INSO
16386-3
1st.Edition
Jul.2013

جمهوری اسلامی ایران
Islamic Republic of Iran
سازمان ملی استاندارد ایران
Iranian National Standardization Organization

استاندارد ملی ایران
۱۶۳۸۶ـ۳
چاپ اول
مرداد ۱۳۹۲

کارت‌های شناسایی ـ واسط‌های برنامه نویسی کارت دارای مدار مجتمع ـ قسمت ۳ : واسط برنامه کاربردی

Identification cards –Integrated circuit card programming interfaces Part 3 : Application interface

ICS:35.240.15

به نام خدا

آشنایی باسازمان ملی استاندارد ایران

مؤسسهٔاستاندارد و تحقیقات صنعتی ایران به موجب بندیک مادهٔ۳ قانون اصلاح قوانین ومقررات مؤسسهٔ استانداردو تحقیقات صنعتی ایران،مصوب بهمن ماه۱۳۷۱تنها مرجع رسمی کشوراست که وظیفه تعیین،تدوین ونشر استانداردهای ملی (رسمی) ایران رابه 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 درحوزه های مختلف درکمیسیون های فنی مرکب ازکارشناسان سازمان ،صاحبنظران مراکز و مؤسسات علمی،پژوهشی،تولیدی واقتصادی آگاه ومرتبط انجام میشود وکوششی همگام با مصالح ملی وباتوجه به شرایط تولیدی، فناوری وتجاری است که از مشارکت آگاهانه ومنصفانهٔ صاحبان حق و نفع ، شامل تولیدکنندگان ، مصرف‌کنندگان، صادرکنندگان و واردکنندگان،مراکز علمی وتخصصی ،نهادها ، سازمان های دولتی و غیر دولتی حاصل میشود.پیش نویس استانداردهای ملی ایران برای نظرخواهی به مراجع ذینفع و اعضای کمیسیون های فنی مربوط ارسال می‌شود و پس از دریافت نظرها و پیشنهادها درکمیتهٔ ملی مرتبط با آن رشته طرح و در صورت تصویب به عنوان استاندارد ملی (رسمی) ایران چاپ و منتشر میشود.

پیش نویس استانداردهایی که مؤسسات و سازمان های علاقه مند وذیصلاح نیز با رعایت ضوابط تعیین شده تهیه میکنند درکمیتهٔ ملی طرح و بررسی و در صورت تصویب ،به عنوان استاندارد ملی ایران چاپ و منتشر میشود .بدین ترتیب، استانداردهایی ملی تلقی میشوندکه براساس مفاد نوشته شده در استاندارد ملی ایران شمارهٔ ۵ تدوین و در کمیتهٔ ملی استاندارد مربوط که سازمان ملی استان دارد ایران تشکیل می‌دهد به تصویب رسیده باشد.

سازمان ملی استانداردایران از اعضای اصلی سازمان بین المللی استاندارد (ISO)[1]،کمیسیون بین المللی الکتروتکنیک (IEC)[2]وسازمان بین المللی اندازه شناسی قانونی(OIML)[3] است و به عنوان تنها رابط[4]کمیسیون کدکس غذایی (CAC)[5]درکشورفعالیت می کند. درتدوین استانداردهای ملی ایران ضمن توجه به شرایط کلی ونیازمندی های خاص کشور ،ازآخرین پیشرفت های علمی،فنی وصنعتی جهان و استانداردهای بین المللی بهره‌گیری می‌شود .

سازمان ملی استانداردایران میتواند 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 ازاستانداردهای ملی ایران را برای محصولات تولیدی داخل کشور و /یا اقلا م 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 ومدیریت زیست محیطی،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 میکند 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 گران بها 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 (OrganisationInternationale de MetrologieLegale)
4 - Contact point
5 - Codex Alimentarius Commission

کمیسیون فنی تدوین استاندارد

" کارت‌های شناسایی ـ واسط‌های برنامه نویسی کارت دارای مدار مجتمع ـ

قسمت ۳ :

واسط برنامه کاربردی"

رئیس:	**سمت و/ یا نمایندگی**
تهرانی طریقت ، محمدابراهیم (کارشناسی ارشد مدیریت فناوری اطلاعات)	مشاور ریاست سازمان ثبت احوال و قائم مقام مجری طرح کارت ملی هوشمند
دبیر:	
داوری تبریزی ، بیژن (لیسانس مهندسی صنایع)	مدیر عامل شرکت مهندسی و بهبود کیفیت شریف
اعضاء: (اسامی به ترتیب حروف الفبا)	
بداغی ، امیرحسین (کارشناسی ارشد مهندسی الکترونیک)	کارشناس سازمان فناوری اطلاعات
جمیل پناه ، ناصر (کارشناسی ارشد مدیریت)	کارشناس سازمان فناوری اطلاعات
جهان‌شاه ، فرناد (کارشناسی مهندسی نرم‌افزار)	کارشناس شرکت مهندسی و بهبود کیفیت شریف
سعیدی ، عذرا (کارشناسی ارشد مهندسی مخابرات)	کارشناس سازمان فناوری اطلاعات
صفرنیا، فتانه (کارشناسی فیزیک)	نماینده حوزه طرح کارت ملی هوشمند سازمان ثبت احوال

زنده نام ، مهدی
(کارشناسی فناوری اطلاعات)

کارشناس حوزه طرح کارت ملی هوشمند سازمان ثبت احوال

نوروزی زاده ، حمیرا
(کارشناسی مهندسی صنایع)

کارشناس استاندارد

فهرست مندرجات

پیش گفتار

استاندارد " کارت‌های شناسایی ـ واسط‌های برنامه نویسی کارت دارای مدار مجتمع ـ قسمت **۳** : واسط برنامه کاربردی" که پیش‌نویس آن در کمیسیون‌های مربوط توسط شرکت مهندسی و بهبود کیفیت شریف تهیه و تدوین شده است و در صد و شصت و دومین اجلاس کمیته ملی استاندارد خدمات مورخ ۹۱/۱۲/۲۱ مورد 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 می شود.

برای حفظ همگامی و هماهنگی با تحولات و پیشرفت‌های ملی و جهانی در زمینه صنایع، علوم و خدمات، استانداردهای ملی ایران در مواقع لزوم تجدید نظر خواهد شد و هر پیشنهادی که برای اصلاح و تکمیل این استانداردها 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.

منبع و مأخذی که برای تهیه این استاندارد مورد استفاده قرار گرفته به شرح زیر است:

ISO/IEC 24727-3:2008+Cor1:2010, Identification cards - Integrated circuit card programming interfaces - Part 3 :Application interface

مقدمه

استانداردهای ملی ایران شماره ۱۶۳۸۶ مجموعه‌ای از واسط‌های برنامه‌نویسی برای برهم‌کنش(تعامل)[۱] بین کارت‌های دارای مدار مجتمع و برنامه‌های کاربردی خارجی است که خدمات عمومی[۲] برای مصارف چند بخشی[۳] را شامل می‌شود. سازمان و عملکرد کارت های دارای مدار مجتمع با استاندارد ISO/IEC 7816-4 مطابقت دارد .

این استاندارد، قسمتی از مجموعه استانداردهای ملی ایران ۱۶۸۳۶ می‌باشد .

1 -Interaction
2-Generic services
3 - Multi-sector use

کارت‌های شناسایی- واسط‌های برنامه‌نویسی کارت دارای مدار مجتمع ـ
قسمت ۳:واسط برنامه کاربردی

۱ هدف و دامنه کاربرد

هدف از تدوین این استاندارد، تعیین و ارائه خدماتی به صورت درخواست‌های عمل و پاسخ‌های عمل مورد نظر است که در واسط خدمت برنامه کاربردی سرویس‌گیرنده[1] (کارخواه) پشتیبانی می‌شود. این خدمات، به صورت مستقل از زبان برنامه‌نویسی، توصیف شده‌اند.

این استاندارد، واسط کاربردی مدل مرجع اتصال متقابل سامانه‌های باز تعریف شده در استاندارد ملی ایران شماره ۱ـ ۱۶۲۷۴ می‌باشد. این استاندارد، یک واسط سطح بالا برای استفاده برنامه‌کاربردی سرویس‌گیرنده از انباره اطلاعات و عملیات پردازشی یک برنامه‌کاربردی کارت به صورتی که در واسط عمومی کارت در نظر گرفته شده، فراهم می‌نماید.

این استاندارد، روش پیاده‌سازی مشخصی را برای این واسط، الزام نمی‌نماید.

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 ملی ایران محسوب می‌شوند.

در صورتی که به مدرکی با ذکر تاریخ انتشار ارجاع داده شده باشد، اصلاحیه‌ها و تجدیدنظرهای 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 آن مورد نظر است.

۲ ـ ۱ استاندارد ملی ایران شماره ۱ـ۱۶۳۸۶،کارت‌های شناسایی ـ واسط‌های برنامه نویسی کارت دارای مدارمجتمع ـ قسمت ۱: معماری

۲ ـ ۲ استاندارد ملی ایران شماره ۲ـ۱۶۳۸۶، کارت‌های شناسایی ـ واسط‌های برنامه نویسی کارت دارای مدارمجتمع ـ قسمت ۲: واسط عمومی کارت

2-3 ISO/IEC 7816-11, Identification cards — Integrated circuit cards — Part 11: Personal verification through biometric methods

2-4 IETF RFC 2141, URN Syntax, May 1997

۳ اصطلاحات و تعاریف

در این استاندارد علاوه بر اصطلاحات و تعاریف تعیین‌شده در استانداردهای ملی ایران شماره ۱ـ ۱۶۳۸۶و شماره ۲ ـ ۱۶۳۸۶، اصطلاحات و تعاریف زیر نیز به‌کار می‌رود :

1- Client application

۱-۳

فهرست کنترل دسترسی[1]

مجموعه قواعد دسترسی

۲-۳

مجوز دسترسی[2]

قابلیت اعطا شده برای انجام یک عمل

۳-۳

قاعده دسترسی[3]

ارتباط یک عمل و یک وضعیت امنیتی در زمینه یک برنامه‌کاربردی کارت

۴-۳

عمل[4]

کاری که یک برنامه‌کاربردی سرویس‌گیرنده می‌تواند در واسط کاربردی این استاندارد از یک برنامه کارت، تقاضا نماید

۵-۳

برنامه‌کاربردی کارت آلفا[5]

برنامه مدیریتی کارت، مربوط به استاندارد ملی ایران شماره ۱۶۳۸۶ که قابلیت شناسایی کارت و برنامه کاربردی و خدمات مدیریتی را فراهم می‌نماید

۶-۳

مسیر برنامه کاربردی کارت[6]

مجموعه منظم از نقاط پایانی پروتکل در شبکه که برنامه کاربردی سرویس گیرنده را به برنامه کاربردی کارت، مرتبط می‌کند

۷-۳

خدمت برنامه کاربردی کارت[7]

مجموعه‌ای از عمل‌ها

۸-۳

کانال[8]

مسیر فیزیکی جابجایی بیت‌های اطلاعات میان یک برنامه کاربردی سرویس گیرنده و یک برنامه کاربردی کارت

1- Access control list
2 - Access permission
3 - Access rule
4 - Action
5 - Alpha Card-application
6 - Card-application-path
7 - Card-application-service
8 - Channel

۳-۹

برنامه کاربردی سرویس گیرنده[1]

جزء نرم‌افزاری که بر روی یک پلاتفرم[2]، اجرا می‌شود و از انباره داده‌ها و خدمات محاسباتی ارائه شده به وسیله یک برنامه کاربردی کارت، استفاده می‌نماید

۳-۱۰

اتصال[3]

کانالی که به‌طور منطقی به آن ارجاع شده است

۳-۱۱

هویت متمایز کننده سراسری[4]

هویت متمایز کننده‌ای که در تمام برنامه‌های کاربردی کارت که به وسیله برنامه کاربردی کارت آلفای یکسان، مدیریت می‌شوند، شناخته می‌شود

۳-۱۲

هویت متمایز کننده محلی[5]

هویت متمایز کننده‌ای که فقط در یک برنامه کاربردی کارت مشخص که درون آن تعریف شده است، شناخته می-شود

۳-۱۳

پارامتر[6]

اطلاعات لازم برای تعریف یک عمل یا تاثیر بر آن

۳-۱۴

کد بازگشتی[7]

اطلاعاتی حاوی وضعیت، که به عنوان نتیجه یک عمل، برگردانده می‌شود

۳-۱۵

وضعیت امنیتی[8]

عبارت بولین[9] به صورت وضعیت‌های احراز هویت مربوط به هویت متمایز کننده

۳-۱۶

جلسه[10]

اتصال مورد استفاده تحت شرایط امنیتی خاص

1- Client-application
2-Platform
3 - Connection
4 - Global Differential-Identity
5 - Local Differential-Identity
6 - Parameter
7 - Return Code
8 - Security Condition
9 - Boolean
10 - Session

۳-۱۷

هدف[1]

هستار پایداری که باید به وسیله عمل‌های خدمت برنامه کاربردی کارت، تغییر داده شود[2]

۴ کوته‌نوشت‌ها

در این استاندارد علاوه بر کوته‌نوشت‌های مشخص شده در استاندارد ملی ایران شماره ۱ ـ ۱۶۳۸۶، موارد زیر نیز بکار می‌رود.

ACL فهرست کنترل دسترسی

AR قانون دسترسی

DID هویت متمایز کننده

DSI ساختار داده‌ای برای تعامل‌پذیری[3]

۵ سازماندهی برای تعامل پذیری

۵-۱ کلیات

یک برنامه سرویس‌گیرنده و یک برنامه کاربردی کارت، دو هستار سطح همتا[4] را تشکیل می‌دهند که از طریق تراکنشی که به خوبی تعریف شده، و سازوکارهای ارتباطی، با هم تعامل دارند. واسط برنامه کاربردی این استاندارد باید تنها سازوکاری باشد که از طریق آن یک برنامه کاربردی سرویس‌گیرنده با یک برنامه کاربردی کارت، تعامل می‌نماید.

این بند، مدل محاسباتی و هستارهای پایدار روی واسط برنامه کاربردی این استاندارد را تعریف می‌نماید که به وسیله آن‌ها برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، تعامل می‌کنند. یک مدل امنیتی، سازوکارهای امنیتی که از طریق آن‌ها در این تعامل، اعتماد حاصل می‌شود اداره می‌نماید.

بند ۵-۲ هستارهای مفهومی روی واسط برنامه کاربردی این استاندارد را معرفی می‌نماید و رفتار عملیاتی و روابط آن‌ها را شرح می‌دهد. بند ۵-۳ هستارها و روابط ارائه شده روی واسط برنامه کاربردی این استاندارد را مشخص می‌نماید. زیربند ۵-۴، مدل امنیتی فراهم‌شده روی واسط برنامه کاربردی این استاندارد را مشخص می‌کند.

۵-۲ مدل محاسباتی

با استفاده از اصطلاحات تعریف شده در استاندارد ملی ایران شماره ۱ ـ ۱۶۳۸۶:

۵-۲-۱ یک برنامه کاربردی سرویس‌گیرنده، مجاز است که از یک مسیر برنامه کاربردی کارت به یک برنامه کاربردی کارت، مطلع باشد. با استفاده از این مسیر، یک برنامه کاربردی سرویس‌گیرنده می‌تواند یک اتصال به یک برنامه

---

1 - Target
2 - Manipulate
3 - Interoperability
4 - Peer-level

کاربردی کارت را آغاز نماید. برنامه کاربردی کارت باید به عنوان برنامه کاربردی کارت جاری، برای آن اتصال، شناخته شده باشد.

۵-۲-۲ واسط برنامه کاربردی این استاندارد مجموعه‌ای از خدمات برنامه کاربردی کارت است که برای برنامه کاربردی سرویس‌گیرنده، ارائه شده‌اند. هر خدمت برنامه کاربردی کارت باید متشکل باشد از اعمالی که ممکن است به وسیله یک برنامه کاربردی سرویس‌گیرنده، درخواست شوند و تاییدیه‌هایی[۱] که به وسیله SAL[۲] برگردانده شده-اند.

۵-۲-۳ اگر برنامه کاربردی سرویس‌گیرنده، از یک مسیر یک برنامه کاربردی کارت به برنامه کاربردی کارت، مطلع نباشد، مجاز است که در واسط برنامه کاربردی این استاندارد برای کشف یک مسیر به برنامه کاربردی کارت مورد تقاضا، درخواست نماید.

۵-۲-۴ یک برنامه کاربردی کارت باید به وسیله یک AID[۳] بطور یکتا مشخص شود.

۵-۲-۵ برنامه کاربردی کارت آلفا، به وسیله برنامه کاربردی سرویس‌گیرنده، اساس مدیریت قابل اعتماد برنامه‌های کاربردی کارت را فراهم می‌کند.

۵-۲-۶ یک برنامه کاربردی سرویس‌گیرنده، مجاز است که بیش از یک اتصال را باز (برقرار) نماید. یک برنامه کاربردی سرویس‌گیرنده، مجاز است که بیش از یک اتصال را با همان برنامه کاربردی کارت، برقرار نماید، به‌طوری که هر اتصال به وسیله یک اداره کردن اتصال[۴] متفاوت، ارجاع داده شود.

۵-۲-۷ با استفاده از یک اتصال باز، یک برنامه کاربردی سرویس‌گیرنده مجاز است که با یک برنامه کاربردی کارت، یک جلسه آغاز نماید.

۵-۲-۸ یک برنامه کاربردی سرویس‌گیرنده مجاز است که بیش از یک جلسه، را باز کند. درون یک اتصال، فقط یک جلسه مجاز است که باز باشد.

۵-۲-۹ وضعیت جاری[۵]، وضعیت جاری یک اتصال است که به وسیله برنامه کاربردی کارت جاری، مجموعه داده[۶] جاری، وضعیت احراز هویت‌های متمایز کننده شناخته شده، و وجود یک جلسه در اتصال، تعریف می‌شود.

۵-۲-۱۰ یک برنامه کاربردی کارت، دربرگیرنده یک یا چند خدمت برنامه کاربردی کارت است.

۵-۲-۱۱ یک خدمت برنامه کاربردی کارت مشخص، خدمت داده نامگذاری شده[۷]، دسترسی به هیچ یا چند مجموعه مجموعه داده را فراهم می‌نماید.

۵-۲-۱۲ یک مجموعه داده شامل هیچ یا چند ساختارداده برای تعامل‌پذیری است (DSI)[۸]. یک مجموعه داده باید دامنه نامگذاری و قواعد دسترسی را به DSIهای درون خود، ارائه کند.

۵-۲-۱۳ هر مجموعه داده باید مطابق قواعد دسترسی حاکم بر عمل‌های موجود از طریق خدمت داده نامگذاری شده، قابل دسترسی باشد.

---

1 - Confirmations
2- Service Access Layer (SAL)
3- Application Access Identifier (AID)
4 - Connection Handle
5 - Current State
6- Data-set
7 - Named Data Service
8- Data Structure for Interoperability (DSI)

۵-۲-۱۴ مجموعه داده اخیرا انتخاب شده در یک برنامه کاربردی کارت باید به عنوان مجموعه داده جاری، شناخته شود.

۵-۲-۱۵ یک درخواست عمل منفرد در واسط برنامه کاربردی این استاندارد مجاز است که به چند درخواست عمومی در واسط عمومی کارت استاندارد ملی ایران شماره ۲ ـ ۱۶۳۸۶ ترجمه شود.

۵-۲-۱۶ یک برنامه کاربردی کارت مجاز است چند عمل را که ممکن است به وسیله برنامه کاربردی سرویس‌گیرنده درخواست شوند، به وسیله واسط برنامه کاربردی این استاندارد پشتیبانی کند.

۵-۲-۱۷ یک درخواست عمل باید یک تاییدیه عمل[1] تولید کند.

۵-۲-۱۸ یک قاعده دسترسی، از نام یک عمل و یک وضعیت امنیتی تشکیل می‌شود. فرض می‌شود که وضعیت امنیتی به وسیله قاعده دسترسی، به عمل، مرتبط است.

۵-۲-۱۹ یک فهرست کنترل دسترسی[2]، مجموعه‌ای از هیچ یا چند قاعده دسترسی است. یک فهرست کنترل دسترسی باید با هر هدف، مرتبط باشد.

۵-۲-۲۰ عملی با یک هدف، ممکن است که با موفقیت انجام شود اگر و تنها اگر وضعیت امنیتی مرتبط با آن عمل، به وسیله یک قاعده دسترسی درون فهرست کنترل دسترسی مربوط به آن هدف، به TRUE[3] ارزیابی شود.

۵-۲-۲۱ یک پروتکل احراز هویت، فرآیندی است که به وسیله آن یک هویت متمایزکننده، مالکیت یک علامت‌گذار[4] را نشان می‌دهد.

۵-۲-۲۲ احراز هویت، اجرای موفقیت‌آمیز یک پروتکل احراز هویت می‌باشد. در این مورد، فرض می‌شود که هویت متمایزکننده، احراز هویت شده است.

۵-۲-۲۳ وضعیت احراز هویت یک هویت متمایزکننده باید یک متغیر بولین باشد که در صورت احراز هویت متمایزکننده، TRUE و در غیر این صورت، FALSE است.

۵-۲-۲۴ یک برنامه کاربردی سرویس‌گیرنده، مجاز است که اطلاعات، وضعیت یا خدمات ارائه شده به وسیله یک برنامه کاربردی کارت را از طریق یک فرآیند کشف یاد بگیرد که این فرآیند از طریق اعمال مختلف List، Get و Describe ارائه شده در واسط برنامه کاربردی استاندارد ملی ایران شماره ۳ ـ ۱۶۳۸۶ به وجود آمده‌اند.

۵-۲-۲۵ قواعد دسترسی، در واسط برنامه کاربردی این استاندارد قابل کشف هستند.

۵-۲-۲۶ بطورکلی، درجه تعامل‌پذیری یک برنامه کاربردی کارت با برنامه‌های کاربردی سرویس‌گیرنده، به اطلاعات قابل کشف ارائه شده به وسیله برنامه کاربردی کارت در واسط برنامه کاربردی این استاندارد بستگی دارد.

---

1 - Action Confirmation
2 - Access Control List

۳ ـ کلمات انگلیسی به‌کاررفته در داخل متن یا دستور و یا پارامتر مربوط به زبان برنامه‌نویسی هستند .

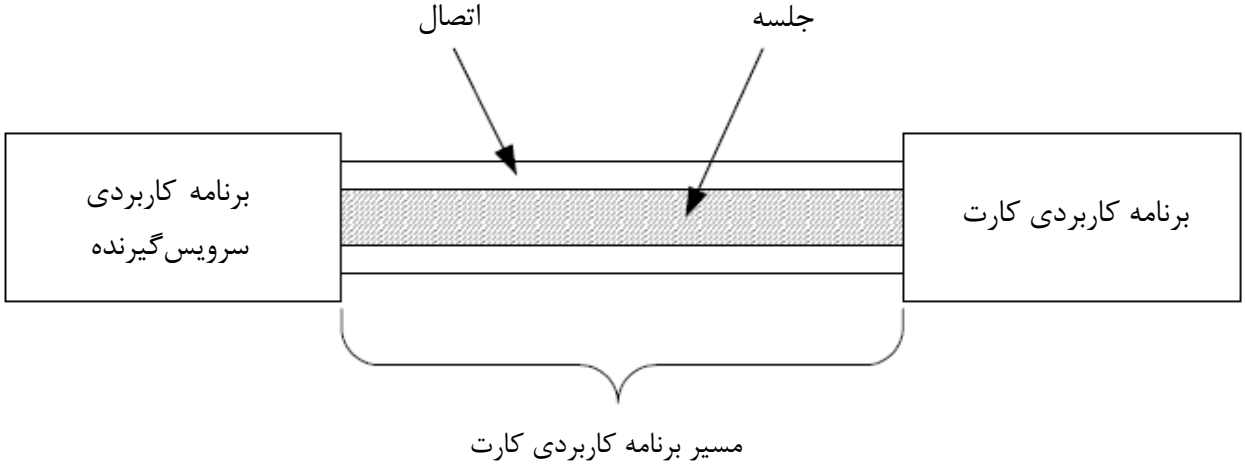
4- Marker

۵-۳ روابط هستار در واسط برنامه کاربردی

۵-۳-۱ کلیات

این بند، هستارهای قابل دسترسی از طریق یک برنامه کاربردی کارت، بخصوص برنامه کاربردی کارت آلفا و برنامه-های کاربردی که به وسیله آن مدیریت می‌شوند را شرح می‌دهد. هستارهایی که به طور مستقیم، بیشتر درگیر مدل امنیتی هستند، در بند ۵-۴ توضیح داده شده‌اند.

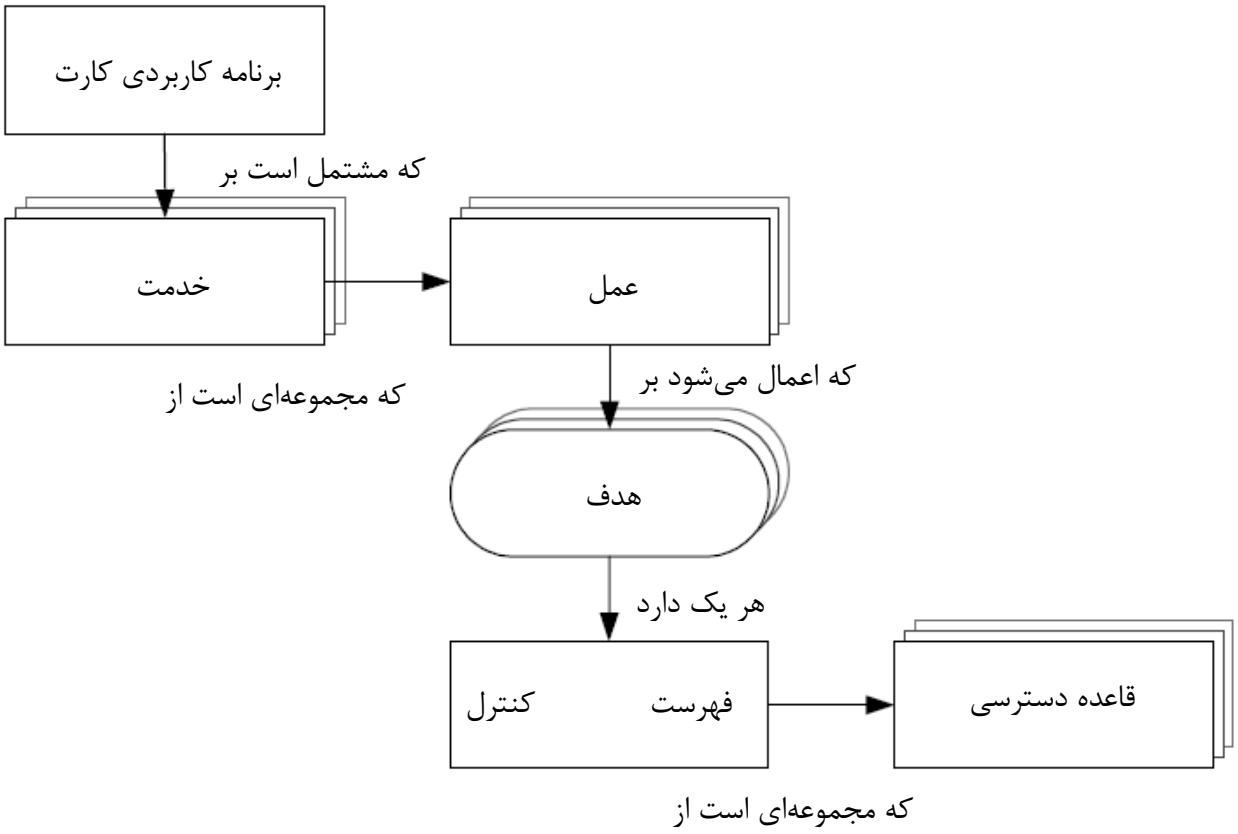
شکل ۱، الگوی برنامه کاربردی سرویس‌گیرنده به برنامه کاربردی کارت را نشان می‌دهد که این استاندارد بر پایه آن قرار دارد. واسط برنامه کاربردی این استاندارد، سازوکارهای اتصال و جلسه نشان داده شده در این شکل را تسهیل می‌نماید.



شکل ۱- الگوی اتصال برنامه کاربردی

برای اتصال یک برنامه کاربردی سرویس‌گیرنده به یک برنامه کاربردی کارت، می‌توان یک مسیر برنامه کاربردی کارت را مشخص نمود. از طریق چنین اتصالی، برنامه کاربردی سرویس‌گیرنده می‌تواند با استفاده از واسط برنامه کاربردی این استاندارد، به خدمات برنامه کاربردی کارت دسترسی یابد. یک جلسه، علاوه بر خصوصیات امنیتی ذاتی خود که می‌توانند برای حفاظت از داده‌های در جریان بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، مورد استفاده قرار گیرند، یک زمینه امنیتی نیز به اتصال می‌افزاید.

شکل ۲ هستارهای تعریف شده در مدل محاسباتی را نشان می‌دهد و روابط بین این هستارها را برقرار می‌نماید. این نمودار هستار رابطه برای توصیف تک‌تک خدمات برنامه کاربردی کارت، در زیر توضیح داده شده است. نوع هدف قابل استفاده برای هر عمل، در کادرهای با گوشه‌های گرد مربوط شده به هر عمل، نشان داده شده است.



شکل۲- مدل محاسبه هستارها و روابط

یک برنامه کاربردی کارت، مجموعه‌ای از هدف‌ها و خدمات است. یک خدمت، مجموعه‌ای از عمل‌ها است. یک عمل، کاری است که بر روی یک هدف، انجام می‌شود. با استفاده از واسط برنامه کاربردی این استاندارد، عمل‌ها و خدمات یک برنامه کاربردی کارت، به وسیله برنامه کاربردی سرویس‌گیرنده، مورد دسترسی قرار می‌گیرند.

۵-۳-۲ برنامه کاربردی کارت آلفا

برنامه کاربردی کارت آلفا باید در واسط برنامه کاربردی در دسترس باشد. این برنامه مجاز است که در ICC [1]وجود داشته باشد یا به وسیله SAL یا پیاده‌سازی GCAL[2] شبیه‌سازی شود. در تمام موارد، یک اتصال به برنامه کاربردی کارت آلفا به وسیله برنامه کاربردی سرویس‌گیرنده، دسترس‌پذیری فهرست برنامه کاربردی کارت را فراهم می‌کند.

۵-۳-۳ دسترسی به خدمات برنامه کاربردی کارت

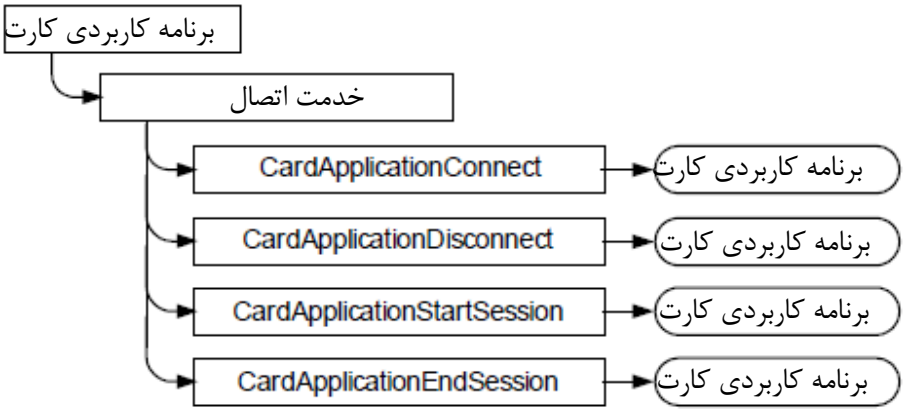
یک برنامه کاربردی سرویس‌گیرنده برای دسترسی به خدمات برنامه کاربردی کارت، از نقاط ورودی[3] Initialize و CardApplicationPath در واسط برنامه کاربردی این استاندارد، استفاده می‌کند. یک برنامه کاربردی سرویس‌گیرنده، برای پایان دادن به دسترسی به خدمات برنامه کاربردی کارت، از نقطه ورودی Terminate استفاد می‌نماید.

1-Integrated Circuit Card
2- Generic Card Access Layer
3 - Entry ponits

یک برنامه کاربردی سرویس‌گیرنده باید قبل از دسترسی به خدمات برنامه کاربردی کارت، نقطه ورودی Initialize روی واسط برنامه کاربردی این استاندارد را درخواست نماید. پس از تایید Initialize، برنامه کاربردی سرویس‌گیرنده مجاز است که اتصال‌ها را باز کند و از برنامه‌های کاربردی کارت متعددی به طور همزمان یا پی در پی، درخواست عمل نماید. توصیه می‌شود هنگامی که استفاده از خدمات برنامه کاربردی کارت، دیگر مورد نیاز نیست، برنامه کاربردی سرویس‌گیرنده، نقطه ورودی Terminate را درخواست کند.

۵-۳-۴ واسط خدمت اتصال

این خدمت برنامه کاربردی کارت، عمل‌هایی را برای برقراری یک اتصال بین یک برنامه کاربردی سرویس‌گیرنده و یک برنامه کاربردی کارت، فراهم می‌کند. به محض اینکه یک اتصال، برقرار شد برای بهبود خصوصیات امنیتی ارتباط بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، جلسه ممکن است که از طریق این اتصال، تحت تاثیر واقع شود. شکل ۳، روابط هستار خدمت اتصال را نشان می‌دهد.

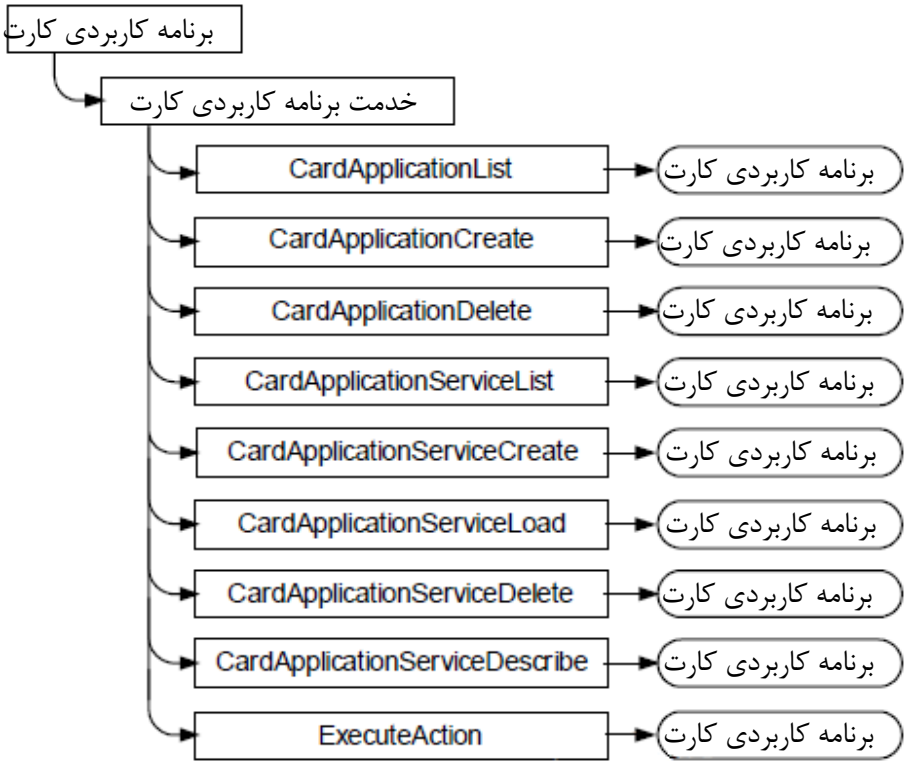


شکل ۳- خدمت اتصال

همانگونه که در نمودار بالا مشخص شده است، هدف هر یک از این عمل‌ها، برنامه کاربردی کارت جاری یا برنامه کاربردی کارتی است که برنامه کاربردی سرویس‌گیرنده تلاش می‌کند به آن متصل شود.

۵-۳-۵ واسط خدمت برنامه کاربردی کارت

این خدمت برنامه کاربردی کارت، عمل‌هایی را برای ایجاد و تغییر برنامه‌های کاربردی کارت، ارائه می‌کند. شکل ۴، روابط هستار خدمت برنامه کاربردی کارت را نشان می‌دهد.



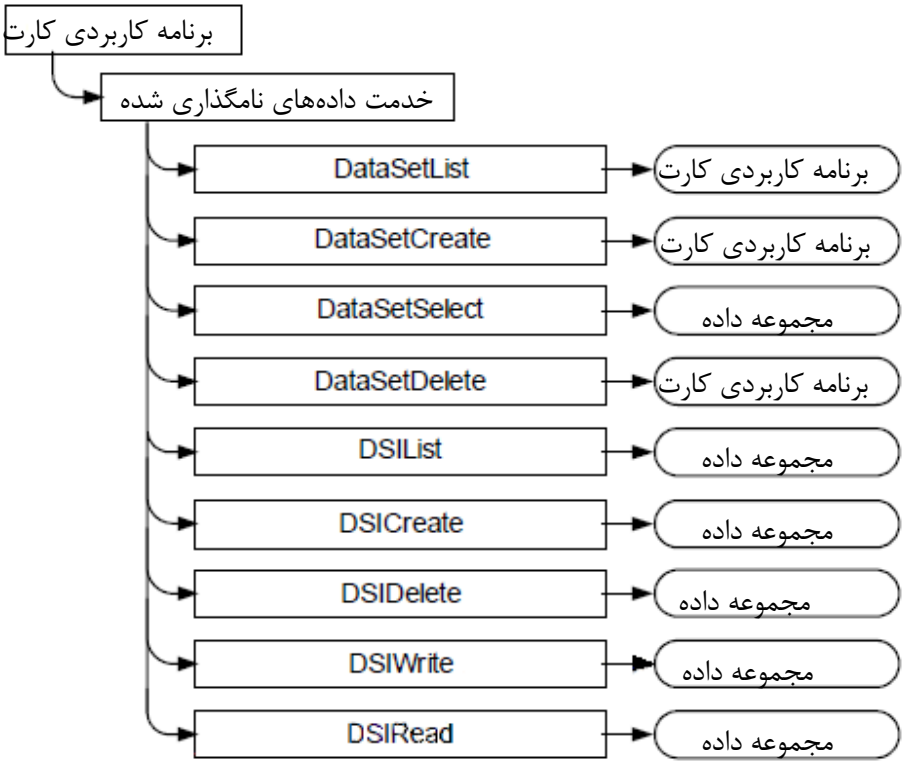
شکل ۴- خدمت برنامه کاربردی کارت

هدف عمل‌های CardApplicationCreate و CardApplicationDelete، برنامه کاربردی کارت آلفا است. هنگام درخواست نمودن این عمل‌ها، برنامه کاربردی کارت آلفا باید برنامه کاربردی کارت جاری باشد.

هدف هر یک از عمل‌های دیگر، برنامه کاربردی کارت جاری است.

۵-۳-۶ واسط خدمت داده‌های نامگذاری شده

این خدمت برنامه کاربردی کارت، عمل‌هایی را برای ایجاد و تغییر مجموعه‌های داده‌ها فراهم می‌کند، یک سازوکار محدودکننده که برقراری قواعد دسترسی مشترک را برای داده‌های درون ساختارهای داده‌ای برای تعامل‌پذیری، فراهم می‌کند. مجموعه‌های داده‌ها مجاز هستند که شامل هر تعدادی از DSI باشند. شکل ۵، روابط هستار خدمت داده‌های نامگذاری شده را نشان می‌دهد.

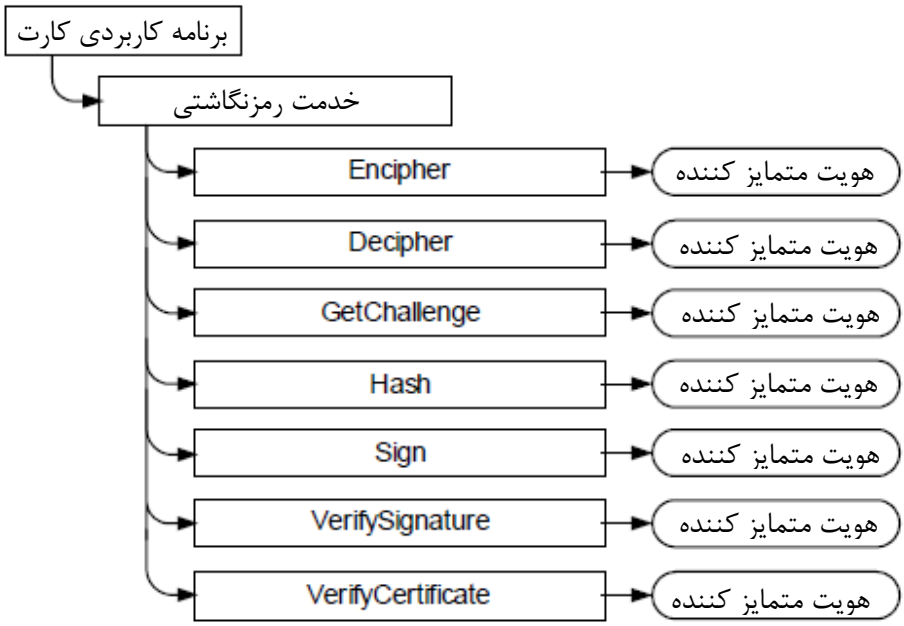


شکل ۵- خدمت داده‌های نامگذاری شده

همانگونه که در نمودار بالا مشخص شده است، هدف هر یک از این عمل‌ها، برنامه کاربردی کارت جاری، مجموعه داده جاری یا مجموعه داده است که برنامه کاربردی سرویس‌گیرنده تلاش می‌کند آن را انتخاب نماید.

۵-۳-۷ واسط خدمت رمزنگاشتی

این خدمت برنامه کاربردی کارت، عمل‌هایی را برای گستره‌ای از عملیات رمزنگاشتی، فراهم می‌نماید که قرار است بر روی پارامترهای درون یک هویت متمایز کننده یا به وسیله آن‌ها انجام شوند. شکل ۶، روابط هستار خدمت رمزنگاشتی را نشان می‌دهد.

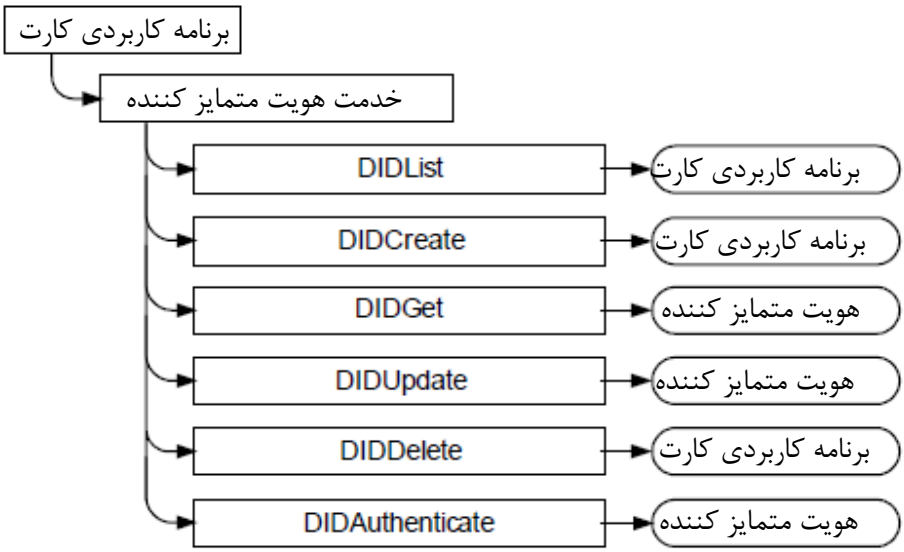


شکل ۶- خدمت رمزنگاشتی

همانگونه که در نمودار بالا مشخص شده است، هدف هر یک از این عمل‌ها، هویت متمایزکننده نامگذاری‌شده در درخواست عمل است.

۵-۳-۸ واسط خدمت هویت متمایزکننده

این خدمت برنامه کاربردی کارت، عمل‌هایی را برای ایجاد و تغییر هویت‌های متمایزکننده، فراهم می‌کند.شکل ۷ ، روابط هستار خدمت هویت متمایزکننده را نشان می‌دهد.

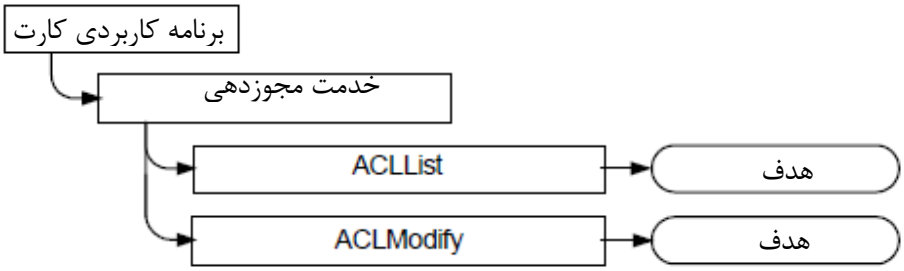


شکل ۷ - خدمت هویت متمایز کننده

همانگونه که در نمودار بالا مشخص شده است، هدف هر یک از این عمل‌ها، برنامه کاربردی کارت جاری یا هویت متمایزکننده نامگذاری‌شده در درخواست عمل است.

۵-۳-۹ واسط خدمت مجوزدهی[1]

این خدمت برنامه کاربردی کارت، عمل‌هایی را برای تغییر فهرست‌های کنترل دسترسی، فراهم می‌کند. شکل ۸، روابط هستار خدمت مجوزدهی را نشان می‌دهد.



شکل ۸ – خدمت مجوزدهی

همانگونه که در نمودار بالا مشخص شده است، هدف هر یک از این عمل‌ها، می‌تواند هر هدفی باشد. هدف، در درخواست عمل، مشخص شده است.

۵-۴ مدل امنیتی

۵-۴-۱ کلیات

به وسیله اجرای موفقیت‌آمیز پروتکل‌های لازم بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، یک وضعیت امنیتی اشتراکی بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، برقرار می‌شود. برقراری وضعیت امنیتی لازم، به عمل‌های درخواستی برنامه کاربردی سرویس‌گیرنده از برنامه کاربردی کارت، اجازه اجرا می‌دهد.

سازوکار مشخص نمودن وضعیت امنیتی لازم برای اجرای یک عمل، باید همان قاعده دسترسی باشد. یک وضعیت امنیتی، به وسیله احراز هویت‌های متمایزکننده، برقرار می‌شود.

