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

Handout 7
September 19, 2002

Problem Set 2

This problem set is due on *Thursday, September 26, 2002* at the beginning of class. Late homeworks will *not* be accepted. This entire problem set is a group problem set. You must, however, continue to cite your sources properly. At the end of each problem, tell us how each person contributed (writing, designing, coding, proof reading, never showed up, etc). There is a fair amount of freestyle programming involved; start early!

Problem 2-1. One-Time Pads

The *Digitarian* government encrypts confidential messages to its embassies using the following system:

- The message (which begins as English text consisting only of uppercase letters, digits, spaces, colons and periods) is first encoded by representing each character as a pair of decimal digits:

0	→	00
1	→	01
	. . .	
9	→	09
space	→	10
A	→	11
	. . .	
Z	→	36
:	→	37
.	→	38

  Each digit is then represented by its ASCII (8-bit) character (0x30 – 0x39).
- The message is then "XOR-ed" with a "one-time pad" of the same length to form the ciphertext. The ciphertext is then represented as a sequence of 3-decimal-digit values, each value between 0 and 255.

Unfortunately for the *Digitarians*, the person responsible for making up the one-time pads is secretly an agent for the opposing *Binarians.* He distributes the one-time pads so that each is duplicated and sent to the embassies in such a way that the embassies will end up using each "one-time pad" several times, rather than just once.

Wednesday 25 September 2002, 7:52 pm, *Binarian* counter-espionage HQ . . .

Jand Bomes, your best agent has just died in mysterious circumstances. However, just before his death, he managed to communicate to you the following critical information about how the *Digitarian* undercover agents arrange their meetings:

- To avoid sending a suspicious message to an agent, all communications are posted on a public forum, in an encrypted form.
- In all communications, all agents and locations are referred to using codenames.

- The messages exchanged are short, and follow a strict syntax. A typical message would be the concatenation of fields containing the critical information (sender and recipient of message, date of message, date/time and location of meeting). If any data is shorter than the space allocated for it, it is padded with spaces to fit the allocated space.
- To arrange a meeting, the two parties follow a strict protocol:
  - first, the requester broadcasts a message addressed to the recipient with blank fields at the place of the meeting information;
  - then, the recipient broadcasts back a message addressed to the requester the same day as the meeting (or the day before) indicating the meeting details.
- The messages only use the characters allowed in the *Digitarian* encryption system. Besides, to make them harder to break, they are "rotated randomly"[1]before applying the standard *Digitarian* encryption scheme. A "random rotation" consists of a circular permutation of the characters: instead of starting with the first character, you pick a starting character at random, then enumerate all the subsequent characters until the end of the original message, at which point you continue at the beginning; you stop when you arrive at your starting point.

A preliminary analysis of the numerous messages on the public forum has permitted you to isolate 24 messages believed to be dealing with meeting arrangements.

The 24 ciphertext messages are available in the course locker in:

```
attach 6.857
cd /mit/6.857/extra/otp/
```

(a) Explain how you can determine whether two messages have been encoded with the same "one-time pad".

How many pads have been used for these 24 messages? Divide the messages into groups that used the same "one-time pad".

(b) Before passing away, Jand Bomes discovered that "Mary" and "Alice" had a secret meeting of critical importance two weeks earlier in "Copenhagen".

You know "Paul" is eager to meet "Mary" to discuss the aftermath of the decisions taken at the meeting. Where and when (date/time) are they meeting? Can you intercept them before the meeting?

Explain how you reached your conclusion.

(c) After you find out the "Paul"-"Mary" meeting passed, you remember that "Alice" has to report to "Bob" about the meeting she had with "Paul". This meeting is imminent. Will you discover the place and date/time of the meeting in time?

[1]A "random rotation" of the message `abcdefghij` would be `ijabcdefgh` or else `defghijabc`.

General remarks:

- You should develop a program for:
  1. determining when two ciphertexts were produced using the same one-time pad.
  2. assisting a human operator in deciphering the plaintexts that were so encrypted.
- The more elaborate your program is, the less trial and error you need to do. Let the computer do the dirty work!
- Feel free to look at the solutions for similar problems from previous years.

Hints:

- The more messages you possess that have been encrypted with the same "one-time pad", the easier it is to decipher one of them.
- Remember that the protocol implies that for each meeting, two messages are broadcast, one of them having lots of blanks.

Problem 2-2. Hashes do grow in trees, you know

Here is a proposal for a signature scheme that blends together Merkle's tree authentication and Lamport's one-time signature scheme. Let h be a suitable hash function that maps strings of arbitrary length to 160-bit values.

Let t be a suitable parameter; let's assume that $t = 28$ in this problem.

The signer will be able to authenticate a sequence of 2^{t-2} one-bit values. By consecutively authenticating 160 such one-bit values, he can authenticate the hash of an arbitrary message, thus signing the message.

The scheme starts by having the signer create a Merkle tree of depth t. This has a root $X = x_\epsilon$ (here ϵ is the empty string) and 2^t leaves, each of which has a 160-bit value. The value of a node x_j is the hash of the concatenation of the values of its two children x_{j0} and x_{j1}. The values of the leaves may be chosen randomly. The signer publishes the root X as part of his public key, but remembers and keeps secret all other values (for now).

