Outline

- Principle of digital signatures
- RSA digital signature scheme
- ElGamal digital signature scheme
- One Standard: **Digital Signature Algorithm (DSA)**

Introduction

- A person is used to sign a document to confirm that the document is originated/approved by him.
- How will Bob ensure that the message comes from Alice, not from Eve when Alice sends a message to Bob?
- In practice, Bob can ask Alice to sign the message cryptographically.
- In other words, an digital signature can authenticate Alice as a valid sender of the message.

Contd…

- Suppose you want to sign a document.
- Why can't you simply digitize your signature and append it to the document?
- Anyone who has access to it can simply remove the signature and add it to something else, for example, a check for a large amount of money.
- With normal signatures, this would require cutting the signature off the document, or photocopying it, and posting it on the check.
- This would rarely pass for an acceptable signature. However, such a forgery is quite easy and cannot be distinguished from the original.
- Therefore, we require digital signatures that cannot be separated from the message and attached to another.

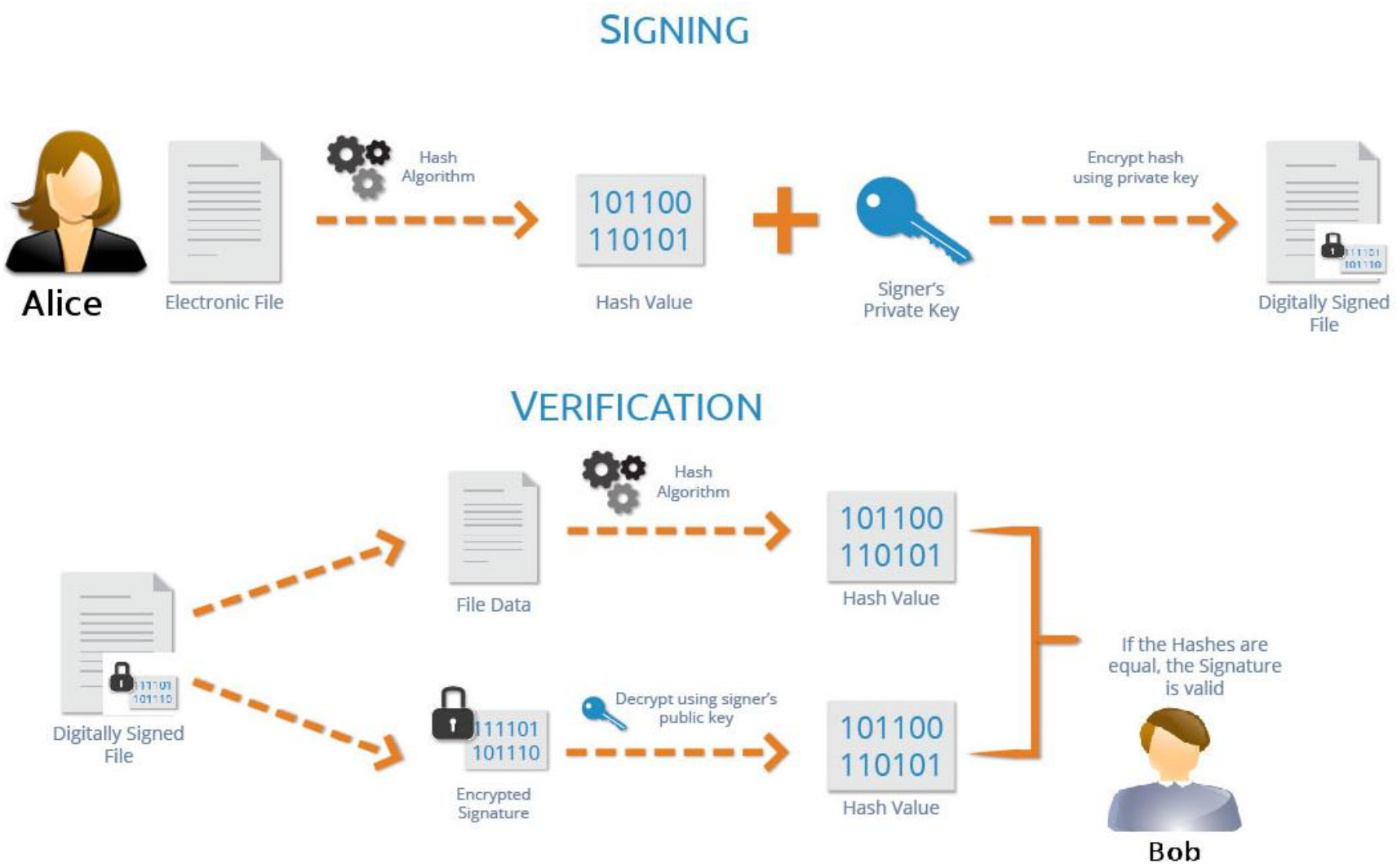
Contd…

- That is, the signature is not only tied to the signer but also to the message that is being signed.
- Also, the digital signature needs to be easily verified by other parties.
- Digital signature schemes therefore consist of two distinct steps: the **signing process**, and the **verification process**.

Basic Requirements

- Easy for the signer to sign
- Easy for anyone to verify a message
- Hard for anyone to forge a digital signature

Working Procedure



Comparison

Digital Signature	Public Key Encryption
Only the holder of a secret can digitally sign data	**Anyone** can encrypt data
Anyone can verify that a digital signature is valid	**Only the holder** of a secret can decrypt the encrypted data

Security Services

- Digital signatures provide the following security services:
  - **Integrity:** Messages have not been modified in transit.
  - **Message Authentication:** The sender of a message is authentic.
  - **Nonrepudiation:** The sender of a message can not deny the creation of the message.

