Seguridad de la Información

TEMA 4 (PARTE B)

Seguridad y Privacidad
En Aplicaciones Telemáticas

SEGURIDAD EN PAGOS ELECTRÓNICOS

Conceptos generales

- En el ámbito del e-commerce se han desarrollado esquemas de pago electrónico que proporcionan en el mundo digital la misma heterogeneidad que los sistemas de pago tradicionales

- En la mayoría de los sistemas de pagos electrónicos disponibles, los pagos se realizan a través de redes abiertas como Internet
  - pero la correspondencia entre los pagos electrónicos y la transferencia del valor real es realizada y garantizada por los bancos, a través de los **sistemas financieros de compensación**
    - estos sistemas utilizan para su funcionamiento las redes cerradas de las instituciones bancarias, las cuales son consideradas comparativamente más seguras

- En general, los pagos electrónicos involucran a un **comprador y a un vendedor**
  - Adicionalmente, existen sistemas que involucran a **TTPs**
- Más aún, puede existir algún tipo de entidad que ejerza de mediador para la **resolución de disputas**
  - por lo general, las disputas se resuelven fuera del sistema de pago, y en muchos casos el protocolo ni siquiera especifica cómo gestionarlas

- Existen varias formas de clasificar los sistemas de pagos electrónicos, y dependen de:
  - Cuando el vendedor contacta con el banco para verificar el proceso de pago
  - Cuando el comprador procede con la transacción y carga de dinero en la cuenta del vendedor
  - La cantidad de dinero implicada en cada transacción

- Los sistemas de pagos electrónicos se pueden clasificar según **cuando el vendedor contacta con el banco**:
  - **On-line**: antes de enviar el producto, el vendedor contacta con la entidad financiera para verificar la validez del pago del comprador

  - **Off-line**: cierto tiempo después de que el vendedor haya aceptado el pago y enviado el producto, realiza el depósito del dinero que le ha dado el comprador
    - para que la entidad financiera lo verifique y lo ingrese en su cuenta
    - es decir, el vendedor no contacta con el banco durante el proceso de compra-venta

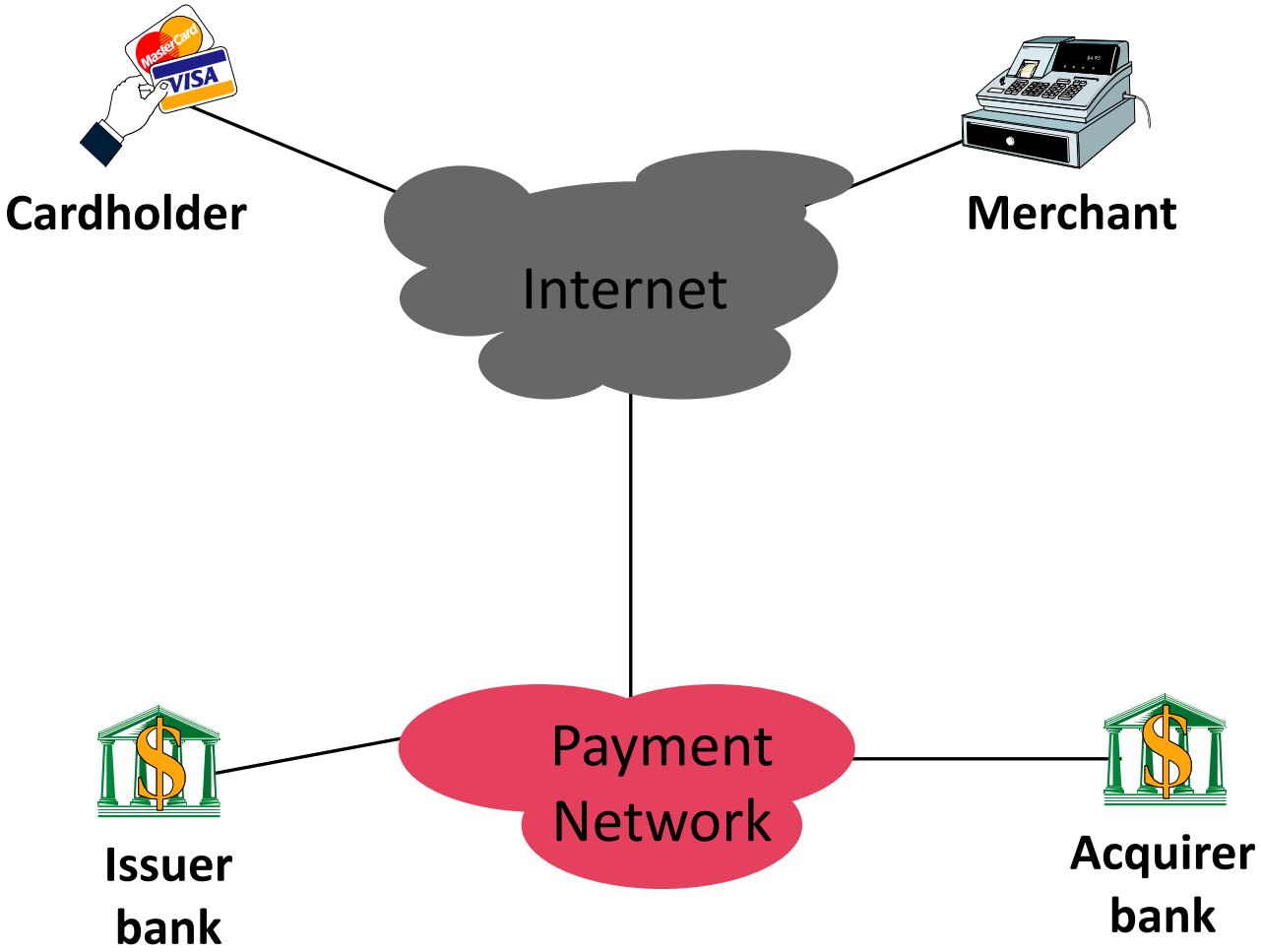
- Existe otra forma de clasificar los pagos electrónicos, atendiendo al **momento en que se retira el dinero de la cuenta del <u>comprador</u>**:
  - **Sistemas de pre-pago**: el comprador ve decrementada su cuenta bancaria antes de realizar la compra
    - este método se correspondería con los sistemas de **monedero electrónico** y **tarjetas telefónicas**
    - éste sería el sistema más análogo al papel moneda tradicional
  - **Sistemas de pago instantáneo**: cuando al comprador se le realiza el cargo en cuenta justo en el momento de realizar la compra
    - se correspondería con los sistemas actuales de pagos con **tarjeta de débito**
  - **Sistemas de post-pago**: cuando Alice realiza la compra, el Banco asegura al vendedor que se le hará efectiva la cantidad acordada
    - pero Alice sólo verá decrementada su cuenta cierto tiempo después de haberse realizado la compra

- Otro criterio de clasificación es **según la cantidad implicada en la transacción**. De esta forma se clasifican los pagos electrónicos como:
  - **Macropagos**: cualquier pago superior a 10 euros
  - **Pagos**: la cantidad está comprendida entre 1 y 10 euros
  - **Micropagos**: cualquier pago inferior a 1 euro

- Normalmente los **pagos inferiores a 10 euros** presentan el problema del **coste de implementación**
  - no tendría sentido utilizar un sistema de pago cuyo coste económico sea de orden de magnitud o superior al importe de la transacción

- Ejemplos de protocolos:
  - **On-line y trazables**:
    - First Virtual
    - CyberCash
    - iKP
    - SET
    - …
  - **Micropagos**:
    - PayPal
    - Google Checkout
    - Amazon Payments
    - iTunes Store
    - …

Tarjetas de crédito



- Problemas en estos medios de pagos:
  - Ataques de escucha
  - Suplantación de identidad (cliente o comerciante)
  - Generación de dato
  - Modificación del dato
  - Etc.

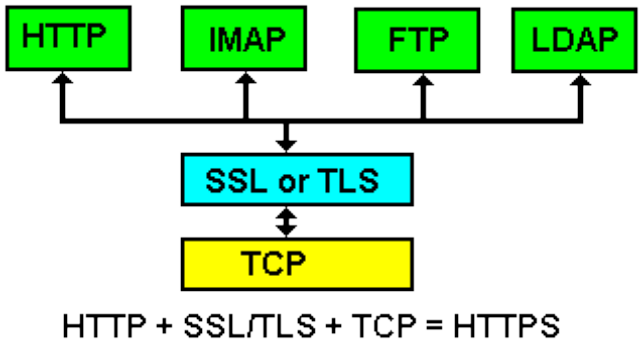
- Soluciones para evitar estos problemas:
  - Mecanismos criptográficos
  - Mecanismos de autenticación de usuarios
  - Firma digitales
  - Certificados digitales

Y protocolos específicos que implementen estas soluciones

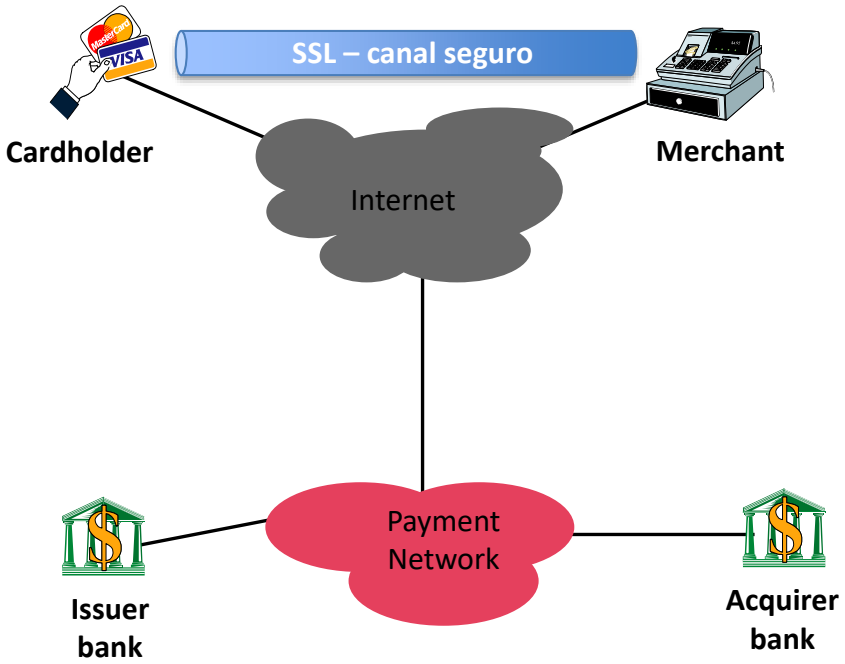
Protocolo SSL (protocolo original) vs TLS (protocolo actual)

- SSL (Secure Sockets Layer) es un protocolo de propósito general (como TLS) para establecer conexiones seguras
  - No es un protocolo de pago, pero se usaba por seguridad
  - SSL lo creó originalmente Netscape (1994).
    - La última versión: SSLv3 – *No se utiliza en la actualidad !!*

- **TLS** (Transport Secure Layer) se creó dentro del IETF
  - Utiliza el mismo formato para la cabecera de los paquetes que SSL
  - La primera versión de TLS puede verse como SSLv3.1, pero difiere en:
    - Número de versión
    - En el código de autenticación del mensaje (MAC)
    - En la función pseudo-aleatoria
    - En los códigos de alerta
    - En la lista de algoritmos de cifrado
    - En los mensajes de verificación del certificado y de finalización
    - En algunas partes del algoritmo criptográfico



