6.915 Computer and Network Security **Fall Term, 1995**

Lecture 21 : November 21, 1995

Lecturer: Ron Rivest *Scribe: Christiana Toutet*

1 Outline

- Finish Digital Cash
- Pseudonyms

2 Digital Cash

The coin protocols we have looked at only work with single denomination coins. However, it would be nice to give the user the flexibility of choosing a denomination. One idea for doing this is to use a tree representation for coins. A user would buy a whole tree and could then spend pieces of the tree.

Example of a one dollar tree:

To spend a piece of the tree the user would send a signature on the root and the portion of the tree he wishes to spend. Notice that we still have not solved the usual problems associated with double spending. One way to avoid double spending would be to disallow spending coins that are ancestors of each other. A user can spend at most $1 without violating this condition. If a user spends two coins that are ancestors of each other, we could imagine a system which breaks the user's anonymity. A weakness of this scheme is that a user's spending can be correlated (linkability) since there is a way to tell which coins (or nodes of the tree) are ancestors of each other.

3 Digital Pseudonyms

3.1 The Idea Behind Digital Pseudonyms

The topic of digital pseudonyms is part of the larger theme of digital cash anonymity which Chaum is a major proponent of. The idea is that you want a different identity, pseudonym, for each organization you deal with and the orginizations you deal with should not be able to make a link between your different identities. Some common fragments of identity we usually give organizations are our address, social security number, name, date, and place of birth, etc.

In which cases would it be legitimate to correlate our identities?

- IRS to verify wages
- for loans; credit checks. You normally need to release info. Chaum believes that it should be your decision, but what about negative information?
- security clearance
- criminal records? What if you commit a crime under a pseudonym and never use it again?

3.2 What is a Pseudonym?

A random number used to deal with a particular organization, or a public key also used to deal with a particular organization are valid implementations of pseudonyms.

3.3 Establishing a pseudonym with an organization X and getting a credential from organization X to transact with organization Y

Each person should be able to establish only one pseudonym per organization. Let's say we have the "is-a-person" organization, Z, which does not allow one to have more than one pseudonym. Any other organization X would only accept a credential from the "is-a-person" organization before transacting with them. Our goal is to be able to get a pseudonym for organization X without revealing our "is-a-person" organization pseudonym. We also don't want the "is-a-person" organization to know that we are getting a pseudonym for organization X because then the "is-a-peson" organization knows wich organizations we are dealing with.

What would the credential from the "is-a-person" organization look like? $\{``p_1\ is\ a\ person"\}_{is-a-person-org}$. This is not quite right since it gives away the pseudonym p_1.

Our other goal is to get a credential from an organization X to transact with Y without X knowing that the credential it is giving us is for transacting with Y. For example getting a credential from the birth cetificate organization saying that you are over 21 so that you can purchase liquor from a liquor store Y.

Description of the protocol:

Let f be a publicly known one way function (i.e. f can be easily computed but its inverse is hard to compute).

Let c and $\overline{c}$ be respectively the public and private keys of organization X.

Let z and $\overline{z}$ be respectively the public and private keys of organization Z (the is-a-person-organization).

Let u be the pseudonym I have with organization Z.

$Me - \rightarrow X$

[1.1] $f(u \cdot s_x^c)^{\overline{z}}$

[1.2] $u \cdot s_x^c$

$Me \leftarrow - X$

[2] $u^{\overline{c}} \cdot s_x$

$Me - \rightarrow Y$

[3.1] $f(u \cdot s_y^c)^{\overline{z}}$

[3.2] $u \cdot s_y^c$

[3.3] $u^{\overline{c}} \cdot s_y$

[1.1]
s_x is a random number I have chosen. X can verify if $[1.1]^z = f([1.2])$ since f is a publicly known one way function and z is Z's public key.

[2]
I can verify that $[2]^c = [1.2]$

[3.1] and [3.2]
Y can check $[3.1]^z = f([3.2])$ since f is a publicly known one way function and z is Z's public key. $[3.3]^c = [3.2]$ since c is X's public key.

[3.3]
I can compute [3.3] since $[3.3] = [2]/(s_x \cdot s_y)$

Every time I want to transact with a different organization I have to go through Z. In order to get $f(u \cdot s_x^c)^{\overline{z}}$ to transact with X I need to go through a protocol with Z. In the same way, in order to get $f(u \cdot s_y^c)^{\overline{z}}$ to transact with Y I need to go through a protocol with Z.

