Electronic payment

E-commerce sites use electronic payment, where electronic payment refers to paperless monetary transactions. Electronic payment has revolutionized the business processing by reducing the paperwork, transaction costs, and labor cost. Being user friendly and less time-consuming than manual processing, it helps business organization to expand its market reach/expansion. Listed below are some of the modes of electronic payments −

* Credit Card
* Debit Card
* Smart Card
* E-Money
* Electronic Fund Transfer (EFT)

Credit Card

Payment using credit card is one of most common mode of electronic payment. Credit card is small plastic card with a unique number attached with an account. It has also a magnetic strip embedded in it which is used to read credit card via card readers. When a customer purchases a product via credit card, credit card issuer bank pays on behalf of the customer and customer has a certain time period after which he/she can pay the credit card bill. It is usually credit card monthly payment cycle. Following are the actors in the credit card system.

* **The card holder** − Customer
* **The merchant** − seller of product who can accept credit card payments.
* **The card issuer bank** − card holder's bank
* **The acquirer bank** − the merchant's bank
* **The card brand** − for example , visa or Mastercard.

Credit Card Payment Proces

|  |  |
| --- | --- |
| **Step** | **Description** |
| Step 1 | Bank issues and activates a credit card to the customer on his/her request. |
| Step 2 | The customer presents the credit card information to the merchant site or to the merchant from whom he/she wants to purchase a product/service. |
| Step 3 | Merchant validates the customer's identity by asking for approval from the card brand company. |
| Step 4 | Card brand company authenticates the credit card and pays the transaction by credit. Merchant keeps the sales slip. |
| Step 5 | Merchant submits the sales slip to acquirer banks and gets the service charges paid to him/her. |
| Step 6 | Acquirer bank requests the card brand company to clear the credit amount and gets the payment. |
| Step 7 | Now the card brand company asks to clear the amount from the issuer bank and the amount gets transferred to the card brand company. |

Debit Card

Debit card, like credit card, is a small plastic card with a unique number mapped with the bank account number. It is required to have a bank account before getting a debit card from the bank. The major difference between a debit card and a credit card is that in case of payment through debit card, the amount gets deducted from the card's bank account immediately and there should be sufficient balance in the bank account for the transaction to get completed; whereas in case of a credit card transaction, there is no such compulsion.

Debit cards free the customer to carry cash and cheques. Even merchants accept a debit card readily. Having a restriction on the amount that can be withdrawn in a day using a debit card helps the customer to keep a check on his/her spending.

Smart Card

Smart card is again similar to a credit card or a debit card in appearance, but it has a small microprocessor chip embedded in it. It has the capacity to store a customer’s work-related and/or personal information. Smart cards are also used to store money and the amount gets deducted after every transaction.

Smart cards can only be accessed using a PIN that every customer is assigned with. Smart cards are secure, as they store information in encrypted format and are less expensive/provides faster processing. Mondex and Visa Cash cards are examples of smart cards.

E-Money

E-Money transactions refer to situation where payment is done over the network and the amount gets transferred from one financial body to another financial body without any involvement of a middleman. E-money transactions are faster, convenient, and saves a lot of time.

Online payments done via credit cards, debit cards, or smart cards are examples of emoney transactions. Another popular example is e-cash. In case of e-cash, both customer and merchant have to sign up with the bank or company issuing e-cash.

Electronic Fund Transfer

It is a very popular electronic payment method to transfer money from one bank account to another bank account. Accounts can be in the same bank or different banks. Fund transfer can be done using ATM (Automated Teller Machine) or using a computer.

Nowadays, internet-based EFT is getting popular. In this case, a customer uses the website provided by the bank, logs in to the bank's website and registers another bank account. He/she then places a request to transfer certain amount to that account. Customer's bank transfers the amount to other account if it is in the same bank, otherwise the transfer request is forwarded to an ACH (Automated Clearing House) to transfer the amount to other account and the amount is deducted from the customer's account. Once the amount is transferred to other account, the customer is notified of the fund transfer by the bank.

