Chapter 6:Esoteric Protocols

Dulal C Kar

Secure Elections

- Ideal voting protocol has at least following six properties
  1. Only authorized voters can vote
  2. No one can vote more than once
  3. No one can determine who voted for whom
  4. No one can duplicate anyone else's vote
  5. No one can change anyone else's vote without being discovered
  6. Every voter can make sure that his vote has been taken into account in the final tabulation
- Additionally
  7. Everyone knows who voted and who did not

Simplistic Voting Protocol #1

1. Each voter encrypts his vote with the public key of a Central Tabulating Facility (CTF)
2. Each voter sends his vote in to the CTF
3. The CTF decrypts the votes, tabulates, and makes the results public

Simplistic Voting Protocol #2

1. Each voter signs his vote with his private key
2. Each voter encrypts his signed vote with the CTF's public key
3. Each voter sends his vote to a CTF
4. CTF decrypts the votes, checks the signatures, tabulates the votes, and makes the results public

Voting with Blind Signatures

1. Each voter generates 10 sets of messages, each set containing a valid vote for each possible outcome and a randomly generated unique identification number for the voter
2. Each voter individually blinds all of the messages and sends them, with their blinding factors to the CTF
3. CTF checks its database to make sure the voter has not submitted his blinded votes for signature previously. It opens nine of the sets to check that they are properly formed. Then it individually signs each message in the set. It sends them back to the voter, storing the name of the voter in its database

Voting with Blind Signatures (cont'd)

4. The voter unblinds the messages and is left with a set of votes signed by CTF
5. The voter chooses one of the votes and encrypts it with the CTF's public key
6. The voter sends his vote in
7. CTF decrypts the votes, checks the signatures, checks its database for a duplicate identification number, saves the serial number, and tabulates the votes. It publishes the results of the election, along with every serial number and its associated vote

Voting with Two Central Facilities

- The following protocol uses a Central Legitimization Agency (CLA) to certify voters and a separate CTF to count votes
  1. Each voter sends a message to CLA asking for a validation number
  2. CLA sends the voter back a random validation number. CLA keeps a list of validation numbers and a list of their recipients to prevent multiple voting
  3. CLA sends the list of validation numbers to CTF
  4. Each voter chooses a random identification number. He creates a message with that number, the validation number he received from CLA, and his vote. He sends this message to CTF
  5. CTF checks the validation number against its list it received from CLA in step (3), If found, CTF crosses it off to prevent someone from voting twice. CTF adds the identification number to the list of people who voted for a particular candidate and adds one to the tally
  6. After all votes have been received, the CTF publishes the outcome, as well as the lists of identification numbers and for whom their owners voted

Improved Voting with a Single Central Facility

- Satisfies first six requirements and the following two properties
  1. A voter can retract his vote and vote again within a given period of time
  2. If a voter finds out that his vote is miscounted, he can identify and correct the problem without jeopardizing the secrecy of his ballot

Protocol: Improved Voting with a Single Central Facility

1. CTF publishes a list of all legitimate voters
2. Within a specific deadline, each voter tells CTF whether he intends to vote
3. CTF publishes a list of voters participating in the election
4. Each voter receives an identification number I using some protocol (sec 4.13)
5. Each voter generates a public-key/private-key pair: k, d. If v is the vote, he generates the following message and sends it to CTF anonymously: $I,E_k(I,v)$
6. CTF acknowledges receipt of the vote by publishing: $E_k(I,v)$
7. Each voter sends the CTF: I,d
8. CTF decrypts the votes. At end of election, it publishes results and for each different vote, the list of all $E_k(I,v)$ values that contained that vote
9. If a voter observes that his vote is not properly counted, he protests by sending CTF: $I,E_k(I,v),d$
10. If a voter wants to change his vote from v to v', he sends CTF: $I, E_k(I,v'),d$

Voting without a Central Tabulating Facility

- Voters watch each other
- But cannot be implemented practically for more than a handful of people

- You may check pages 130-132

Secure Multiparty Computation

- How can a group of people calculate their average salary without learning others' salaries?

1. Alice sends to Bob: $K_B(S_A+R_A)$,
   where S_A-Alice's salary; R_A-secret random number; K_B: Bob's public key
2. Bob decrypts and then sends Carol: $K_C(S_A+R_A+S_B)$
   where S_B-Bob's salary; K_C: Carol's public key
3. Carol decrypts and then sends Dave: $K_D(S_A+R_A+S_B+S_C)$
4. Dave decrypts and then sends Alice: $K_A(S_A+R_A+S_B+S_C+S_D)$
5. Alice decrypts and then subtract her secret number R_A from the sum to obtain the sum of salaries
6. Alice divide the sum by four (no. of people) to obtain the average

Digital Cash

- Checks and credit cards have reduced the amount of physical cash flow but have an audit trail
- Digital cash
  - Anonymous, untraceable
  - One time use for purchase or transfer or change (cannot be copied)
- Digital cash protocols are very complex

Digital Cash: Analogous Protocol 1

- Alice prepares 100 anonymous money orders for $1000 each
- Alice puts each money order and a piece of carbon paper in an envelope. She sends all 100 envelopes to bank
- Bank opens 99 envelopes and confirms that each is a money order for $1000.
- Bank signs the one remaining unopened envelope, which goes through the carbon paper to the money order. Bank hands the unopened envelope back to Alice, and deducts $1000 from her account
- Alice opens the envelope and spends the money order with a merchant
- Merchant checks bank's signature for legitimacy of the money order
- Merchant takes the money order to bank
- Bank verifies and credits $1000 to merchant's account