RSA Signature Scheme

Alice

$k_{pr} = d, k_{pub} = (n, e)$

(n,e) →

compute signature:
$s = \text{sig}_{k_{pr}}(x) \equiv x^d \bmod n$

(x,s) →

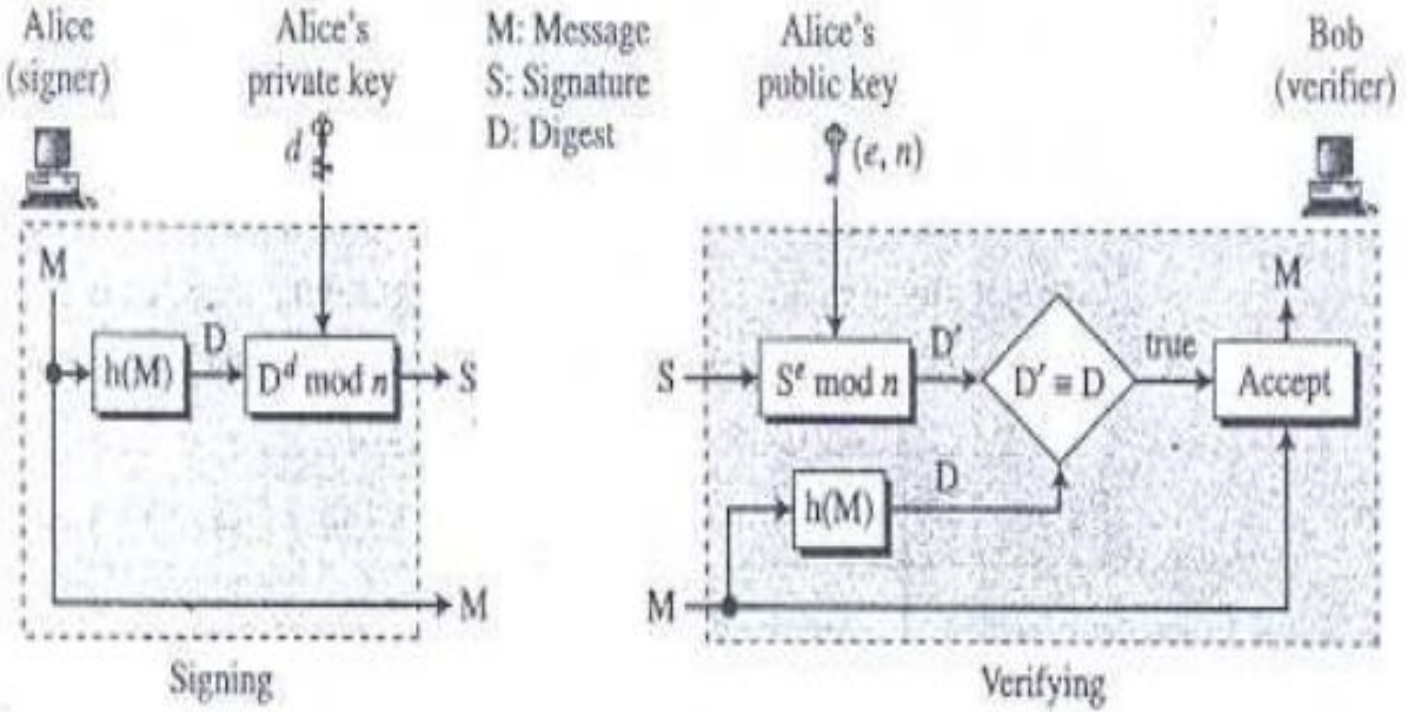
Bob

verify: $\text{ver}_{k_{pub}}(x, s)$
$x' \equiv s^e \bmod n$

$$x' \begin{cases} \equiv x \bmod n & \Longrightarrow \text{valid signature} \\ \not\equiv x \bmod n & \Longrightarrow \text{invalid signature} \end{cases}$$

RSA Signature

Figure 13.8 *The RSA signature on the message digest*



Attacks on RSA Signature

Attacks on RSA Signed Digests

How susceptible to attack is the RSA digital signature scheme when the digest is signed?

Key-Only Attack We can have three cases of this attack:

a. Eve intercepts the pair (S, M) and tries to find another message M′ that creates the same digest, h(M) = h(M′). As we learned in Chapter 11, if the hash algorithm is *second preimage resistant*, this attack is very difficult.

b. Eve finds two messages M and M′ such that h(M) = h(M′). She lures Alice to sign h(M) to find S. Now Eve has a pair (M′, S) which passes the verifying test, but it is the forgery. We learned in Chapter 11 that if the hash algorithm is *collision resistant*, this attack is very difficult.

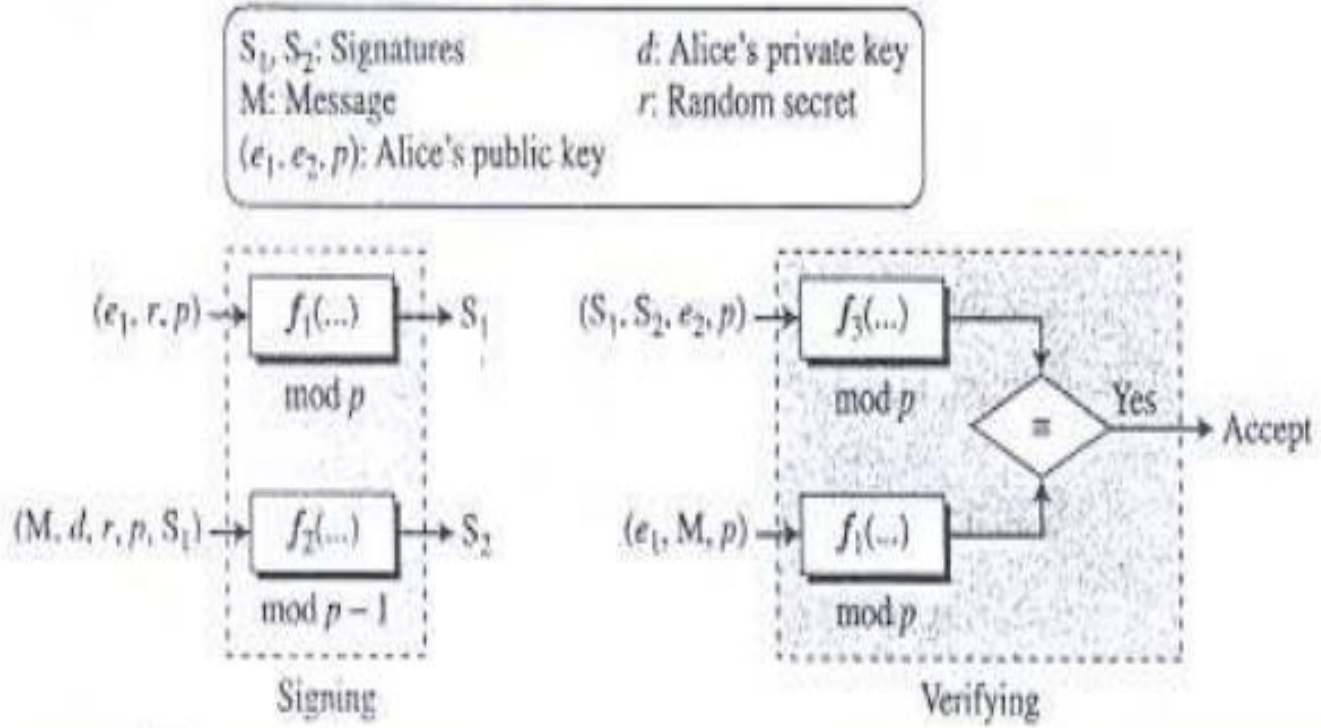
c. Eve may randomly find message digest D, which may match with a random signature S. She then finds a message M such that D = h(M). As we learned in Chapter 11, if the hash function is *preimage resistant*, this attack is very difficult to launch.

