Midterm Exam **COS 432, Fall 2004**

There are three questions, all of equal weight. You may use your own lecture notes, copies of other people's lecture notes, and any summaries you have made of those notes. You may not use any other materials.

Some problems may have more than one correct answer, so don't be too worried if you can think of good alternative answers. You must give a single answer. You'll get full credit if your answer is one of the correct ones.

If you can't figure out a complete or perfect answer to a question, tell us what you have figured out. If your answer is flawed, you'll get more credit by showing us that you know the limitations of your answer, rather than pretending not to notice what's wrong with it.

Please do your work in an exam book.

Undergraduates: Before turning in your exam, please hand-write and sign the Honor Code pledge, "I pledge my honor that I have not violated the Honor Code during this examination."

(Graduate students: No need to sign anything. You are bound automatically by university regulations.)

---

1. The Diebold AccuVote-TS is an electronic voting machine that has been used in several real elections and will be used in several states in the upcoming national election. Independent researchers have found serious security flaws in these machines. One such flaw involves the method for ensuring that a voter does not vote more than once.

   Suppose Alice is a voter. When Alice arrives at the polling place, she goes to a desk manned by election worker Bob. Alice tells Bob her name and Bob verifies that she is authorized to vote in that polling place. Then Bob takes a smartcard and uses a small computer at the sign-in desk to put the smartcard into "active mode". Bob gives the smartcard to Alice.

   (A smartcard is size and shape of a credit card. For the purposes of this question, you can think of it as being a tiny computer that has a computer program wired into it when it is manufactured. A smartcard can communicate with any device if it has physical contact to that device. Diebold has preprogrammed the smartcards Bob is using, but anybody else can make their own smartcards and preprogram them to run whatever program they like.)

Alice inserts the smartcard into the AccuVote voting machine. The smartcard and the AccuVote exchange electronic messages, and if the smartcard is in "active mode" it is switched to "inactive mode" and Alice is allowed to cast her vote. On her way out of the polling place, Alice gives the (now inactive) smartcard back to Bob.

This procedure is supposed to ensure that Alice can vote only once, because it is supposed to be impossible to cast a vote without presenting a smartcard that has been activated by Bob.

Here is the protocol (i.e., the series of messages) exchanged between the smartcard and the AccuVote when Alice inserts the smartcard:

1. AccuVote to smartcard: "Are you in active mode?"
2. smartcard to AccuVote: "Yes."
3. AccuVote to smartcard: "Please put yourself in inactive mode."
4. smartcard to AccuVote: "Done."

This protocol is insecure because, although Bob's smartcards will follow the rules, the AccuVote does not authenticate the smartcard. Alice can cast an extra vote by making her own smartcard that sends the two messages "Yes" and "Done", sneaking that smartcard into the voting booth, and plugging it into the AccuVote. Indeed, if Alice makes her own smartcard that sends those two messages over and over, then she can cast as many votes as she likes – each time she inserts her homemade smartcard into the AccuVote it will allow her to cast one more vote.

Explain how you would fix this protocol. You may assume the following: Bob is honest; Bob can preprogram his smartcards any way he likes; Alice cannot steal or reprogram one of Bob's smartcards; Alice can make her own smartcards and program them however she likes. Alice's goal is to sign in once at the front desk and then cast multiple votes. Your goal is to design a system that stops her from doing so.

Your answer should explain these things:

- which cryptographic key or keys are used, and which devices know which key(s);
- what program runs on Bob's legitimate smartcards;
- what messages are exchanged between the AccuVote and the smartcard that is inserted into the AccuVote;
- why a smartcard programmed by Alice (or any smartcard other than Bob's) cannot trick the AccuVote into allowing a vote to be cast.

2. RFID is a technology for identifying objects via radio. (Prox cards use a form of RFID technology.) RFID uses small objects called "tags" that are implanted in everyday items. Each tag has a unique identifying number that we'll call its "tagID". An RFID "reader" can detect any tags that are nearby (within a few meters) and learn their tagIDs. To detect nearby tags, a reader emits a fixed radio signal called a "beacon". The beacon conveys no information but merely activates any tags that are nearby. Whenever a tag hears the beacon, it responds by sending a radio signal containing its tagID.

   Shoe stores want to use RFID to keep track of their inventory, by implanting an RFID tag into every shoe when it is manufactured. A shoe store employee could then walk around the store with an RFID reader and take a census of which particular shoes were present. An RFID reader at the store's entrance could also detect which shoes leave the store and when.

   This is nice for shoe stores, but it raises privacy concerns, because your shoes would still have functioning RFID tags after you bought them. If somebody could learn the tagIDs of your shoes (say, by pinging your shoes with an RFID reader), then they could build automated devices to detect when your shoes (and hence you) were present.
   Design an improved RFID system that better protects privacy. Assume that a reader still sends out a fixed beacon signal that conveys no information. On hearing the beacon, each tag will run a simple computation (which you devise) to generate a sequence of bits that it will send in response. On receiving the response, the shoe store should be able to recover the tag's tagID. But anybody else hearing the response should be unable to determine which tag is present, or even whether a tag that is present now is the same tag that was present at some earlier time.

   Your answer should explain these things:
   - which cryptographic key or keys are to be used, and which devices know which key(s);
   - how a tag generates the bits it will send on hearing the beacon;
   - how the shoe store can recover a tag's tagID after hearing the bits that the tag sends; and
   - why other parties who hear a tag's response to the beacon cannot learn the tag's tagID and cannot learn whether it is the same tag that was present at some earlier time.

3. Internet users are sometimes asked to give their email addresses to merchants, and the addresses they give sometimes end up being used by spammers. To prevent this, Charlie wants to be able to give a different email address to each merchant, in such a way that he can detect if one of those addresses is later used by somebody other than the merchant to whom he gave it.

   Here's an example. Suppose Charlie wants to give an email address to amazon.com. Charlie uses some algorithm to generate an email address such as d85jbk3s829a@charlie.com. He gives this address to Amazon. If he later receives an email message sent to d85jbk3s829a@charlie.com, fromsomebody@amazon.com, he will accept that message because it came from amazon.com. But if he later receives an email to that address from foo@bar.com, he will know that amazon.com gave the address to somebody else.

   So Charlie needs two algorithms. The first one, which we'll call "generate", takes an Internet address (in the form of a string like "amazon.com") and produces an email address in the charlie.com domain. The second algorithm, which we'll call "check", takes an email address A in charlie.com, and an Internet address S, and returns True if and only if S is supposed to be able to send email to A – that is, if A could have been produced by a call to generate(S).

   There are two more requirements. (1) Spammers should be unable to generate addresses for which the "check" algorithm returns True. (2) No matter how many addresses Charlie generates, he should need only a constant amount of storage space.

   Your answer should explain these things:

   - which cryptographic key or keys are to be used;
   - which "generate" algorithm Charlie should use;
   - which "check" algorithm Charlie should use; and
   - why your solution meets the requirements.