بسمه تعالی

شرکت پرداخت الکترونیک

بانک پاسارگاد

مستندات فنی اتصال به درگاه پرداخت اینترنتی دو مرحله‌ای
شرکت پرداخت الکترونیک بانک پاسارگاد

PEP Two-Step-EPayment

Technical Specification

آبان 1392

مقدمه

خرید اینترنتی یکی از تراکنش‌های کارتی است که در مرکز شتاب نیز جزو تراکنش‌های مجاز محسوب می‌شود. در این مستند قدم‌های لازم برای ایجاد بستر پرداخت الکترونیکی در سمت وب سایت فروشنده که مایل است از طریق درگاه پرداخت اینترنتی دو مرحله‌ای شرکت پرداخت الکترونیک بانک پاسارگاد به خریداران خود سرویس ارائه دهد، توضیح داده شده است.

تعاریف

تعاریف مرتبط با خریدار

- **خریدار**: هویتی است که توسط یکی از انواع کارت‌های بانکی عضو شبکه شتاب و با مراجعه به وب سایت مورد نظر خود تقاضای خرید کالا یا 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 (MerchantCode)**: کدی است که توسط بانک به فروشنده اختصاص می‌یابد و در حین انجام تراکنش برای شناسایی فروشنده از آن استفاده می‌گردد.
- **شماره شناسائی ترمینال (TerminalCode)**: کدی است که توسط بانک 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 بانک را با آن کلید، امضای دیجیتال می‌کند.
- **کلید عمومی فروشنده (Public Key)** : کلیدی است که بانک جهت تایید امضای دیجیتال فروشگاه از آن استفاده می‌کند.
- **سپرده فروشنده**: سپرده کوتاه مدت، جاری یا پس‌اندازی است که فروشنده جهت انجام عمل تسویه حساب با بانک در یکی از بانک پاسارگاد افتتاح نموده و آنرا به بانک جهت تسویه حساب تراکنش‌های انجام شده، اعلام می‌نماید.
- **مبلغ فاکتور (Amount)**: مبلغی می‌باشد که فروشنده می‌خواهد از خریدار دریافت نماید.
- **شماره فاکتور (InvoiceNumber)**: هر خرید از فروشنده باید دارای شماره فاکتور خاص خود باشد، این شماره تماماً به صورت عدد است.
- **تاریخ فاکتور (InvoiceDate)**: تاریخ فاکتور خرید است و فرمت آن به انتخاب فروشگاه می‌باشد (لازم به ذکر است که تاریخ و شماره فاکتور، باید به‌گونه‌ای تخصیص داده شوند که از ترکیب آنها شناسه یکتایی بدست آید تا همیشه بتوان برای شناسایی یک تراکنش خرید از آن استفاده کرد).

- **امضای دیجیتال (Digital Signature)**: امضای دیجیتال روشی مبتنی بر الگوریتم های رمزنگاری نا متقارن می‌باشد که به کمک آن می‌توان اطمینان حاصل کرد که داده‌های ارسالی از جانب فروشگاه مشخصی ارسال شده است.
- **RedirectAddress**: آدرس صفحه‌ای در سایت فروشنده است که خریدار پس از انجام عملیات خرید به آن فرستاده می‌شود. این آدرس باید به صورت Absolute در هر تراکنش برای سایت بانک ارسال شود.
- **Timestamp**: زمان ارسال داده به سایت بانک 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 بانک ارسال شود. نکته‌ی مهم در اینجا این است که مقداری که در فیلد Timestamp قرار می‌گیرد باید دقیقا با مقداری که تحت همین عنوان در امضای دیجیتال قرار می‌گیرد یکی باشد، همچنین هیچ دو درخواستی، نمی‌توانند دارای Timestamp یکسان باشند.

تعاریف مرتبط با بانک و عملیات مالی

- **بانک** : منظور از بانک در این مستند بانک پاسارگاد می باشد.
- **درگاه پرداخت اینترنتی بانک (Internet Payment Gateway)**: سایتی است متعلق به بانک 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 بانک 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 می‌باشد. برای خرید کد 1003 و برای برگشت کد 1004 در نظر گرفته شده است.
- **شماره رهگیری (TransactionReferenceID)**: شماره‌ای است که سایت بانک 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 سپرده فروشنده توسط بانک می باشد که در صورت موفق بودن تراکنش خرید و تایید خرید انجام شده توسط فروشنده، پس از کسر کارمزد انجام می شود.

