Massachusetts Institute of Technology
6.857: Network and Computer Security
Professor Ronald L. Rivest

Handout 24
October 24, 2002

Fall 2002 Midterm

1. There are six (6) problems and a total of 100 points. This midterm is due on *Thursday, October 31, 2002* at the beginning of class. Late midterms will *not* be accepted. If you can't hand the midterm in or have a friend hand it in, then drop it off with the TA or course secretary BEFORE the deadline.

2. **Policy:** The midterm is open book, but *no collaboration is allowed!* You may not communicate with any person (except for the course staff) about any aspect of the midterm until after the hand-in deadline, even if you have already handed in your midterm. Violation of this policy will be dealt with severely.

   You may use your notes from this course, class handouts, textbooks, and resources online. Give citations for any material (other than your lecture notes and class handouts) that you use. Keep in mind that citations are not always correct or sufficient for justification. It is best to rely on your own reasoning and the material presented in class.

3. **Format:** Type up your solutions, staple each problem separately (subparts of a problem can be stapled together in proper order), and put your name and problem number on each page. Use at least an 11pt font size and 1 inch margins. You have a week for the midterm, so there is no excuse for sloppiness. You will receive 1 point for each problem in which you follow these directions correctly.

4. **Page limit:** Each problem has a specified a page limit. Pages beyond the specified limit will be ignored. Do not feel obliged to use the entire allotment; many problems have concise answers.

5. **Bugs, etc.:** Eventual bugs in the midterm or any hint the staff may feel inclined to give will be reported to you through the `6.857-students-public@mit.edu` mailing list. Check your e-mail regularly during the examination period! If you have not received e-mail on this list yet, then you are probably not on it. Tell the TAs ASAP.

Problem Q-1. I don't like spam [20 pts]

In this problem, limit your write up to three pages total. Additional pages will be ignored.

Email has become nowadays a wonderful means of communication: it is widely accessible, simple to use, virtually cost-free, and you can reach people at the other end of the world almost instantaneously.

But there is a flip side to this coin. Email has become as well a fantastic way of sending unsolicited advertising or marketing to people. Post a message in a newsgroup about digital rectal thermometers, and companies selling health equipment may send you information about the latest health devices that are completely useless to you. Enter your email address in an online raffle, and you may end up receiving an endless series of emails inviting you to enter even more of them.

As a consequence, fighting spam has become an ever-increasing need. Hoping to lay his hand on a golden opportunity, Ben Bitdiddle decides to enter the anti-spam business.

(a) Describe the objectives of an anti-spam personal agent. What would you expect an anti-spam personal agent to do for you? How would you rate the effectiveness of an anti-spam personal agent?

In your discussion, remember that the agent is fallible: there can be false positives (a non-spam email is detected as spam) and false negatives (a spam email is not detected as such).

Although many current anti-spam agents rely on AI techniques to smartly filter the emails based on content or other properties of spam email or spam senders, Ben decides to focus on a computational approach: the sender has to complete a certain amount of work for his email to get into the recipient's inbox.

Recall the hash-cash technique discussed in lecture: upon reception of an email, the recipient sends a challenge (x, z) (where z is a given b-bit long string) to which the sender needs to reply with r such that $h(x||r)$ starts with z, where h is a one-way hash function and $||$ denotes concatenation. To make the protocol non-interactive, one can set x to be the concatenation $(m||e||d)$ of the message m, the recipient's email address e and the current date/time d.

Instead of basing the work on the difficulty of finding a partial inversion, Ben Bitdiddle decides to base his scheme on the difficulty of finding a partial *collision*: the sender now needs to find $r \neq r'$ such that $h(m||e||d||r)$ and $h(m||e||d||r')$ have the same b leading bits.

(b) Evaluate the amount of work needed by the sender to meet the challenge in Ben's scheme, in terms of processing time and storage space. Compare it to the hash-cash technique.

How do the time and memory requirements grow with the number of recipients?

You must specify any assumptions you are making.

(c) Discuss the efficiency of this scheme: what would be an appropriate value for b? can the sender perform any pre-computation?

You can assume for this question that a typical *personal* computer can compute around one million hash evaluations per second.

(d) What are the advantages and drawbacks of this scheme compared to the more traditional approaches, based on heuristics? Remember to consider the different issues in actually implementing this scheme (deployment, ease of use, customization, ...).

Problem Q-2. Security of embedded devices [20 pts]

In this problem, limit your write up to three pages total. Additional pages will be ignored.

As technology progresses, we will likely see embedded computers in almost everything, from toasters to food packaging (perhaps in the form of RFID chips).

One way to manage these devices securely is to let every such device have an "owner" that can give it commands. For example, your PDA might be the "owner" for your toaster. Commands from other parties (including previous owners) are to be ignored by the device.

(a) Explain how such a device might be programmed to recognize commands from its owner, and only its owner. Show how this can be done with both classical (symmetric private-key) technology and with public-key technology.

(b) Describe a protocol explaining how "transfer of ownership" can be accomplished, under the public-key framework of your answer in part (a). That is, how can Alice instruct the device that Bob is now the new owner? (Remember that once this is done, the device should no longer respond to commands from Alice.)

(c) Describe a protocol explaining how "transfer of ownership" can be accomplished, under the classical (symmetric-key) framework of your answer in part (a), similarly.

(d) Compare the public-key and private-key approaches to this problem, in terms of efficiency, security, and flexibility.

Problem Q-3. Palladium? But I just met 'em. [25 pts]

In this problem, limit your write up to three pages total. Additional pages will be ignored.

For each of the four key components of Palladium (curtained memory, sealed storage, secure input/output, and attestation):

(a) Describe very briefly what the component does.

