مستندات فنی

اتصال به درگاه چند پرداختی اینترنتی شرکت پرداخت الکترونیک پاسارگاد

پرداخت الکترونیک
پاسارگاد
واحد پایانه های اینترنتی

چکیده

خرید اینترنتی یکی از تراکنشهای کارتی محسوب می شود که در مرکز شتاب نیز جزو تراکنش های مجاز برشمرده می شود. در این مستند قدمهای لازم برای ایجاد بستر خرید چند پرداختی در سمت وب سایت فروشنده که مایل است از طریق درگاه چند پرداختی اینترنتی شرکت پرداخت الکترونیک پاسارگاد به خریداران خود سرویس ارائه دهد، توضیح داده شده است.

کلمات کلیدی: چند پرداختی، گزارش‌های بلند، گزارش‌های بلند برون‌سازمانی

توجه : در صورت درخواست چنین سرویسی، به منظور اعمال تغییرات لازم از سمت شرکت لطفا اطلاع رسانی فرمایید.

تعاریف

تعاریف مرتبط با خریدار

- **خرید چند پرداختی**: منظور از خرید در این مستند، تراکنش خریدی است که در آن فروشنده می تواند با ارسال یک دستور پرداخت هنگام خرید، مبالغی را در تاریخ‌های مشخص و به شماره حساب های مشخص واریز نماید. به عنوان مثال آژانس مسافرتی را در نظر بگیرید که می خواهد پس از پرداخت وجه از طریق در گاه پرداخت شرکت پرداخت الکترونیک پاسارگاد توسط مشتری، قسمتی از این مبلغ به حساب هتل، قسمتی از آن به حساب آژانس هواپیمایی و .. در تاریخ های مشخص واریز گردد.
- **خریدار:** هویتی است که توسط یکی از انواع کارتهای بانکی (عضو شبکه شتاب) و با مراجعه به وب سایت مورد نظر خود تقاضای ارائه خدمات را دارد.

تعاریف مرتبط با فروشنده

- **فروشنده:** هویتی است که با آماده‌سازی بستر پرداخت اینترنتی اقدام به فروش کالا و خدمات از طریق وب سایت خود می‌نماید.
- **شماره شناسائی فروشنده (MerchantID)**: کدی است که توسط شرکت پرداخت الکترونیک پاسارگاد به فروشنده اختصاص می‌یابد و در حین انجام تراکنش برای شناسایی فروشنده از آن استفاده می‌گردد.
- **شماره شناسائی ترمینال (TerminalID)**: کدی است که توسط شرکت پرداخت الکترونیک پاسارگاد به فروشنده اختصاص می‌یابد و در حین انجام تراکنش از آن استفاده می‌گردد.

- **کلید خصوصی فروشنده (Private Key)**: کلیدی است که فروشنده برای احراز هویت از آن استفاده می‌کند و تمامی داده‌های ارسالی خود به شرکت پرداخت الکترونیک پاسارگاد را با آن کلید، امضا دیجیتال می‌کند.
- **کلید عمومی فروشنده (Public Key):** کلیدی است که شرکت پرداخت الکترونیک پاسارگاد جهت تایید امضای دیجیتالی فروشگاه از آن استفاده می‌کند.
- **سپرده فروشنده:** سپرده کوتاه مدت، جاری یا پس‌اندازی است که فروشنده جهت انجام عمل تسویه حساب با شرکت پرداخت الکترونیک پاسارگاد در یکی از شعب بانک پاسارگاد افتتاح نموده و آنرا به بانک جهت تسویه حساب تراکنش‌های انجام شده، اعلام می‌نماید.
- **مبلغ فاکتور (Invoice Amount):** مبلغ خرید است که فروشنده می‌خواهد از خریدار دریافت نماید. (این مبلغ باید کوچکتر و یا برابر با جمع مبالغ ریز پرداخت‌ها باشد.)
- **شماره فاکتور (Invoice Number):** هر خرید از فروشنده دارای شماره فاکتور خاص خود باید باشد که تماماً عددی است.
- **تاریخ فاکتور (Invoice Date):** تاریخ فاکتور خرید است و فرمت آن به انتخاب فروشگاه است. (لازم به تذکر است که تاریخ و شماره فاکتور، به گونه‌ای باید تخصیص داده شوند که ترکیب آنها شناسه یکتایی را به وجود آورد تا همیشه بتوان برای شناسایی یک تراکنش خرید از آن استفاده کرد).
- **Timestamp:** زمان ارسال داده به سایت شرکت پرداخت الکترونیک پاسارگاد را Timestamp می‌گویند که فرمت آن به شکل "YYYY/MM/DD HH:MM:SS" بوده و به تاریخ میلادی ارسال می‌گردد. اگر هرکدام از عددهای ماه، روز، ساعت، دقیقه یا ثانیه یک رقمی باشد با قراردادن یک صفر در سمت چپ آن باید عدد دو رقمی تولید شده و برای شرکت پرداخت الکترونیک پاسارگاد ارسال شود.
- **SubPayment :**یک ریز پرداخت است که دارای چندین مؤلفه است:
  - **SubPayID:** شماره اختصاص یافته به هر ریزپرداخت در یک دستور پرداخت است. این شماره باید 32 بیتی (int) و در هر subpaymentList یکتا باشد.

