Diffie-Hellman

Key issue of using symmetric key crypto

- Question: what is the key challenge of using symmetric key crypto in practice?

Key issue of using symmetric key crypto

- Question: what is the key challenge of using symmetric key crypto in practice?

Users

Key distribution

Businesses

amazon.com

Overview of Diffie-Hellman

- Invented by Malcolm Williamson (GCHQ, British Equivalent of NSA) and, independently, by Diffie and Hellman (Stanford)
  - **Diffie and Hellman won ACM Turing award for this!**

- A **"key exchange" algorithm**
  - Used to establish a shared symmetric key

- ***Not* for encrypting or signing**

Based on the discrete logarithm algorithm

- Based on **discrete log** problem, which is believed to be difficult:
  - **Given:** g, p, and $g^k \bmod p$
  - **Find:** exponent k
    - For example, in real numbers, log2(8)=3 because $2^3=8$
    - But for discrete log, finding the k is not feasible to do
- Example
  - Question 1: $g = 2$, $p = 17$, $(g^k \bmod p) = 13$. What is k?

Computational cost

- We can try: $k = 0, 1, \ldots,$ until $g^k \bmod p = 13$
  - $\mathbf{g^k \bmod p}$ has **p** possibilities

- We may have to try (possibly more than) p times to find all possible $g^k \bmod p$
  - $\mathbf{2^0 \bmod 17 = 1}$, $2^1 \bmod 17 = 2$, $\mathbf{2^2 \bmod 17 = 4}$, $2^3 \bmod 17 = 8$
  - $2^4 \bmod 17 = 16$, $2^5 \bmod 17 = 15$, $\mathbf{2^6 \bmod 17 = 13}$
  - $2^7 \bmod 17 = 9$, $\mathbf{2^8 \bmod 17 = 1}$, $2^9 \bmod 17 = 2$, $\mathbf{2^{10} \bmod 17 = 4}$

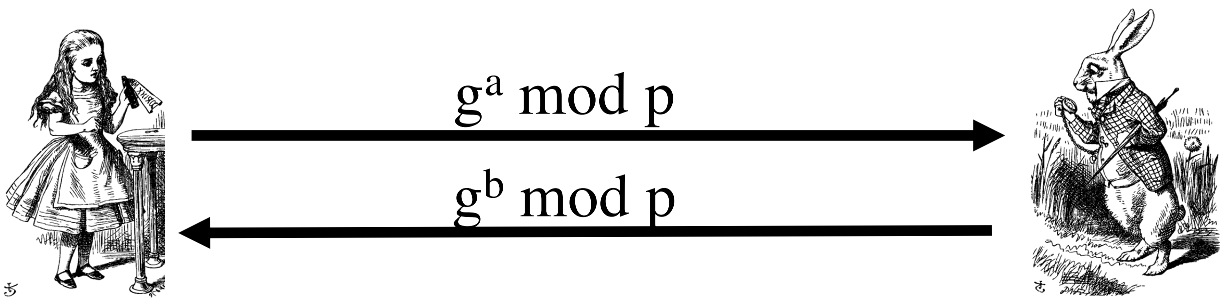
- The algorithm is **linear with p,** but it's **exponential** in terms of **the number of bits** needed to represent p
  - 1024 bits for p

Diffie-Hellman procedure

- Let p be prime, let g be a **generator**
- Alice selects her private value a
- Bob selects his private value b
- Alice sends $g^a \bmod p$ to Bob
- Bob sends $g^b \bmod p$ to Alice
- Both compute shared secret, $g^{ab} \bmod p$
- Shared secret can be used as symmetric key