۵-۴-۲ هویت متمایزکننده

برای برقرار کردن یک وضعیت امنیتی شامل یک برنامه کاربردی سرویس‌گیرنده و یک برنامه کاربردی کارت، باید از سازوکار احراز هویت متمایزکننده استفاده شود. در جدول ۱، اجزاء هر هویت متمایزکننده، نشان داده شده‌اند. در واسط عمومی کارت، یک هویت متمایزکننده، مجاز است به صورتی که در پیوست پ، به وسیله ASN.1 تعریف شده، منتقل شود.

1 - Authorization

جدول ۱ – اجزاء هویت متمایز کننده

جزء ۱	جزء ۲	جزء ۳	جزء ۴	جزء ۵
DIDName	پروتکل احراز هویت	علامت‌گذار	دامنه	توصیف‌کننده
اجباری	اجباری	اجباری	ضمنی	اختیاری

در زمان ایجاد هویت متمایز کننده، جزء DIDName باید به وسیله برنامه کاربردی سرویس‌گیرنده‌ای که آن هویت متمایز کننده را ایجاد می‌کند، مشخص شود.

جزء پروتکل احراز هویت باید به صورت یک شناسانه شیء (OID)[1] مشخص شود و تعیین می‌نماید که هویت متمایزکننده برای کدام عمل‌ها می‌تواند استفاده شود. تنها اگر پروتکل احراز هویت مشخص شده به وسیله این شناسانه شیء، همانگونه که در پیوست الف تعریف شده، برای استفاده در احراز هویت، تعریف شده باشد، اتمام موفقیت‌آمیز آن برای هویت متمایز کننده باید وضعیت احراز هویت متمایز کننده را به TRUE تنظیم نماید. یک DIDName فقط باید به یک پروتکل احراز هویت، مرتبط باشد. عمل‌های خدمت رمزنگاشتی، مجاز هستند که همانگونه که در پیوست الف، تعریف شده از اطلاعات درون جزء علامت‌گذار، استفاده کنند.

یادآوری۱ ــ پروتکل احراز هویت مشخص‌شده به وسیله جزء ۲، برای احراز هویت یا برای سایر خدمات رمزنگاشتی استفاده می‌شود.

جزء دامنه، ضمنی است. هویت‌های متمایز کننده تعریف شده درون برنامه کاربردی کارت آلفا، از لحاظ دامنه، سراسری هستند و در نتیجه درون تمام برنامه‌های کاربردی کارت مدیریت شده به وسیله این برنامه کاربردی کارت آلفا، شناخته می‌شوند. تمام هویت‌های متمایز کننده دیگر، از لحاظ دامنه، برای برنامه کاربردی کارتی که درون آن تعریف شده‌اند، محلی هستند. دامنه هویت متمایز کننده باید با دامنه وضعیت احراز هویت مربوطه، یکسان باشد.

جزء توصیف‌کننده اختیاری، مجاز است که جزئیاتی را در مورد استفاده از هویت متمایزکننده، ارائه کند. به عنوان مثال، مجاز است که به وسیله شناسانه شیء به یک مشخصه خارجی، اشاره کند.

جزء علامت‌گذار، اطلاعات درون برنامه کاربردی کارت یا یک ارجاع به آن است، که باید در اجرای پروتکل احراز هویت مشخص شده درون جزء پروتکل احراز هویت، مورد استفاده قرار گیرد.

نمونه‌هایی از علامتگذارها شامل موارد زیر هستند:

- یک PIN
- یک کلمه عبور
- یک کلید متقارن
- یک کلید نامتقارن
- یک گواهی دیجیتال
- یک تصویر یا الگوی زیست سنجی

1-Object Identifier (OID)

- یک جفت کلید متقارن؛ به عنوان مثال، یکی برای رمزگذاری و یکی برای تولید کد احراز هویت پیام (MAC)

همانگونه که در سطر آخر فهرست قبلی مشاهده شد، علامت‌گذار مجاز است که از چند بخش، به عنوان مثال از چند کلید، تشکیل شود. این امر، پروتکل‌های احراز هویت پیچیده‌ای را فراهم می‌آورد که مجاز هستند چند کلید را به طور موازی یا پی در پی استفاده کنند.

علامت‌گذار، اغلب ممکن است درون یک برنامه کاربردی کارت، به صورت یک کلید، پیاده‌سازی شود که همانگونه که در استاندارد ISO/IEC 7816-4 مشخص شده، به وسیله یک مرجع کلید به آن اشاره می‌شود. در بعضی از پروتکل‌های احراز هویت پیچیده، ممکن است که یک هویت متمایز کننده و علامت‌گذار آن، به چند مرجع کلید، مرتبط باشند. مرجع کلید، مجاز است همان‌گونه که در شرایط امنیتی ISO/IEC 7816-4 تعریف شده است، کد-گذاری شود. یک مرجع کلید مشخص، ممکن است در چندین هویت متمایز کننده، وجود داشته باشد.

رابطه DIDName با یک علامت‌گذار و پروتکل احراز هویت، سازوکاری است که به وسیله آن، هویت متمایز کننده به یک علامت‌گذار، مرتبط می‌شود. این نگاشت باید به عنوان بخشی از اطلاعات عمل کشف، نگهداری شود. این نگاشت به برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، امکان می‌دهد که اطلاعات یک هویت متمایز-کننده را شناسایی، مبادله و به اشتراک بگذارند.

۵-۴-۳ پروتکل‌های احراز هویت

پروتکل احراز هویت باید سازوکاری باشد که به وسیله برنامه کاربردی سرویس‌گیرنده، برای تنظیم وضعیت احراز هویت برای هویت متمایز کننده مشخص شده به وسیله DIDName استفاده می‌شود. پروتکل‌های احراز هویت، در جدول ۲ ارائه شده و در پیوست الف، شرح داده شده‌اند.

جدول ۲ – پروتکل‌های احراز هویت

تایید ساده	پروتکل کنترل دسترسی توسعه یافته پودمانی (M-EAC)
احراز هویت داخلی نامتقارن	انتقال کلید با احراز هویت دوطرفه مبتنی بر RSA
احراز هویت خارجی نامتقارن	دستیابی به سن
احراز هویت داخلی متقارن	برقراری کلید جلسه نامتقارن
احراز هویت خارجی متقارن	مقایسه PIN ایمن
مقایسه	توافق کلید EC با احراز هویت برنامه کاربردی کارت
مقایسه PIN	توافق کلید EC با احراز هویت دوطرفه
مقایسه زیست سنجی	توافق کلید EC-DH ساده
احراز هویت دوطرفه با استقرار کلید	احراز هویت نامتقارن GP
احراز هویت دوطرفه برنامه کاربردی سرویس‌گیرنده با استقرار کلید	احراز هویت متقارن GP (حالت صریح)
احراز هویت خارجی نامتقارن برنامه کاربردی سرویس‌گیرنده	احراز هویت متقارن GP (حالت ضمنی)

برای تنظیم کردن وضعیت احراز هویت یک هویت متمایز کننده به TRUE، پروتکل احراز هویت آن هویت متمایز-کننده باید با استفاده از عمل CardApplicationStartSession یا عمل DIDAuthenticate به طور موفقیت‌آمیز اتمام یافته باشد.

همانگونه که در پیوست الف تعریف شده، درخواست‌های پی در پی عمل یکسان برای اداره کردن اتصال یکسان و هویت متمایز کننده نامگذاری شده، ممکن است برای اتمام پروتکل احراز هویت، ضروری باشند. این درخواست‌ها نمایانگر مراحل منفرد در یک پروتکل احراز هویت چند مرحله‌ای می‌باشند.

وضعیت‌های احراز هویت مربوط به هویت متمایز کننده محلی، برای اداره کردن اتصال مشخص شده در خلال احراز هویت آن‌ها، معتبر هستند. هنگامی که برای آن اداره کردن اتصال، عمل CardApplicationDisconnect درخواست می‌شود، باید وضعیت‌های احراز هویت محلی، به FALSE تنظیم شود یا در غیر این صورت اداره کردن اتصال، نا-معتبر می‌شود. وضعیت‌های احراز هویت سراسری باید برای تمام اتصال‌ها به برنامه‌های کاربردی کارت، که به وسیله این برنامه کاربردی کارت آلفا مدیریت می‌شوند، معتبر باشند. تا زمانی که اداره کردن اتصالی که برای برقراری وضعیت‌های احراز هویت سراسری استفاده شده بود، قطع شود، وضعیت‌های احراز هویت سراسری، موثر باقی می-ماند یا در غیر این صورت، نامعتبر می‌شود.

هنگامی که عمل DIDAuthenticate یا عمل CardApplicationStartSession درخواست می‌شود، وضعیت احراز هویت یک هویت متمایز کننده باید به FALSE تنظیم شود و تنها در زمان تایید عمل ایجاد شده برای آن درخواست که به طور موفقیت‌آمیز، پروتکل احراز هویت را پایان می‌دهد به TRUE تنظیم می‌شود.

وضعیت احراز هویت یک هویت متمایز کننده، که در زمان ارزیابی وضعیت احراز هویت آن، نامشخص است، باید FALSE فرض شود.

۵-۴-۴ کلیدهای جلسه

یک پروتکل، مجاز است که شامل ایجاد کلیدهای رمزنگاشتی برای استفاده در خلال یک جلسه باشد.

هنگامی که عمل CardApplicationDisconnect درخواست می‌شود، کلیدهای جلسه باید نامعتبر بشوند، در غیر این صورت، اتصال، نامعتبر می‌شود. علاوه بر این، اگر پروتکل به وسیله درخواست نمودن عمل CardApplicationStartSession اتمام یابد،کلیدهای جلسه به هنگام درخواست عمل CardApplicationEndSession، باید نامعتبر شوند.

۵-۴-۵ فهرست‌های کنترل دسترسی

یک فهرست کنترل دسترسی (ACL)، مجموعه‌ای از قواعد دسترسی (ARهای) قابل اعمال به یک هدف است. یک AR، یک عمل خدمت برنامه کاربردی کارت را با یک وضعیت امنیتی، مرتبط می‌نماید. یک وضعیت امنیتی، یک عبارت منطقی بر حسب وضعیت‌های احراز هویت متمایزکننده است. یک هدف، عبارت از یک مجموعه داده، یک هویت متمایز کننده یا یک برنامه کاربردی کارت است.

احراز هویت کردن یک هویت متمایز کننده باید وضعیت احراز هویت آن را به TRUE تنظیم کند. در غیر این صورت، وضعیت احراز هویت آن باید FALSE باشد. اگر وضعیت امنیتی به TRUE ارزیابی شود، عمل مربوطه، مجاز است. اگر وضعیت امنیتی به FALSE ارزیابی شود، عمل مربوطه، مجاز نیست.

یک عمل شامل یک هدف، باید فقط در صورتی اجرا شود که با توجه به وضعیت‌های احراز هویت جاری هویت‌های متمایز کننده شناخته شده درون برنامه کاربردی کارت جاری، یک AR در ACL قابل اعمال برای آن هدف، مربوط به آن عمل وجود داشته باشد با یک وضعیت امنیتی مربوط به آن عمل که به TRUE ارزیابی می‌شود. یک AR فقط برای درخواست‌های عمل خود و فقط هنگامی که درخواست، شامل هدفی است که به ACL آن مربوط است، قابل اعمال می‌باشد.

یک ACL بخشی از هدفی است که به آن اعمال می‌شود و خود، هدف جداگانه‌ای نیست. بنابراین، هنگامی‌که یک عمل خدمت مجوزدهی در یک AR ظاهر می‌شود، AR کنترل می‌کند که آیا آن عمل مجاز است که روی ACL که در آن وجود دارد، اجرا شود. دسترسی کنترلی ARها به یک ACL، درون خود ACL وجود دارد.

به وسیله شامل کردن یک هویت متمایز کننده در یک وضعیت امنیتی، ممکن است که احراز هویت آن هویت متمایز کننده، برای مجوزدهی به عمل مربوط به آن در یک قاعده دسترسی، ضروری شود. اگر هویت متمایز کننده، با پروتکل احراز هویتی که کلیدهای جلسه را تولید می‌کند، مرتبط باشد، ممکن است که شامل کردن آن در وضعیت امنیتی، برقرار کردن یک زمینه امنیتی را ضروری نماید.

۶ دسترسی به خدمت برنامه کاربردی کارت

۶-۱ کلیات

این بند، آن نقاط ورودی روی واسط برنامه کاربردی این استاندارد را تعریف می‌کند که به وسیله آن‌ها یک برنامه کاربردی سرویس‌گیرنده می‌تواند یک برنامه کاربردی کارت خاص را بیابد و کانال‌های ارتباطی را با آن برقرار نماید. یک برنامه کاربردی سرویس‌گیرنده مجاز است که به بیش از یک برنامه کاربردی کارت، متصل شود.

۶-۲ Initialize

۶-۲-۱ هدف

این نقطه ورودی باید لایه این استاندارد شامل اتصال به سایر اجزاء پشته پروتکل استاندارد ملی ایران شماره ۱۶۳۸۶را مقداردهی اولیه کند. خدمات برنامه کاربردی کارت باید تنها پس از یک فراخوانی موفقیت‌آمیز این نقطه ورودی، روی واسط برنامه کاربردی این استاندارد در دسترس برنامه کاربردی سرویس‌گیرنده باشند.

وضعیت مقداردهی اولیه (متغیرها، مقدارهای احتمالی، منابع، پشتیبانی از خدمات برنامه کاربردی کارت) مربوط به پیاده‌سازی واسط برنامه کاربردی این استاندارد، خارج از دامنه این استاندارد است.

۶-۲-۲ نقطه ورودی

```
OUT ReturnCode Initialize(
);
```

۶-۲-۳ پارامترها

هیچ

۶-۲-۴ پیش‌نیازها

هیچ

۶-۲-۵ کدهای بازگشتی

API_OK

API_COMMUNICATION_FAILURE
API_INCORRECT_PARAMETER

۶-۲-۶ تاثیر روی وضعیت جاری

به محض اتمام موفقیت‌آمیز، لایه این استاندارد باید مقداردهی اولیه شود.

۶-۳ Terminate

۶-۳-۱ هدف

این نقطه ورودی باید به دسترسی به خدمات برنامه کاربردی کارت ارائه شده روی واسط برنامه کاربردی این استاندارد، پایان دهد. هر اتصالی که در حال حاضر بین برنامه کاربردی سرویس‌گیرنده و هر برنامه کاربردی کارت، برقرار باشد باید قطع شود. پیش از آنکه هر خدمت برنامه کاربردی روی واسط برنامه کاربردی این استاندارد برای برنامه کاربردی سرویس‌گیرنده، در دسترس باشد، نقطه ورودی مقداردهی اولیه باید فراخوانی شود.

۶-۳-۲ نقطه ورودی

OUT ReturnCode **Terminate(**
);

۶-۳-۳ پارامترها

هیچ

۶-۳-۴ پیش‌نیازها

هیچ

۶-۳-۵ کدهای بازگشتی

API_OK
API_WARNING_CONNECTION_DISCONNECTED
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_COMMUNICATION_FAILURE

۶-۳-۶ تاثیر روی وضعیت جاری

کد بازگشتی API_WARNING_CONNECTION_DISCONNECTED یک اتمام موفقیت‌آمیز درخواست است با اثر جانبی قطع کردن حداقل یک اتصال فعال.

کد بازگشتی API_COMMUNICATION_FAILURE یک اتمام ناموفق این عمل است. در این مورد، وضعیت اتصال‌های موجود، تعریف نشده، می‌باشد.

۶-۴ CardApplicationPath

۶-۴-۱ هدف

این نقطه ورودی باید مسیرهای برنامه کاربردی کارت از برنامه کاربردی سرویس‌گیرنده به یک برنامه کاربردی نامگذاری شده را تعیین کند.

۶-۴-۲ نقطه ورودی

OUT ReturnCode **CardApplicationPath(**

```
IN CardApplicationPath cardAppPathRequest,
OUT CardApplicationPathSet cardAppPathResultSet
);
```

۶-۴-۳ پارامترها

cardAppPathRequest	ترتیبی از نقاط پایانی پروتکل که نشان‌دهنده بخش نهایی یک مسیر برنامه کاربردی کارت از برنامه کاربردی سرویس‌گیرنده به آن برنامه کاربردی کارت است، که AID مربوط به آن، یک شناسانه میزبان یا پایانه و به طور اختیاری یک نام یا شناسانه IFD را شامل می‌شود. ساختار خاص این بخش نهایی در استاندارد IETF RFC 2141 بیان شده است.
cardAppPathResultSet	مجموعه‌ای از مسیرهای برنامه کاربردی کارت بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت نامگذاری شده، که هر یک شامل مقدار cardAppPathRequest به عنوان بخش نهایی خود است.

۶-۴-۴ پیش‌نیازها

هیچ

۶-۴-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_TOO_MANY_RESULTS
API_COMMUNICATION_FAILURE
```

۶-۴-۶ تاثیر روی وضعیت جاری

هیچ

۷ خدمت اتصال

۷-۱ کلیات

این بند، عمل‌هایی را تعیین می‌کند که از طریق آن‌ها یک برنامه کاربردی سرویس‌گیرنده می‌تواند به یک برنامه کاربردی کارت، متصل شود یا ارتباط با آن را قطع نماید و یک جلسه را بر پایه یک اتصال، آغاز کند یا پایان دهد.

۷-۲ CardApplicationConnect

۷-۲-۱ هدف

این عمل، باید یک اتصال احراز هویت نشده را بین برنامه کاربردی سرویس‌گیرنده و یک برنامه کاربردی کارت، برقرار کند.

۷-۲-۲ عمل

```
OUT ReturnCode CardApplicationConnect(
IN CardApplicationPath cardApplicationPath,
IN BOOLEAN exclusiveUse,
OUT ConnectionHandle connectionHandle
);
```

۷-۲-۳ پارامترها

cardApplicationPath	مسیر برنامه کاربردی کارت از برنامه کاربردی سرویس‌گیرنده تا برنامه کاربردی کارت
exclusiveUse	اگر TRUE باشد، به لایه این استاندارد نشان می‌دهد که اگر در حال حاضر، اتصال دیگری با برنامه کاربردی کارت وجود ندارد، این اتصال باید باز شود و تا زمانی که این اتصال بسته شود، نباید هیچ اتصال دیگری با برنامه کاربردی کارت، برقرار شود؛ اگر FALSE باشد، مجاز است که چند اتصال همزمان به این برنامه کاربردی کارت، باز شوند
connectionHandle	ارجاع غیرشفاف[1] برای استفاده در واسط برنامه کاربردی این استاندارد در درخواست‌های عمل

۷-۲-۴ پیش‌نیازها

هیچ

۷-۲-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_EXCLUSIVE_NOT_AVAILABLE
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۷-۲-۶ تاثیر روی وضعیت جاری

به محض اتمام موفقیت‌آمیز، برنامه کاربردی کارت نامبرده شده در مسیر برنامه کاربردی کارت، برنامه کاربردی کارت جاری است و برنامه کاربردی کارت، با اداره کردن اتصال، مرتبط است.

۷-۳ CardApplicationDisconnect

۷-۳-۱ هدف

این عمل، باید به یک اتصال بین برنامه کاربردی سرویس‌گیرنده و یک برنامه کاربردی کارت، خاتمه دهد.

۷-۳-۲ عمل

```
OUT ReturnCode CardApplicationDisconnect(
IN ConnectionHandle connectionHandle,
IN ReaderAction action
);
```

۷-۳-۳ پارامترها

connectionHandle	اداره کردن اتصال
action	آرگومان اختیاری نشان‌دهنده یک عمل IFD مورد درخواست برای راه‌اندازی مجدد یا خاموش کردن IFD یا بیرون راندن یا ضبط کردن ICC

1 - Opaque Reference

۷-۳-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۷-۳-۵ کدهای بازگشتی

```
API_OK
API_WARNING_SESSION_ENDED
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۷-۳-۶ تاثیر روی وضعیت جاری

کد بازگشتی API_WARNING_SESSION_ENDED نشان‌دهنده یک اتمام موفقیت‌آمیز برای این عمل، محسوب می‌شود که دارای اثر جانبی پایان دادن به حداقل یک جلسه فعال در اتصال می‌باشد.

۷-۴ CardApplicationStartSession

۷-۴-۱ هدف

این عمل باید یک جلسه را بین یک برنامه کاربردی سرویس‌گیرنده و یک برنامه کاربردی کارت، برقرار نماید. به ازای هر اداره کردن اتصال، مجاز است که فقط یک جلسه برقرار شود.

یک اداره کردن اتصال به یک برنامه کاربردی کارت ثانویه، به عنوان مثال، یک پیمانه دسترسی امنیتی (SAM)، مجاز است که برای انجام عملیات رمزنگاشتی از طرف برنامه کاربردی سرویس‌گیرنده، در برقراری جلسه، ارائه شود.

درخواست‌های پی‌درپی این عمل برای اداره کردن اتصال یکسان و هویت متمایزکننده نامگذاری شده، ممکن است برای تکمیل پروتکل به صورتی که در پیوست الف تعریف شده، ضروری باشد.

۷-۴-۲ عمل

```
OUT ReturnCode CardApplicationStartSession(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName didName,
IN/OUT DIDAuthenticationData authenticationProtocolData,
IN ConnectionHandle samConnectionHandle
);
```

۷-۴-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایزکننده نامگذاری شده
didName	نام هویت متمایزکننده حاوی پروتکل احراز هویت که برای برقراری جلسه بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، اجرا می‌شود
AuthenticationProtocolData	داده‌های تبادل شده مطابق با پروتکل
samConnectionHandle	آرگومان اختیاری برای یک برنامه کاربردی کارت، که می‌تواند به وسیله لایه این استاندارد، استفاده شود

۷-۴-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۷-۴-۵ کدهای بازگشتی

API_OK
API_NEXT_REQUEST
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_ACTIVE_SESSION
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۷-۴-۶ تاثیر روی وضعیت جاری

به پیوست الف، مراجعه شود.

۷-۵ CardApplicationEndSession

۷-۵-۱ هدف

این عمل باید یک جلسه بین یک برنامه کاربردی سرویس‌گیرنده و یک برنامه کاربردی کارت را خاتمه دهد.

۷-۵-۲ عمل

OUT ReturnCode **CardApplicationEndSession(**
IN ConnectionHandle **connectionHandle**
);

۷-۵-۳ پارامترها

connectionHandle	اداره کردن اتصال

۷-۵-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۷-۵-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NO_ACTIVE_SESSION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE

۷-۵-۶ تاثیر روی وضعیت جاری

به محض اتمام موفقیت‌آمیز، مشخصات امنیتی اتصال، به مشخصات امنیتی ذاتی اتصال، برگردانده می‌شود.

۸ خدمت برنامه کاربردی کارت

۸-۱ کلیات

این بند، عمل‌هایی که به وسیله آن‌ها برنامه‌های کاربردی کارت می‌تواند ایجاد و حذف شود و به وسیله آن‌ها خدمات درون برنامه‌های کاربردی کارت، می‌تواند ایجاد و حذف شود را تعریف می‌کند. عمل

CardApplicationCreate() قابلیت مدیریتی[1] را برای استقرار هستار برنامه کاربردی کارت، فراهم می‌کند. کد اجرایی برنامه کاربردی کارت، به وسیله عمل CardApplicationServiceLoad() بارگذاری می‌شود.

۸-۲ CardApplicationList

۸-۲-۱ هدف

این عمل باید نام برنامه‌های کاربردی کارت فهرست شده در برنامه کاربردی کارت آلفا را برگرداند. قاعده دسترسی قابل استفاده برای این عمل، باید در فهرست کنترل دسترسی برنامه کاربردی کارت جاری، وجود داشته باشد.

۸-۲-۲ عمل

```
OUT ReturnCode CardApplicationList(
IN ConnectionHandle connectionHandle,
OUT CardApplicationNameList cardApplicationNameList
);
```

۸-۲-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationNameList	فهرست نام برنامه‌های کاربردی کارت

۸-۲-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۸-۲-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۸-۲-۶ تاثیر روی وضعیت جاری

هیچ

۸-۳ CardApplicationCreate

۸-۳-۱ هدف

این عمل باید یک برنامه کاربردی کارت جدید، ایجاد کند.

۸-۳-۲ عمل

```
OUT ReturnCode CardApplicationCreate(
IN ConnectionHandle connectionHandle,
IN CardApplicationName cardApplicationName,
IN AccessControlList cardApplicationACL
);
```

۸-۳-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationName	نام برنامه کاربردی کارتی که قرار است ایجاد شود

1 - Administrative functionality

cardApplicationACL فهرست کنترل دسترسی برای این برنامه کاربردی کارت جدید

۸-۳-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است. برنامه کاربردی کارت جاری، باید برنامه کاربردی کارت آلفا باشد.

۸-۳-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAME_EXISTS
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_PREREQUISITE_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۸-۳-۶ تاثیر روی وضعیت جاری

هیچ

۸-۴ CardApplicationDelete

۸-۴-۱ هدف

این عمل باید برنامه کاربردی کارت نامگذاری شده، شامل تمام خدمات، مجموعه دادهها و هویت‌های متمایزکننده آن را حذف کند.

۸-۴-۲ عمل

OUT ReturnCode **CardApplicationDelete(**
IN ConnectionHandle **connectionHandle,**
IN CardApplicationName **cardApplicationName**
);

۸-۴-۳ پارامترها

connectionHandle اداره کردن اتصال

cardApplicationName نام برنامه کاربردی کارتی که قرار است حذف شود

۸-۴-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است. برنامه کاربردی کارت جاری، باید برنامه کاربردی کارت آلفا باشد.

۸-۴-۵ کدهای بازگشتی

API_OK
API_WARNING_CONNECTION_DISCONNECTED
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_PREREQUISITE_NOT_SATISFIED
API_COMMUNICATION_FAILURE

۸-۴-۶ تاثیر روی وضعیت جاری

به محض اتمام موفقیت‌آمیز، تمام اتصال‌ها به برنامه کاربردی کارت نامگذاری‌شده، باید قطع شوند.

۸-۵ CardApplicationServiceList

۸-۵-۱ هدف

این عمل باید خدمات برنامه کاربردی کارت درون برنامه کاربردی کارت جاری را فهرست کند.

۸-۵-۲ عمل

```
OUT ReturnCode CardApplicationServiceList(
IN ConnectionHandle connectionHandle,
OUT CardApplicationServiceNameList cardApplicationServiceNameList
);
```

۸-۵-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationServiceNameList	فهرست نام خدمات برنامه کاربردی کارت در برنامه کاربردی کارت جاری

۸-۵-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۸-۵-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۸-۵-۶ تاثیر روی وضعیت جاری

هیچ

۸-۶ CardApplicationServiceCreate

۸-۶-۱ هدف

این عمل باید در برنامه کاربردی کارت جاری، یک خدمت برنامه کاربردی کارت جدید، ایجاد کند.

۸-۶-۲ عمل

```
OUT ReturnCode CardApplicationServiceCreate(
IN ConnectionHandle connectionHandle,
IN CardApplicationServiceName cardApplicationServiceName
);
```

۸-۶-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationServiceName	نام خدمت برنامه کاربردی کارتی که قرار است ایجاد شود

۸-۶-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۸-۶-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAME_EXISTS
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
```

API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۸-۶-۶ تاثیر روی وضعیت جاری

هیچ

[

۸-۷ CardApplicationServiceLoad

۸-۷-۱ هدف

این عمل باید کد اجرایی را که یک خدمت برنامه کاربردی کارت را درون برنامه کاربردی کارت جاری، پیاده‌سازی می‌کند، بارگذاری نماید.

۸-۷-۲ عمل

```
OUT ReturnCode CardApplicationServiceLoad(
IN ConnectionHandle connectionHandle,
IN CardApplicationServiceName cardApplicationServiceName,
IN CardApplicationServiceLoadPackage code
);
```

۸-۷-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationServiceName	نام خدمت برنامه کاربردی کارتی که به وسیله کد، پیاده‌سازی شده است
code	بسته بارگذاری خدمت

۸-۷-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۸-۷-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۸-۷-۶ تاثیر روی وضعیت جاری

هیچ

۸-۸ CardApplicationServiceDelete]

۸-۸-۱ هدف

این عمل باید خدمت برنامه کاربردی کارت نامگذاری شده، شامل کدی که آن را پیاده‌سازی می‌کند، را از برنامه کاربردی کارت جاری، حذف کند.

۸-۸-۲ عمل

```
OUT ReturnCode CardApplicationServiceDelete(
IN ConnectionHandle connectionHandle,
IN CardApplicationServiceName cardApplicationServiceName
);
```

۸-۸-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationServiceName	نام خدمت برنامه کاربردی کارتی که قرار است حذف شود

۸-۸-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۸-۸-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE

۸-۸-۶ تاثیر روی وضعیت جاری

هیچ

۸-۹ CardApplicationServiceDescribe

۸-۹-۱ هدف

این عمل باید یک URL[1] یا توصیف کامل خدمت برنامه کاربردی کارت نامگذاری شده را برگرداند. توصیف به دست آمده، به برنامه کاربردی سرویس‌گیرنده، امکان می‌دهد که عملکردی فراتر از مجموعه استاندارد شده خدمات برنامه کاربردی کارت توصیف شده در این استاندارد، را کشف نماید.

۸-۹-۲ عمل

OUT ReturnCode **CardApplicationServiceDescribe(**
IN ConnectionHandle **connectionHandle**,
IN CardApplicationServiceName **cardApplicationServiceName,**
OUT CardApplicationServiceDescription **serviceDescription**
);

۸-۹-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationServiceName	نام خدمت برنامه کاربردی کارتی که قرار است توصیف شود
serviceDescription	URI، URL، یا توصیف کامل (ساختار) خدمت برنامه کاربردی کارت نامگذاری شده (مجاز است که به عنوان یک اساس برای ترکیب صحیح برای درخواست‌های SAL در این خدمت برنامه کاربردی کارت نامگذاری شده، به وسیله عمل ExecuteAction استفاده شود)

۸-۹-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۸-۹-۵ کدهای بازگشتی

API_OK

1- Uniform Resource Locator

API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE

۸-۹-۶ تاثیر روی وضعیت جاری

هیچ

۸-۱۰ ExecuteAction

۸-۱۰-۱ هدف

این عمل باید دسترسی به یک عمل در یک خدمت برنامه کاربردی کارت که در این استاندارد، تعریف نشده، را درخواست نماید.

عملکرد عمل‌های مورد دسترسی به وسیله این عمل، خارج از دامنه (این استاندارد) است.

۸-۱۰-۲ عمل

OUT ReturnCode **ExecuteAction(**
IN ConnectionHandle **connectionHandle**,
IN CardApplicationServiceName **cardApplicationServiceName**,
IN ActionName **actionName**,
IN ExecuteActionRequest **request,**
OUT ExecuteActionConfirmation **confirmation**
);

۸-۱۰-۳ پارامترها

connectionHandle	اداره کردن اتصال
cardApplicationServiceName	نام خدمت برنامه کاربردی کارت حاوی عملی که قرار است درخواست شود
actionName	نام عملی که قرار است درخواست شود
request	داده‌های درخواست عمل
Confirmation	داده‌های تایید عمل

۸-۱۰-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۸-۱۰-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۸-۱۰-۶ تاثیر روی وضعیت جاری

هیچ

۹ خدمت داده‌های نام‌گذاری شده

۹-۱ کلیات

این بند، عمل‌هایی که به وسیله آن‌ها داده‌ها می‌توانند درون یک برنامه کاربردی کارت، مدیریت شوند را تعریف می-کند.

۹-۲ DataSetList

۹-۲-۱ هدف

این عمل، باید نام مجموعه‌داده‌های تعریف شده در برنامه کاربردی کارت جاری را فهرست کند.

۹-۲-۲ عمل

```
OUT ReturnCode DataSetList(
IN ConnectionHandle connectionHandle,
OUT DataSetNameList dataSetNameList
);
```

۹-۲-۳ پارامترها

connectionHandle	اداره کردن اتصال
dataSetNameList	فهرست نام مجموعه داده‌های تعریف‌شده درون برنامه کاربردی کارت جاری

۹-۲-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۹-۲-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۹-۲-۶ تاثیر روی وضعیت جاری

هیچ

۹-۳ DataSetCreate

۹-۳-۱ هدف

این عمل، باید در برنامه کاربردی کارت جاری، یک مجموعه داده جدید، ایجاد کند.

۹-۳-۲ عمل

```
OUT ReturnCode DataSetCreate(
IN ConnectionHandle connectionHandle,
IN DataSetName dataSetName,
IN AccessControlList dataSetACL
);
```

۹-۳-۳ پارامترها

connectionHandle	اداره کردن اتصال

dataSetName	نام مجموعه داده
dataSetACL	فهرست کنترل دسترسی متعلق به مجموعه‌داده

۹-۳-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۹-۳-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAME_EXISTS
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۹-۳-۶ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز، در برنامه کاربردی کارت جاری، مجموعه داده جدید، مجموعه داده جاری می‌شود.

۹-۴ DataSetSelect

۹-۴-۱ هدف

این عمل، باید در برنامه کاربردی کارت جاری، مجموعه داده نامگذاری شده را انتخاب کند.

۹-۴-۲ عمل

OUT ReturnCode **DataSetSelect(**
IN ConnectionHandle **connectionHandle**,
IN DataSetName **dataSetName**
);

۹-۴-۳ پارامترها

connectionHandle	اداره کردن اتصال
dataSetName	نام مجموعه داده

۹-۴-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۹-۴-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE

۹-۴-۶ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز، در برنامه کاربردی کارت جاری، مجموعه داده انتخاب شده، مجموعه داده جاری، می-شود.

۹-۵ DataSetDelete

۹-۵-۱ هدف

این عمل، باید مجموعه داده نام‌گذاری‌شده درون برنامه کاربردی کارت جاری، شامل تمام DSIهای درون آن مجموعه داده را حذف کند.

۹-۵-۲ عمل

```
OUT ReturnCode DataSetDelete(
IN ConnectionHandle connectionHandle,
IN DataSetName dataSetName
);
```

۹-۵-۳ پارامترها

connectionHandle	اداره کردن اتصال
dataSetName	نام مجموعه داده

۹-۵-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۹-۵-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۹-۵-۶ تاثیر روی وضعیت جاری

اگر در زمان درخواست این عمل، مجموعه داده جاری، مجموعه داده‌ای باشد که قرار است حذف شود، پس از تایید عمل، مجموعه داده جاری، تعریف نشده، خواهد بود.

۹-۶ DSIList

۹-۶-۱ هدف

این عمل، باید نام DSIهای درون مجموعه داده جاری را فهرست کند.

۹-۶-۲ عمل

```
OUT ReturnCode DSIList(
IN ConnectionHandle connectionHandle,
OUT DSINameList dsiNameList
);
```

۹-۶-۳ پارامترها

connectionHandle	اداره کردن اتصال
dsiNameList	فهرست نام DSIهای درون مجموعه داده جاری

۹-۶-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است. یک مجموعه داده باید انتخاب شده باشد.

۹-۶-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_PREREQUISITE_NOT_SATISFIED
API_COMMUNICATION_FAILURE

۹-۶-۶ تاثیر روی وضعیت جاری

هیچ

۹-۷ DSICreate

۹-۷-۱ هدف

این عمل، باید در مجموعه داده جاری، یک DSI جدید، ایجاد کند.

۹-۷-۲ عمل

OUT ReturnCode **DSICreate(**
IN ConnectionHandle **connectionHandle**,
IN DSIName **dsiName,**
IN DSIContent **dsiContent**
);

۹-۷-۳ پارامترها

connectionHandle	اداره کردن اتصال
dsiName	نام DSIی که قرار است در مجموعه داده جاری، ایجاد شود
dsiContent	محتوایی که قرار است در DSI، ذخیره شود

۹-۷-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است. یک مجموعه داده باید انتخاب شده باشد.

۹-۷-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAME_EXISTS
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_PREREQUISITE_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۹-۷-۶ تاثیر روی وضعیت جاری

هیچ

۹-۸ DSIDelete

۹-۸-۱ هدف

این عمل، باید DSI نام‌گذاری‌شده را از مجموعه داده جاری، حذف کند.

۹-۸-۲ عمل

OUT ReturnCode **DSIDelete(**
IN ConnectionHandle **connectionHandle**,
IN DSIName **dsiName**

```
);
```

۹-۸-۳ پارامترها

connectionHandle	اداره کردن اتصال
dsiName	نام DSI در مجموعه داده جاری، که قرار است حذف شود

۹-۸-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است. یک مجموعه داده باید انتخاب شده باشد.

۹-۸-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_PREREQUISITE_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۹-۸-۶ تاثیر روی وضعیت جاری

هیچ

۹-۹ DSIWrite

۹-۹-۱ هدف

این عمل، باید محتویات DSI نامگذاری شده در مجموعه داده جاری را با داده‌های فراهم شده، جایگزین کند.

۹-۹-۲ عمل

```
OUT ReturnCode DSIWrite(
IN ConnectionHandle connectionHandle,
IN DSIName dsiName,
IN DSIContent dsiContent
);
```

۹-۹-۳ پارامترها

connectionHandle	اداره کردن اتصال
dsiName	نام DSI در مجموعه داده جاری، که قرار است نوشته شود
dsiContent	محتوایی که قرار است درون DSI، ذخیره شود

۹-۹-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است. یک مجموعه داده باید انتخاب شده باشد.

۹-۹-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_PREREQUISITE_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۹-۹-۶ تاثیر روی وضعیت جاری

هیچ

۹-۱۰ DSIRead

۹-۱۰-۱ هدف

این عمل، باید محتوای DSI نامگذاری شده درون مجموعه داده جاری را برگرداند.

۹-۱۰-۲ عمل

```
OUT ReturnCode DSIRead(
IN ConnectionHandle connectionHandle,
IN DSIName dsiName,
OUT DSIContent dsiContent
);
```

۹-۱۰-۳ پارامترها

connectionHandle	اداره کردن اتصال
dsiName	نام DSI در مجموعه داده جاری، که قرار است خوانده شود
dsiContent	محتوای DSI

۹-۱۰-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است. یک مجموعه داده باید انتخاب شده باشد.

۹-۱۰-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_PREREQUISITE_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۹-۱۰-۶ تاثیر روی وضعیت جاری

هیچ

۱۰ خدمت رمزنگاشتی

۱۰-۱ کلیات

این بند، عمل‌هایی را که به وسیله آن‌ها عملیات رمزنگاشتی، انجام می‌شوند، تعریف می‌کند.

الگوریتم‌های استفاده شده در اجرای این عمل‌ها به پروتکل‌های احراز هویت، بستگی دارد، همانگونه که در پیوست الف، تعریف شده است.

۱۰-۲ Encipher

۱۰-۲-۱ هدف

این عمل باید داده‌های ارائه شده را مطابق با عمل رمزنگاشتی مشخص شده در پروتکل احراز هویت درون هویت متمایزکننده نامگذاری شده، رمزگذاری[1] کند.

۱۰-۲-۲ عمل

```
OUT ReturnCode Encipher(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName didName,
IN CipherBuffer plainText,
OUT CipherBuffer cipherText
);
```

۱۰-۲-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایز کننده نام‌گذاری‌شده
didName	نام هویت متمایز کننده فراهم کننده الگوریتم رمزنگاشتی
plainText	داده‌هایی که قرار است رمزنگاشتی شود
cipherText	متن رمزنگاشتی شده

۱۰-۲-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۰-۲-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۱۰-۲-۶ تاثیر روی وضعیت جاری

هیچ

۱۰-۳ Decipher

۱۰-۳-۱ هدف

این عمل باید داده‌های ارائه شده را مطابق با عمل رمزنگاشتی پروتکل درون هویت متمایزکننده نامگذاری‌شده، رمزگشایی[2] کند.

---

1 - Encipher
2 - Decipher

۱۰-۳-۲ عمل

```
OUT ReturnCode Decipher(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName didName,
IN CipherBuffer cipherText,
OUT CipherBuffer plainText
);
```

۱۰-۳-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایز کننده نامگذاری شده
didName	نام هویت متمایز کننده فراهم کننده الگوریتم رمزنگاشتی
cipherText	داده‌هایی که قرار است رمزگشایی شود
plainText	متن رمزنگاشتی‌نشده

۱۰-۳-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۰-۳-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۱۰-۳-۶ تاثیر روی وضعیت جاری

هیچ

۱۰-۴ GetRandom

۱۰-۴-۱ هدف

این عمل باید یک مقدار تصادفی تولید شده مطابق با پروتکل هویت متمایز کننده نامگذاری شده را برگرداند.

۱۰-۴-۲ عمل

```
OUT ReturnCode GetRandom(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName didName,
OUT RandomDataBuffer random
);
```

۱۰-۴-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایز کننده نامگذاری شده

didName	نام هویت متمایز کننده‌ای که خصوصیات آن مقدار تصادفی را تعریف می‌کند
Random	مقدار تصادفی

۱۰-۴-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۰-۴-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۱۰-۴-۶ تاثیر روی وضعیت جاری

هیچ

۱۰-۵ Hash

۱۰-۵-۱ هدف

این عمل باید پیام ارائه شده را مطابق با پروتکل احراز هویت و علامت گذار هویت متمایز کننده نام‌گذاری‌شده، درهم‌سازی[1] کند.

۱۰-۵-۲ عمل

OUT ReturnCode **Hash(**
IN ConnectionHandle **connectionHandle**,
IN DIDScope **didScope,**
IN DIDName **didName,**
IN MessageBuffer **message,**
OUT HashBuffer **hash**
);

۱۰-۵-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایز کننده نامگذاری شده
didName	نام هویت متمایز کننده مورد نظر برای استفاده در تولید درهم‌سازی
message	پیامی که قرار است درهم‌سازی شود
hash	نتیجه درهم‌سازی پیام ارائه شده

۱۰-۵-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

1 - Hash

۱۰-۵-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۱۰-۵-۶ تاثیر روی وضعیت جاری

هیچ

۱۰-۶ Sign

۱۰-۶-۱ هدف

این عمل باید پیام ارائه شده را مطابق با پروتکل احراز هویت و علامت گذار هویت متمایز کننده نامگذاری شده، امضاء کند.

۱۰-۶-۲ عمل

OUT ReturnCode **Sign(**
IN ConnectionHandle **connectionHandle**,
IN DIDScope **didScope,**
IN DIDName **didName,**
IN MessageBuffer **message,**
OUT SignatureBuffer **signature**
);

۱۰-۶-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایز کننده نامگذاری شده
didName	نام هویت متمایز کننده مورد نظر برای استفاده در تولید آن امضای دیجیتال
message	پیامی که قرار است امضاء شود
signature	امضای پیام ارائه شده

۱۰-۶-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۰-۶-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۱۰-۶-۶ تاثیر روی وضعیت جاری

هیچ

۱۰-۷ VerifySignature

۱۰-۷-۱ هدف

این عمل باید با استفاده از پروتکل احراز هویت و علامت گذار هویت متمایز کننده نامگذاری شده، تایید یک امضای دیجیتال را انجام دهد.

۱۰-۷-۲ عمل

```
OUT ReturnCode VerifySignature(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName didName,
IN SignatureBuffer signature,
IN MessageBuffer message
);
```

۱۰-۷-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایز کننده نامگذاری شده
didName	نام هویت متمایز کننده مورد نظر برای استفاده در تایید آن امضاء
signature	امضایی‌که قرار است احراز شود
message	داده‌های امضاء شده

۱۰-۷-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۰-۷-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_INVALID_SIGNATURE
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۱۰-۷-۶ تاثیر روی وضعیت جاری

هیچ

۱۰-۸ VerifyCertificate

۱۰-۸-۱ هدف

این عمل باید با استفاده از پروتکل احراز هویت و علامت گذار هویت متمایز کننده نامگذاری شده، تایید یک گواهی دیجیتال را انجام دهد.

اگر گواهی ارائه شده، معتبر باشد ولی به وسیله یک هویت متمایز کننده روی کارت، امضاء نشده باشد، درخواست-های بعدی می‌توانند با ارائه کردن گواهی امضاء کننده گواهی قبلی، ایجاد شوند. تا زمانی که گواهی ارائه شده، به وسیله یک هویت متمایز کننده روی کارت، امضاء شده باشد، تکرار این کار، زنجیره اعتماد از هویت متمایز کننده روی کارت تا صاحب گواهی اصلی ارائه شده را تایید می‌نماید.

۱۰-۸-۲ عمل

```
OUT ReturnCode VerifyCertificate(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName rootCert,
IN CertificateType certificateType,
IN Certificate certificate
);
```

۱۰-۸-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایزکننده نام‌گذاری‌شده
rootCert	نام هویت متمایز کننده اختیاری مورد نظر برای تایید امضای روی گواهی
certificateType	قالب گواهی، شامل یک اندیس و شناسانه شیء
certificate	گواهی که قرار است تایید شود

۱۰-۸-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۰-۸-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_INVALID_KEY
API_INVALID_SIGNATURE
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۱۰-۸-۶ تاثیر روی وضعیت جاری

هیچ

۱۱ خدمت Differential-identity

۱۱-۱ کلیات

این بند، عمل‌هایی که به وسیله آن‌ها، هویت‌های متمایز کننده می‌توانند درون یک برنامه کاربردی کارت، ایجاد و مدیریت شوند را تعریف می‌کند.

۱۱-۲ DIDList

۱۱-۲-۱ هدف

این عمل باید نام‌های هویت‌های متمایز کننده تعریف شده درون برنامه کاربردی کارت جاری را فهرست کند.

۱۱-۲-۲ عمل

```
OUT ReturnCode DIDList(
IN ConnectionHandle connectionHandle,
IN DIDQualifier filter,
OUT DIDNameList didNameList
);
```

۱۱-۲-۳ پارامترها

connectionHandle	اداره کردن اتصال
filter	didNameList فقط شامل نام‌های آن DIDهایی است که دارای فیلدهای توصیف‌کننده دقیقا مطابق با فیلتر باشند، یک فیلتر Null با تمام فیلدهای توصیف‌کننده و تمام مقادیر، مطابقت دارد.
didNameList	فهرست نام‌های هویت‌های متمایزکننده درون برنامه کاربردی کارت جاری

۱۱-۲-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۱-۲-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۱۱-۲-۶ تاثیر روی وضعیت جاری

هیچ

۱۱-۳ DIDCreate

۱۱-۳-۱ هدف

این عمل باید درون برنامه کاربردی کارت جاری، یک هویت متمایزکننده جدید، ایجاد کند.

قالب پارامتر didUpdateData برای هر پروتکل احراز هویت، در پیوست الف، تعریف شده است.

۱۱-۳-۲ عمل

```
OUT ReturnCode DIDCreate(
```

```
IN ConnectionHandle connectionHandle,
IN DIDName didName,
IN ObjectIdentifier authProtocolOID,
IN DIDUpdateData didUpdateData,
IN AccessControlList didACL
);
```

۱۱-۳-۳ پارامترها

connectionHandle	اداره کردن اتصال
didName	نام هویت متمایز کننده‌ای که قرار است ایجاد شود
authProtocolOID	شناسانه شیء متعلق به یک پروتکل احراز هویت شناسایی شده در پیوست الف یا ثبت شده در جای دیگری در استاندارد ملی ایران شماره ۱۶۳۸۶
didUpdataData	ساختار شامل پارامترهایی برای ایجاد هویت متمایزکننده، مربوط به پروتکل احراز هویت مشخص شده به وسیله authProtocolOID
didACL	فهرست کنترل دسترسی حاکم بر دسترسی به هویت متمایز کننده

۱۱-۳-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۱-۳-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAME_EXISTS
API_PROTOCOL_NOT_RECOGNIZED
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۱۱-۳-۶ تاثیر روی وضعیت جاری

هیچ

۱۱-۴ DIDGet

۱۱-۴-۱ هدف

این عمل باید اطلاعات یک هویت متمایز کننده شناخته شده درون برنامه کاربردی کارت جاری، را برگرداند.

داده‌های برگردانده شده باید شامل فیلد پروتکل، تمام اجزای قابل صدور علامت‌گذار مطابق تعریف پروتکل در پیوست الف و در صورت وجود، فیلد توصیف کننده، باشد.

۱۱-۴-۲ عمل

```
OUT ReturnCode DIDGet(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName didName,
OUT DIDStructure didStructure
);
```

۱۱-۴-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایزکننده نام‌گذاری‌شده
didName	نام هویت متمایزکننده‌ای که اطلاعات کشف مربوط به آن، درخواست شده است
didStructure	ساختار شامل پارامترهای قابل کشف آن هویت متمایز کننده

۱۱-۴-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۱-۴-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۱۱-۴-۶ تاثیر روی وضعیت جاری

هیچ

۱۱-۵ DIDUpdate

۱۱-۵-۱ هدف

این عمل باید علامت‌گذار جدیدی را برای هویت متمایز کننده نامگذاری شده که درون برنامه کاربردی کارت جاری به طور مناسب برای پروتکل موجود آن تعریف شده، ذخیره یا تولید کند.

آرگومان didUpdateData شامل علامت گذار جدیدی که قرار است ذخیره شود یا پارامترهایی که قرار است مطابق پروتکل هویت متمایز کننده، در تولید یک علامت‌گذار جدید، استفاده شود، می‌باشد. آرگومان didUpdateData ممکن است که شامل اطلاعات متفاوت هویت متمایز کننده نیز باشد.

۱۱-۵-۲ عمل

```
OUT ReturnCode DIDUpdate(
IN ConnectionHandle connectionHandle,
IN DIDName didName,
IN DIDUpdateData didUpdateData
);
```

۱۱-۵-۳ پارامترها

connectionHandle	اداره کردن اتصال
didName	نام هویت متمایز کننده‌ای که قرار است به‌روز رسانی شود
didUpdatData	داده‌هایی برای به‌روز رسانی یک هویت متمایز کننده

۱۱-۵-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۱-۵-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE

۱۱-۵-۶ تاثیر روی وضعیت جاری

هیچ

۱۱-۶ DIDDelete

۱۱-۶-۱ هدف

این عمل باید هویت متمایز کننده نامگذاری شده که درون برنامه کاربردی کارت جاری تعریف شده است را حذف کند.

۱۱-۶-۲ عمل

OUT ReturnCode **DIDDelete(**
IN ConnectionHandle **connectionHandle**,
IN DIDName **didName**
);

۱۱-۶-۳ پارامترها

connectionHandle	اداره کردن اتصال
didName	نام هویت متمایز کننده‌ای که قرار است از برنامه کاربردی کارت، حذف شود

۱۱-۶-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۱-۶-۵ کدهای بازگشتی

API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE

۱۱-۶-۶ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز، وضعیت احراز هویت مربوط به هویت متمایزکننده حذف شده، از وضعیت جاری، حذف می‌شود.

۱۱-۷ DIDAuthenticate

۱۱-۷-۱ هدف

این عمل باید پروتکل احراز هویت مربوط به هویت متمایز کننده نامگذاری شده که در برنامه کاربردی کارت جاری، شناخته شده است را اجرا کند.

ممکن است برای تکمیل این پروتکل به صورتی که در پیوست الف یا در جایی در استاندارد ملی ایران شماره ۱۶۳۸۶ تعریف شده، درخواست‌های پی‌درپی این عمل برای اداره کردن اتصال و هویت متمایزکننده یکسان، لازم باشد.

۱۱-۷-۲ عمل

```
OUT ReturnCode DIDAuthenticate(
IN ConnectionHandle connectionHandle,
IN DIDScope didScope,
IN DIDName didName,
IN/OUT DIDAuthenticationData authenticationProtocolData,
IN ConnectionHandle samConnectionHandle
);
```

۱۱-۷-۳ پارامترها

connectionHandle	اداره کردن اتصال
didScope	دامنه هویت متمایزکننده نام‌گذاری‌شده
didName	نام هویت متمایزکننده‌ای که قرار است احراز شود
authenticationProtocolData	داده‌های مبادله شده مطابق پروتکل
samConnectionHandle	آرگومان اختیاری برای یک برنامه کاربردی کارت که مجاز است به وسیله لایه این استاندارد، مورد استفاده قرار گیرد

۱۱-۷-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۱-۷-۵ کدهای بازگشتی

```
API_OK
API_NEXT_REQUEST
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_PROTOCOL_NOT_RECOGNIZED
API_INAPPROPRIATE_PROTOCOL_FOR_ACTION
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۱۱-۷-۶ تاثیر روی وضعیت جاری

به پیوست الف، رجوع شود.