- **Amount**: مبلغ ریز پرداخت است که در تاریخ مشخص و به حساب مشخصی انتقال پیدا می کند. لازم به ذکر است که مجموع مبالغ ریزپرداخت ها باید کوچکتر و یا مساوی با مبلغ ارسال شده تحت عنوان invoiceAmount باشد.
- **Date**: تاریخ انجام ریزپرداخت است، که فرمت آن به شکل "YYYY/MM/DD" بوده و به تاریخ میلادی ارسال می‌گردد. اگر هرکدام از عددهای ماه یا روز یک رقمی باشد با قراردادن یک صفر در سمت چپ آن باید عدد دو رقمی تولید شده و برای درگاه پرداخت ارسال شود.
- **Account**: شماره حساب یا شماره شبای مقصد جهت واریز وجه ریزپرداخت است، که می تواند حاوی یک شماره حساب باشد که قسمت های آن توسط نقطه از هم جدا می شوند (به عنوان مثال 219.10.44039.1) و یا حاوی شماره شبای مقصد باشد که در این صورت انتقال وجه به آن توسط صدور پایا انجام خواهد شد. (مثال : IR910120020000005013136523)

توجه: ترمینال مورد استفاده دارای یک مشخصه است که تعیین می کند ریزپرداخت‌ها توسط انتقال وجه به حساب پرداخت شوند و یا توسط صدورپایا. تنظیم این مشخصه توسط کارشناسان شرکت پرداخت الکترونیک پاسارگاد صورت میگیرد. ریزپرداخت ارسالی باید مطابق با نوع انتقال مشخص شده در ترمینال باشد. در صورتی که هریک از شماره حساب های مشخص شده در ریز پرداخت ها نامعتبر باشند، یعنی در سیستم بانک پاسارگاد وجود نداشته و یا اجازه ی واریز نداشته باشند فروشگاه، هنگام ارسال فاکتور به درگاه پرداخت با پیغام "شماره سپرده ارسالی نامعتبر است" مواجه می شود. همچنین در صورتی که انتقال وجه از نوع پایا باشد، فرمت آن باید با فرمت استاندارد شبا مطابقت داشته باشد. (مثال : IR910120020000005013136523)

توجه: با توجه به آنی نبودن عملیات پایا، پذیرفته شدن انتقال وجه بین بانکی توسط بانک به منزله ی موفقیت آمیز بودن تراکنش پایا نمی باشد. به عنوان مثال ممکن است بعد از گذشت چندین ساعت به هر دلیلی تراکنش مورد نظر در نهایت برگشت خورده و موفقیت آمیز نباشد. لذا پذیرنده می بایست جهت آگاه شدن از وضعیت نهایی عملیات پایا، این مورد را از درگاه پرداخت استعلام نماید. درگاه پرداخت همواره آخرین وضعیت انتقال وجه های پایا را از سیستم بانکی استعلام نموده و برای پذیرنده ارسال می نماید.

- **Description**: توضیحات مربوط به ریزپرداخت است که در سند آن قرار می گیرد.

- **SubPaymentList**: یک xml از لیست ریز پرداخت ها به صورت زیر می باشد که به رشته ای با فرمت base64String تبدیل می شود.

```
<?xml version="1.0" encoding="utf-8"?>
<SubPaymentList>
  <SubPayments>
<SubPayment>
      <SubPayID>1</SubPayID>
      <Amount>5000000</Amount>
      <Date>2010/02/02</Date>
      <Account>219.10.44039.1</Account>
      <Description></Description>
    </SubPayment>
    <SubPayment>
      <SubPayID>2</SubPayID>
      <Amount>30000</Amount>
      <Date> 2010/04/08 </Date>
      <Account>201.800.981313.1</Account>
      <Description></Description>
    </SubPayment>
            .
            .
            .
  </SubPayments>
</SubPaymentList>
```

- **invoiceUpdateList**

یک xml از لیست ریزپرداخت‌های به روز رسانی شده به صورت زیر است که به رشته‌ای با فرمت base64String تبدیل می‌شود.

```
<?xml version="1.0" encoding="utf-8"?>
<invoiceUpdateList>
<invoiceAction invoiceUID="634253256472082172">
      <action type="Add" subPayID="4" amount="1" date="2010/02/02"
      account="219.10.44039.1"/>
      <action type="Delete" subPayID="5" amount="2" date="2010/02/02"
      account="219.10.44039.1"/>
      <action type="Edit" subPayID="1" amount="1" date="2010/02/02"
      account="219.10.44039.1"/>
      <action type="Add" subPayID="2" amount="1" date="2010/02/02"
      account="219.10.44039.1"/>
</invoiceAction>
</invoiceUpdateList>
```

لازم به ذکر است که تا زمانی می‌توان هر یک از عملیات Delete و Edit را برروی یک ریز پرداخت انجام داد که آن ریز پرداخت در حالت فعال باشد، یعنی هنوز زمان انجام آن فرا نرسیده و یا به هر دلیل انجام نشده باشد. همچنین عملیات Add (اضافه کردن یک ریز پرداخت) را در صورتی می توان انجام داد که مجموع مبالغ ریزپرداخت‌ها پس از به روزرسانی بیشتر از مبلغ پرداخت شده برای فاکتور نباشد. در این صورت به‌روز رسانی انجام نشده و ریزپرداخت‌ها در همان حالت قبلی خود باقی می‌مانند.

توجه: نوع انتقال در زمان ویرایش یک رکورد باید با نوع قبلی آن یکی باشد. به طور مثال چنان چه نوع انتقال ریز پرداخت واریز به حساب است نمی توان نوع آن را به انتقال به شبا تغییر داد.

