Topic 3.1: NetBill

Outline

- Why is NetBill developed?
- NetBill Transaction Model
- NetBill Transaction Protocol
  - Basic Protocol
  - Optimizations for zero-priced goods
- Failure Analysis

E-Commerce over the Internet

- Internet is attractive for e-commerce
  - Search for suppliers
  - Price negotiation
  - Ordering
  - Payment for goods
  - Delivery of information goods
    - Software, electronic books, etc.
- Challenges
  - No easily identifiable places of business
  - Transactions are subject to observation by their parties
  - Privacy
NetBill

- NetBill is a system developed to facilitate selling and delivery of low-priced information goods over the Internet.
  - Maintain accounts for customers as well as merchants, which are linked to banks
  - Transfer information goods from merchant to customer
  - Transfer money from customer’s account to merchant’s account.
  - Combine small transactions into larger conventional transactions, reducing transaction cost.

NetBill Transaction Model

- Three phases
  - Phase 1: Price negotiation
  - Phase 2: Goods delivery
  - Phase 3: Payment

NetBill Transaction Objectives

- Only authorized customers can charge against a NetBill account
- The customer and merchant must agree on the purchase item and the price
- A customer can optionally protect her identity from merchants
- Customers and merchants are provided with proof of transaction results from NetBill
- There is a negotiation phase between customer and merchant
- A customer may present credentials identifying her for special treatment
- A customer receives the goods if and only if she is charged for the goods
- A customer may need approval from a fourth party before the NetBill server will allow a transaction.
- The privacy and integrity of communications is protected from observation or alteration by external parties.
NetBill Transaction Protocol

• The basic protocol
  – Phase 1: price negotiation
    • C ⇒ M: price request
    • M ⇒ C: price quote
  – Phase 2: goods delivery
    • C ⇒ M: goods request
    • M ⇒ C: goods, encrypted with a key K
  – Phase 3: payment
    • C ⇒ M: signed electronic payment order (EPO)
    • M ⇒ N: endorsed EPO (including K)
    • N ⇒ M: signed result (including K)
    • M ⇒ C: signed result (including K)

Notations

• $T_{CM}(Id)$: Kerberos ticket proving to Y that X is named by Id, and establish a session key $XY$ shared between them.
• $CC(M)$: cryptographic checksum of M.
• $E_K(M)$: M encrypted using key K.
• $E_{X-PUB}(M)$: M encrypted using X’s RSA public key.
• $E_{X-PR}(M)$: M signed using X’s RSA private key.
• $[M]_X$: M signed (with RSA) and timestamped by X.
• $[M]_{X-DSA}$: M signed and timestamped by X with DSA.
• ${M}_X$: M encrypted for X using RSA.

The Price Request Phase

1. C ⇒ M: $T_{CM}(Id)$, $E_{CM}$(Credentials, PRD, Bid, RequestFlags, TID)
2. M ⇒ C: $E_{XM}$(ProductID, Price, RequestFlags, TID)

• $T_{CM}(Id)$: prove the identity of the customer
• Credentials: establish the customer’s membership
• PRD: product description
• RequestFlags:
  – Message 1: request for the disposition of the transaction (e.g., Delivery method)
  – Message 2: merchant’s response to customer’s request
• TID:
  – Message 1: if this is a repeated request
  – Message 2: if this is not supplied by the customer
The Goods Delivery Phase

3. \( C \Rightarrow M: T_{\text{Md}}(\text{Id}), E_{\text{Md}}(\text{TID}) \)
4. \( M \Rightarrow C: E_{\text{K}}(\text{Goods}), E_{\text{Md}}(\text{CC}(E_{\text{K}}(\text{Goods})), \text{EPOID}) \)

- M sends to C
  - An encrypted version of the goods
  - The cryptographic checksum of the encrypted goods
  - EPOID: electronic purchase order ID.
    - Merchant ID + a timestamp (delivery time) + a serial number
- Intuition:
  - Reduce the transaction to a fair exchange of K and the payment from C.
  - This fair exchange depends on the NetBill server.