مراحل انجام تراکنش خرید دو مرحله ای

1. خریدار با مراجعه به وب سایت فروشنده و انتخاب کالا یا خدمات مورد نیاز، آماده پرداخت مبلغ فاکتور می‌شود.
2. سایت فروشنده اطلاعات مربوط به تراکنش خرید دو مرحله‌ای را با PrivateKey خود امضا کرده و با متد POST به سایت (https://pep.shaparak.ir/gateway.aspx) ارسال می‌کند. به دلیل اینکه تراکنش از نوع خرید می‌باشد به همراه ارسال داده‌ها خریدار نیز به سایت بانک فرستاده(redirect)می‌شود. مواردی که برای این تراکنش به صورت POST به وب سایت بانک ارسال می‌شوند عبارتند از:
    - MerchantCode
    - TerminalCode
    - InvoiceNumber
    - InvoiceDate
    - Amount
    - RedirectAddress
    - Action
    - TimeStamp
    - امضا دیجیتالی

مراحل تولید امضای دیجیتال عبارت است از:

1. اتصال داده‌های ذکر شده بصورت زیر:

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

2. اجرای الگوریتم درهم‌سازی SHA1 بر روی رشته بالا.
3. امضای رشته‌ی حاصل از بند دوم به وسیله PrivateKey، که نتیجه آن یک رشته‌ی باینری می‌باشد.
4. تبدیل رشته‌ی باینری به رشته ای با فرمت base64String، که این رشته امضای دیجیتال پذیرنده برای تراکنش محسوب می‌شود.

3. خریدار با وارد کردن شماره کارت (PAN)، کلمه عبور اینترنتی (PIN2)، کد اعتبارسنجی دوم (CVV2) و تاریخ انقضای کارت (Expiration Date) درخواست انجام تراکنش را برای بانک ارسال می‌کند.

4. در این مرحله تراکنش توسط بانک پردازش گردیده و عملیات لازم در مرکز بانک پاسارگاد، شتاب و بانک صادر کننده کارت انجام می‌پذیرد. در صورت صحت ورود داده‌ها و وجود وجه کافی در حساب خریدار، عملیات مالی در این مرحله توسط بانک صورت می‌گیرد.

5. وب سایت بانک پس از انجام تراکنش، نتیجه تراکنش را به خریدار نشان داده و با فشردن دکمه ادامه توسط خریدار ، او را به آدرسی که در فیلد RedirectAddress قرار داده شده می‌فرستد و در Query String آن مقادیر زیر را قرار می‌دهد.
   - InvoiceNumber (در فیلد iN)
   - InvoiceDate (در فیلد iD)
   - TransactionReferenceID (در فیلد tref)

6. دریافت نتیجه تراکنش

سایت فروشنده با ارسال TransactionReferenceID دریافت شده از جانب بانک بصورت POST، به سایت بانک (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 می‌تواند با فرستادن شماره فاکتور، تاریخ فاکتور، شماره شناسایی فروشنده و شماره شناسایی ترمینال به صورت POST به صفحه ذکر شده از نتیجه تراکنش باخبر شود. سایت بانک صفحه XML زیر را برای فروشنده ارسال می‌کند. فروشنده پس از تطبیق **نوع تراکنش**، **شماره فاکتور**، **تاریخ فاکتور**، **شماره شناسائی فروشنده** و **شماره شناسائی ترمینال** موجود در XML ارسالی، با موارد مشابه در 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 بانک، 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>
```

7. تایید خرید

در صورتی که عملیات خرید با موفقیت انجام شده باشد، سایت بانک مدت زمان مشخصی منتظر می‌ماند تا خرید انجام شده توسط فروشنده تایید شود. در صورتی که خرید طی این مدت زمان توسط فروشنده تایید نشود بانک به صورت خودکار آن را برگشت زده و پول به حساب خریدار بازمی‌گردد. لازم به ذکر است که مدت زمانی که بانک برای تایید خرید از سمت فروشنده منتظر می‌ماند طبق توافق بانک و فروشنده می‌باشد. توجه شود که برای تایید خرید، فروشنده باید شماره فاکتور و تاریخ فاکتور تراکنش خرید (تراکنش اصلی) را ارسال کند ولی TimeStamp باید با تاریخ جاری سیستم مقداردهی شود. مواردی که جهت تایید خرید به صورت post به وب سایت بانک(https://pep.shaparak.ir/VerifyPayment.aspx) ارسال می‌شوند عبارتند از:

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

مراحل تولید امضای دیجیتال عبارت است از:

1. اتصال داده‌های ذکر شده بصورت زیر:

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

2. اجرای الگوریتم درهم‌سازی SHA1 بر روی رشته بالا.
3. امضای رشته‌ی حاصل از بند دوم به وسیله PrivateKey، که نتیجه آن یک رشته‌ی باینری می‌باشد.

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

پس از ارسال موارد فوق به سایت بانک، درخواست تایید خرید 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>
```

