INFS 767
Secure Electronic Commerce
Fall 2001

Lecture 6
Unlinkable Serial Transactions

Prof. Ravi Sandhu

CASE STUDY OF AN ELECTRONIC COMMERCE PROTOCOL

- **Unlinkable Serial Transactions: Protocols and Application**

  by

  Stubblebine, Syverson and Goldschlag
- **Paper in November 1999 issue of ACM Transactions on Information and System Security (TISSEC)**

CONFLICTING OBJECTIVES

- **Provider expects**
  - **to be paid**
  - **to make subscription cloning difficult**
  - **to terminate rogue subscriptions**
- **Customer expects**
  - **privacy of usage profile**

CHAUM'S SOLUTIONS (1981 on)

- **Payment is by anonymous e-cash**
- **Customers register using anonymous pseudonyms**

CHAUM'S SOLUTIONS (1981 on)

- **Cannot accommodate credit card payments**
- **Pseudonyms are a single point of anonymity failure**
- **Rogue customer can re-register using different pseudonym**

UST SOLUTION

- **customer may be known to vendor but behavior is untraceable**
- **independent of payment mechanism**
- **no single point of anonymity failure**
- **makes subscription cloning difficult**
- **does not allow multiple concurrent transactions from a single customer**

UST SOLUTION

- **A legitimate subscriber can set up a proxy pirate server**
  - **not a problem in small scale**
    - **requires technical skill**
    - **little motivation**
  - **on large scale**
    - **can use detect and prosecute strategy to deter**

BASIC IDEA

- **Customer gets a one-time unlinkable token**
- **Fresh token is generated after every use**

MAJOR CAVEAT

- **Cryptographic protocol cannot prevent against linkage through application data**
- **Trade-off between functionality and anonymity**

APPLICATIONS

- **Subscription to on-line information service**
- **pay-per-use**
- **pay-per-view**
- **multivendor packages (digital coupon books)**
- **anonymous proof of membership**

OPERATING ENVIRONMENT ASSUMPTIONS

A1. Anonymity protected network communications are unlinkable to prior communications provided application content does not enable linkage.

A2. Entities may collude. However, we assume that collusion among customers is insignificant in the sense that there will always be a sufficient number of non-colluding customers and associated transactions to mask legitimate customer activity.

OPERATING ENVIRONMENT ASSUMPTIONS

A3. We assume that cryptographic keys, nonces, blinding factors, etc. are adequately randomly chosen from an adequately large space to prevent random collisions or revealing of secrets by cryptanalytic attacks.

A4. We assume that keyed cryptographic operations prevent any undetectable modification of fields to which those operations are applied. Furthermore, we assume the inability of an entity to forge signatures without knowledge of the key.

OPERATING ENVIRONMENT ASSUMPTIONS

A5. We assume that every message is received as sent after a finite number of attempts to send it.

A6. The vendor will provide services for which he accepts payment.

FRAUD REQUIREMENTS

R1. Eliminate high volume fraud

R2. Detect, and reduce activity related to low volume fraud.

R3. Payments (including refunds if applicable) cannot be stolen.

CUSTOMER PRIVACY REQUIREMENTS

R4. Protect the identity of the customer in a transaction from vendors, other customers, and outsiders.

R5. Prevent the building of customer profiles (including pseudonymous profiles) by vendors, other customers, and outsiders.

SERVICE GUARANTEE REQUIREMENTS

R6. Customers cannot be denied service for which they contracted.

NOTATION

$[X]_K$: Message integrity of X using K

$\{X\}_K$: Message integrity and confidentiality of X using K

$\underline{X}$: Blinding of X

h(X): Hash of X

BLINDING

- $\underline{X}$ is computed from X using a secret blinding factor by Alice
- $[\ \underline{X}\ ]_B$ is signed by Bob's private key
- Alice removes blinding factor by use of secret to get $[X]_B$
- Without secret blinding factor cannot associate $\underline{X}$ with X or $[X]_B$