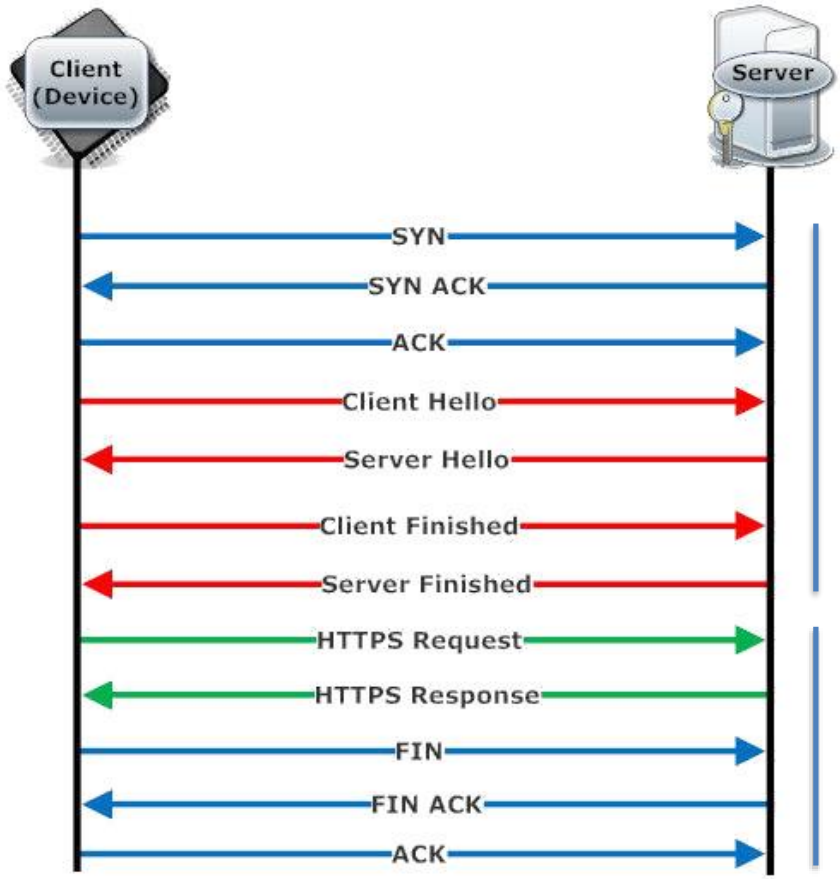
- SSL aplica:
  - Criptografía de asimétrica:
    - Gestión de certificados digitales
    - RSA o Diffie-Hellman
  - Criptografía simétrica:
    - DES, 3DES, RC2, RC4 o IDEA



- SSL autentifica al servidor
  - Utilizando certificados digitales X.509 v3
  - Opcionalmente, también puede certificar al cliente

- SSL asegura la integridad de los datos
  - Mediante códigos de autentificación de mensajes (MAC) y una clave secreta
  - MD5 o SHA-1

- Originalmente SSL proveía **confidencialidad en el pago electrónico**
  - Garantizaba la creación de un canal seguro entre cliente y servidor



Negociación de las credenciales de seguridad y el modo de protección de los canales de comunicación – criptografía de clave pública

Comunicación segura mediante Cifrado simétrico

- Sin embargo, SSL presentaba algunos **problemas importantes**:
  - Sólo protege transacciones entre dos puntos, mientras que una transacción electrónica basada en una tarjeta de crédito involucra al menos a un banco
  - SSL no protege al comprador frente al vendedor
    - El vendedor puede obtener información de la tarjeta que podría utilizar en un futuro de forma ilícita
  - No hay mecanismos de autentificación de tarjetas
  - No hay mecanismos de facturación o de gestión de recibos
    - cualquier reclamación queda a la buena voluntad del vendedor

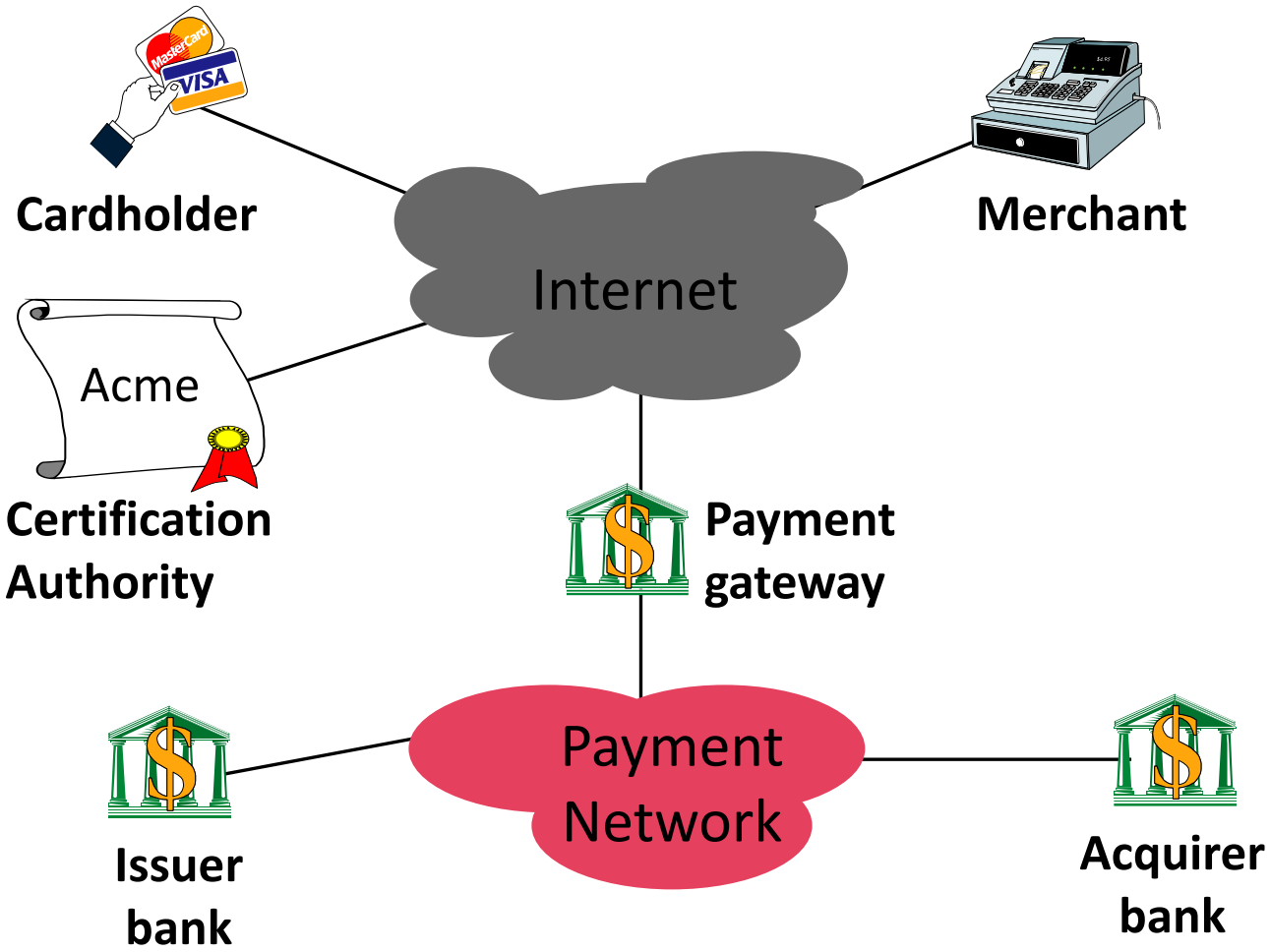
Por tanto, se requerían de otros tipos de protocolos más específicos ...

Protocolo SET (Secure Electronic Transactions)

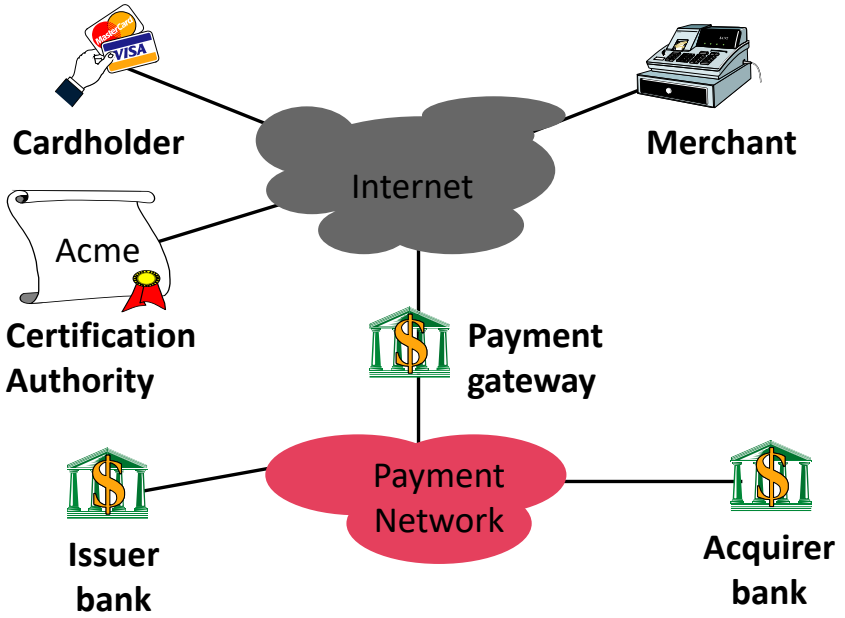
- Protocolo desarrollado en 1996 por VISA y Mastercard (+ American Express), en colaboración con:
  - IBM
  - Microsoft
  - Verisign
  - RSA
  - Netscape
  - GTE

- Objetivo: proporcionar seguridad a las **transacciones electrónicas basadas en tarjetas de crédito**
  - para poder reducir el fraude mercantil
  - y garantizar el pago a través de esas mismas redes

- Arquitectura SET:



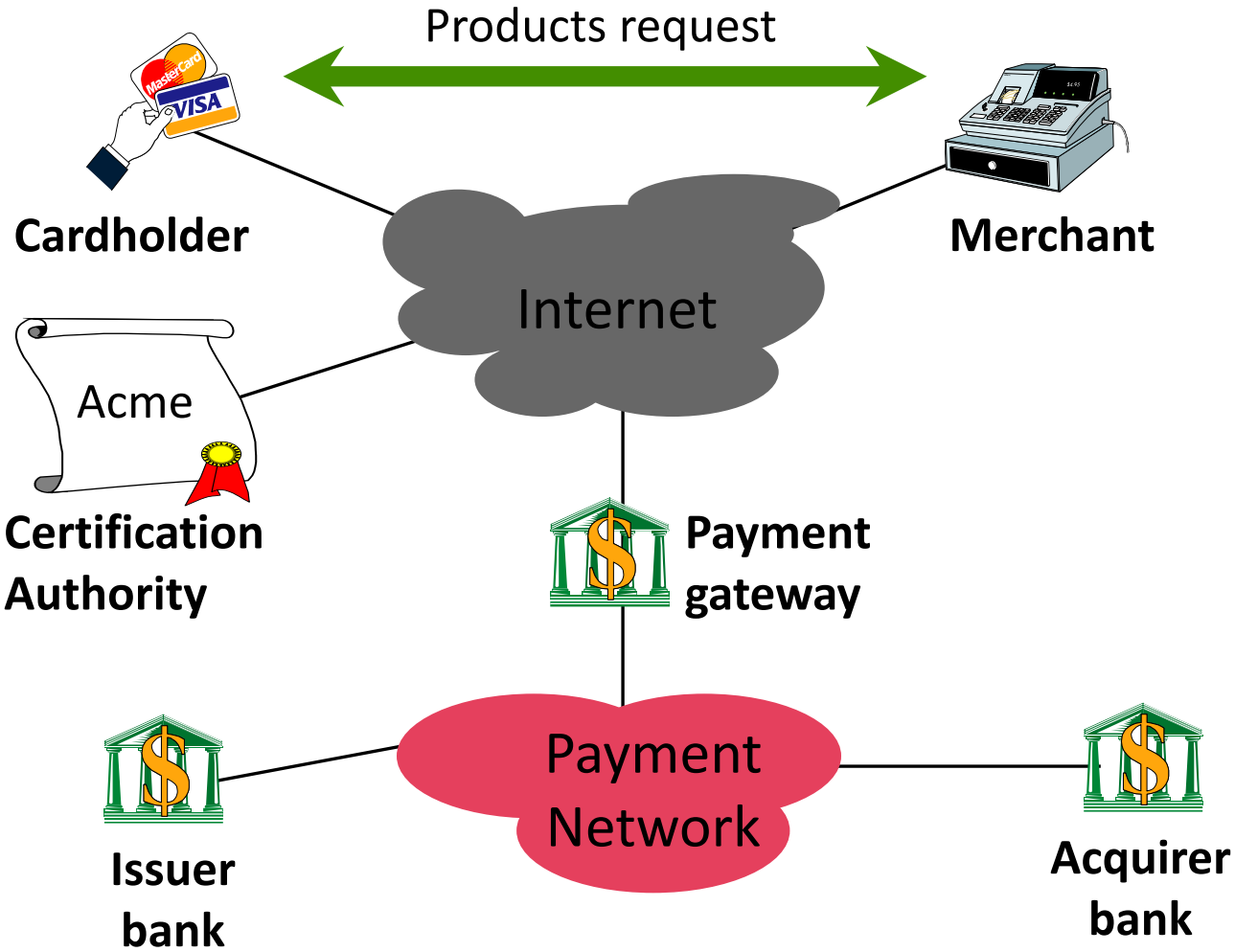
- SET
  - A diferencia de SSL, fue diseñado para el comercio electrónico
  - Sin embargo, no es un sistema de pago en sí mismo, sino un **conjunto de protocolos de seguridad y de formatos estándar**
    - que permiten a los usuarios usar de una forma segura a través de Internet la infraestructura ya existente de tarjetas de crédito