8. انجام عمل تسویه حساب توسط بانک که جزئیات آن در قرارداد منعقده فیمابین بانک و فروشنده درج گردیده است.

برگشت خرید

در صورتی که فروشنده مایل به برگشت دادن خرید باشد می تواند حداکثر تا پایان روز ارسال تراکنش خرید این کار را انجام دهد. 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 (https://pep.shaparak.ir/doRefund.aspx) فرستاده می‌شوند عبارتند از:

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

مراحل تولید امضای دیجیتال عبارت است از:

1. اتصال داده‌های ذکر شده بصورت زیر:

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

2. اجرای الگوریتم درهم‌سازی SHA1 بر روی رشته بالا.
3. امضای رشته‌ی حاصل از بند دوم به وسیله PrivateKey، که نتیجه آن یک رشته‌ی باینری می‌باشد.
4. تبدیل رشته‌ی باینری به رشته ای با فرمت base64String، که این رشته امضای دیجیتال پذیرنده برای برگشت زدن تراکنش خرید می‌باشد.

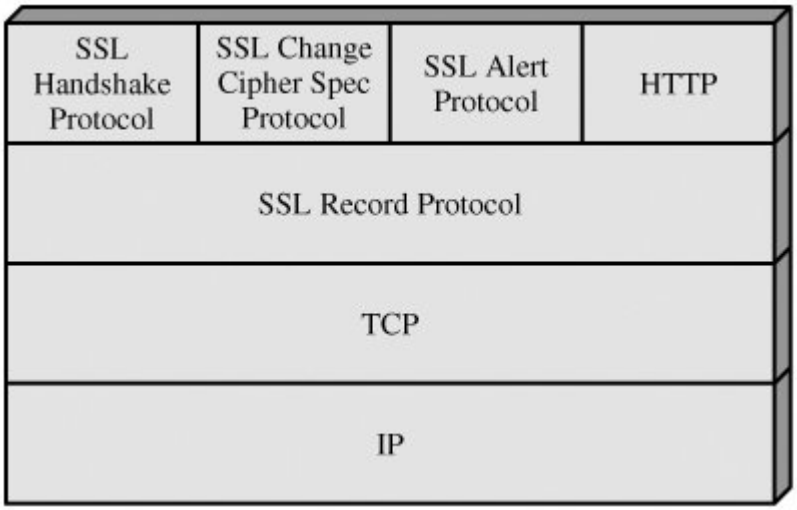
پس از ارسال موارد فوق به سایت بانک، درخواست برگشت خرید توسط بانک پردازش گردیده و xml زیر برای فروشنده ارسال می‌شود.

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

پیوست 1: نیازمندی‌های امنیتی

جهت برقراری ارتباط امن فیمابین سایت فروشنده و سایت بانک، سایت بانک از پروتکل 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 بصورت رمز شده ارسال می‌شوند و بدین ترتیب اطلاعات مبادله شده از دزدیده شدن یا استراق سمع محافظت می‌شوند. SSL برای شرکت‌ها و مشتریان این امکان را فراهم می‌کند که بتوانند با اطمینان اطلاعات خود (مانند شماره کارت اعتباری و ...) را به یک وب سایت بطور محرمانه ارسال کنند. برای برقراری یک اتصال SSL نیاز به یک SSL Certificate می‌باشد.

