Princeton University — Computer Science
COS 432: Information Security (Fall 2013)

This test has 13 questions worth a total of 50 points.

That's a lot of questions. Work through the ones you're comfortable with first. Keep in mind that many questions have partial credit for incomplete answers, and some may have more than one correct answer. Notes are allowed, whether hand-written or printed.

Name and NetID:

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 by university rules against cheating.)

Optional: mark how you feel at the beginning of the test...

...and at the end

Q1. [7 points] Provide short (1-sentence) answers.

a. Tor encrypts traffic all the way from the sender to the exit node, but not from the exit node to the receiver. Why not?

b. Can you receive bitcoins even when your computer is not connected to the Internet?

c. Does storing hash(username || password) on the server better defend against an attacker who breaks into the server and tries to crack passwords than just storing hash(password)? Here || refers to concatenation.

d. Consider the family of functions $f_k(x) = x^k \bmod p$ where p is a large prime. Is this a PRF?

For each of the following, state whether or not it uses asymmetric encryption. Provide 1-sentence explanations. [Added clarification: asymmetric encryption refers to just encryption, not asymmetric cryptography in general. Specifically, digital signatures and such are excluded.]

e. Connecting to a secure website

f. Bitcoin

g. Tor

Q2. [2 points] Alice has a procedure to generate a random number as follows: she uses a true random number generator to pick a random index in the RAND corporation book of 1 million random digits. She then picks the sequence of 10 digits starting at that index (wrapping around if necessary), resulting in a number between 0 and 10 billion. Roughly how many bits of entropy does this procedure provide per invocation? Explain.

Q3. [2 points] Acme Security, the security division of Acme Corporation, is worried about backdoors in crypto standards, but has heard the advice that homebrew ciphers are likely to have vulnerabilities. Therefore they come up with a plan to get the best of both approaches: they invent their own cipher AcmeSec but use it in conjunction with AES. That is, their encryption algorithm is $AES_{\mathbf{k}}(AcmeSec_{\mathbf{k}}(\mathbf{m}))$ where **k** is the key and **m** is the message.

What is one reason this might be worse for security than just using AES?

Q4. [2 points] Intuitively it seems like an encryption scheme should be secure if an adversary can't decrypt the ciphertext without knowledge of the key. Why not define security in terms of this property? Why bother with the "encryption game?"

Q5. [2 points] Consider this example from the notes: *Zfone, a secure VoIP system, protects against MITM attacks with a short authentication string. After two Zfone terminals exchange keys, both computers display a four-character string. The users are supposed to manually verify that both strings are the same — "my screen says 5C19; what does yours say?" — to ensure that the phones are communicating directly with each other and not with a man in the middle.*

How come a short string is enough for security here, whereas we require cryptographic keys to be at least 128 bits? Assume that the two parties have never met and can't recognize each other's voice.

Q6. [2 points] Mallory breaks into Alice's computer and steals her Bitcoin wallet (that contains her Bitcoin private keys). Fortunately Alice has a backup of her wallet on another computer. What determines who has the bitcoins now?

Q7. [2 points] Following best security practice, Trippy Games Inc., makers of addictive games for Windows, develop their software on machines behind a firewall, but distribute them via a web server in the "de-militarized zone" (DMZ) outside this firewall. What additional security measure can they take to decrease the chance that their customers will download and install malware instead of addicting games, even in the event of the DMZ computers being taken over by an attacker?

Q8. [4 points] When connecting to a machine for the first time using SSH on the command line, the following transcript is typical (user input is in bold):

$ **ssh portal.cs.princeton.edu**
The authenticity of host 'portal.cs.princeton.edu (128.112.155.166)' can't be established.
DSA key fingerprint is d3:74:ca:f9:62:fe:2b:15:cd:9b:be:d4:00:7e:58:74.
Are you sure you want to continue connecting (yes/no)? **yes**
Warning: Permanently added 'portal.cs.princeton.edu,128.112.155.166' (DSA) to the list of known hosts.

Explain what's going on here. What does it mean for authenticity to be established? Is there something the user could have done to establish authenticity before connecting? What is the key fingerprint and why is that being shown to the user? What does it mean to permanently add something to the list of known hosts, and what are the pluses and minuses of doing that? (If you don't know, take your best guess based on what we've learnt in class.)

Q9. [3 points] NotSpam Toys Inc., firm in their conviction that they are not spammers, decide to make it really easy for recipients of their emails to opt out.

