
A FEW E-COMMERCE APPLICATIONS

CIS 400/628 — Spring 2005
Introduction to Cryptography

This is based on Chapter 9 of Trappe and Washington

E-COMMERCE: SET

SET = **S**ecure **E**lectronic **T**ransaction

Consider a credit card transaction over the net.

We want:

Authenticity

No impersonations. No forgeries.

Integrity

Documents cannot be altered after the fact.

Privacy

The details of the transaction should be private.

Security

Credit card numbers & the like must be protected.

SET - a collection of crypto protocols for credit card transactions (\approx 1997)

SET EXAMPLE

Characters

Cardholder }
Merchant } **These guys don't trust one another**
Bank }

H - a public hash function

PKC - say RSA

Encryption/Decryption Functions

E_C E_M E_B Public Private D_C D_M D_B

Cardholder

GSO - goods and services order (cardholder's name, merchant's name, prices, etc.)

PI - payment instructions (merchant's name, card number, total price, etc.)

SET EXAMPLE CONTINUED

Q: Who knows what from the transactions?

Cardholder

Computes

$$\begin{aligned} \text{GSO}_{md} &= H(E_M(\text{GSO})) & \text{PI}_{md} &= H(E_B(\text{PI})) \\ \text{PO}_{md} &= H(\text{PI}_{md} || \text{GSO}_{md}) & \text{DS} &= D_C(\text{PO}_{md}) \end{aligned}$$

Sends $E_M(\text{GSO}), \text{DS}, \text{PI}_{md}, E_B(\text{PI})$ to Merchant

Merchant (on receiving $E_M(\text{GSO}), \text{DS}, \text{PI}_{md}, E_B(\text{PI})$)

Computes

$$\begin{aligned} gso_{md} &= H(E_M(\text{GSO})) & gso &= D_M(E_M(\text{GSO})) \\ b &= H(\text{PI}_{md} || gso_{md}) & c &= E_C(\text{DS}) \end{aligned}$$

Checks

$$gso_{md} \stackrel{?}{=} H(E_M(gso)) \ \& \ b \stackrel{?}{=} c$$

Sends $\text{GSO}_{md}, E_B(\text{PI}), \text{DS}$ to the Bank

SET EXAMPLE, CONTINUED

Bank (on receiving GSO_{md} , $E_B(\text{PI})$, DS)

Computes

$$\begin{aligned} pi_{md} &= H(E_B(\text{PI})) & c &= E_C(\text{DS}) \\ pi &= D_B(E_B(\text{PI})) & b &= H(pi_{md} | GSO_{md}) \end{aligned}$$

Checks

$$b \stackrel{?}{=} c$$

Sends

E_M (a payment auth. + signature) to Merchant.

Merchant

Sends

E_C (receipt + signature) to Cardholder.

DIGITAL CASH

Okamoto & Ohta's Criteria

- ▶ Cash can be sent securely through computer networks
- ▶ Cash cannot be copied or reused.
- ▶ The spender can remain anonymous
 - neither the merchant nor the bank can identify the spender
- ▶ The transactions can be done off-line
 - the bank does not have to be involved.
- ▶ Cash can be transferred to others.
- ▶ Cash can be divided into smaller amounts.

BRANDS'S DIGITAL CASH SCHEME

Characters

- ▶ Bank
- ▶ Spender
- ▶ Merchant
- ▶ Central Authority
- ▶ Eve L. Dewar

Central Authority Chooses:

- ▶ A prime $p \ni q = (p - 1)/2$ is also prime.
- ▶ α – a primitive element of \mathbb{Z}_p^* .
- ▶ $g = \alpha^2 \pmod{p}$.
(So: $g^{e_1} \equiv g^{e_2} \pmod{p} \iff e_1 \equiv e_2 \pmod{q}$)
- ▶ $e_1, e_2 \in \mathbb{Z}_{p-1}^*$ – secret exponents.
- ▶ $g_1 = g^{e_1}$ and $g_2 = g^{e_2}$.
- ▶ $H: \mathbb{Z}^5 \rightarrow \mathbb{Z}_q$ and $H_0: \mathbb{Z}^4 \rightarrow \mathbb{Z}_q$. Hash functions

Public: $p, q, g, g_1, g_2, H,$ and H_0 .

Private: e_1 and e_2

BRANDS'S DIGITAL CASH: THE SETUP CONTINUED

The Bank

Chooses $x \stackrel{\text{ran}}{\infty} \mathbb{Z}_q$ – The bank's private ID

Computes $h = g^x$, $h_1 = g_1^x$, and $h_2 = g_2^x$. (All $(\text{mod } p)$)
 h, h_1, h_2 – the bank's public ID

The Spender

Chooses $u \stackrel{\text{ran}}{\infty} \mathbb{Z}_q$ – The spender's private ID

Computes $I = g_1^u \pmod{p}$ & sends I to the bank.

The Bank

Saves I + info on the spender

Computes $z' = (I g_2)^x \pmod{p}$ and sends z' to the spender.

The Merchant

Chooses an ID number M and sends it to the bank.

