February 23, 2000

Diffie-Hellman practice

Silly (multiplicative) Diffie-Hellman

You are Eve, sitting half-asleep in the corridor outside the cells of Alice and Bob, and listening to Alice and Bob. You hear the following dialog:

Bob: Oh, come on, we'll just multiply, it's easier than exponentiating!

Alice: Oh, fine. *Maybe* we can compensate by using a rather large prime so that *she* [INDICATING EVE] will have trouble. We'll use **1211211211211**.

Bob: That's such a lovely prime! Why don't we start with another neat number like **842842842**.

Alice: So here's my result: **763030992727**.

Bob: And here's mine: **370114094368**.

Bob: Now, without much work, we can get a common key ...

Alice: I'm still worried! [GLANCING TOWARDS A DROWSY EVE] Well, convert the letters in your message to an integer using **A**$\to$**01**, **B**$\to$**02**, ..., **Y**$\to$**25**, and **Z**$\to$**26**, and then add the result, mod **1211211211211**, to our common key.

Bob: No problem, no problem. You're being silly. After all, Eve's just a **794453124413** ... [LAUGHS LOUDLY.] Hah, hah, hah.

[THIS NOISE WAKES UP EVE COMPLETELY. SHE SNIFFS, THINKS FOR A FEW SECONDS, AND THROWS A BUCKET OF COLD WATER ON BOB.]

True (exponential) Diffie-Hellman

Again, you are Eve, listening to Alice and Bob. You hear the following dialog:

Alice: We need a prime.

Bob: How about **3**?

Alice: No, no, a big prime!

Bob: How about **3**?

Alice: Nitwit. She's sitting there, our executions are in the morning, and the sun is beginning to make the sky glow faintly.

Bob: O.k., I'll consult an oracle ... let's use **120001201**, a lovely prime.

Alice: And we'll use **2** as the base for our powers. Let's get to work.

[TIME PASSES.]

Bob: Here's mine: **75056128**.

Alice: I've got **60616573**.

[MORE TIME PASSES.]

Alice: Done!

Bob: Me, too.

Alice: Convert the letters in your message to an integer using **A**$\to$**01**, **B**$\to$**02**, ..., **Y**$\to$**25**, and **Z**$\to$**26**, and then add the result, mod **120001201**, to our common key.

Bob: So I say to you, **95368077**!

Eve: [SNARLS]

After you finish reading and enjoying these magnificent dramatic scenes, please consider your homework assignment on the other side of this page. Golly, the true human drama and depth of characterization shown here is remarkable.

Homework/Contest

Of course I want you to discover the messages that Bob has sent. Note that there is a discussion of Diffie-Hellman key exchange in section 5.4 (pages 120–124) of the text. You may work alone or in a group of at most 4 students.

1. Send me e-mail with either message (or both). Your message should contain a list of the people who worked on the problem, a short description of the strategy that you used to decode the messages, and should also have some accounting of what computing power (and how much computing time) you needed to discover the message.

2. a) The first person/group to send me Bob's exponential message decoded properly accompanied by the description requested above wins **First Prize with reward**.

b) The first person/group to send me Bob's multiplicative message decoded properly accompanied by the description requested above wins **Second Prize with reward**.

To help distribute the rewards more equitably

No winning group may contain more than one (1) person from a previous winning group!!!

Some useful `Maple` comments

I remind you that `Maple` can remember numbers if you tell it to. The statements

key:=12345; number:=55511;

will cause `Maple` to remember 12345 and 55511 with the names designated. This is useful if you are similar to me: my error rate in typing (and retyping!) numbers is high.

`Maple` can also do modular arithmetic. One way is to use the *mod* command.

42 mod 5;

The result is just 2 (the remainder after dividing 42 by 5). Using *mod* and labeling numbers is likely to really save time and effort. For example, with the values of *key* and *modulus* mentioned above, the command

*frog:=72*key^2 mod number;*

produces the answer 36963 and assigns it the name *frog*. `Maple` still has the values of *key* and *modulus* (and *frog* also!) to work with later.