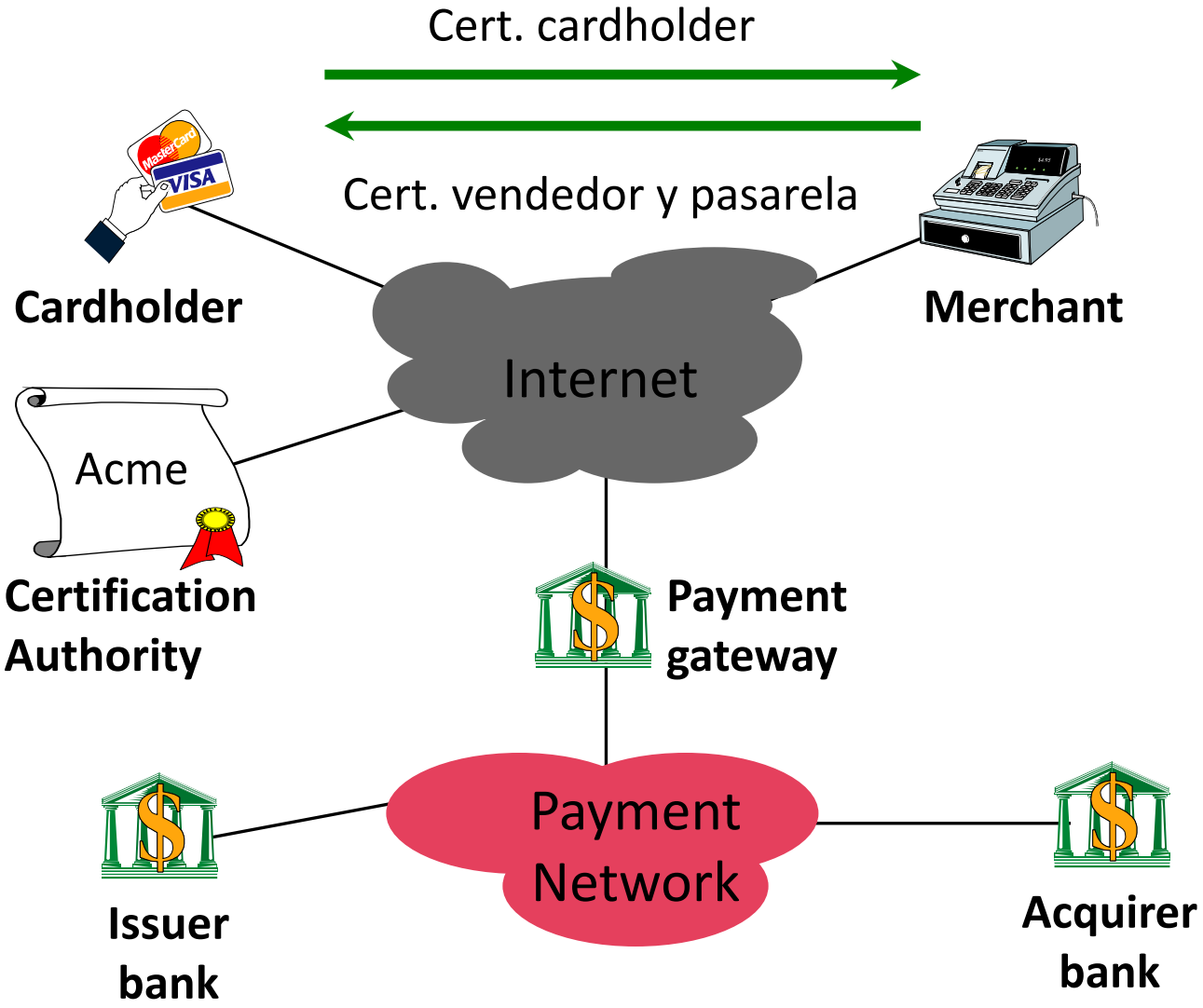
- SET proporciona:
  - **Confidencialidad** en las comunicaciones entre las entidades que intervienen en la transacción
  - **Autenticidad**, a través del uso de certificados digitales X.509
    - Todas las entidades, incluyendo el cliente, el vendedor y la pasarela de pago, han de tener certificados X.509
    - Es necesario el servicio de una o más Autoridades de Certificación
  - **Privacidad**, porque la información sólo está disponible para las diferentes entidades cuando y donde es necesario
  - **Integridad**
  - **Reduce las disputas debido al no repudio**
  - Autorización de pago
  - Confirmación de la transacción
  - Garantía de pago al vendedor

- Pasos de una transacción:
  1. **Petición de producto**
  2. **Inicialización**: envío de certificados
  3. **Información del pedido e instrucciones de pago**: descripción de la compra
  4. **Petición de autorización**: vendedor-pasarela y pasarela-banco emisor
  5. **Aprobación de autorización**: el banco emisor autoriza el pago
  6. **Finalización**: el vendedor reclama la cantidad a la pasarela
     - Petición de compensación hacia el banco del vendedor

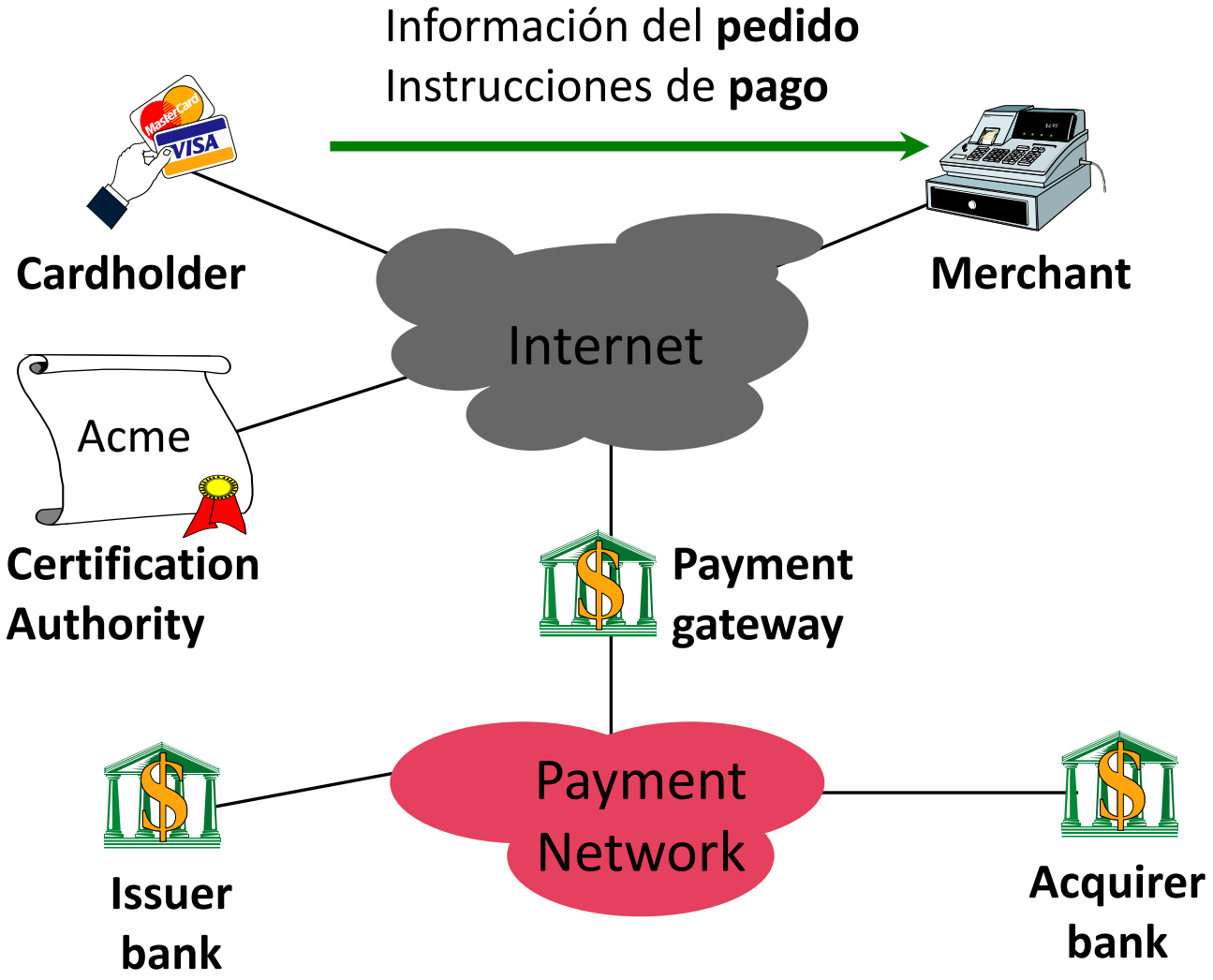
1. Petición del producto



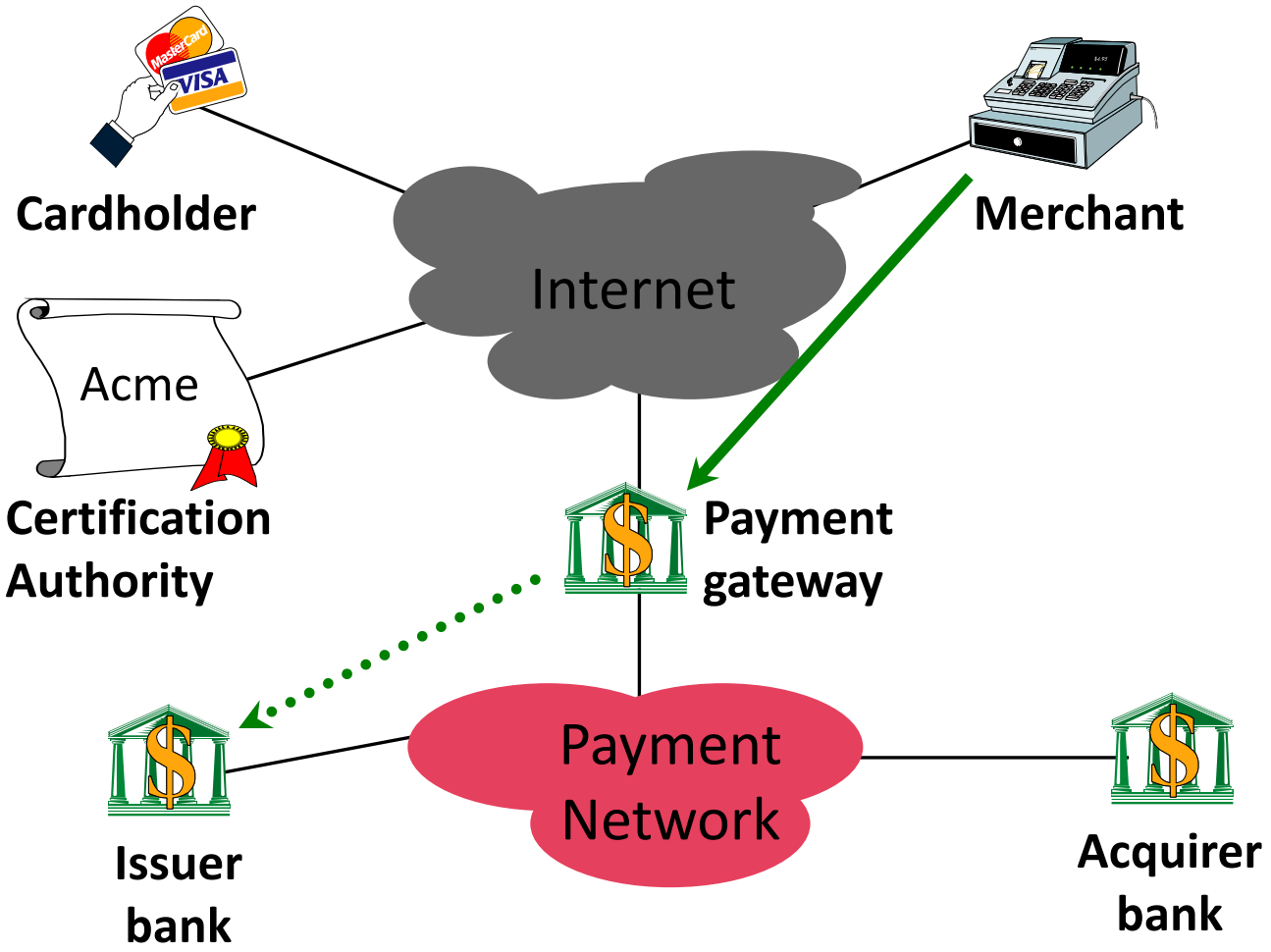
2. **Inicialización** (envío de certificados y autenticación)



