Cryptography 4/22

Reagan Shirk

April 22, 2020

RSA in Sage

```
sage: p = ZZ.random_element(2^1020).next_prime()
sage: q = ZZ.random_element(2^1028).next_prime()
sage: e = 2^16 + 1
sage: gcd(e, (p-1)*(q-1))
1
sage: n = p * q
sage: "public key n and e"
'public key n and e'
sage: d = Integers((p-1)*(q-1))(e)^(-1)
sage: "secret key d"
'secret key d'
sage: m = 202004221041
sage: c = Integers(n)(m)^e
sage: Integers(n)(c)^d
202004221041
```

RSA Security

- Moral of the story: there are a shit ton of ways to break an RSA and that's why it's normally just used for signature and not for encryption/decryption
- We think that factoring n is harder than:
  - computing $\phi(n)$ from n
  - calculating d from n and e
  - calculating m from $n,\ e$, and the cipher
- What we want to do is prove that calculating m from $n,\ e$, and cipher is hard, but no body can prove it. . . ?
- On Friday, we'll prove that factoring n is equivalent two the first two things, but that no one can prove for the last one
- We want to prove that computing $\phi(n)$ from n is equivalent to factoring n
  - Observe that:
    $$p \times (q = n,\ (p-1)(q-1) = \phi(n)$$
  - p and q are unknown, n and $\phi(n)$ are known
  - You have two equations and two unknowns so you can solve the system of equations
  - This can be solved in polynomial time
- Proving that calculating d from n and e is equivalent to factoring n is harder than the proof above but it's still possible
  - If we can find n, e, and d we can factor n in polynomial time (poly(log(n)))
    - n and e are public, only d is unkown
  - Modern approach: find x such that $\gcd(x, n)$ is nontrivial (anything $\neq 1,\ n$ is considered nontrivial)

$$\forall a \in (\mathbb{Z}/n)^*,\ a^{ed-1} = 1(\mathrm{mod} n)$$

$$\gcd(a^{ed-1} - 1, n) = n$$

– What happens when we try $\gcd(a^{\frac{ed-1}{4}} - 1,\ n)$? This $\neq n$
– Alright I think he's lost me now