تعاریف مرتبط با بانک و عملیات مالی

- **درگاه پرداخت اینترنتی (Internet Payment Gateway):** سایتی است متعلق به شرکت پرداخت الکترونیک پاسارگاد که در آن خریدار پس از انتخاب موارد مورد خرید خود در سایت فروشنده، به آنجا هدایت می‌شود و در آنجا مشخصات کارت و رمز خود را وارد می‌نماید و سپس شرکت پرداخت الکترونیک پاسارگاد تراکنش مورد نظر خریدار را انجام داده و در نهایت فروشنده را از نتیجه آن آگاه می‌سازد.
- **نوع تراکنش (Action):** نشان دهنده نوع عملیات مالی مورد نظر که در این سیستم شامل خرید و یا برگشت خرید است. برای خرید کد ۱۰۰۳ و برای برگشت کد ۱۰۰۴ در نظر گرفته شده است.
- **شماره رهگیری (TransactionReferenceID):** شماره‌ای است که سایت شرکت پرداخت الکترونیک پاسارگاد پس از موفقیت‌آمیز بودن تراکنش به سایت فروشنده ارسال می‌کند که به وسیله آن فروشنده می تواند از موفقیت آمیز بودن تراکنش اطلاع یابد.
- **تسویه حساب:** واریز وجوه دریافتی از خریدار به حساب فروشنده توسط شرکت پرداخت الکترونیک پاسارگاد ، در صورت موفق بودن تراکنش خرید پس از کسر کارمزد

مراحل انجام تراکنش خرید

۱. خریدار با مراجعه به وب سایت فروشنده و انتخاب کالا یا خدمات مورد نیاز، آماده پرداخت مبلغ فاکتور می‌شود.

۲. سایت فروشنده اطلاعات مربوط به تراکنش خرید را با PrivateKey خود امضا کرده و با متد POST به سایت شرکت پرداخت الکترونیک پاسارگاد (https://pep.shaparak.ir/gateway.aspx) ارسال می‌کند. به دلیل اینکه تراکنش از نوع خرید است به همراه ارسال داده‌ها خریدار نیز به سایت شرکت پرداخت الکترونیک پاسارگاد فرستاده (redirect) می‌شود. مواردی که برای این تراکنش به صورت POST به وب سایت بانک ارسال می‌شوند عبارتند از:

- InvoiceNumber
- InvoiceDate
- TerminalCode
- MerchantCode
- RedirectAddress
- Amount
- TimeStamp
- Action
- SubPaymentList (با فرمت base64)
- امضا دیجیتالی

مراحل تولید امضای دیجیتال عبارت است از:

۱. اتصال داده‌های ذکر شده به صورت زیر:

#merchantCode#terminalCode#invoiceNumber#invoiceDate#amount#redirectAddress#action#timeStamp#SubPaymentList#

۲. اجرای الگوریتم درهم‌سازی SHA1 بر روی رشته بالا.

۳. امضای رشته‌ی حاصل از بند دوم به وسیله PrivateKey، که نتیجه آن یک رشته‌ی باینری است.

۴. تبدیل رشته‌ی باینری به رشته ای با فرمت base64String، که این رشته امضای دیجیتال پذیرنده برای تراکنش خرید محسوب می‌شود.

۳. خریدار با وارد کردن شماره کارت (PAN)، کلمه عبور (PIN2)، کد اعتبارسنجی دوم (CVV2) و تاریخ انقضای کارت (Expiration Date) درخواست انجام تراکنش را برای شرکت پرداخت الکترونیک پاسارگاد ارسال می‌کند.

۴. در این مرحله تراکنش توسط شرکت پرداخت الکترونیک پاسارگاد پردازش گردیده و عملیات لازم در مرکز شتاب و بانک صادر کننده کارت انجام می‌پذیرد و در صورت صحت ورود داده‌ها و وجود وجه کافی در حساب خریدار عملیات مالی در این مرحله توسط بانک صورت می‌گیرد. یعنی وجه از حساب خریدار برداشته شده و در پایان روز کاری به حساب اصلی فروشگاه واریز می‌شود. این مبلغ بر اساس لیست ریز پرداختهای ارسالی، در تاریخ های مشخص شده از حساب فروشگاه برداشت شده و به حساب های تعیین شده واریز می‌گردد.

۵. وب سایت شرکت پرداخت الکترونیک پاسارگاد پس از انجام تراکنش خریدار را به آدرسی که در فیلد RedirectAddress قرار داده شده می‌فرستد (redirect می‌کند) و در Query String آن مقادیر زیر را قرار می‌دهد.

- InvoiceNumber (در فیلد iN)
- InvoiceDate (در فیلد iD)
- TransactionReferenceID یا همان InvoiceUID (در فیلد tref)

۶. انجام عمل تسویه حساب توسط شرکت پرداخت الکترونیک پاسارگاد که جزئیات آن در قرارداد منعقده فیمابین شرکت پرداخت الکترونیک پاسارگاد و فروشنده درج گردیده است.

برگشت خرید

در صورتی که فروشنده مایل به برگشت دادن کل خرید باشد، می تواند حداکثر تا پایان روز ارسال دستور پرداخت این کار را انجام دهد در این حالت پول به حساب خریدار برگشت داده شده و ریزپرداخت‌های مربوطه غیر فعال

می‌شوند. لازم به ذکر است که هر خرید فقط تا زمانی که هیچ یک از ریزپرداخت‌های آن انجام نشده است قابل برگشت است. توجه شود که برای تراکنش‌های برگشت از خرید، فروشنده باید شماره فاکتور و تاریخ فاکتور تراکنش خرید (تراکنش اصلی) را ارسال کند ولی TimeStamp باید با تاریخ جاری سیستم مقداردهی شود. بدین منظور مواردی که به صورت post به وب سایت شرکت پرداخت الکترویک پاسارگاد (https://pep.shaparak.ir/doRefund.aspx) فرستاده می‌شوند عبارتند از:

- InvoiceNumber
- InvoiceDate
- TerminalCode
- MerchantCode
- Amount
- TimeStamp
- امضا دیجیتالی

مراحل تولید امضای دیجیتال عبارت است از:

۱. اتصال داده‌های ذکر شده به صورت زیر:

#merchantCode#terminalCode#invoiceNumber#invoiceDate#amount# timeStamp#

۲. اجرای الگوریتم درهم‌سازی SHA1 بر روی رشته بالا.