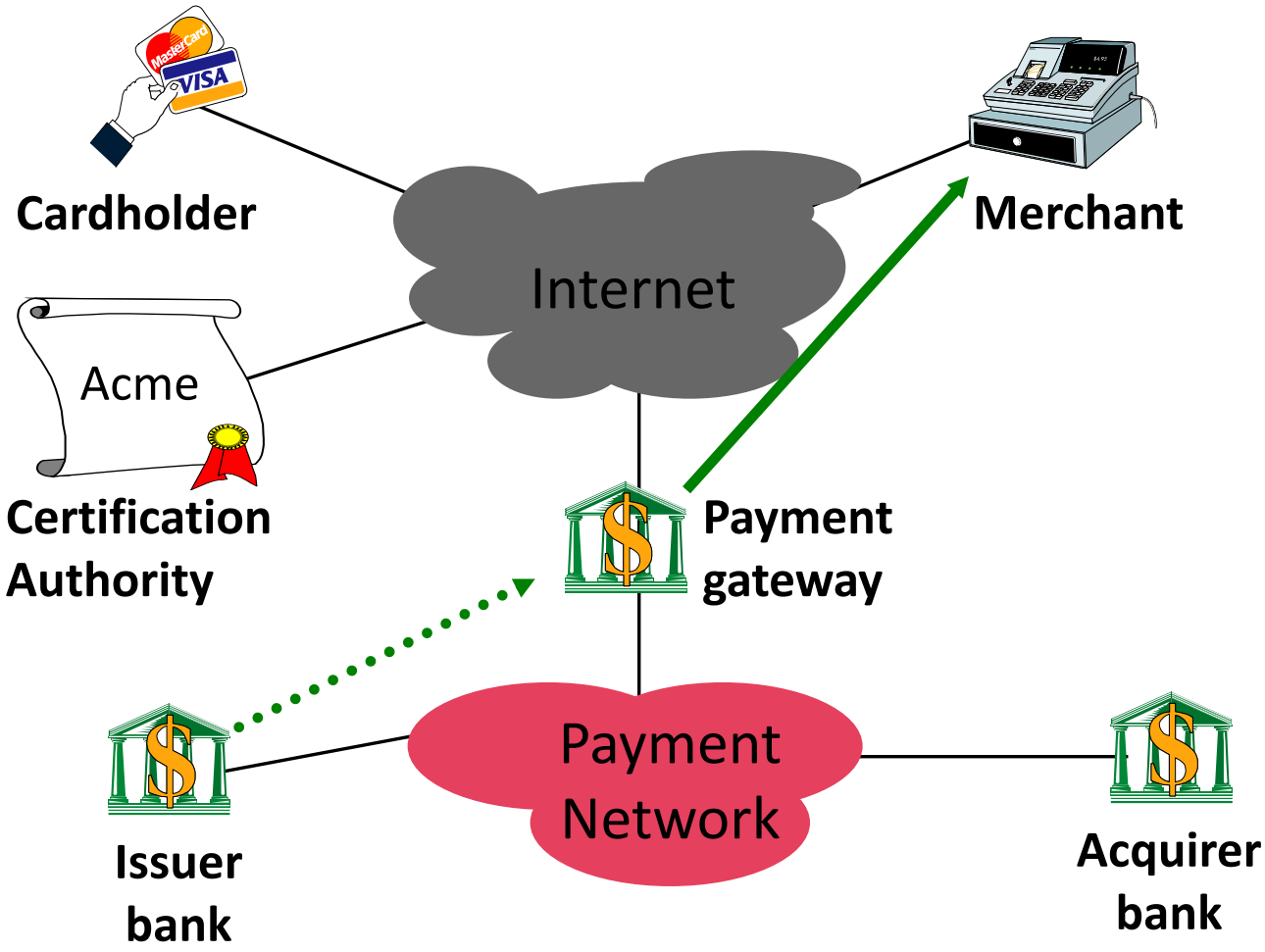
3. Información del pedido e instrucciones de pago



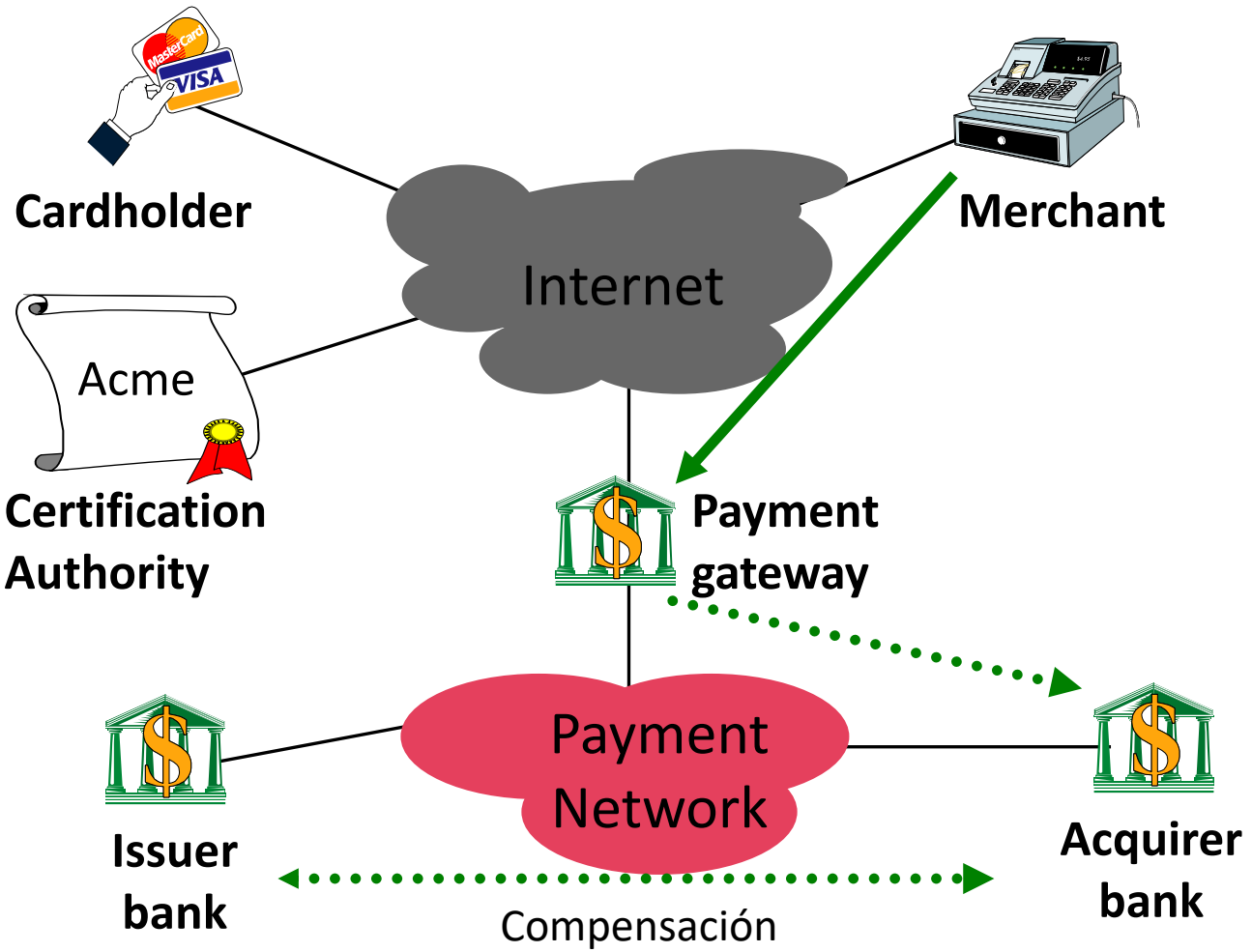
4. Petición de autorización



5. Aprobación de autorización

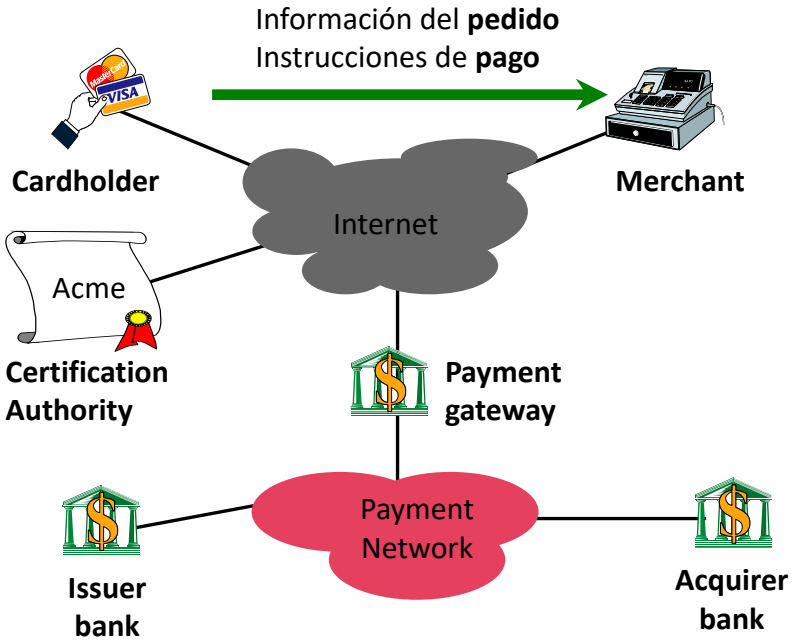


6. Finalización



FIRMA DUAL (en SET) – PRIVACIDAD

- SET introduce una importante innovación técnica: la **firma dual**
  - El propósito de este tipo de firma es enlazar dos mensajes que han de ir a receptores diferentes
- En este caso, el **cliente** quiere enviar:
  - la información del pago (Payment Information) al banco
  - la información del pedido (Order Information) al comerciante

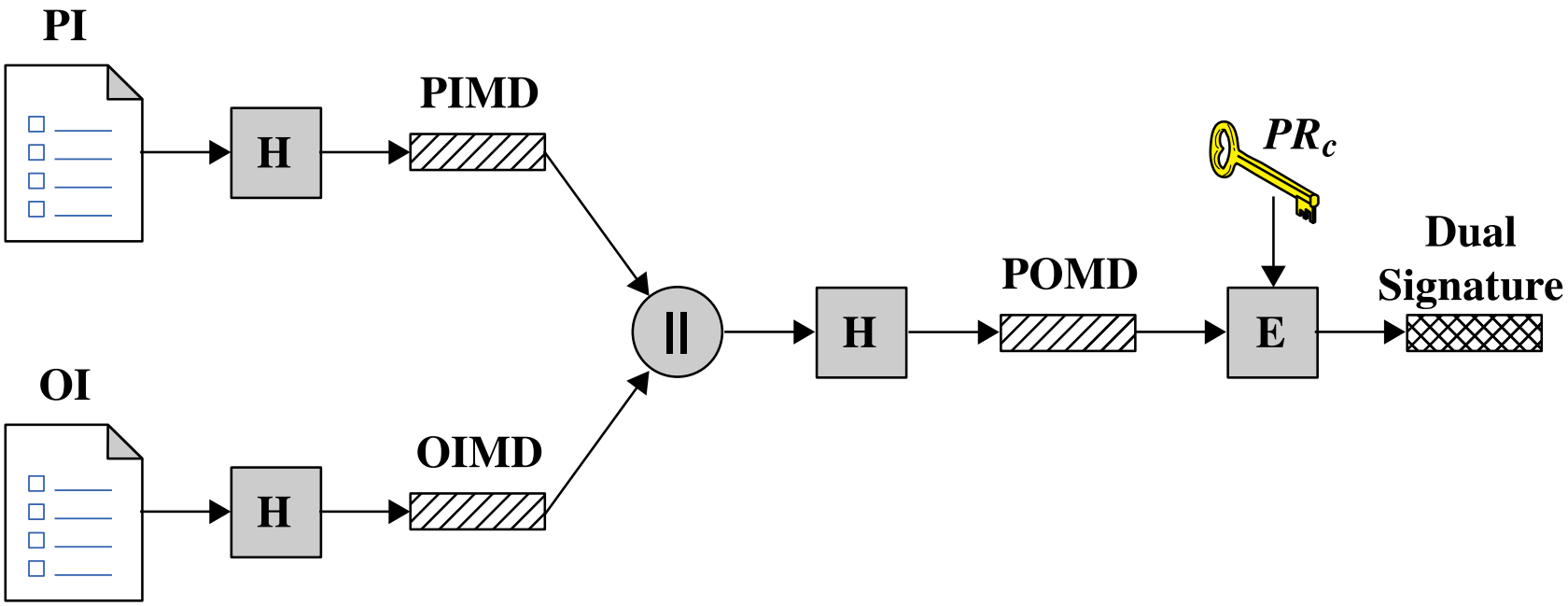


FIRMA DUAL (en SET) – PRIVACIDAD

- SET introduce una importante innovación técnica: la **firma dual**
  - El propósito de este tipo de firma es enlazar dos mensajes que han de ir a receptores diferentes
- En este caso, el **cliente** quiere enviar:
  - la información del pago (Payment Information) al banco
  - la información del pedido (Order Information) al comerciante
- Pero se ofrece mayor **privacidad** al cliente si ambos ítems se mantienen por separado:
  - ni el comerciante necesita conocer el número de tarjeta del cliente
  - ni el banco necesita conocer los detalles del pedido del cliente
- No obstante, es necesario que ambos ítems queden enlazados de alguna forma, para una posible **resolución de disputas** posterior

¡¡HAY NO REPUDIO!!

