6.915 Computer and Network Security	**Fall Term, 1995**
Lecture 19 : November 14, 1995	
Lecturer: Ron Rivest	*Scribe: Douglas Schreiber*

1 Administrivia

Take Home mid-term results:

average(mean) = 94.75/100

median = 97/100

standard deviation = 7.5

2 Outline

- Digital Cash (*a la Chaum*)

3 Desirable Properties for Digital Cash

- forgery is hard
- duplication
  - prevent
  - detect
- avoid requiring that the bank keep a database
- preserve customers' anonymity
- anonymity can be preserved by providing transferability (like in real cash) ...of course, then you need to solve the duplication problem

4 An Analysis of Three Possible Digital Cash Protocols

4.1 A First-Try Bare-Bones Protocol

1. Customer tells bank she would like to withdraw \$100.
2. Bank returns a \$100 bill which looks like this: $\{$I am a \$100 bill, #4527)$\}_{SK_B}$
3. The customer pays the merchant with the bill.
4. The merchant gives the bill to the bank which credits the merchant's account.

Properties of/problems with this protocol:

- verifyable, but copyable
- forgery is impossible (except copying) (this is based upon the assumption that the signature scheme is secure and makes forgery impossible)

4.2 A second try that perserves anonymity

This approach involves blind signatures. The user presents the bank with a bill inside a container. The bank signs the bill without seeing the contents of the bill. This way, the bank cannot determine the source of a bill when a merchant presents it for deposit.

A useful analogy: The user covers a check with a piece of carbon paper and then seals both of them inside an envelope. The user gives the envelope to the bank. The bank then signs the outside of the envelope with a ball-point pen and returns the envelope to the user (without opening it - acutally the bank is *unable* to open the envelope in the digital version). The user then removes the signed check from the envelope and can spend it. The bank has never seen what it signed, so it cannot associate it with the user when it is returned to be deposited, but it can verify the signature on the check and thus guarantee the validity of the check.

There is, *of course*, a problem with this: The bank can be fooled into signing phony bills. For example, a user could tell the bank he's making a \$1 withdrawal and then present a \$100 bill to be signed. The bank will, unknowingly, sign the \$100 bill and allow the user to cheat the bank out of \$99.

It is possible to use RSA to create blinded signatures: In general, the RSA signature scheme works like this:

If $\mathcal{M}$ is the message to be signed, then

$$x = RSA - Signature(\mathcal{M}) = \mathcal{M}^{\frac{1}{d_b}}(mod n_b)$$

where n_b and d_b are publicly known values for the bank's signature. The secret information that the bank possesses is the inverse of $d_b(mod n_b)$, which we represented above as $\frac{1}{d_b}$.

The signature can be verified by calculating x^{d_b} and verifying that it is equal to $\mathcal{M}(mod n_b)$. (The customer knows n_b, so he can do $mod n_b$ arithmetic.)

In this case, the coustomer wants the bank to provide him with x, without revealing $\mathcal{M}$ to the bank.

The customer does this as follows:

1. Choose some random number, $r(mod n_b)$.
2. Calculate $\mathcal{M}' = \mathcal{M} \cdot r^{d_b}(mod n_b)$.
3. Give the bank $\mathcal{M}'$.
4. The bank returns a signature for $\mathcal{M}'$, x'.

   Note that $x' = (\mathcal{M}')^{\frac{1}{d_b}} = \mathcal{M}^{\frac{1}{d_b}} \cdot (r^{d_b})^{\frac{1}{d_b}} = \mathcal{M}^{\frac{1}{d_b}} \cdot r$.

   Since the customer knows r, he can divide x' by r to obtain $x = \mathcal{M}^{\frac{1}{d_b}}$.
5. The customer can then calculate x^{d_b} to verify the correctness of the signature, x.

This solves the problem of preserving the customer's anonymity, but other problems remain. The bank can still be fooled into signing something that it shouldn't (like a \$100 bill that it thinks is a \$1 bill), as we discussed before.

How can we fix the dollar amount?

- only have one denomination (per public key, for example)
- use a "cut and choose" procedure:

1. customer makes up 100 $20 bills
2. blinds them all
3. gives them to the bank
4. The bank picks one to sign (at random), and requires that the customer unblind all of the rest (by revealing the rs). Before the signature is returned, the bank ensures that all of the bills that were unblinded were correct.

Of course, one could set up the protocol to create an arbitrarily small chance that a cheater will succeed (by requiring that the user provided more blinded messages, for example).

So, now this protocol satisfies the anonymity requirement and can provide sufficiently small possibilities for cheating. However, there is no way to stop double spending, and it hard to track down and prosecute double-spenders.

In order to stop double-spending, the bank must record all of the bills it receives in a database. However, this database would have to keep growing and growing forever, unless the bills had some expiration. Unfortunately, expiration is not a desirable property of cash!

There is also a more subtle problem: two customers could create the same serial number accidentally. However, this could be solved by requiring that the serial number on a bill includes a hash of the user's unique id, or requiring that the serial numbers be chosen randomly out of a sufficiently space (to provide a sufficiently small chance that serial numbers will be duplicated).

4.3 A Third Protocol Which Allows Identification of Cheaters

ModIfy the second protocol from above so that the customer writes a "random identity string", or RIS, on the bill when presenting that bill as payment to a merchant. The RIS must be guaranteed to be different for every payment. The spender can then be identified by the RIS. If the bank receives two identical bills with different RIS values, then the customer has cheated. If the bank receives two identical bills with the same RIS values, then the merchant has cheated.

The problem then becomes: How does one create a bill that only the customer can write on (inside the signature).

Let $\mathcal{M}$ = (I'm \$20, #4527,$y_1, y'_1, y_2, y'_2, \ldots, y_{100}, y'_{100}$)

where $y_i = \mathcal{H}(x_i)$, $y'_i = \mathcal{H}(x'_i)$, and $\mathcal{H}$ is a one-way hash function.

Then, $RIS = \{oneof\{x_1, x'_1\}, oneof\{x_2, x'_2\}, \ldots, oneof\{x_{100}, x'_{100}\}\}$.

However, we still cannot trace the customer. We need a property such that the pattern of x,x's reveals the customer's name.

If we let $x_i \oplus x'_i$ = the customer's name, then if the customer double spends and uses two different RIS's, her name will be revealed.

Now the bank needs to make sure that the customer is behaving...and keep the customer's anonymity without associating the customer with the serial numbers.

We will continue to discuss this next lecture.