The Payment Phase

5. \( C \Rightarrow M: T_{\text{Md}}(\text{Id}), E_{\text{Md}}([\text{EPO}]_{C}) \)

- EPO consists of
  - Clear part:
    - C’s ID, Product ID, Price, M’s ID
    - CC(E_{K}(\text{Goods})), CC(\text{PRD}), CC(\text{C acct}, \text{AcctVN})
    - EPOID
  - Encrypted part:
    - \( T_{\text{M}}(\text{TrueID}) \)
    - \( E_{\text{CN}}(\text{Authorization, C acct, AcctVN, Cmemo}) \)

The Payment Phase (Cont’d)

6. \( M \Rightarrow N: T_{\text{M}}(\text{M}), E_{\text{Md}}([\text{EPO}]_{C}, \text{M acct}, \text{MMemo}, \text{K}_{C}) \)

- The merchant endorse and submit the EPO
  - M acct: Merchant’s NetBill account
  - MMemo: merchant’s memo field
  - K: the key used to deliver the goods
- Point of no return
  - The merchant cannot reverse the transaction.
The Payment Phase (Cont’d)

7. $N \Rightarrow M: E_{MN}(\{\text{Receipt}\}, E_{CN}(\text{EPOID}, \text{CAcct}, \text{Bal}, \text{Flags}))$

- The NetBill server makes decision based on verification of
  - The signatures
  - Privileges of the users involved
  - Customer’s account balance
  - Uniqueness and freshness of the EPOID

- Receipt
  - Result, Identity, Price, ProductID, M, K, EPOID
  - The signed receipt certifies the transaction

The Payment Phase (Cont’d)

8. $M \Rightarrow C: E_{CM}(\{\text{Receipt}\}, E_{CN}(\text{EPOID}, \text{CAcct}, \text{Bal}, \text{Flags}))$

- Merchant forwards NetBill server’s response to customer
  - M needs to decrypt and re-encrypt

Status Query Exchange

- Needed when there is communication failure

  The merchant requests the transaction status from NetBill

  1. $M \Rightarrow N: T_{MN}(\text{Id}, E_{MN}(\text{EPOID}))$
  2. $N \Rightarrow M: E_{MN}(\{\text{Receipt}\}, E_{CN}(\text{EPOID}, \text{CAcct}, \text{Bal}, \text{Flags}))$

  The customer requests the transaction status from the merchant

  1. $C \Rightarrow M: T_{CM}(\text{Id}, E_{CM}(\text{EPOID}))$
  2. $M \Rightarrow C: E_{CM}(\{\text{Receipt}\}, E_{CN}(\text{EPOID}, \text{CAcct}, \text{Bal}, \text{Flags}))$
Status Query Exchange (Cont’d)

The customer requests the transaction status from NetBill

1. \( C \Rightarrow N: T_{CS}(\text{TrueId}), E_{CN}(\text{EPOID}) \)
2. \( N \Rightarrow C: E_{CN}([\text{Receipt}]_{\text{NetBill}}, E_{CN}(\text{EPOID}, \text{CAcct, Bal, Flags})) \)

The customer requests the transaction status from the merchant for a non-NetBill transaction

1. \( C \Rightarrow M: T_{CM}(\text{Id}), E_{CM}(\text{EPOID}) \)
2. \( M \Rightarrow C: E_{CM}(\text{Result, K}) \)

Zero-Priced Goods

- Protocol can be simplified
- Four variations
  - Type indicated in \( \text{RequestFlags} \) in the price request message
  - Zero-price certified delivery
  - Certified delivery without NetBill server
  - Verified delivery
  - Unverified delivery

Zero-Price Certified Delivery

1. \( C \Rightarrow M: T_{CS}(\text{Identn}, E_{CS}(\text{Credentials, PRD, Bid, RequestFlags, TID})) \)
2. \( M \Rightarrow C: E_{CS}(\text{ProductID, Price=0, RequestFlags, TID}, E_{CS}(\text{CAcct, EPOID})) \)
3. \( C \Rightarrow M: T_{CM}(\text{Identn}, E_{CM}(\text{EPOID})) \)
4. \( M \Rightarrow N: E_{CS}(\text{Receipt} || \text{FPA, CAcct, Bid, Flags}) \)
5. \( N \Rightarrow M: E_{CS}(\text{Receipt} || \text{FPA, CAcct, Bid, Flags}) \)

Price negotiation can be omitted.