Known-Message Attack Let us assume Eve has two message-signature pairs (M_1, S_1) and (M_2, S_2) which have been created using the same private key. Eve calculates $S \equiv S_1 \times S_2$. If she can find a message M such that $h(M) \equiv h(M_1) \times h(M_2)$, she has forged a new message. However, finding M given h(M) is very difficult if the hash algorithm is *preimage resistant*.

Chosen-Message Attack Eve can ask Alice to sign two legitimate messages M_1 and M_2 for her. Eve then creates a new signature $S \equiv S_1 \times S_2$. Since Eve can calculate $h(M) \equiv h(M_1) \times h(M_2)$, if she can find a message M given h(M), the new message is a forgery. However, finding M given h(M) is very difficult if the hash algorithm is *preimage resistant*.

ElGamal Digital Signature Scheme (Idea)

Figure 13.9 *General idea behind the ElGamal digital signature scheme*



ElGamal Signature Scheme

- Alice's public key: (p, α, β) where $\beta \equiv \alpha^a \bmod p$ and private key: a
- In order for Alice to sign a message m, she does the following:

1. Selects a secret random k such that $\gcd(k, p-1) = 1$
2. Computes $r \equiv \alpha^k \pmod{p}$
3. Computes $s \equiv k^{-1}(m - ar) \pmod{p-1}$

The signed message is the triple (m, r, s).

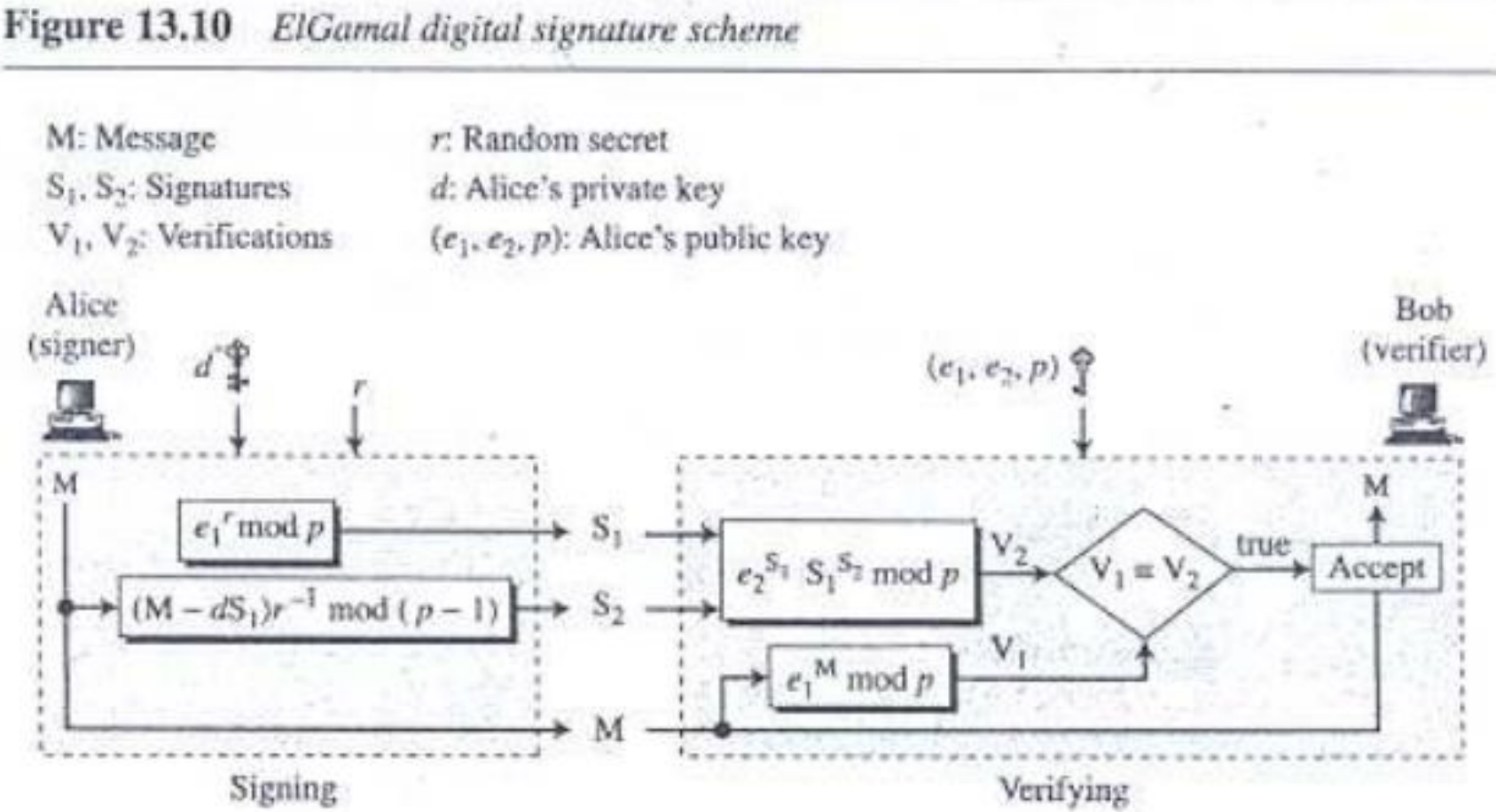
Bob can verify the signature as follows:

1. Download Alice's public key (p, α, β).
2. Compute $v_1 \equiv \beta^r r^s \pmod{p}$, and $v_2 \equiv \alpha^m \pmod{p}$.
3. The signature is declared valid if and only if $v_1 \equiv v_2 \pmod{p}$.