۳. امضای رشته‌ی حاصل از بند دوم به وسیله PrivateKey، که نتیجه آن یک رشته‌ی باینری است.

۴. تبدیل رشته‌ی باینری به رشته ای با فرمت base64String، که این رشته امضای دیجیتال پذیرنده برای برگشت زدن تراکنش خرید است.

پس از ارسال موارد فوق به سایت شرکت پرداخت الکترویک پاسارگاد ، درخواست برگشت خرید توسط شرکت پردازش گردیده و xml زیر برای فروشنده ارسال می‌شود.

```
<?xml version="1.0" encoding="utf-8"?>
<actionResult>
  <result>{true|false}</result>
  <resultMessage>{پیغام خطا|عملیات با موفقیت انجام شد}</resultMessage>
</actionResult>
```

دریافت نتیجه تراکنش

سایت فروشنده با ارسال TransactionReferenceID دریافت شده از جانب شرکت پرداخت الکترویک پاسارگاد به صورت POST، به سایت شرکت ، (https://pep.shaparak.ir/CheckTransactionResult.aspx) می تواند از نتیجه تراکنش باخبر شود. لازم به ذکر است که اگر سایت فروشنده به هر دلیل موفق به دریافت TransactionReferenceID نشود می تواند با فرستادن InvoiceNumber، InvoiceDate، MerchantCode و TerminalCode به صورت POST به سایت ذکر شده از نتیجه تراکنش باخبر شود. سایت شرکت پرداخت الکترویک پاسارگاد صفحه XML زیر را برای فروشنده ارسال می کند. فروشنده پس از تطبیق **نوع تراکنش**، **شماره فاکتور**، **تاریخ فاکتور**، **شماره شناسائی فروشنده** و **شماره شناسائی ترمینال** موجود در XML با شماره و تاریخ فاکتور اصلی نتیجه تراکنش را خوانده و اقدام مقتضی را انجام می‌دهد. لازم به ذکر است که پذیرنده می‌بایست نتیجه تراکنش را چک کرده و از موفق بودن تراکنش اطمینان حاصل کند و به صرف دریافت TransactionReferenceID از شرکت پرداخت الکترونیک پاسارگاد تراکنش را موفقیت آمیز تلقی نکند.

```
<?xml version=\"1.0\" encoding=\"utf-8\"?>
<resultObj>
  <result>{true|false}</result>
  <action>{1003|1004}</action>
  <invoiceNumber>{فاکتور شماره}</invoiceNumber>
  <invoiceDate>{فاکتور تاریخ}</invoiceDate>
  <transactionReferenceID>{تراکنش شماره}</transactionReferenceID>
  <traceNumber>{پیگیری شماره}</traceNumber>
  <referenceNumber>{ارجاع شماره}</referenceNumber>
  <transactionDate>{تراکنش تاریخ}</transactionDate>
  <terminalCode>{شماره ترمینال}</terminalCode>
  <merchantCode>{شماره فروشگاه}</merchantCode>
  <amount>{مبلغ}</amount>
</resultObj>
```

نحوه به روز رسانی یک دستور پرداخت

فروشنده در دو حالت می‌تواند ریزپرداخت‌های فاکتور ارسالی را به روز کند:

۱. در حالتی که دستور پرداخت برای شرکت پرداخت الکترویک پاسارگاد ارسال شده اما به هر دلیلی پرداخت صورت نگرفته است. در این صورت فروشنده می‌تواند با ارسال مجدد همان فاکتور (شماره، تاریخ و مبلغ فاکتور، مطابق با فاکتور قبلی) لیست ریزپرداخت جدید (به روز رسانی شده) را ارسال کند. بدین ترتیب لیست جدید جایگزین لیست قبلی شده و در تاریخ های مشخص شده پرداخت صورت می گیرد.

۲. در حالتی که دستور پرداخت ارسال شده برای شرکت پرداخت الکترویک پاسارگاد با موفقیت پرداخت شده است. برای به روز رسانی ریزپرداخت‌های این دستور پرداخت سایت فروشنده اطلاعات مربوط به دستور پرداخت ویرایش شده را با PrivateKey خود امضا کرده و با متد POST به سایت شرکت پرداخت الکترویک پاسارگاد (https://pep.shaparak.ir/UpdateInvoiceSubpayment.aspx) ارسال می‌کند. مواردی که به وب سایت به صورت POST ارسال می‌شوند عبارتند از:

- TerminalCode
- MerchantCode
- InvoiceUpdateList
- TimeStamp
- Digital Signature

مراحل تولید امضای دیجیتال عبارت است از:

۱. اتصال داده‌های ذکر شده به صورت زیر:

#merchantCode#terminalCode# invoiceUpdateList#timeStamp#

۲. اجرای الگوریتم درهم‌سازی SHA1 بر روی رشته بالا.

۳. امضای رشته‌ی حاصل از بند دوم به وسیله PrivateKey، که نتیجه آن یک رشته‌ی باینری است.

۴. تبدیل رشته‌ی باینری به رشته ای با فرمت base64String، که این رشته امضای دیجیتال پذیرنده برای به روزرسانی یک دستور پرداخت است.