۱۲ خدمت مجوزدهی

۱۲-۱ کلیات

این بند، عمل‌های مورد استفاده برای کشف و تغییر فهرست‌های کنترل دسترسی را تعریف می‌کند.

۱۲-۲ ACLList

۱۲-۲-۱ هدف

این عمل باید فهرست کنترل دسترسی برای هدف نامگذاری شده را برگرداند.

اگر نوع هدف، مجموعه‌داده باشد، نام این هدف باید فقط برای مجموعه داده‌های تعریف شده در برنامه کاربردی کارت جاری، به کار رود. اگر نوع هدف، هویت متمایز کننده باشد، نام این هدف باید فقط برای هویت‌های متمایز کننده تعریف شده در برنامه کاربردی کارت جاری، به کار رود.

۱۲-۲-۲ عمل

OUT ReturnCode **ACLList(**
IN ConnectionHandle **connectionHandle**,
IN TargetType **targetType**,
IN TargetName **targetName**,
OUT AccessControlList **targetACL**
);

۱۲-۲-۳ پارامترها

connectionHandle	اداره کردن اتصال
targetType	نوع هدف
targetName	نام هدف
targetACL	فهرست کنترل دسترسی

۱۲-۲-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۲-۲-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_COMMUNICATION_FAILURE
```

۱۲-۲-۶ تاثیر روی وضعیت جاری

هیچ

۱۲-۳ ACLModify

۱۲-۳-۱ هدف

این عمل باید قاعده دسترسی برای عمل نام‌گذاری‌شده را در فهرست کنترل دسترسی هدف نام‌گذاری‌شده تغییر دهد.

اگر نوع هدف، مجموعه داده باشد، نام این هدف باید فقط برای مجموعه داده‌های تعریف شده در برنامه کاربردی کارت جاری، به کار رود. اگر نوع هدف، هویت متمایز کننده باشد، نام این هدف باید فقط برای هویت‌های متمایز کننده تعریف شده در برنامه کاربردی کارت جاری، به کار رود.

۱۲-۳-۲ عمل

```
OUT ReturnCode ACLModify(
IN ConnectionHandle connectionHandle,
IN TargetType targetType,
IN TargetName targetName,
IN CardApplicationServiceName cardApplicationServiceName,
IN ActionName actionName,
IN SecurityCondition securityCondition
);
```

۱۲-۳-۳ پارامترها

connectionHandle	اداره کردن اتصال
targetType	نوع هدف
targetName	نام هدف
cardApplicationServiceName	نام خدمت برنامه کاربردی کارت حاوی عمل نام‌گذاری‌شده
actionName	نام عملی که قاعده دسترسی آن قرار است تغییر داده شود
securityCondition	عبارت کامل بولین بر حسب نام هویت‌های متمایزکننده

۱۲-۳-۴ پیش‌نیازها

یک connectionHandle معتبر، مورد انتظار است.

۱۲-۳-۵ کدهای بازگشتی

```
API_OK
API_INCORRECT_PARAMETER
API_NAMED_ENTITY_NOT_FOUND
API_NOT_INITIALIZED
API_SECURITY_CONDITION_NOT_SATISFIED
API_INSUFFICIENT_RESOURCES
API_COMMUNICATION_FAILURE
```

۱۲-۳-۶ تاثیر روی وضعیت جاری

هیچ

پیوست الف

(الزامی)

پروتکل‌های احراز هویت

الف-۱ کلیات

پروتکل مربوط به یک هویت متمایز کننده در زمان ایجاد آن هویت، باید یکی از پروتکل‌های تعریف شده در این پیوست یا ثبت شده در جایی در استاندارد ملی ایران شماره ۱۶۳۸۶ باشد. یک برنامه کاربردی کارت، نیاز نیست تا تمام پروتکل‌های تعریف شده در این پیوست را پیاده‌سازی کند.

احراز هویت مربوط به هویت‌های متمایزکننده، از طریق اجرای موفقیت‌آمیز پروتکل‌های احراز هویت تعریف شده در این پیوست، انجام می‌شود.

در شکل‌های این پیوست، به جای کدگذاری واقعی خود درخواست‌ها، توصیف‌های کلی برای درخواست‌های قرار داده شده روی واسط استاندارد ملی ایران شماره ۲ ـ ۱۶۳۸۶به وسیله لایه این استاندارد، ارائه شده است. کدگذاری واقعی درخواست‌ها، بر عهده پیاده‌سازی این استاندارد است. برخی از تنظیمات پشته استاندارد ملی ایران شماره ۱۶۳۸۶، ممکن است به پیاده‌سازی لایه GCI استاندارد ملی ایران شماره ۲ ـ ۱۶۳۸۶ نیاز نداشته باشند.

الف-۲ تعاریف مشترک

الف-۲-۱ نمادها و عملگرها

در تعریف محاسبات این پیوست، از قراردادهای زیر استفاده شده است.

=?=	یک عمل مقایسه را نشان می‌دهد
||	یک عمل الحاق را نشان می‌دهد
algorithm (input)	به کارگیری algorithm برای input را نشان می‌دهد
algorithm [key] (input)	به کارگیری algorithm با استفاده از key برای input را نشان می‌دهد
algorithm $^{-1}$ (input)	به کارگیری معکوس algorithm برای input را نشان می‌دهد

(برای یک الگوریتم رمزنگاشتی، این نماد، به کارگیری آن را برای رمزگشایی، نشان می‌دهد.)

RNG(size) به کارگیری یک مولد عدد تصادفی را به منظور تولید size بایت داده‌ها، نشان می‌دهد، در حالی که مولد مورد استفاده، خارج از دامنه کاربرد این استاندارد است

الف-۲-۲ ساختارها

عمل‌های درون API این استاندارد که از ایجاد و به‌روز رسانی هویت‌های متمایزکننده، پشتیبانی می‌کنند باید یک ساختار DIDUpdateData را به عنوان یک آرگومان ورودی برای آن عمل‌ها، فراهم نمایند. این ساختار، به صورت زیر تعریف شده است:

DIDUpdateData ::= SEQUENCE {

```
marker OCTET STRING,
qualifier DiDQualifier OPTIONAL
}
```

به طوری که marker، یک علامت‌گذار بخصوص است به همان صورتی‌که در هر زیربند بخصوص، برای هر پروتکل احراز هویت، تعریف شده است.

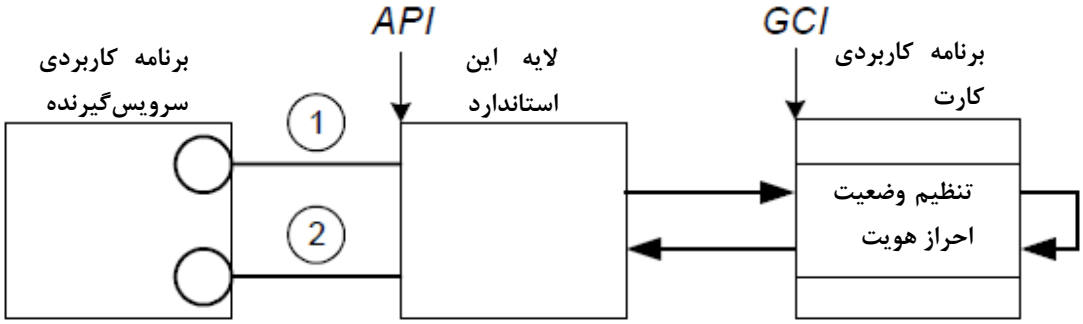
یک تایید برای عمل DIDGet باید شامل یک پارامتر DIDStructure به صورت تعریف شده در زیر، باشد.

```
DIDStructure ::= SEQUENCE {
name DIDName,
authProtocol OBJECT IDENTIFIER,
scope DIDScope,
authenticated BOOLEAN,
marker OCTET STRING,
qualifier DiDQualifierType OPTIONAL
}
```

تایید DIDGet حداقل باید، مقادیر صحیح name، authProtocol، scope و authenticated را برگرداند. منوط به انتخاب‌های پیاده‌سازی و خط‌مشی[1]، ممکن است مقادیر ناقص یا کامل marker و qualifier، در آن تایید، برگردانده شود.

الف-۳ اعلان ساده

احراز یک هویت متمایز کننده مربوط به این پروتکل احراز هویت از طریق اعلان ساده، به‌دست‌می‌آید.



شکل الف-۱ تایید ساده

شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء ،

{INSO(1) standard(0) INSO 16386(16386) part3(3) annex-a(0) simple-assertion(3) }

شناسایی می‌شود.

الف-۳-۱ علامت‌گذار

یک هویت متمایز کننده که از این پروتکل، استفاده می‌کند یک علامت‌گذار خالی دارد (OCTET STRING با طول صفر).

1 - Policy

الف-۳-۲ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالیکه برای این پروتکل احراز هویت، marker یک OCTET STRING با طول صفر است.

الف-۳-۳ DIDUpdate

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل DIDUpdate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۳-۴ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک DIDStructure باشد.

الف-۳-۵ Authentication

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData خالی، به صورتی که در زیر، تعریف شده، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSessionبایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۳-۵-۱ رویه

(۱) authenticationProtocolData ::= empty OCTET STRING

(۲) authenticationProtocolData ::= empty OCTET STRING

الف-۳-۵-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایز کننده نامگذاری شده باید درون برنامه کاربردی کارت، به TRUE تنظیم شود.

الف-۳-۶ Encipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۳-۷ Decipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۳-۸ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل GetRandom باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۳-۹ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Hash باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۳-۱۰ Sign

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۳-۱۱ VerifySignature

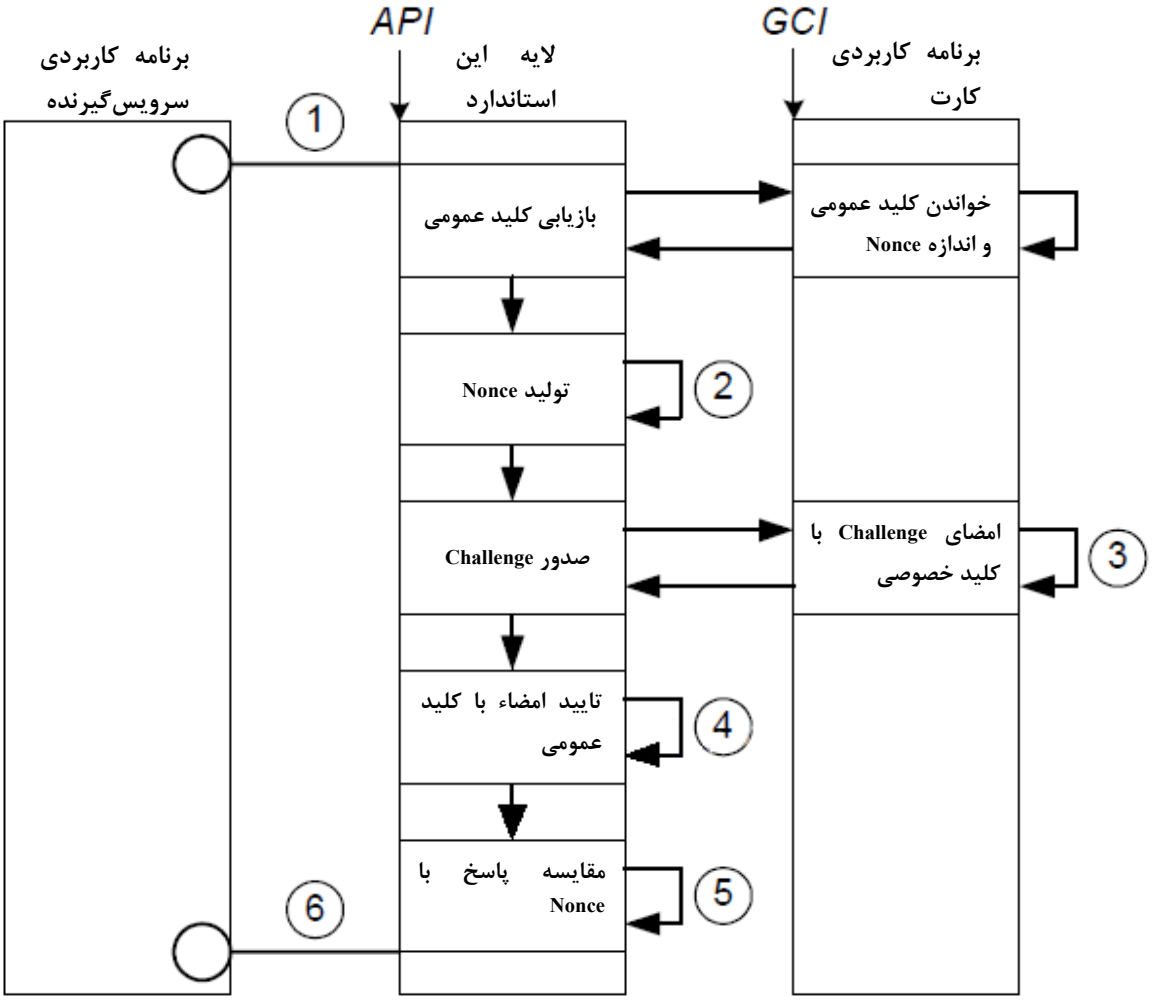
در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifySignature باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۳-۱۲ VerifyCertificate

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifyCertificate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۴ احراز هویت داخلی نامتقارن

این پروتکل احراز هویت، یک پروتکل چالش/ پاسخ با استفاده از رمزنگاشتی کلید عمومی است.



شکل الف-۲ احراز هویت داخلی نامتقارن

الف-۴-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO (1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) asymmetric-internal-authenticate(4) } شناسایی می‌شود.

الف-۴-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP004 ::= SEQUENCE {
signatureAlgorithm OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
CHOICE {
SEQUENCE {
publicKeyMaterial OCTET STRING,
privateKey OCTET STRING
},
generateFlag NULL
}
nonceSize INTEGER
}
```

الف-۴-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۴-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل احراز هویت است.

الف-۴-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد، در حالی که همانگونه که در الف-۴-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

اگر کلیدهای اصلی، به صورتی که به وسیله حضور گزینه generateFlag در زمان درخواست عمل DIDCreate مشخص شده، درون برنامه کاربردی کارت، ایجاد شده باشند، یک درخواست عمل DIDUpdate که حاوی کلیدها است باید کد بازگشتی API_INCORRECT_PARAMETER را برگرداند.

اگر کلیدهای اصلی، به صورتی که به وسیله حضور گزینه‌های publicKeyMaterial و privateKey در زمان درخواست عمل DIDCreate مشخص شده، به وسیله برنامه کاربردی سرویس‌گیرنده، فراهم شده باشند، یک درخواست عمل DIDUpdate به همراه گزینه generateFlag، باید کد بازگشتی API_INCORRECT_PARAMETER را برگرداند.

الف-۴-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۴-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData، به صورتی که در زیر، تعریف شده، اجرا می‌شود.

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSessionبایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۴-۶-۱ رویه

(۱) authenticationProtocolData ::= empty OCTET STRING

(۲) nonce = RNG (nonceSize)

(۳) challenge = signatureAlgorithm [privateKey] (nonce)

(۴) response = signatureAlgorithm-1 [publicKey] (challenge)

(۵) result = (response =?= nonce)

(۶) authenticationProtocolData ::= result BOOLEAN as OCTET STRING

الف-۴-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایزکننده نامگذاری‌شده درون برنامه کاربردی کارت، تغییری نمی‌کند.

الف-۴-۷ Encipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۴-۸ Decipher

outBuffer = encryptionAlgorithm [privateKey] (inBuffer)

الف-۴-۹ GetRandom

random = RNG (nonceSize)

الف-۴-۱۰ Hash

hash = hashAlgorithm (message)

الف-۴-۱۱ Sign

signature = encryptionAlgorithm [privateKey] (message)

الف-۴-۱۲ VerifySignature

message =?= encryptionAlgorithm [publicKey] (signature)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت، API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۴-۱۳ VerifyCertificate

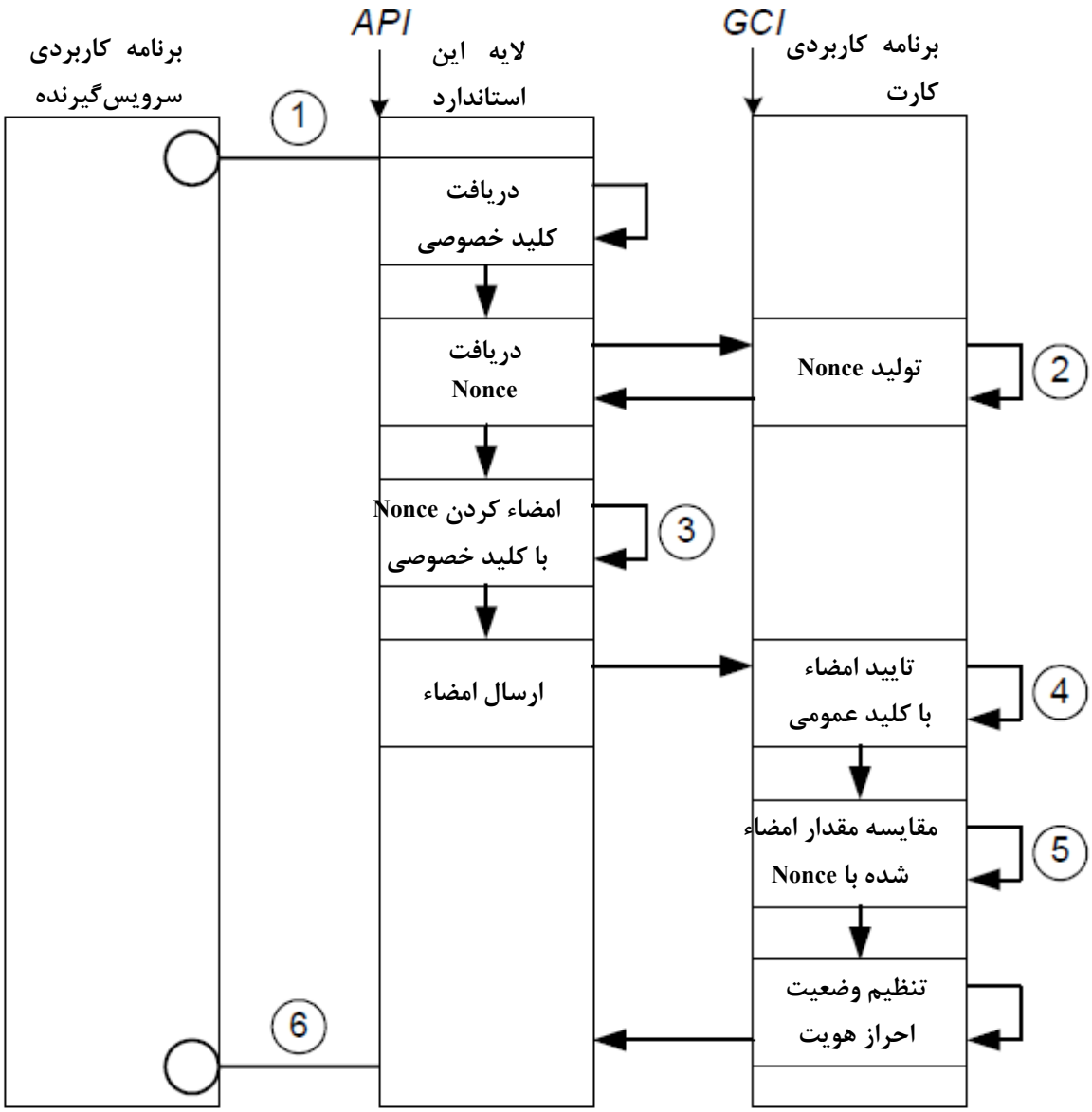
بسته به نوع گواهی،

hash = hashAlgorithm (certificate without signature)
hash =?= encryptionAlgorithm [publicKey] (signature from certificate)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت، API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۵ احراز هویت خارجی نامتقارن

این پروتکل احراز هویت برای برقراری یک وضعیت احراز هویت‌شده یک هویت متمایزکننده در یک برنامه کاربردی کارت، استفاده می‌شود.



شکل الف-۳ احراز هویت خارجی نامتقارن

الف-۵-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386(16386) part3(3) annex-a(0) asymmetric-external-authenticate(5) } شناسایی می‌شود.

الف-۵-۲ علامت‌گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP005 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
publicKeyMaterial OCTET STRING,
nonceSize INTEGER
}
```

الف-۵-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۵-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۵-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد، در حالی‌که همانگونه که در الف-۵-۲ تعریف‌شده، marker، علامت-گذار خاص این پروتکل است.

الف-۵-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۵-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData، به صورتی که در زیر، تعریف شده، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSessionبایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۵-۶-۱ رویه

(۱) authenticationProtocolData ::= privateKey OCTET STRING

یا empty OCTET STRING اگر گزینه samConnectionHandle در درخواست DIDAuthenticate استفاده شده باشد.

(۲) nonce = RNG (nonceSize)

(۳) signature = encryptionAlgorithm [privateKey] (nonce)

اگر گزینه samConnectionHandle در درخواست DIDAuthenticate استفاده شده باشد، لایه این استاندارد باید از برنامه کاربردی کارتی که به وسیله این samConnectionHandle به آن متصل است، تقاضا کند که این کار را انجام دهد.

(۴) message = $\text{encryptionAlgorithm}^{-1}$ [publicKey] (signature)

(۵) result = (nonce =?= message)

(۶) **authenticationProtocolData** ::= empty OCTET STRING

الف-۵-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، اگر مقدار result برابر با TRUE باشد، باید وضعیت احراز هویت مربوط به هویت متمایزکننده نامگذاری‌شده درون برنامه کاربردی کارت، به TRUE تنظیم شود.

الف-۵-۷ Encipher

outBuffer = encryptionAlgorithm [publicKey] (inBuffer)

الف-۵-۸ Decipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۵-۹ GetRandom

random = RNG (nonceSize)

الف-۵-۱۰ Hash

hash = hashAlgorithm (message)

الف-۵-۱۱ Sign

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۵-۱۲ VerifySignature

message =?= encryptionAlgorithm [publicKey] (signature)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت، API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۵-۱۳ VerifyCertificate

بسته به نوع گواهی،

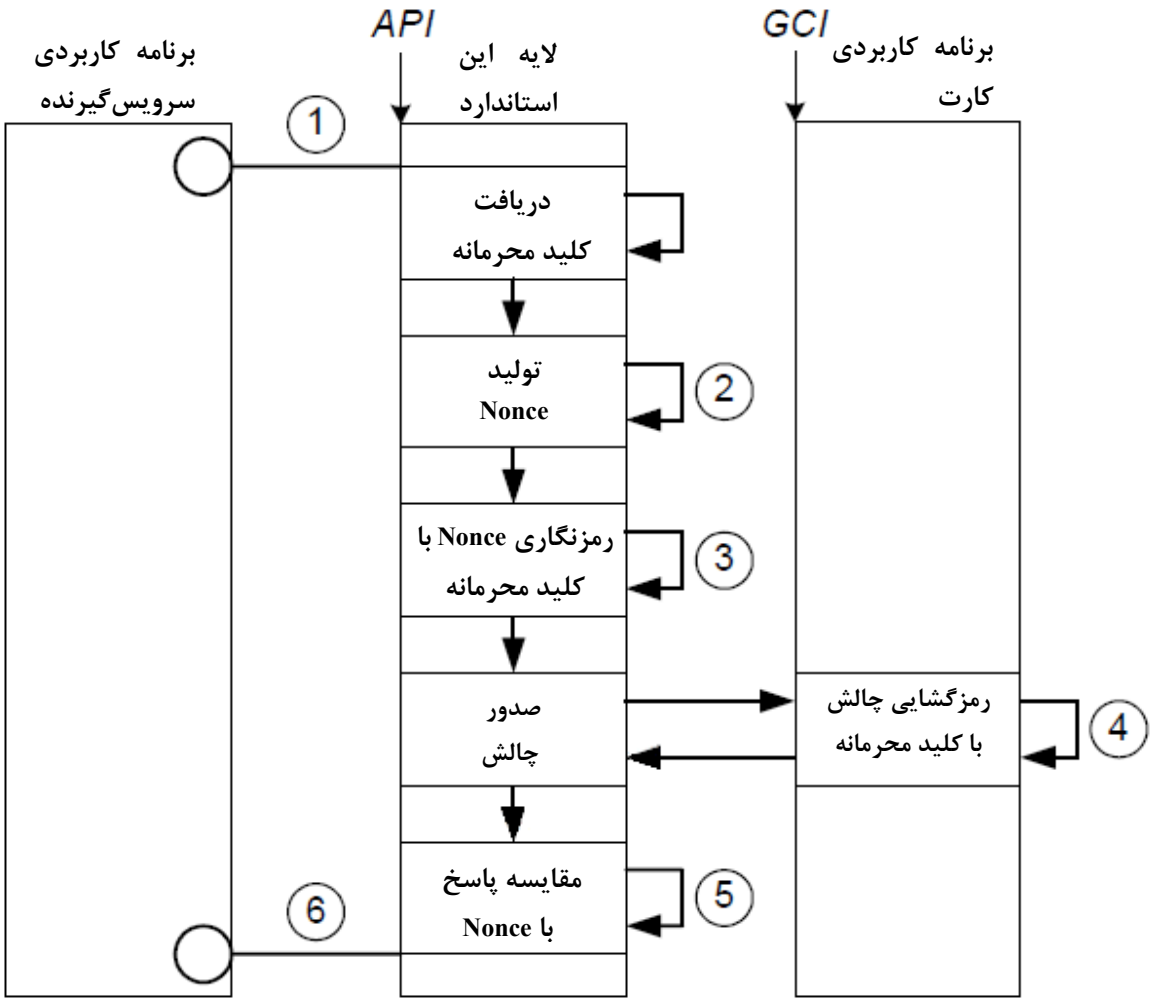
hash = hashAlgorithm (certificate without signature)
hash =?= encryptionAlgorithm [publicKey] (signature from certificate)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت، API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۶ احراز هویت داخلی متقارن

این پروتکل احراز هویت، یک پروتکل چالش/پاسخ[1] با استفاده از رمزنگاشتی متقارن می‌باشد.

1 - Challenge/Response



شکل الف-۴ احراز هویت داخلی متقارن

الف-۶-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386(16386) part3(3) annex-a(0) symmetric-internal-authenticate(6) } شناسایی می‌شود.

الف-۶-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP006 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
secretKey OCTET STRING,
nonceSize INTEGER
}
```

پارامترهای keySize و nonceSize به ترتیب، نشان‌دهنده تعداد بیت‌های key و nonce هستند.

پارامتر secretKey، رشته بیتی خود key است.

الف-۶-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۶-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۶-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد، در حالی‌که همانگونه که در الف-۶-۲ تعریف‌شده، marker، علامت-گذار خاص این پروتکل است.

الف-۶-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۶-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData، به صورت تعریف شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSessionبایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۶-۶-۱ رویه

(۱) authenticationProtocolData ::= secretKey OCTET STRING

یا empty OCTET STRING اگر گزینه samConnectionHandle در درخواست DIDAuthenticate استفاده شده باشد.

(۲) nonce = RNG (nonceSize)

(۳) challenge = encryptionAlgorithm [secretKey] (nonce)

(۴) response = encryptionAlgorithm^{-1} [secretKey] (challenge)

(۵) result = (nonce =?= response)

(۶) **authenticationProtocolData** ::= result BOOLEAN as OCTET STRING

الف-۶-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایزکننده نامگذاری‌شده درون برنامه کاربردی کارت، تغییری نمی‌کند.

الف-۶-۷ Encipher

outBuffer = encryptionAlgorithm [secretKey] (inBuffer)

الف-۶-۸ Decipher

outBuffer = encryptionAlgorithm^{-1} [secretKey] (inBuffer)

الف-۶-۹ GetRandom

random = RNG (nonceSize)

الف-۶-۱۰ Hash

hash = hashAlgorithm (message)

الف-۶-۱۱ Sign

digest = hashAlgorithm (message)
signature = encryptionAlgorithm [secretKey] (digest)

الف-۶-۱۲ VerifySignature

digest = hashAlgorithm (message)
digest =?= encryptionAlgorithm^{-1} [secretKey] (signature)

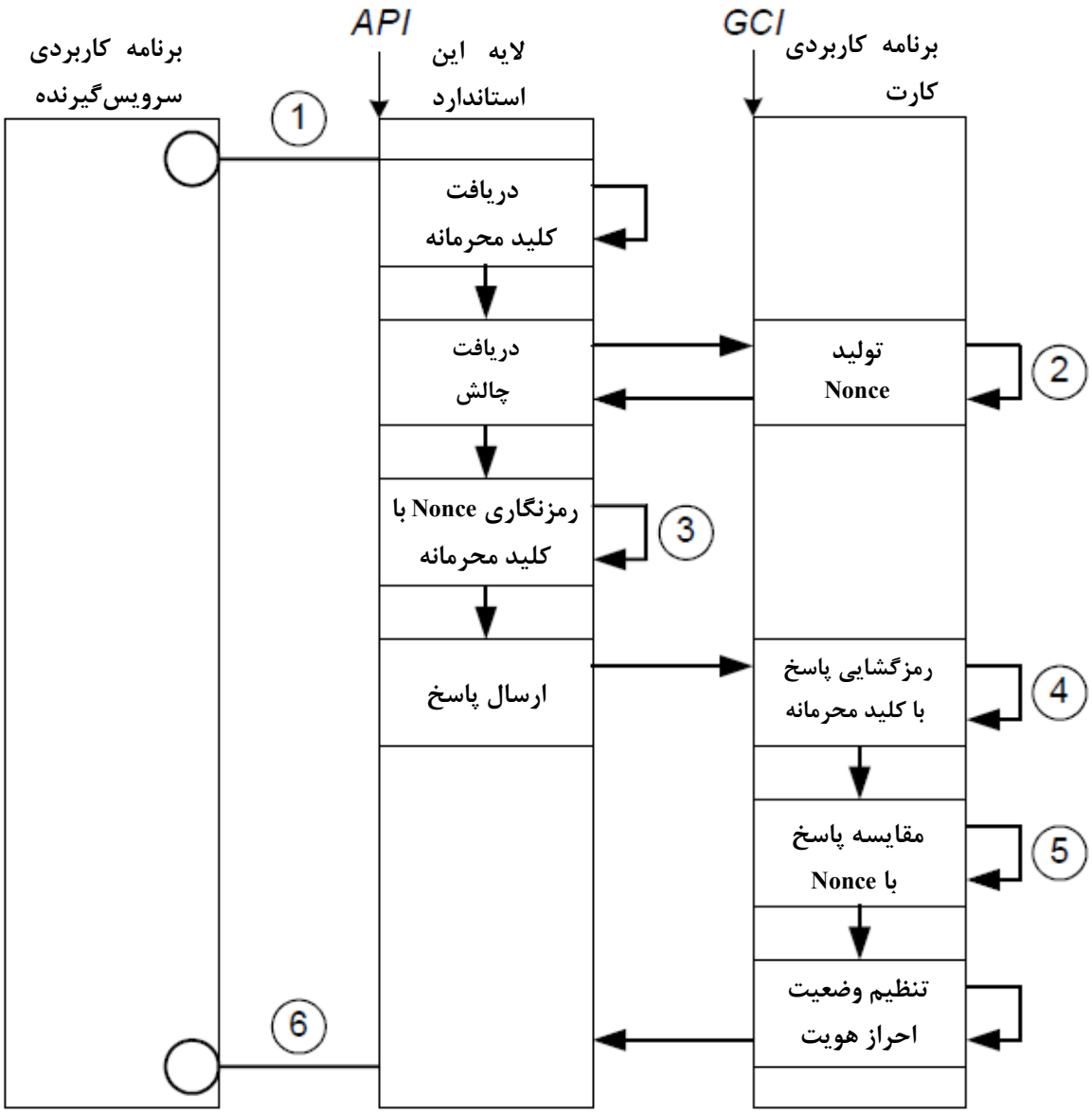
اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت، API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۶-۱۳ VerifyCertificate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۷ احراز هویت خارجی متقارن

این پروتکل احراز هویت، برای برقراری یک وضعیت احراز هویت‌شده یک هویت متمایزکننده در یک برنامه کاربردی کارت، استفاده می‌شود.



شکل الف- ۵ احراز هویت خارجی متقارن

الف-۷-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، { INSO (1) standard(0) INSO 16386(16386) part3(3) annex-a(0) symmetric-external-authenticate(7) } شناسایی می‌شود.

الف-۷-۲ علامت گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP007 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
secretKey OCTET STRING,
nonceSize INTEGER
}
```

الف-۷-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۷-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۷-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد، در حالی که همانگونه که در الف-۷-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۷-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۷-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData، به صورتی که در زیر، تعریف شده، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSessionبایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۷-۶-۱ رویه

(۱) authenticationProtocolData ::= secretKey OCTET STRING

یا empty OCTET STRING اگر گزینه samConnectionHandle در درخواست DIDAuthenticate استفاده شده باشد.

(۲) nonce = RNG (nonceSize)

(۳) response = encryptionAlgorithm [secretKey] (nonce)

(۴) challenge = $\text{encryptionAlgorithm}^{-1}$ [secretKey] (nonce)

(۵) result = (nonce =?= challenge)

(۶) **authenticationProtocolData** ::= empty OCTET STRING

الف-۷-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایز کننده نامگذاری شده، باید در برنامه کاربردی کارت به مقدار result تنظیم شود.

الف-۷-۷ Encipher

outBuffer = encryptionAlgorithm [secretKey] (inBuffer)

الف-۷-۸ Decipher

outBuffer = encryptionAlgorithm^{-1} [secretKey] (inBuffer)

یادآوری – encryptionAlgorithm برای رمزگشایی استفاده می‌شود و ممکن است شکل متفاوتی از "encryptionAlgorithm" باشد.

الف-۷-۹ GetRandom

random = RNG (nonceSize)

الف-۷-۱۰ Hash

hash = hashAlgorithm (message)

الف-۷-۱۱ Sign

digest = hashAlgorithm (message)
signature = encryptionAlgorithm [secretKey] (digest)

الف-۷-۱۲ VerifySignature

digest = hashAlgorithm (message)
digest =?= encryptionAlgorithm^{-1} [secretKey] (signature)

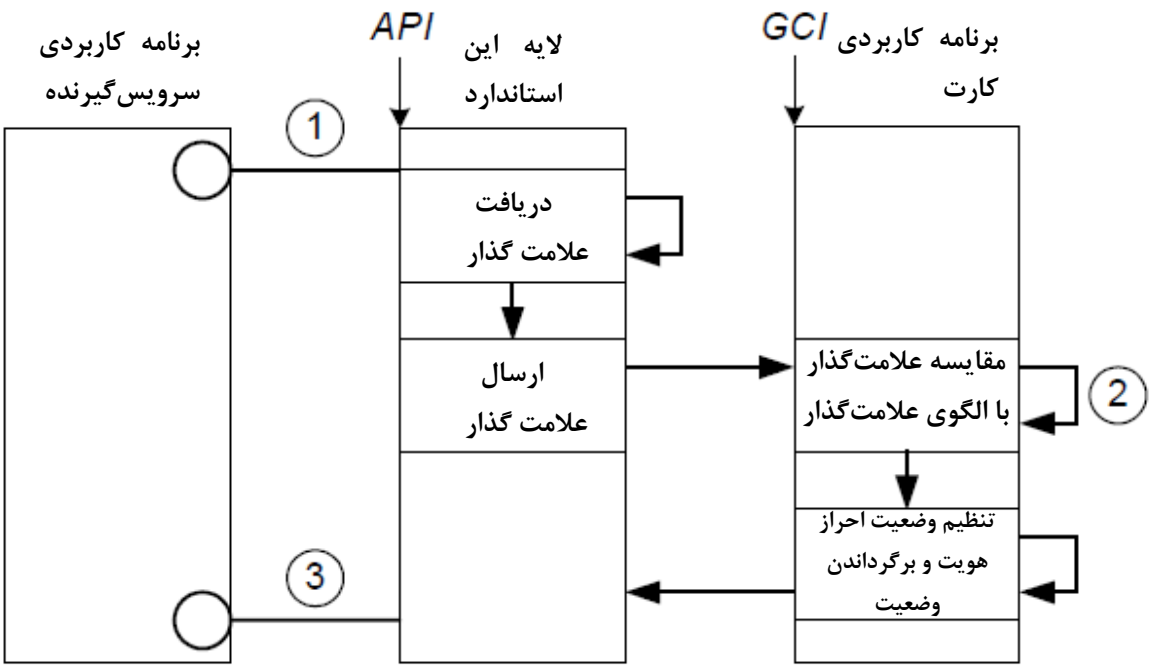
اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت، API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۷-۱۳ VerifyCertificate

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۸ مقایسه

این پروتکل احراز هویت، یک مقدار ارائه شده را با مقدار ذخیره شده در علامت‌گذار یک هویت متمایز کننده، مقایسه می‌کند.



شکل الف-۶ مقایسه

الف-۸-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، { INSO (1) standard(0) INSO 16386(16386) part3(3) annex-a(0) compare (8) } شناسایی می‌شود.

الف-۸-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP008 ::= SEQUENCE {
minDataLength INTEGER,
maxDataLength INTEGER,
paddingCharacter OCTET STRING,
markerTemplate OCTET STRING
}
```

پارامترهای minDataLength و maxDataLength به ترتیب، نشان‌دهنده تعداد بیت‌های key و nonce هستند. پارامتر paddingCharacter، یک تک بایت است.

الف-۸-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۸-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۸-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد، در حالی که همانگونه که در الف-۸-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۸-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۸-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData، به صورتی‌که در زیر، تعریف شده، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSessionبایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۸-۶-۱ رویه

(۱) authenticationProtocolData ::= markerTemplate OCTET STRING SIZE(minDataLength..maxDataLength)

marker باید به ترتیب بایت big-endian و با پر کردن کم ارزش‌ترین بایت در انتها قالب‌بندی‌شده باشد.

(۲) result = (marker =?= markerTemplate)

مقایسه به صورت بیت به بیت است.

(۳) authenticationProtocolData ::= empty OCTET STRING

الف-۸-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایزکننده نامگذاری شده، باید درون برنامه کاربردی کارت به مقدار result تنظیم شود.

الف-۸-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۸-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۸-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل GetRandom باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۸-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Hash باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۸-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۸-۱۲ VerifySignature

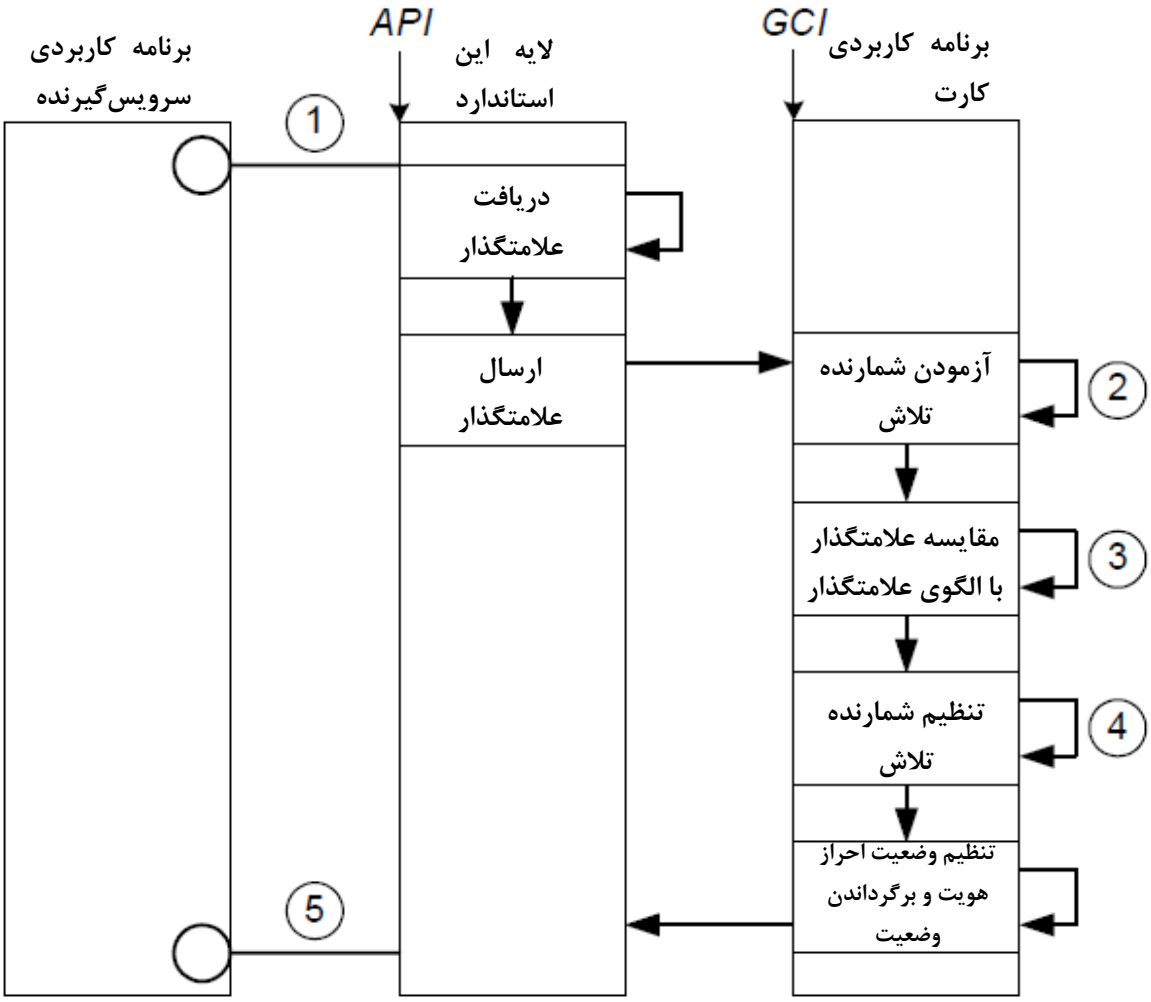
در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifySignature باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۸-۱۳ VerifyCertificate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifyCertificate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۹ مقایسه PIN

این پروتکل احراز هویت، یک PIN ارائه شده را با مقدار ذخیره شده در علامتگذار یک هویت متمایزکننده، مقایسه می‌کند.