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 بانک وارد می‌شود.

پیوست 2: 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"; // 1003: برای درخواست خرید
        RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
        rsa.FromXmlString("<RSAKeyValue><Modulus>oQRshGhLf2Fh...");
        //کلید خصوصی فروشنده
        string data = "#" + merchantCode + "#" + terminalCode + "#"
   +invoiceNumber + "#" + invoiceDate + "#" + amount + "#" +redirectAddress
   + "#" + action + "#" + timeStamp + "#";
        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 با زدن کلید ارسال از سوی مشتری ، 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="submit" name="submit" value="ارسال" />
</form>
```

- **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: برای درخواست برگشت خرید
       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/DoRefun
              d.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();
                      در این مرحله Result شامل نتیجه برگشت خرید به صورت XML میباشد//
       }
</script>
```

- **نمونه کد دریافت نتیجه**

```
<script language="C#" runat="server">

        private ReadPaymentResult() {
        HttpWebRequest request =
               (HttpWebRequest)WebRequest.Create("https://pep.shaparak.ir
               /CheckTransactionResult.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;
         ServicePointManager.ServerCertificateValidationCallback =new
               RemoteCertificateValidationCallback(RemoteCertificateValidation);
                                        //برای اطمینان حاصل کردن از اینکه مشتری داده های خود را به سایت بانک ارسال
                                        کرده و response دریافت شده فقط و فقط از جانب سایت بانک می باشد.
          request.GetRequestStream().Write(textArray, 0, textArray.Length);
          HttpWebResponse response = (HttpWebResponse)request.GetResponse();
          StreamReader reader = new StreamReader(response.GetResponseStream());
          string result = reader.ReadToEnd();
                                  //در این مرحله Result شامل نتیجه به صورت XML میباشد
        }
```

تابع RemoteCertificateValidation که در زیر آمده است مقدار True یا False را در نتیجه ی چک کردن SSl Certificate برمی‌گرداند. اگر خطایی در ارتباط با certificate وجود نداشته باشد True، در غیر اینصورت False برگردانده می شود.

```
        private static bool RemoteCertificateValidation(object
        sender,X509Certificate certificate, X509Chain chain, SslPolicyErrors
        sslPolicyErrors)
        {
                if (sslPolicyErrors == SslPolicyErrors.None)
                    return true;
                return false;
        }

</script>
```

- **نمونه کد تایید خرید**

```
<script language="C#" runat="server">
       private VerifyPayment() {
       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
       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 + "#" + timeStamp +"#";
       byte[] signMain = rsa.SignData(Encoding.UTF8.GetBytes(data), new
                              SHA1CryptoServiceProvider());
       sign = Convert.ToBase64String(signMain);

       HttpWebRequest request =
              (HttpWebRequest)WebRequest.Create("https://pep.shaparak.ir/
              VerifyPayment.aspx");
       string text = " InvoiceNumber =" + invoiceNumber +"& InvoiceDate=" +
       invoiceDate +"&MerchantCode=" + merchantCode +"&TerminalCode=" +
       terminalCode +"& Amount=" + amount +"& 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();
                       //در این مرحله Result شامل نتیجه تایید خرید به صورت XML میباشد
       }
</script>
```

پیوست 3:الگوریتم رمز نگاری نامتقارن

الگوریتم‌های رمز گذاری نامتقارن نوعی از الگوریتم‌های رمز نگاری هستند که دارای دو کلید مختلف می‌باشند 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 می‌باشد.

مفهوم امضای دیجیتال :

امضای دیجیتال روشی مبتنی بر الگوریتم های رمزنگاری نا متقارن می باشد که به کمک ان 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 رمز نگاری 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 توضیع 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 رمز گشایی 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 مانند Documentها، Emailها و پیغامها را امضا کرد و امضای صورت گرفته را در انتهای Document، Email و یا پیغام قرار داد. در این حالت گیرنده پیغام با داشتن اصل پیغام، 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.