One final weird problem

- If Alice is anonymous when transacting with Carol, can't she just cheat by quitting the protocol?
- Alice sends Carol and Dave (M,S), Carol and Dave demand all but one nonce piece
- If they ask for the same set, Alice wins, hands them over, and has double-spent the money
- If they ask for different sets, Alice just logs off, waits a random amount of time, and tries again.

Better solution: oblivious transfer

- I have separate pieces of information $\{M_1, M_0\}$
- You want one of them, but you don't want me to know which one I am giving you.
- Imagine if a money order has two nonce halves, and Carol can demand one without Alice knowing which was transmitted. Then she can't game the system with Carol and Dave.

Better solution: oblivious transfer

- I give you two envelopes with different PO Box addresses
- Somehow we guarantee that I only can access a single PO Box (nevermind how)
- You place one document each in the two envelopes, and mail them
- I open whatever arrives in my PO box.

Blinded transfer of message M

1. Alice: encryption key E_A, decryption key D_A
   Alice→Carol: E_A

   This is an anonymous protocol right now, so Carol can't look up "Alice's" encryption key. She can, however, make one up for the purpose of this exchange.

Blinded transfer of message M

1. Carol→Alice blinding factor $T=\text{Encrypt}[E_A, R]$
2. Alice→Carol: $C=M\times\text{Decrypt}[D_A, T] = M\times R$
3. Carol: $M = CR^{-1}$

   This only works because Carol knows the decryption of T, and she only knows that because she made T the encryption of a known value R.

Blinded transfer of message M

1. Carol: blinding factor T_1=Encrypt[E_A, R],
   T_0 = [string reverse T_1]

2. Carol→Alice: {T_1, T_0} (unlabeled, in random order, no way to tell which message is the forwards one)

3. Alice→Carol:
   C_1=M_1×Decrypt[D_A, T_1] = M_1×R
   C_0=M_0×Decrypt[D_A, T_0] = M_0×??????

   Carol only knows the decryption of one blinding factor, and can only unblind one message. In practice, Alice doesn't know which one is reversed, and doesn't know which transmitted message can be unblinded.

Weird things we can do:

- Prove you know a secret “password” that you never transmit or share with anyone.
- Safely sign a document without looking at it, to preserve the anonymity of another party.
- Create an anonymous channel that magically identifies the participants if they break a rule.
- Produce a “spendable” bit string that can not be copied and spent multiple times.

Weird things we can do:

- Create a system where completely anonymous people have message integrity (i.e., they can prove a message is from them, and unaltered.)
- Lock up messages to become readable only under certain conditions, e.g. 3 separate people try to read them.
- Entangle messages so that only one is readable, but people can’t tell which.

Voting

The voting problem

- Set of voters {Alice, Bob, Carol, … }
- Set of nonvoters {Eve, etc}
- Counting authority or authorities
  - Can be multiple: balloting server, counting server, voter registration server, etc
- Anyone can have whatever keys or datagrams are necessary for communication

The voting problem

- Every registered voter should be able to cast exactly one vote. Nonvoters can not cast votes.
- The authority can count votes without knowing who voted for whom
- It should be possible to prove that:
  - Your vote was counted
  - Your vote was *correctly* counted
  - The totals are honest