But delivery must be certified by NetBill.
Certified Delivery without NetBill

1. \( C \rightarrow M \quad T_{CM}(Identity), E_{CM}(Credentials, PRD, Bid, RequestFlags, TID) \)

2/4. \( M \rightarrow C \quad E_{CM}(ProductID, Price=0), RequestFlags, TID, E_{C}(Goods), ECM(CCM(E_{C}(Goods), EPOID)) \)

5. \( C \rightarrow M \quad T_{CM}(Identity), E_{CM}(EPOID, CCE_{C}(Goods)) \)

8. \( M \rightarrow C \quad E_{CM}(Result, K) \)

- No need to go through NetBill.
- But C cannot recover if M decides not to send message 8.

Verified Delivery

1. \( C \rightarrow M \quad T_{CM}(Identity), E_{CM}(Credentials, PRD, Bid, RequestFlags, TID) \)

2/4. \( M \rightarrow C \quad E_{CM}(ProductID, Price=0), RequestFlags, TID, Goods, CC(Goods), EPOID \)

5. \( C \rightarrow M \quad T_{CM}(Identity), E_{CM}(EPOID, CC(Goods)) \)

8. \( M \rightarrow C \quad E_{CM}(Result) \)

- Goods is encrypted with shared session key.
- C doesn’t have to wait for K.

Unverified Delivery

1. \( C \rightarrow M \quad T_{CM}(Identity), E_{CM}(Credentials, PRD, Bid, RequestFlags, TID) \)

2/4. \( M \rightarrow C \quad E_{CM}(ProductID, Price=0), RequestFlags, TID, Goods, CC(Goods)) \)

- Eliminate the acknowledgement of goods delivery.
Failure Analysis

• Customer complaints
  – Incorrect or damaged goods
    • Can be resolved with the EPO, which contains a cryptographic checksum of the encrypted goods
    – Cannot deal with false advertisement
  – No decryption key
    • Can be resolved by a status query exchange with the NetBill server

Failure Analysis (Cont’d)

• Transaction dispute
  – Inconsistent price
    • Can be resolved by checking the EPO signed by the customer
  – Fraudulent transactions
    • Same resolution as above.

Failure Analysis (Cont’d)

• Merchant Complaints
  – Insufficient payment
    • Can be resolved by checking the receipt signed by NetBill
### Identification and Authentication

- **Public key based Kerberos**
  - Each entity has public/private key pair with a certificate for the public key
  - Public key certificate is used to obtain a Kerberos server ticket

1. \( C \rightarrow M \ [\{ \text{Identity, M, Timestamp, K} \}_M] \)
2. \( M \rightarrow C \ E_K(T_{CM} \{\text{Identity}, \text{CM}\}) \)

### Privacy protection

- **Pseudonym mechanism**
  - Implemented through a pseudonym-granting server \( P \).
  - Two methods
    - Per transaction
      - Use a unique pseudonym for each transaction
    - Per merchant
      - Use a unique pseudonym for each customer-merchant pair

### Authorization

1. \( C \rightarrow A \ T_{CA}(\text{Identity}), E_{CA}(M, \text{ProductID}, \text{Price}, \text{CC}(E_{K}(\text{Goods})), \text{EPOID, CAccct}) \)
2. \( A \rightarrow C \ E_{CA}(E_{A, PKG}(\text{CC}(\text{Identity, M, ProductID, Price, CC(E_{K}(\text{Goods}), EPOID, CAccct})) \)

- Performed through an access control server \( A \).
  - Message returned by \( A \) is used as the authorization token in an EPO.