A. Their first implementation is to allow users to opt out by clicking on a (user-specific) link to the URL notspamtoys.com/opt-out/<email>, and they put the opt out link at the bottom of all their emails. This turns out to be a bad idea. Why?

B. Humbled, they switch to an implementation where they set an authentication cookie in the user's browser when the user first creates an account on notspamtoys.com. Now the opt-out URL is simply notspamtoys.com/opt-out/. When a user visits the URL, the server identifies and authenticates the user via the cookie and opts them out. (If the user had deleted the cookie they fall back on asking the user to log in again.) This also turns out to be a bad idea. Why?

Q10. [5 points] Genius Cards Ltd., smart card manufacturers across the pond, make smart cards that have an RSA signing key **d** in tamper-resistant memory. When provided with input **x** the card computes and returns $\mathbf{x}^{\mathbf{d}}$ mod **N**. Here **N** is a public RSA modulus of length **n** bits. Sadly for Genius, the cards turn out not to be so smart: when placed on a table and hit with a mallet at just the right speed, a random bit of **d** gets flipped (but the attacker has no control over which bit).

A. Show how the attacker can extract the secret key **d**. Note that the attacker can provide arbitrary **n**-bit inputs and observe the corresponding outputs, and can intersperse this with mallet strikes in any order.

B. (Extra credit, 2 points) What is the expected worst-case running time of your attack in terms of computation performed by the smart card, other computation performed by the attacker, and mallet applications? Assume that the complexity of multiplying or dividing two **n**-bit integers is **O(n log n)**. Ignore constant factors.

Design

Q11. [5 points] Design a system for two people to be able to obtain each other's authentic public key when they meet physically, so that they can later securely email each other. Analyze the security and usability of your design. What are your trust assumptions, and what adversaries does it protect against? (This is completely open-ended; you could have them just use pen and paper, or smartphones, or some custom device.)

Q12. [6 points] CryptoLocker is a piece of malware that encrypts files on victims' computers and demands a ransom in exchange for the ability to decrypt. The creator wants to achieve certain properties: the user shouldn't be able to decrypt without payment (even if they hire a forensics expert). Further, different victims should not be able to co-operate to avoid paying separately: if victim Alice pays, that should in no way help victim Bob decrypt his files. Finally, the creator wants to minimize the cost of administering the malware as well as the probability of getting shut down. How can CryptoLocker achieve these properties (to the extent possible)? What key material, if any, should be part of the malware binary? How would encryption and decryption work? Analyze the pluses and minuses of your design (from the malware author's point of view).

Q13. [8 points] Capable Essays (capable-essays.com), purveyor of high-quality user-generated essays, provides the following functionality: users can create new essays and view, edit or delete existing essays. With the proper authorization, of course. Capable wants a capability-based access control system, i.e., access to any of the 3 operations (view/edit/delete) on an existing essay is allowed if and only if the user possesses the right capability URL. Anyone can create a new essay on the site. When they do, they get 3 capability URLs (for view/edit/delete), each of which they are then free to share with others. Sharing the URL identifies the essay in question and simultaneously shares the corresponding permission.

A. Design a URL scheme that lets Capable achieve the above properties. What will the three URLs corresponding to a newly created essay look like? Note that a capability URL like `https://capable-essays.com/view/essay-id` is insecure — it allows a clever user to simply type in `https://capable-essays.com/edit/essay-id` and edit an essay for which she only has view permissions. (It might also allow users to guess essay IDs unless essay-id values are long and random.) Hint: the above attack, trivial as it is, is a forgery. How can we make URLs unforgeable?

Briefly describe how access control enforcement on the server would work, given your URL scheme.

B. Some users misbehaved and shared their URLs with the world, resulting in angry customers, so the site wants a new URL scheme that gives users the ability to revoke capabilities they've given out. One way to do this is to have multiple URLs for the same (essay, action) pair. We want the following two functionalities:

1. A user can create a new capability URL for any (action, essay) pair that she already holds a capability for.
2. A user can revoke any capability that she created, which also revokes all capabilities created from it (and so on transitively).

Describe how you would design a URL scheme for this system. How would you implement the above two functionalities? What data would you need to store on the server? How would access control enforcement work?

Note: beyond the capability creation, sharing, and revocation details described here, users shouldn't have to worry about any other security details, such as accounts, passwords or keys.

C. (Extra credit, 2 points) Provide an answer for part B that does not involve storing new information on the server every time a new capability is created; only a list of revoked capabilities should be stored.

Hint: URLs can be arbitrarily long.