Online versus offline voting

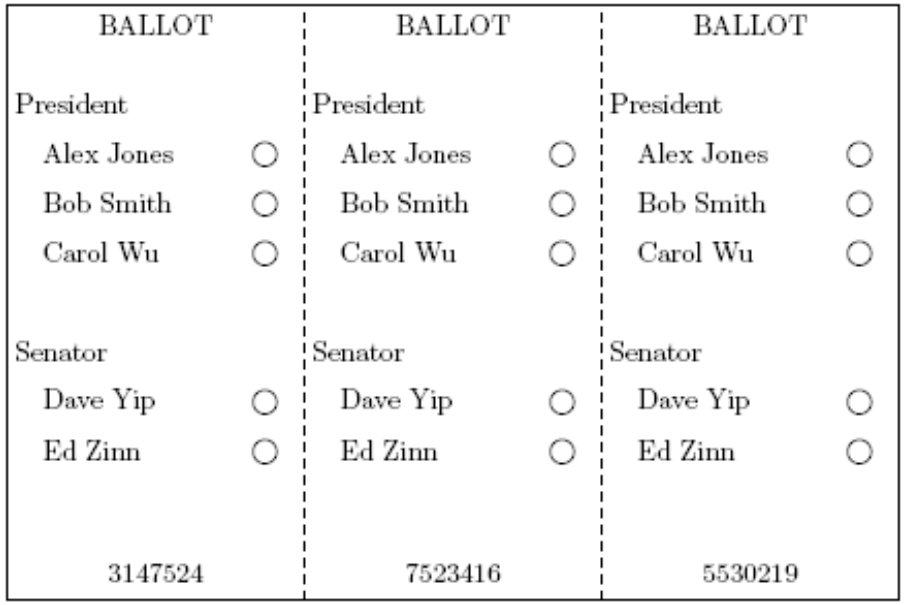
- Voting “over the Internet” is hard
- Voting at a balloting location allows certain security policies to be put in place:
  - No recording vote, or letting people watch you vote (“unsticks” the threat of bribery or coersion)
  - Repeated attempts are not easy
  - Fraud is riskier
  - Process can be independently observed
  - Physical process to guarantee counting, anonymity of votes

Online versus offline voting

- Still some clever attacks available to attackers:
  - “Chain voting”
  - Attackers can exploit physical limitations of voting booth, e.g. sabotage of machines or impeding voting process to slow down votes in targeted precincts.
- Generally, offline voting can offer more security, but presents a challenge to achieve some stranger, “cryptographic” goals.

ThreeBallot

- Offline voting system that attempts to prove ballots are fully and correctly counted, while preserving anonymity
- Each ballot has three parts, each with a random ID number



BALLOT		BALLOT		BALLOT	
President		President		President	
Alex Jones	○	Alex Jones	○	Alex Jones	○
Bob Smith	○	Bob Smith	○	Bob Smith	○
Carol Wu	○	Carol Wu	○	Carol Wu	○
Senator		Senator		Senator	
Dave Yip	○	Dave Yip	○	Dave Yip	○
Ed Zinn	○	Ed Zinn	○	Ed Zinn	○
3147524		7523416		5530219	

- Every row is marked with a randomly assigned dot; one votes for a candidate by marking a second dot.
- The ballot is sliced, all three are counted, and a copy of one slice is given to the voter as a receipt.

Online voting

- Server has list of registered voters (and their public keys)

- Attempt one:

1. Voter sends in ballot, signed with voter's private key, then encrypted with server's public key.

2. Server verifies signature, counts ballot, publishes totals.

Online voting

- Attempt two:

1. Voter uses cut-and-choose protocol to get the server to sign an unknown ballot. (How?!)

2. Voter sends signed ballot to server anonymously

3. Server verifies its own signature, publishes all submitted ballots

Online voting

- Attempt three:

1. Voter authenticates with Voter Registration Server, is given a random token

2. Voter Registration Server gives list of all tokens to Vote Counting Server

3. Voter sends $\{token,vote\}_{ES}$ over anonymous channel.

Online voting

- Attempt four (attempt three with one server):

1. Voter sends Server signed intent to vote, uses cut-and-choose protocol to receive a signed anonymous token:

$$\text{token} = \{ \text{hash[vote-token]}, \text{election \#}, \text{date} \}_{DS}$$

2. Server publishes list of signed intent certificates (establishes number of voters by start of election)

3. Voter publishes token over anonymous channel.

Online voting

- Attempt four (attempt three with one server):

4. Voter publishes $\{\text{vote-token, vote}\}_{ES}$ anonymously.

5. Server publishes list of vote-tokens and votes.

Online voting

- Attempt four (attempt three with one server):

4. Voter publishes Encrypt(vote-token, vote)$_K$ anonymously.

5. After commitment deadline, voter publishes K anonymously