در این مرحله درخواست به‌روزرسانی توسط بانک پردازش گردیده و xml زیر برای فروشنده ارسال می‌شود.

```
<?xml version="1.0" encoding="utf-8"?>
<actionResult>
  <result>{true|false}</result>
  <resultMessage>{پیغام خطا|عملیات با موفقیت انجام شد}</resultMessage>
</actionResult>
```

استعلام ریزپرداخت های ارسالی بر اساس تاریخ ریزپرداخت ها

سایت فروشنده اطلاعات مربوط به استعلام ریز پرداخت ها را با Private Key خود امضا کرده و به صورت POST، به سایت شرکت پرداخت الکترویک پاسارگاد (https://pep.shaparak.ir/GetSubPaymentsResult.aspx) ارسال می کند. مواردی که به وب سایت به صورت POST ارسال می‌شوند عبارتند از:

- terminalCode
- merchantCode
- startDate (ابتدای بازه زمانی جهت دریافت ریز پزداخت)
- endDate (انتهای بازه زمانی جهت دریافت ریز پزداخت)
- timeStamp
- digital Signature

در این مرحله درخواست استعلام ریز پرداخت ها توسط بانک پردازش گردیده و xml زیر برای فروشنده ارسال می‌شود. این xml شامل آخرین حالت ریزپرداخت هایی است، که تاریخ انجام آنها در بازه ی زمانی ارسالی است.

```
<?xml version=\"1.0\" encoding=\"utf-8\"?>
<resultObj>
  <result>{true|false}</result>
  <resultMessage>{پیغام خطا|عملیات با موفقیت انجام شد}</resultMessage>
```

```
<subPayments>
 <subPayment>
   <invoiceUID>{شماره رهگیری}</invoiceUID>
   <subPayID>{شماره ریز پرداخت}</subPayID>
   <date>{تاریخ ریز پرداخت}</date>
   <amount>{مبلغ ریز پرداخت }</amount>
   <account>{شماره حساب}</account>
   <description>{شرح}</description>
   <status>{Inactive| Active | Done| Canceled}</status>
   <depositName>{نام سپرده}</depositName>
   <transactionDate>{زمان انجام ریز پرداخت}</transactionDate>
   <payaRefNumber>{شماره پیگیری پایا}</payaRefNumber>
   <payaStatus>{آخرین وضیعت پایا}</payaStatus>
   <payaStatusDesc>{شرح آخرین وضیعت پایا}</payaStatusDesc>
   <bankName>{نام بانک مقصد در انتقال پایا}</bankName>
 </subPayment>
 <subPayment>
   <invoiceUID>{شماره رهگیری}</invoiceUID>
   <subPayID>{شماره ریز پرداخت}</subPayID>
   <date>{تاریخ ریز پرداخت}</date>
   <amount>{مبلغ ریز پرداخت }</amount>
   <account>{شماره حساب}</account>
   <description>{شرح}</description>
   <status>{Inactive| Active | Done| Canceled}</ status >
   <depositName>{نام سپرده}</depositName>
   <transactionDate>{زمان انجام ریز پرداخت}</transactionDate>
   <payaRefNumber>{شماره پیگیری پایا}</payaRefNumber>
   <payaStatus>{آخرین وضیعت پایا}</payaStatusDesc>
   <payaStatusDesc>{شرح آخرین وضیعت پایا}</payaStatusDesc>
   <bankName>{نام بانک مقصد در انتقال پایا}</bankName>
 </subPayment>
</subPayments>
        .
        .
        .
</resultObj>
```

توجه : چنانچه ریز پرداختی به صورت پایا انجام شود ، چهار فیلد payaRefNumber، payaStatus، payaStatusDesc، bankName به خروجی xml اضافه می شود.

فیلد payaStatus شامل مقادیر زیر است:

- Unknown
- Confirmed
- Sent
- CentralBank_Rejected
- Sent_Recieved

- Rejected

فیلد payaStatusDesc شامل مقادیر زیر است که در واقع عنوان فارسی متناظر با payaStatus است:

- نامشخص(Unknown) : وضعیت انتقال نامشخص است.
- ارسالی - تایید شده(Confirmed) : درخواست انتقال با موفقیت ثبت شده است.
- ارسالی - ارسال شده به بانک مرکزی(Sent): درخواست انتقال به بانک مرکزی ارسال شده است.
- ارسالی - رد شده توسط بانک مرکزی(CentralBank_Rejected): درخواست انتقال ارسالی ، توسط بانک مرکزی مردود شده است.
- ارسالی - دریافت شده توسط بانک مرکزی(Sent_Recieved): درخواست انتقال توسط بانک مرکزی دریافت شده است.
- ارسالی - رد شده توسط بانک مقابل(Rejected) : بانک مقصد درخواست انتقال را به هر دلیل رد کرده است.

فیلد status می تواند شامل حالت های زیر باشد:

- Inactive: به این معناست که فاکتور ارسالی پرداخت نشده است و ریز پرداخت مورد نظر در حالت غیر فعال است. به محض پرداخت مبلغ فاکتور ریز پرداخت مورد نظر فعال می شود.

  توجه: در خرید دو مرحله ای ریز پرداخت ها پس از تایید خرید توسط فروشگاه فعال می شوند.
- Active: به این معناست که فاکتور ارسالی پرداخت شده است و ریز پرداخت مورد نظر آماده است تا در تاریخ مشخص شده انجام شود، و یا به هر دلیلی ریز پرداخت در تاریخ مورد نظر پرداخت نشده و مجددا برای پرداخت آن تلاش می شود.

توجه: تا زمانی که ریز پرداخت در حالت فعال باشد و به هر دلیلی (به عنوان مثال کافی نبودن موجودی حساب فروشگاه) با موفقیت انجام نشود، هر روز برای انجام آن تلاش خواهد شد مگر اینکه که فروشگاه آن را لغو کند.