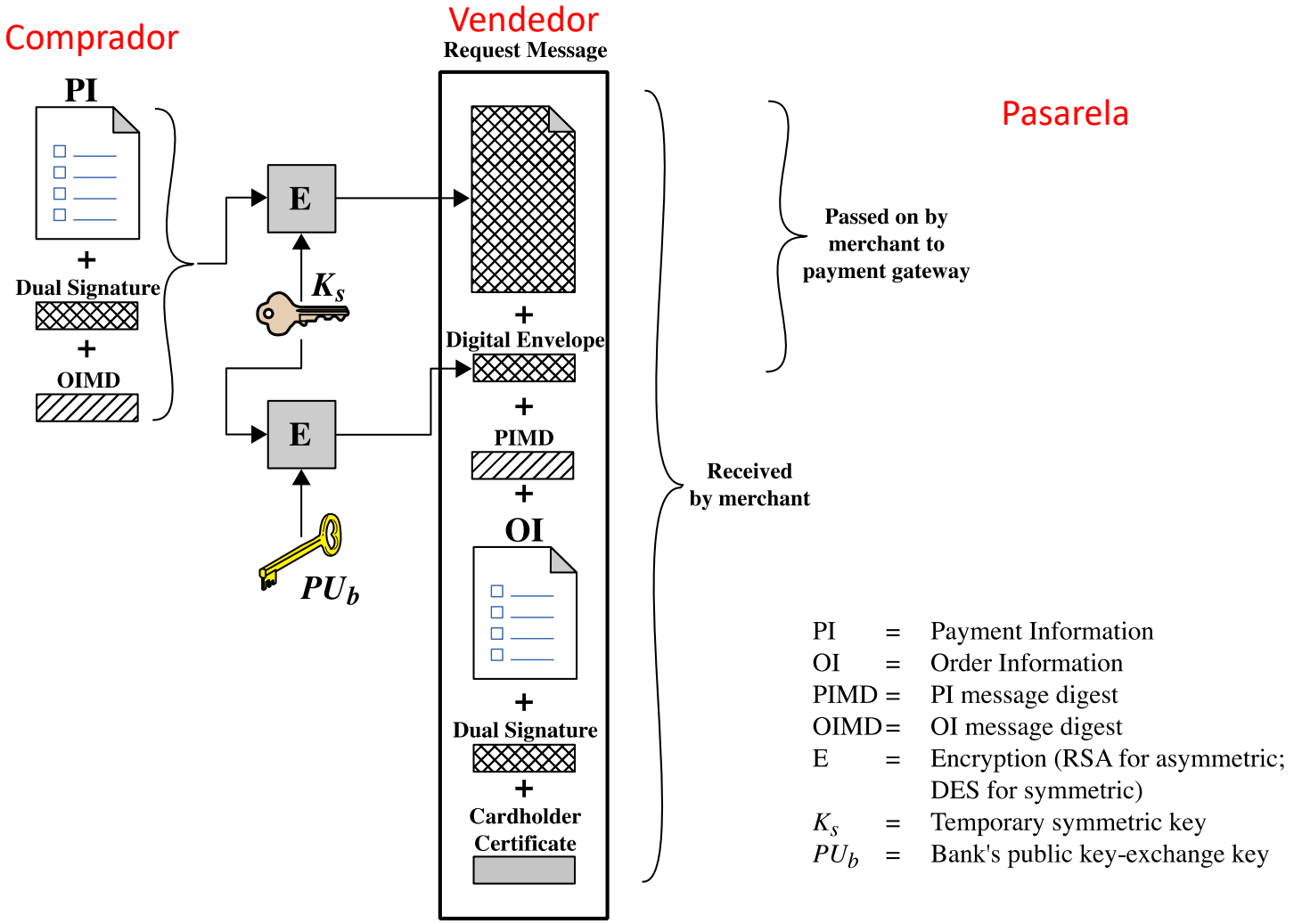
Firma dual



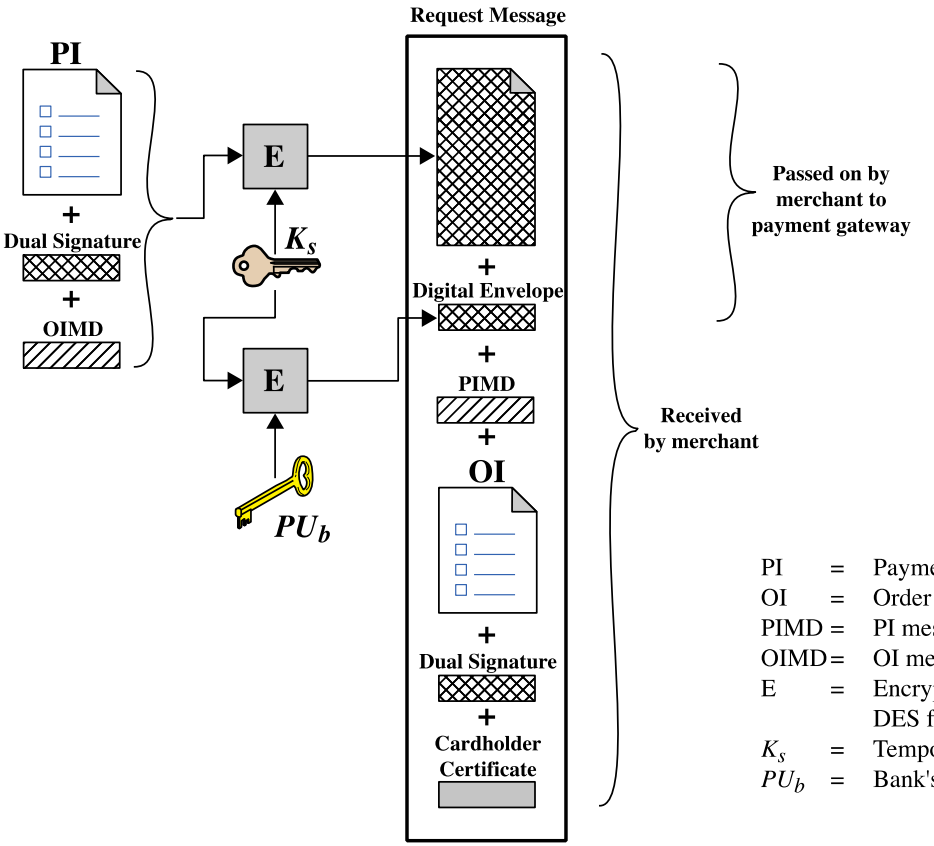
PI = Payment Information
OI = Order Information
H = Hash function (SHA-1)
|| = Concatenation

PIMD = PI message digest
OIMD = OI message digest
POMD = Payment Order message digest
E = Encryption (RSA)
PR_c = Customer's private signature key

- Petición de compra enviada por el comprador al vendedor:



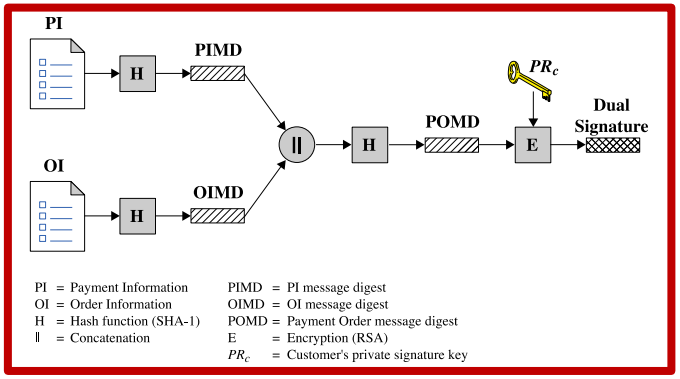
PI = Payment Information
OI = Order Information
PIMD = PI message digest
OIMD = OI message digest
E = Encryption (RSA for asymmetric; DES for symmetric)
K_s = Temporary symmetric key
PU_b = Bank's public key-exchange key

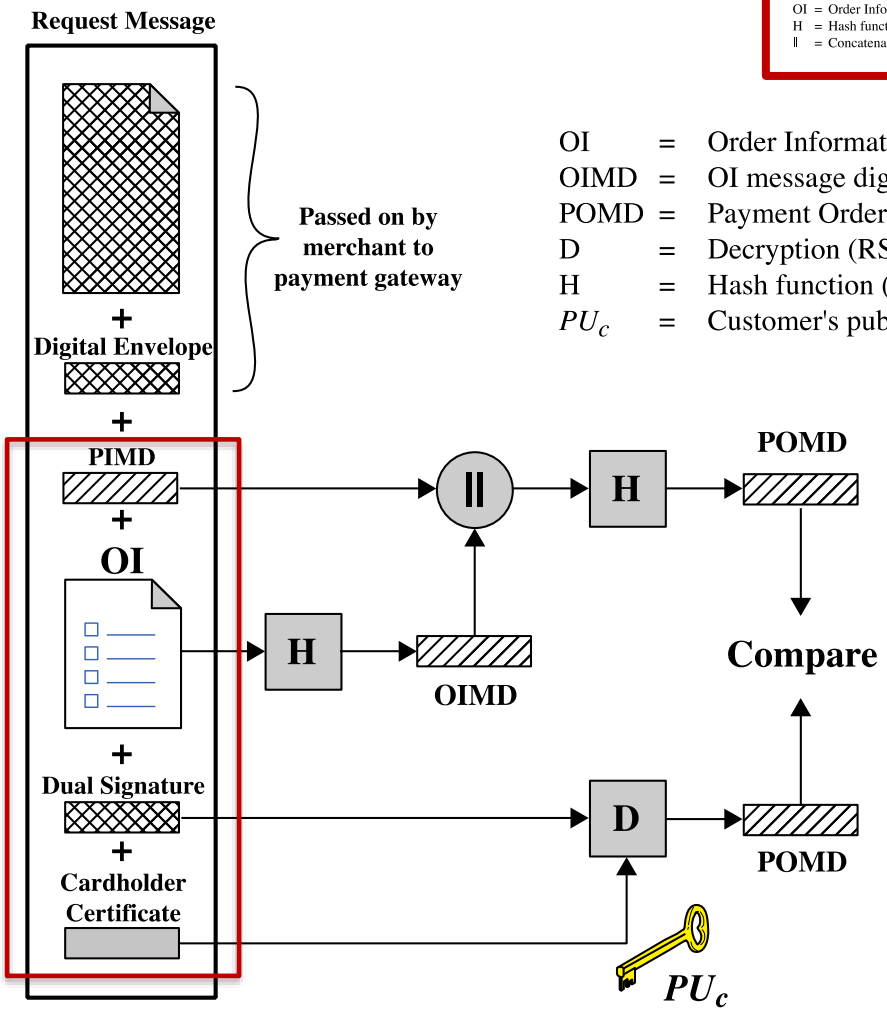


¿Qué hace el vendedor al recibir el mensaje?