شکل الف-۷ مقایسه PIN

الف-۹-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، { INSO (1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) pin-compare (9) } شناسایی می‌شود.

الف-۹-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP009 ::= SEQUENCE {
minDataLength INTEGER,
maxDataLength INTEGER,
storedLength INTEGER,
paddingCharacter OCTET STRING,
markerTemplate OCTET STRING,
maxAttempts INTEGER,
attemptsCounter INTEGER,
pinRef OCTET STRING,
pinValue VisibleString
}
```

پارامترهای minDataLength و maxDataLength به ترتیب، نشان‌دهنده تعداد بیت‌های key و nonce هستند.

پارامتر paddingCharacter، یک تک بایت است.

markerTemplate باید به ترتیب بایت big-endian و با پر کردن کم ارزش‌ترین بایت در انتها قالب بندی شده باشد.

الف-۹-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۹-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۹-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد، در حالی که همانگونه که در الف-۹-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

اجرای این عمل باید attemptsCounter را به صفر، تنظیم نماید.

الف-۹-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر didStructure باشد.

الف-۹-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData، به صورت تعریف شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSession بایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۹-۶-۱ رویه

(۱) authenticationProtocolData ::= markerTemplate OCTET STRING SIZE(minDataLength..maxDataLength)

marker باید به ترتیب بایت big-endian قالب‌بندی‌شده باشد با پر کردن کم ارزش‌ترین بایت در انتها.

(۲) اگر attemptsCounter برابر با یا بزرگتر از maxAttempts باشد، ادامه اجرای این درخواست باید از شماره ۵ انجام شود.

(۳) result = (marker =?= markerTemplate)

مقایسه به صورت بیت به بیت است.

(۴) اگر result برابر با TRUE باشد، attemptsCounter به صفر تنظیم می‌شود. در غیر این صورت، attemptsCounter افزایش می‌یابد.

retries = maxAttempts – attemptsCounter

(۵) authenticationProtocolData ::= retries INTEGER OPTIONAL as OCTET STRING

الف-۹-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایز کننده نامگذاری شده، باید درون برنامه کاربردی کارت به مقدار result تنظیم شود.

الف-۹-۷ Encipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۹-۸ Decipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۹-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل GetRandom باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۹-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Hash باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۹-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۹-۱۲ VerifySignature

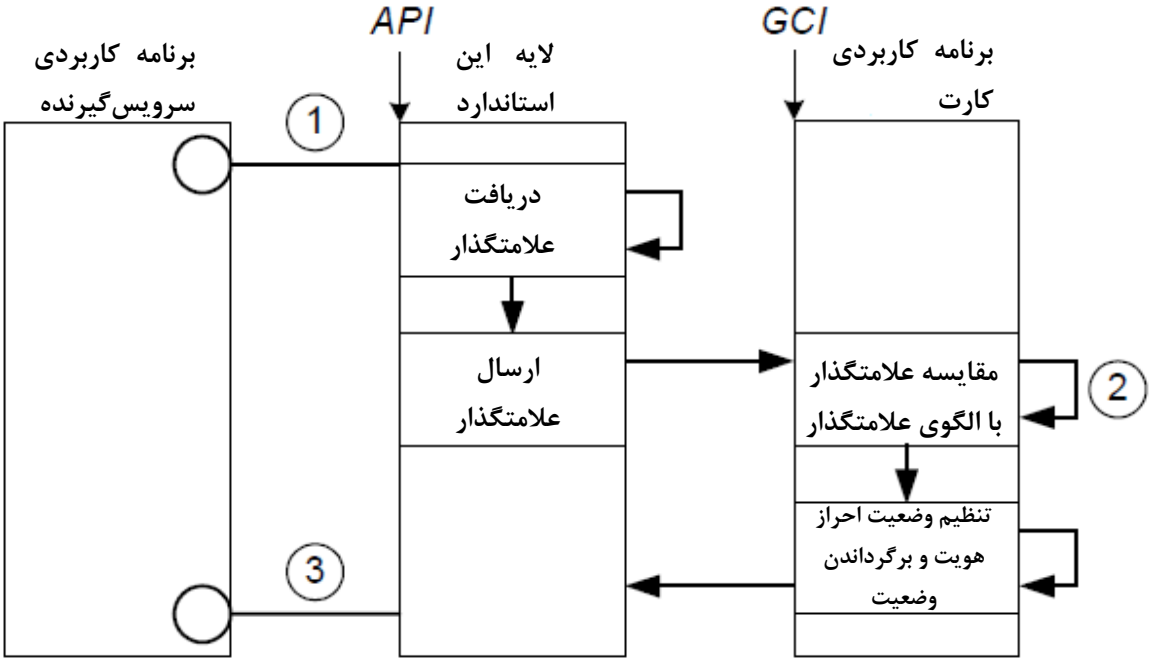
در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifySignature باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۹-۱۳ VerifyCertificate

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifyCertificate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۰ مقایسه زیست‌سنجی

این پروتکل احراز هویت، یک مقدار ارائه شده را با مقدار ذخیره شده در علامت‌گذار یک هویت متمایز کننده، مقایسه می‌کند.



شکل الف-۸ مقایسه زیست سنجی

الف-۱۰-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، { INSO (1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) biometric-compare (10) } شناسایی می‌شود.

الف-۱۰-۲ علامت‌گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP010 ::= SEQUENCE {
bit OCTET STRING,
markerTemplate OCTET STRING
}
```

پارامتر bit باید به صورت یک الگوی اطلاعات زیست‌سنجی استاندارد ISO/IEC 7816-11 قالب‌بندی شود. پارامتر markerTemplate باید به صورت یک الگوی زیست‌سنجی استاندارد ISO/IEC 7816-11 قالب‌بندی شود.

الف-۱۰-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته‌شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۰-۲ تعریف‌شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۰-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد، در حالی که همانگونه که در الف-۱۰-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۰-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۰-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData، به صورتی که در زیر، تعریف شده، اجرا می‌شود.

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSession بایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۱۰-۶-۱ رویه

(۱) authenticationProtocolData ::= MarkerAP010 as OCTET STRING

(۲) result = (marker =?= markerTemplate)

مقایسه به صورتی است که در الگوی اطلاعات زیست سنجی استاندارد 7816-11 تعریف شده است.

(۳) authenticationProtocolData ::= empty OCTET STRING

الف-۱۰-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایزکننده نامگذاری‌شده، باید درون برنامه کاربردی کارت به مقدار result تنظیم شود.

الف-۱۰-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۰-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۰-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل GetRandom باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۰-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Hash باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۰-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

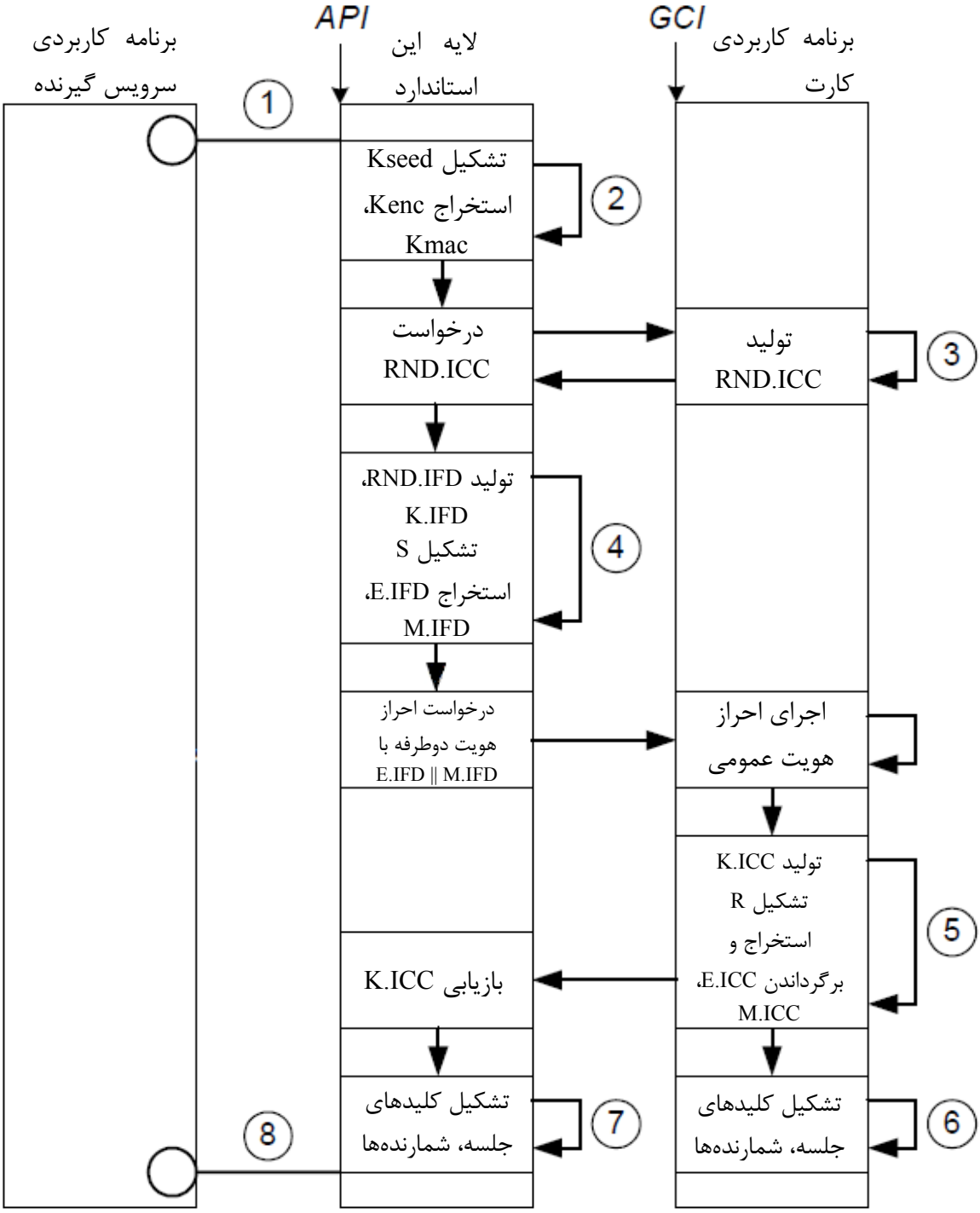
الف-۱۰-۱۲ VerifySignature

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifySignature باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۰-۱۳ VerifyCertificate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifyCertificate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱ احراز هویت دوطرفه با استقرار کلید



شکل الف ۹- احراز هویت دوطرفه با استقرار کلید

الف-۱۱-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) mutual-authentication-with-key-establishment (11) } شناسایی می‌شود.

الف-۱۱-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP011 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
macAlgorithm OBJECT IDENTIFIER,
derivationAlgorithmK_enc OBJECT IDENTIFIER,
derivationAlgorithmK_mac OBJECT IDENTIFIER,
derivationAlgorithmK_IFD OBJECT IDENTIFIER,
derivationAlgorithmSessionKeysAndCounters OBJECT IDENTIFIER
}
```

الف-۱۱-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته‌شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۱-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۱-۴ DIDUpdate

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل DIDUpdate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۱-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل‌های CardApplicationStartSession با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل DIDAuthenticate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۶-۱ رویه

authenticationProtocolData ::= K_seed OCTET STRING (۱)

K_enc = derivationAlgorithmK_enc (Kseed) (۲)

K_mac = derivationAlgorithmK_mac (Kseed)

RND.ICC = RNG (8 bytes) (۳)

RND.IFD = RNG (8 bytes) (۴)

K.IFD = RNG (16 bytes)
S = RND.IFD || RND.ICC || K.IFD
E.IFD = encryptionAlgorithm [K_enc] (S)
M.IFD = macAlgorithm [K_mac] (E.IFD)

M.IFD =?= macAlgorithm [K_mac] (E.IFD) (۵)

S' = $\text{encryptionAlgorithm}^{-1}$ [K_enc] (E.IFD)

بازیابی RND.ICC از 'S و مقایسه با مورد اصلی آن

اگر برابر نباشند، یک خطا برگردانده شود.

K.IFD از 'S بازیابی می‌شود.

K.ICC = RNG (16 bytes)
R = RND.ICC || RND.IFD || K.ICC
E.ICC = encryptionAlgorithm [K_enc] (R)
M.ICC = macAlgorithm [K_mac] (E.ICC)

(۶) استخراج کلیدهای جلسه و شمارنده ترتیب ارسال (SSC)

K_enc || K_mac || SSC = derivationAlgorithmSessionKeysAndCounters(K.ICC)

result = (M.ICC =?= macAlgorithm [K_mac] (E.ICC)) (۷)

R' = encryptionAlgorithm^{-1} [K_enc] (E.ICC)

RND.IFD بازیابی از 'R و مقایسه با مورد اصلی آن

اگر برابر نباشند، یک خطا برگردانده شود.

K.ICC از R بازیابی می‌شود.

K_enc || K_mac | SSC = derivationAlgorithmSessionKeysAndCounters(K.ICC)
authenticationProtocolData ::= result BOOLEAN as OCTET STRING (۸)

الف-۱۱-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایز کننده نامگذاری شده، باید درون برنامه کاربردی کارت به مقدار TRUE تنظیم شود و یک جلسه، شروع خواهد شد.

الف-۱۱-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل GetRandom باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Hash باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۱۲ VerifySignature

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifySignature باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۱۳ VerifyCertificate

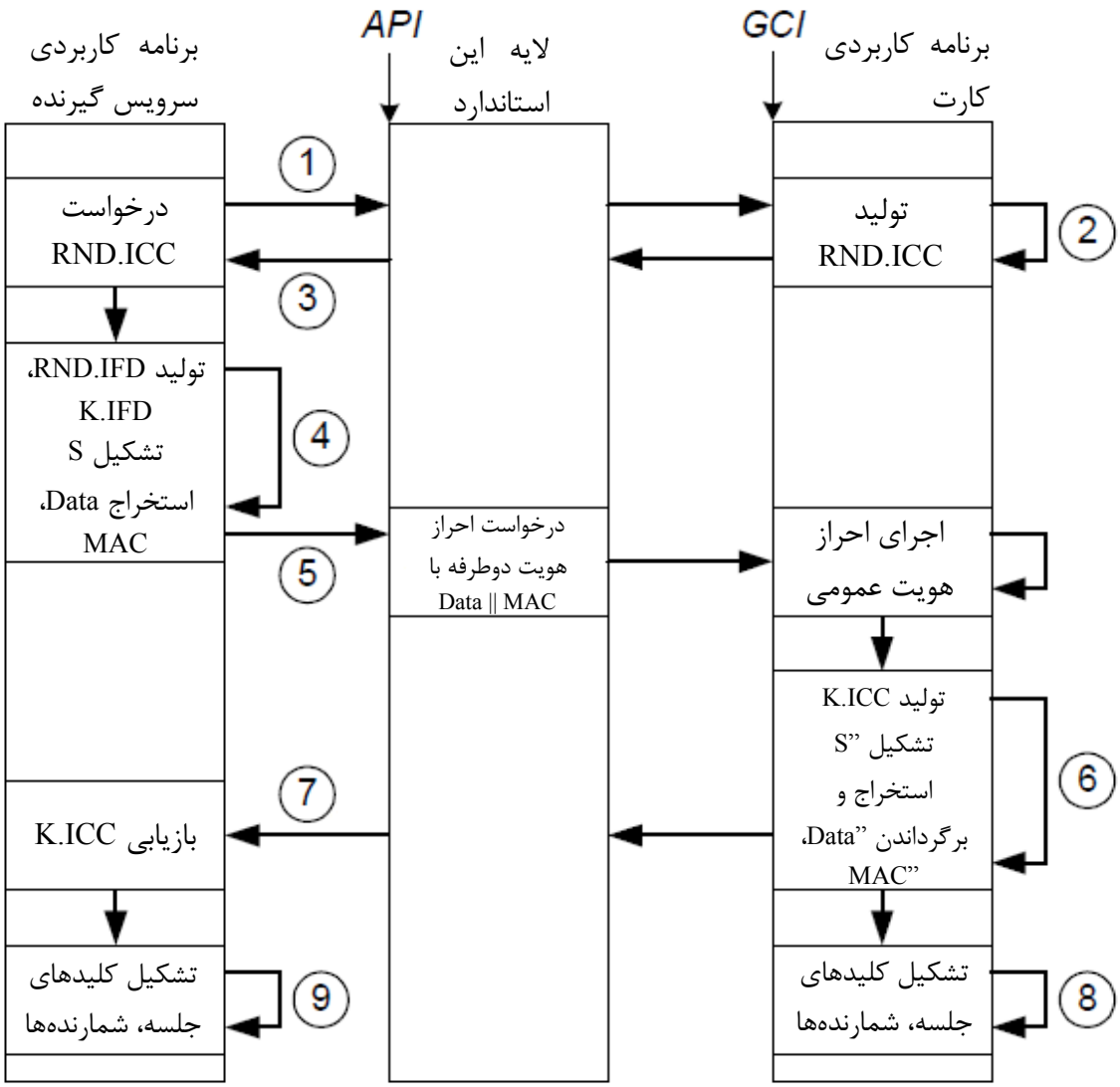
در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifyCertificate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۱-۱۴ CardApplicationEndSession

پس از این که هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، احراز هویت شد، یک درخواست برای این عمل، باید وضعیت احراز هویت مربوط به این هویت متمایز کننده را به FALSE تنظیم کند و کد بازگشتی API_OK را برگرداند.

الف-۱۲ احراز هویت دوطرفه برنامه کاربردی سرویس‌گیرنده با استقرار کلید

این پروتکل احراز هویت، پس از تعریف شدن پروتکل احراز هویت دوطرفه با استقرار کلید، به وسیله استانداردهای CEN/EN 14890-1 و CEN/EN 14890-2، شکل گرفت.



شکل الف ۱۰- احراز هویت دوطرفه برنامه کاربردی سرویس گیرنده با استقرار کلید

الف-۱۲-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) client-application-mutual-authentication-wke (12) } شناسایی می‌شود.

الف-۱۲-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP012 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
macAlgorithm OBJECT IDENTIFIER,
encryptionAlgorithmForSessionKey OBJECT IDENTIFIER,
macAlgorithmForSessionKey OBJECT IDENTIFIER,
derivationAlgorithmSessionKeysAndCounter OBJECT IDENTIFIER,
K_enc OCTET STRING,
K_mac OCTET STRING,
DIV_IFD OCTET STRING
```

}

مقدار DIV_IFD، ۸ بایت، طول دارد.

الف-۱۲-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایز کننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۲-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۲-۴ DIDUpdate

یک درخواست عمل DIDUpdate در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، باید شامل پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۲-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۲-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۲-۶ احراز هویت

این پروتکل احراز هویت، به‌وسیله درخواست‌های پی‌درپی عمل DIDAuthenticate یا CardApplicationStartSession با اداره کردن اتصال و هویت متمایزکننده یکسان، به همراه یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

الف-۱۲-۶-۱ پیش نیاز

برنامه کاربردی سرویس‌گیرنده باید قبل از مرحله ۱، DIV.ICC ۸ بایتی را به دست آورد.

الف-۱۲-۶-۲ رویه

authenticationProtocolData ::= empty OCTET STRING (۱)

RND.ICC = RNG (8 bytes) (۲)

authenticationProtocolData ::= RND.ICC OCTET STRING (۳)

RND.IFD = RNG (8 bytes) (۴)

K.IFD = RNG (32 bytes)
S = RND.IFD || SN.IFD || RND.ICC || SN.ICC || K.IFD
Data = encryptionAlgorithm [K_enc] (S)
MAC = macAlgorithm [K_mac] (Data)

authenticationProtocolData ::= SEQUENCE { (۵)

Data OCTET STRING,
MAC OCTET STRING
}

MAC =?= macAlgorithm [K_mac] (Data) (۶)

S' = encryptionAlgorithm^{-1} [K_enc] (Data)

RND.ICC و DIV.ICC از 'S بازیابی می‌شوند و با موارد اصلی آن‌ها، مقایسه می‌شوند.

K.IFD از 'S بازیابی می‌شود.

K.ICC = RNG (32 bytes)
S" = RND.ICC || DIV.ICC || RND.IFD || DIV.IFD || K.ICC
Data" = encryptionAlgorithm [K_enc] (S")
MAC" = macAlgorithm [K_mac] (Data")

authenticationProtocolData ::= SEQUENCE { (۷)

Data" OCTET STRING,
MAC" OCTET STRING
}

(۸) کلیدهای جلسه و شمارنده ترتیب ارسال (SSC)، از کلید با مقدار اولیه $K_{IFD/ICC}$ محاسبه می‌شوند.

K_enc || K_mac || SSC = derivationAlgorithmSessionKeysAndCounters(K.ICC)
result = (MAC" =?= macAlgorithm [K_mac] (Data")) (۹)
S" ' = $encryptionAlgorithm^{-1}$ [K_enc] (Data")

RND.IFD و DIV.IFD از '"S بازیابی می‌شوند و با موارد اصلی آن‌ها، مقایسه می‌شوند.

K.ICC از '"S بازیابی می‌شود.

کلیدهای جلسه و شمارنده ترتیب ارسال (SSC)، از کلید با مقدار اولیه $K_{IFD/ICC}$ محاسبه می‌شوند.

K_enc || K_mac || SSC = derivationAlgorithmSessionKeysAndCounters(K.ICC)

الف-۱۲-۶-۳ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایز کننده نامگذاری شده، باید درون برنامه کاربردی کارت به مقدار TRUE تنظیم شود و یک جلسه، شروع خواهد شد.

الف-۱۲-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Encipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۲-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۲-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل GetRandom باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۲-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Hash باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۲-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۲-۱۲ VerifySignature

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifySignature باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۲-۱۳ VerifyCertificate

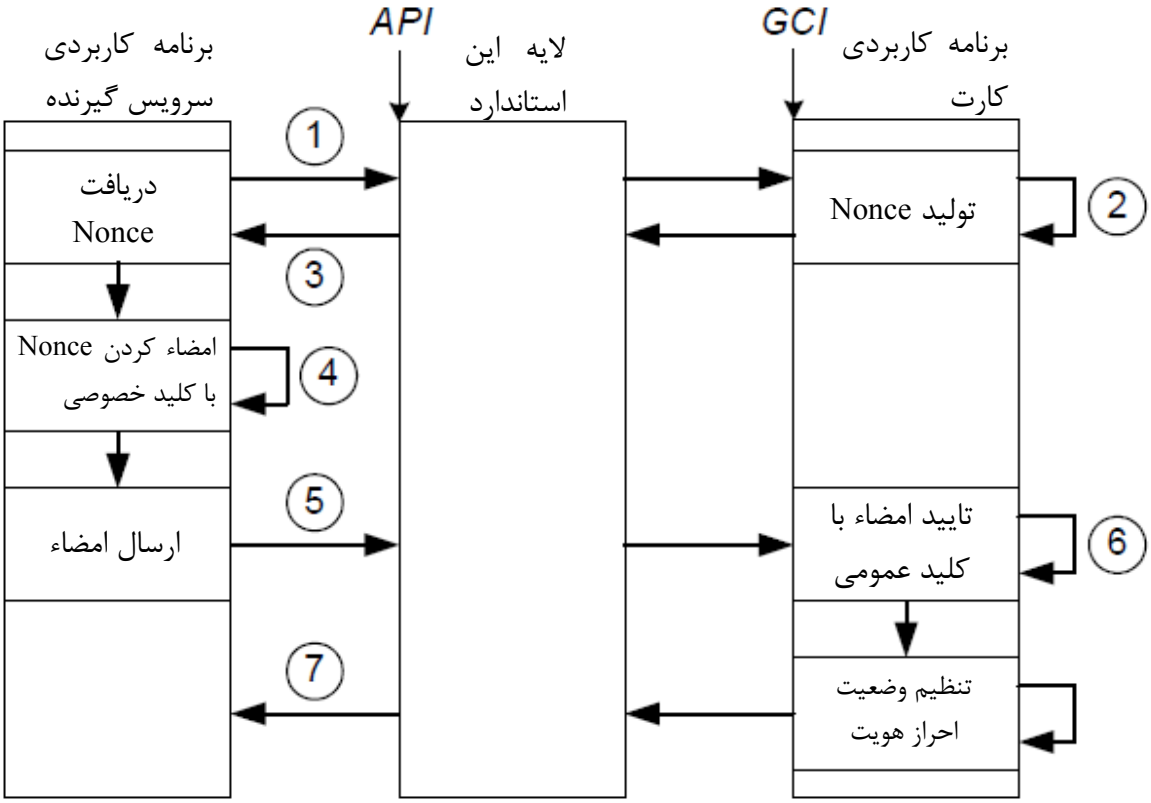
در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل VerifyCertificate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۲-۱۴ CardApplicationEndSession

پس از این که هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، احراز هویت شد، یک درخواست برای این عمل، باید وضعیت احراز هویت مربوط به این هویت متمایز کننده را به FALSE تنظیم کند و کد بازگشتی API_OK را برگرداند.

الف-۱۳ احراز هویت خارجی نامتقارن برنامه کاربردی سرویس‌گیرنده

این پروتکل احراز هویت، در یک برنامه کاربردی کارت، برای برقرار کردن یک وضعیت احراز هویت شده مربوط به یک هویت متمایز کننده، استفاده می‌شود.



شکل الف ۱۱- احراز هویت خارجی نامتقارن برنامه کاربردی سرویس‌گیرنده

الف-۱۳-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO (1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) client-app-asymmetric-external-authentication (13) } شناسایی می‌شود.

الف-۱۳-۲ علامت‌گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP013 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
publicKeyMaterial OCTET STRING,
nonceSize INTEGER
}
```

الف-۱۳-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۳-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۳-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۳-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۳-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۳-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSession بایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۳-۶-۱ رویه

(۱) authenticationProtocolData ::= empty OCTET STRING

(۲) nonce = RNG (nonceSize)

(۳) authenticationProtocolData ::= nonce OCTET STRING

(۴) signature = encryptionAlgorithm [privateKey] (nonce)

(۵) authenticationProtocolData ::= signature OCTET STRING

(۶) message = $encryptionAlgorithm^{-1}$ [publicKey] (signature)

result = (message =?= nonce)

authenticationProtocolData ::= empty OCTET STRING (۷)

الف-۱۳-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، اگر مقدار result برابر با TRUE باشد، وضعیت احراز هویت مربوط به هویت متمایزکننده نامگذاری شده، باید درون برنامه کاربردی کارت به مقدار TRUE تنظیم شود.

الف-۱۳-۷ Encipher

outBuffer = encryptionAlgorithm [publicKey] (inBuffer)

الف-۱۳-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Decipher باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۳-۹ GetRandom

random = RNG (nonceSize)

الف-۱۳-۱۰ Hash

hash = hashAlgorithm (message)

الف-۱۳-۱۱ Sign

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل Sign باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۳-۱۲ VerifySignature

message =?= encryptionAlgorithm [publicKey] (signature)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۱۳-۱۳ VerifyCertificate

منوط به نوع گواهی،

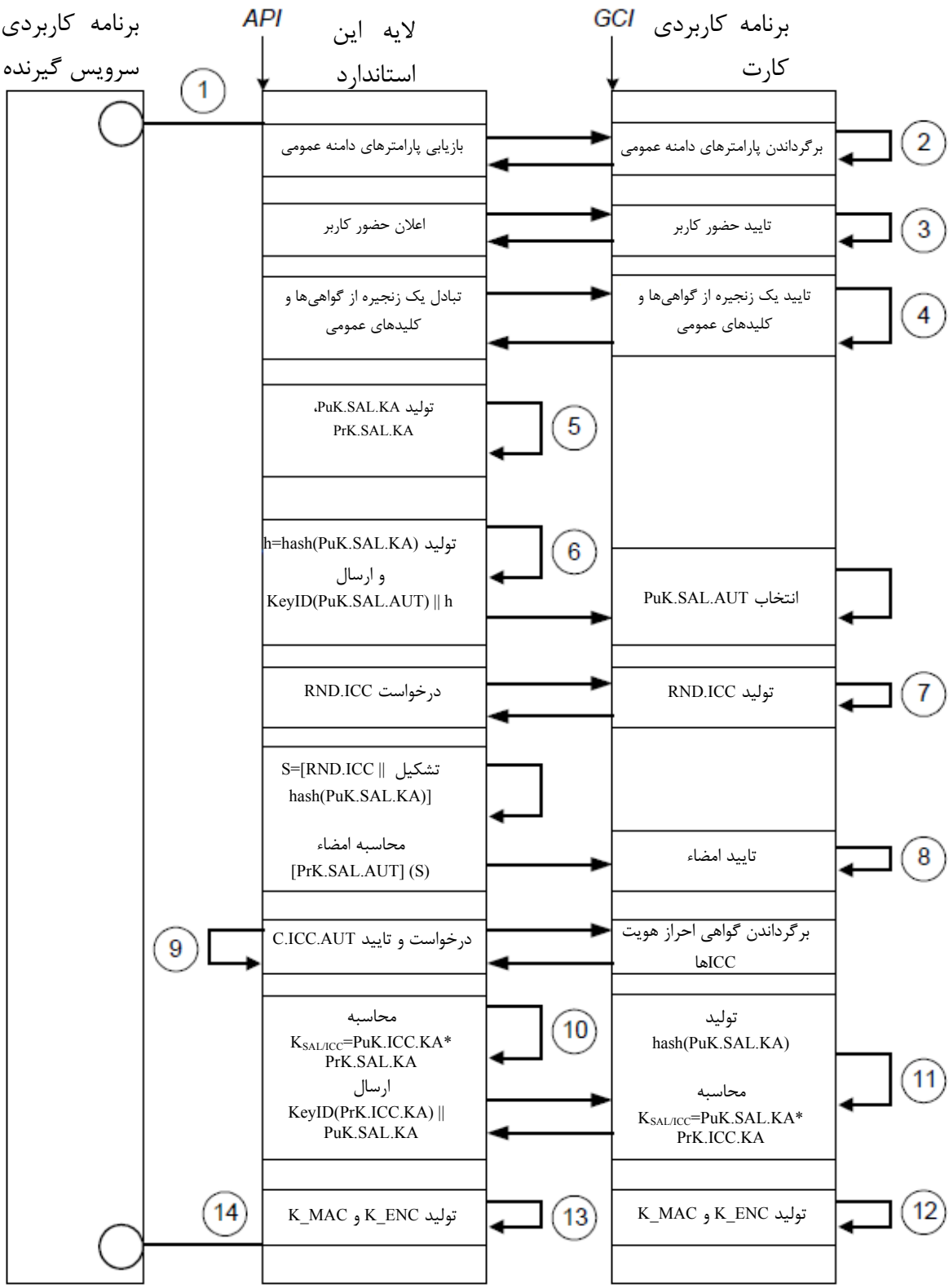
hash = hashAlgorithm (certificate without signature)
hash =?= encryptionAlgorithm [publicKey] (signature from certificate)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۱۴ پروتکل کنترل دسترسی توسعه یافته پودمانی (M-EAC)

برنامه کاربردی سرویس‌گیرنده می‌تواند به وسیله فراخوانی Differential-IdentityAuthenticate به همراه یک DIDName، که دارای احراز هویت دوطرفه با عدم قابلیت ردگیری مشخص شده به عنوان پروتکل احراز هویت است، اجرای این پروتکل را آغاز نماید.

این پروتکل، بر اساس الگوی "احراز هویت دستگاه اجبار حریم خصوصی با عدم قابلیت ردگیری منطبق بر ELC" شرح داده شده در prEN 14890-1 فصل ۸-۶ شکل گرفته است.



شکل الف ۱۲ـ پروتکل کنترل دسترسی توسعه یافته پودمانی (M-EAC)

الف-۱۴-۱ شناسانه شیء پروتکل

این پروتکل احراز هویت به وسیله شناسانه‌های شیء زیر، شناسایی می‌شود:

1- {INSO (1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) authentication-protocol(0) Modular-extended-access-control-protocol(14) without-local-authentication(0)}
2- {INSO (1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) authentication-protocol(0) Modular-extended-access-control-protocol(14) with-non-traceability(1)}
3- {INSO (1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) authentication-protocol(0) Modular-extended-access-control-protocol(14) with-local-authentication(2)}

الف-۱۴-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP014 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
encryptionAlgorithmForSessionKey OBJECT IDENTIFIER,
macAlgorithmForSessionKey OBJECT IDENTIFIER,
K_enc OCTET STRING,
K_mac OCTET STRING,
derivationAlgorithmK_enc OBJECT IDENTIFIER,
derivationAlgorithmK_mac OBJECT IDENTIFIER,
keySize INTEGER,
CHOICE {
SEQUENCE {
publicKeyMaterial OCTET STRING,
privateKey OCTET STRING
},
generateFlag NULL
},
nonceSize INTEGER
}
```

الف-۱۴-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته‌شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همانگونه که در الف-۱۴-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۴-۴ DIDUpdate

یک درخواست عمل DIDUpdate در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۴-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۴-۶ احراز هویت

این پروتکل احراز هویت، به‌وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData تعریف‌شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSession بایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۱۴-۶-۱ رویه

(۱) authenticationProtocolData ::= publicKeyMaterial OCTET STRING

(۲) بازیابی پارامترهای دامنه عمومی

(۳) اثبات مشروط حضور کاربر

---انتخاب تایید کلید و گواهی---

(۴) تایید Root CA مربوط به کلید عمومی و گواهی CA متعلق به SAL

verifySiganture (PuK.RCA.AUT)
verifyCertificate [PuK.RCA.AUT](C_CV.CA_{SAL}.CS_AUT)

تایید امضای کلید عمومی متعلق به گواهی احراز هویت CA مربوط به SAL و گواهی احراز هویت متعلق به SAL

verifySiganture (PuK.CA_{SAL}.CS_AUT)
verifyCertificate [PuK.CA_{SAL}.CS_AUT](C_CV.SAL.AUT)

ICC، PuK.SAL.AUT را در مکان موقتی، ذخیره می‌کند.

--- احراز هویت SAL (احراز هویت خارجی)---

(۵) SAL جفت کلید توافق کلید را به صورت تصادفی، تولید می‌کند

PuK.SAL.KA = RNG (keySize)
PrK.SAL.KA = RNG (keySize)

(۶) hash = hashAlgorithm (PuK.SAL.KA)

message = (keyId (PuK.SAL.AUT) || hash)

(۷) RND.ICC = RNG (nonceSize)

(۸) S = [RND.ICC || hashAlgorithm (PuK.SAL.KA)]

تایید امضاء signature = encryptionAlgorithm [PrK.SAL.AUT](S)

digest = encryptionAlgorithm [PuK.SAL.AUT](signature)

RND.ICC بازیابی می‌شود و با مورد اصلی آن، مقایسه می‌شود.

--- احراز هویت ICC (احراز هویت داخلی)---

(۹) تایید گواهی احراز هویت متعلق به ICC

verifyCertificate [PuK.ICC.KA](C.ICC.AUT)

(۱۰) SAL کلید توافق کلید را محاسبه می‌کند

$K_{SAL/ICC}$ = [PuK.ICC.KA * PrK.SAL.KA]
message = (keyId(PrK.ICC.KA) || PuK.SAL.KA)

(۱۱) ICC کلید توافق کلید را محاسبه می‌کند

hash =?= hashAlgorithm(PuK.SAL.KA)
$K_{ICC/SAL}$ = [PuK.SAL.KA * PrK.ICC.KA]

(۱۲) ICC، کلیدهای جلسه برای رمزنگاشتی و محاسبه MAC را تولید می‌کند

K_enc = derivationAlgorithmK_enc ()
K_mac = derivationAlgorithmK_mac ()

(۱۳) SAL، کلیدهای جلسه برای رمزنگاشتی و محاسبه MAC را تولید می‌کند

K_enc = derivationAlgorithmK_enc ()
K_mac = derivationAlgorithmK_mac ()

--- برقرار کردن جلسه ایمن---

authenticationProtocolData ::= result BOOLEAN as OCTET STRING (۱۴)

}

الف-۱۴-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایزکننده نامگذاری‌شده، درون برنامه کاربردی کارت، تغییری نمی‌کند.

الف-۱۴-۷ Encipher

outBuffer = encryptionAlgorithm [privateKey] (inBuffer)
outBuffer = encryptionAlgorithmForSessionKey [K_enc] (inBuffer)

الف-۱۴-۸ Decipher

outBuffer = encryptionAlgorithm [publicKey] (inBuffer)
outBuffer = $encryptionAlgorithmForSessionKey^{-1}$ [K_enc] (inBuffer)

الف-۱۴-۹ GetRandom

random = RNG (nonceSize)

الف-۱۴-۱۰ Hash

hash = hashAlgorithm (message)

الف-۱۴-۱۱ Sign

signature = encryptionAlgorithm [privateKey] (message)

الف-۱۴-۱۲ VerifySignature

message =?= encryptionAlgorithm [publicKey] (signature)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۱۴-۱۳ VerifyCertificate

منوط به نوع گواهی،

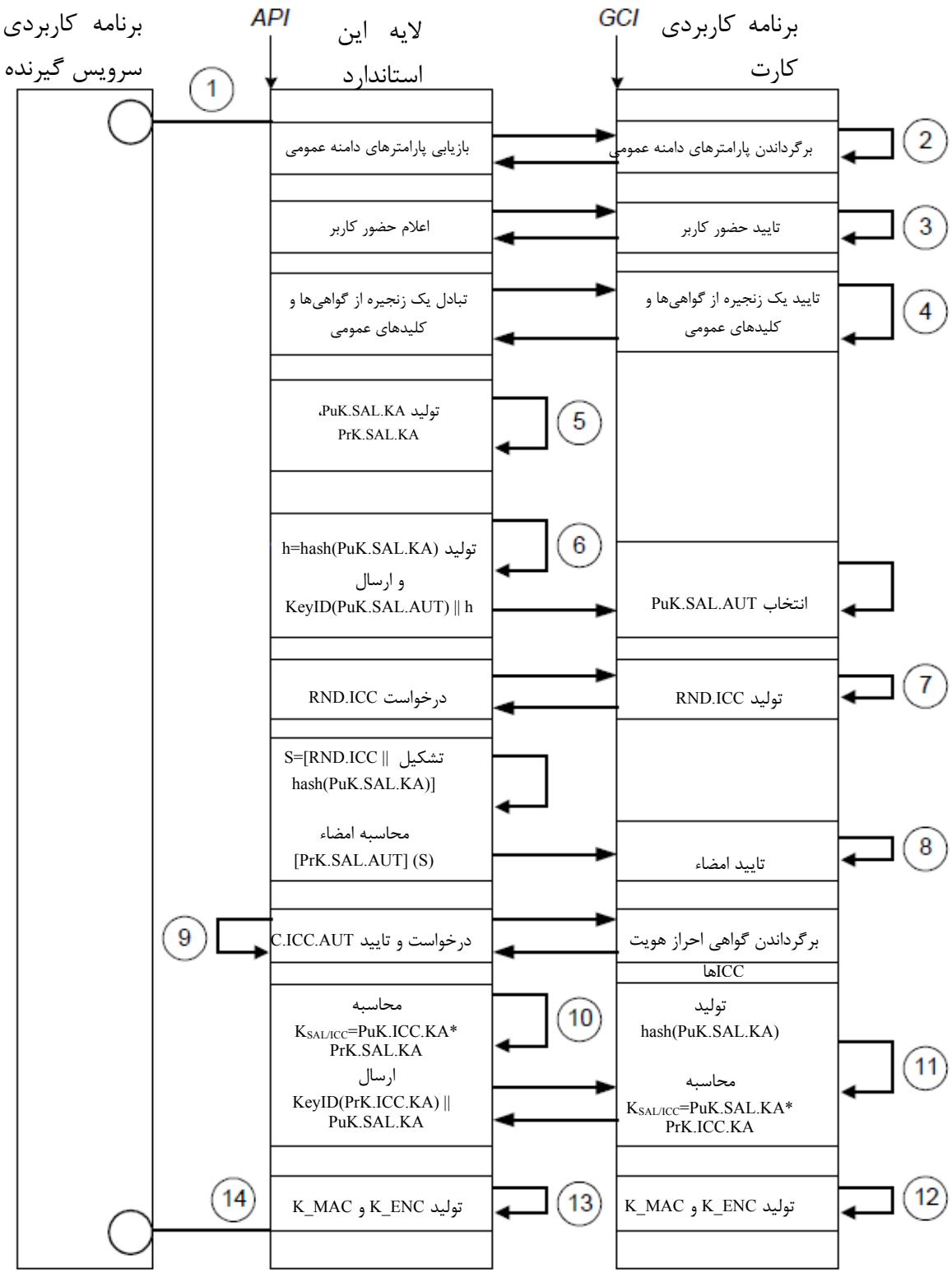
hash = hashAlgorithm (certificate without signature)
hash =?= encryptionAlgorithm [publicKey] (signature from certificate)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۱۵ انتقال کلید با احراز هویت دوطرفه مبتنی بر RSA

برنامه کاربردی سرویس‌گیرنده می‌تواند به وسیله فراخوانی Differential-IdentityAuthenticate به همراه یک DIDName، که دارای انتقال کلید با احراز هویت دوطرفه مبتنی بر RSA مشخص شده به عنوان پروتکل احراز هویت است، اجرای این پروتکل را آغاز نماید.

این پروتکل، بر اساس الگوی «پروتکل انتقال کلید مبتنی بر RSA» شرح داده‌شده در استاندارد prEN 14890-1 بند ۸-۴ شکل گرفته است.



شکل الف ۱۳- انتقال کلید با احراز هویت دوطرفه مبتنی بر RSA

الف-۱۵-۱ شناسانه شیء پروتکل

این پروتکل احراز هویت به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) key-transport-with-mutual-authentication(15)} شناسایی می‌شود.