Diffie-Hellman

- **Public:** g and p
- **Private:** Alice's exponent a, Bob's exponent b

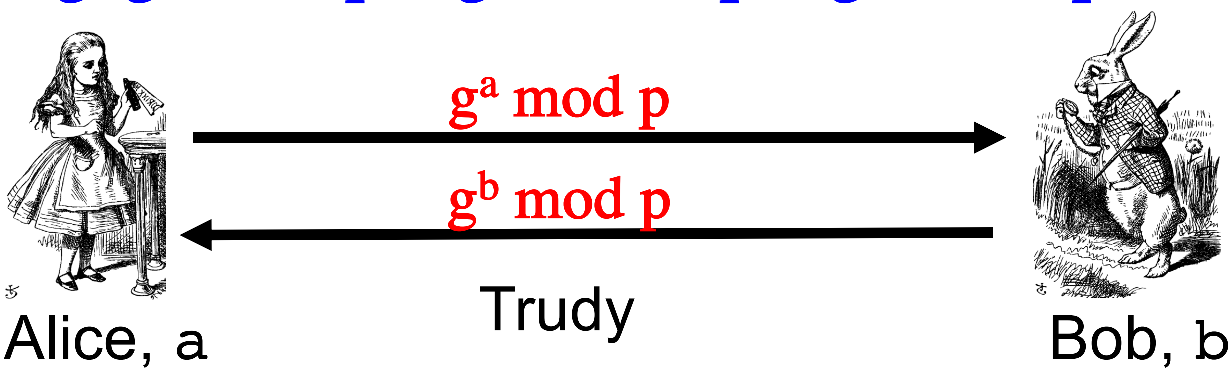


Alice, a　　　　　　　　Bob, b

- Alice computes $(g^b)^a = g^{ba} = g^{ab} \bmod p$
- Bob computes $(g^a)^b = g^{ab} \bmod p$
- Use $K = g^{ab} \bmod p$ as symmetric key

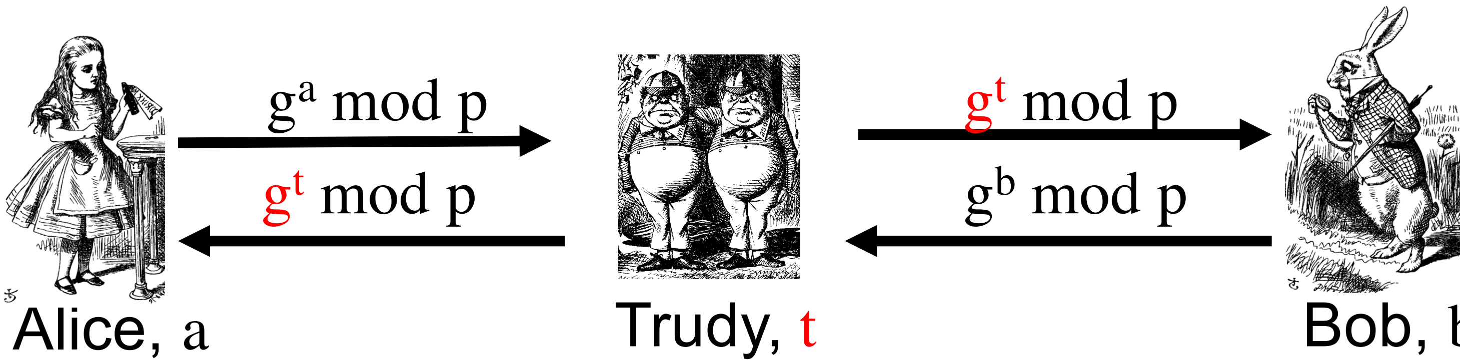
Can Trudy break Diffie-Hellman?

- Suppose Bob and Alice use Diffie-Hellman to determine symmetric key $K = g^{ab} \bmod p$
- Trudy can see $g^a \bmod p$ and $g^b \bmod p$
  - But… $g^a g^b \bmod p = g^{a+b} \bmod p \neq g^{ab} \bmod p$



Diffie-Hellman MiM attack

- Subject to **man-in-the-middle (MiM) attack**



- Trudy shares secret g^{at} mod p with Alice
- Trudy shares secret g^{bt} mod p with Bob
- Alice and Bob don’t know Trudy exists!

How to prevent MiM attack?

- How to prevent MiM attack?
  - Encrypt DH exchange with symmetric key
  - Encrypt DH exchange with public key
  - Sign DH values with private key
- At this point, DH may look pointless…
  - …but it's not (more on this later)
- In any case, you **MUST** be aware of MiM attack on Diffie-Hellman

El Gamal Cryptosystem

El Gamal Cryptosystem

- Invented by Tather El Gamal in 1984
- Based on the discrete logarithm problem
  - **Given:** $\mathbf{g}$, $\mathbf{p}$, and $\mathbf{g}^{\mathbf{k}} \bmod \mathbf{p}$
  - **Find:** exponent $\mathbf{k}$
- Example
  - <u>Question 1</u>: g = 2, p = 17, g^k mod p = 13. What is k?