UST REGISTRATION

M1. C -> V: {Payment, K_{CV}}$_V$,

[Request for certificate of type S, C, $\underline{h(N1)}$]K_{CV}

M2. V -> C: [$\underline{h(N1)}$]$_S$

M3. C -> V: [Ack] K_{CV}

UST REGISTRATION

- **Multiple certificates are required if customer needs to access from multiple machines**
- **Alternately customer can use single certificate at some web page through which all access is proxied**
  - **customer must trust web page host**
- **Customer authentication is not part of this protocol**

CERTIFICATE REDEMPTION

M1. C -> V: { [h(Ni)]$_S$, Ni, K_{CV} }$_V$,

[Request for transaction of type S, $\underline{h(Ni+1)}$]K_{CV}

M2. V -> C: [Approved OR Not Approved]K_{CV}

M3. C <-> V: [Transaction]K_{CV}

M4. V -> C: [$\underline{h(Ni+1)}$]$_S$

M5. C -> V: [Ack]K_{CV}

CERTIFICATE REDEMPTION

- **New certificate is obtained after transaction ends**
  - **makes it harder for subscriber to run proxy subscription service**
- **K_{CV} is used for integrity protection in protocol**
  - **can optionally be used for confidentiality protection**
  - **C has incentive to choose it to be unique**
  - **use of K_{CV} has to be serialized within a transaction otherwise sharing of K_{CV} can lead to concurrent use by sharing K_{CV} instead of sharing certificate**

NOT APPROVED

- **[Not Approved]K_{CV} is sent only if nonce does not match signed certificate**
- **other Not Approved errors sent without K_{CV}**
- **makes protocol fail-stop**

SUBSCRIPTION TERMINATION

M1. C -> V: { [h(Ni)]$_S$, Ni, K_{CV}}$_V$,

[Request for transaction of type S terminate, C]K_{CV}

M2. V -> C: {Refund}K_{CV} OR

[Not approved]K_{CV}

M3. C -> V: [Ack]K_{CV}

SUBSCRIPTION TERMINATION

- May require multiple certificates to be returned
- refund can take any form: e-cash, credit card, paper check
- customer authentication (for refund) is not part of this protocol
- termination by vendor is not so easy: requires change of service key S

RECOVERY

- Broken connection
  - if protocol breaks too early can replay in entirety (except for transaction)
  - ack is assumed after suitable time-out
  - registration gets committed after ack (explicit or assumed)
- Disk crash
  - reinitialize
  - backup certificates

SERVICE KEYS

- Service key S should be well known to prevent selection on per-subscriber basis
- S can be changed to terminate subscriptions

SUBSCRIPTION SHARING

- Certificate sharing
- proxy server
- session sharing

UST WITH AUDIT (USTA) REGISTRATION

M1. C -> V: {Payment, I_{audit}, K_{CV}}$_V$,
[Request for certificate of type S, C, h(N1)]K_{CV}

M2. V -> C: [h(N1)]$_S$

M3. C -> V: [Ack] K_{CV}

USTA CERTIFICATE REDEMPTION

M1. C -> V: { [h(Ni)]$_S$, Ni, K_{CV} }$_V$,
[Request for transaction of type S,
h(Ni, I_{audit}, Salt), h(Ni+1)] K_{CV}

M2. V -> C: [Approved OR Not Approved OR Audit]K_{CV}

M3. C <-> V: [Transaction]K_{CV}

M4. V -> C: [h(Ni+1)]$_S$

M5. C -> V: [Ack]K_{CV}

AUDIT PROTOCOL

M1. C -> V: $\{ [h(Ni)]_S, Ni, K_{CV} \}_V$,
[Request for transaction of type S,
$h(Ni, I_{audit}, Salt)$, $\underline{h(Ni+1)}$]K_{CV}

M2. V -> C: [Audit]K_{CV}

M3. C -> V: $\{C, Ni, I_{audit}, Salt\}$ K_{CV}

M4. V -> C: [$\underline{h(Ni+1)}$ $]_S$ OR
[Not approved]K_{CV}

M5. C -> V: [Ack]K_{CV}

AUDIT PROTOCOL

- **Must use K_{CV} and not V in message 3**

APPLICATIONS

- **Pay-per-use (digital tokens)**
- **third party subscription management**
- **multivendor packages**
- **membership and voting**

RELATED WORK

- **Digital cash**
- **Anonymity services**