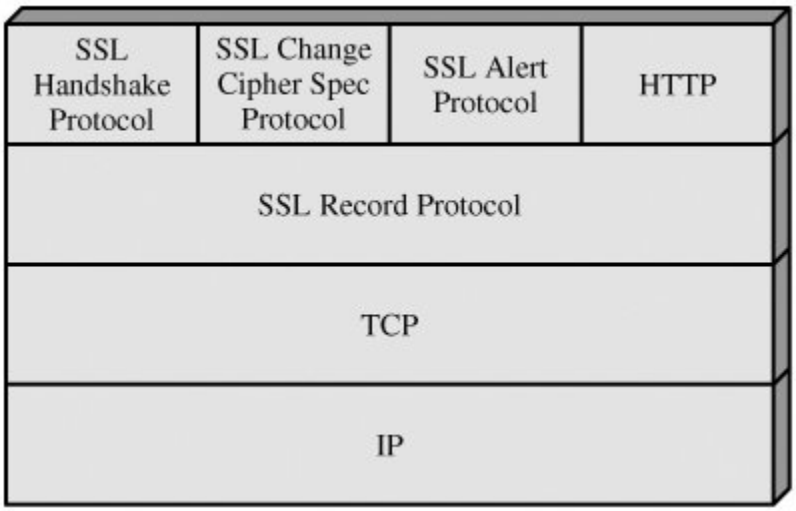
- Done: بدین معناست که ریز پرداخت با موفقیت پرداخت شده است.
- Canceled: بدین معناست که ریز پرداخت توسط فروشگاه لغو شده است.

توجه : در صورتی که فروشگاه از خرید دو مرحله ایی استفاده می کند در صورت تایید نکردن خرید، خرید برگشت خورده و ریز پرداخت های آن نیز لغو می شوند.

پیوست ۱: نیازمندی‌های امنیتی

جهت برقراری ارتباط امن فیمابین سایت فروشنده و سایت شرکت پرداخت الکترونیک پاسارگاد ، سایت شرکت پرداخت الکترونیک پاسارگاد از Protocol SSL استفاده می‌کند. پروتکل SSL (Secure Socket Layer) یک استاندارد وب برای رمزنگاری اطلاعات بین کاربر و وب سایت است. اطلاعاتی که توسط یک اتصال SSL مبادله می‌شوند به صورت رمز شده ارسال می‌شوند و بدین ترتیب اطلاعات مبادله شده از دزدیده شدن یا استراق سمع محافظت می‌شوند. پروتکل SSL برای شرکتها و مشتریان این امکان را فراهم می‌کند که به توانند با اطمینان اطلاعات خود (مانند شماره کارت اعتباری و ...) را به یک وب سایت به طور محرمانه ارسال کنند. برای برقراری یک اتصال SSL نیاز به یک SSL Certificate است.

همچنین پیشنهاد می‌شود که سایت فروشنده نیز از پروتکل SSL استفاده کند اما اجباری نیست.



یکی دیگر از نیازمندی‌های امنیتی این است که فروشنده نباید از هیچ‌کدام از اطلاعات مالی خریدار (همانند مشخصات کارت، کلمه رمز کارت و ...) مطلع شود. به همین دلیل فروشنده از خریدار هیچ نوع اطلاعات مالی و بانکی دریافت نمی‌کند و تمامی این اطلاعات توسط خریدار صرفاً در سایت شرکت پرداخت الکترونیک پاسارگاد وارد می‌شود.

پیوست ۲: نمونه کدهای مورد نیاز با زبان C# برای سمت فروشگاه

- **نمونه کد ارسال داده‌ها برای تراکنش خرید**

```
<script language="C#" runat="server">
       private setSendingData() {
         merchantCode = 115; // کد پذیرنده
         terminalCode = 12; // کد ترمینال
         amount = 2000000;  // مبلغ فاکتور
         redirectAddress = "http://merchantsite.com/redirectAddress.aspx";
           //آدرس سایتی که مشتری پس از انجام تراکنش باید به آن فرستاده شود
         timeStamp = DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss");
         invoiceNumber = 1949945; // شماره فاکتور
         invoiceDate = 1387/10/12 12:45:32; // تاریخ فاکتور
          action = "1003";
        string subPaymentsXml = "<?xml version="1.0" encoding="utf-8"?>
       <SubPaymentList><SubPayments><SubPayment><SubPayID>1</SubPayID>
       <Amount>5000000</Amount><Date>2010/02/02</Date><Account>219.10.44039.1</Account
       ><Description></Description></SubPayment><SubPayment><SubPayID>2</SubPayID><Amo
       unt>30000</Amount><Date>                                                2010/04/08
       </Date><Account>201.800.981313.1</Account><Description></Description></SubPayme
       nt></SubPayments><SubPaymentList>";
       String                                subPaymentList                                =
       Convert.ToBase64String(Encoding.UTF8.GetBytes(subPaymentsXml));
       RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
       //کلید خصوصی فروشنده
       rsa.FromXmlString("<RSAKeyValue><Modulus>oQRshGhLf2Fh...");
       string data = "#" + merchantCode + "#" + terminalCode + "#" +      invoiceNumber
       + "#" + invoiceDate + "#" + amount + "#" + redirectAddress + "#" + action + "#"
       + timeStamp + "#"+ subPaymentList + "#";
       byte[] signMain = rsa.SignData(Encoding.UTF8.GetBytes(data), new
                              SHA1CryptoServiceProvider());
           sign = Convert.ToBase64String(signMain);
       }
   </script>
```

بخشی از کد که در سایت پذیرنده قرار می‌گیرد و برای ارسال داده‌ها به سیستم پرداخت استفاده می‌شود. در واقع صفحه وبی است که پذیرنده در آن اطلاعات تراکنش را قرار می‌دهد و مشتری با زدن کلید ارسال، داده‌ها برای سایت پرداخت فرستاده می‌شود.

```
<form id="Form2" method="post" Action="https://pep.shaparak.ir/gateway.aspx">
    <input type="hidden" name="invoiceNumber" value="<%= invoiceNumber %>" />
    <input type="hidden" name="invoiceDate" value="<%= invoiceDate %>" />
    <input type="hidden" name="amount" value="<%= amount %>" />
    <input type="hidden" name="terminalCode" value="<%= terminalCode %>" />
    <input type="hidden" name="merchantCode" value="<%= merchantCode %>" />
    <input type="hidden" name="redirectAddress" value="<%= redirectAddress %>" />
    <input type="hidden" name="timeStamp" value="<%= timeStamp %>" />
    <input type="hidden" name="action" value="<%= action %>" />
    <input type="hidden" name="sign" value="<%= sign %>" />
    <input type="hidden" name="subPaymentList" value="<%= subPaymentList %>" />
    <input type="submit" name="submit" value="ارسال" />
</form>
```