El Gamal - Key generation

- Alice generates the public/private key pair as follows:
- (1) Generate large prime **p** and generator **g**
- (2) Select a **random** integer **a**, where $1 <= a <= p - 2$, and compute $g^a \bmod p$
- (3) Alice's public key is $(p, g, g^a \bmod p)$, and her private key is a.
  - **Discrete logarithm problem**: hard to find **a** from $\mathbf{g^a \bmod p}$

El Gamal - Encryption

- Bob encrypts a message m to Alice

- (1) Obtain Alice's public key $(p, g, g^a \bmod p)$
- (2) Represent message m in the range $\{0, 1, \ldots, p - 1\}$
- (3) Select a random integer k, where $1 <= k <= p - 2$
- (4) Compute $c_1 = g^k \bmod p$ and $c_2 = \mathbf{m} * (g^a)^k \bmod p$
- (5) Bob sends ciphertext $c = (c_1, c_2)$ to Alice
  - **Discrete logarithm problem**: hard to find $\mathbf{k}$ from $\mathbf{g^k \bmod p}$

El Gamal - Decryption

- Alice receives ciphertext $c = (c_1, c_2)$ from Bob
- Alice recovers the plaintext by computing:
- $(c_1^{-a}) * c_2 \bmod p$
- where a is her private key
- Why it works with: $c_1 = g^k \bmod p$ and $c_2 = m * (g^a)^k \bmod p$?
- Because: $c_2 = m * (g^a)^k \bmod p$
- $c_2 = m * (g^k)^a \bmod p = m * c_1^a \bmod p$
- $m = c_2 * c_1^{-a} \bmod p$

El Gamal - Example

- Alice chooses her public key (17, 6, 7) and private key 5
  - Prime $p = 17$
  - Generator $g = 6$
  - Private key $a = 5$
  - Public key part: $g^a \bmod p = 6^5 \bmod 17 = 7$
- Bob encrypts her message $m = 13$
  - He chooses a random number $k = 10$
  - He calculates $c_1 = g^k \bmod p = 6^{10} \bmod 17 = 15$
  - He encrypts $c_2 = m * (g^a \bmod p)^k \bmod p = 13 * 7^{10} \bmod 17 = 9$
- Bob sends $(c_1, c_2) = (15, 9)$ to Alice
- Alice decrypts it by: $m = c_2 * c_1^{-a} \bmod p = 9 * 15^{-5} \bmod 17 = 13$
  - $15^{-1} \bmod 17 = 8$ because $8 * 15 = 120 = 17 * 7 + 1$

Quiz

- Suppose that we want to replace CBC with ECB or CTR for generating MAC. Which of the following is true?

- A: Only ECB can be used to replace CBC.
- B: Only CTR can be used to replace CBC.
- C: CBC can be replaced with either of ECB and CTR.
- **D: CBC can be replaced with neither ECB nor CTR.**

Quiz: (Check all answers that apply)

- Which of the following is correct about ECB (Electronic Codebook)?

- **A: Under ECB, blocks are encrypted independently**
- B: Under ECB, an IV is required
- **C: ECB suffers the cut and paste attack**
- **D: The same plaintext leads to the same ciphertext**

Quiz

- MAC (Message Authentication Code) can be achieved with the following mode:

- A: Electronic Codebock (ECB)
- **B: Cipher Block Chaining (CBC)**
- C: Counter Mode (CTR)
- D: None of the above

Quiz

- What is 5^{-1} mod 10?

- A: 1
- B: 2
- C: 3
- **D: Doesn't exist**
  - **5 has factors 1, 5**
  - **10 has factors 1, 2, 5, 10**
  - **More than 1 common factors**

Quiz

- Suppose that Bob's knapsack private key consists of (3,5,10, 23) along with the multiplier $m^{-1} = 6$ and modulus $n = 47$
- Find the plaintext given the ciphertext $C = 20$. Give your answer in binary.

- To decrypt,
  - $20 \cdot m^{-1} = 20 \cdot 6 = 26 \bmod 47$
  - Solve SIK with $S = 26$

    (3, 5, 10, 23)
  - Obtain plaintext 1 0 0 1

Elliptic Curve Cryptography(ECC)

Elliptic Curve Crypto (ECC)

- "Elliptic curve" is **not** a cryptosystem
- Elliptic curves are a different way to do the math in public key system
- Elliptic curve versions of DH, RSA, etc.
  - Compare to the exponential version
- Why would we want them if we already have DH and RSA?