We now show that the verification procedure works. Assume the signature is valid. Since $s \equiv k^{-1}(m - ar) \pmod{p-1}$, we have $sk \equiv m - ar \pmod{p-1}$, so $m \equiv sk + ar \pmod{p-1}$. Therefore (recall that a congruence mod $p-1$ in the exponent yields an overall congruence mod p),

$$v_2 \equiv \alpha^m \equiv \alpha^{sk+ar} \equiv (\alpha^a)^r (\alpha^k)^s \equiv \beta^r r^s \equiv v_1 \pmod{p}.$$

Figure 13.10 *ElGamal digital signature scheme*



ElGamal Signature Example

Here is a trivial example. Alice chooses $p = 3119$, $e_1 = 2$, $d = 127$ and calculates $e_2 = 2^{127} \bmod 3119 = 1702$. She also chooses r to be 307. She announces e_1, e_2, and p publicly; she keeps d secret. The following shows how Alice can sign a message.

$$M = 320$$

$$S_1 = e_1^{\,r} = 2^{307} = 2083 \bmod 3119$$

$$S_2 = (M - d \times S_1) \times r^{-1} = (320 - 127 \times 2083) \times 307^{-1} = 2105 \bmod 3118$$

Alice sends M, S_1, and S_2 to Bob. Bob uses the public key to calculate V_1 and V_2.

$$V_1 = e_1^{\,M} = 2^{320} = 3006 \bmod 3119$$

$$V_2 = d^{S_1} \times S_1^{\,S_2} = 1702^{2083} \times 2083^{2105} = 3006 \bmod 3119$$

Because V_1 and V_2 are congruent, Bob accepts the message and he assumes that the message has been signed by Alice because no one else has Alice's private key, d.

Credit Cards on the Internet

Problem: communicate credit card and purchasing data securely to gain consumer trust

- Authentication of buyer and merchant
- Confidential transmissions

Systems vary by

- Type of public-key encryption
- Type of symmetric encryption
- Message digest algorithm
- Number of parties having private keys
- Number of parties having certificates

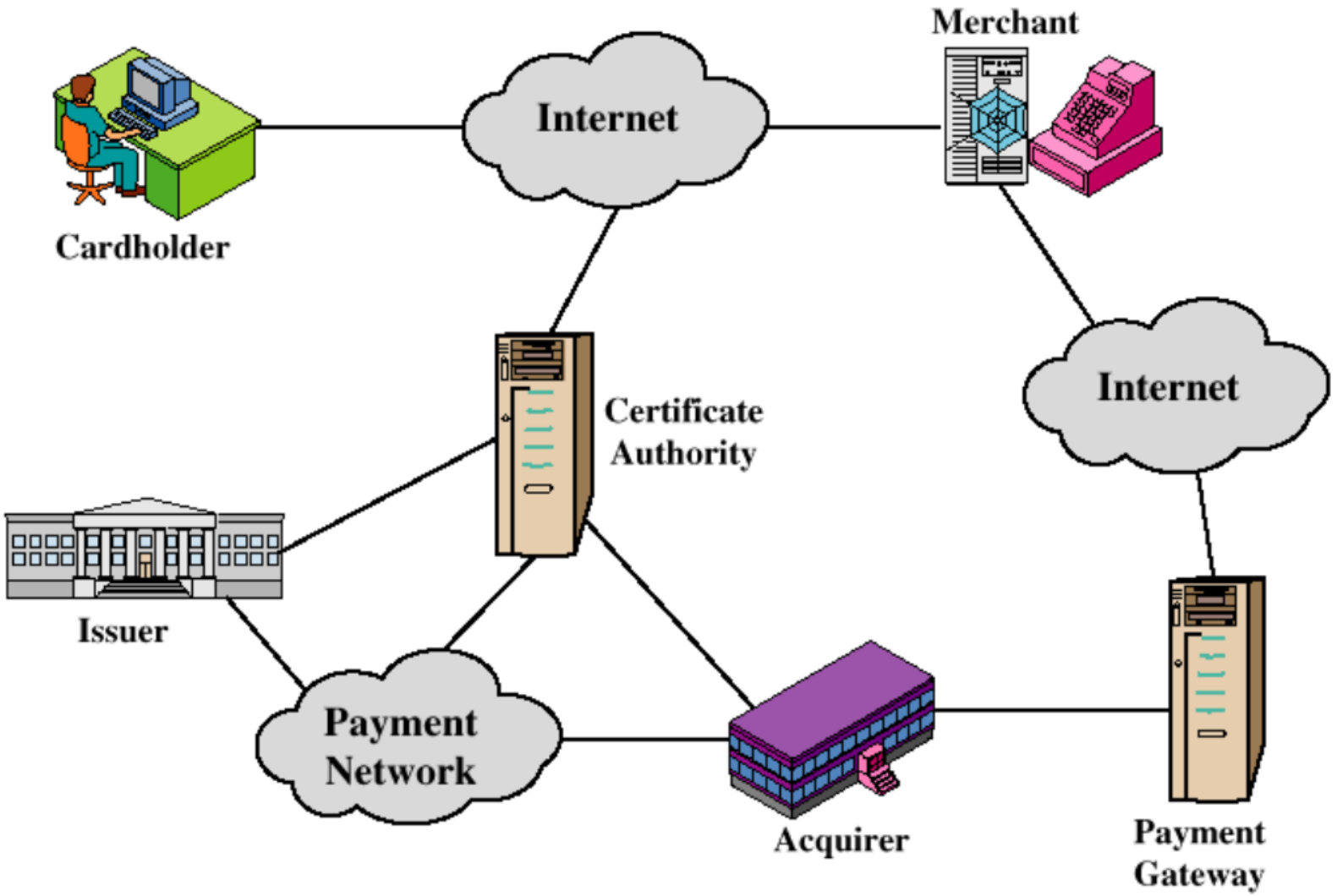
Credit Card Protocols

- SSL 1 or 2 parties have private keys
- TLS (Transport Layer Security)
  IETF version of SSL
- i KP (IBM)
- SEPP (Secure Encryption Payment Protocol)
  MasterCard, IBM, Netscape
- STT (Secure Transaction Technology)
  VISA, Microsoft
- SET (Secure Electronic Transactions)
  MasterCard, VISA

Secure Electronic Transaction (SET)