- **نمونه کد دریافت نتیجه تراکنش**

```
<script language="C#" runat="server">
       private ReadPaymentResult() {
       HttpWebRequest request =
              (HttpWebRequest)WebRequest.Create("https://pep.shaparak.ir/CheckTransacti
              onResult.aspx");
        string text = "invoiceUID=" + Request.QueryString["tref"];
        byte[] textArray = Encoding.UTF8.GetBytes(text);
        request.Method = "POST";
        request.ContentType = "application/x-www-form-urlencoded";
        request.ContentLength = textArray.Length;
        request.GetRequestStream().Write(textArray, 0, textArray.Length);
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();
        StreamReader reader = new StreamReader(response.GetResponseStream());
        string result = reader.ReadToEnd();
              است// XML صورت به نتیجه تراکنش شامل Result مرحله این در
       }
</script>
```

- **نمونه کد برگشت خرید**

```
<script language="C#" runat="server">
       private DoRefund() {
        merchantCode = 115; // کد پذیرنده
        terminalCode = 12; // کد ترمینال
        amount = 2000000;  // مبلغ فاکتور
        invoiceNumber = 1949945; // شماره فاکتور
        invoiceDate = 1387/10/12 12:45:32; // تاریخ فاکتور
       action = "1004"; // 1004: براي درخواست برگشت خريد
       timeStamp = DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss");
       RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
       rsa.FromXmlString("<RSAKeyValue><Modulus>oQRshGhLf2Fh...");
       string data = "#" + merchantCode + "#" + terminalCode + "#" +
       invoiceNumber + "#" + invoiceDate + "#" + amount + "#" + action+ "#" +
       timeStamp + "#";
       byte[] signMain = rsa.SignData(Encoding.UTF8.GetBytes(data), new
                           SHA1CryptoServiceProvider());
       sign = Convert.ToBase64String(signMain);

       HttpWebRequest request =
              (HttpWebRequest)WebRequest.Create("https://pep.shaparak.ir/DoRefund.aspx"
              );
       string text = " InvoiceNumber =" + invoiceNumber +"& InvoiceDate=" +
       invoiceDate +"&MerchantCode=" + merchantCode +"&TerminalCode=" +
       terminalCode +"& Amount=" + amount +"& action=" + action +"&TimeStamp=" +
       timeStamp + "&Sign=" + sign;

        byte[] textArray = Encoding.UTF8.GetBytes(text);
        request.Method = "POST";
```

```
        request.ContentType = "application/x-www-form-urlencoded";
        request.ContentLength = textArray.Length;
        request.GetRequestStream().Write(textArray, 0, textArray.Length);
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();
        StreamReader reader = new StreamReader(response.GetResponseStream());
        string result = reader.ReadToEnd();
            است// XML صورت به نتیجه برگشت خرید شامل Result مرحله این در
    }
</script>
```

- **نمونه کد به روز رسانی ریزپرداخت**

```
<script language="C#" runat="server">
    private UpdateInvoiceSubPayment() {
    merchantCode = 115; // کد پذیرنده
     terminalCode = 12; // کد ترمینال
    timeStamp = DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss");
    string updateXml = "";
    updateXml += "<?xml version='1.0' encoding='utf-8'?>";
    updateXml += "<invoiceUpdateList>";
    updateXml += "<invoiceAction  invoiceUID=\"634253256472082172\">";
    updateXml += "<action type=\"Add\" subPayID=\"1\" amount=\"1\"
    date=\"2010/12/13\" account=\"219.10.44039.1\"/>";
    updateXml += "<action type=\"Delete\" subPayID=\"2\"  amount=\"1\"
    date=\"2010/12/13\" account=\"219.10.44039.1\"/>";
    updateXml += "<action type=\"Edit\" subPayID=\"3\" amount=\"1\"
    date=\"2010/12/13\" account=\"219.10.44039.1\"/>";
    updateXml += "</invoiceAction>";
    updateXml += "</invoiceUpdateList >";
    string inoviceUpdateList =
    Convert.ToBase64String(Encoding.UTF8.GetBytes(updateXml));
    string data = "#" + merchantCode + "#" + terminalCode + "#" +
    inoviceUpdateList + "#" + timeStamp + "#";
    RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
    rsa.FromXmlString("<RSAKeyValue><Modulus>…");
    byte[] signMain = rsa.SignData(Encoding.UTF8.GetBytes(data), new
    SHA1CryptoServiceProvider());
    string sign = Convert.ToBase64String(signMain);
    HttpWebRequest request
          =(HttpWebRequest)WebRequest.Create("https://pep.shaparak.ir/UpdateInvoi
          ceSubpayment.aspx");
    string text = "invoiceUpdateList=" + inoviceUpdateList +"&timeStamp=" +
    timeStamp +"&MerchantCode=" + merchantCode +"&TerminalCode=" + terminalCode
    + "&Sign=" + sign;
    byte[] textArray = Encoding.UTF8.GetBytes(text);
    request.Method = "POST";
    request.ContentType = "application/x-www-form-urlencoded";
    request.ContentLength = textArray.Length;
    request.GetRequestStream().Write(textArray, 0, textArray.Length);
     HttpWebResponse response = (HttpWebResponse)request.GetResponse();
     StreamReader reader = new StreamReader(response.GetResponseStream());
     string result = reader.ReadToEnd();
```

```
است// XML صورت به نتیجه به روزرساني دستورپرداخت شامل Result مرحله این در
        }
</script>
```