الف-۱۵-۲ علامت‌گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP015 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
CHOICE {
SEQUENCE {
publicKey OCTET STRING,
privateKey OCTET STRING
},
generateFlag NULL
},
nonceSize INTEGER
}
```

الف-۱۵-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همانگونه که در الف-۱۵-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۵-۴ DIDUpdate

یک درخواست عمل DIDUpdate در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۵-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۵-۶ احراز هویت

این پروتکل احراز هویت، به‌وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData تعریف‌شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSession بایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۱۵-۶-۱ رویه

(۱) authenticationProtocolData ::= empty OCTET STRING

(۲) تایید چند مرحله‌ای کلیدهای عمومی و گواهی‌های SAL.

(۲-۱) تایید گواهی CA[1] با کلید عمومی CA ریشه

verifyCertificate [PuK.RCA.AUT] (C_CV.CA.CS_AUT)

1 - Certificate Authority

پس از موفقیت تایید، ICC، مقدار $PuK.CA_{SAL}.CS\_AUT$ درون گواهی را ذخیره می‌کند.

(۲-۲) تایید گواهی احراز هویت SAL

verifyCertificate [$PuK.CA_{SAL}.CS\_AUT$] (C_CV.SAL.AUT)

پس از موفقیت تایید، ICC، مقدار PuK.SAL.AUT درون گواهی را در یک مکان موقتی، ذخیره می‌کند.

(۳) تایید چند مرحله‌ای کلیدهای عمومی و گواهی‌های ICC.

SAL از ICC بازیابی می‌کند:

گواهی C.CA.AUT CA متعلق به ICC، حاوی کلید عمومی $PuK.CA_{ICC}.CS\_AUT$ متعلق به CA.

گواهی C.ICC.AUT متعلق به ICC، حاوی کلید عمومی PuK.ICC.AUT متعلق به ICC، مورد استفاده برای احراز هویت.

verifyCertificate [$PuK.CA_{ICC}.CS\_AUT$] (C.ICC.AUT)

پس از موفقیت تایید، SAL، مقدار PuK.ICC.AUT درون گواهی را به طور موقتی، ذخیره می‌کند.

(۴) ICC برای انتخاب کلید احراز هویت خصوصی PrK.ICC.AUT

(۵) ICC برای کلید عمومی متعلق به SAL، Puk.SAL.AUT برای رمزنگاشتی

--- احراز کردن هویت ICC---

(۶) SAL تولید و ارسال می‌کند

(8 LSB)شماره سریال SN.SAL = SAL

RND.SAL = RNG (nonceSize)
challenge = (RND.SAL || SN.SAL)

(۷) ICC یک امضاء، تولید می‌کند و پاسخ می‌دهد

اطلاعات نشانه کلید[1] متعلق به K_{ICC} = ICC

signature = encryptionAlgorithm [PrK.ICC.AUT](challenge || K_{ICC})
response = encryptionAlgorithm [PuK.ICC.AUT](signature)

(۸) SAL امضای درون پاسخ ICC را تایید می‌کند

signature = encryptionAlgorithm [PrK.ICC.AUT](response)
result =?= encryptionAlgorithm [PuK.ICC.AUT](signature)

--- احراز کردن هویت SAL---

(۹) SAL امضاء تولید می‌کند تا به وسیله ICC تایید شود

اطلاعات نشانه کلید متعلق به K_{SAL} = SAL

(8 LSB) شماره سریال SN.ICC = ICC

RND.ICC = RNG (nonceSize)

(۱۰) S = (RND.ICC || SN.ICC)

signature = encryptionAlgorithm [PrK.SAL.AUT] (K_{SAL} || S)
challenge = encryptionAlgorithm [PuK.ICC.AUT] (signature)

(۱۱) ICC تایید می‌کند که K_{SAL} با K_{ICC} متفاوت است و چالش فرستاده شده به وسیله SAL را تایید می‌کند

K_{SAL} =?= K_{ICC}
signature = encryptionAlgorithm [PrK.ICC.AUT](challenge)

1 - Key Token

result = encryptionAlgorithm [PuK.SAL.AUT](signature)

authenticationProtocolData ::= result BOOLEAN as OCTET STRING (۱۲)

}

الف-۱۵-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به هویت متمایز کننده نامگذاری شده، درون برنامه کاربردی کارت، تغییری نمی‌کند.

الف-۱۵-۷ Encipher

outBuffer = encryptionAlgorithm [privateKey] (inBuffer)

الف-۱۵-۸ Decipher

outBuffer = encryptionAlgorithm [publicKey] (inBuffer)

الف-۱۵-۹ GetRandom

random = RNG (nonceSize)

الف-۱۵-۱۰ Hash

hash = hashAlgorithm (message)

الف-۱۵-۱۱ Sign

signature = encryptionAlgorithm [privateKey] (message)

الف-۱۵-۱۲ VerifySignature

message =?= encryptionAlgorithm [publicKey] (signature)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت API_INVALID_SIGNATURE را برمی‌گرداند.

الف-۱۵-۱۳ VerifyCertificate

منوط به certificateType،

hash = hashAlgorithm (certificate without signature)
hash =?= encryptionAlgorithm [publicKey] (signature from certificate)

اگر TRUE باشد، API_OK را برمی‌گرداند، در غیر این صورت API_INVALID_SIGNATURE را برمی‌گرداند.

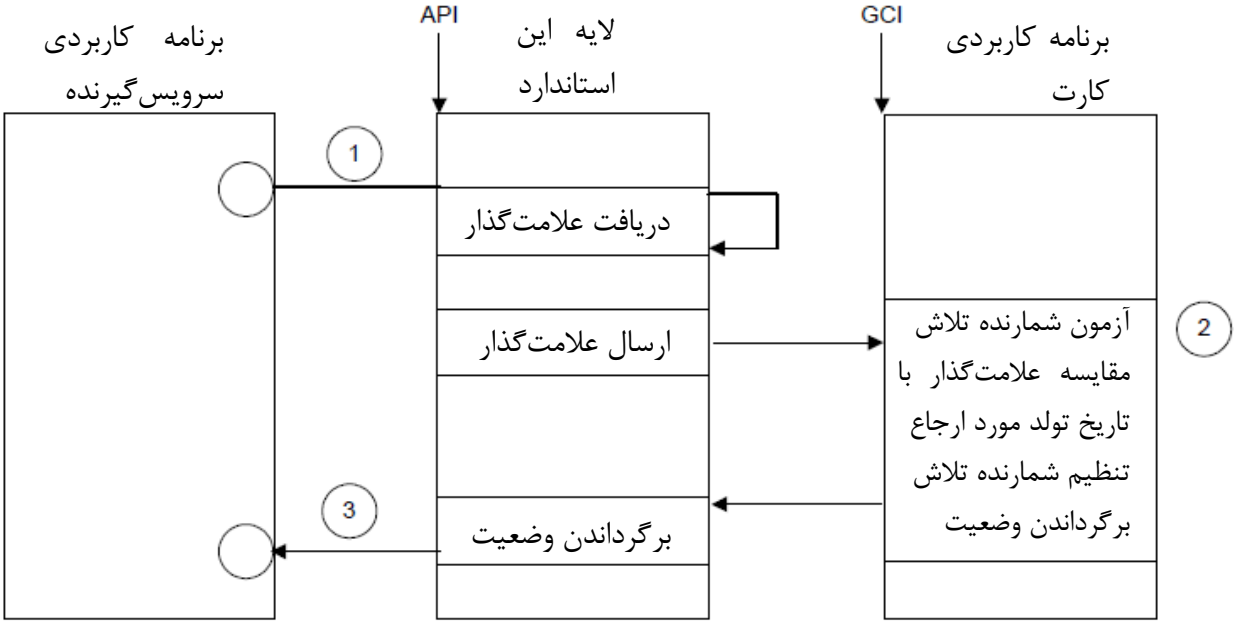
الف-۱۶ دستیابی به سن

هدف این پروتکل، امکان دادن به یک صاحب کارت برای اثبات این است که او به یک سن مشخصی رسیده است، بدون اینکه تاریخ تولد او فاش شود.

ممکن است که بعضی از پیاده‌سازی‌های کارت هوشمند، تاریخ تولد صاحب کارت را ذخیره کنند. اغلب، تاریخ تولد، محرمانه تلقی می‌شود، با این حال، موقعیت‌هایی وجود دارد که در آن‌ها صاحب کارت باید اثبات کند که به سن مشخصی رسیده است، مثلا برای دسترسی به خدماتی که برای آن‌ها محدودیت سنی، قرار داده شده است. این پروتکل احراز هویت برای نشان دادن اینکه صاحب کارت، به سن مشخصی رسیده است بدون فاش شدن تاریخ تولد واقعی او، طراحی شده است.

این پروتکل احراز هویت، یک تاریخ ورودی را با تاریخ تولد ذخیره شده در یک DSI، مقایسه می‌کند و در صورتی که تاریخ ورودی، برابر با یا جدیدتر از آن تاریخ تولد باشد، نتیجه موفقیت‌آمیز برمی‌گرداند.

مقدار آن تاریخ تولد هرگز نباید به وسیله این پروتکل احراز هویت، فاش شود.



شکل الف-۱۴ دستیابی به سن

الف-۱۶-۱ شناسانه شیء پروتکل

این پروتکل احراز هویت به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) age-attainment(16)} شناسایی می‌شود.

الف-۱۶-۲ علامت گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، باید علامت‌گذار زیر را داشته باشد.

```
MarkerAP016 ::= SEQUENCE {
dateOfBirthReference DSIReference,
attainedDate OCTET STRING
}
```

در حالی‌که dateofBirthReference باید به تاریخ ذخیره‌شده درون برنامه کاربردی کارت، ارجاع کند. ساختار این ارجاع به صورت زیر تعریف می‌شود:

```
DSIReference ::= SEQUENCE {
aID APPLICATION IDENTIFIER,
dataSetName DataSetName,
dSIName DSIName
}
```

attainedDate و تاریخ تولد ذخیره شده در DSI که به‌وسیله dateOfBirthReference به آن ارجاع شده است باید به صورت yyyymmdd باشد.

الف-۱۶-۳ DIDCreate

یک درخواست عمل DIDCreate مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۱۶-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۶-۴ DIDUpdate

یک درخواست عمل DIDUpdate در هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۱۶-۲ تعریف شده، marker، علامت-گذار خاص این پروتکل است.

الف-۱۶-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۶-۶ احراز هویت

این پروتکل احراز هویت، به وسیله یک درخواست تکی عمل DIDAuthenticate با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل CardApplicationStartSession بایدکدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTIO را برگرداند.

الف-۱۶-۶-۱ رویه

(۱) authenticationProtocolData ::= attainedDate OCTET STRING

attainedDate باید به همان صورت تاریخ تولد ارجاع شده درون dateOfBirthReference، به صورت yyyymmdd قالب بندی شود.

(۲) اطمینان از اینکه وضعیت احراز هویت، تنظیم نشده است، زیرا برای هر جلسه، باید فقط یک تلاش احراز هویت، مجاز باشد.

مقایسه attainedDate با تاریخ ذخیره شده در ارجاع DSI درون فیلد dateOfBirthReference.

اگر attainedDate برابر با یا جدیدتر از آن تاریخ تولد باشد، نتیجه موفقیت‌آمیز برگردانده می‌شود. در غیر این صورت، نتیجه شکست، برگردانده می‌شود.

(۳) برگرداندن وضعیت

الف-۱۶-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام این پروتکل احراز هویت، موفقیت‌آمیز یا ناموفق، وضعیت احراز هویت مربوط به این هویت متمایز کننده، باید به "true" تنظیم شود. این باید نشان دهد که احراز هویت دستیابی به سن، اتفاق افتاده است و نمی-تواند برای این جلسه، تکرار شود.

الف-۱۶-۷ Encipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۶-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست باید کدبازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۶-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۶-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۶-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۶-۱۲ VerifySignature

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۶-۱۳ VerifyCertificate

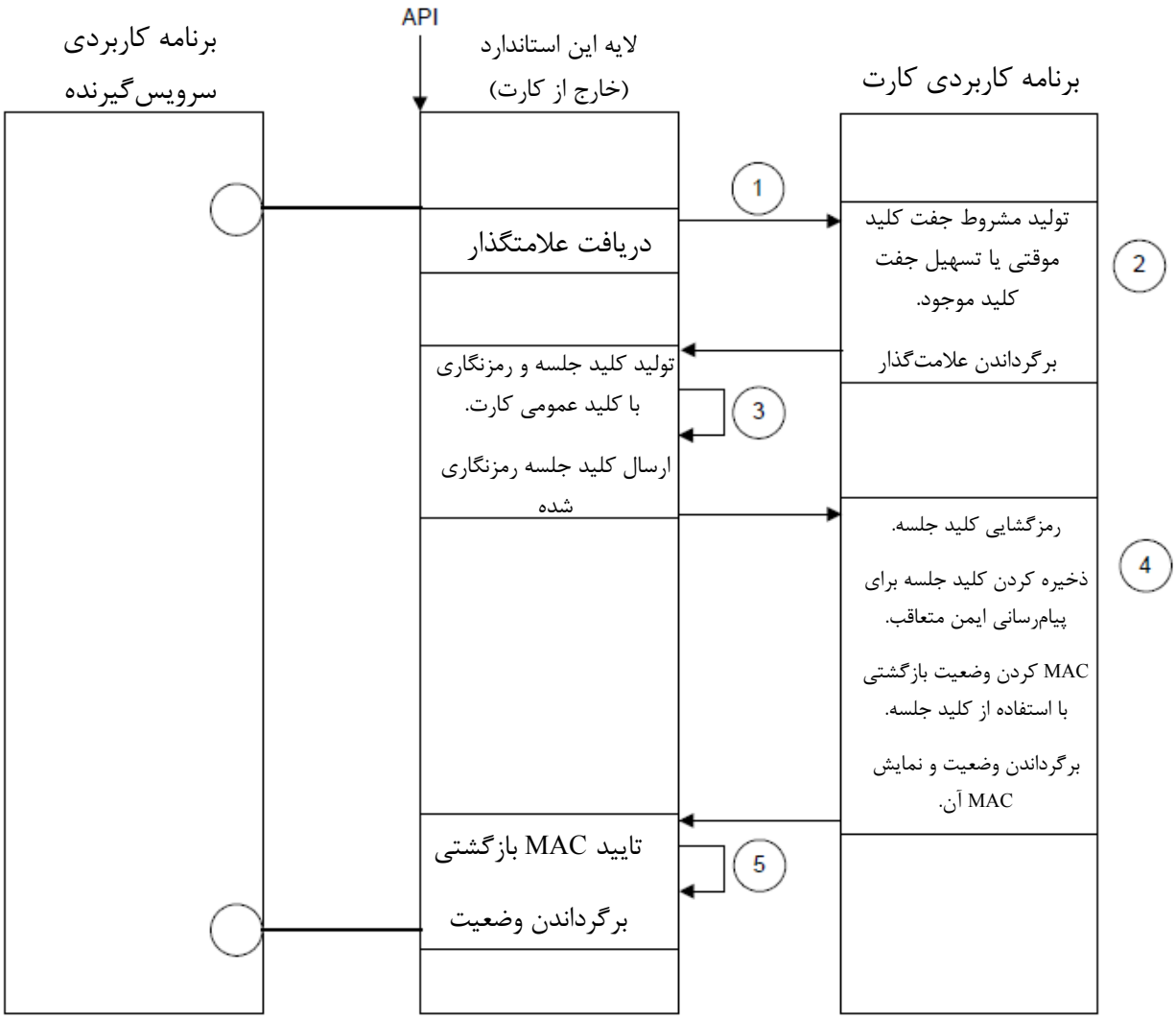
در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷ برقراری نامتقارن کلید جلسه

این پروتکل، به وسیله تسهیل رمزنگاشتی نامتقارن برای انتقال کلیدهای جلسه تولید شده خارج از کارت، به کارت، کلیدهای جلسه متقارن را برقرار می‌کند.

یک سازوکار اختیاری برای ذخیره کردن یک جفت کلید نامتقارن ایستا (شامل یک گواهی دیجیتال) یا یک جفت کلید نامتقارن موقتی، برای صدور مجوز برای تقویت برقراری کلید جلسه، با احراز هویت‌های کارت نامتقارن یا بدون آن را انعطاف پذیر می‌سازد.

در مورد یک جفت کلید ایستا و گواهی دیجیتال، ممکن است که برنامه کاربردی سرویس‌گیرنده، اصالت کارت را بررسی نماید.



شکل الف-۱۵- برقراری نامتقارن کلید جلسه

الف-۱۷-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) asymmetric-session-key-establishment(17)} شناسایی می‌شود.

الف-۱۷-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP017 ::= SEQUENCE {
encryptionAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
keyTransportProtectionType INTEGER,
privateTransKeyReference DIDReference
CHOICE {
SEQUENCE {
privateKey OCTET STRING,
publicKeyMaterial OCTET STRING
},
SEQUENCE {
privateKeyReference DIDReference,
```

```
publicKeyReference DIDReference
},
generateFlag BOOLEAN
},
sessionMACAlgorithm OBJECT IDENTIFIER,
sessionENCAlgorithm OBJECT IDENTIFIER
}
```

در حالی که dateofBirthReference باید به تاریخ ذخیره شده درون برنامه کاربردی کارت، ارجاع کند. ساختار این ارجاع به صورت زیر تعریف می‌شود:

```
DSIReference ::= SEQUENCE {
aID APPLICATION IDENTIFIER,
dataSetName DataSetName,
dSIName DSIName
}
```

keyTransportProtectionType باید نشان دهد که آیا کلید خصوصی به صورت متن واضح (0) یا رمزنگاشتی شده (1) منتقل می‌شود. publicKeyMaterial می‌تواند یک کلید عمومی یا یک گواهی دیجیتال حاوی یک کلید عمومی باشد.هریک از پارامترهای privateKeyReference، publicKeyMaterialReference و privateTransKeyReference به یک کلید رمزنگاشتی، ارجاع می‌دهند. ساختار این مرجع، به صورت تعریف شده در زیر است:

```
DIDReference ::= SEQUENCE {
scope DIDScope,
dIDName DIDName
}
```

الف-۱۷-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۱۷-۲ تعریف‌شده، marker، علامت‌گذار خاص این پروتکل است.

در موردی که یک جفت کلید خصوصی و عمومی، خارج از برنامه کاربردی سرویس‌گیرنده، ایجاد شده باشند و کلید خصوصی قرار است در حین عمل DIDCreate بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، به صورت محرمانه منتقل شود (یعنی keyTransportProtectionType به (1) «رمزنگاشتی‌شده» تنظیم‌شده است)، موارد زیر به کار می‌روند:

- کلید خصوصی باید با استفاده از قالب ترکیب انتقال ایمن (به الف-۱۷-۱۵ مراجعه شود)، بسته بندی شود
- یک کلید خصوصی موجود باید درون پارامتر privateTransKeyReference ارجاع شود
- کلید عمومی متناظر با privateTransKeyReference باید در زمان ایجاد بسته ترکیب انتقال ایمن (به الف-۱۷-۱۵ مراجعه شود)، مورد استفاده قرار گیرد
- پیاده‌سازی این پروتکل احراز هویت مربوط به این برنامه کاربردی کارت باید بسته ترکیب انتقال ایمن (به الف-۱۷-۱۵ مراجعه شود) را به منظور استخراج کلید خصوصی برای ذخیره‌سازی در DID، پردازش نماید.

اگر generateFlag به TRUE تنظیم شود، آنگاه باید کلید عمومی و کلید خصوصی، در هر عمل DIDAuthenticate تولید شود.

اگر keyTransportProtectionType به رمزنگاشتی شده (1) تنظیم شود، آنگاه کلید خصوصی باید یک ارجاع به یک کلید موجود (هویت متمایز کننده) باشد که در دسترس برنامه کاربردی سرویس‌گیرنده قرار دارد.

الف-۱۷-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۷-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

فیلدهای ساختار marker دقیقا به همان صورت توصیف شده برای عمل DIDCreate در بالا، تفسیر می‌شوند.

الف-۱۷-۵ DIDGet

تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر didStructure باشد.

برای این پروتکل بخصوص، اگر generateFlag به TRUE تنظیم شود، آنگاه نباید هیچ اطلاعات کلید عمومی یا ارجاعی به اطلاعات یک کلید عمومی، برگردانده شود. اگر generateFlag تنظیم نشود، آنگاه جفت کلید نامتقارن، ایستا یا ارجاع شده، است و publicKeyMaterial باید برگردانده شود.

الف-۱۷-۶ احراز هویت

این پروتکل،به وسیله یک درخواست تکی عمل CardApplicationStartSession با یک پارامتر authenticationProtocolData خالی، به صورت تعریف شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل DIDAuthentication باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۶-۱ رویه

```
authenticationProtocoldata ::= SEQUENCE { (۰)
certVerificationProcess INTEGER,
crlAddress OCTET STRING OPTIONAL,
ocspAddress OCTET STRING OPTIONAL
}
```

به طوری که پارامتر certVerificationProcess یک نشان‌دهنده شامل یک پوشش بیتی استخراج شده از مقدارهای زیر است:

- (0) فاقد تایید
- (1) تایید امضاء
- (2) تایید تاریخ انقضاء
- (4) بررسی CRL
- (8) بررسی پاسخ دهنده OCSP

یعنی یک تاییدیه که نیازمند تایید امضاء (1)، تایید تاریخ انقضاء (2) و مشاوره یک پاسخ دهنده OCSP (8) می‌باشد، یک پوشش بیتی 11 (8+2+1) را تشکیل خواهد داد.

(۱) SAL باید علامت‌گذار برقراری نامتقارن کلید جلسه را درخواست کند.

(۲) اگر علامت‌گذار generateFlag DID به TRUE تنظیم شود، باید یک جفت کلید نامتقارن تولید شود و کلید عمومی باید در علامت‌گذار، برگردانده شود.

اگر علامت‌گذار generateFlag DID تنظیم نشود، باید publicKeyMaterial ایستا، که ممکن است یک کلید عمومی نامتقارن یا یک گواهی دیجیتال باشد، در علامت‌گذار، برگردانده شود.

(۳) در صورت ارائه گواهی دیجیتال، ممکن است که SAL به طور اختیاری، آن را مطابق با نشان‌دهنده certVerificationProcess تایید کند.

SAL باید یک کلید جلسه متقارن، تولید کند که باید با کلید عمومی درون علامت‌گذار DID، رمزنگاشتی شود (یعنی در صورت ارائه، کلید عمومی به طور مستقیم، یا در صورت ارائه، کلید عمومی درون گواهی دیجیتال).

```
K_Mac = sessionMacAlgorithm()
K_Enc = sessionEncAlgorithm()
sessionKeys ::= SEQUENCE {
K_Mac OCTET STRING,
K_Enc OCTET STRING
}
```

اگر علامت‌گذار برگردانده شده، حاوی کلید عمومی باشد آنگاه

encryptedData = encryptionAlgorithm(publicKey, sessionKeys)

در غیر این صورت، اگر علامت‌گذار برگردانده شده، حاوی گواهی دیجیتال باشد

encryptedData = encryptionAlgorithm(digitalCertificate(publicKey),sessionKeys)

SAL باید کلیدهای جلسه رمزنگاشتی شده را برگرداند.

(۴) برنامه کاربردی کارت، با استفاده از کلید خصوصی، کلیدهای جلسه رمزنگاشتی شده را رمزگشایی می‌کند و کلیدهای جلسه را برای پیام‌رسانی ایمن بعدی استاندارد ملی ایران شماره ۱۶۳۸۶، ذخیره می‌نماید.

sessionKeys = $\text{encryptionAlgorithm}^{-1}$ (privateKey, encryptedData)

status code کردن MAC

macData = macAlgorithm(K_Mac, statusCode)

برنامه کاربردی کارت باید یک status code و نیز macData را برگرداند.

(۵) SAL، macData برگردانده شده را بررسی می‌کند تا اصالت کلید خصوصی ارجاع شده را تصدیق نماید.

الف-۱۷-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل احراز هویت، وضعیت احراز هویت مربوط به این هویت متمایزکننده نامگذاری‌شده، باید درون برنامه کاربردی کارت، به TRUE تنظیم شود.

الف-۱۷-۷ Encipher

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۱۲ VerifySignature

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۱۳ VerifyCertificate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۷-۱۴ CardApplicationEndSession

درخواست این عمل، پس از اینکه هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، احراز هویت شد، باید وضعیت احراز هویت مربوط به این هویت متمایز کننده را به FALSE تنظیم کند و کد بازگشتی API_OK را برگرداند.

الف-۱۷-۱۵ ترکیب انتقال ایمن

یادآوری: این ترکیب، تطبیقی از نوع محتوای envelopedData PKCS#7- RFC 2315 می‌باشد.

الف-۱۷-۱۵-۱ تعریف بسته انتقال

```
ContentInfo ::= SEQUENCE {
contentType ContentType,
content EnvelopedData
}
ContentType ::= OBJECT IDENTIFIER
EnvelopeDataVersion ::= INTEGER (0)
CertificateSerialNumber ::= INTEGER
KeyEncryptionAlgorithmIdentifier ::= AlgorithmIdentifier
ContentEncryptionAlgorithmIdentifier ::= AlgorithmIdentifier
AlgorithmIdentifier ::= SEQUENCE {
algorithm OBJECT IDENTIFIER,
parameters ANY DEFINED BY algorithm OPTIONAL
}
EnvelopedData ::= SEQUENCE {
version EnvelopeDataVersion,
recipientInfos RecipientInfos,
encryptedContentInfo EncryptedContentInfo
}
RecipientInfos ::= SET OF RecipientInfo
RecipientInfo ::= SEQUENCE {
```

```
version EnvelopeDataVersion,
issuerAndSerialNumber IssuerAndSerialNumber,
keyEncryptionAlgorithm KeyEncryptionAlgorithmIdentifier,
encryptedKey EncryptedKey
}
IssuerAndSerialNumber ::= SEQUENCE {
issuer Name,
serialNumber CertificateSerialNumber
}
EncryptedKey ::= OCTET STRING
EncryptedContentInfo ::= SEQUENCE {
contentType ContentType,
contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier,
encryptedContent [0] IMPLICIT EncryptedContent OPTIONAL
}
EncryptedContent ::= OCTET STRING
```

الف-۱۷-۱۵-۲ توضیحات

ContentInfo – contentType: نشان‌دهنده نوع محتوا است. این استاندارد، یک نوع محتوای envelopedData را تعریف می‌کند که باید یک شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) asymmetric-session-key-establishment(17) envelopedData(1)} داشته باشد.

ContentInfo – content: محتوایی که قرار است منتقل شود. در این مورد باید envelopedData باشد.

EnvelopedData – EnvelopeDataVersion: شماره نسخه ترکیب است. برای این نسخه از قالب داده دربرگرفته شده[1]، باید 0 باشد.

EnvelopedData - recipientInfos: مجموعه‌ای از اطلاعات مختص-هر-گیرنده است. در این مجموعه، باید حداقل یک جزء وجود داشته باشد.

EnvelopedData- encryptedContentInfo: اطلاعات محتوای رمزنگاشتی شده، می‌باشد.

RecipientInfo-version: شماره نسخه ترکیب است. برای این نسخه از این استاندارد، باید 0 باشد.

RecipientInfo-issuerAndSerialNumber گواهی دریافت‌کننده (و در نتیجه نام متمایز دریافت‌کننده و کلید عمومی) را به وسیله نام متمایز صادرکننده و شماره سریال مختص صادرکننده، مشخص می‌کند.

RecipientInfo-keyEncryptionAlgorithm: الگوریتم کلید-رمزنگاشتی (و پارامترهای مربوطه) را که تحت کلید محتوا-رمزنگاشتی با کلید عمومی دریافت‌کننده، رمزنگاشتی می‌شود، مشخص می‌نماید.

RecipientInfo-encryptedKey حاصل رمزنگاشتی کردن کلید محتوا-رمزنگاشتی با کلید عمومی دریافت‌کننده است.

EncryptedContentInfo-contentType: نشان‌دهنده نوع محتوا است. این استاندارد، یک نوع محتوای envelopedData را تعریف می‌کند که باید یک شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) asymmetric-session-key-establishment(17) envelopedData(1)} داشته باشد.

EncryptedContentInfo-contentEncryptionAlgorithm: الگوریتم محتوا-رمزنگاشتی (و پارامترهای مربوطه) را که تحت آن، محتوا رمزنگاشتی می‌شود، مشخص می‌نماید.

EncryptedContentInfo-encryptedContent: حاصل رمزنگاشتی کردن محتوا است.

1- Enveloped

الف-۱۷-۱۵-۳ فرآیند ساختن بسته

تولید یک کلید محتوا-رمزنگاشتی برای یک الگوریتم محتوا-رمزنگاشتی مشخص، به طور تصادفی.

رمزنگاشتی کلید رمزنگاشتی محتوا با کلید عمومی دریافت‌کننده (یعنی برنامه کاربردی سرویس‌گیرنده).

تلفیق کلید رمزنگاشتی محتوای رمزنگاشتی شده و سایر اطلاعات مختص دریافت‌کننده، درون مقدار RecipientInfo.

رمزنگاشتی محتوا با کلید رمزنگاشتی (در صورت نیاز، از پر کردن، استفاده شود).

تلفیق مقدارهای RecipientInfo با محتوای رمزنگاشتی شده درون یک مقدار EnvelopedData.

الف-۱۷-۱۵-۴ فرآیند تفسیر بسته

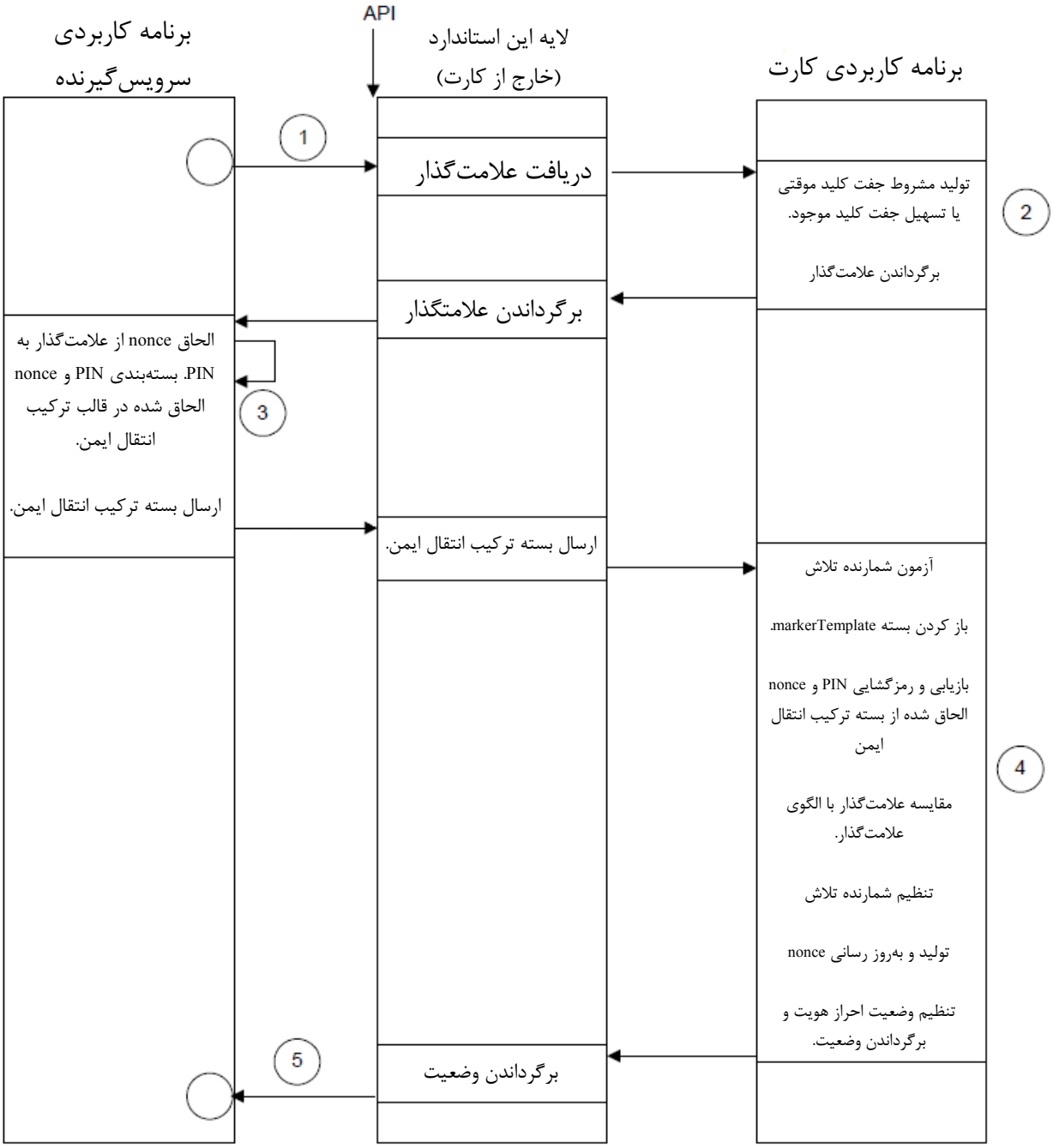
کلید رمزنگاشتی محتوای رمزنگاشتی شده با کلید خصوصی دریافت‌کننده (یعنی کارت هوشمند)، رمزگشایی می-شود.

محتوای رمزنگاشتی شده با کلید رمزنگاشتی محتوای بازیابی شده، رمزگشایی می‌شود.

الف-۱۸ مقایسه PIN ایمن

این پروتکل احراز هویت، برای انتقال به طور ایمن و ناشناس PIN، ترکیب انتقال ایمن (به الف-۱۷-۱۴ مراجعه کنید) را به کار می‌برد. برنامه کاربردی کارت باید PIN را از بسته ترکیب انتقال ایمن (به الف-۱۷-۱۴ مراجعه کنید) استخراج کند و آن را با یک مقدار ذخیره شده در علامت‌گذار یک هویت متمایز کننده، مقایسه نماید.

مقدار attemptsCounter در این پروتکل احراز هویت، برای تسهیل قفل کردن پروتکل احراز هویت، باید قابل به‌روز رسانی باشد (یعنی تنظیم attemptsCounter به maxAttempts).



شکل الف-۱۶- مقایسه PIN ایمن

الف-۱۸-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) secure-pin-compare(18)} شناسایی می‌شود.

الف-۱۸-۲ علامت‌گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
markerAP018 ::= SEQUENCE {
minDataLength INTEGER,
```

```
maxDataLength INTEGER,
paddingCharacter OCTET STRING,
markerTemplate OCTET STRING,
maxAttempts INTEGER,
attemptsCounter INTEGER,
encryptionAlgorithm OBJECT IDENTIFIER,
keySize INTEGER,
keyTransportProtectionType INTEGER,
nonceSize INTEGER,
nonce INTEGER,
CHOICE {
SEQUENCE {
privateKey OCTET STRING,
publicKeyMaterial OCTET STRING
},
SEQUENCE {
privateKeyReference DIDReference,
publicKeyMaterialReference DIDReference
},
generateFlag NULL
}
}
```

keyTransportProtectionType باید نشان دهد که آیا در حین عمل DIDCreate یا عمل DIDUpdate، markerTemplate به صورت متن واضح (0) یا رمزنگاشتی شده (1) منتقل می‌شود.

پارامترهای privateKeyReference و publicKeyMaterialReference باید یک کلید رمزنگاشتی شناخته شده به وسیله یک هویت متمایزکننده را ارجاع دهند. ساختار این مرجع، به صورت تعریف شده در زیر است:

```
DIDReference ::= SEQUENCE {
scope DIDScope,
dIDName DIDName
}
```

الف-۱۸-۳ DIDCreate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۱۸-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

در موردی که markerTemplate قرار است در حین عمل DIDCreate بین برنامه کاربردی سرویس‌گیرنده و برنامه کاربردی کارت، به صورت محرمانه منتقل شود (یعنی keyTransportProtectionType به «رمزنگاشتی‌شده» تنظیم شده است)، موارد زیر به کار می‌رود:

- markerTemplate باید با استفاده از قالب ترکیب انتقال ایمن (به الف-۱۷-۱۴ مراجعه شود)، بسته بندی شود
- یک کلید خصوصی ایستا و یک کلید عمومی باید به ترتیب در پارامتر privateKeyReference و پارامتر publicKeyMaterialReference مورد ارجاع واقع شوند؛
- کلید عمومی ارجاع شده با publicKeyMaterialReference باید در زمان ایجاد بسته ترکیب انتقال ایمن (به الف-۱۷-۱۴ مراجعه شود)، مورد استفاده قرار گیرد

- پیاده‌سازی این پروتکل احراز هویت مربوط به این برنامه کاربردی کارت باید بسته ترکیب انتقال ایمن (به الف-۱۷-۱۴ مراجعه شود) را به منظور استخراج markerTemplate برای ذخیره‌سازی در DID، پردازش نماید.

در صورتی که generateFlag به "true" تنظیم شود، markerTemplate نباید به طور محرمانه منتقل شود و در حین عمل DIDCreate، بسته‌بندی ایمن، ضروری نیست.

الف-۱۸-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۱۸-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است. فیلدهای ساختار marker دقیقا به همان صورت توصیف شده برای عمل DIDCreate در بالا، تفسیر می‌شوند.

شمارنده attemptsCounter ممکن است برای غیرفعال کردن یک هویت متمایز کننده که از این پروتکل احراز هویت، استفاده می‌کند، به maxAttempts تنظیم شود.

الف-۱۸-۵ DIDGet

یک درخواست عمل DIDGet در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۸-۶ احراز هویت

این پروتکل احراز هویت، باید به وسیله دو درخواست اجرا شود. اولین درخواست، عمل DIDGet است که علامتگذار حاوی کلید عمومی مورد نیاز را برمی‌گرداند. درخواست دوم، عمل DIDAuthenticate با یک پارامتر authenticationProtocolData به صورت تعریف شده در زیر، است.

درهویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند،یک درخواست عمل CardApplicationStartSession() بایدکدبازگشتیAPI_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۸-۶-۱ رویه

(۱) درخواست برای بازیابی علامت‌گذاری که حاوی کلید عمومی است.

(۲) اگر علامت‌گذار generateFlag DID به TRUE تنظیم شود، یک جفت کلید نامتقارن به وسیله برنامه کاربردی کارت باید برای جلسه جاری، تولید شود.

اگر علامت‌گذار generateFlag DID به FALSE تنظیم شود، جفت کلید ارجاع شده باید به وسیله برنامه کاربردی کارت، برای جلسه جاری، مورد استفاده قرار گیرد.

علامت‌گذار حاوی اطلاعات کلید عمومی و nonce جاری باید به برنامه کاربردی سرویس‌گیرنده برگردانده شوند.

(۳) تایید این که attemptsCounter کوچکتر از maxAttempts است. برنامه کاربردی سرویس‌گیرنده باید یک کلید متقارن، تولید کند و باید PIN الحاق شده با nonce را با استفاده از کلید متقارن تولید شده و کلید عمومی ارائه شده به‌وسیله برنامه کاربردی کارت، به صورت مشخص‌شده به‌وسیله ترکیب انتقال ایمن تعریف‌شده در الف-۱۷-۱۴، بسته‌بندی نماید.

markerTemplate، به همراه بسته ترکیب انتقال ایمن (به الف-۱۷-۱۴ مراجعه کنید)، به برنامه کاربردی کارت، برگردانده می‌شود.

authenticationProtocolData ::= ContentInfo(secure transport syntax A.17.14, 7. General syntax)

(۴) اگر مقدار attemptsCounter بزرگتر از یا برابر با مقدار maxAttempts باشد، این درخواست باید از مرحله شماره ۵، ادامه یابد.

برنامه کاربردی کارت، الگوی ترکیب انتقال ایمن (به الف-۱۷-۱۴ مراجعه کنید) را با استفاده از privateKey از بسته‌بندی خارج می‌کند تا encryptedKey درون ContentInfo مربوط به الگوی ترکیب انتقال ایمن را رمزگشایی کند.

encryptedKey به همراه ContentInfo برای رمزگشایی PIN و nonce الحاق شده، به کار می‌رود.

result = (decryptedPIN =?= markerTemplate)

مقایسه PIN و nonce الحاق شده، به صورت بیت به بیت است.

اگر حاصل، TRUE باشد، attemptsCounter به صفر بازتنظیم می‌شود. در غیر این صورت، مقدار attemptsCounter افزایش می‌یابد.

تولید و ذخیره کردن nonce جدید.

nonce = RNG(nonceSize)

(۵) authenticationProtocolData ::= SEQUENCE {

maxAttempts INTEGER OPTIONAL,
attemptsCounter INTEGER OPTIONAL
}

الف-۱۸-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل، وضعیت احراز هویت مربوط به این هویت متمایز کننده نامگذاری شده، باید درون برنامه کاربردی کارت، به مقدار نتیجه، تنظیم شود.

الف-۱۸-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۸-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۸-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۸-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۸-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۸-۱۲ VerifySignature

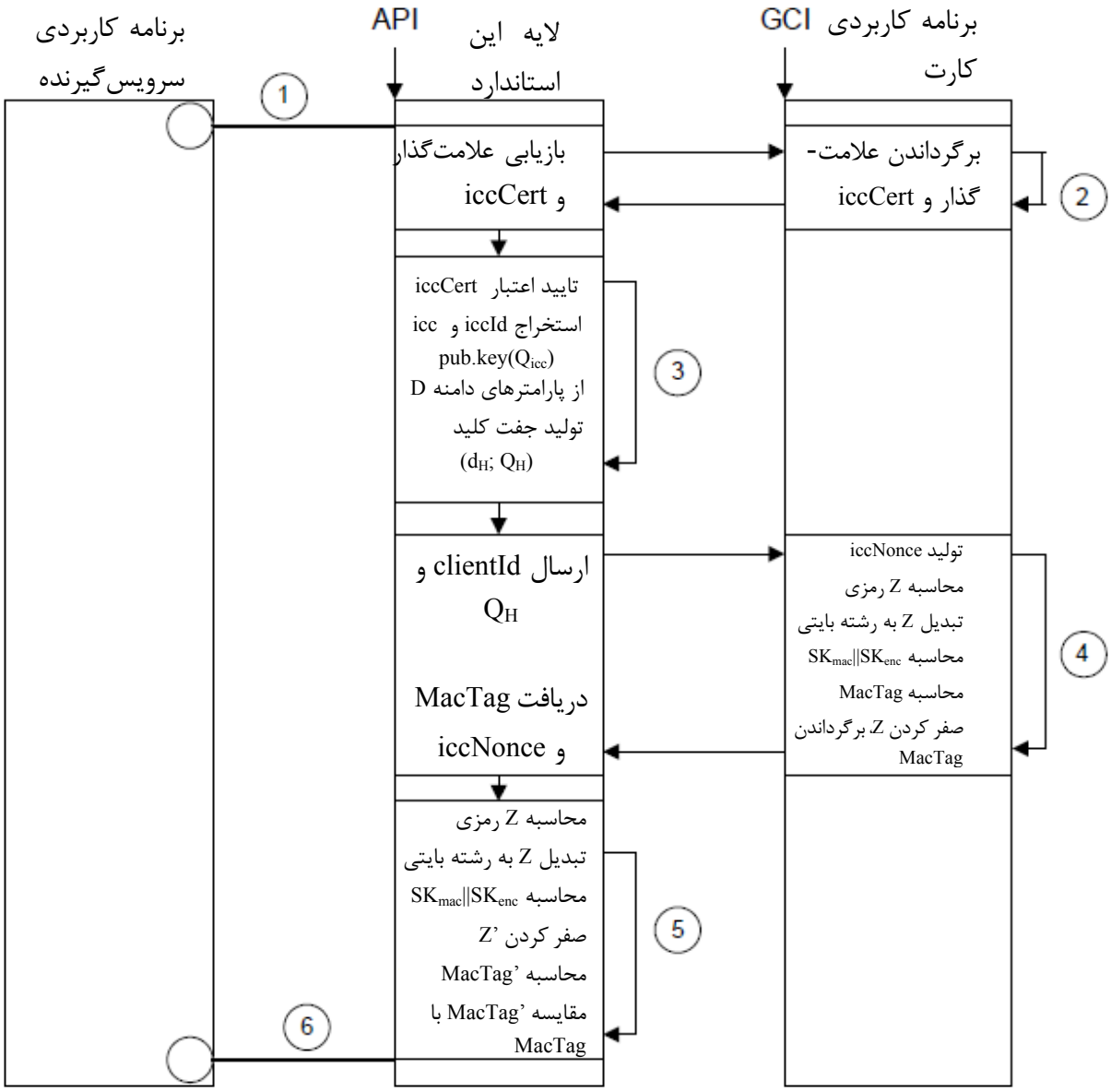
در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۸-۱۳ VerifyCertificate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۹ توافق کلید EC با احراز هویت برنامه کاربردی کارت

این پروتکل احراز هویت، یک پروتکل احراز هویت داخلی برنامه کاربردی کارت و توافق کلید با استفاده از رمزنگاشتی EC است. آغازگر، یک جفت کلید موقتی، تولید می‌کند ولی جفت کلید ایستا ندارد؛ پاسخ‌دهنده، فقط یک جفت کلید ایستا دارد. این پروتکل برای برقرار کردن احراز هویت (SKmac) و رمزنگاشتی (SKenc) کلیدهای جلسه برای پیام‌رسانی ایمن‌تر بین لایه این استاندارد و برنامه کاربردی کارت، استفاده می‌شود.



شکل الف-۱۷- توافق کلید EC با احراز هویت برنامه کاربردی کارت

الف-۱۹-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) ec-key-agreement-with-card-application-authentication(19)} شناسایی می‌شود.

الف-۱۹-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP019 ::= SEQUENCE {
domainParameters OBJECT IDENTIFIER,
keyEstablishmentAlgorithm OBJECT IDENTIFIER,
kDFHashAlgorithm OBJECT IDENTIFIER,
sessionMacAlgorithm OBJECT IDENTIFIER,
sessionEncAlgorithm OBJECT IDENTIFIER,
nonceSize INTEGER,
```

```
CHOICE {
SEQUENCE {
iccPublicKey OCTET STRING,
iccPrivateKey OCTET STRING
},
genKeyPairFlag NULL
}
iccIdentifier OCTET STRING,
iccCert OCTET STRING
}
```

الف-۱۹-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۹-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۹-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۱۹-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۱۹-۵ DIDGet

یک تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۱۹-۶ احراز هویت

این پروتکل، به وسیله یک درخواست ساده عمل CardApplicationStartSession با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

الف-۱۹-۶-۱ رویه

(۱) authenticationProtocolData ::= clientIdentifier OCTET STRING

(۲) تایید اعتبار iccCert. تولید جفت کلید موقتی $(d_H;Q_H)$ از پارامترهای دامنه D. تایید این که Q_H برای پارامترهای دامنه D معتبر است.

(۳) iccNonce = RNG(nonceSize)

$Z = keyEstablishmentAlgorithm(d_{ICC};Q_H)$

تبدیل Z به یک رشته بایتی با استفاده از field-element-to-byte-string.

محاسبه کلیدهای جلسه با استفاده از تابع مشتق‌گیری:

$SK_{mac} \| SK_{enc} = KDF(Z, keyDataLen, otherInfo)$

به‌طوری‌که:

```
otherInfo ::= SEQUENCE {
sessionMacAlgorithm OBJECT IDENTIFIER,
sessionEncAlgorithm OBJECT IDENTIFIER,
clientIdentifier OCTET STRING,
iccIdentifier OCTET STRING,
iccNonce OCTET STRING
```

}
keyDataLen = algoKeyLengthInBits(seesionMacAlgorithm) +
algoKeyLengthInBits(sessionEncAlgorithm)
N = keyDataLen / (hashLengthInBits(kDFHashAlgorithm))
DerivedKeyingMaterial = kDFHashAlgorithm(0x00000001 || Z || otherInfo) ||
kDFHashAlgorithm(0x00000002 || Z || otherInfo) ||
kDFHashAlgorithm((0x00000000 + N) || Z || otherInfo)

KDF، سمت چپ‌ترین بیت‌های DerivedKeyingMaterial، (keyDataLen) است.

مختصات Q_H، از اجزاء فیلد، به رشته‌های بایتی Q_H' تبدیل می‌شوند

MacData = "KC_1_V" || iccIdentifier || clientIdentifier || NULL || Q_H'
MacTag = sessionMacAlgorithm(SK_{mac}, MacData)

(۴) تایید این که Q_{ICC} برای پارامترهای دامنه D معتبر هستند

Z = keyEstablishmentAlgorithm(dH;QICC)

تبدیل Z به یک رشته بایتی با استفاده از field-element-to-byte-string.

استفاده از KDF تعریف شده در مرحله (۳)

SK_{mac} || SK_{enc} = KDF(Z; KeyDataLen; otherInfo)

مختصات QH، از اجزاء فیلد، به رشته‌های بایتی QH' تبدیل می‌شوند

MacData' = "KC_1_V" || iccIdentifier || clientIdentifier || NULL || Q_H'
MacTag' = sessionMacAlgorithm(SK_{mac}, MadData)
authenticationProtocol ::= empty OCTET STRING (۵)
return_code = API_OK

الف-۱۹-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل احراز هویت، وضعیت احراز هویت مربوط به این هویت متمایزکننده نامگذاری‌شده، درون برنامه کاربردی کارت تغییری نمی‌کند، با این حال، یک جلسه با قابلیت پیام‌رسانی ایمن، آغاز می‌شود.