Security is an essential part of any transaction that takes place over the internet. Customers will lose his/her faith in e-business if its security is compromised. Following are the essential requirements for safe e-payments/transactions −

* **Confidentiality** − Information should not be accessible to an unauthorized person. It should not be intercepted during the transmission.
* **Integrity** − Information should not be altered during its transmission over the network.
* **Availability** − Information should be available wherever and whenever required within a time limit specified.
* **Authenticity** − There should be a mechanism to authenticate a user before giving him/her an access to the required information.
* **Non-Repudiability** − It is the protection against the denial of order or denial of payment. Once a sender sends a message, the sender should not be able to deny sending the message. Similarly, the recipient of message should not be able to deny the receipt.
* **Encryption** − Information should be encrypted and decrypted only by an authorized user.
* **Auditability** − Data should be recorded in such a way that it can be audited for integrity requirements.

## Measures to ensure Security

Major security measures are following −

* **Encryption** − It is a very effective and practical way to safeguard the data being transmitted over the network. Sender of the information encrypts the data using a secret code and only the specified receiver can decrypt the data using the same or a different secret code.
* **Digital Signature** − Digital signature ensures the authenticity of the information. A digital signature is an e-signature authenticated through encryption and password.
* **Security Certificates** − Security certificate is a unique digital id used to verify the identity of an individual website or user.

## Security Protocols in Internet

We will discuss here some of the popular protocols used over the internet to ensure secured online transactions.

## Secure Socket Layer (SSL)

It is the most commonly used protocol and is widely used across the industry. It meets following security requirements −

* Authentication
* Encryption
* Integrity
* Non-reputability

"https://" is to be used for HTTP urls with SSL, where as "http:/" is to be used for HTTP urls without SSL.

### **Secure Hypertext Transfer Protocol (SHTTP)**

SHTTP extends the HTTP internet protocol with public key encryption, authentication, and digital signature over the internet. Secure HTTP supports multiple security mechanism, providing security to the end-users. SHTTP works by negotiating encryption scheme types used between the client and the server.

### **Secure Electronic Transaction**

It is a secure protocol developed by MasterCard and Visa in collaboration. Theoretically, it is the best security protocol. It has the following components −

* **Card Holder's Digital Wallet Software** − Digital Wallet allows the card holder to make secure purchases online via point and click interface.
* **Merchant Software** − This software helps merchants to communicate with potential customers and financial institutions in a secure manner.
* **Payment Gateway Server Software** − Payment gateway provides automatic and standard payment process. It supports the process for merchant's certificate request.
* **Certificate Authority Software** − This software is used by financial institutions to issue digital certificates to card holders and merchants, and to enable them to register their account agreements for secure electronic commerce.

# **Using Digital Signatures to Secure E-Commerce**

Digital signatures—which have been de rigueur for years among both professional and amateur security advocates—have emerged from those relatively small populations to become an important part of mainstream security discussions.

The driving force was the Electronic Signatures in Global and National Commerce Act (the so-called “E-Sign” bill) that, beginning in October of 2000, allowed an electronic signature to be accepted as binding in a commercial transaction.

E-Sign allows the parties in an agreement to accept an E-Signature; but, contrary to popular belief, it does not require that digital signatures be accepted.

Well defer a real discussion of this legislation; but [check out this GigaLaw article](http://www.gigalaw.com/articles/alston-2000-06-p1.html) for a legally focused explanation of the bills provisions.

Well also defer a spelunk into the nitty-gritty math of encryption and decryption, largely because RSA has a series of papers (PCKS 1-10) [at its Web site](http://www.rsa.com/) that covers the math involved quite adequately. PCKS Nos. 1, 5, and 7 are a great a place to start.

Here well talk about the concepts delineated in those papers, as well as how to apply them in the non-mathematical world.

**Definitions**

One simple word definition: a “digital signature” is extra data appended to a message which identifies and authenticates the identity of the sender and the message data using [public-key encryption.](http://en.wikipedia.org/wiki/Public-key_encryption)

The sender uses a [one-way hash function](http://en.wikipedia.org/wiki/One-way_hash_function) to generate a hash-code from the message data.