Elliptic curves are more efficient

- Fewer bits needed for same security
- For example, a **256-bit ECC public key** should provide comparable security to a **3072-bit RSA public key**
- Faster than standard RSA
- Good for handhelds and phones

NIST recommended key sizes

Symmetric algorithm (bit)	RSA and DH (bit)	ECC (bit)
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

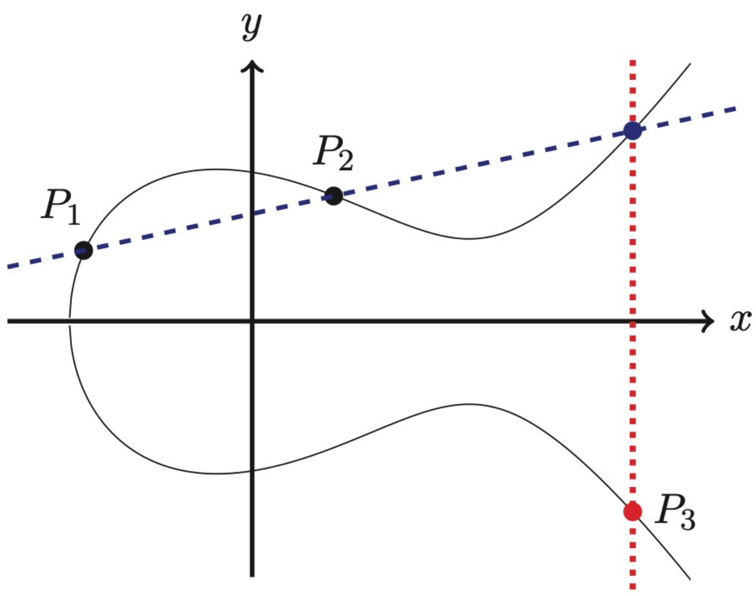
What is an Elliptic Curve?

- An elliptic curve `E` is the graph of an equation of the form

$$y^2 = x^3 + ax + b$$

- Also includes a “point at infinity”
- What do elliptic curves look like?

Elliptic Curve - Example



- Consider elliptic curve

  `E: y`^{2}` = x`^{3}` - x + 1`

- If P_1 and P_2 are on E, we can define addition,

  $P_3 = P_1 + P_2$

  as shown in picture
- Addition is all we need…

Points on Elliptic Curve

- Discrete version: $y^2 = x^3 + ax + b (\bmod p)$
- Consider $y^2 = x^3 + 2x + 3 \pmod 5$

$$
\begin{aligned}
x = 0 &\Rightarrow y^2 = 3 \Rightarrow \text{no solution} \pmod 5 \\
x = 1 &\Rightarrow y^2 = 6 = 1 \Rightarrow y = 1,4 \pmod 5 \\
x = 2 &\Rightarrow y^2 = 15 = 0 \Rightarrow y = 0 \pmod 5 \\
x = 3 &\Rightarrow y^2 = 36 = 1 \Rightarrow y = 1,4 \pmod 5 \\
x = 4 &\Rightarrow y^2 = 75 = 0 \Rightarrow y = 0 \pmod 5
\end{aligned}
$$

- Then points on the elliptic curve are
  (1,1), (1,4), (2,0), (3,1), (3,4), (4,0), and the point at infinity: ∞

BINGHAMTON UNIVERSITY STATE UNIVERSITY OF NEW YORK

Addition on Elliptic Curve

- Addition on: $y^2 = x^3 + ax + b \pmod{p}$

  $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$

  $P_1 + P_2 = P_3 = (x_3, y_3)$

  First: $m = (y_2 - y_1) * (x_2 - x_1)^{-1} \bmod p$, if $P_1 \neq P_2$

  $m = (3x_1^2 + a) * (2y_1)^{-1} \bmod p$, if $P_1 = P_2$

  Second: $x_3 = m^2 - x_1 - x_2 \pmod{p}$

  $y_3 = m(x_1 - x_3) - y_1 \pmod{p}$

  Special cases:

  Special point $P_3 = \infty$, and $\infty + P = P$ for all P

Elliptic Curve Addition Example

- Consider $y^2 = x^3 + 2x + 3 \pmod 5$.
- Points on the curve are (1,1), (1,4), (2,0), (3,1), (3,4), (4,0), and ∞
- What is $(x_1,y_1) + (x_2,y_2) = (1, 4) + (3, 1) = P_3 = (x_3, y_3)$?