الف-۱۹-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۹-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۹-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۹-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۹-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۹-۱۲ VerifySignature

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۱۹-۱۳ VerifyCertificate

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

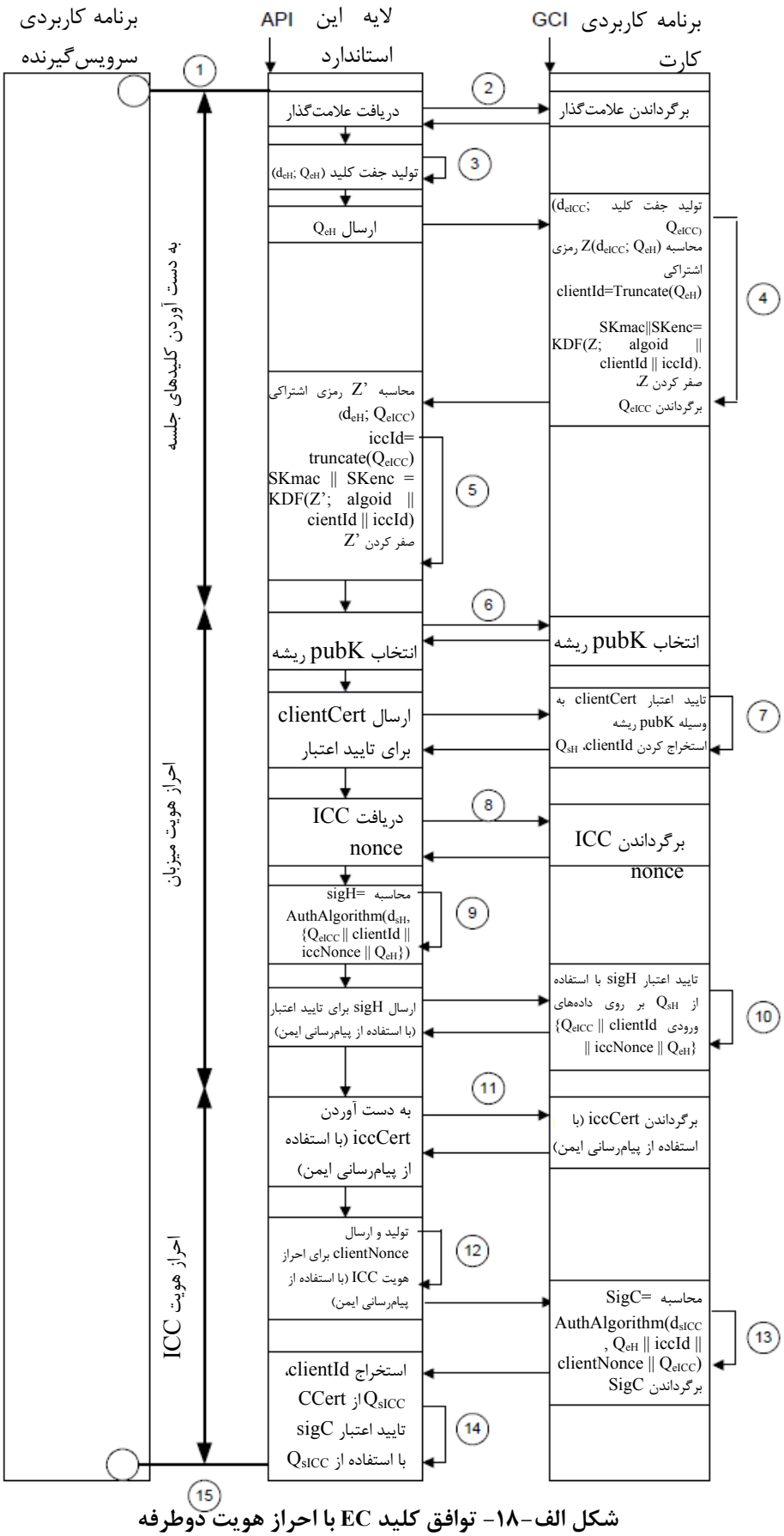
الف-۲۰ توافق کلید EC با احراز هویت دوطرفه

این پروتکل احراز هویت، یک پروتکل احراز هویت دو طرفه با توافق کلید با استفاده از رمزنگاشتی منحنی بیضوی[1] است.

برای سرّی بودن بیشتر، هم سرویس‌گیرنده و هم ICC برای برقرار کردن کلیدهای جلسه، از یک جفت کلید EC موقتی و یک جفت کلید EC ایستا استفاده می‌کنند.

این پروتکل برای برقرار کردن احراز هویت (SKmac) و رمزنگاشتی (SKenc) کلیدهای جلسه برای پیام‌رسانی ایمن‌تر بین لایه این استاندارد و برنامه کاربردی کارت، استفاده می‌شود.

1 - Elliptic Curve



شکل الف-۱۸- توافق کلید EC با احراز هویت دوطرفه

الف-۲۰-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) ec-key-agreement-with-mutual-authentication(20)} شناسایی می‌شود.

الف-۲۰-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP020 ::= SEQUENCE {
DomainParameters OBJECT IDENTIFIER,
KeyEstablishmentAlgorithm OBJECT IDENTIFIER,
KDFHashAlgorithm OBJECT IDENTIFIER,
SessionMacAlgorithm OBJECT IDENTIFIER,
SessionEncAlgorithm OBJECT IDENTIFIER,
AuthAlgorithm OBJECT IDENTIFIER,
nonceSize INTEGER,
CHOICE {
SEQUENCE {
iccPublicKey OCTET STRING,
iccPrivateKey OCTET STRING
},
genKeyPairFlag NULL
},
iccIdentifier OCTET STRING,
iccCert OCTET STRING,
rootIdentifier OCTET STRING,
rootPublicKey OCTET STRING,
iccCertKnown BOOLEAN,
verifyClientCert BOOLEAN,
EnforcePrivacy BOOLEAN
}
```

الف-۲۰-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۲۰-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۲۰-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همانگونه که در الف-۲۰-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۲۰-۵ DIDGet

یک تایید برای عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۲۰-۶ احراز هویت

این پروتکل، به وسیله یک درخواست ساده عمل CardApplicationStartSession() با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

الف-۲۰-۶-۱ رویه

(۱) authenticationProtocolData ::= SEQUENCE {

clientIdentifier OCTET STRING
}

(۲) به دست آوردن پارامترهای دامنه D از الگوی علامت‌گذار. اطلاعات برگردانده شده نباید شامل هیچ‌گونه داده‌های مخصوص آن ICC باشد. تمام کلیدها، گواهی‌ها و شناسانه‌ها باید به NULL تنظیم شوند.

(۳) تولید جفت کلید موقتی (d_{eH}, Q_{eH}) از پارامترهای دامنه D. ارسال Q_{eH} به ICC.

(۴) تایید این که Q_{eH} برای پارامترهای دامنه D، معتبر است. تولید جفت کلید موقتی (d_{eICC}, Q_{eICC}) از پارامترهای دامنه D.

Z = KeyEstablishmentAlgorithm(d_{eICC}, Q_{eH})

تبدیل Z به یک رشته بایتی با استفاده از Field-element-to-Byte-String.

تبدیل مختصات Q_{eICC} و Q_H، از اجزاء فیلد، به رشته‌های بایتی Q_{ICC}' و Q_H'.

clientIdentifier = Truncate(QICC')
iccIdentifier = Truncate(QH')

به‌طوری‌که ()Truncate، ۴ بایت سمت چپ پارامتر خود را برمی‌گرداند.

محاسبه کلیدهای جلسه با استفاده از تابع مشتق‌گیری KDF:

SKmac || SKenc = KDF(Z, keyDataLen, otherInfo)

به طوری که:

otherInfo ::= SEQUENCE {
sessionMacAlgorithm OBJECT IDENTIFIER,
sessionMacAlgorithm OBJECT IDENTIFIER,
clientIdentifier OBJECT IDENTIFIER,
iccIdentifier OBJECT IDENTIFIER
}
keyDataLen = algoKeyLengthInBits(sessionMacAlgorithm)
+ algokeyLengthInBits(sessionEncAlgorithm)
N = keyDataLen / (hashLengthInBits(KDFHashAlgorithm))
DerivedKeyingMaterial = KDFHashAlgorithm(0x00000001 || Z || otherInfo) ||
KDFHashAlgorithm(0x00000002 || Z || otherInfo) ||
…
KDFHashAlgorithm((0x0000000 + N) || Z || otherInfo)

KDF is the keyDataLen left-most bits from DerivedKeyingMaterial.
Zeroize Z and return Q_{eICC}.

(۵) تایید این که Q_{eICC} برای پارامترهای دامنه D، معتبر است.

Z = KeyEstablishmentAlgorithm(D_{eH};Q_{eICC})

تبدیل Z به یک رشته بایتی با استفاده از Field-element-to-Byte-String

تبدیل مختصات Q_{eICC} و Q_H، از اجزاء فیلد، به رشته‌های بایتی Q_{ICC}' و Q_H'.

clientIdentifier = Truncate(Q_{ICC}')
iccIdentifier = Truncate(Q_H')

به‌طوری‌که ()Truncate، ۴ بایت سمت چپ پارامتر خود را برمی‌گرداند.

استفاده از KDF تعریف شده در مرحله (۴)

SKmac || SKenc = KDF(Z; keyDataLen; otherInfo)

صفر کردن Z.

یک کانال برای پیام‌رسانی ایمن (SKmac, SKenc) باز می‌شود.

(۶) انتخاب rootPublicKey برای تایید بیشتر گواهی

(۷) اگر verifyClientCert به TRUE تنظیم شده باشد، تایید clientCert بر اساس rootPublicKey

استخراج clientIdentifier از clientCert و کلید عمومی احراز هویت میزبان Q_{SH}.

تایید این که Q_{SH} برای پارامترهای دامنه D، معتبر است.

اگر verifyClientCert به FALSE تنظیم شده باشد، آنگاه فرض می‌شود که

clientIdentifier = rootIdentifier

و

Q_{SH} = rootPublicKey

(۸) استفاده از پیام‌رسانی ایمن برای انتقال iccNonce.

iccNonce = RNG(nonceSize)

(۹) تبدیل مختصات Q_{eICC} و Q_{eH}، از اجزاء فیلد، به رشته‌های بایتی Q_{ICC}' و Q_H'.

sigH = AuthAlgorithm(d_{sH}, { Q_{ICC}' || clientIdentifier || iccNonce || Q_H'})

AuthAlgorithm باید با پارامترهای دامنه، مطابقت داشته باشد و اجازه احیاء پیام را ندهد.

(۱۰) sigH باید با استفاده از پیام‌رسانی ایمن، منتقل شود.

تایید اعتبار sigH با استفاده از Q_{sH} بر روی داده‌های ورودی {Q_{ICC}' || clientIdentifier || iccNonce || Q_H'}

برنامه کاربردی سرویس‌گیرنده، احراز هویت شده است.

(۱۱) اگر iccCertKnown به TRUE تنظیم شود، آنگاه برگرداندن iccIdentifier به جای iccCert. در صورتی که مقدار enforcePrivacy به TRUE تنظیم شده باشد، رمزنگاشتی iccCert با استفاده از پیام‌رسانی ایمن.

اگر مقدار enforcePrivacy به FALSE تنظیم شده باشد، آنگاه ضروری نیست که گواهی برنامه کاربردی کارت، با پیام‌رسانی ایمن، رمزنگاشتی یا MAC شود.

(۱۲) اگر iccCertKnown به FALSE تنظیم شود، استخراج Q_{sICC} از iccCert. تایید این که Q_{sICC} برای پارامترهای دامنه D، معتبر است.

clientNonce باید با استفاده از پیام‌رسانی ایمن، منتقل شود:

clientNonce = RNG(nonceSize)

(۱۳) تبدیل مختصات Q_{eICC} و Q_{eH}، از اجزاء فیلد، به رشته‌های بایتی Q_{ICC}' و Q_H'

یافتن d_{sICC}، کلید خصوصی احراز هویت متناظر با Q_{sICC}، iccIdentifier، و iccCert.

sigC = AuthAlgorithm(d_{sICC}, { Q_{ICC}' || iccIdentifier || clientNonce || Q_H' })

AuthAlgorithm باید با پارامترهای دامنه، مطابقت داشته باشد و اجازه احیاء پیام را ندهد.

(۱۴) sigC باید با استفاده از پیام‌رسانی ایمن، منتقل شود.

تایید اعتبار sigC با استفاده از Q_{sICC} بر روی داده‌های ورودی {Q_{ICC}' || iccIdentifier || clientNonce || Q_H'}

برنامه کاربردی سرویس‌گیرنده، احراز هویت شده است.

authenticationProtocolData ::= empty OCTET STRING (۱۵)

return_code = API_OK

الف-۲۰-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل احراز هویت، وضعیت احراز هویت مربوط به این هویت متمایز کننده نام‌گذاری شده، باید درون برنامه کاربردی کارت به مقدار نتیجه، تنظیم شود و یک جلسه با قابلیت پیام‌رسانی ایمن، آغاز می‌شود.

الف-۲۰-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۰-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۰-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۰-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۰-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۰-۱۲ VerifySignature

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

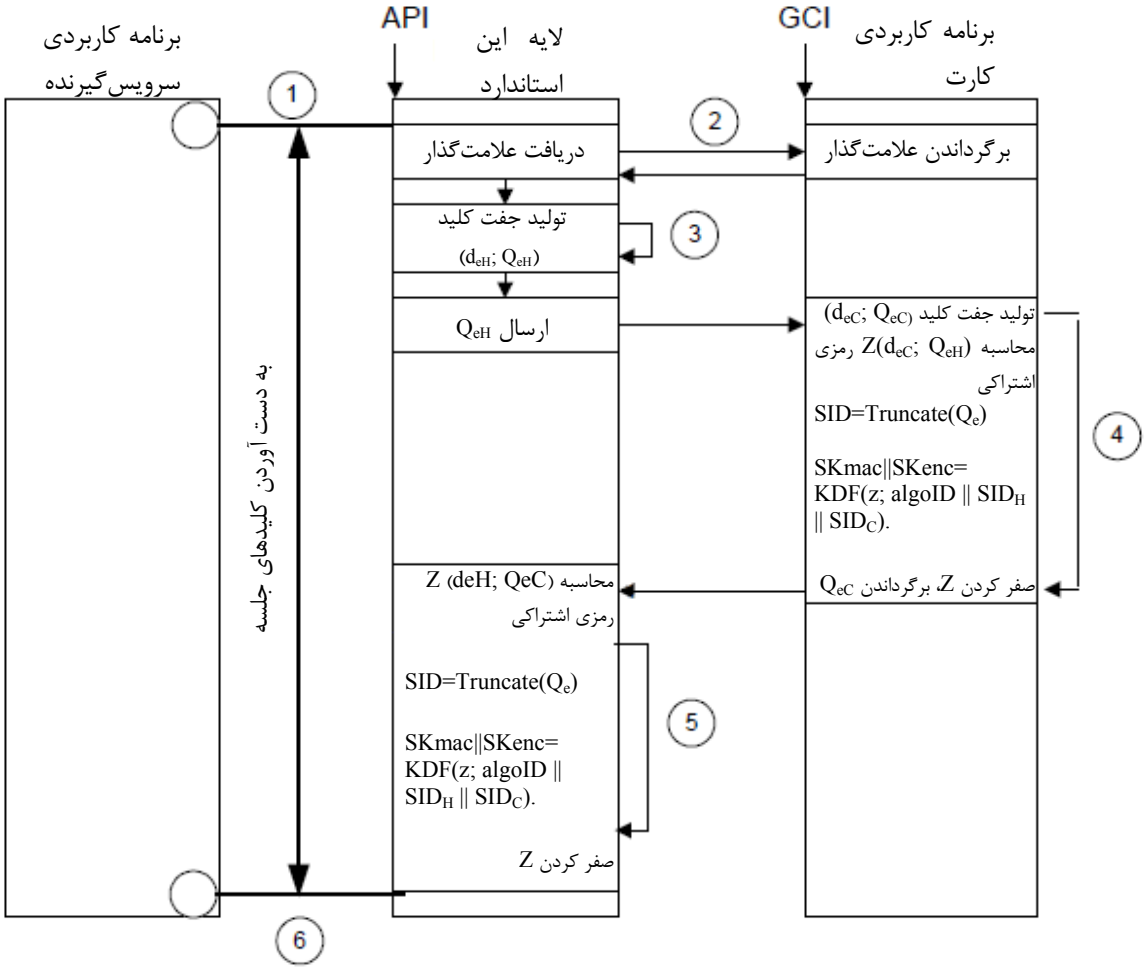
الف-۲۰-۱۳ VerifyCertificate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱ توافق کلید EC-DH ساده

این پروتکل احراز هویت، یک پروتکل ساده با توافق کلید با استفاده از رمزنگاشتی EC است. هم برنامه کاربردی سرویس‌گیرنده و هم برنامه کاربردی کارت، برای توافق کلید، یک جفت کلید موقتی ولی بدون جفت کلید ایستا، تولید می‌کنند و از آن استفاده می‌نمایند.

این پروتکل برای برقرار کردن احراز هویت (SKmac) و رمزنگاشتی (SKenc) کلیدهای جلسه برای پیام‌رسانی ایمن‌تر بین لایه این استاندارد و برنامه کاربردی کارت، استفاده می‌شود.



شکل الف-۱۹- توافق کلید EC-DH ساده

الف-۲۱-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، {INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) simple-ec-dh-key-agreement (21)} شناسایی می‌شود.

الف-۲۱-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار زیر را دارد.

```
MarkerAP021 ::= SEQUENCE {
DomainParameters OBJECT IDENTIFIER,
KeyEstablishmentAlgorithm OBJECT IDENTIFIER,
KDFHashAlgorithm OBJECT IDENTIFIER,
SessionMacAlgorithm OBJECT IDENTIFIER,
SessionEncAlgorithm OBJECT IDENTIFIER
}
```

الف-۲۱-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل، باید شامل یک پارامتر DIDUpdateData باشد در حالی که همانگونه که در الف-۲۱-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۲۱-۴ DIDUpdate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱-۵ DIDGet

یک تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۲۱-۶ احراز هویت

این پروتکل، به‌وسیله یک درخواست ساده عمل CardApplicationStartSession() با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

(۱) authenticationProtocolData ::= empty OCTET STRING

(۲) به دست آوردن پارامترهای دامنه D از الگوی علامت‌گذار. اطلاعات برگردانده شده نباید شامل هیچ‌گونه داده‌های مخصوص آن ICC باشد. تمام کلیدها، گواهی‌ها و شناسانه‌ها باید به NULL تنظیم شوند.

(۳) تولید جفت کلید موقتی $(d_{eH}; Q_{eH})$ از پارامترهای دامنه D. ارسال Q_{eH} به ICC.

(۴) تایید این که Q_{eH} برای پارامترهای دامنه D، معتبر است.

تولید جفت کلید موقتی $(d_{eICC}; Q_{eICC})$ از پارامترهای دامنه D.

$Z = KeyEstablishmentAlgorithm(d_{eICC};Q_{eH})$

تبدیل Z به یک رشته بایتی با استفاده از field-element-to-byte-string

تبدیل مختصات Q_{eICC} و Q_H، از اجزاء فیلد، به رشته‌های بایتی Q_{ICC}' و Q_H'

$clientIdentifier = Truncate(Q_{ICC}')$
$iccIdentifier = Truncate(Q_H')$

به طوری که Truncate()، ۴ بایت سمت چپ پارامتر خود را برمی‌گرداند.

محاسبه کلیدهای جلسه با استفاده از تابع مشتق‌گیری KDF:

```
SKmac || SKenc = KDF( Z, keyDataLen, otherInfo)
otherInfo ::= SEQUENCE {
SessionMacAlgorithm OBJECT IDENTIFIER,
SessionEncAlgorithm OBJECT IDENTIFIER,
clientIdentifier OCTET STRING,
iccIdentifier OCTET STRING,
iccNonce OCTET STRING
}
keyDataLen = algoKeyLengthInBits(SessionMacAlgorithm) +
AlgoKeyLengthInBits(SessionEncAlgorithm)
N = keyDataLen / (hashLengthInBits (KDFHashAlgorithm))
DerivedKeyingMaterial = KDFHashAlgorithm(0x00000001 || Z || otherInfo) ||
```

KDFHashAlgorithm(0x00000002 || Z || otherInfo) ||
…
KDFHashAlgorithm((0x00000000 + N) || Z || otherInfo)

KDF، به تعداد keyDataLen بیت، سمت چپ‌ترین بیت‌های DerivedKeyingMaterial است.

صفر کردن Z و برگرداندن Q_{eICC}.

(۵) تایید این که QeICC برای پارامترهای دامنه D، معتبر است.

$Z = KeyEstablishmentAlgorithm(D_{eH};Q_{eICC})$

تبدیل Z به یک رشته بایتی با استفاده از field-element-to-byte-string

تبدیل مختصات QeICC و QeH، از اجزاء فیلد، به رشته‌های بایتی QICC' و QH'

clientIdentifier = Truncate(QICC')
iccIdentifier = Truncate(QH')

به طوری که ()Truncate، ۴ بایت سمت چپ پارامتر خود را برمی‌گرداند.

استفاده از KDF تعریف شده در مرحله (۴)

SKmac || SKenc = KDF(Z; keyDataLen; otherInfo)

صفر کردن Z

باز شدن یک کانال برای پیام‌رسانی ایمن (SKmac, SKenc)

(۶) authenticationProtocolData ::= empty OCTET STRING

return_code = API_OK

الف-۲۱-۶-۱ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل احراز هویت، وضعیت احراز هویت مربوط به این هویت متمایزکننده نام‌گذاری شده، درون برنامه کاربردی کارت، تغییری نمی‌کند؛ با این حال، یک جلسه با قابلیت پیام‌رسانی ایمن، آغاز می‌شود.

الف-۲۱-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱-۱۱ Sign

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱-۱۲ VerifySignature

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۱-۱۳ VerifyCertificate

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک عمل درخواست باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

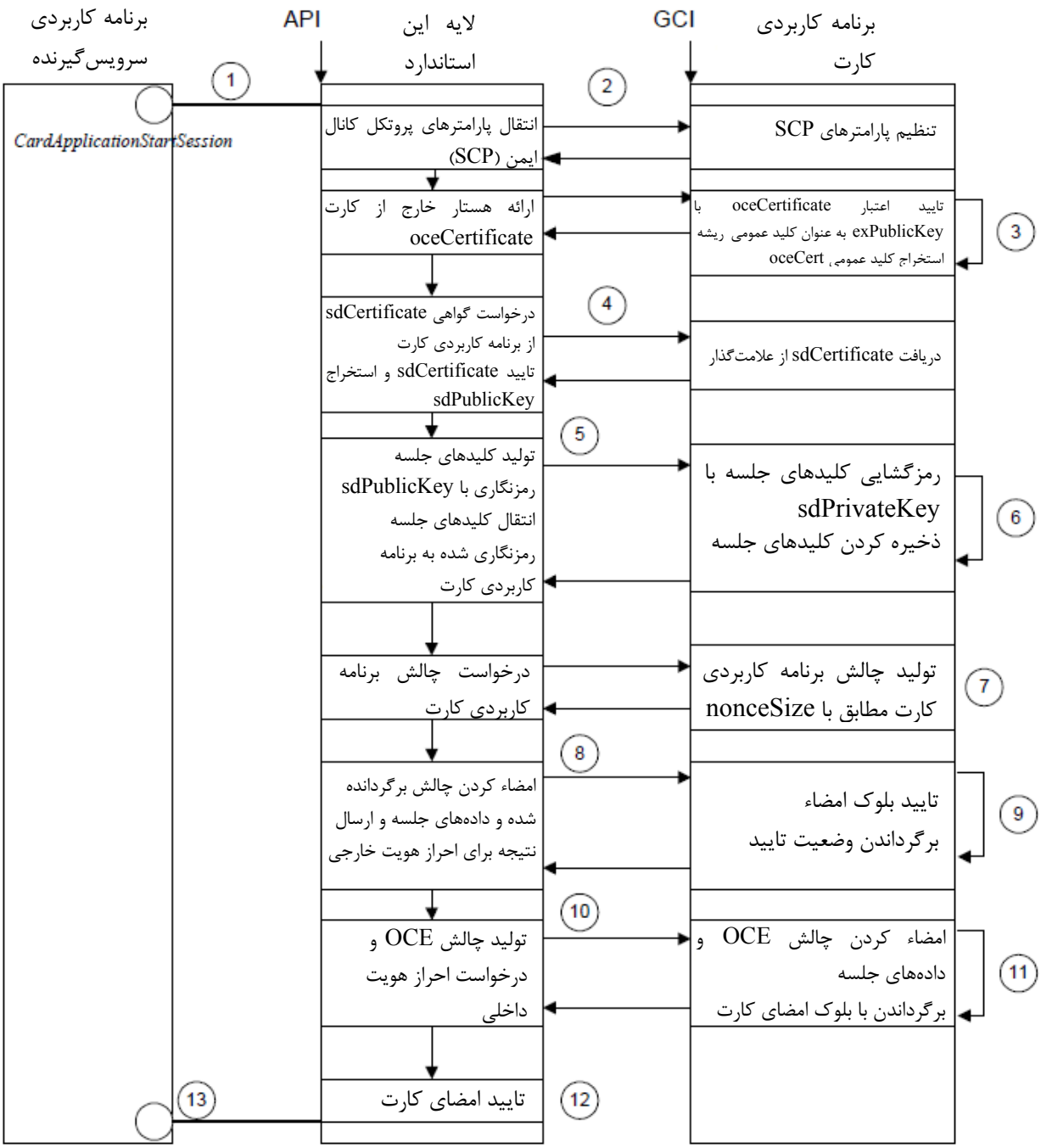
الف-۲۲ احراز هویت نامتقارن GP

این پروتکل احراز هویت، یک پروتکل چالش-پاسخ با استفاده از رمزنگاشتی کلید عمومی است[۱].

این پروتکل هم برای برقرار کردن وضعیت احراز هویت مربوط به یک هویت متمایزکننده در یک برنامه کاربردی کارت و هم برای انتقال دو کلید جلسه برای احراز هویت پیام (SKmac) و برای رمزنگاشتی (SKenc) برای پیام‌رسانی ایمن‌تر بین لایه این استاندارد و برنامه کاربردی کارت، استفاده می‌شود[۲].

۱ـ در GlobalPlatform 2.2 Card Specification، به صورت "SCP10" با گزینه انتقال کلید (i=0x02) تعریف شده است.

۲ ـ با این پروتکل احراز هویت، کلید خصوصی، درون SAL، میزبانی می‌شود. این اشاره می‌کند که احراز هویت سرویس‌گیرنده و SAL باید در محیط مورد اعتماد یکسانی اجرا شوند. تنظیمات مناسب برای این طرح (به استاندارد ملی ایران شماره ۴ ـ ۱۶۳۸۶ مراجعه کنید)، Loyal-stack، Opaque-ICC-stack، و Remote-ICC-Stack هستند.



شکل الف-۲۰- احراز هویت نامتقارن GP

الف-۲۲-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه‌های شیء زیر شناسایی می‌شود:

{ INSO(1) INSO 16386 (16386) part3(3) annex-a(0) gp-asymmetric-authentication (22) }

اگر فقط احراز هویت خارجی مورد نیاز باشد،

{ INSO(1) INSO 16386 (16386) part3(3) annex-a(0) gp-asymmetric-authentication (22) mutual-auth(0) }

اگر احراز هویت داخلی (مرحله‌های (۱۰) تا (۱۲)) مورد نیاز باشد.

الف-۲۲-۲ علامت‌گذار

هویت متمایز کننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار تعریف شده با الگوی زیر را دارد:

```
MarkerAP022 ::= SEQUENCE {
sdPublicKey OCTET STRING,
sdPrivateKey OCTET STRING,
sdCertificate OCTET STRING,
exPublicKey OCTET STRING, // PK.TP_EX.AUT as defined in GP 2.2
(F.1.2.1 Overview)
ocePubliKey OCTET STRING, // PK.OCE.AUT as defined in GP 2.2
(F.1.2.1 Overview)
kaExPublicKey OCTET STRING, // PK.KA_EX.AUT as defined in GP 2.2
(F.1.2.1)
kaInCertificate OCTET STRING, // CERT.KA_IN.AUT as defined in GP
2.2 (F.1.2.1 Overview)
sessionMacAlgo OBJECT IDENTIFIER,
sessionEncAlgo OBJECT IDENTIFIER,
keyEncryptionAlgo OBJECT IDENTIFIER,
hashAlgorithm OBJECT IDENTIFIER,
nonceSize INTEGER
}
```

الف-۲۲-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۲۲-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۲۲-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۲۲-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۲۲-۵ DIDGet

یک تایید برای عمل DIDGet مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۲۲-۶ احراز هویت

این پروتکل، به‌وسیله یک درخواست ساده عمل CardApplicationStartSession با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل DIDAuthentication باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۶-۱ رویه

```
authenticationProtocolData ::= SEQUENCE { (۱)
oceCertificate OCTET STRING,
securityLevel OCTET STRING, // As specified in GlobalPlatform
Specifications 2.2 Table 10-1
```

SCPParametersP1P2 OCTET STRING, // As specified in GlobalPlatform Specifications 2.2 F4.5.2 and F4.5.4
}

(۲) پارامترهای SCP با برنامه کاربردی کارت، ارتباط برقرار می‌کنند

(۳) دامنه امنیتی (SD) برنامه کاربردی کارت، oceCertificate را تایید می‌کند و اعتبار ocePublicKey، کلید عمومی متناظر، را با استفاده از exPublicKey به عنوان کلید عمومی دارای اختیار ذخیره شده در علامت‌گذار، برقرار می‌نماید.

(۴) لایه این استاندارد، sdCertificate را تایید می‌کند و اعتبار sdPublicKey را بر اساس گواهی دارای اختیار و سیاست‌های صادر کننده sdCertificate برقرار می‌نماید.

(۵) لایه این استاندارد، کلیدهای جلسه را برای MAC و ENC تولید می‌کند، آن‌ها را با sdPublicKey رمزنگاشتی می‌نماید و بلوک داده‌های رمزنگاشتی شده encData را به برنامه کاربردی کارت، منتقل می‌کند.

skMac = sessionMacAlgo()
skEnc = sessionEncAlgo()
crtMac ::= SEQUENCE {
crtTag OCTET STRING, // 'B4' (CCT) or 'B8' (CT)
crtLength OCTET STRING,
usageQualifier OCTET STRING, // '95.01.XX'
skMac OCTET STRING, // 'D1.18.XX…XX'
cryptoAlgo OCTET STRING, // GP 2.2 Table F-11 and 11.1.8, with a value corresponding to sessionMacAlgo OID
sequenceCounter OCTET STRING // '91.08.XX…XX'
}
crtEnc ::= SEQUENCE {
crtTag OCTET STRING, // 'B4' (CCT) or 'B8' (CT)
crtLength OCTET STRING,
usageQualifier OCTET STRING, // '95.01.XX'
skEnc OCTET STRING, // 'D1.18.XX…XX'
cryptoAlgo OCTET STRING, // GP 2.2 Table F-11 and 11.1.8, with a value corresponding to sessionMacAlgo OID
sequenceCounter OCTET STRING // '91.08.XX…XX'
}
keyData ::= SEQUENCE {
paddingPattern OCTET STRING, // '00.02.FF…FF.00'
securityLevelTag OCTET STRING, // 'D3'
securityLevelLen OCTET STRING, // 'D1'
securityLevel OCTET STRING,
crtMac OCTET STRING,
crtEnc OCTET STRING
}
encData = keyEncryptionAlgo[sdPublicKey](keyData)

(۶) برنامه کاربردی کارت، بلوک داده‌های دریافت شده را با sdPrivateKey رمزگشایی می‌کند تا کلیدهای جلسه MAC و ENC را استخراج نماید، و آن‌ها را برای استفاده بعدی پیام ایمن، ذخیره می‌کند.

keyData = keyEncryptionAlgo-1[sdPrivateKey](encData)

(۷) برنامه کاربردی کارت، یک چالش برای احراز هویت خارجی، تولید می‌کند

sdNonce = RNG(nonceSize)

(۸) لایه این استاندارد، امضای خود را برای احراز هویت خارجی، به برنامه کاربردی کارت می‌فرستد. امضای لایه این استاندارد، حاصل ایجاد یک تغییر (درهم‌سازی) روی یک مجموعه داده‌ها، ایجاد یک بلوک امضاء، و امضاء کردن بلوک امضاء با ocePrivateKey، کلید خصوصی متناظر با oceCertificate، است.

```
dataToHash ::= SEQUENCE {
securityLevelTag OCTET STRING, // 'D3'
securityLevelLen OCTET STRING, // '01'
securityLevel OCTET STRING,
crtMac OCTET STRING,
crtEnc OCTET STRING,
sdNonce OCTET STRING
}
hashData = hashAlgorithm(dataToHash)
sigBlock ::= SEQUENCE {
paddingPattern OCTET STRING, // '00.01.FF…FF.00'
derHashAlgorithm OCTET STRING, // DER encoded OID
(of GP's hash algo)
derHashData OCTET STRING // DER encoded hashData
}
oceSignature = authAltorithm[ocePrivateKey](sigBlock)
```

(۹) برنامه کاربردی کارت، oceSignature را با استفاده از ocePublicKey استخراج شده در مرحله (۲) تایید می‌کند. اکنون، اگر تایید، موفقیت‌آمیز باشد کانال ایمن، در دسترس است. اگر احراز هویت داخلی، مورد نیاز نباشد (اولین شناسانه شیء پشتیبانی شده به وسیله این پروتکل احراز هویت)، از مرحله (۱۳) ادامه می‌یابد.

(۱۰) اگر احراز هویت داخلی، مورد نیاز باشد (برای دومین شناسانه شیء پشتیبانی شده به وسیله این پروتکل احراز هویت)، آنگاه مرحله‌های (۱۰)، (۱۱) و (۱۲) اجرا می‌شوند.

OCE برای چالش برنامه کاربردی کارت، یک nonce تولید می‌کند:

oceNonce = RNG(nonceSize)

(۱۱) برنامه کاربردی کارت، امضای خود را برای احراز هویت داخلی به OCE برمی‌گرداند. امضای برنامه کاربردی کارت، حاصل ایجاد یک تغییر (درهم‌سازی) روی یک مجموعه داده‌ها، ایجاد یک بلوک امضاء، و امضاء کردن آن بلوک با sdPrivateKey است:

```
sdSigBlock ::= SEQUENCE {
paddingPattern OCTET STRING, // '00.01.FF…FF.00'
derHashAlgorithm OCTET STRING, // DER encoded OID
derHashData OCTET STRING // DER encoded hashData
hashData = hashAlgorithm(dataToHash)
sdDataToHash ::= SEQUENCE {
crtMac OCTET STRING,
crtEnc OCTET STRING,
oceNonce OCTET STRING
}
sdSignature = authAlgorithm[sdPrivateKey](sdSigBlock)
```

(۱۲) لایه این استاندارد، sdSignature را با استفاده از sdPublicKey استخراج شده در مرحله (۳) تایید می‌کند. این تایید می‌نماید که برنامه کاربردی کارت، کلیدهای جلسه ارسال شده به وسیله لایه این استاندارد را به طور صحیح، استخراج کرده است.

(۱۳) authenticationProtocolData ::= empty OCTET STRING

اگر احراز هویت، با موفقیت به پایان برسد return_code = API_OK.

الف-۲۲-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل احراز هویت، وضعیت احراز هویت مربوط به این هویت متمایز کننده نامگذاری شده، باید درون برنامه کاربردی کارت، به TRUE تنظیم شود، و یک جلسه با قابلیت پیام‌رسانی ایمن، آغاز می‌شود.

الف-۲۲-۷ Encipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست این عمل باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۸ Decipher

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست این عمل باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۹ GetRandom

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست این عمل باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۱۰ Hash

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست این عمل باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۱۱ Sign

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست این عمل باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۱۲ VerifySignature

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست این عمل باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۱۳ VerifyCertificate

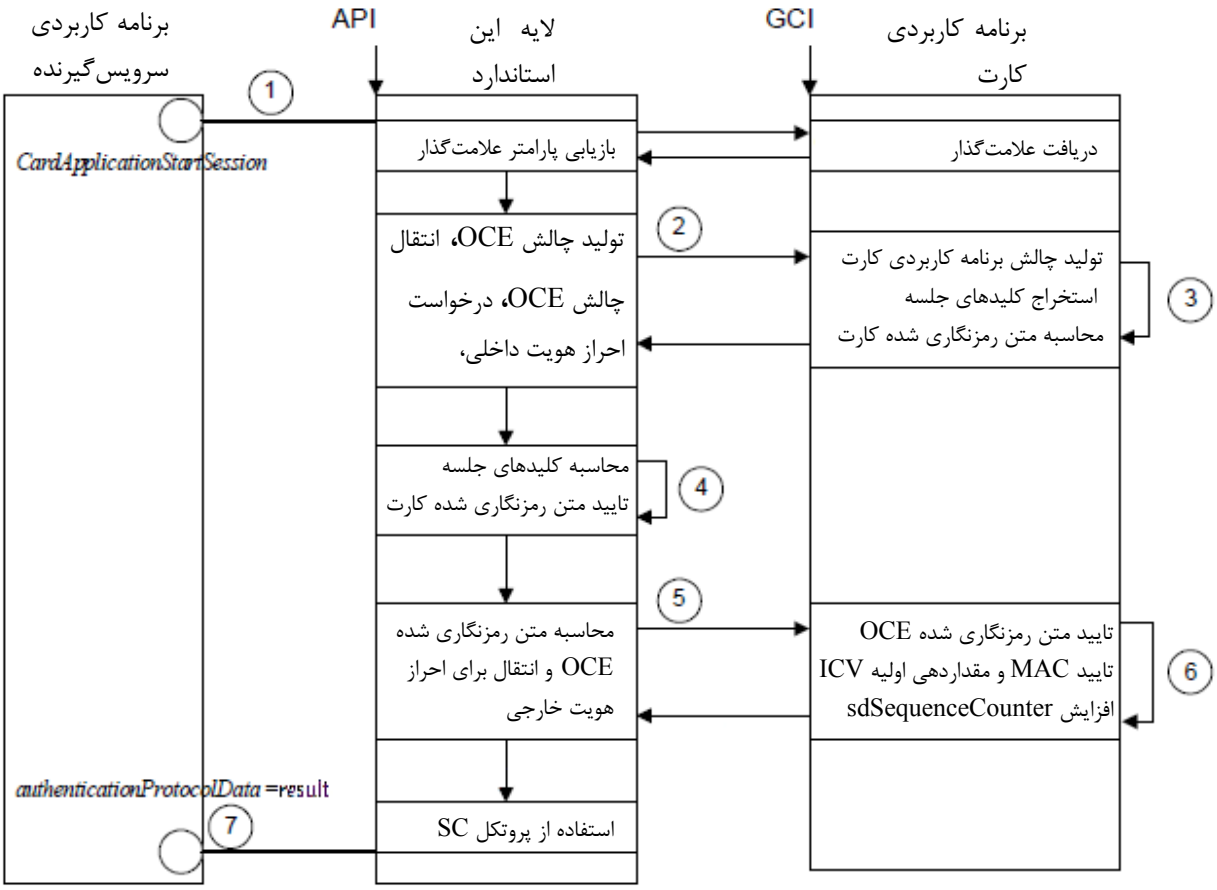
در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست این عمل باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۲-۱۴ CardApplicationEndSession

یک درخواست این عمل، پس از این که هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، احراز هویت شد، باید وضعیت احراز هویت مربوط به این هویت متمایزکننده را به FALSE تنظیم کند، و کد بازگشتی API_OK را برگرداند.

الف-۲۳ احراز هویت متقارن GP (حالت صریح)

این پروتکل احراز هویت، یک پروتکل چالش-پاسخ با استفاده از رمزنگاشتی کلید متقارن است[۱]. این پروتکل هم برای برقرار کردن وضعیت احراز هویت مربوط به یک هویت متمایز کننده در یک برنامه کاربردی کارت و هم برای توافق بر سر دو کلید جلسه برای احراز هویت پیام (skMac) و برای رمزنگاشتی (skEnc) برای پیام‌رسانی ایمن‌تر بین لایه این استاندارد و برنامه کاربردی کارت، استفاده می‌شود.



شکل الف-۲۱- احراز هویت متقارن GP (حالت صریح)

الف-۲۳-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء، { INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) gp-symmetric-authentication-explicit-mode (23) } شناسایی می‌شود.

الف-۲۳-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، علامت‌گذار تعریف شده به‌صورت زیر را دارد:

---

۱ ـ در GlobalPlatform 2.2 Card Specification، به صورت SCP02 در حالت صریح (i=0x25) تعریف شده است.

```
MarkerAP023 ::= SEQUENCE {
sdStaticEncKey OCTET STRING,
sdStaticMacKey OCTET STRING,
sdStaticDekKey OCTET STRING,
sdSequenceCounter OCTET STRING,
sessionMacAlgo OBJECT IDENTIFIER,
sessionEncAlgo OBJECT IDENTIFIER,
keyDerivationAlgo OBJECT IDENTIFIER,
nonceSize INTEGER
}
```

الف-۲۳-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل احراز هویت، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۲۳-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۲۳-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایز کننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک پارامتر DIDUpdateData باشد در حالی‌که همان‌گونه که در الف-۲۳-۲ تعریف شده، marker، علامت‌گذار خاص این پروتکل است.

الف-۲۳-۵ DIDGet

یک تایید یک درخواست عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک پارامتر DIDStructure باشد.

الف-۲۳-۶ احراز هویت

این پروتکل، به وسیله یک درخواست ساده عمل CardApplicationStartSession با یک پارامتر authenticationProtocolData تعریف شده در زیر، اجرا می‌شود.

در هویت متمایز کننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل DIDAuthentication باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۳-۶-۱ رویه

(۱) authenticationProtocolData ::=

```
securityLevel OCTET STRING // As specified in GlobalPlatform
Card Specification 2.2 Table E-11
```

(۲) لایه این استاندارد، یک چالش برای برنامه کاربردی کارت برای احراز هویت داخلی، ایجاد می‌کند:

```
oceNonce = RNG(nonceSize)
```

(۳) برنامه کاربردی کارت نیز یک nonce تولید می‌کند، سپس کلیدهای جلسه را استخراج می‌نماید و متن رمزنگاشتی شده برنامه کاربردی کارت را محاسبه می‌کند:

```
sdNonce = RNG(nonceSize)
skMac = keyDerivationAlgo[sdStaticEncKey](sdSequenceCounter)
skEnc = keyDerivationAlgo[sdStaticMacKey](sdSequenceCounter)
sdInput ::= SEQUENCE {
```

```
oceNonce OCTET STRING,
sdSequenceCounter OCTET STRING,
sdNonce OCTET STRING,
padding OCTET STRING
}
sdCryptogram = authAlgorithm[skEnc](sdInput)
```

(۴) همچنین، لایه این استاندارد، به صورت مرحله (۳) برنامه کاربردی کارت را محاسبه می‌کند، و یک مقایسه، انجام می‌دهد. اگر مقایسه، موفقیت‌آمیز باشد، برنامه کاربردی کارت را احراز هویت می‌کند.

(۵) لایه این استاندارد، متن رمزنگاشتی‌شده[1] هستار خارج از کارت را محاسبه می‌کند و آن را به همراه یک الگوی CMAC برای احراز هویت خارجی، به برنامه کاربردی کارت، می‌فرستد.