He/she then encrypts the hash-code with his/her private key. The receiver recomputes the hash-code from the data and decrypts the received hash with the senders public key.

If the two hash-codes are equal, the receiver is given an indication that the data has not been corrupted while in transit from one machine to another, and that it appears on first glance to have come from the given sender.

**Now for Some Detail**

Now, lets look at this definition with some intermediate steps left in, instead of glossing them over:

A message is a collection of some data into a unit. This “message” has a finite, but variable, length. What is called a message hash results from the reduction (by a mathematical operation) of this variable length message into a fixed length result. This result is called, understandably, the hash.

For most routine computational situations, the fixed length used for a hash is 128 bits; though some transaction systems will cut this down to 32 bits for speed. One property of a hash function is that it gives differing results for each different input. This means that hashes produce a unique resultant value given a unique message input.

The Secure Hash Algorithm (SHA-1, described in [FIPS 181-1](https://www.eweek.com/security/using-digital-signatures-to-secure-e-commerce/csrc.nist.gov/cryptval/140-1/140sp/140sp197.pdf)) is one commonly used algorithm for a one-way hash. In general, commercial products (like MD2 or MD5 from RSA Data Security Inc.) are used by most programmers to compute the hash on the fly (stream processing) inside a program.

MD5 is designed specifically to be fast when used in real-time operations, and has been widely accepted. Not having to reinvent the wheel and then optimize it as well is a strong attraction when a project needs to be finished on time.

Now, (and here comes the payoff) if one then encrypts the resulting hash of a message with ones private key , you get the digital signature of a message. This signature is usually appended to the message itself.

The digital signature is the result of binding someones private key via encryption to a specific message digest hash, thereby binding the person (by extension) to that messages specific content.

In the RSA-style method, to verify that a person has signed message with their private key; one decrypts the encoded message digest (which has been bound to the message) using the public key of the assumed sender to get the message hash.

If that hash is the same as the one that the receiver then performs on the message that accompanied the encrypted digest, the receiver then makes the assumption that (1) the sender (or someone who has the senders private key) encrypted the digest with their private key and (2) the message did not change in transmission.

**Some Caveats**

Its important to remember that the presence of a digital signature does not imply anything about the state or authenticity of the message to which it is linked. A signature links a sender to a message; it doesnt authenticate the message itself.

The message telling the Air Force to start bombing Moscow Right Now may have a correct signature; but it will not be acted upon (hopefully—see the movie “Dr. Strangelove ” for more on this) until it has been authenticated by some other independent means.

The message may be left in plaintext, and thus any receiver of the message can verify that it was indeed sent by the purported sender. If the message is encrypted in some way (say with a secret key) only those possessing the key can decode the message, perform a hash operation on the plaintext to get a digest, compare the digest to the encrypted digest decoded using the senders public key and thus determine the validity of the digital signature.

For now the important thing is to remember that digital signatures can work to bind someones private key (again, by being able to decrypt with the public key we consider the message to have been encrypted by the private key) to a message.

Lets take signatures one step further in degree: That message can be “countersigned” by someone else (like a receiver-trusted Certificate Authority) to form what is called a certificate. Certificates are defined structurally and technically in RFC1422. This implies compliance with the X.509 v3 standard as well.

**Dual Signatures**

Dual signatures can link two messages within a message unit. These segments may be effectively addressed to differing people such that the message parts may only be read by the intended recipient, yet provide a way to check authenticity of the overall message.

Lets consider how to actually do this. A dual signature is generated by creating a message digest (hash) of both plaintext messages individually (H(m1) and H(m2)), concatenating these digests into one value [m3] and then taking the hash of that value [H(m3)]. Encrypting H(m3) with the signers private key provides the dual signature.

A recipient operates upon the message by first computing the message digest of the message sent, then concatenating it with the sender-provided message digest of the other message, and finally computing the result. This should equal the decrypted dual signature. Thus, dual signatures link two individual messages into one functional unit for transmission, yet will keep the information content compartmentalized.