  $m = (1-4)*(3-1)^{-1} = (-3)*2^{-1} = 2(3) = 6 = 1 \pmod 5$

  $x_3 = 1 - 1 - 3 = 2 \pmod 5$

  $y_3 = 1(1-2) - 4 = 0 \pmod 5$

- On this curve, (1,4) + (3,1) = (2,0)

First: $c = (y_2 - y_1) * (x_2 - x_1)^{-1} \bmod p$, if $P_1 \neq P_2$

$c = (3x_1^2 + a) * (2y_1)^{-1} \bmod p$, if $P_1 = P_2$

Second: $x_3 = c^2 - x_1 - x_2 \pmod p$

$y_3 = c(x_1 - x_3) - y_1 \pmod p$

ECC Diffie-Hellman

- **Public:** Elliptic curve and point (x,y) on curve
- **Private:** Alice's multiplier A and Bob's multiplier B

$$\xrightarrow{A(x,y)}$$

$$\xleftarrow{B(x,y)}$$

Alice, A

Bob, B

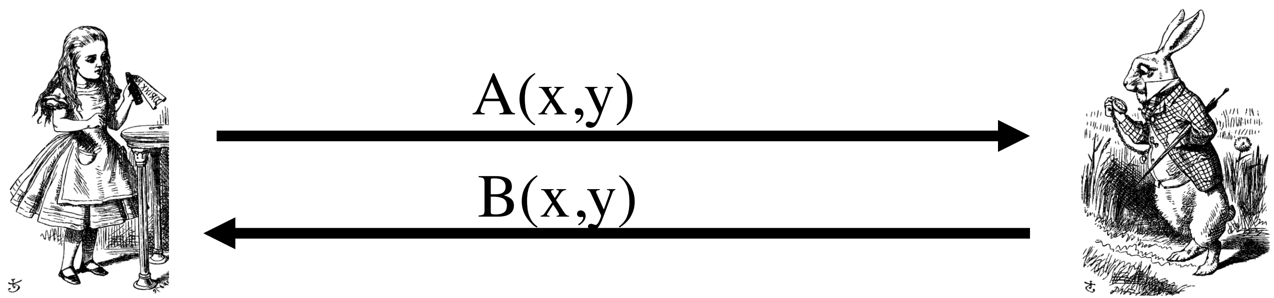
- Computes multiplication as repeated addition
- Alice computes $A(B(x,y))$
- Bob computes $B(A(x,y))$
- These are the same since $AB = BA$

ECC Diffie-Hellman Example

- **Public:** Curve $y^2 = x^3 + 7x + b \pmod{37}$ and point $(2,5) \Rightarrow b = 3$
- **Alice's private:** $A = 4$
- **Bob's private:** $B = 7$
- Alice sends Bob: $4(2,5) = (7,32)$
- Bob sends Alice: $7(2,5) = (18,35)$
- Alice computes: $4(18,35) = (22,1)$
- Bob computes: $7(7,32) = (22,1)$
- Both computes: $28(2,5)$

How to break ECC Diffie-Hellman?

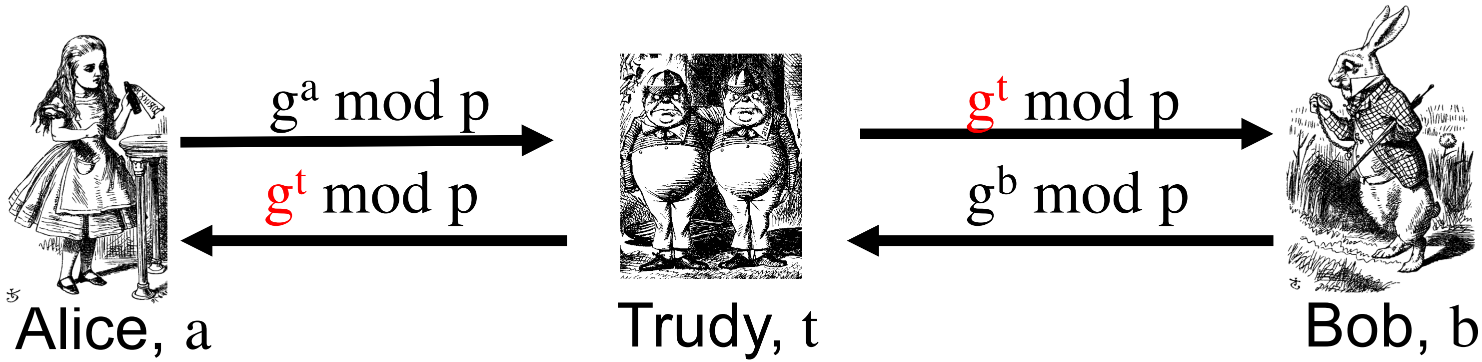
- **Public:** Elliptic curve and point (x,y) on curve
- **Private:** Alice's A and Bob's B