```
oceInput ::= SEQUENCE {
sdSequenceCounter OCTET STRING,
sdNonce OCTET STRING,
oceNonce OCTET STRING,
padding OCTET STRING
}
oceCryptogram = authAlgorithm[skEnc](oceInput)
macInput ::= SEQUENCE {
apduHeader OCTET STRING,
oceCryptogram OCTET STRING
}
CMAC = sessionMacAlgo[skMac](macInput)
```

(۶) برنامه کاربردی کارت، همان متن رمزنگاشتی شده و CMAC را محاسبه می‌کند و با داده‌های دریافت شده در مرحله (۵) یک مقایسه، انجام می‌دهد. اگر مقایسه، موفقیت‌آمیز باشد، برنامه کاربردی کارت، لایه این استاندارد را احراز هویت می‌کند و sdSequenceCounter را افزایش می‌دهد و مقدار اولیه ICV را به مقدار CMAC تنظیم می‌نماید.

(۷) authenticationProtocolData ::= empty OCTET STRING

اگر احراز هویت، با موفقیت به پایان برسد return_code = API_OK.

الف-۲۳-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل احراز هویت، وضعیت احراز هویت مربوط به این هویت متمایزکننده نامگذاری شده، باید درون برنامه کاربردی کارت، به مقدار 'result' تنظیم شود، و یک جلسه با قابلیت پیام‌رسانی ایمن، آغاز می‌شود.

الف-۲۳-۷ Encipher

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

1 - Cryptogram

الف-۲۳-۸ Decipher

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۳-۹ GetRandom

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۳-۱۰ Hash

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۳-۱۱ Sign

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۳-۱۲ VerifySignature

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۳-۱۳ VerifyCertificate

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۳-۱۴ CardApplicationEndSession

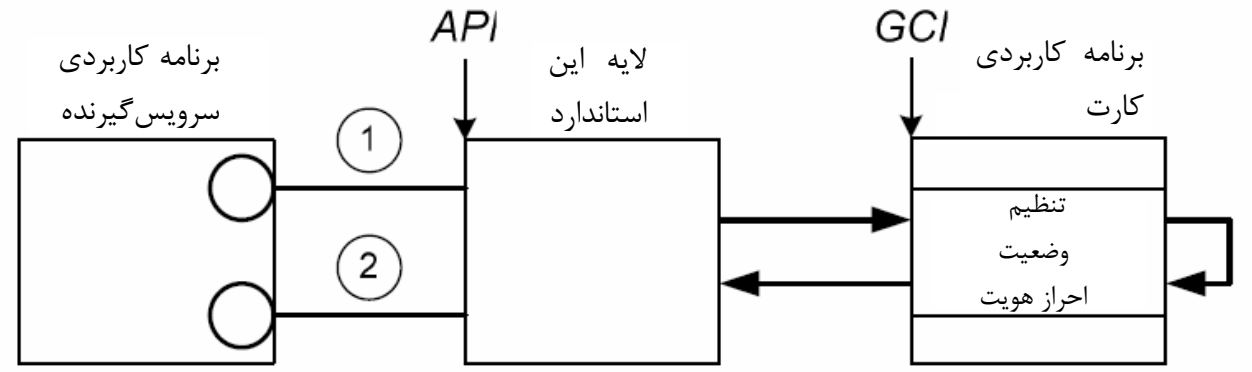
یک درخواست این عمل، پس از احراز هویت موفقیت‌آمیز یک هویت متمایزکننده، باید وضعیت احراز هویت مربوط به این هویت متمایز کننده را به FALSE تنظیم کند، و کد بازگشتی API_OK را برگرداند.

الف-۲۴ احراز هویت متقارن GP (حالت ضمنی)

این پروتکل احراز هویت، یک اعلان ساده است که نشان می‌دهد ارتباط متعاقب با برنامه کاربردی کارت به صورت حالت ضمنی پیام‌رسانی ایمن، خواهد بود[1]. این پروتکل هم برای برقرار کردن وضعیت احراز هویت مربوط به یک هویت متمایز کننده در یک برنامه کاربردی کارت و هم برای توافق بر سر دو کلید جلسه برای احراز هویت پیام (skMac) و برای رمزنگاشتی (skEnc) برای پیام‌رسانی ایمن‌تر بین لایه این استاندارد و برنامه کاربردی کارت، استفاده می‌شود.

---

۱ ـ در GlobalPlatform Card Specification 2.2، به صورت SCP02 در حالت ضمنی (i=0x29) تعریف شده است.



شکل الف-۲۲- احراز هویت متقارن GP (حالت ضمنی)

الف-۲۴-۱ شناسانه شیء پروتکل

این پروتکل به وسیله شناسانه شیء زیر شناسایی می‌شود: { INSO(1) standard(0) INSO 16386 (16386) part3(3) annex-a(0) gp-asymmetric-authentication-implicit-mode (24) }

الف-۲۴-۲ علامت‌گذار

هویت متمایزکننده‌ای که از این پروتکل، استفاده می‌کند، باید یک علامت‌گذار، به صورت تعریف شده در زیر داشته باشد:

```
SpecificMarkerStruct ::= SEQUENCE {
sdStaticBaseKey OCTET STRING,
sdSequenceCounter OCTET STRING,
sessionMacAlgo OBJECT IDENTIFIER,
sessionEncAlgo OBJECT IDENTIFIER,
keyDerivationAlgo OBJECT IDENTIFIER
}
```

الف-۲۴-۳ DIDCreate

یک درخواست عمل DIDCreate در نظر گرفته‌شده برای ایجاد یک هویت متمایزکننده برای استفاده با این پروتکل، باید شامل یک پارامتر didStructure به‌صورت تعریف شده در زیر، باشد:

```
didStructure ::= SEQUENCE {
authProtocol OBJECT IDENTIFIER,
marker OCTET STRING,
scope BOOLEAN OPTIONAL,
qualifier DIDQualifier OPTIONAL
}
```

الف-۲۴-۴ DIDUpdate

یک درخواست عمل DIDUpdate مرتبط با هویت متمایزکننده‌ای که از این پروتکل احراز هویت استفاده می‌کند، باید شامل یک SpecificMarkerStruct به صورت تعریف شده در الف-۲۴-۲ باشد.

الف-۲۴-۵ DIDGet

یک تایید درخواست عمل DIDGet مرتبط با هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، باید شامل یک SpecificMarkerStruct تعریف شده در الف-۲۴-۲ باشد.

الف-۲۴-۶ احراز هویت

این پروتکل، به‌وسیله یک درخواست ساده عمل CardApplicationStartSession با یک پارامتر NULL authenticationProtocolData، به همان صورتی‌که به طور ضمنی به وسیله طرفین، شناخته می‌شود، اجرا می‌گردد.

در هویت متمایزکننده‌ای که از این پروتکل استفاده می‌کند، یک درخواست عمل DIDAuthenticate باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۶-۱ رویه

(۱) authenticationProtocolData ::= empty OCTET STRING

(۲) authenticationProtocolData ::= empty OCTET STRING

اگر احراز هویت، با موفقیت به پایان برسد return_code = API_OK.

الف-۲۴-۶-۲ تاثیر روی وضعیت جاری

پس از اتمام موفقیت‌آمیز این پروتکل احراز هویت، وضعیت احراز هویت مربوط به این هویت متمایز کننده نامگذاری شده، باید درون برنامه کاربردی کارت، به TRUE تنظیم شود.

پس از دریافت متعاقب اولین APDU در پیام‌رسانی ایمن، کلیدهای جلسه به صورت زیر، محاسبه می‌شوند:

skMac = keyDerivationAlgo[sdStaticBaseKey](sdSequenceCounter)
skEnc = keyDerivationAlgo[sdStaticBaseKey](sdSequenceCounter)

برای محاسبه اولین CMAC متصل به اولین APDU محافظت‌شده با پیام‌رسانی ایمن، همان‌گونه که در GlobalPlatform card Specification 2.2 Table E-1 مشخص شده است، ICV بر روی AID به MAC تنظیم می‌شود. پس از تایید موفقیت‌آمیز مقدار CMAC، برنامه کاربردی کارت، sdSequenceCounter را افزایش می‌دهد.

الف-۲۴-۷ Encipher

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۸ Decipher

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۹ GetRandom

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۱۰ Hash

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۱۱ Sign

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۱۲ VerifySignature

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۱۳ VerifyCertificate

یک درخواست این عمل که دارای ارجاع به هویت متمایزکننده‌ای است که از این پروتکل استفاده می‌کند، باید کد بازگشتی API_INAPPROPRIATE_PROTOCOL_FOR_ACTION را برگرداند.

الف-۲۴-۱۴ CardApplicationEndSession

یک درخواست این عمل، پس از احراز هویت موفقیت‌آمیز یک هویت متمایز کننده، باید وضعیت احراز هویت مربوط به این هویت متمایز کننده را به FALSE تنظیم کند، و کد بازگشتی API_OK را برگرداند.

پیوست ب

(الزامی)

الگوریتم‌های رمزنگاشتی

ب-۱ الزامات تعامل‌پذیری

این پیوست، منبع تعریف کننده الگوریتم‌های رمزنگاشتی استفاده شده در پروتکل‌های احراز هویت مشخص شده در پیوست الف را ارائه می‌نماید.

یک ماژول ASN.1، «ALGO- استاندارد ملی ایران ۳ ـ ۱۶۳۸۶» به‌صورت زیر تعریف شده است:

```
ISO24727-3-ALGO {iso(1) standard(0) iso24727(24727) part3(3) annexB (13) }
-- Version 1.3, 03-Mar-2010
--
-- IF-PROFILE value '01'
--
-- *According to ISO/IEC 24727-2, the optional IF-PROFILE field in the CCD is
-- used to indicate that a card provides an implementation of ISO/IEC 24727-3.
-- © ISO/IEC 2008-2010
-- All rights reserved. Unless otherwise specified, no part of this publication
-- may be reproduced or utilized in any form or by any means, electronic or
-- mechanical, including photocopying and microfilm, without permission in
-- writing from either ISO at the address below or ISO's member body in the
-- country of the requester.
--
-- ISO copyright office
-- Case postale 56 • CH-1211 Geneva 20
-- Tel. + 41 22 749 01 11
-- Fax + 41 22 749 09 47
-- E-mail copyright@iso.org
-- Web www.iso.org
DEFINITIONS AUTOMATIC TAGS EXTENSIBILITY IMPLIED ::=
BEGIN
-- EXPORTS; Exports all
IMPORTS;
-- Major and Minor Revision values for this ASN.1 Module
revMajISO24727-3-ALGO INTEGER ::= 1
revMinISO24727-3-ALGO INTEGER ::= 3
AlgorithmIDParameters ::= SEQUENCE {
algorithm
ALGORITHMIDENTIFIERPARAMETERS.&id({SupportedAlgorithms}),
parameters
ALGORITHMIDENTIFIERPARAMETERS.&Type
({SupportedAlgorithms}{@algorithm}) OPTIONAL}
ALGORITHMIDENTIFIERPARAMETERS ::= CLASS {&id OBJECT IDENTIFIER,
&Type OPTIONAL }
WITH SYNTAX {ID &id
[TYPE &Type]}
-- Algorithm OIDs
-- Add an Unknown
id-unknownAlgorithmIdentifier OBJECT IDENTIFIER ::=
{iso(1) standard(0) iso24727(24727) part3(3) annexB(13)
algorithmIdentifiers(1) unknown(0)}
```

unknownAlgorithmIdentifier ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-unknownAlgorithmIdentifier}

ب-۲ الگوریتم‌های متقارن

ب-۲-۱ مراجع

18033-2 refers to hash functions in 10118-2 and 10118-3.
18033-2 describes key derivation functions KDF1 and KDF2 with dependences on hash functions.
18033-2 refers to MAC algorithms in 9797-1 and 9797-2.
18033-2 refers to block ciphers in 18033-3.
18033-2 describes symmetric ciphers SC1 and SC2 with dependences on block ciphers and key derivation functions.
18033-2 describes key encapsulation mechanisms (KEM).
18033-2 describes data encapsulation mechanisms (DEM).
18033-2 describes hybrid ciphers.
18033-2 describes ElGamal-based KEM.
18033-2 describes RSA-based asymmetric cipher and KEM.
18033-3 defines TDEA (which is 3DES) in terms of DES.
18033-3 defines DES (in Annex A).
18033-3 defines MISTY1.
18033-3 defines CAST-128.
18033-3 defines AES.
18033-3 defines Camellia.
18033-3 defines SEED.
18033-4 describes the construction of stream ciphers from block ciphers.
18033-4 defines MUGI keystream generator.
18033-4 defines SNOW 2.0 keystream generator.

ب-۲-۲ الگوریتم‌های متقارن از استاندارد ملی ایران شماره ۳ـ۱۰۸۲۴

id-is18033-3 OBJECT IDENTIFIER ::= { 1 0 18033 3 }
id-bc64 OBJECT IDENTIFIER ::= { 1 0 18033 3 1 }
id-bc128 OBJECT IDENTIFIER ::= { 1 0 18033 3 2 }
id-bc64-tdea OBJECT IDENTIFIER ::= { 1 0 18033 3 1 1 }
id-bc64-misty1 OBJECT IDENTIFIER ::= { 1 0 18033 3 1 2 }
id-bc64-cast128 OBJECT IDENTIFIER ::= { 1 0 18033 3 1 3 }
id-bc128-aes OBJECT IDENTIFIER ::= { 1 0 18033 3 2 1 }
id-bc128-camellia OBJECT IDENTIFIER ::= { 1 0 18033 3 2 2 }
id-bc128-seed OBJECT IDENTIFIER ::= { 1 0 18033 3 2 3 }
is18033-3 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-is18033-3 }
bc64 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc64 }
bc128 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc128 }
bc64-tdea ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc64-tdea }
bc64-misty1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc64-misty1 }
bc64-cast128 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc64-cast128 }
bc128-aes ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc128-aes }
bc128-camellia ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc128-camellia }
bc128-seed ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-bc128-seed }

ب-۲-۳ AES (کلید ۱۲۸ بیتی)

```
id-aes128-ECB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 1 }
id-aes128-CBC OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 2 }
id-aes128-OFB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 3 }
id-aes128-CFB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 4 }
aes128-ECB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes128-ECB }
aes128-CBC ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes128-CBC }
aes128-OFB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes128-OFB }
aes128-CFB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes128-CFB }
```

ب-۲-۴ AES (کلید ۱۹۲ بیتی)

```
id-aes192-ECB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 21 }
id-aes192-CBC OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 22 }
id-aes192-OFB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 23 }
id-aes192-CFB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 24 }
aes192-ECB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes192-ECB }
aes192-CBC ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes192-CBC }
aes192-OFB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes192-OFB }
aes192-CFB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes192-CFB }
```

ب-۲-۵ AES (کلید ۲۵۶ بیتی)

```
id-aes256-ECB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 41 }
id-aes256-CBC OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 42 }
id-aes256-OFB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 43 }
id-aes256-CFB OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 1 44 }
aes256-ECB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes256-ECB }
aes256-CBC ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes256-CBC }
aes256-OFB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes256-OFB }
aes256-CFB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-aes256-CFB }
```

ب-۲-۶ DES

```
id-des-ECB OBJECT IDENTIFIER ::= { 1 3 14 3 2 6 }
id-des-CBC OBJECT IDENTIFIER ::= { 1 3 14 3 2 7 }
-- Carries an IV as a parameter.
id-des-OFB OBJECT IDENTIFIER ::= { 1 3 14 3 2 8 }
-- Carries an FBParameter as a parameter.
-- where,
-- FBParameter ::= SEQUENCE {
-- iv IV,
-- numberOfBits NumberOfBits
-- }
-- and,
-- IV ::= OCTET STRING
-- NumberOfBits ::= INTEGER
-- number of feedback bits (1-64)
des-ECB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-des-ECB }
```

```
des-CBC ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-des-CBC }
des-OFB ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-des-OFB }
```

ب-۲-۷ موارد دیگر

```
id-des-EDE3-CBC OBJECT IDENTIFIER ::= { 1 2 840 113549 3 7}
id-rc2CBC OBJECT IDENTIFIER ::= { 1 2 840 113549 3 2}
id-rc5-CBC-PAD OBJECT IDENTIFIER ::= { 1 2 840 113549 3 9}
des-EDE3-CBC ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-des-EDE3-CBC }
rc2CBC ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-rc2CBC }
rc5-CBC-PAD ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-rc5-CBC-PAD }
```

ب-۳ الگوریتم‌های نامتقارن

ب-۳-۱ کلیدهای عمومی

```
id-rsa-public-key OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 1 }
id-dsa-public-key OBJECT IDENTIFIER ::= { 1 2 840 10040 4 1 }
id-ec-public-key OBJECT IDENTIFIER ::= { 1 2 840 10045 2 1 }
rsa-public-key ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-rsa-public-key }
dsa-public-key ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dsa-public-key }
ec-public-key ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ec-public-key }
```

ب-۳-۲ رمزنگاشتی نامتقارن

```
id-rsa-oaep OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 7 }
rsa-oaep ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-rsa-oaep }
```

ب-۳-۳ امضاها

```
id-dsa-with-SHA-1 OBJECT IDENTIFIER ::= { 1 2 840 10040 4 3 }
id-md2WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 2 }
id-md5WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 4 }
id-sha-1WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 5 }
id-sha224WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 14 }
id-sha256WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 11 }
id-sha384WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 12 }
id-sha512WithRSAEncryption OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 13 }
id-rsaSSA-PSS OBJECT IDENTIFIER ::= { 1 2 840 113549 1 1 10 }
-- same OID for all hash algorithms (hash algorithm is specified as a parameter).
dsa-with-SHA-1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dsa-with-SHA-1 }
md2WithRSAEncryption ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-md2WithRSAEncryption }
md5WithRSAEncryption ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-md5WithRSAEncryption }
sha-1WithRSAEncryption ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-sha-1WithRSAEncryption }
sha224WithRSAEncryption ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-sha224WithRSAEncryption }
sha256WithRSAEncryption ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-sha256WithRSAEncryption }
sha384WithRSAEncryption ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-sha384WithRSAEncryption }
```

```
sha512WithRSAEncryption ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-sha512WithRSAEncryption }
rsaSSA-PSS ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-rsaSSA-PSS }
```

ب-۴ الگوریتم‌های منحنی بیضوی

ب-۴-۱ منحنی‌های بیضوی

```
id-ansiX9p192r1 OBJECT IDENTIFIER ::= { 1 2 840 10045 3 1 1 }
id-ansiX9t163k1 OBJECT IDENTIFIER ::= { 1 3 132 0 1 }
id-ansiX9t163r2 OBJECT IDENTIFIER ::= { 1 3 132 0 15 }
id-ansiX9p224r1 OBJECT IDENTIFIER ::= { 1 3 132 0 33 }
id-ansiX9t233k1 OBJECT IDENTIFIER ::= { 1 3 132 0 26 }
id-ansiX9t233r1 OBJECT IDENTIFIER ::= { 1 3 132 0 27 }
id-ansiX9p256r1 OBJECT IDENTIFIER ::= { 1 2 840 10045 3 1 7 }
id-ansiX9t283k1 OBJECT IDENTIFIER ::= { 1 3 132 0 16 }
id-ansiX9t283r1 OBJECT IDENTIFIER ::= { 1 3 132 0 17 }
id-ansiX9p384r1 OBJECT IDENTIFIER ::= { 1 3 132 0 34 }
id-ansiX9t409k1 OBJECT IDENTIFIER ::= { 1 3 132 0 36 }
id-ansiX9t409r1 OBJECT IDENTIFIER ::= { 1 3 132 0 37 }
id-ansiX9p521r1 OBJECT IDENTIFIER ::= { 1 3 132 0 35 }
id-ansiX9t571k1 OBJECT IDENTIFIER ::= { 1 3 132 0 38 }
id-ansiX9t571r1 OBJECT IDENTIFIER ::= { 1 3 132 0 39 }
ansiX9p192r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9p192r1 }
ansiX9t163k1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t163k1 }
ansiX9t163r2 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t163r2 }
ansiX9p224r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9p224r1 }
ansiX9t233k1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t233k1 }
ansiX9t233r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t233r1 }
ansiX9p256r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9p256r1 }
ansiX9t283k1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t283k1 }
ansiX9t283r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t283r1 }
ansiX9p384r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9p384r1 }
ansiX9t409k1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t409k1 }
ansiX9t409r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t409r1 }
ansiX9p521r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9p521r1 }
ansiX9t571k1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t571k1 }
ansiX9t571r1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ansiX9t571r1 }
```

ب-۴-۲ امضاها

```
id-ecDSA-with-SHA-1 OBJECT IDENTIFIER ::= { 1 2 840 10045 4 1 }
id-ecDSA-with-SHA-224 OBJECT IDENTIFIER ::= { 1 2 840 10045 4 3 1 }
```

```
id-ecDSA-with-SHA-256 OBJECT IDENTIFIER ::= { 1 2 840 10045 4 3 2 }
id-ecDSA-with-SHA-384 OBJECT IDENTIFIER ::= { 1 2 840 10045 4 3 3 }
id-ecDSA-with-SHA-512 OBJECT IDENTIFIER ::= { 1 2 840 10045 4 3 4 }
ecDSA-with-SHA-1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ecDSA-with-SHA-1 }
ecDSA-with-SHA-224 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ecDSA-with-SHA-224 }
ecDSA-with-SHA-256 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ecDSA-with-SHA-256 }
ecDSA-with-SHA-384 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ecDSA-with-SHA-384 }
ecDSA-with-SHA-512 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-ecDSA-with-SHA-512 }
```

ب-۵ توابع درهم‌سازی

```
id-md2 OBJECT IDENTIFIER ::= { 1 2 840 113549 2 2 }
id-md5 OBJECT IDENTIFIER ::= { 1 2 840 113549 2 5 }
id-sha1 OBJECT IDENTIFIER ::= { 1 3 14 3 2 26 }
id-sha224 OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 2 4 }
id-sha256 OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 2 1 }
id-sha384 OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 2 2 }
id-sha512 OBJECT IDENTIFIER ::= { 2 16 840 1 101 3 4 2 3 }
md2 ALGORITHMIDENTIFIERPARAMETERS ::= {ID id-md2 }
md5 ALGORITHMIDENTIFIERPARAMETERS ::= {ID id-md5 }
sha1 ALGORITHMIDENTIFIERPARAMETERS ::= {ID id-sha1 }
sha224 ALGORITHMIDENTIFIERPARAMETERS ::= {ID id-sha224 }
sha256 ALGORITHMIDENTIFIERPARAMETERS ::= {ID id-sha256 }
sha384 ALGORITHMIDENTIFIERPARAMETERS ::= {ID id-sha384 }
sha512 ALGORITHMIDENTIFIERPARAMETERS ::= {ID id-sha512 }
```

ب-۶ کدهای احراز هویت پیام

```
id-hmacWithSHA1 OBJECT IDENTIFIER ::= { 1 2 840 113549 2 7}
id-hmacWithSHA224 OBJECT IDENTIFIER ::= { 1 2 840 113549 2 8}
id-hmacWithSHA256 OBJECT IDENTIFIER ::= { 1 2 840 113549 2 9}
id-hmacWithSHA384 OBJECT IDENTIFIER ::= { 1 2 840 113549 2 10}
id-hmacWithSHA512 OBJECT IDENTIFIER ::= { 1 2 840 113549 2 11}
id-hmac-MD5 OBJECT IDENTIFIER ::= { 1 3 6 1 5 5 8 1 1 }
hmacWithSHA1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-hmacWithSHA1 }
hmacWithSHA224 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-hmacWithSHA224 }
hmacWithSHA256 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-hmacWithSHA256 }
hmacWithSHA384 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-hmacWithSHA384 }
hmacWithSHA512 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-hmacWithSHA512 }
hmac-MD5 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-hmac-MD5 }
```

ب-۷ استقرار کلید

```
id-dhSinglePass-stdDH-sha1kdf-scheme OBJECT IDENTIFIER ::=
{ 1 3 133 16 840 63 0 2 }
id-dhSinglePass-cofactorDH-sha1kdf-scheme OBJECT IDENTIFIER ::=
{ 1 3 133 16 840 63 0 3 }
id-mqvSinglePass-sha1kdf-scheme OBJECT IDENTIFIER ::=
{ 1 3 133 16 840 63 0 16 }
```

```
id-x9-63-scheme OBJECT IDENTIFIER ::=
{ 1 2 840 63 0 }
id-secg-scheme OBJECT IDENTIFIER ::= { 1 3 132 1 }
id-dhSinglePass-stdDH-sha1kdf OBJECT IDENTIFIER ::=
{ 1 2 840 63 0 2 }
id-dhSinglePass-cofactorDH-sha1kdf OBJECT IDENTIFIER ::=
{ 1 2 840 63 0 3 }
id-dhSinglePass-cofactorDH-recommendedKDF OBJECT IDENTIFIER ::= { 1 3 132 1 1 }
id-dhSinglePass-cofactorDH-specifiedKDF OBJECT IDENTIFIER ::= { 1 3 132 1 2 }
id-iso-kdf1 OBJECT IDENTIFIER ::=
{ 1 0 18033 2 5 1 }
dhSinglePass-stdDH-sha1kdf-scheme ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dhSinglePass-stdDH-sha1kdf-scheme }
dhSinglePass-cofactorDH-sha1kdf-scheme ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dhSinglePass-cofactorDH-sha1kdf-scheme }
mqvSinglePass-sha1kdf-scheme ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-mqvSinglePass-sha1kdf-scheme }
x9-63-scheme ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-x9-63-scheme }
secg-scheme ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-secg-scheme }
dhSinglePass-stdDH-sha1kdf ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dhSinglePass-stdDH-sha1kdf }
dhSinglePass-cofactorDH-sha1kdf ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dhSinglePass-cofactorDH-sha1kdf }
dhSinglePass-cofactorDH-recommendedKDF ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dhSinglePass-cofactorDH-recommendedKDF }
dhSinglePass-cofactorDH-specifiedKDF ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-dhSinglePass-cofactorDH-specifiedKDF }
iso-kdf1 ALGORITHMIDENTIFIERPARAMETERS ::=
{ID id-iso-kdf1 TYPE OBJECT IDENTIFIER}
```

ب-۸ برشمردن الگوریتم‌های پشتیبانی‌شده

```
SupportedAlgorithms ALGORITHMIDENTIFIERPARAMETERS ::=
{ unknownAlgorithmIdentifier |
-- B.2 Symmetric Algorithms
is18033-3 |
bc64 |
bc128 |
bc64-tdea |
bc64-misty1 |
bc64-cast128 |
bc128-aes |
bc128-camellia |
bc128-seed |
-- B 2.3 AES (128-bit key)
aes128-ECB |
aes128-CBC |
aes128-OFB |
aes128-CFB |
-- B.2.4 AES (192-bit key)
aes192-ECB |
aes192-CBC |
aes192-OFB |
aes192-CFB |
-- B.2.5 AES (256 bit key)
aes256-ECB |
aes256-CBC |
aes256-OFB |
```

aes256-CFB |
-- B.2.6 DES
des-ECB |
des-CBC |
des-OFB |
-- B.2.7 Others
des-EDE3-CBC |
rc2CBC |
rc5-CBC-PAD |
-- B.3 Asymmetric Algorithms
-- B.3.1 Public Keys
rsa-public-key |
dsa-public-key |
ec-public-key |
-- B.3.2 Asymmetric Encryption
rsa-oaep |
-- B.3.3 Signatures
dsa-with-SHA-1 |
md2WithRSAEncryption |
md5WithRSAEncryption |
sha-1WithRSAEncryption |
sha224WithRSAEncryption |
sha256WithRSAEncryption |
sha384WithRSAEncryption |
sha512WithRSAEncryption |
rsaSSA-PSS |
-- B.4 Elliptic Curve Algorithms
-- B.4.1 Elliptic Curves
ansiX9p192r1 |
ansiX9t163k1 |
ansiX9t163r2 |
ansiX9p224r1 |
ansiX9t233k1 |
ansiX9t233r1 |
ansiX9p256r1 |
ansiX9t283k1 |
ansiX9t283r1 |
ansiX9p384r1 |
ansiX9t409k1 |
ansiX9t409r1 |
ansiX9p521r1 |
ansiX9t571k1 |
ansiX9t571r1 |
-- B.4.2 Signatures
ecDSA-with-SHA-1 |
ecDSA-with-SHA-224 |
ecDSA-with-SHA-256 |
ecDSA-with-SHA-384 |
ecDSA-with-SHA-512 |
-- B.5 Hash Functions
md2 |
md5 |
sha1 |
sha224 |
sha256 |
sha384 |
sha512 |
-- B.6 Message Authentication Codes
hmacWithSHA1 |

```
hmacWithSHA224 |
hmacWithSHA256 |
hmacWithSHA384 |
hmacWithSHA512 |
hmac-MD5 |
-- B.7 Key Establishment
dhSinglePass-stdDH-sha1kdf-scheme |
dhSinglePass-cofactorDH-sha1kdf-scheme |
mqvSinglePass-sha1kdf-scheme |
x9-63-scheme |
secg-scheme |
dhSinglePass-stdDH-sha1kdf |
dhSinglePass-cofactorDH-sha1kdf |
dhSinglePass-cofactorDH-recommendedKDF |
dhSinglePass-cofactorDH-specifiedKDF |
iso-kdf1 }
END
```

پیوست پ

(الزامی)

ارائه ASN.1

پ-۱ کلیات

```
INSO 16386-3-API {INSO (1) standard(0) INSO 16386 (16386) part3(3) annexC(14) }
-- Version 1.74, 05-Mar-2010
--
-- IF-PROFILE value '01'
--
-- *According to ISO/IEC 24727-2, the optional IF-PROFILE field in the CCD is
-- used to indicate that a card provides an implementation of ISO/IEC 24727-3.
-- © ISO/IEC 2008-2010
-- All rights reserved. Unless otherwise specified, no part of this publication
-- may be reproduced or utilized in any form or by any means, electronic or
-- mechanical, including photocopying and microfilm, without permission in
-- writing from either ISO at the address below or ISO's member body in the
-- country of the requester.
--
-- ISO copyright office
-- Case postale 56 • CH-1211 Geneva 20
-- Tel. + 41 22 749 01 11
-- Fax + 41 22 749 09 47
-- E-mail copyright@iso.org
-- Web www.iso.org
DEFINITIONS AUTOMATIC TAGS EXTENSIBILITY IMPLIED ::=
BEGIN
--EXPORTS (all)
IMPORTS size-max-NameLength, size-max-NodePathLength, size-max-Padding,
size-max-SecurityCondition,
ByteValue, URIType,
ApplicationIdentifier, ObjectIdentifier, Name, TransactionIdentifier,
IFDAction, IFDName, GenericHandleType, GenericIdentifierType
FROM ISO24727-COMMON { iso(1) standard(0) iso24727(24727) }
AlgorithmIDParameters
FROM ISO24727-3-ALGO {iso(1) standard(0) iso24727(24727)
part3(3) annexB (13) };
-- Major and Minor Revision values for this ASN.1 Module
revMajISO24727-3-API INTEGER ::= 1
revMinISO24727-3-API INTEGER ::= 74
```

پ-۱-۱ ثابت‌ها

```
size-max-NameLength INTEGER ::= 255
size-max-NodePathLength INTEGER ::= 255
size-max-Padding INTEGER ::= 16
size-max-SecurityCondition INTEGER ::= 255
```

پ-۱-۲ نوع‌های داده‌ای

```
ConnectionHandle ::= INTEGER
CardApplicationName ::= ApplicationIdentifier
CardApplicationNameList ::= SET OF CardApplicationName
DSIName ::= Name
DSINameList ::= SET OF DSIName
DSIContent ::= OCTET STRING
```

```
DataSetName ::= Name
DataSetNameList ::= SET OF DataSetName
DIDScope ::= CHOICE {
local NULL,
global NULL
}
CardApplicationServiceLoadPackage ::= OCTET STRING
ExecuteActionRequest ::= OCTET STRING
ExecuteActionConfirmation ::= OCTET STRING
CipherBuffer ::= OCTET STRING
MessageBuffer ::= OCTET STRING
HashBuffer ::= OCTET STRING
SignatureBuffer ::= OCTET STRING
RandomDataBuffer ::= OCTET STRING
```

پ-۱-۳ ساختارهای داده‌ای

```
DifferentialIdentityAuthenticationState ::= SEQUENCE {
dIDName DIDName,
dIDScope DIDScope,
dIDState BOOLEAN
}
SecurityCondition ::= CHOICE {
didAuthentication DifferentialIdentityAuthenticationState,
always BOOLEAN (TRUE),
never BOOLEAN (FALSE),
not SecurityCondition,
and SEQUENCE SIZE (1..size-max-SecurityCondition) OF
SecurityCondition,
or SEQUENCE SIZE (1..size-max-SecurityCondition) OF
SecurityCondition
}
AccessRule ::= SEQUENCE {
cardApplicationService CardApplicationServiceName,
action ActionName,
securityCondition SecurityCondition
}
AccessControlList ::= SET OF AccessRule
CardApplicationPathInfo ::= SEQUENCE {
pathSecurity PathSecurityType OPTIONAL,
-- The pathSecurity element specifies the protection
-- between the local dispatcher and the remote dispatcher
-- which is located at channelHandle.protocolTerminationPoint
channelHandle ChannelHandleType OPTIONAL,
contextHandle ContextHandleType OPTIONAL,
iFDName IFDName OPTIONAL,
slotIndex INTEGER OPTIONAL,
cardApplication ApplicationIdentifier
}
PathSecurityType ::= SEQUENCE {
protocol PATHSECURITYPARAMETERS.&id({SupportedPathSecurityProtocols}),
parameters PATHSECURITYPARAMETERS.&Type({SupportedPathSecurityProtocols} {@protocol}) OPTIONAL
}
unknownPathSecurityProtocolOID OBJECT IDENTIFIER ::= { iso(1) standard(0) iso24727(24727) part3(3)
annex-c(2) pathSecurityProtocols(0) unknown(0) }
unknownPathSecurityProtocol PATHSECURITYPARAMETERS ::= {
ID unknownPathSecurityProtocolOID}

SupportedPathSecurityProtocols PATHSECURITYPARAMETERS ::=
{unknownPathSecurityProtocol}
PATHSECURITYPARAMETERS ::= CLASS {
&id OBJECT IDENTIFIER,
&Type OPTIONAL
} WITH SYNTAX {ID &id
[TYPE &Type]}
ProtocolTerminationPoint ::= URIType
ChannelHandleType ::= SEQUENCE {
protocolTerminationPoint ProtocolTerminationPoint OPTIONAL,
sessionIdentifier GenericIdentifierType OPTIONAL,
binding URIType OPTIONAL
}
ContextHandleType ::= GenericHandleType
CardApplicationPathSet ::= SET OF CardApplicationPathInfo
CardApplicationServiceDescription ::= CHOICE {
serviceDescriptionText VisibleString,
```

```
serviceDescriptionURL URL
}
CertificateInfo ::= CHOICE {
efidOrPath OCTET STRING,
certificateContent OCTET STRING
}
CertificateType ::= CHOICE {
typeIndex INTEGER,
typeID OBJECT IDENTIFIER
}
DSI ::= SEQUENCE {
dsiName DSIName,
dsiContent DSIContent
}
DSIReference ::= SEQUENCE {
aID ApplicationIdentifier,
dataSetName DataSetName,
dsiName DSIName
}
DataSet ::= SEQUENCE {
dsName DataSetName,
dsContent SEQUENCE OF DSI,
dsACL AccessControlList
}
DIDQualifier ::= CHOICE {
applicationIdentifier ApplicationIdentifier,
objectIdentifier OBJECT IDENTIFIER,
applicationFunction BIT STRING
}
DIDUpdateData ::= SEQUENCE {
marker OCTET STRING,
qualifier DIDQualifier OPTIONAL
}
DIDStructure ::= SEQUENCE {
name DIDName,
protocol OBJECT IDENTIFIER,
scope DIDScope,
authenticated BOOLEAN,
marker OCTET STRING,
qualifier DIDQualifier OPTIONAL
}
DIDAuthenticationData ::= OCTET STRING
DIDName ::= Name
DIDReference ::= SEQUENCE {
scope DIDScope,
dIDName DIDName
}
DIDNameList ::= SET OF DIDName

TargetName ::= CHOICE {
datasetName DataSetName,
didName DIDName,
cardApplicationName CardApplicationName
}
TargetType ::= VisibleString (CONSTRAINED BY{
```

" -- مجموعه داده"

```
-- "DIFFERENTIAL-IDENTITY"
-- "CARD-APPLICATION"
})
URL ::= CHOICE {
printable PrintableString,
ia5 IA5String
}
-- Secure Transport Syntax, an adaptation of PKCS#7 envelopedData content type – RFC 2315
ContentInfo ::= SEQUENCE {
contentType ContentType,
content EnvelopedData
}
ContentType ::= OBJECT IDENTIFIER
EnvelopeDataVersion ::= INTEGER (0)
CertificateSerialNumber ::= INTEGER
KeyEncryptionAlgorithmIdentifier ::= AlgorithmIDParameters
ContentEncryptionAlgorithmIdentifier ::= AlgorithmIDParameters
```

```
EnvelopedData ::= SEQUENCE {
version EnvelopeDataVersion,
recipientInfos RecipientInfos,
encryptedContentInfo EncryptedContentInfo
}
RecipientInfos ::= SET OF RecipientInfo
RecipientInfo ::= SEQUENCE {
version EnvelopeDataVersion,
issuerAndSerialNumber IssuerAndSerialNumber,
keyEncryptionAlgorithm KeyEncryptionAlgorithmIdentifier,
encryptedKey EncryptedKey
}
IssuerAndSerialNumber ::= SEQUENCE {
issuer Name,
serialNumber CertificateSerialNumber
}
EncryptedKey ::= OCTET STRING
EncryptedContentInfo ::= SEQUENCE {
contentType ContentType,
contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier,
encryptedContent EncryptedContent OPTIONAL
}
EncryptedContent ::= OCTET STRING
```

پ-۱-۴ خدمات برنامه کاربردی کارت

```
CardApplicationServiceName ::= VisibleString (CONSTRAINED BY{
-- "Connection"
-- "CardApplication"
-- "NamedData"
-- "Cryptographic"
-- "DifferentialIdentity"
-- "Authorization"
})
CardApplicationServiceNameList ::= SET OF CardApplicationServiceName
ActionName ::= CHOICE {
apiAccessEntryPoint APIAccessEntryPointName,
connectionServiceAction ConnectionServiceActionName,
cardApplicationServiceAction CardApplicationServiceActionName,
namedDataServiceAction NamedDataServiceActionName,
cryptographicServiceAction CryptographicServiceActionName,
differentialIdentityServiceAction DifferentialIdentityServiceActionName,
authorizationServiceAction AuthorizationServiceActionName
}
```

پ-۱-۵ عمل‌ها/نقاط ورودی

```
APIAccessEntryPointName ::= VisibleString (CONSTRAINED BY{
-- "Initialize"
-- "Terminate"
-- "CardApplicationPath"
})
ConnectionServiceActionName ::= VisibleString (CONSTRAINED BY{
-- "CardApplicationConnect"
-- "CardApplicationDisconnect"
-- "CardApplicationStartSession"
-- "CardApplicationEndSession"
))
CardApplicationServiceActionName ::= VisibleString (CONSTRAINED BY{
-- "CardApplicationList"
-- "CardApplicationCreate"
-- "CardApplicationDelete"
-- "CardApplicationServiceList"
-- "CardApplicationServiceCreate"
-- "CardApplicationServiceLoad"
-- "CardApplicationServiceDelete"
-- "CardApplicationServiceDescribe"
-- "ExecuteAction"
})
NamedDataServiceActionName ::= VisibleString (CONSTRAINED BY{
-- "DataSetList"
-- "DataSetCreate"
-- "DataSetSelect"
```

```
-- "DataSetDelete"
-- "DSIList"
-- "DSICreate"
-- "DSIDelete"
-- "DSIWrite"
-- "DSIRead"
})
CryptographicServiceActionName ::= VisibleString (CONSTRAINED BY{
-- "Encipher"
-- "Decipher"
-- "GetRandom"
-- "Sign"
-- "VerifySignature"
-- "VerifyCertificate"
})
DifferentialIdentityServiceActionName ::= VisibleString (CONSTRAINED BY{
-- "DIDList"
-- "DIDCreate"
-- "DIDGet"
-- "DIDUpdate"
-- "DIDDelete"
-- "DIDAuthenticate"
})
AuthorizationServiceActionName ::= VisibleString (CONSTRAINED BY{
-- "ACLList"
-- "ACLModify"
})
```

پ-۱-۶ کدهای بازگشتی

```
InitializeReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_COMMUNICATION_FAILURE"
-- "API_INCORRECT_PARAMETER"
})
TerminateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_WARNING_CONNECTION_DISCONNECTED"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationPathReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_TOO_MANY_RESULTS"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationConnectReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_EXCLUSIVE_NOT_AVAILABLE"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationDisconnectReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_WARNING_SESSION_ENDED"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationStartSessionReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_NEXT_REQUEST"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_DID_ALREADY_AUTHENTICATED"
-- "API_NOT_INITIALIZED"
```

```
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationEndSessionReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NO_ACTIVE_SESSION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationListReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationCreateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAME_EXISTS"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_PREREQUISITE_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationDeleteReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_WARNING_CONNECTION_DISCONNECTED"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_PREREQUISITE_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationServiceListReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationServiceCreateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAME_EXISTS"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationServiceLoadReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationServiceDeleteReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
CardApplicationServiceDescribeReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
```

```
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
ExecuteActionReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DataSetListReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
DataSetCreateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAME_EXISTS"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DataSetSelectReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
DataSetDeleteReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
DSIListReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_PREREQUISITE_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
DSICreateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAME_EXISTS"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_PREREQUISITE_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DSIDeleteReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_PREREQUISITE_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
```

```
DSIWriteReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_PREREQUISITE_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DSIReadReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_PREREQUISITE_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
EncipherReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DecipherReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
GetRandomReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
HashReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
SignReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
```

```
VerifySignatureReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_INVALID_SIGNATURE"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
VerifyCertificateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_INVALID_KEY"
-- "API_INVALID_SIGNATURE"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DIDListReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
DIDCreateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAME_EXISTS"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DIDGetReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
DIDUpdateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
DIDDeleteReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
DIDAuthenticateReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_NEXT_REQUEST"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
```

```
-- "API_PROTOCOL_NOT_RECOGNIZED"
-- "API_INAPPROPRIATE_PROTOCOL_FOR_ACTION"
-- "API_DID_ALREADY_AUTHENTICATED"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
ACLListReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_COMMUNICATION_FAILURE"
})
ACLModifyReturnCode ::= VisibleString (CONSTRAINED BY{
-- "API_OK"
-- "API_INCORRECT_PARAMETER"
-- "API_NAMED_ENTITY_NOT_FOUND"
-- "API_NOT_INITIALIZED"
-- "API_SECURITY_CONDITION_NOT_SATISFIED"
-- "API_INSUFFICIENT_RESOURCES"
-- "API_COMMUNICATION_FAILURE"
})
```

پ-۲ واسط برنامه‌نویسی کاربردی

پ-۲-۱ Initialize

```
Initialize ::= SEQUENCE {
argument NULL OPTIONAL,
result NULL OPTIONAL,
return InitializeReturnCode
}
```

پ-۲-۲ Terminate

```
Terminate ::= SEQUENCE {
argument NULL OPTIONAL,
result NULL OPTIONAL,
return TerminateReturnCode
}
```

پ-۲-۳ CardApplicationPath

```
CardApplicationPathArgument ::= SEQUENCE {
cardAppPathRequest CardApplicationPathInfo
}
CardApplicationPathResult ::= SEQUENCE {
cardAppPathResultSet CardApplicationPathSet
}
CardApplicationPath ::= SEQUENCE {
argument CardApplicationPathArgument,
result CardApplicationPathResult OPTIONAL,
return CardApplicationPathReturnCode
}
```

پ-۲-۴ CardApplicationConnect

```
CardApplicationConnectArgument ::= SEQUENCE {
cardApplicationPath CardApplicationPathInfo,
exclusiveUse BOOLEAN
}
CardApplicationConnectResult ::= SEQUENCE {
connectionHandle ConnectionHandle
}
CardApplicationConnect ::= SEQUENCE {
argument CardApplicationConnectArgument,
result CardApplicationConnectResult OPTIONAL,
return CardApplicationConnectReturnCode
}
```

پ-۲-۵ CardApplicationDisconnect

```
CardApplicationDisconnectArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
action IFDAction OPTIONAL
```

```
}
CardApplicationDisconnect ::= SEQUENCE {
argument CardApplicationDisconnectArgument,
result NULL OPTIONAL,
return CardApplicationDisconnectReturnCode
}
```

پ-۲-۶ CardApplicationStartSession

```
CardApplicationStartSessionArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName,
authenticationProtocolData DIDAuthenticationData,
samConnectionHandle ConnectionHandle OPTIONAL
}
CardApplicationStartSessionResult ::= SEQUENCE {
authenticationProtocolData DIDAuthenticationData
}
CardApplicationStartSession ::= [APPLICATION 1031] SEQUENCE {
argument CardApplicationStartSessionArgument,
result CardApplicationStartSessionResult OPTIONAL,
return CardApplicationStartSessionReturnCode
}
```

پ-۲-۷ CardApplicationEndSession

```
CardApplicationEndSessionArgument ::= SEQUENCE {
connectionHandle ConnectionHandle
}
CardApplicationEndSession ::= [APPLICATION 1033] SEQUENCE {
argument CardApplicationEndSessionArgument,
result NULL OPTIONAL,
return CardApplicationEndSessionReturnCode
}
```

پ-۲-۸ CardApplicationList

```
CardApplicationListArgument ::= SEQUENCE {
connectionHandle ConnectionHandle
}
CardApplicationListResult ::= SEQUENCE {
cardApplicationNameList CardApplicationNameList
}
CardApplicationList ::= SEQUENCE {
argument CardApplicationListArgument,
result CardApplicationListResult OPTIONAL,
return CardApplicationListReturnCode
}
```

پ-۲-۹ CardApplicationCreate

```
CardApplicationCreateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
cardApplicationName CardApplicationName,
cardApplicationACL AccessControlList
}
CardApplicationCreate ::= SEQUENCE {
argument CardApplicationCreateArgument,
result NULL OPTIONAL,
return CardApplicationCreateReturnCode
}
```

پ-۲-۱۰ CardApplicationDelete

```
CardApplicationDeleteArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
cardApplicationName CardApplicationName
}
CardApplicationDelete ::= SEQUENCE {
argument CardApplicationDeleteArgument,
result NULL OPTIONAL,
return CardApplicationDeleteReturnCode
}
```

پ-۲-۱۱ CardApplicationServiceList

```
CardApplicationServiceListArgument ::= SEQUENCE {
connectionHandle ConnectionHandle
```

```
}
CardApplicationServiceListResult ::= SEQUENCE {
cardApplicationServiceNameList CardApplicationServiceNameList
}
CardApplicationServiceList ::= SEQUENCE {
argument CardApplicationServiceListArgument,
result CardApplicationServiceListResult OPTIONAL,
return CardApplicationServiceListReturnCode
}
```

پ-۲-۱۲ CardApplicationServiceCreate

```
CardApplicationServiceCreateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
cardApplicationServiceName CardApplicationServiceName
}
CardApplicationServiceCreate ::= SEQUENCE {
argument CardApplicationServiceCreateArgument,
result NULL OPTIONAL,
return CardApplicationServiceCreateReturnCode
}
```

پ-۲-۱۳ CardApplicationServiceLoad

```
CardApplicationServiceLoadArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
cardApplicationServiceName CardApplicationServiceName,
code CardApplicationServiceLoadPackage
}
CardApplicationSeviceLoad ::= SEQUENCE {
argument CardApplicationServiceLoadArgument,
result NULL OPTIONAL,
return CardApplicationServiceLoadReturnCode
}
```

پ-۲-۱۴ CardApplicationServiceDelete

```
CardApplicationServiceDeleteArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
cardApplicationServiceName CardApplicationServiceName
}
CardApplicationServiceDelete ::= SEQUENCE {
argument CardApplicationServiceDeleteArgument,
result NULL OPTIONAL,
return CardApplicationServiceDeleteReturnCode
}
```

پ-۲-۱۵ CardApplicationServiceDescribe

```
CardApplicationServiceDescribeArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
cardApplicationServiceName CardApplicationServiceName
}
CardApplicationServiceDescribeResult ::= SEQUENCE {
serviceDescription CardApplicationServiceDescription
}
CardApplicationServiceDescribe ::= SEQUENCE {
argument CardApplicationServiceDescribeArgument,
result CardApplicationServiceDescribeResult OPTIONAL,
return CardApplicationServiceDescribeReturnCode
}
```

پ-۲-۱۶ ExecuteAction

```
ExecuteActionArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
cardApplicationServiceName CardApplicationServiceName,
actionName ActionName,
request ExecuteActionRequest
}
ExecuteActionResult ::= SEQUENCE {
confirmation ExecuteActionConfirmation
}
ExecuteAction ::= SEQUENCE {
argument ExecuteActionArgument,
result ExecuteActionResult OPTIONAL,
return ExecuteActionReturnCode
}
```

پ-۲-۱۷ DataSetList

```
DataSetListArgument ::= SEQUENCE {
connectionHandle ConnectionHandle
}
DataSetListResult ::= SEQUENCE {
dataSetNameList DataSetNameList
}
DataSetList ::= SEQUENCE {
argument DataSetListArgument,
result DataSetListResult OPTIONAL,
return DataSetListReturnCode
}
```

پ-۲-۱۸ DataSetCreate

```
DataSetCreateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
dataSetName DataSetName,
dataSetACL AccessControlList
}
DataSetCreate ::= SEQUENCE {
argument DataSetCreateArgument,
result NULL OPTIONAL,
return DataSetCreateReturnCode
}
```

پ-۲-۱۹ DataSetSelect

```
DataSetSelectArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
dataSetName DataSetName
}
DataSetSelect ::= SEQUENCE {
argument DataSetSelectArgument,
result NULL OPTIONAL,
return DataSetSelectReturnCode
}
```

پ-۲-۲۰ DataSetDelete

```
DataSetDeleteArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
dataSetName DataSetName
}
DataSetDelete ::= SEQUENCE {
argument DataSetDeleteArgument,
result NULL OPTIONAL,
return DataSetDeleteReturnCode
}
```

پ-۲-۲۱ DSIList

```
DSIListArgument ::= SEQUENCE {
connectionHandle ConnectionHandle
}
DSIListResult ::= SEQUENCE {
dsiNameList DSINameList
}
DSIList ::= SEQUENCE {
argument DSIListArgument,
result DSIListResult OPTIONAL,
return DSIListReturnCode
}
```

پ-۲-۲۲ DSICreate

```
DSICreateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
dsiName DSIName,
dsiContent DSIContent
}
DSICreate ::= SEQUENCE {
argument DSICreateArgument,
result NULL OPTIONAL,
return DSICreateReturnCode
}
```

پ-۲-۲۳ DSIDelete

```
DSIDeleteArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
dsiName DSIName
}
DSIDelete ::= SEQUENCE {
argument DSIDeleteArgument,
result NULL OPTIONAL,
return DSIDeleteReturnCode
}
```

پ-۲-۲۴ DSIWrite