**Enveloping**

Now, what if one part of the message needs to be more “secure” than the other? The simple way out of this situation is to apply another layer of encryption on top of the “message plus signature” data. A so-called “digital envelope “is a way to encrypt data and to send the key for that encryption along with the data. Most enveloping schemes use a symmetric method to encrypt the data, and an asymmetric one to encrypt the key.

**Randomness and Signatures**

One aspect about the mathematics of key encoding is illuminating as to how real life can intrude on the best ordered calculations. The encoding scheme needs to start its mathematical operations from some initialization vector, or random sequence of numbers, to provide the best possible security.

Pseudo-random number generator (pRNG) programs can provide these initializing values, or random human actions like the time between hitting keys can be used to provide them. pRNGs have a topic of some controversy amongst the sci.crypt newsgroup, where some serious cipher-lovers hang out. The discussion consensus evolved that pRNGs work, but only up to a point.

The problem is that pRNGs may also require an initial seed value to generate randomized numbers. Providing these seed numbers can be a problem for the designer. Netscape suffered some rather public embarrassment when it was discovered by an observer that the time of day was being used as a seed value in Navigator 2.0. Knowing the seed value allowed an easier computational attack on the encrypted messages that Navigator was generating.

For very secure (read: military) situations, the sci.crypt folk like other types of RNGs such as physical radiation-decay emission counters. That sort of hardware is obviously impractical in the normal user universe, but it behooves the designer using encryption to consider carefully where his or her random seeds come from. Eastlake, Crocker and Shillers RFC 1750 *Randomness Recommendations for Security* should be consulted for further details on randomization of seeds. Chaos, in this instance, can work for you.

[Next page: So What Good is It?](https://www.eweek.com/security/using-digital-signatures-to-secure-e-commerce)

So What Good is

It?”>

**Whats the use of all this?**

So, what good are these “signatures”? Where do they help make things work? E-mail systems come to mind, of course. There are lots of commercial applications that use these concepts to either control or filter email.

It may well be that the real “killer app” for signatures is e-commerce. Microsoft was reported to have gotten an “electronic agent” clause inserted into the E-Sign bill that allows such an agent (i.e., a computer program ) to enter into a legally binding contract through the use of digital signatures.

This clause would enable a mouse click (“Buy me!”) sent over the web to legally bind a user in a contract. One can only surmise what unscrupulous sites will do in the future with this concept. A button that is viewed as “Free Stuff” but instead charges you seems a no-brainer to implement. Being able to “bet the house” on a casino site may now mean your real house!

**The Validity of Signatures in Authentication**

The best use of signatures overall may be that they can serve as a negative indicator. That is, if you expect to find a digital signature with a message and find none (or one that computes out to an unexpected value) it forces you to authenticate the message by some other means or reject it entirely. It can be a flag that focuses attention on a message when that message is an exception to what is expected. The biggest conceptual problem with digital signatures is that a positive result (that is, everything seems OK) may not be a validator of message accuracy.

Lets setup a classic scenario:

By initiating a “man-in-the-middle” attack, an evil-doer would intercept the message from its original sender; usually by having the sender route the message to an address that appears to be for the intended recipient, but actually is used by the interceptor.

The interceptor can then alter the message using public keys. How? Since it was sent to the interceptor, its most likely also been encoded with the interceptors public key. If thats the case, it means the interceptors private key will decode the message. (The sender thinks he has used the recipients public key.)

If that part of the attack doesnt come off and the correct public key has been used, the interceptor cant alter the message, but may still be able to gain information about the sender and recipient through message analysis.

He can also replace the entirely message with his own. It can then be sent it on its way to the intended recipient. The recipient decodes what seems to be a good signature by use of his private key. But, at worst, the attack altered the contents of the message during transit. Even if the contents are not altered, it should fail authentication since it did not come directly from the sender.

In short, a digital signature may seem fine from a mathematical standpoint but actually signify nothing for authentication purposes. To rely on a signature as a positive indicator (as the E-Sign bill seems to do) is asking for trouble.

Well look at some of the security issues raised by this bill in a later article. At least, we all should understand what a digital signature is—and what it is not.