنترل Batch Request پردازش شده باشد، مقدار CMCStatus برابر با success خواهد بود. پاسخ‌ها با کد CMCStatus مربوط به خود ایجاد می‌گردند. خطاها در هر درخواست مستقل به سطح بالاتر انتشار نمی‌یابند.

درهنگامی‌که یک پاسخ PKI با مقدار partial در CMCStatus بازمی‌گردد، کنترل Query Pending (طبق زیربند الف-۱۰) برای بازگرداندن نتایج بیشتری استفاده می‌شود. وضعیت بازگشت داده شده دربردارنده یک زمان پیشنهادی است که بعد از آن کارساز باید نتایج اضافه‌تر را درخواست کند.

الف-۱۴ کنترل Control Processed

این کنترل به یک RA اجازه می‌دهد که به پردازش‌کننده‌های بعدی کنترل‌ها نشان دهد که یک کنترل مشخص درقبل پردازش شده است. این امر به RA امکانی می‌دهد که در بین رشته پردازشی یک کنترل تعریف شده را به صورت محلی یا در مستند بعدی پردازش کند.
OID این کنترل به فرم زیر است:

id-cmc-controlProcessed ::= { id-cmc 32 }

ساختار ASN.1 این کنترل نیز به صورت زیر تعریف شده است:

```
ControlList ::= SEQUENCE {
        bodyList        SEQUENCE SIZE (1..MAX) OF BodyPartReference
```

- bodyList – مجموعه‌ای از شناسانه‌های بخش بدنه‌ای است که یک مسیر به هر کنترلی کـه توسـط RA پردازش می‌شود را شکل می‌دهد. این کنترل فقط برای آن دسته از کنترل‌هایی است که در این استاندارد تعریف نشده‌اند. زیرا این کنترل‌ها می‌توانند منجر به ایجاد حالت خطا در کارسازی گردند که قصد اجرای آن‌ها را دارد. هیچ وضعیت خطایی لازم نیست چرا که منجر مـی‌شـود RA درخواسـت را بـه کـارخواهی همراه با خطا بازگرداند؛ به جای این که درخواست به کارساز بعدی در فهرست پردازش ارسال شود.