PI = Payment Information
OI = Order Information
PIMD = PI message digest
OIMD = OI message digest
E = Encryption (RSA for asymmetric; DES for symmetric)
K_s = Temporary symmetric key
PU_b = Bank's public key-exchange key

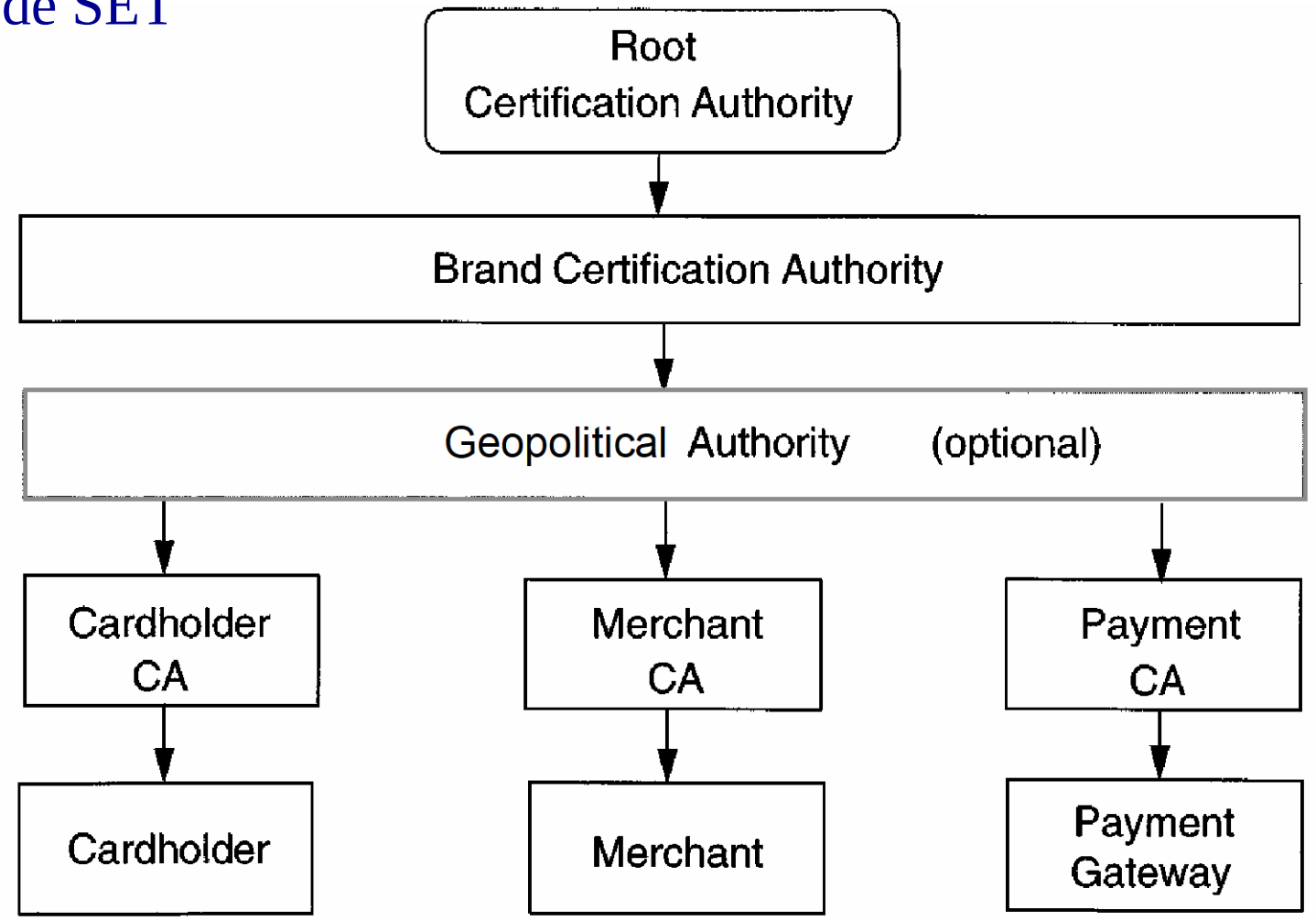
- Verificación del vendedor:





OI = Order Information
OIMD = OI message digest
POMD = Payment Order message digest
D = Decryption (RSA)
H = Hash function (SHA-1)
PU_c = Customer's public signature key

- PKI de SET


Root
Certification Authority
Brand Certification Authority
Geopolitical Authority (optional)
Cardholder CA
Merchant CA
Payment CA
Cardholder
Merchant
Payment Gateway

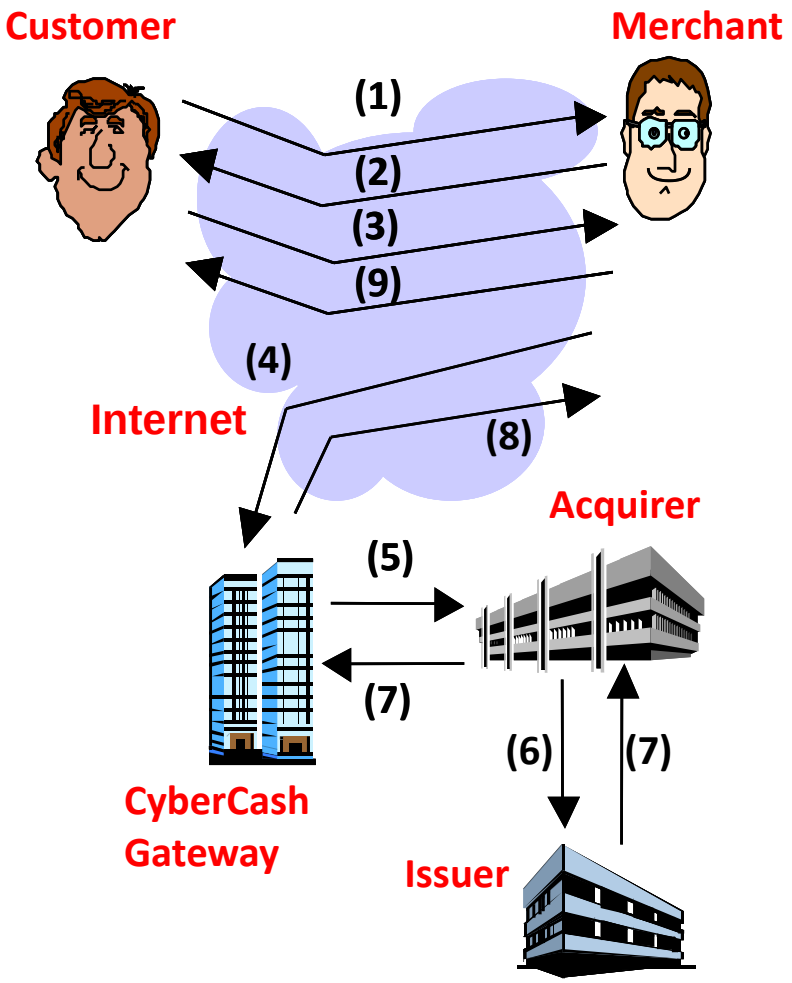
- Ejercicio 1 y 2:

- Ventajas de uso del SET:
  - Muy seguro y bien diseñado
  - Garantiza autenticación, confidencialidad, integridad, no-repudio y privacidad
    - Si el banco del comprador autoriza el pago, el vendedor tiene la garantía de ese pago (no-repudio)
    - Evita que el vendedor acceda a los datos de la tarjeta
    - Evita que el banco acceda a la información de los productos comprados

- Desventajas de uso del SET:
  - Es dependiente de algoritmos específicos (RSA, DES, SHA1)
  - Gestión de certificados digitales
  - Fuerte esfuerzo para la implantación (especialmente para el vendedor)
  - No adaptado a micropagos

Protocolo Cybercash

- Se basa en el uso de una **pasarela propia** que gestiona los pagos electrónicos
- Permite el uso de cualquier tipo de tarjeta
- Integra el software de cliente (**cyberwallet**) con la red financiera del banco del comprador
- Se realiza la **autenticación** de todas las entidades y el **cifrado** de los datos relativos al pago



1. Purchase order (description).
2. Payment request (price).
3. Payment order (signed by the CyberWallet).
4. Redirection of the payment order.
5. Verification of the order. Authorization request.
6. Request for authorization of the issuer bank.
7. Authorization reply (acquirer and gateway).
8. Sending the **encrypted bills** (customer and merchant)
9. Redirection of the **customer's bill**.

- Algunos problemas:
  - Parte de la información del cliente es conocida por la pasarela, por lo que se pueden analizar los hábitos del cliente
    - **Problema de privacidad** que no existía en SET
  - Uso de DES (56 bits) y RSA (1024 bits)

- Tras caer en bancarrota, Verisign adquirió los derechos sobre la marca y el protocolo de pago
- Posteriormente Paypal compró la solución a Verisign

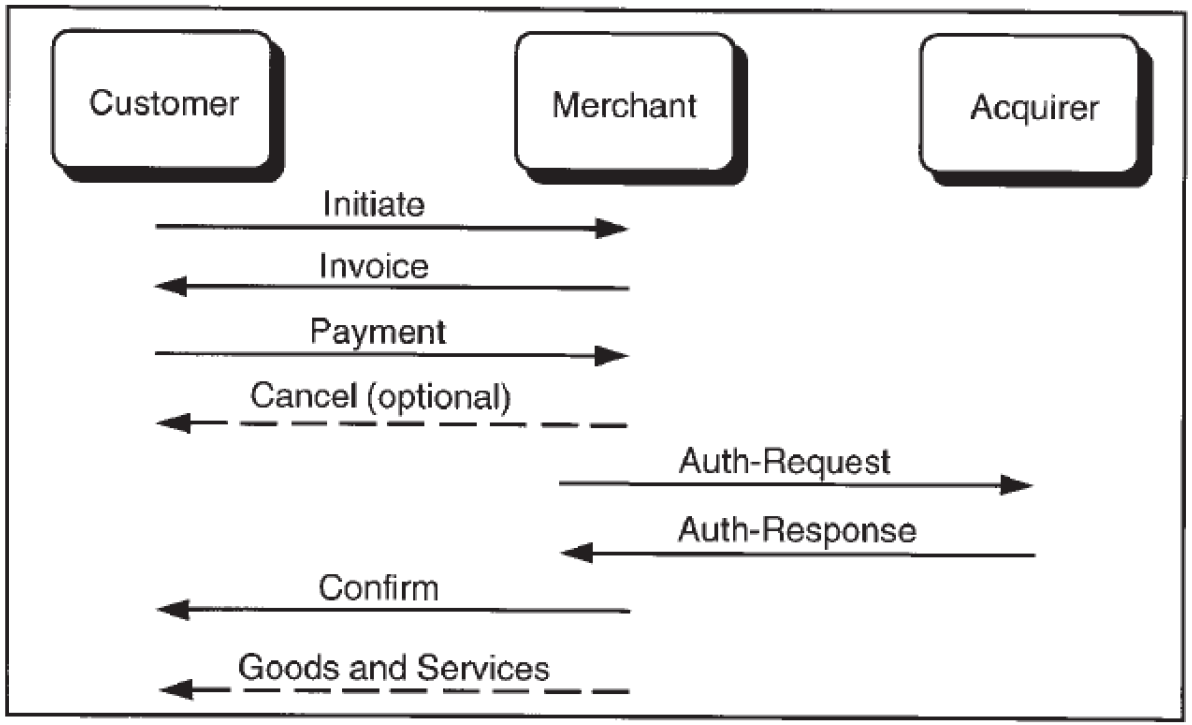
Protocolos iKP

- iKP (i = 1, 2, 3) es una familia de protocolos de pago, **basado en criptografía de clave pública**, y desarrollado por IBM

- Estos protocolos (1KP, 2KP, 3KP) se diferencian entre sí en el número de entidades que poseen su propio par de claves públicas
  - Cuanto mayor sea el número de entidades que posean un par <*clave pública*, *clave privada*>, mayor es el nivel de seguridad proporcionado

- La implantación de los protocolos 2KP y 3KP se realiza de forma gradual para conseguir **un pago seguro multiparte** completo, requiriendo una infraestructura de certificación avanzada

- Cada uno de los protocolos consta de seis pasos:



- El contenido de estos pasos varía dependiendo de a qué protocolo nos estemos refiriendo

- Los elementos intercambiados en una transacción iKP, y los campos formados por la combinación de tales elementos, son:

Item	Description
CAN	Customer's account number (e.g., credit card number)
ID_M	Merchant ID; identifies merchant to acquirer
TID_M	Transaction ID; uniquely identifies the transaction
DESC	Description of the goods; includes payment information such as credit card holder's name and bank identification number
$SALT_C$	Random number generated by C; used to randomize DESC and thus ensure privacy of DESC on the M to A link
$NONCE_M$	Random number generated by a merchant to protect against replay
DATE	Merchant's current date/time
PIN	Customer's PIN which, if present, can be optionally used in 1KP to enhance security
Y/N	Response from card issuer; Yes/No or authorization code
R_C	Random number chosen by C to form CID
CID	A customer pseudo-ID which uniquely identifies C; computed as $CID = H(R_C, CAN)$
V	Random number generated in 2KP and 3KP by merchant; used to bind the Confirm and Invoice message flows

Item	Description
Common	Information held in common by all parties: $PRICE, ID_M, TID_M, DATE, NONCE_M, CID, H(DESC, SALT_C), [H(V)]$
Clear	Information transmitted in the clear: $ID_M, TID_M, DATE, NONCE_M, H(Common), [H(V)]$
SLIP	Payment instructions: $PRICE, H(Common), CAN, R_C, [PIN]$
EncSlip	Payment instruction encrypted with the public key of the acquirer: $PK_A (SLIP)$
$CERT_X$	Public-key certificate of X, issued by a CA
Sig_A	Acquirer's signature: $SK_A[H(Y/N, H(Common))]$
Sig_M	Merchant's signature in Auth-Request: $SK_M[H(H(Common), [H(V)])]$
Sig_C	Cardholder's signature: $SK_C[H(EncSlip, H(Common))]$

- El **protocolo 1KP** es el más básico. Sólo el Acquirer necesita poseer (y distribuir) su certificado de clave pública $CERT_A$.
- La información de partida de cada una de las entidades es:

Actor	Information Items
Customer	DESC, CAN, PK_{CA}, [PIN], $CERT_A$
Merchant	DESC, PK_{CA}, $CERT_A$
Acquirer	SK_A, $CERT_A$

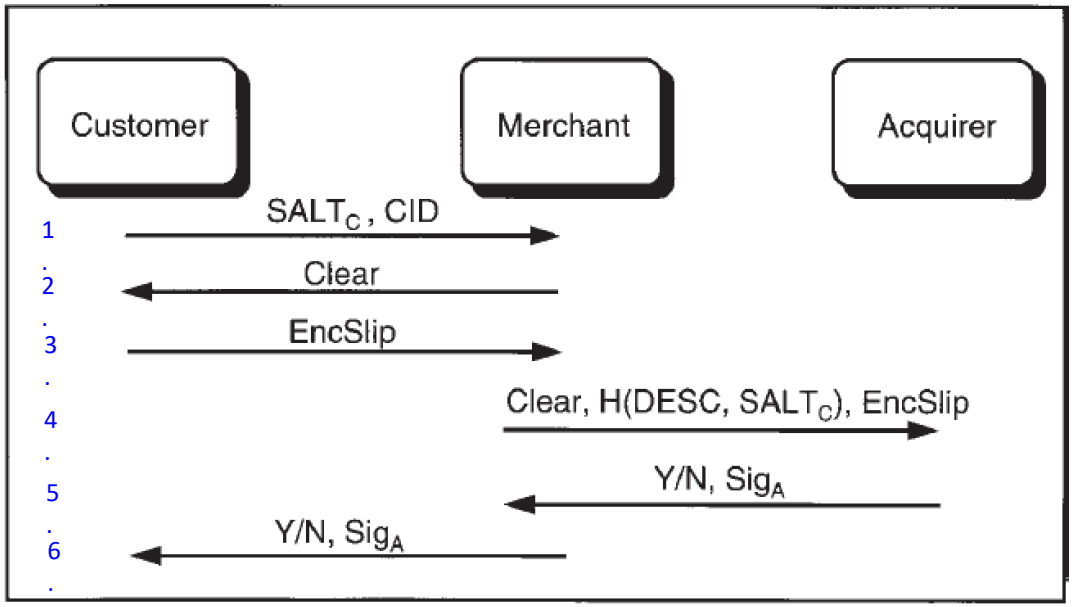
Item	Description
CAN	Customer's account number (e.g., credit card number)
ID_M	Merchant ID; identifies merchant to acquirer
TID_M	Transaction ID; uniquely identifies the transaction
DESC	Description of the goods; includes payment information such as credit card holder's name and bank identification number
$SALT_C$	Random number generated by C; used to randomize DESC and thus ensure privacy of DESC on the M to A link
$NONCE_M$	Random number generated by a merchant to protect against replay
DATE	Merchant's current date/time
PIN	Customer's PIN which, if present, can be optionally used in 1KP to enhance security
Y/N	Response from card issuer; Yes/No or authorization code
R_C	Random number chosen by C to form CID
CID	A customer pseudo-ID which uniquely identifies C; computed as $CID = H(R_C, CAN)$
V	Random number generated in 2KP and 3KP by merchant; used to bind the Confirm and Invoice message flows

- El **protocolo 1KP** es el más básico. Sólo el Acquirer necesita poseer (y distribuir) su certificado de clave pública $CERT_A$.
- La información de partida de cada una de las entidades es:

Actor	Information Items
Customer	DESC, CAN, PK_{CA}, [PIN], $CERT_A$
Merchant	DESC, PK_{CA}, $CERT_A$
Acquirer	SK_A, $CERT_A$

- Los pasos del protocolo son:

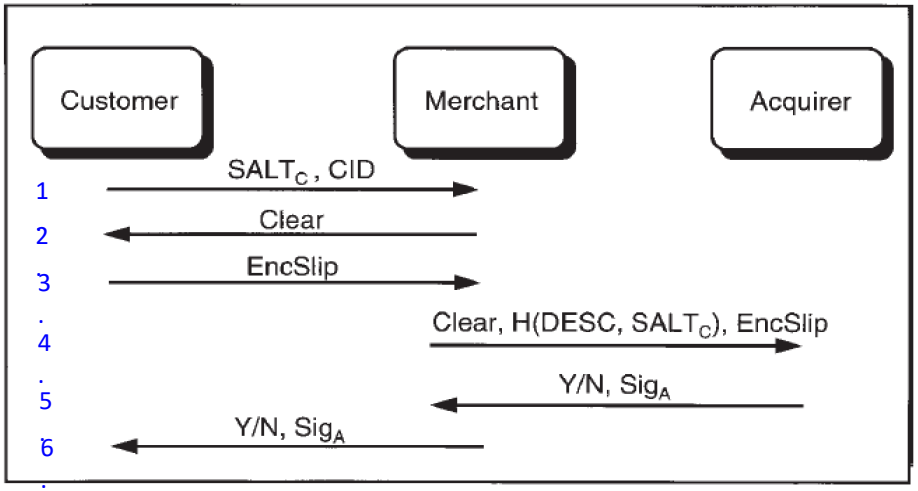
Item	Description
Common	Information held in common by all parties: PRICE, ID_M, TID_M, DATE, $NONCE_M$, CID, H(DESC, $SALT_C$), [H(V)]
Clear	Information transmitted in the clear: ID_M, TID_M, DATE, $NONCE_M$, H(Common), [H(V)]
SLIP	Payment instructions: PRICE, H(Common), CAN, R_C, [PIN]
EncSlip	Payment instruction encrypted with the public key of the acquirer: PK_A (SLIP)
$CERT_X$	Public-key certificate of X, issued by a CA
Sig_A	Acquirer's signature: SK_A[H(Y/N, H(Common))]
Sig_M	Merchant's signature in Auth-Request: SK_M[H(H(Common), [H(V)])]
Sig_C	Cardholder's signature: SK_C[H(EncSlip, H(Common))]



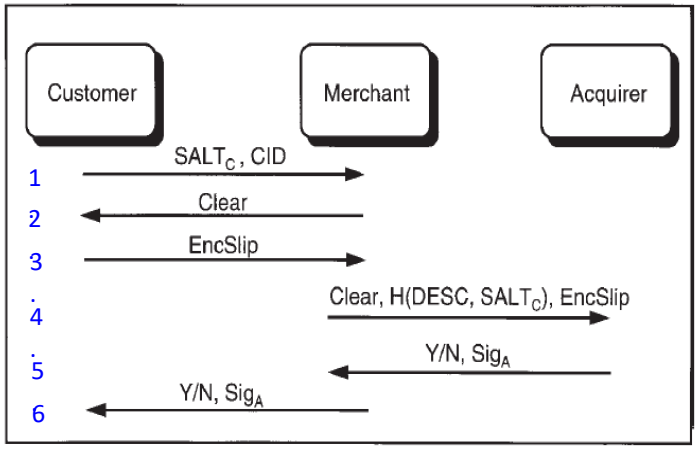
- Ejercicio 3:

Item	Description
CAN	Customer's account number (e.g., credit card number)
ID_M	Merchant ID; identifies merchant to acquirer
TID_M	Transaction ID; uniquely i
DESC	Description of the goods; i holder's name and bank i
$SALT_C$	Random number generate privacy of DESC on the M
$NONCE_M$	Random number generate
DATE	Merchant's current date/t
PIN	Customer's PIN which, if enhance security
Y/N	Response from card issue
R_C	Random number chosen l
CID	A customer pseudo-ID wh CID = $H(R_C, CAN)$
V	Random number generate Confirm and Invoice mess

Item	Description
Common	Information held in common by all parties: PRICE, ID_M, TID_M, DATE, $NONCE_M$, CID, H(DESC, $SALT_C$), [H(V)]
Clear	Information transmitted in the clear: ID_M, TID_M, DATE, $NONCE_M$, H(Common), [H(V)]
SLIP	Payment instructions: PRICE, H(Common), CAN, R_C, [PIN]
EncSlip	Payment instruction encrypted with the public key of the acquirer: PK_A (SLIP)
$CERT_X$	Public-key certificate of X, issued by a CA
Sig_A	Acquirer's signature: SK_A[H(Y/N, H(Common))]
Sig_M	Merchant's signature in Auth-Request: SK_M[H(H(Common), [H(V)])]
Sig_C	Cardholder's signature: SK_C[H(EncSlip, H(Common))]



- Pasos en detalle:



1. $SALT_C$, $H(R_C, CAN)$

2. ID_M, TID_M, DATE, $NONCE_M$, $H(PRICE, ID_M, TID_M, DATE, NONCE_M, H(R_C , CAN), H(DESC, SALT_C))$

3. P_{KA} $(PRICE, H(PRICE, ID_M, TID_M, DATE, NONCE_M, H(R_C, CAN), H(DESC, SALT_C)), CAN, R_C, [PIN])$

4. ID_M, TID_M, DATE, $NONCE_M$, $H(PRICE, ID_M, TID_M, DATE, NONCE_M, H(R_C , CAN), H(DESC, SALT_C))$, $H(DESC, SALT_C)$,
   P_{KA} $(PRICE, H(PRICE, ID_M, TID_M, DATE, NONCE_M, H(R_C, CAN), H(DESC, SALT_C)), CAN, R_C, [PIN])$

5. $Y/N, Sig_A$

6. $Y/N, Sig_A$

Item	Description
CAN	Customer's account number (e.g., credit card number)
ID_M	Merchant ID; identifies merchant to acquirer
TID_M	Transaction ID; uniquely identifies the transaction
DESC	Description of the goods; includes payment information such as credit card holder's name and bank identification number
$SALT_C$	Random number generated by C; used to randomize DESC and thus ensure privacy of DESC on the M to A link
$NONCE_M$	Random number generated by a merchant to protect against replay
DATE	Merchant's current date/time
PIN	Customer's PIN which, if present, can be optionally used in 1KP to enhance security
Y/N	Response from card issuer; Yes/No or authorization code
R_C	Random number chosen by C to form CID
CID	A customer pseudo-ID which uniquely identifies C; computed as $CID = H(R_C, CAN)$
V	Random number generated in 2KP and 3KP by merchant; used to bind the Confirm and Invoice message flows

