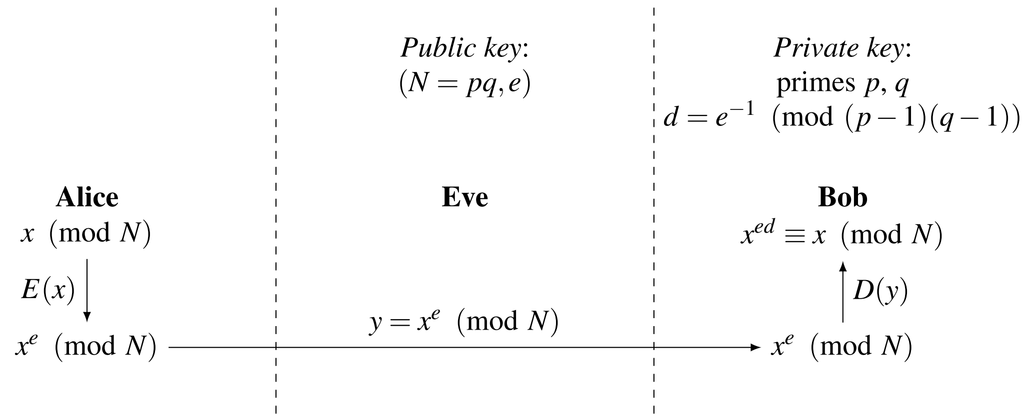
CS 70 Discrete Mathematics and Probability Theory
Spring 2025 Rao **DIS 4A**

1 RSA Intro

Note 7

Fermat's Little Theorem: For all primes p, $a^{p-1} \equiv 1 \pmod{p}$ if $a \neq 0$, and $a^p \equiv a \pmod{p}$ for all a.

RSA Scheme:



(a) Evaluate $3123^{30} \pmod{31}$.

(b) Suppose we would like to evaluate $141^{161} \pmod{187}$.

(i) First, evaluate $141^{161} \pmod{11}$ and $141^{161} \pmod{17}$. Use the results of those computations to evaluate $141^{161} \pmod{187}$.

(ii) Alternatively we can evaluate $141^{161} \pmod{187}$ by thinking of the computation as an instance of the RSA equation $x^{ed} \equiv x \pmod{pq}$. What are p, q, e, and d? What is the final result of the computation? (*Hint: We know that* $187 = 11 \times 17$ *and* $161 = 23 \times 7$.)

2 RSA Warm-Up

Note 7

Consider an RSA scheme with modulus $N = pq$, where p and q are distinct prime numbers larger than 3.

(a) What is wrong with using the exponent $e = 2$ in an RSA public key?

(b) Now suppose that $p = 5$, $q = 17$, and $e = 3$. What is the public key?

(c) What is the private key?

(d) Alice wants to send a message $x = 10$ to Bob. What is the encrypted message $E(x)$ she sends using the public key?

(e) Suppose Bob receives the message $y = 19$ from Alice. What equation would he use to decrypt the message? What is the decrypted message?

(f) In RSA, we rely on the hardness of two different problems in order to guarantee the security of the scheme. Which two problems are these? If their hardness is not guaranteed, what goes wrong?

3 RSA with Multiple Keys

Note 7

Members of a secret society know a secret word. They transmit this secret word x between each other many times, each time encrypting it with the RSA method. Eve, who is listening to all of their communications, notices that in all of the public keys they use, the exponent e is the same. Therefore the public keys used look like $(N_1, e), \ldots, (N_k, e)$ where no two N_i's are the same. Assume that the message is x such that $0 \leq x < N_i$ for every i.

Further, in all of the subparts, you may assume that Eve knows the details of the modified RSA schemes (i.e. Eve knows the format of the N_i's, but not the specific values used to compute the N_i's).

(a) Suppose Eve sees the public keys $(p_1q_1, 7)$ and $(p_1q_2, 7)$ as well as the corresponding transmissions. Can Eve use this knowledge to break the encryption? If so, how? Assume that Eve cannot compute prime factors efficiently. Think of p_1, q_1, q_2 as massive 1024-bit numbers. Assume p_1, q_1, q_2 are all distinct and are valid primes for RSA to be carried out.

(b) The secret society has wised up to Eve and changed their choices of N, in addition to changing their word x. Now, Eve sees keys $(p_1q_1, 3)$, $(p_2q_2, 3)$, and $(p_3q_3, 3)$ along with their transmissions. Argue why Eve cannot break the encryption in the same way as above. Assume $p_1, p_2, p_3, q_1, q_2, q_3$ are all distinct and are valid primes for RSA to be carried out.

(c) Let's say the secret x was not changed ($e = 3$), so they used the same public keys as before, but did not transmit different messages. How can Eve figure out x?

4 RSA for Concert Tickets

Note 7

Alice wants to tell Bob her concert ticket number, m, which is an integer between 0 and 100 inclusive. She wants to tell Bob over an insecure channel that Eve can listen in on, but Alice does not want Eve to know her ticket number.

(a) Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's ticket number is. How did she do it?

(b) Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m? (You may assume that r is coprime to N.)