- Developed by Visa and MasterCard
- Designed to protect credit card transactions
- Confidentiality: all messages encrypted
- Trust: all parties must have digital certificates
- Privacy: information made available only when and where necessary

Participants in the SET System



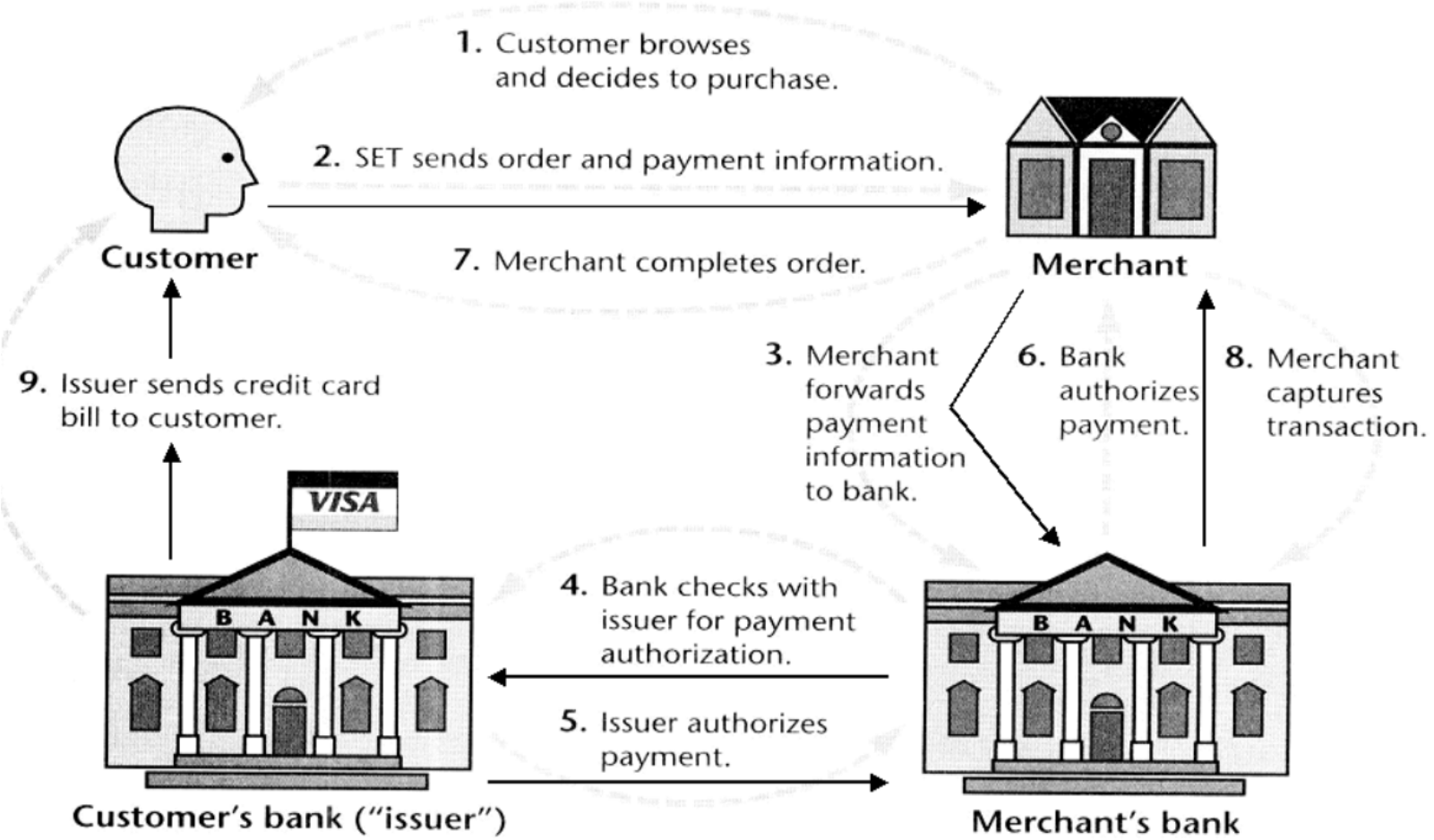
SET Business Requirements

- Provide confidentiality of payment and ordering information
- Ensure the integrity of all transmitted data
- Provide authentication that a cardholder is a legitimate user of a credit card account
- Provide authentication that a merchant can accept credit card transactions through its relationship with a financial institution

SET Business Requirements cont.

- Ensure the use of the best security practices and system design techniques to protect all legitimate parties in an electronic commerce transaction
- Create a protocol that neither depends on transport security mechanisms nor prevents their use
- Facilitate and encourage interoperability among software and network providers

SET Transactions



SET Transactions

- The customer opens an account with a card issuer.
  MasterCard, Visa, etc.
- The customer receives a X.509 V3 certificate signed by a bank.
  X.509 V3
- A merchant who accepts a certain brand of card must possess two X.509 V3 certificates.
  One for signing & one for key exchange
- The customer places an order for a product or service with a merchant.
- The merchant sends a copy of its certificate for verification

SET Transactions cont.