Digital Cash: Analogous Protocol 2

- Previous protocol has double spending problem. How? Alice can copy the money order and spend it twice
- Protocol

1. Alice includes a different random unique string on each 100 anonymous money orders for $1000 each

2-7 steps are the same as in protocol 1

8. Bank verifies its signature and checks its database to make sure a money order with the same uniqueness string has not been previously deposited. If not, it credits $1000 to merchant's account and records the string in a database
9. If previously deposited, bank does not accept the money order

Digital Cash: Analogous Protocol 3

- In previous protocol, bank cannot identify the cheater (Alice or Merchant). Explain.
- Protocol

Steps 1-6 are same as in protocol 2

7. Merchant asks Alice to write a random identity number string on the money order
8. Alice complies
9. Merchant takes the money order to bank
10. Bank verifies its signature and checks its database to make sure a money order with the same uniqueness string has not been previously deposited. If it has not, bank credits $1000 to merchant's account. Bank records uniqueness string and identity string in a database
11. If uniqueness string is in database, bank refuses the money order. Then, it compares the identity string on the money order with the one stored in the database. If it is the same, the bank knows that the merchant photocopied the money order. If it is different, the bank knows that the person who bought the money order photocopied it.

– Alice can easily frame the merchant. How?

Digital Cash: Protocol 4 using cryptography

- How to identify the cheater (Alice)
- Technique of secret splitting can be used to hide Alice's name in the digital money order
- Protocol

1. Alice prepares *n* anonymous money orders for a given amount. Each money order contains:

   Amount

   Uniqueness string: X

   Identity strings: $I_1 = (I_{1L}, I_{1R})$

   $I_2 = (I_{2L}, I_{2R})$

   . . . .

   $I_n = (I_{nL}, I_{nR})$

   Each of the pair is generated as:

   Alice creates a string that gives her name, address, etc. that the bank wants to see. Then, she splits it into two pieces using the secret splitting protocol. Then she commits to each piece using a bit-commitment protocol. Any pair reveals Alice's identity.

Protocol 4 (cont'd)

2. Alice blinds all *n* money orders, using a blind signature protocol. She gives them all to the bank
3. Bank asks Alice to unblind any (*n*-1) money orders at random. After verification, bank asks Alice to reveal all of the identity strings
4. Bank signs the remaining blinded money order, gives it to Alice, and deducts the amount from her account
5. Alice unblinds the money order and spends it with a merchant
6. Merchant verifies bank's signature
7. Merchant asks Alice to randomly reveal either the left half or the right half of each identity string on the money order. Essentially, merchant gives Alice a random *n*-bit selector string $b_1, b_2, \ldots, b_n$. Alice opens either left or right half of Ii, depending on whether b_i is a 0 or a 1
8. Alice complies
9. Merchant takes the money order to bank

Protocol 4 (cont'd)

10. Bank verifies signature, checks its database for the uniqueness string, and credits merchant's account with the amount, if not deposited earlier. Bank records uniqueness string and all of identity information in a database
11. [Finding the cheater]

    If uniqueness string is found in the database, bank refuses the money order. It checks the identity string to find out whether the merchant or the buyer has copied the money order. If the identity string is different, the bank knows that the buyer has copied it. To reveal identity of buyer, bank finds a bit position where one merchant had Alice open the left half and the other merchant the right half and XORs first merchant's half and second merchant's half.

Notes about Protocol 4

- The odds that two random selector strings will be the same is 1 in 2^n . Why?
- The protocol does not keep Alice from trying to cheat. How?
- Bank doesn't know the identity of spender unless he or she copies the money order. Explain
- Alice cannot prevent her identity being revealed if she cheats. Explain
- Alice could try to use somebody else's name. The odd is in this case is 1 in n. Why?

Notes about Protocol 4 (cont'd)

- Can the merchant cheat?
  - No, bank will notice repeated use of the selector string
  - Even collusion between Alice and merchant can't cheat bank. How? Bank signs a money order with the uniqueness string
- Can bank cheat?
  - Bank cannot figure out from the money order who was the buyer
  - Bank and merchant together cannot figure out who Alice is

Notes about Protocol 4 (cont'd)

- Eve can cheat. How?
  - Eavesdrops on the communication between Alice and merchant
  - Goes to bank before merchant and deposits digital cash first
  - Bank accepts it
  - Though wrongfully, bank identifies merchant as cheater
  - If Eve steals and spends Alice's cash before Alice, then Alice will be identified as cheater

Digital Cash and Perfect Crime

- Digital cash has its dark side
- Watch Alice commit the perfect crime
  1. Alice kidnaps a baby
  2. Alice prepares 10,000 anonymous money orders for $1000
  3. Alice blinds all using a blind signature protocol. She sends them to authorities with the threat to kill the baby unless the following instructions are met:
     a) Have a bank sign all 10,000 money orders
     b) Publish the results in a newspaper
  4. Authorities comply
  5. Alice buys a newspaper, unblinds the money orders, and starts spending them. There is no way for the authorities to trace the money orders to her
  6. Alice frees the baby

Notes

- Digital cash, in general, isn’t good deal for criminals. Why?
- Spender is anonymous but merchant is not

Properties of Ideal Digital Cash System

- 1.Independence
  - Security is independent of physical location
- 2. Security
  - Cannot be copied and reused
- 3. Privacy (untraceability)
- 4. Offline payment
- Transferability
- Divisibility
- Previous protocols satisfy properties 1, 2, 3, and 4. There is a protocol that satisfies all