Description of the protocol with Z:

1. I randomly create n pairs s'_i and t'_i for $i = 1 \ldots n$. I compute $s_i = f(s'_i)$ and $t_i = f(t'_i)$. Then I compute q_i such that $q_i = f(u \cdot s_i^c) \cdot t_i^z$ and I send the q_i's to Z.

2. Z selects $n - 1$ candidates out of the q_i's I sent him.

3. For the candidates Z selected in step 2, I reveal s'_i and t'_i.

4. Z can verify the $n - 1$ candidates he selected in step 2 by verifying that $q_i = f(u \cdot f(s'_i)^c) \cdot f(t'_i)^z$ with the s'_i's and t'_i's I sent him in step 3.

5. If all the verifications are correct, Z signs the non-verified candidate.

3.4 Possible Problems/Comments (brought up in lecture) with the Protocol

- In the protocol with Z, it seems that Z can only verify that $g_n = f(u \cdot f(s'_n)^c) \cdot f(t'_n)^z$ if it knows c. Knowing c implies that Z knows which organization, in this case X. This goes against our goals.

- We also noticed that if all organizations had c,$\overline{c}$ as keys then anyone could compute [3.3]. It was also pointed out that Y only needs to know c and not $\overline{c}$ to verify that $[3.1]^z = f([3.2])$ and $[3.3]^c = [3.2]$. Therefore it seems that Z needs to know what organization we are dealing with to obtain c rather than having all organizations have the same public and private keys.

- What prevents me from getting a second credential?

Since it seems that Z needs to know which organization we are dealing with, Z could ensure that each person only gets one pseudonym for each organization.

In order to maintain privacy, one could get a new pseudonym for each organization, so that Z cannot find out which organizations we are conducting transactions with and which organizations we are not conducting transactions with.

3.5 Comments on the Scheme as a Whole

- What if someone steals someone else's credential?
- This scheme seems awkward to use from a user interface point of view. A system enforced by the legislature, similar to what they have in England, seems more practical and/or better to achieve the goal.

3.6 What about the cases mentioned at the beginning of the lecture that need to correlate information?

- IRS to verify wages.

  You could tell your company your IRS pseudonym, but the company needs to verify you are giving it the right pseudonym. A protocol would be needed to set up an ID so that the company and the IRS could communicate.

- Credit checks.

  In this case, there needs to be a credit clearinghouse that would collect information about everyone. It seems that a central place would be needed for each category. But some amount of independence must be kept. (For example, driver's license versus credit)

- Security clearance

  You would have to reveal all your pseudonyms, but what keeps you from not revealing some? (Those that would reveal negative information for example)

4 Handouts

There were two handouts:

- Readings for Lecture 21 (handout #33)
- Problem Set #9

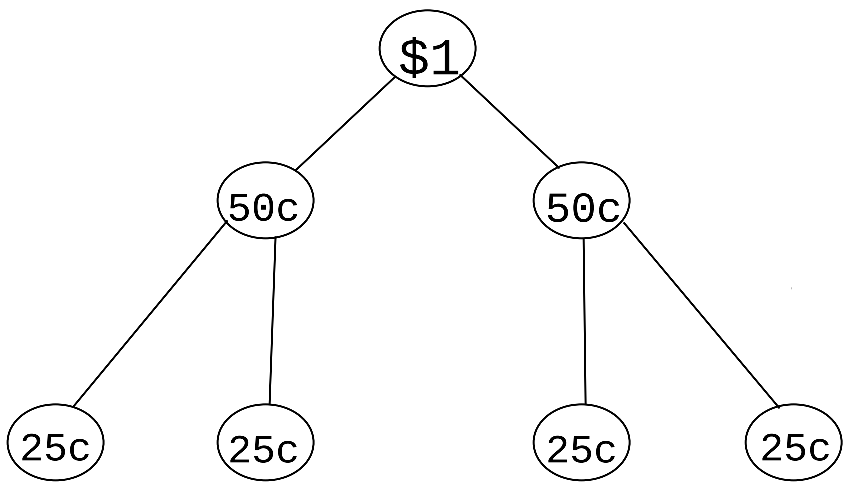

Figure 1: The root of the tree corresponds to the value of the tree: in this case $1. The children of the root correspond to coins of 50 cents, etc.