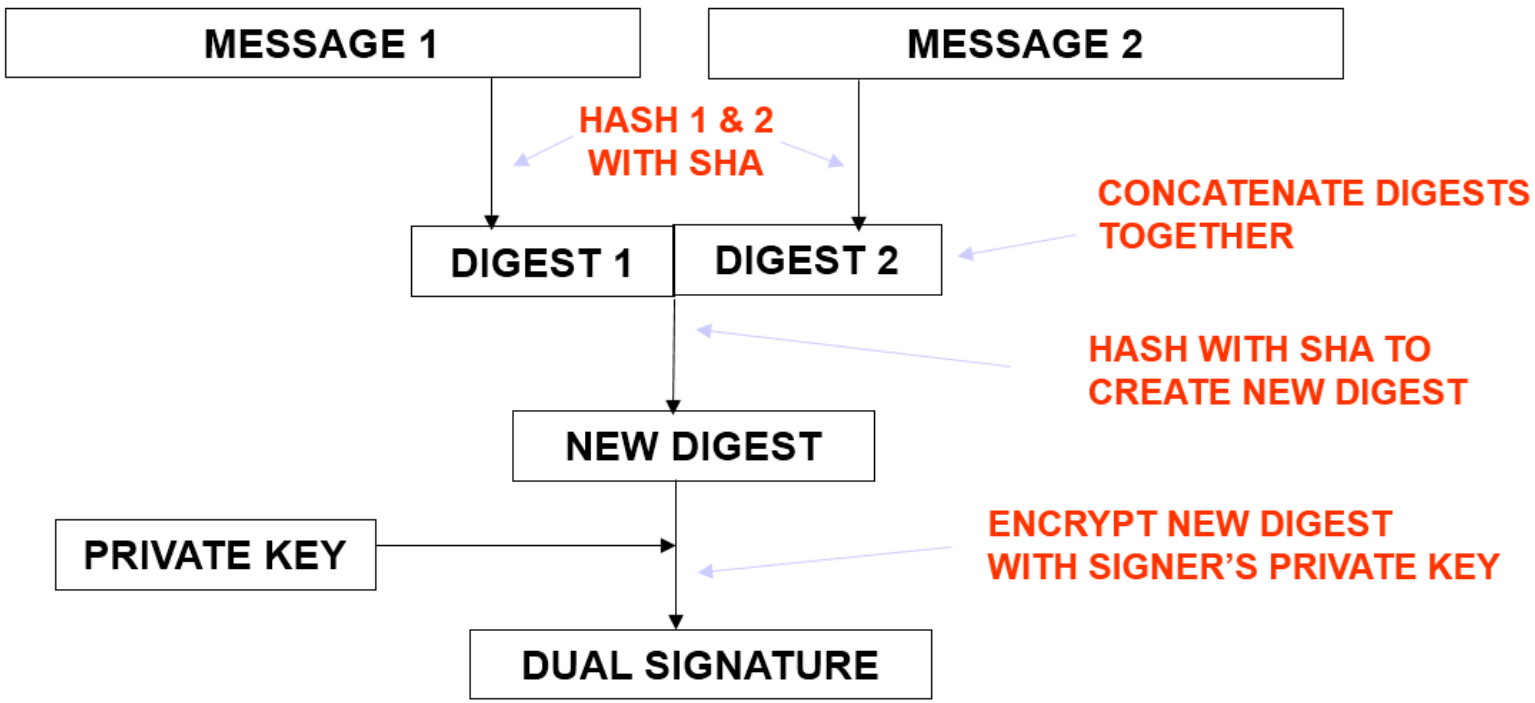
- The customer sends order and payment information to the merchant.
- The merchant requests payment authorization from the payment gateway prior to shipment.
- The merchant confirms order to the customer.
- The merchant provides the goods or service to the customer.
- The merchant requests payment from the payment gateway.

Key Technologies of SET

- Confidentiality of information: DES
- Integrity of data: RSA digital signatures with SHA-1 hash codes
- Cardholder account authentication: X.509v3 digital certificates with RSA signatures
- Merchant authentication: X.509v3 digital certificates with RSA signatures
- Privacy: separation of order and payment information using dual signatures

Dual Signatures

- Links two messages securely but allows only one party to read each.



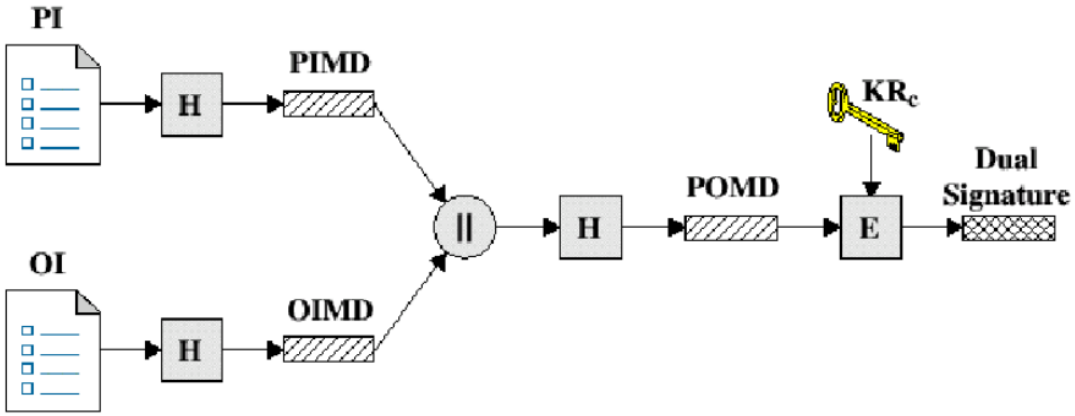
Dual Signature for SET

- Concept: Link Two Messages Intended for Two Different Receivers:
  Order Information (OI): Customer to Merchant
  Payment Information (PI): Customer to Bank
- Goal: Limit Information to A âĂIJNeed-to-KnowâĂİ Basis:
  Merchant does not need credit card number.
  Bank does not need details of customer order.
  Afford the customer extra protection in terms of privacy by keeping these items separate.
- This link is needed to prove that payment is intended for this order and not some other one.

Why Dual Signature?

- Suppose that customers send the merchant two messages:
  The signed order information (OI).
  The signed payment information (PI).
  In addition, the merchant passes the payment information (PI) to the bank.
- If the merchant can capture another order information (OI) from this customer, the merchant could claim this order goes with the payment information (PI) rather than the original.

Dual Signature Operation



- The operation for dual signature is as follows:
  Take the hash (SHA-1) of the payment and order information.
  These two hash values are concatenated [H(PI)||H(OI)] and then the result is hashed.
  Customer encrypts the final hash with a private key creating the dual signature.
  $DS = E_{KRC}[H(H(PI)||H(OI))]$

DS Verification by Merchant

- The merchant has the public key of the customer obtained from the customer's certificate.
- Now, the merchant can compute two values:
  H(PIMD || H(OI))
  $D_{KUC}[DS]$
- Should be equal!

DS Verification by Bank

- The bank is in possession of DS, PI, the message digest for OI (OIMD), and the customer's public key, then the bank can compute the following:
  H(H(PI) || OIMD)
  ‘ $D_{KUC}[DS]$

What did we accomplish?

- The merchant has received OI and verified the signature.
- The bank has received PI and verified the signature.
- The customer has linked the OI and PI and can prove the linkage.