```
DSIWriteArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
dsiName DSIName,
dsiContent DSIContent
}
DSIWrite ::= SEQUENCE {
argument DSIWriteArgument,
result NULL OPTIONAL,
return DSIWriteReturnCode
}
```

پ-۲-۲۵ DSIRead

```
DSIReadArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
dsiName DSIName
}
DSIReadResult ::= SEQUENCE {
dsiContent DSIContent
}
DSIRead ::= SEQUENCE {
argument DSIReadArgument,
result DSIReadResult OPTIONAL,
return DSIReadReturnCode
}
```

پ-۲-۲۶ Encipher

```
EncipherArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName,
plainText CipherBuffer
}
EncipherResult ::= SEQUENCE {
cipherText CipherBuffer
}
Encipher ::= SEQUENCE {
argument EncipherArgument,
result EncipherResult OPTIONAL,
return EncipherReturnCode
}
```

پ-۲-۲۷ Decipher

```
DecipherArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName,
cipherText CipherBuffer
}
DecipherResult ::= SEQUENCE {
plainText CipherBuffer
}
Decipher ::= SEQUENCE {
argument DecipherArgument,
result DecipherResult OPTIONAL,
return DecipherReturnCode
}
```

پ-۲-۲۸ GetRandom

```
GetRandomArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
```

```
didScope DIDScope,
didName DIDName
}
GetRandomResult ::= SEQUENCE {
random RandomDataBuffer
}
GetRandom ::= SEQUENCE {
argument GetRandomArgument,
result GetRandomResult OPTIONAL,
return GetRandomReturnCode
}
```

پ-۲-۲۹ Hash

```
HashArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName,
message MessageBuffer
}
HashResult ::= SEQUENCE {
hash HashBuffer
}
Hash ::= SEQUENCE {
argument HashArgument,
result HashResult OPTIONAL,
return HashReturnCode
}
```

پ-۲-۳۰ Sign

```
SignArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName,
message MessageBuffer
}
SignResult ::= SEQUENCE {
signature SignatureBuffer
}
Sign ::= SEQUENCE {
argument SignArgument,
result SignResult OPTIONAL,
return SignReturnCode
}
```

پ-۲-۳۱ VerifySignature

```
VerifySignatureArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName,
signature SignatureBuffer,
message MessageBuffer
}
VerifySignature ::= SEQUENCE {
argument VerifySignatureArgument,
result NULL OPTIONAL,
return VerifySignatureReturnCode
}
```

پ-۲-۳۲ VerifyCertificate

```
VerifyCertificateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
rootCert DIDName,
certificateType CertificateType,
certificate CertificateInfo
}
VerifyCertificate ::= SEQUENCE {
argument VerifyCertificateArgument,
result NULL OPTIONAL,
return VerifyCertificateReturnCode
}
```

پ-۲-۳۳ DIDList

```
DIDListArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
filter DIDQualifier OPTIONAL
}
DIDListResult ::= SEQUENCE {
didNameList DIDNameList
}
DIDList ::= SEQUENCE {
argument DIDListArgument,
result DIDListResult OPTIONAL,
return DIDListReturnCode
}
```

پ-۲-۳۴ DIDCreate

```
DIDCreateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didName DIDName,
authProtocolOID ObjectIdentifier,
didUpdateData DIDUpdateData,
didACL AccessControlList
}
DIDCreate ::= SEQUENCE {
argument DIDCreateArgument,
result NULL OPTIONAL,
return DIDCreateReturnCode
}
```

پ-۲-۳۵ DIDGet

```
DIDGetArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName
}
DIDGetResult ::= SEQUENCE {
didStructure DIDStructure
}
DIDGet ::= SEQUENCE {
argument DIDGetArgument,
result DIDGetResult OPTIONAL,
return DIDGetReturnCode
}
```

پ-۲-۳۶ DIDUpdate

```
DIDUpdateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didName DIDName,
didUpdateData DIDUpdateData
}
DIDUpdate ::= SEQUENCE {
argument DIDUpdateArgument,
result NULL OPTIONAL,
return DIDUpdateReturnCode
}
```

پ-۲-۳۷ DIDDelete

```
DIDDeleteArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didName DIDName
}
DIDDelete ::= SEQUENCE {
argument DIDDeleteArgument,
result NULL OPTIONAL,
return DIDDeleteReturnCode
}
```

پ-۲-۳۸ DIDAuthenticate

```
DIDAuthenticateArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
didScope DIDScope,
didName DIDName,
authenticationProtocolData DIDAuthenticationData,
```

```
samConnectionHandle ConnectionHandle OPTIONAL
}
DIDAuthenticateResult ::= SEQUENCE {
authenticationProtocolData DIDAuthenticationData
}
DIDAuthenticate ::= SEQUENCE {
argument DIDAuthenticateArgument,
result DIDAuthenticateResult OPTIONAL,
return DIDAuthenticateReturnCode
}
```

پ-۲-۳۹ ACLList

```
ACLListArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
targetType TargetType,
targetName TargetName
}
ACLListResult ::= SEQUENCE {
targetACL AccessControlList
}
ACLList ::= SEQUENCE {
argument ACLListArgument,
result ACLListResult OPTIONAL,
return ACLListReturnCode
}
```

پ-۲-۴۰ ACLModify

```
ACLAccessRuleModifyArgument ::= SEQUENCE {
connectionHandle ConnectionHandle,
targetType TargetType,
targetName TargetName,
cardApplicationServiceName CardApplicationServiceName,
actionName ActionName,
securityCondition SecurityCondition
}
ACLAccessRuleModify ::= SEQUENCE {
argument ACLAccessRuleModifyArgument,
result NULL OPTIONAL,
return ACLModifyReturnCode
}
```

پ-۳ واسط فراخوانی از راه دور (Marshalling/Unmarshalling)

```
ServiceChoice ::= [APPLICATION 2101] CHOICE {
apiAccessChoice APIAccessChoice,
connectionServiceChoice ConnectionServiceChoice,
cardApplicationServiceChoice CardApplicationServiceChoice,
namedDataServiceChoice NamedDataServiceChoice,
cryptographicServiceChoice CryptographicServiceChoice,
differentialIdentityServiceChoice DifferentialIdentityServiceChoice,
authorizationServiceChoice AuthorizationServiceChoice
}
APIAccessChoice ::= [APPLICATION 2102] CHOICE {
initialize InitializeCall,
terminate TerminateCall,
cardApplicationPath CardApplicationPathCall
}
ConnectionServiceChoice ::= [APPLICATION 2103] CHOICE {
cardApplicationConnect CardApplicationConnectCall,
cardApplicationDisconnect CardApplicationDisconnectCall,
cardApplicationStartSession CardApplicationStartSessionCall,
cardApplicationEndSession CardApplicationEndSessionCall
}
CardApplicationServiceChoice ::= [APPLICATION 2104] CHOICE {
cardApplicationList CardApplicationListCall,
cardApplicationCreate CardApplicationCreateCall,
cardApplicationDelete CardApplicationDeleteCall,
cardApplicationServiceList CardApplicationServiceListCall,
cardApplicationServiceCreate CardApplicationServiceCreateCall,
cardApplicationServiceLoad CardApplicationServiceLoadCall,
cardApplicationServiceDelete CardApplicationServiceDeleteCall,
cardApplicationServiceDescribe CardApplicationServiceDescribeCall,
executeAction ExecuteActionCall
}
NamedDataServiceChoice ::= [APPLICATION 2105] CHOICE {
```

```
dataSetList DataSetListCall,
dataSetCreate DataSetCreateCall,
dataSetSelect DataSetSelectCall,
dataSetDelete DataSetDeleteCall,
dSIList DSIListCall,
dSICreate DSICreateCall,
dSIDelete DSIDeleteCall,
dSIWrite DSIWriteCall,
dSIRead DSIReadCall
}
CryptographicServiceChoice ::= [APPLICATION 2106] CHOICE {
encipher EncipherCall,
decipher DecipherCall,
getRandom GetRandomCall,
hash HashCall,
sign SignCall,
verifySignature VerifySignatureCall,
verifyCertificate VerifyCertificateCall
}
DifferentialIdentityServiceChoice ::= [APPLICATION 2107] CHOICE {
didList DIDListCall,
didCreate DIDCreateCall,
didGet DIDGetCall,
didUpdate DIDUpdateCall,
didDelete DIDDeleteCall,
didAuthenticate DIDAuthenticateCall
}
AuthorizationServiceChoice ::= [APPLICATION 2108] CHOICE {
aclList ACLListCall,
aclModify ACLModifyCall
}
```

پ-۴ خدمت اتصال

پ-۴-۱ Initialize

```
InitializeCall ::= [APPLICATION 2001] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument NULL OPTIONAL
}
InitializeReturn ::= [APPLICATION 2002] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode InitializeReturnCode
}
```

پ-۴-۲ Terminate

```
TerminateCall ::= [APPLICATION 2003] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument NULL OPTIONAL
}
TerminateReturn ::= [APPLICATION 2004] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode TerminateReturnCode
}
```

پ-۴-۳ CardApplicationPath

```
CardApplicationPathCall ::= [APPLICATION 2005] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationPathArgument
}
CardApplicationPathReturn ::= [APPLICATION 2006] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result CardApplicationPathResult OPTIONAL,
returnCode CardApplicationPathReturnCode
}
```

پ-۴-۴ CardApplicationConnect

```
CardApplicationConnectCall ::= [APPLICATION 2007] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationConnectArgument
}
CardApplicationConnectReturn ::= [APPLICATION 2008] SEQUENCE {
```

```
transactionIdentifier TransactionIdentifier OPTIONAL,
result CardApplicationConnectResult OPTIONAL,
returnCode CardApplicationConnectReturnCode
}
```

پ-۴-۵ CardApplicationDisconnect

```
CardApplicationDisconnectCall ::= [APPLICATION 2009] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationDisconnectArgument
}
CardApplicationDisconnectReturn ::= [APPLICATION 2010] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode CardApplicationDisconnectReturnCode
}
```

پ-۴-۶ CardApplicationStartSession

```
CardApplicationStartSessionCall ::= [APPLICATION 2011] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationStartSessionArgument
}
CardApplicationStartSessionReturn ::= [APPLICATION 2012] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result CardApplicationStartSessionResult OPTIONAL,
returnCode CardApplicationStartSessionReturnCode
}
```

پ-۴-۷ CardApplicationEndSession

```
CardApplicationEndSessionCall ::= [APPLICATION 2013] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationEndSessionArgument
}
CardApplicationEndSessionReturn ::= [APPLICATION 2014] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode CardApplicationEndSessionReturnCode
}
```

پ-۵ خدمت برنامه کاربردی کارت

پ-۵-۱ CardApplicationList

```
CardApplicationListCall ::= [APPLICATION 2015] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationListArgument
}
CardApplicationListReturn ::= [APPLICATION 2016] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result CardApplicationListResult OPTIONAL,
returnCode CardApplicationListReturnCode
}
```

پ-۵-۲ CardApplicationCreate

```
CardApplicationCreateCall ::= [APPLICATION 2017] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationCreateArgument
}
CardApplicationCreateReturn ::= [APPLICATION 2018] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode CardApplicationCreateReturnCode
}
```

پ-۵-۳ CardApplicationDelete

```
CardApplicationDeleteCall ::= [APPLICATION 2019] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationDeleteArgument
}
CardApplicationDeleteReturn ::= [APPLICATION 2020] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode CardApplicationDeleteReturnCode
```

```
}
```

پ-۵-۴ CardApplicationServiceList

```
CardApplicationServiceListCall ::= [APPLICATION 2021] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationServiceListArgument
}
CardApplicationServiceListReturn ::= [APPLICATION 2022] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result CardApplicationServiceListResult OPTIONAL,
returnCode CardApplicationServiceListReturnCode
}
```

پ-۵-۵ CardApplicationServiceCreate

```
CardApplicationServiceCreateCall ::= [APPLICATION 2023] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationServiceCreateArgument
}
CardApplicationServiceCreateReturn ::= [APPLICATION 2024] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode CardApplicationServiceCreateReturnCode
}
```

پ-۵-۶ CardApplicationServiceLoad

```
CardApplicationServiceLoadCall ::= [APPLICATION 2025] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationServiceLoadArgument
}
CardApplicationServiceLoadReturn ::= [APPLICATION 2026] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode CardApplicationServiceLoadReturnCode
}
```

پ-۵-۷ CardApplicationServiceDelete

```
CardApplicationServiceDeleteCall ::= [APPLICATION 2027] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationServiceDeleteArgument
}
CardApplicationServiceDeleteReturn ::= [APPLICATION 2028] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode CardApplicationServiceDeleteReturnCode
}
```

پ-۵-۸ CardApplicationServiceDescribe

```
CardApplicationServiceDescribeCall ::= [APPLICATION 2029] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument CardApplicationServiceDescribeArgument
}
CardApplicationServiceDescribeReturn ::= [APPLICATION 2030] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result CardApplicationServiceDescribeResult OPTIONAL,
returnCode CardApplicationServiceDescribeReturnCode
}
```

پ-۵-۹ ExecuteAction

```
ExecuteActionCall ::= [APPLICATION 2031] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument ExecuteActionArgument
}
ExecuteActionReturn ::= [APPLICATION 2032] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result ExecuteActionResult OPTIONAL,
returnCode ExecuteActionReturnCode
}
```

پ-۶ خدمت داده‌های نام‌گذاری‌شده

پ-۶-۱ DataSetList

```
DataSetListCall ::= [APPLICATION 2033] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
```

```
argument DataSetListArgument
}
DataSetListReturn ::= [APPLICATION 2034] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result DataSetListResult OPTIONAL,
returnCode DataSetListReturnCode
}
```

پ-۶-۲ DataSetCreate

```
DataSetCreateCall ::= [APPLICATION 2035] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DataSetCreateArgument
}
DataSetCreateReturn ::= [APPLICATION 2036] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DataSetCreateReturnCode
}
```

پ-۶-۳ DataSetSelect

```
DataSetSelectCall ::= [APPLICATION 2037] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DataSetSelectArgument
}
DataSetSelectReturn ::= [APPLICATION 2038] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DataSetSelectReturnCode
}
```

پ-۶-۴ DataSetDelete

```
DataSetDeleteCall ::= [APPLICATION 2039] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DataSetDeleteArgument
}
DataSetDeleteReturn ::= [APPLICATION 2040] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DataSetDeleteReturnCode
}
```

پ-۶-۵ DSIList

```
DSIListCall ::= [APPLICATION 2041] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DSIListArgument
}
DSIListReturn ::= [APPLICATION 2042] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result DSIListResult OPTIONAL,
returnCode DSIListReturnCode
}
```

پ-۶-۶ DSICreate

```
DSICreateCall ::= [APPLICATION 2043] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DSICreateArgument
}
DSICreateReturn ::= [APPLICATION 2044] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DSICreateReturnCode
}
```

پ-۶-۷ DSIDelete

```
DSIDeleteCall ::= [APPLICATION 2045] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DSIDeleteArgument
}
DSIDeleteReturn ::= [APPLICATION 2046] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DSIDeleteReturnCode
}
```

پ-۶-۸ DSIWrite

```
DSIWriteCall ::= [APPLICATION 2047] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DSIWriteArgument
}
DSIWriteReturn ::= [APPLICATION 2048] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DSIWriteReturnCode
}
```

پ-۶-۹ DSIRead

```
DSIReadCall ::= [APPLICATION 2049] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DSIReadArgument
}
DSIReadReturn ::= [APPLICATION 2050] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result DSIReadResult OPTIONAL,
returnCode DSIReadReturnCode
}
```

پ-۷ خدمت رمزنگاشتی

پ-۷-۱ Encipher

```
EncipherCall ::= [APPLICATION 2051] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument EncipherArgument
}
EncipherReturn ::= [APPLICATION 2052] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result EncipherResult OPTIONAL,
returnCode EncipherReturnCode
}
```

پ-۷-۲ Decipher

```
DecipherCall ::= [APPLICATION 2053] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DecipherArgument
}
DecipherReturn ::= [APPLICATION 2054] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result DecipherResult OPTIONAL,
returnCode DecipherReturnCode
}
```

پ-۷-۳ GetRandom

```
GetRandomCall ::= [APPLICATION 2055] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument GetRandomArgument
}
GetRandomReturn ::= [APPLICATION 2056] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result GetRandomResult OPTIONAL,
returnCode GetRandomReturnCode
}
```

پ-۷-۴ Hash

```
HashCall ::= [APPLICATION 2057] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument HashArgument
}
HashReturn ::= [APPLICATION 2058] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result HashResult OPTIONAL,
returnCode HashReturnCode
}
```

پ-۷-۵ Sign

```
SignCall ::= [APPLICATION 2059] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument SignArgument
}
```

```
SignReturn ::= [APPLICATION 2060] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result SignResult OPTIONAL,
returnCode SignReturnCode
}
```

پ-۷-۶ VerifySignature

```
VerifySignatureCall ::= [APPLICATION 2061] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument VerifySignatureArgument
}
VerifySignatureReturn ::= [APPLICATION 2062] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode VerifySignatureReturnCode
}
```

پ-۷-۷ VerifyCertificate

```
VerifyCertificateCall ::= [APPLICATION 2063] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument VerifyCertificateArgument
}
VerifyCertificateReturn ::= [APPLICATION 2064] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode VerifyCertificateReturnCode
}
```

پ-۸ خدمت هویت متمایزکننده

پ-۸-۱ DIDList

```
DIDListCall ::= [APPLICATION 2065] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DIDListArgument
}
DIDListReturn ::= [APPLICATION 2066] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result DIDListResult OPTIONAL,
returnCode DIDListReturnCode
}
```

پ-۸-۲ DIDCreate

```
DIDCreateCall ::= [APPLICATION 2067] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DIDCreateArgument
}
DIDCreateReturn ::= [APPLICATION 2068] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DIDCreateReturnCode
}
```

پ-۸-۳ DIDGet

```
DIDGetCall ::= [APPLICATION 2069] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DIDGetArgument
}
DIDGetReturn ::= [APPLICATION 2070] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result DIDGetResult OPTIONAL,
returnCode DIDGetReturnCode
}
```

پ-۸-۴ DIDUpdate

```
DIDUpdateCall ::= [APPLICATION 2071] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DIDUpdateArgument
}
DIDUpdateReturn ::= [APPLICATION 2072] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DIDUpdateReturnCode
}
```

پ-۸-۵ DIDDelete

```
DIDDeleteCall ::= [APPLICATION 2073] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DIDDeleteArgument
}
DIDDeleteReturn ::= [APPLICATION 2074] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode DIDDeleteReturnCode
}
```

پ-۸-۶ DIDAuthenticate

```
DIDAuthenticateCall ::= [APPLICATION 2075] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument DIDAuthenticateArgument
}
DIDAuthenticateReturn ::= [APPLICATION 2076] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result DIDAuthenticateResult OPTIONAL,
returnCode DIDAuthenticateReturnCode
}
```

پ-۹ خدمت مجوزدهی

پ-۹-۱ ACLList

```
ACLListCall ::= [APPLICATION 2077] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument ACLListArgument
}
ACLListReturn ::= [APPLICATION 2078] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result ACLListResult OPTIONAL,
returnCode ACLListReturnCode
}
```

پ-۹-۲ ACLModify

```
ACLModifyCall ::= [APPLICATION 2079] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
argument ACLAccessRuleModifyArgument
}
ACLModifyReturn ::= [APPLICATION 2080] SEQUENCE {
transactionIdentifier TransactionIdentifier OPTIONAL,
result NULL OPTIONAL,
returnCode ACLModifyReturnCode
}
```

پ-۱۰ ساختارهای پروتکل احراز هویت

یادآوری ۱ــ بخش‌های پ-۱۰-۱ تا پ-۱۰-۳ به‌طور عمدی حذف شده‌اند، در نتیجه مخاطب ممکن است به راحتی بین الف-n و پ-۱۰-n ارجاع کند.

پ-۱۰-۴ احراز هویت داخلی نامتقارن

```
MarkerAP004 ::= SEQUENCE {
signatureAlgorithm AlgorithmIDParameters,
hashAlgorithm AlgorithmIDParameters,
keySize INTEGER,
keyPair CHOICE {
keyPairInline SEQUENCE {
publicKeyMaterial OCTET STRING,
privateKey OCTET STRING
},
generateFlag NULL
},
nonceSize INTEGER
}
```

پ-۱۰-۵ احراز هویت خارجی نامتقارن

```
MarkerAP005 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
hashAlgorithm AlgorithmIDParameters,
```

```
keySize INTEGER,
publicKeyMaterial OCTET STRING,
nonceSize INTEGER
}
```

پ-۱۰-۶ احراز هویت داخلی متقارن

```
MarkerAP006 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
hashAlgorithm AlgorithmIDParameters,
keySize INTEGER,
secretKey OCTET STRING,
nonceSize INTEGER
}
```

پ-۱۰-۷ احراز هویت خارجی متقارن

```
MarkerAP007 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
hashAlgorithm AlgorithmIDParameters,
keySize INTEGER,
secretKey OCTET STRING,
nonceSize INTEGER
}
```

پ-۱۰-۸ مقایسه

```
MarkerAP008 ::= SEQUENCE {
minDataLength INTEGER,
maxDataLength INTEGER,
paddingCharacter OCTET STRING,
markerTemplate OCTET STRING
}
```

پ-۱۰-۹ مقایسه PIN

```
MarkerAP009 ::= SEQUENCE {
minDataLength INTEGER,
maxDataLength INTEGER,
storedLength INTEGER,
paddingCharacter OCTET STRING,
markerTemplate OCTET STRING,
maxAttempts INTEGER,
attemptsCounter INTEGER,
pinRef OCTET STRING,
pinValue VisibleString
}
```

پ-۱۰-۱۰ مقایسه زیست سنجی

```
MarkerAP010 ::= SEQUENCE {
bit OCTET STRING,
markerTemplate OCTET STRING
}
```

پ-۱۰-۱۱ احراز هویت دوطرفه با استقرار کلید

```
MarkerAP011 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
macAlgorithm AlgorithmIDParameters,
derivationAlgorithmK-enc AlgorithmIDParameters,
derivationAlgorithmK-mac AlgorithmIDParameters,
derivationAlgorithmK-IFD AlgorithmIDParameters,
derivationAlgorithmSessionKeysAndCounters AlgorithmIDParameters
}
```

پ-۱۰-۱۲ احراز هویت دوطرفه برنامه کاربردی سرویس‌گیرنده با استقرار کلید

```
MarkerAP012 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
macAlgorithm AlgorithmIDParameters,
encryptionAlgorithmForSessionKey AlgorithmIDParameters,
macAlgorithmForSessionKey AlgorithmIDParameters,
derivationAlgorithmSessionKeysAndCounter AlgorithmIDParameters
}
```

پ-۱۰-۱۳ احراز هویت خارجی نامتقارن برنامه کاربردی سرویس‌گیرنده

```
MarkerAP013 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
```

```
hashAlgorithm AlgorithmIDParameters,
keySize INTEGER,
publicKeyMaterial OCTET STRING,
nonceSize INTEGER
}
```

پ-۱۰-۱۴ پروتکل کنترل دسترسی توسعه یافته پودمانی (M-EAC)

```
MarkerAP014 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
hashAlgorithm AlgorithmIDParameters,
encryptionAlgorithmForSessionKey AlgorithmIDParameters,
macAlgorithmForSessionKey AlgorithmIDParameters,
k-enc OCTET STRING,
k-mac OCTET STRING,
derivationAlgorithmK-enc AlgorithmIDParameters,
derivationAlgorithmK-mac AlgorithmIDParameters,
keySize INTEGER,
keyPair CHOICE {
keyPairInline SEQUENCE {
publicKeyMaterial OCTET STRING,
privateKey OCTET STRING
},
generateFlag NULL
},
nonceSize INTEGER
}
```

پ-۱۰-۱۵ انتقال کلید با احراز هویت دوطرفه مبتنی بر RSA

```
MarkerAP015 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
hashAlgorithm AlgorithmIDParameters,
keySize INTEGER,
keyPair CHOICE {
keyPairInline SEQUENCE {
publicKeyMaterial OCTET STRING,
privateKey OCTET STRING
},
generateFlag NULL
},
nonceSize INTEGER
}
```

پ-۱۰-۱۶ دستیابی به سن

```
MarkerAP016 ::= SEQUENCE {
dateOfBirthReference DSIReference,
attainedDate INTEGER
}
```

پ-۱۰-۱۷ برقراری نامتقارن کلید جلسه

```
MarkerAP017 ::= SEQUENCE {
encryptionAlgorithm AlgorithmIDParameters,
keySize INTEGER,
keyTransportProtectionType INTEGER,
privateTransKeyReference DIDReference,
keyPair CHOICE {
byValue SEQUENCE {
privateKey OCTET STRING,
publicKeyMaterial OCTET STRING
},
ByReference SEQUENCE {
privateKeyReference DIDReference,
publicKeyReference DIDReference
},
generateFlag NULL
},
sessionMACAlgorithm AlgorithmIDParameters,
sessionENCAlgorithm AlgorithmIDParameters
}
```

پ-۱۰-۱۸ مقایسه PIN ایمن

```
MarkerAP018 ::= SEQUENCE {
minDataLength INTEGER,
maxDataLength INTEGER,
```

```
paddingCharacter OCTET STRING,
markerTemplate OCTET STRING,
maxAttempts INTEGER,
attemptCounter INTEGER,
encryptionAlgorithm AlgorithmIDParameters,
keySize INTEGER,
keyTransportProtectionType INTEGER,
nonceSize OCTET STRING,
nonce INTEGER,
keyPair CHOICE {
byValue SEQUENCE {
privateKey OCTET STRING,
publicKeyMaterial OCTET STRING
},
byReference SEQUENCE {
privateKeyReference DIDReference,
publicKeyMaterialReference DIDReference
},
generateFlag NULL
}
}
```

پ-۱۰-۱۹ توافق کلید EC با احراز هویت برنامه کاربردی کارت

```
MarkerAP019 ::= SEQUENCE {
domainParameters AlgorithmIDParameters,
keyEstablishmentAlgorithm AlgorithmIDParameters,
kDFHashAlgorithm AlgorithmIDParameters,
sessionMacAlgorithm AlgorithmIDParameters,
sessionEncAlgorithm AlgorithmIDParameters,
nonceSize INTEGER,
keyPair CHOICE {
keyPairInline SEQUENCE {
iccPublicKey OCTET STRING,
iccPrivateKey OCTET STRING
},
genKeyPairFlag NULL
},
iccIdentifier OCTET STRING,
iccCert OCTET STRING
}
```

پ-۱۰-۲۰ توافق کلید EC با احراز هویت دوطرفه

```
MarkerAP020 ::= SEQUENCE {
domainParameters AlgorithmIDParameters,
keyEstablishmentAlgorithm AlgorithmIDParameters,
kDFHashAlgorithm AlgorithmIDParameters,
sessionMacAlgorithm AlgorithmIDParameters,
sessionEncAlgorithm AlgorithmIDParameters,
authAlgorithm AlgorithmIDParameters,
nonceSize INTEGER,
keyPair CHOICE {
keyPairInline SEQUENCE {
iccPublicKey OCTET STRING,
iccPrivateKey OCTET STRING
},
genKeyPairFlag NULL
},
iccIdentifier OCTET STRING,
iccCert OCTET STRING,
rootIdentifier OCTET STRING,
rootPublicKey OCTET STRING,
iccCertKnown BOOLEAN,
verifyClientCert BOOLEAN,
enforcePrivacy BOOLEAN
}
```

پ-۱۰-۲۱ توافق کلید EC-DH ساده

```
MarkerAP021 ::= SEQUENCE {
domainParameters AlgorithmIDParameters,
keyEstablishmentAlgorithm AlgorithmIDParameters,
kDFHashAlgorithm AlgorithmIDParameters,
sessionMacAlgorithm AlgorithmIDParameters,
sessionEncAlgorithm AlgorithmIDParameters
}
```

پ-۱۰-۲۲ احراز هویت نامتقارن GP

```
MarkerAP022 ::= SEQUENCE {
sdPublicKey OCTET STRING,
sdPrivateKey OCTET STRING,
sdCertificate OCTET STRING,
exPublicKey OCTET STRING, -- PK.TP EX.AUT as defined in GP 2.2 (F.1.2.1 Overview)
ocePublicKey OCTET STRING, -- PK.OCE.AUT as defined in GP 2.2 (F.1.2.1 Overview)
kaExPublicKey OCTET STRING, -- PK.KA EX.AUT as defined in GP 2.2 (F.1.2.1)
kaInCertificate OCTET STRING, -- CERT.KA IN.AUT as defined in GP 2.2 (F.1.2.1 Overview)
sessionMacAlgo AlgorithmIDParameters,
sessionEncAlgo AlgorithmIDParameters,
keyEncryptionAlgo AlgorithmIDParameters,
hashAlgorithm AlgorithmIDParameters,
nonceSize INTEGER
}
```

پ-۱۰-۲۳ احراز هویت متقارن GP (حالت صریح)

```
MarkerAP023 ::= SEQUENCE {
sdStaticEncKey OCTET STRING,
sdStaticMacKey OCTET STRING,
sdStaticDekKey OCTET STRING,
sdSequenceCounter OCTET STRING,
sessionMacAlgo AlgorithmIDParameters,
sessionEncAlgo AlgorithmIDParameters,
keyDerivationAlgo AlgorithmIDParameters,
nonceSize INTEGER
}
```

پ-۱۰-۲۴ احراز هویت متقارن GP (حالت ضمنی)

```
MarkerAP024 ::= SEQUENCE {
sdStaticBaseKey OCTET STRING,
sdSequenceCounter OCTET STRING,
sessionMacAlgo AlgorithmIDParameters,
sessionEncAlgo AlgorithmIDParameters,
keyDerivationAlgo AlgorithmIDParameters
}
END
```

پیوست ت

(الزامی)

ماژول COMMON– استاندارد ملی ایران شماره ۱۶۳۸۶

```
INSO 16386-COMMON { INSO(1) standard(0) INSO 16386 (16386) }
-- Version 1.5, 03-Mar-2010
--
-- IF-PROFILE value '01'
--
-- *According to ISO/IEC 24727-2, the optional IF-PROFILE field in the CCD is
-- used to indicate that a card provides an implementation of ISO/IEC 24727-3.
-- © ISO/IEC 2008-2010
-- All rights reserved. Unless otherwise specified, no part of this publication
-- may be reproduced or utilized in any form or by any means, electronic or
-- mechanical, including photocopying and microfilm, without permission in
-- writing from either ISO at the address below or ISO's member body in the
-- country of the requester.
--
-- ISO copyright office
-- Case postale 56 • CH-1211 Geneva 20
-- Tel. + 41 22 749 01 11
-- Fax + 41 22 749 09 47
-- E-mail copyright@iso.org
-- Web www.iso.org
DEFINITIONS AUTOMATIC TAGS EXTENSIBILITY IMPLIED ::=
BEGIN
-- EXPORTS (all)
IMPORTS;
-- Major and Minor Revision values for this ASN.1 Module
revMajISO24727-COMMON INTEGER ::= 1
revMinISO24727-COMMON INTEGER ::= 5
NonNegativeInt ::= INTEGER (0..32767)
PositiveInt ::= INTEGER (1..32767)
-- C.1.1 Constants
size-max-NameLength INTEGER ::= 255
size-max-NodePathLength INTEGER ::= 255
size-max-Padding INTEGER ::= 16
size-max-SecurityCondition INTEGER ::= 255
ByteValue ::= INTEGER (0..255)
ApplicationIdentifier ::= OCTET STRING
ObjectIdentifier ::= OBJECT IDENTIFIER
Name ::= VisibleString (SIZE(1..size-max-NameLength))
IFDName ::= Name
GenericHandleType ::= OCTET STRING
GenericIdentifierType ::= OCTET STRING
TransactionIdentifier ::= GenericIdentifierType
URIType ::= UTF8String
-- The URIType is a string, which represents a URI according to RFC 3986.
-- A URI may be
-- * a URL according to RFC 1738 (e.g. http://www.example.com or
-- http://192.168.1.1 )
-- * a URN according to RFC 2141, which may be used to referece for example
-- * OIDs according to RFC 3061 (e.g. oid:1.0.24727.3.0.1)
-- * IETF documents according to RFC 2648, which in turn may define
-- protocols (e.g. ietf:rfc:4346 may indicate that TLS v1.1
```

```
-- shall be used to protect a channel)
-- * phone numbers according to RFC 3966
-- The list of registered names is maintained by IANA
-- (see http://www.iana.org/assignments/urn-namespaces ).
IFDAction ::= CHOICE {
reset NULL,
unpower NULL,
eject NULL,
confiscate NULL
}
IFDSessionIdentifier ::= GenericHandleType
IFDChannelHandle ::= SEQUENCE {
ref OCTET STRING,
protocolTerminationPoint URIType OPTIONAL,
sessionID IFDSessionIdentifier OPTIONAL,
binding URIType OPTIONAL
}
END
```

پیوست ث

(اطلاعاتی)

کتابنامه

[۱] استاندارد ملی ایران ۱ـ ۱۶۲۷۴، فناوری اطلاعات ـ اتصال متقابل سامانه‌های بازـ مدل مرجع پایه ـ مدل پایه

[۲] استاندارد ملی ایران شماره ۱ـ ۸۲۳۱، کارت‌های شناسایی ـ شناسایی صادرکنندگان کارت‌ها ـ قسمت اول: سیستم شماره‌گذاری

[۳] استاندارد ملی ایران ـ ایزو ـ آی ای سی ۳-۷۸۱۶، کارت‌های شناسایی ـ کارت‌های مدار یکپارچه قسمت ۳- کارت‌های دارای اتصالات - واسط الکتریکی و پروتکل‌های انتقال

[۴] استاندارد ملی ایران ـ ایزو ـ آی ای سی ۱۲ ـ ۷۸۱۶،کارت‌های شناسایی -کارت‌های مدار یکپارچه قسمت ۱۲- کارت‌های دارای اتصالات - واسط الکتریکی USB و رویه‌های عامل

[۵] استاندارد ملی ایران ۱ ـ ۸۲۳۲، کارت‌های شناسایی ـ کارت‌های دارای مدارمجتمع قسمت ۱ـ کارت‌های دارای کنتاکت‌ها- مشخصات فیزیکی

[۶] استاندارد ملی ایران ۲ـ۸۲۳۲، کارت‌های شناسایی ـ کارت‌های مدار مجتمع ـ قسمت ۲ـ کارت‌های دارای کنتاکت‌ها - ابعاد و محل قرارگیری کنتاکت‌ها

[۷] استاندارد ملی ایران ۵ ـ۸۲۳۲، کارت های شناسایی -کارت‌های مدار مجتمع - قسمت ۵-ثبت فراهم کنندگان برنامه کاربردی

[۸] استاندارد ملی ایران ۷ـ۸۲۳۲، کارت‌های شناسایی ـ کارت‌های تماسی دارای مدار(های)مجتمع-قسمت هفتم ـ فرمان‌های بین صنایع برای زبان پرس‌وجوی ساخت‌یافته کارت(SCQL)

[۹] استاندارد ملی ایران ـ ایزو ـ آی ای سی ۲ ـ۸۸۲۴، فناوری اطلاعات -نشانه گذاری قاعده ی نحوی انتزاعی یک -(ASN.1) ویژگی شئ اطلاعاتی

[۱۰] استاندارد ملی ایران ـ ایزو ـ آی ای سی ۱ـ ۸۸۲۵، فناوری اطلاعات -قواعد کدبندی نشانه‌گذاری قاعده‌ی نحوی انتزاعی یک (ASN.1) ویژگی قواعد کدبندی پایه (BER) قواعد کدبندی متعارف (CER) و قواعد کدبندی متمایز(DER)

[۱۱] استاندارد ملی ایران ـ ایزو ـ آی ای سی ۴ ـ ۸۸۲۵ ، فناوری اطلاعات ـ قواعد کدبندی نشانه‌گذاری قاعده‌ی نحوی انتزاعی یک (ASN.1) قواعد کدبندی XML(XER)

[۱۲] استاندارد ملی ایران ـ ایزو ـ آی ای سی ۳ـ ۹۷۹۶، فن‌آوری اطلاعات ـ فنون امنیتی - طرح‌های امضای دیجیتال با قابلیت بازیابی پیام ـ قسمت سوم ـ سازوکارهای مبتنی بر لگاریتم گسسته

[۱۳] استاندارد ملی ایران شماره : ۲ ـ ۱۶۱۹۶، فناوری اطلاعات ـ فنون امنیتی ـ طرح‌های امضای رقمی(دیجیتال) با بازیابی پیام - قسمت ۲: سازوکارهای مبتنی بر تجزیه اعداد صحیح

[۱۴] استاندارد ملی ایران ـ ایزو ـ آی ای سی۱ ـ ۹۷۹۷ ، فناوری اطلاعات -فنون امنیتی -کدهای احراز هویت پیام (MAC)قسمت ۱ - سازوکارهای استفاده از رمزگذاری بلوکی

[۱۵] استاندارد ملی ایران شماره ۱ـ ۱۰۸۲۵، فناوری اطلاعات -فنون امنیتی -احراز هویت هستار قسمت ۱ـ کلیات
[۱۶] استاندارد ملی ایران شماره ۲ـ ۱۰۸۲۵، فناوری اطلاعات ـ فنون امنیتی ـ احراز هویت هستار قسمت ۲ـ سازوکارهای استفاده کننده از الگوریتم‌های پوشیده سازی متقارن
[۱۷] استاندارد ملی ایران شماره ۳ ـ ۱۰۸۲۵، فناوری اطلاعات ـ فنون امنیتی ـ احراز هویت هستارقسمت ۲ـ سازوکارهای استفاده کننده از الگوریتم‌های پوشیده‌سازی متقارن
[۱۸] استاندارد ملی ایران شماره ۴ ـ ۱۰۸۲۵، فن‌آوری اطلاعات - فنون امنیتی تشخیص هویت نهاد-قسمت چهارم- مکانیزم‌های استفاده کننده از یک تابع مقابله رمزنگاری
[۱۹] استاندارد ملی ایران شماره ۵ ـ ۱۰۸۲۵، فناوری اطلاعات ـ فنون امنیتی ـ احراز هویت هستار-قسمت ۵: سازوکارهای استفاده‌کننده از فنون دانش ـ صفر
[۲۰] استاندارد ملی ایران شماره ۶ ـ ۱۰۸۲۵، فناوری اطلاعات - فنون امنیتی - احراز هویت هستار قسمت ۶ـ سازوکارهای استفاده از انتقال دستی داده‌ها
[۲۱] استاندارد ملی ایران شماره ۹۶۰۰، فن‌آوری اطلاعات - روش‌های امنیتی - حالت‌های عملیاتی یک الگوریتم رمزنگاری قطعه‌ای N بیتی
[۲۲] استاندارد ملی ایران شماره ۱ـ ۹۵۹۸ ، فن‌آوری اطلاعات - روش‌های امنیتی ـ توابع در هم ساز قسمت اول ـ کلیات
[۲۳] استاندارد ملی ایران شماره۳ـ ۹۵۹۸، فناوری اطلاعات -فنون امنیتی -توابع درهم‌ساز ـ قسمت ۳ـ توابع درهم ساز اختصاصی
[۲۴] استاندارد ملی ایران شماره ۴ـ ۹۵۹۸، فن‌آوری اطلاعات - روش‌های امنیتی ـ توابع در هم‌ساز قسمت چهارم- توابع درهم ساز با استفاده از محاسبات پیمانه‌ای
[۲۵] استاندارد ملی ایران ایزو ـ آی ای سی ۳ـ ۱۰۵۳۶،کارت‌های شناسایی ـ کارت های مدار(های)یکپارچه بدون تماس(CICCS)ـ قسمت۳ـ سیگنال‌های الکترونیکی و رویه‌های باز نشاندن
[۲۶] استاندارد ملی ایران شماره ۱ ـ۱۱۶۸۴، کارت‌های شناسایی -کارت‌های مدار(های) مجتمع غیر تماسی –کار های جفت‌شدگی قوی- قسمت ۱-خصوصیات فیزیکی
[۲۷] استاندارد ملی ایران شماره ۳ ـ۱۰۸۲۲، فناوری اطلاعات ـ فنون امنیتی ـ مدیریت کلید ـ قسمت ۳ـ ساز و کارهای مبتنی بر فنون نامتقارن
[۲۸] استاندارد ملی ایران شماره ۴ ـ۱۰۸۲۲، فن‌آوری اطلاعات ـ فنون امنیتی ـ مدیریت کلید ـ قسمت چهارم ـ مکانیزم مبتنی بر رازهای ضعیف
[۲۹] استاندارد ملی ایران شماره ۲ ـ۱۶۲۹۰، کارت‌های شناسایی –کارت‌های مدار مجتمع بدون تماس -کارت‌های مجاورتی -قسمت ۲- توان بسامد رادیویی و واسط سیگنال
[۳۰] استاندارد ملی ایران شماره ۴ ـ۱۶۲۹۰، کارت‌های شناسایی –کارت‌های مدار مجتمع بدون تماس -کارت‌های مجاورتی ـ قسمت ۴ـ پروتکل انتقال
[۳۱] استاندارد ملی ایران شماره ۱ـ۱۱۴۹۴، فناوری اطلاعات -فنون امنیت امضاهای دیجیتال با پیوست قسمت ۱:کلیات

[۳۲] استاندارد ملی ایران ایزو ـ آی ای سی شماره ۲ ـ ۱۴۸۸۸، فناوری اطلاعات ـ فنون امنیتی - امضاهای رقمی(دیجیتالی)با پیوست قسمت ۲- سازوکارهای بر پایه عامل‌بندی صحیح
[۳۳] استاندارد ملی ایران ایزو ـ آی ای سی شماره ۳ـ ۱۴۸۸۸، فناوری اطلاعات ـ فنون امنیتی ـ- امضاهای رقمی (دیجیتال) با پیوست قسمت ۳- سازوکارهای بر پایه لگاریتم گسسته
[۳۴] استاندارد ملی ایران شماره ۱ـ۱۱۶۸۶، کارت‌های شناسایی ـ کارت‌های مدار(های) مجتمع غیر تماسی -کارت های مجاورتی (دوربرد) قسمت ۱- خصوصیات فیزیکی
[۳۵] استاندارد ملی ایران ۱ ـ ۱۰۸۲۴، فن‌آوری اطلاعات ـ فنون امنیتی الگوریتم‌های رمز نگاری- قسمت اول- کلیات
[۳۶] استاندارد ملی ایران ۳ ـ ۱۰۸۲۴، فناوری اطلاعات ـ فنون امنیتی ـ الگوریتم‌های رمز نگاری ـ قسمت ۳- رمزهای بلوکی
[۳۷] استاندارد ملی ایران ۴ ـ ۱۰۸۲۴، فن‌آوری اطلاعات ـ فنون امنیتی الگوریتم‌های رمزنگاری ـ قسمت چهارم- رمزگذاری جریانی
[38] ISO 3166-1, Codes for the representation of names of countries and their subdivisions — Part 1: Country codes
[39] ISO/IEC TR 9577, Information technology — Protocol identification in the network layer
[40] ISO/IEC 9945 (all parts), Information technology — Portable Operating System Interface (POSIX)
[41] ISO/IEC 9979, Information technology — Security techniques — Procedures for the registration of cryptographic algorithms
[42] ISO 9992-2, Financial transaction cards — Messages between the integrated circuit card and the card
accepting device — Part 2: Functions, messages (commands and responses), data elements and structures
[43] ISO/IEC 13673, Information technology — Document processing and related communication — Conformance testing for Standard Generalized Markup Language (SGML) systems
[44] ISO/IEC 16262, Information technology — ECMAScript language specification
[45] IETF RFC 1738, Uniform Resource Locators (URL), December 1994
[46] IETF RFC 1778, The String Representation of Standard Attribute Syntaxes, March 1995
[47] IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, August 1998
[48] CEN/EN 14890-1, Application interface for smart cards used as secure signature creation devices — Part 1: Basic services
[49] CEN/EN 14890-2, Application interface for smart cards used as secure signature creation devices — Part 2: Additional services
[50]ISO/IEC 7816-4:2013, Identification cards -- Integrated circuit cards -- Part 4: Organization, security and commands for interchange
[51] ISO/IEC 7816-6:2004, Identification cards -- Integrated circuit cards -- Part 6: Interindustry data elements for interchange

[52] ISO/IEC 7816-8:2004, Identification cards -- Integrated circuit cards -- Part 8: Commands for security operations
[53] ISO/IEC 7816-9:2004, Identification cards -- Integrated circuit cards -- Part 9: Commands for card management
[54] ISO/IEC 7816-10:1999, Identification cards -- Integrated circuit(s) cards with contacts -- Part 10: Electronic signals and answer to reset for synchronous cards
[55] ISO/IEC 7816-11:2004, Identification cards -- Integrated circuit cards -- Part 11: Personal verification through biometric methods
[56] ISO/IEC 7816-13:2007, Identification cards -- Integrated circuit cards -- Part 6: Interindustry data elements for interchange
[57] ISO/IEC 7816-15:2004, Identification cards -- Integrated circuit cards -- Part 15: Cryptographic information application
[58] ISO/IEC 9797-2:2011, Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function
[59] ISO/IEC 9797-3:2011, Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 3: Mechanisms using a universal hash-function
[60]ISO/IEC/IEEE 9945:2009,Information technology -- Portable Operating System Interface (POSIX®) Base Specifications, Issue 7
[61] ISO 10118-2: 2010, Information technology -- Security techniques -- Hash-functions -- Part 2 : Hash-functions using an n-bit block cipher
[62] ISO/IEC 10536-2:1995 , Identification cards -- Contactless integrated circuit(s) cards -- Part 2: Dimensions and location of coupling areas
[63] ISO/IEC 11770-1:2010 , Information technology -- Security techniques -- Key management -- Part 1: Framework
[64] ISO/IEC 11770-2:2008 , Information technology -- Security techniques -- Key management -- Part 2: Mechanisms using symmetric techniques
[65] ISO/IEC 11770-5:2011 , Information technology -- Security techniques -- Key management -- Part 5: Group key management
[66] ISO/IEC 14443-1:2008 , Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 1: Physical characteristics
[67] ISO/IEC 14443-3:2011, Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 3: Initialization and anticollision
[68]ISO/IEC 15693-2:2006 , Identification cards -- Contactless integrated circuit cards -- Vicinity cards -- Part 2: Air interface and initialization
[69] ISO/IEC 15693-3:2009, Identification cards -- Contactless integrated circuit cards -- Vicinity cards -- Part 3: Anticollision and transmission protocol
[70] ISO/IEC 18033-2:2006 , Information technology -- Security techniques -- Encryption algorithms -- Part 2: Asymmetric ciphers