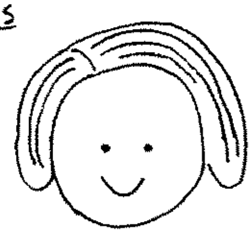
Math Access Notes
Thursday June 14

Public Key Cryptography with Secure Signature

numbering as in Davis' notes

Alice sets up to receive messages



A

1. Alice picks to large primes
   p, q
2. Alice computes her modulus
   $N := pq$

3a. She chooses the encryption power e relatively prime to $(p-1)(q-1)$

$$[\ \gcd(e, (p-1)(q-1)) = 1\ ]$$

3b. $\boxed{\text{The modulus } N \text{ and power } e \text{ are Alice's public key}}$
Anyone wishing to send Alice a "message", i.e a residue $0 \le x < N$, first encrypts it using the function

$$E(x) :\equiv x^e \mod N$$

7. Alice, because she knows number theory and p and q, can find her decryption power d. She solves the multiplicative inverse equation

$$ed \equiv 1 \mod (p-1)(q-1)$$

She will decrypt messages with the function

$$D(y) :\equiv y^d \mod N$$

$$\text{since } D(E(x)) \equiv D(x^e) \equiv (x^e)^d = x^{ed} = x^{1 + k(p-1)(q-1)}$$

$$\equiv x \mod N \text{ by Fermat-Euler's Thm}^{*}$$

(the sender's original message!)

$*$ The version of Euler's Thm we use says that

$$x^{(p-1)(q-1)+1} \equiv x \mod pq$$

You can see this works when $pq = 3 \cdot 5 = 15$, by looking at the power table mod 15, at the end of Wednesday's notes.
In that case $(p-1)(q-1)+1 = 2 \cdot 4 + 1 = 9$.

This version also implies

$$x^{1+2(p-1)(q-1)} \equiv \underbrace{x^{1+(p-1)(q-1)}}_{x}\, x^{(p-1)(q-1)} \equiv x^{1+(p-1)(q-1)} \equiv x \mod N$$

etc.

Bob sends a message

4a. Bob wishes to send a message to Alice. He converts it into numbers x, using a conversion key like Dan's.

4b. Secure signature: Bob has created his own public key: e_B, N_B

and private key: d_B
He thinks of a sensible "signature", makes it numeric, s_B, and <u>decrypts</u> it using is private key,
$$D_B(s_B)$$

5. Bob appends x to $D_B(s_B)$, creating,
$$x \divideontimes D_B(s_B)$$
(breaks this into blocks $< N_A$) and encrypts using Alice's public key
$$y = E_A(x \divideontimes D_B(s_B))$$

6. Bob sends y to Alice, e.g. over the internet

8. Alice wishes to decode the message y
She computes
$$D_A(y) = D_A(E_A(x \divideontimes D_B(s_B)))$$
$$= \underset{\text{message}}{x} \divideontimes \underset{\text{gibberish}}{D_B(s_B)}$$

9. Alice uses Bob's public key to compute
$$E_B(D_B(s_B)) = s_B,$$
Bobs signature!
[Only Bob could make $D_B(s_B)$!]

Doesn't know D_A so can't get to $x \divideontimes D_B(s_B)$;
so can't read the message <u>and</u> can't forge messages to Alice which look like they came from Bob.