Dual Signature Operation

card holder registration	purchase notification
merchant registration	sale transaction
purchase request	authorization reversal
payment authorization	capture reversal
payment capture	credit reversal
certificate query	
purchase inquiry	

Purchase Request

- Browsing, Selecting, and Ordering is Done
- Purchasing Involves 4 Messages:
  Initiate Request
  Initiate Response
  Purchase Request
  Purchase Response

Purchase Request: Initiate Request

- Basic Requirements:
  Cardholder Must Have Copy of Certificates for Merchant and Payment Gateway
- Customer Requests the Certificates in the Initiate Request Message to Merchant
  Brand of Credit Card
  ID Assigned to this Request/response pair by customer
  Nonce

Purchase Request: Initiate Response

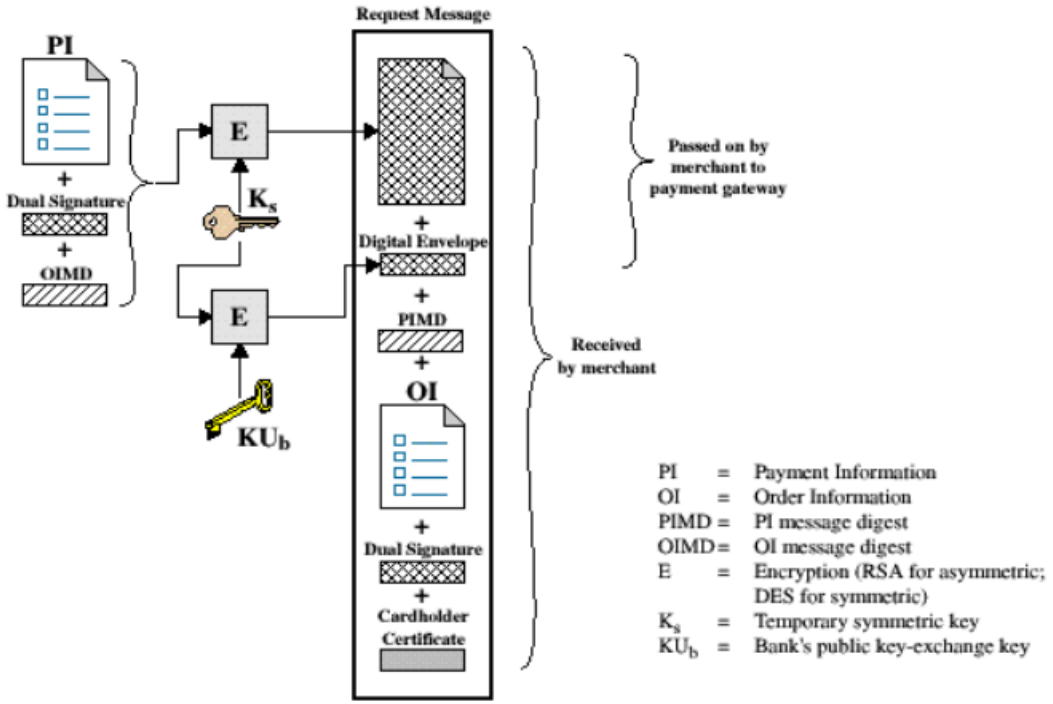
- Merchant Generates a Response
  Signs with Private Signature Key
  Include Customer Nonce
  Include Merchant Nonce (Returned in Next Message)
  Transaction ID for Purchase Transaction
- In Addition
  Merchant's Signature Certificate
  Payment Gateway's Key Exchange Certificate

Purchase Request: Purchase Request

- Cardholder Verifies Two Certificates Using Their CAs and Creates the OI and PI.
- Message Includes:
  Purchase-related Information
  Order-related Information
  Cardholder Certificate

Purchase Request

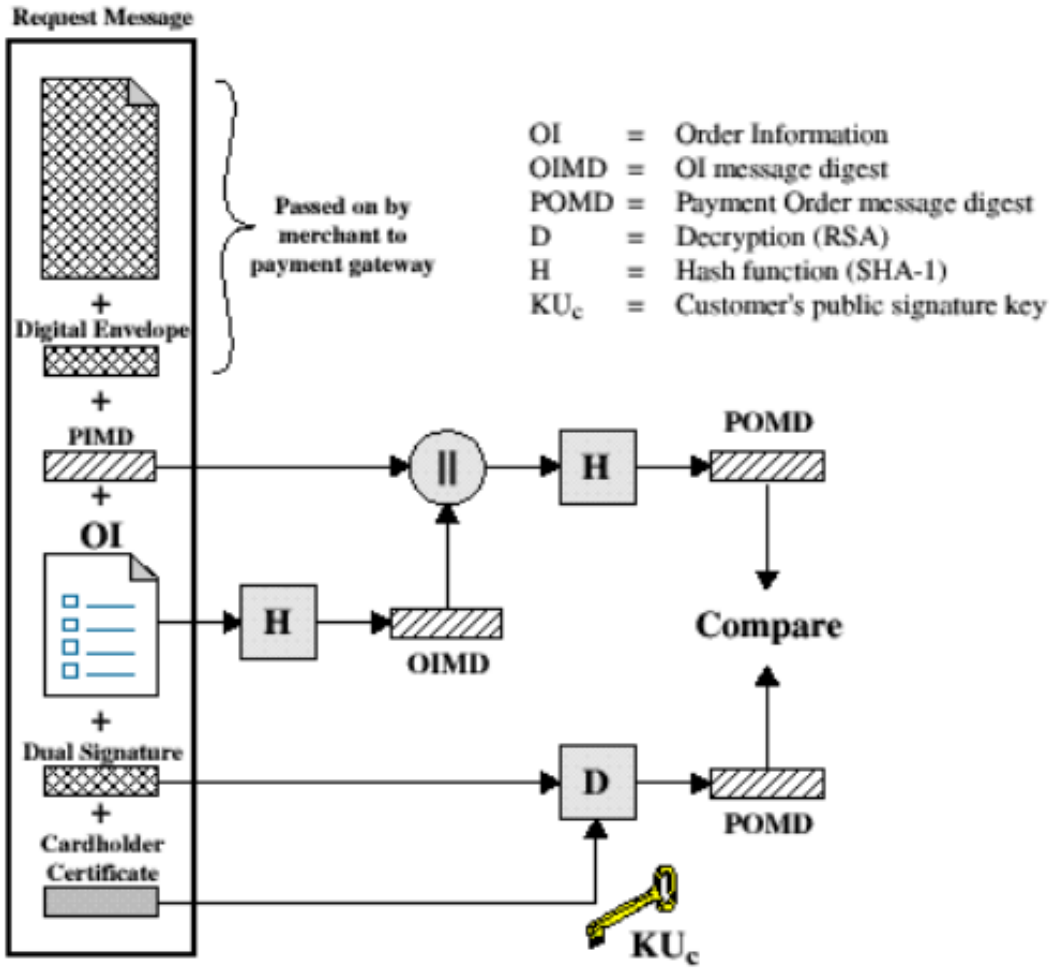
- The cardholder generates a one-time symmetric encryption key, KS.