(b) Describe what vulnerabilities would be introduced if that component were removed from the design.

(c) Describe one or two applications that would probably become infeasible (insufficiently secure) if that component were not present. (Or if you think this component is inessential for all applications, explain why.)

(d) Describe one or two applications that would not be affected by the removal of that component, but which still require the remaining components. (If you think there would be no such applications, explain why.)

Problem Q-4. Universal Translators [25 pts]

In this problem, limit your write up to three pages total. Additional pages will be ignored.

Here is a variant of the ElGamal encryption scheme where the g and y values are swapped during encryption.

Algorithm G: Key generation for user i:

1. Generate a large random prime p and a generator g of the multiplicative group Z_p^* of the integers modulo p.
2. Select a random integer $x_i, 1 \leq x_i \leq p-2$ where $\gcd(x_i, p-1) = 1$, and compute $y_i = g^{x_i} \bmod p$.
3. i's public key is (p, g, y_i); i's private key is x_i.

Algorithm E: Encrypting a message to user i:

1. Obtain i's authentic public key (p, g, y_i).
2. Represent the message as an integer m in the range $\{1, \ldots, p-1\}$.
3. Select a random integer $k, 1 \leq k \leq p-2$.
4. Compute $a = {y_i}^k \bmod p$ and $b = m \cdot g^k \bmod p$.
5. Send the ciphertext $c = (a, b)$ to i.

Algorithm D: Decrypting a ciphertext $c = (a, b)$ sent to user i:

1. Use the private key x_i to compute $g^k = a^{1/x_i} \bmod p$ where the fraction in the exponent is computed $\bmod(p-1)$.
2. Recover m by computing $b/g^k \bmod p$.

Vericosine, a spin-off company of a Verisine, has decided to go into the ElGamal universal translator business. You can assume all parties share and agree upon securely the same p and g values. In your solution, you must state any other properties you require of p and g that are not specified above.

(a) Warm up

Several of Vericosine's customers use this system for encryption. A bug in the implementation of step 3 in E caused k to be a deterministic function of only the message. Unfortunately, the customers cannot upgrade their encryption (it was apparently built into the SSC hardware of a Palladium chip). Thus, if a sender encrypts the same message to two different recipients, an adversary can detect that the messages are the same.

Seeing a new source of revenue, Vericosine explains that users could send their ciphertexts (over a secure connection) to a re-randomizer before sending the final ciphertext over an untrusted network to the recipient. This could be implemented in software near the chip.

Using the notation above, explain how Vericosine can re-randomize ciphertexts generated by E. That is, construct a function R that takes as input the ciphertext (a, b) to produce a uniformly re-randomized ciphertext (a', b') such that $(a, b) \neq (a', b')$ and $D(R(a, b)) = D(a, b)$. Explain why your re-randomization works. You can assume that the re-randomizer has a source of randomness. The re-randomizer does not have access to any user secrets. Make sure to specify the necessary security properties of anything you use.

(b) Translation

Vericosine, tired of the mere re-randomization business, decides to enter the burgeoning ciphertext translation business. Vericosine decides that if it obtains the quotient $t_{j \leftarrow i} = x_i/x_j \bmod (p-1)$, then it can translate ciphertexts originally destined to user i into ciphertexts decryptable by user j.

Construct the translation function $T(a_i, b_i, t_{j \leftarrow i})$ which takes as input a ciphertext for user i and the translation quotient from i to j to produce a new ciphertext (a_j, b_j) which only user j can decrypt to the original message.

Explain why your scheme works. In particular, what property of this variant of ElGamal is necessary for your scheme to work?

(c) Divided between preparations

The Technical Advisory Board of only Vericosine (TABOO-V) proposes two methods of establishing the translation quotient on the Vericosine translation server, T. Below, the $\longrightarrow$ symbol means "sends to":

Preparation Un:

$i \longrightarrow j : x_i$
$j \longrightarrow T : x_i/x_j$

Preparation Deux:

$i \longrightarrow T : x_i$
$j \longrightarrow T : x_j$
(T then computes $x_i/x_j \bmod (p-1)$)

Compare and contrast these two preparations. What are the security ramifications? What are the trust models? Some issues you may consider include resiliency of the system to disclosure of secrets or partial secrets, collusion between various parties, the assumptions about T, the addition of more users, and intrusion tolerance. Use the notion i, j, T, W to represent user i, user j, the translator, and the rest of the world respectively.

(d) Enter the consultant

Gwen Bytediddle[1], a well respected and paid consultant, points out that the scheme could be strengthened by making private keys available to fewer parties. Unfortunately, Gwen was called away to fix a leaky encrypted pipe before she could finish. She left a hint that the two users should first agree upon a random number $r \in Z_{p-1}$.

(1) Using Gwen's hint, construct a "preparation trois" such that Vericosine can translate ciphertexts, but private keys such as x_i and x_j are minimally exposed when all parties are honest.

(2) Explain the new trust model and argue briefly why preparation trois is secure when all parties are honest.

(3) What problems from part (c) remain if parties collude?

(e) Bonus

For 1 point, explain in one sentence how the two companies can become one.

[1] A distant relative of Ben Bitdiddle

Problem Q-5. Survey [5 pts]

On a single page, answer both of the following questions:

(a) Describe your favorite material in 6.857. Why is it your favorite?

(b) What material or paper not yet presented in class would you most like to discuss in 6.857? Why? Limit your suggestion to a single topic. `http://web.mit.edu/6.857/www/references.html` may offer some help.

Problem Q-6. Academic honesty [5 pts]

Write a couple sentences to testify that you have not collaborated with anyone on this midterm and that you have cited all your sources. Affix your pretty signature to this statement.