(a) Let s be a binary string of length t. There is a unique corresponding leaf x_s. What values in the tree should the signer publish in order to reveal x_s and prove it is part of the tree rooted at X? How many hash function evaluations are required on the part of the verifier to check this proof? How long is the proof (how many bytes)?

Let v_i be the i-th one-bit value to be authenticated. Let w_i be the string i (represented with exactly $t-2$ bits) concatenated with the one-bit v_i, concatenated with the single bit "0". Thus w_i is a binary string of length t:

$$w_i \quad = \quad i||v_i||0$$

The signer can now publish a proof of v_i as the i-th value; the proof consists of i, v_i, and the proof that x_{w_i} is in the tree rooted at X.

(b) Why do we force w_i to end with a zero bit?

Now to provide a real "digital signature" of a message M, the signer first computes $h(M)$, and breaks that into a sequence of 160 bits. He can then authenticate these values as the next 160 one-bit values to be authenticated, in the above manner. (Note that there is a counter i that the signer must keep track of, counting the number of one-bit values authenticated so far.)

(c) How long is the signature? How many hash function evaluations are required to check the signature? (Give upper and lower bounds.)

(d) What properties should the hash function have?

(e) How do signature size and verification speed compare with 1024-bit RSA with small public exponent size? (You can find suitable information on the Web...)

Problem 2-3. Block cipher security

In 1996, a group of cryptographers (affectionately known as the Magnificent Seven) wrote a document explaining the inadequacies of block cipher key sizes.[2] At the time, 40-bit DES keys were commonplace because of extremely restrictive U.S. export laws. In this problem, we hope you will discover how the security requirements of block ciphers change both quantitatively (e.g., key size) and qualitatively (e.g., manner of application) over time.

Treat this problem as serious expository writing. It should be clear, well-written, and focused. We assume you know about standard writing practices (a thesis sentence, intro paragraph, body paragraphs, concluding paragraph). Think creatively and write a convincing argument. Who knows, some day you will give Congressional testimony on such a topic!

(a) Key size

To increase key size from 56 bits to 112 bits, one could naively encrypt a plaintext message twice such as $DES_{k_1}(DES_{k_2}(M))$. There is, however, a meet-in-the-middle attack to break this scheme with a mere 2^{57} expected work factor instead of 2^{111}. To thwart this attack, Triple-DES encrypts a plaintext with two 56-bit DES keys, but with an encrypt-decrypt-encrypt method as in $DES_{k_1}(DES^{-1}_{k_2}(DES_{k_1}(M)))$. AES, the replacement for an aging DES, comes in modes with 256-, 192-, and 128-bit keys. This is a vast increase over the 56-bit DES keys. But why design an entirely new algorithm instead of simply re-encrypting multiple times to achieve longer key lengths? After all, DES has stood the test of time, but AES may have subtle faults not yet discovered. Why assume the new risk? In your essay, address these questions by explaining why we have changed to brand new, possibly insecure ciphers instead of simply re-encrypting with established ciphers that have stood the test of time. Or if you disagree with this trend, explain your position. Offer 2-3 clear points to justify your argument. Keep your essay to under one page in length with a reasonable font. Here are some suggestions to help you form an opinion:

- What has qualitatively changed since the 1996 paper on symmetric key sizes? Consider such factors as Moore's law, storage capabilities, new applications, and lifetime of data. What other factors are there?
- Sometimes technology does not scale well. To go from small keys to large keys may require a completely new paradigm. What causes such a change?

[2] Handout 8

- In 1998 the Electronic Frontier Foundation built a DES cracking machine for US\$210,000, \$80,000 of which paid for design, integration, and testing.[3] The remaining \$130,000 paid for components. After 18 months of work, the team created a machine that could recover a DES key under various attack models in as few as 4.524 days. The EFF is a civil liberties group; imagine what a wealthy, terrorist-hunting government could accomplish.

(b) $\mathcal{P} = \mathcal{NP}$ or $\neq \mathcal{NP}$. That is the question.

After attending lecture 4 and accepting Prof. Rivest's challenge, Ben Bitdiddle decides to write a master's thesis on the cryptographic ramifications of discovering that $\mathcal{P} = \mathcal{NP}$. One of Ben's conclusions is that the probability of finding that $\mathcal{P} = \mathcal{NP}$ is more than the probability of recovering a 128-bit block cipher key in a known plaintext attack. Ben concludes that 128-bit key sizes are therefore too large. Is he right or wrong? Offer 2-3 clear points to justify your argument. Keep your essay to under one page in length with a reasonable font.

In his Ph.D. thesis proposal, Ben hopes to write about the effects of $\mathcal{P} = \mathcal{NP}$ on asymmetric cryptography. But that's another story, so don't discuss public key cryptography in your essay.

Here are some suggestions to start your thinking:

- We learned that cryptography should be strong enough to not be the weakest link in a security system. But have we gone too far?
- Does Ben's assignment of a probability of discovering $\mathcal{P} = \mathcal{NP}$ make sense? If not, argue for another quantification or method of comparison.
- Think about risk assessment.
- In making a standard, a single recommendation must cover many situations and therefore be somewhat excessive. Yet it should not cause significant computational pain to users with less than modern computers.

[3]Electronic Frontier Foundation. *Cracking DES*, O'Reilly and Associates, May 1998.