Merchant Verifies Purchase Request

- When the merchant receives the Purchase Request message, it performs the following actions:
  Verify the cardholder certificates by means of its CA signatures.
  Verifies the dual signature using the customer's public key signature.

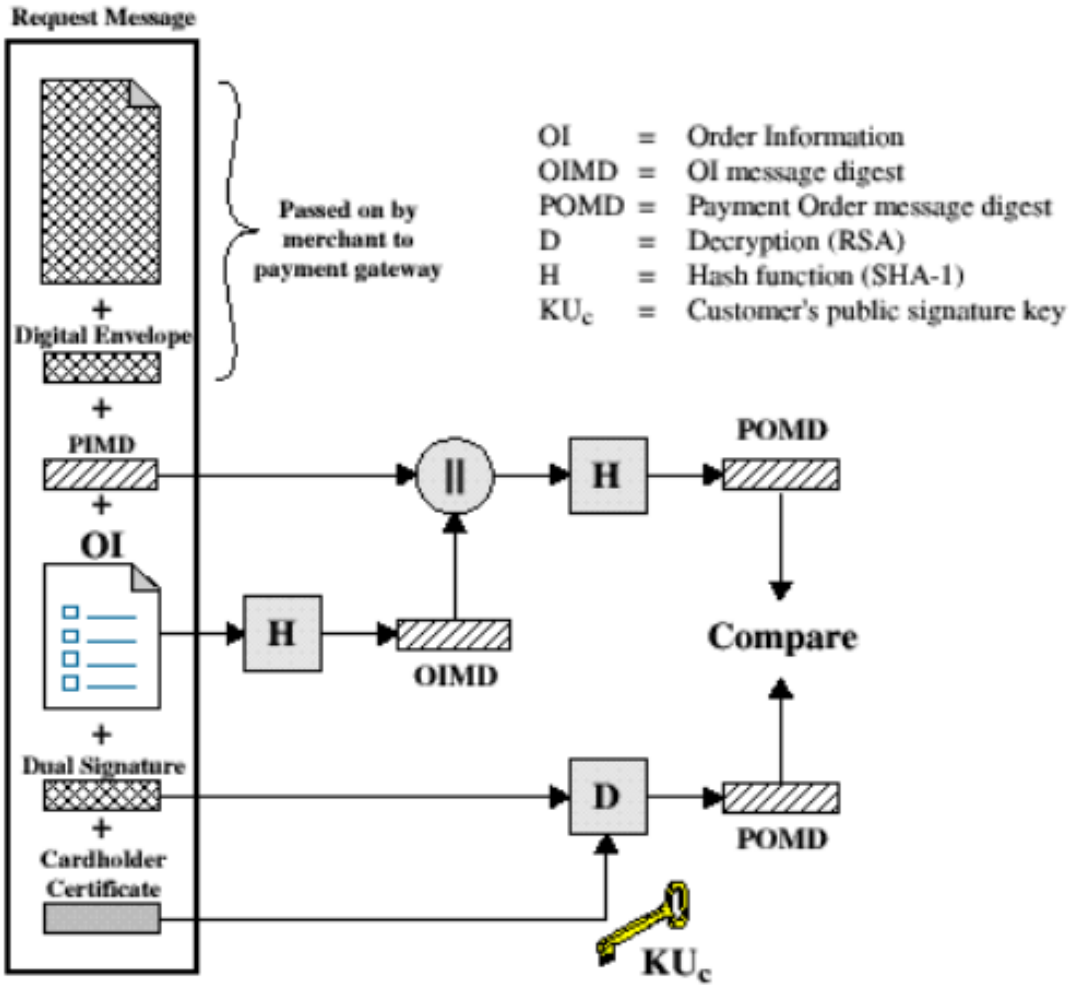
Merchant Verifies Purchase Request



Merchant Verification (cont'd)

- Processes the order and forwards the payment information to the payment gateway for authorization.
- Sends a purchase response to the cardholder.

Merchant Verification (cont'd)



Purchase Response Message

- Message that Acknowledges the Order and References Corresponding Transaction Number
- Block is
  Signed by Merchant Using its Private Key
  Block and Signature Are Sent to Customer Along with MerchantâĂŹs Signature Certificate
- Upon Reception
  Verifies Merchant Certificate
  Verifies Signature on Response Block
  Takes the Appropriate Action

Payment Process

- The payment process is broken down into two steps:
  Payment authorization
  Payment capture

Payment Authorization

- The merchant sends an authorization request message to the payment gateway consisting of the following:
  Purchase-related information
  - PI
  - Dual signature calculated over the PI & OI and signed with customer's private key.
  - The OI message digest (OIMD)
  - The digital envelop

  Authorization-related information

  Certificates

Payment Authorization (cont'd)

- Authorization-related information
  An authorization block including:
  - A transaction ID
  - Signed with merchant's private key
  - Encrypted one-time session key

  Certificates
  - Cardholder's signature key certificate
  - Merchant's signature key certificate
  - Merchant's key exchange certificate

Payment: Payment Gateway

- Verify All Certificates
- Decrypt Authorization Block Digital Envelope to Obtain Symmetric Key and Decrypt Block
- Verify Merchant Signature on Authorization Block
- Decrypt Payment Block Digital Envelope to Obtain Symmetric Key and Decrypt Block
- Verify Dual Signature on Payment Block
- Verify Received Transaction ID Received from Merchant Matches PI Received from Customer
- Request and Receive Issuer Authorization

Authorization Response

- Authorization Response Message
  - Authorization-related Information
  - Capture Token Information
  - Certificate

SET Overhead

- Simple purchase transaction:
  - Four messages between merchant and customer
  - Two messages between merchant and payment gateway
  - 6 digital signatures
  - 9 RSA encryption/decryption cycles
  - 4 DES encryption/decryption cycles
  - 4 certificate verifications
- Scaling:
  - Multiple servers need copies of all certificates

Thank You . . . !