CREATING A COIN

Coin $\equiv (A, B, z, a, b, r) \in \mathbb{Z}^6$ In number theory we trust.

Spender

Asks bank for a coin and sends ID I .

Bank

Chooses: $w \stackrel{\text{ran}}{\in} \mathbb{Z}_q$ and computes:
$$\left. \begin{aligned} g_w &\equiv g^w \\ \beta &\equiv (Ig_2)^w \end{aligned} \right\} \pmod{p}$$

Sends g_w and β to the spender.

Spender

Chooses $(s, x_1, x_2, \alpha_1, \alpha_2) \stackrel{\text{ran}}{\in} \mathbb{Z}^5$

Computes:

$$\left. \begin{aligned} A &\equiv (Ig_2)^s & B &\equiv g_1^{x_1} g_2^{x_2} & z &\equiv (z')^s \\ a &\equiv g_w^{\alpha_1} g^{\alpha_2} & b &\equiv \beta^{s\alpha_1} A^{\alpha_2} \end{aligned} \right\} \pmod{p}$$

$A=1$ not allowed.

CREATING A COIN, CONTINUED

Spender – continued

Computes $c \equiv \alpha_1^{-1} \cdot H(A, B, z, a, b) \pmod{q}$.

Sends c to the bank.

Bank

Computes $c_1 \equiv (c \cdot x + w) \pmod{q}$.

Sends c_1 to the spender.

Spender

Computes $r \equiv (\alpha_1 c_1 + \alpha_2) \pmod{q}$.

The coin (A, B, z, a, b, r) is complete.

The amount of the coin is removed from
the spender's bank account.

SPENDING THE COIN

Spender

Gives the coin (A, B, z, a, b, r) to the merchant.

Merchant

Checks: $g^r \equiv ah^{H(A,B,z,a,b)}$
 $A^r \equiv z^{H(A,B,z,a,b)}b \pmod{q}$ (Check on board)

Computes $d = H_0(A, B, M, t)$, where t = a time stamp.

Sends d to spender.

Spender

Computes $\left. \begin{array}{l} r_1 \equiv d \cdot u \cdot s + x_1 \\ r_2 \equiv d \cdot s + x_2 \end{array} \right\} \pmod{q}$

Sends r_1 and r_2 to merchant.

Merchant

Checks: $g_1^{r_1} \cdot g_2^{r_2} \equiv A^d \cdot B \pmod{p}$ (Check on board)

Accepts the coin **iff** this holds.

DEPOSITING THE COIN IN THE BANK

Merchant

Sends (A, B, z, a, b, r) and (r_1, r_2, d) to the bank.

Bank

Checks that the coin has not yet be deposited. **Fraud control**
(If it has, call the cops.)

Checks that

$$\left. \begin{aligned} g^r &\equiv a \cdot h^{H(A,B,z,a,b)} \\ A^r &\equiv z^{H(A,B,z,a,b)} \cdot b \\ g_1^{r_1} \cdot g_2^{r_2} &\equiv A^d \cdot B \end{aligned} \right\} \pmod{p}$$

Accepts the coin **iff** these check out.

FRAUD CONTROL: I

The spender tries to spend the same coin
with the merchant and the vendor.

Merchant

Sends the coin and (r_1, r_2, d) to the bank.

Vendor

Sends the coin and (r'_1, r'_2, d') to the bank.

Bank

Since

$$r_1 - r'_1 \equiv us(d - d') \pmod{q}$$

$$r_2 - r'_2 \equiv s(d - d') \pmod{q}$$

we have

$$u \equiv (r_1 - r'_1)(r_2 - r'_2)^{-1} \pmod{q}$$

$$I \equiv g_1^u \pmod{q}$$

↑ the ID of the spender

FRAUD CONTROL: II

The merchant tries to deposit the same coin twice

Once with (r_1, r_2, d) ← legit

Once with (r'_1, r'_2, d') ← forged

This is hard to do.

The merchant has to produce r'_1 , r'_2 , and d' \ni

$$g_1^{r'_1} \cdot g_2^{r'_2} \equiv A^{d'} \cdot B \pmod{p}.$$

FRAUD CONTROL: III

Someone tries to make an unauthorized coin

This requires finding numbers \ni

$$\left. \begin{array}{l} g^r \equiv a \cdot h^{H(A,B,z,a,b)} \\ A^r \equiv z^{H(A,B,z,a,b)} \cdot b \end{array} \right\} \pmod{p} \left. \vphantom{\begin{array}{l} g^r \\ A^r \end{array}} \right\} \begin{array}{l} \text{Discrete logs} \\ \text{and worse!} \end{array}$$

Eve L. Dewer dot com receives a coin from the spender
and tries to spend the coin with the merchant

Merchant

Computes d' for Eve, which is unlikely to equal d .

Etc. see text

ANONYMITY

The spender

never needs to show the merchant an ID.

The bank

never sees the values of A, B, z, a, b, r until the coin is deposited.

the bank and the merchant

cannot figure out the ID of the spender
unless there is double spending.

See text for fuller details