Alice, A   Bob, B

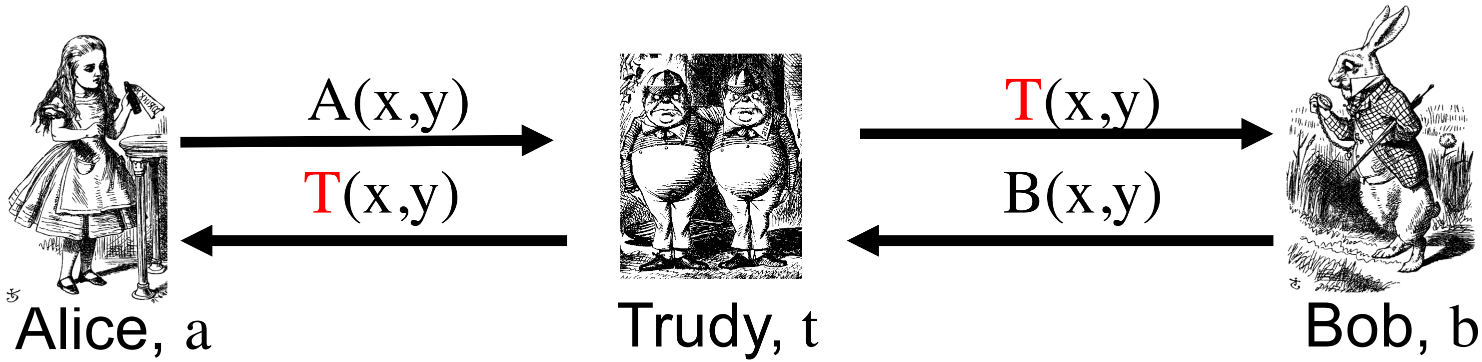
- Given A(x, y), Trudy would need to find A, or given B(x, y), find B.
- Both are difficult!

Recall that Diffie-Hellman suffers Man-in-the-middle attack



- Trudy shares secret g^{at} mod p with Alice
- Trudy shares secret g^{bt} mod p with Bob
- Alice and Bob don’t know Trudy exists!

Does ECC Diffie-Hellman also suffer man-in-the-middle attack?



- Trudy shares secret AT(x,y) with Alice
- Trudy shares secret BT(x,y) with Bob
- Alice and Bob don’t know Trudy exists!

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

Larger ECC Example

- Example from Certicom ECCp-109
  - Challenge problem, solved in 2002
- Curve $E: y^2 = x^3 + ax + b \pmod{p}$
- Where

```
p = 564538252084441556247016902735257
a = 321094768129147601892514872825668
b = 430782315140218274262276694323197
```

- Now what?

ECC Example

- The following point `P` is on the curve `E`

`(x,y) = (97339010987059066523156133908935,`
`149670372846169285760682371978898)`

- Let `k = 281183840311601949668207954530684`
- The `kP` is given by

`(x,y) = (44646769697405861057630861884284,`
`522968098895785888047540374779097)`

- And this point is also on the curve `E`

Really Big Numbers!

- Numbers are big, but not big enough
  - ECCp-109 bit solved in 2002
- Today, ECC DH needs bigger numbers
- But RSA needs ***way*** bigger numbers
  - Minimum RSA modulus today is 1024 bits
  - That is, more than 300 decimal digits
  - That's about 10x the size of ECC example
  - And 2048 bit RSA modulus is common…