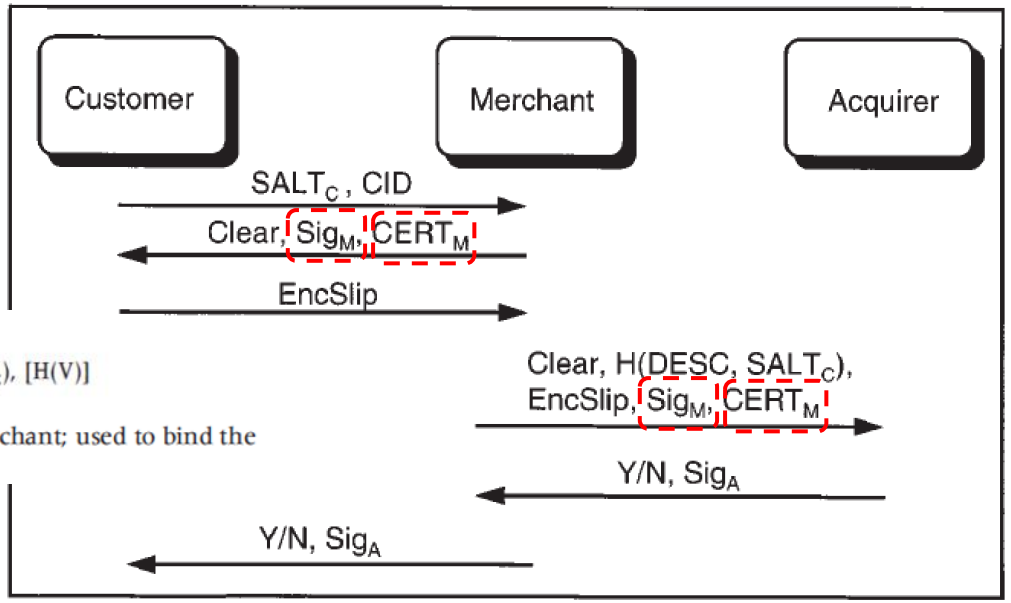
Item	Description
Common	Information held in common by all parties: PRICE, ID_M, TID_M, DATE, $NONCE_M$, CID, $H(DESC, SALT_C)$, [H(V)]
Clear	Information transmitted in the clear: ID_M, TID_M, DATE, $NONCE_M$, H(Common), [H(V)]
SLIP	Payment instructions: PRICE, H(Common), CAN, R_C, [PIN]
EncSlip	Payment instruction encrypted with the public key of the acquirer: PK_A (SLIP)
$CERT_X$	Public-key certificate of X, issued by a CA
Sig_A	Acquirer's signature: $SK_A[H(Y/N, H(Common))]$
Sig_M	Merchant's signature in Auth-Request: $SK_M[H(H(Common), [H(V)])]$
Sig_C	Cardholder's signature: $SK_C[H(EncSlip, H(Common))]$

- Las desventajas de uso de 1KP son:
  - El cliente se autentica utilizando sólo un número de tarjeta de crédito y, opcionalmente, un PIN, en lugar de firmas digitales
  - El vendedor no se autentica ni ante el cliente ni ante el Acquirer
  - Ni el vendedor ni el cliente proporcionan evidencias de intervención en la transacción

- En el **protocolo 2KP**, además del Acquirer, cada vendedor necesita tener un par *<clave pública, clave privada>*, y está obligado a distribuir su certificado $CERT_M$ al cliente y al Acquirer
- La información de partida de cada una de las entidades es:

Actor	Information Items
Customer	DESC, CAN, PK_{CA}, $CERT_A$
Merchant	DESC, PK_{CA}, $CERT_A$, SK_M, $CERT_M$
Acquirer	PK_{CA}, SK_A, $CERT_A$

- Los pasos del protocolo son:



Sig_M	Merchant's signature in Auth-Request: $SK_M[H(H(Common), [H(V)])]$
Common	Information held in common by all parties: PRICE, ID_M, TID_M, DATE, $NONCE_M$, CID, $H(DESC, SALT_C)$, $[H(V)]$
V	Random number generated in 2KP and 3KP by merchant; used to bind the Confirm and Invoice message flows

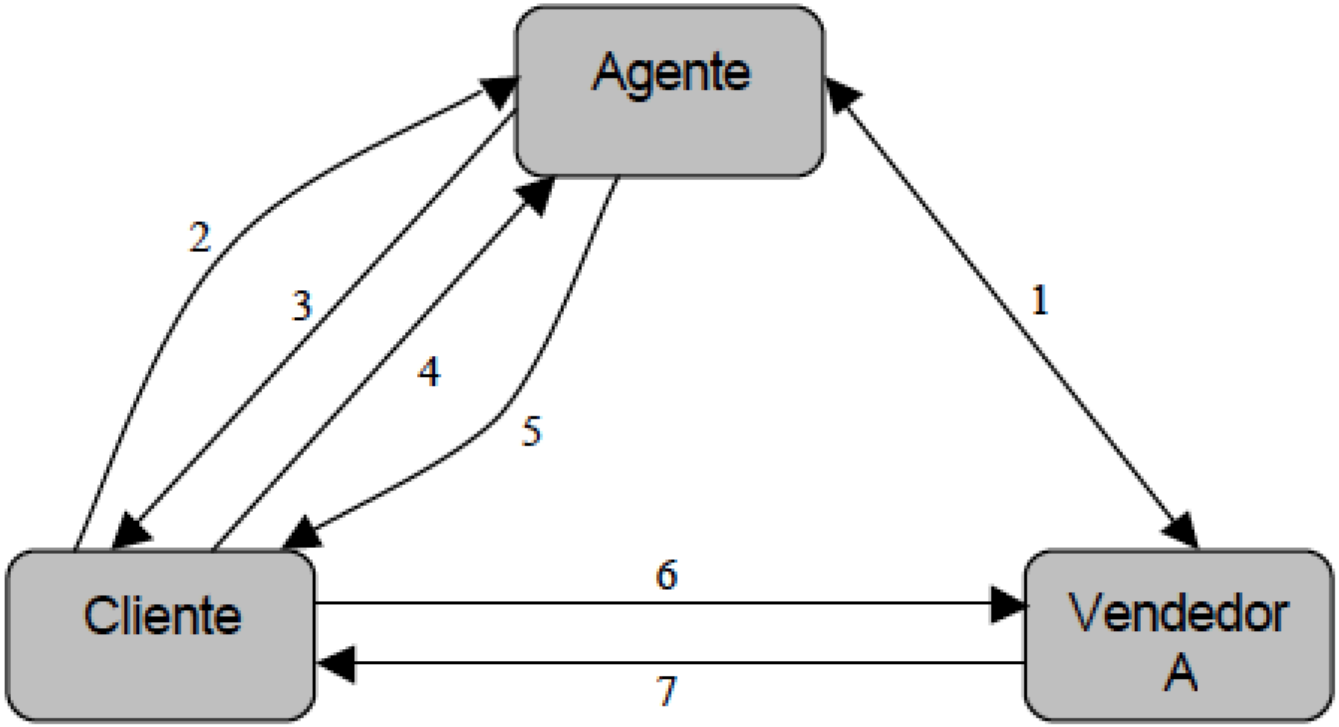
Protocolo Millicent

- En algunos escenarios hay que transferir una cantidad muy pequeña (**micropago**), y por ello, hay que buscar la forma más eficiente y económica posible de hacerlo
  - minimizando el tráfico y los recursos utilizados, para que los costes de realizar el pago sean mínimos en comparación el pago en sí mismo

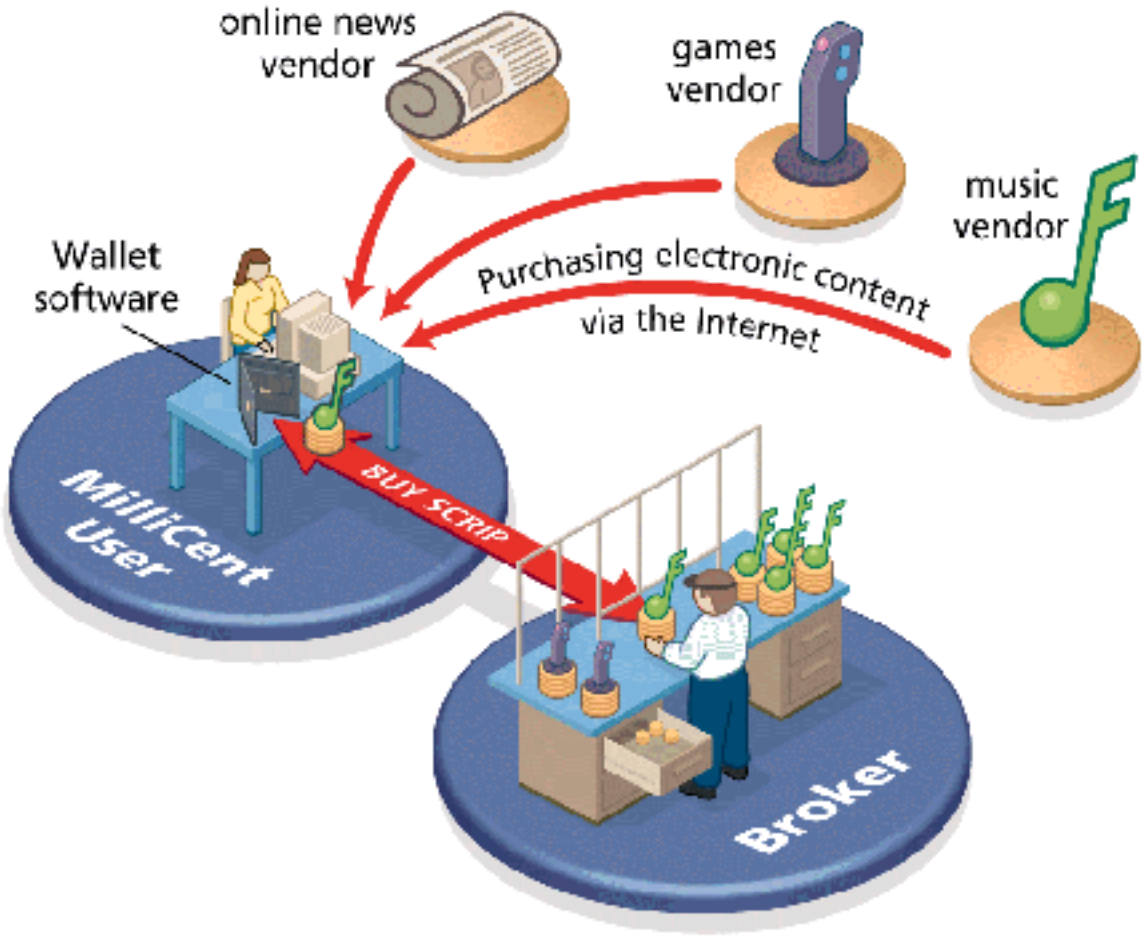
- Para reducir los costes operacionales se utilizan varias soluciones:
  - servicios de prepago
  - autorizaciones off-line
  - agrupación de la facturación de los micropagos en lotes
  - reducción del coste computacional
    - la criptografía de clave pública no es la más adecuada pues resulta cara, e incluso, los criptosistemas simétricos pueden ser cuestionables
      - cada vez más se aplica el empleo de **funciones hash**
    - sin embargo, conlleva también la imposibilidad de proporcionar servicios de no-repudio

- Un ejemplo es **Millicent** que utiliza cifrado simétrico, y no utiliza procesamiento on-line
  - Además de clientes y comerciantes, en Millicent existe la figura del **agente de negocios** (posiblemente una institución financiera)
  - El sistema utiliza una forma de moneda electrónica, el **scrip**
    - los scrips vienen a ser "cupones electrónicos" que representan dinero, con los que el comprador obtiene la mercancía del vendedor

- Para un cliente no sería eficiente comprar lotes de scrips a cada uno de los potenciales vendedores del sistema
  - se puede suponer que, durante un periodo, las compras de un cliente a varios comerciantes alcanzarán un importe equivalente a un macropago
  - la función principal del agente de negocios es la de vender a cada cliente, y dentro de un mismo lote mixto, scrips de distintos vendedores



1. Compra-Venta de scrips de A
2. Compra de scrips de agente (macropago)
3. Envío de scrips de agente
4. Compra de scrips de A (micropago mediante scrips de agente
5. Envío de scrips de A
6. Petición producto + micropago mediante scrips de A
7. Envío de producto



Fuente: http://magsastre.eresmas.com/3-6comer.html

- El modelo a tres bandas (cliente, vendedor y bróker) ayuda a tener cierto grado de anonimato por parte del comprador:
  - el agente conoce la identidad del comprador y su número de tarjeta de crédito, pero nunca llega a conocer qué producto compra
  - el vendedor sabe lo que el cliente compra, pero desconoce su identidad

- Otros sistemas de micropago son:
  - Subscrip
  - Kleline
  - Flattr
  - M-coin
  - etc.