- **نمونه کد استعلام ریزپرداخت های ارسالی بر اساس تاریخ ریزپرداخت ها**

```
<script language="C#" runat="server">
private GetSubPaymentResult() {
merchantCode = 115; // کد پذیرنده
terminalCode = 12; // کد ترمینال
fromDate = "1392/01/01";
toDate = "1392/01/03";
timeStamp = DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss");
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
rsa.FromXmlString("<RSAKeyValue><Modulus>…");
string data = "#"+merchantCode +"#"+terminalCode +"#"+ fromDate +"#"+ toDate + "#" +
timeStamp +"#";
byte[] signMain = rsa.SignData(Encoding.UTF8.GetBytes(data), new
SHA1CryptoServiceProvider());
string sign = Convert.ToBase64String(signMain);
string text = "MerchantCode=" + merchantCode + "&TerminalCode=" + terminalCode
+"&startDate=" + fromDate + "&endDate=" + toDate +"&TimeStamp=" + timeStamp +
"&sign="+ sign;
byte[] textArray = Encoding.UTF8.GetBytes(text);
HttpWebRequest request = (HttpWebRequest)WebRequest.Create("https://pep.shaparak.ir/
GetSubPaymentsResult.aspx");
request.Method = "POST";
request.ContentType = "application/x-www-form-urlencoded";
request.ContentLength = textArray.Length;
request.GetRequestStream().Write(textArray, 0, textArray.Length);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();
StreamReader reader = new StreamReader(response.GetResponseStream());
string xml = reader.ReadToEnd();
نتیجه استعلام ریزپرداخت های ارسالی بر اساس تاریخ ریزپرداخت  شامل Result مرحله این در
صورت ها به XML //است
        }
</script>
```

پیوست ۳: الگوریتم رمز نگاری نامتقارن

الگوریتم‌های رمزگذاری نامتقارن نوعی از الگوریتم‌های رمزنگاری هستند که دارای دو کلید مختلف هستند که از یکی جهت رمزنگاری و از دیگری جهت رمز گشایی استفاده می‌شود. این الگوریتم‌ها در گستره وسیعی از کاربردها به کار می رود. در این الگوریتم‌ها کلید اول را کلید عمومی (**Public Key**) و کلید دوم را کلید خصوصی(**Private Key**) می‌نامند. یکی از کاربردهای مهم الگوریتم های رمز نگاری نامتقارن استفاده از آنها در تولید امضای دیجیتال است.

مفهوم امضای دیجیتال:

امضای دیجیتال روشی مبتنی بر الگوریتم های رمزنگاری نامتقارن است که به کمک آن می‌توان اطمینان حاصل کرد که داده‌های ارسالی از جانب شخص مشخصی ارسال شده است. نمونه ای از این الگوریتم‌ها می توان به RSA و DSA اشاره کرد.

روال کار در امضای دیجیتال به این شکل است که پیش از ارسال داده ها، اطلاعات را با استفاده از الگوریتم‌های درهم سازی یک‌طرفه (Hash Algorithms) به یک کد درهم (Hash) تبدیل می‌شود. از نمونه این الگوریتم ها می توان به MD5, SHA1 و ... اشاره کرد. یک‌طرفه بودن در این الگوریتم‌ها به این معنی است که پس از کد شدن اطلاعات به هیچ عنوان نمی توان از روی این کدها، اطلاعات اصلی را به دست آورد. پس از در هم سازی اطلاعات، به منظور تولید امضای دیجیتال، باید از یکی از الگوریتم‌های رمزنگاری نامتقارن استفاده شود و با استفاده از کلید خصوصی (Private Key) آن الگوریتم، رشته‌ی تولید شده توسط الگوریتم درهم سازی را امضا نمود.

مفهموم کلید عمومی و کلید خصوصی:

کلید عمومی بخشی از کلید است که بین همه توزیع می‌شود و هیچ نگرانی از لو رفتن و دزدیده شدن آن وجود ندارد به واقع لفظ "عمومی" نیز بیان‌گر همین مطلب است. اگر داده‌ای برای صاحب کلید عمومی (پخش کننده کلید عمومی) باید

رمز شود با استفاده از این کلید رمز نگاری شده و ارسال می‌شود. نکته مهم الگوریتم‌های نامتقارن در این مطلب است که داده‌های رمز شده با کلید عمومی فقط و فقط با کلید خصوصی قابل رمزگشایی هستند و دوباره با همان کلید عمومی نمی‌توان آنها را رمزگشایی کرد به همین دلیل داشتن کلید عمومی کمکی به رمزگشایی داده‌ها نخواهد کرد.

کلید خصوصی در واقع بخشی از کلید است که به وسیله آن داده های رمز شده به وسیله کلید عمومی را می‌توان رمز گشایی کرد. صاحب کلید خصوصی باید حداکثر محافظت از این کلید را انجام دهد و به هیچ عنوان اجازه ندهد که این کلید در دست کسی غیر از خودش قرار گیرد. علاوه بر این با استفاده از کلید خصوصی می‌توان اسناد و مدارک (Documentها)، پست الکترونیکی (Emailها) و پیام ها را امضا کرد و امضای صورت گرفته را در انتهای اسناد، پست الکترونیک و یا پیام قرار داد. در این حالت گیرنده پیام با داشتن اصل پیام، امضای دیجیتال زیر آن و کلید عمومی شما می تواند از صحت امضا اطمینان حاصل کند و مطمئن شود که داده‌ها از جانب شما ارسال شده است. اما با کلید عمومی به هیچ عنوان نمی‌تواند امضای شما را جعل کند.