Digital Signature Algorithm (160-bit)

- **Alice – Key Creation**
  - Find 1024-bit prime p, 160-bit prime q where q divides $p-1$
  - Find $\alpha \in \mathbb{Z}_p^*$ where ord(α)=q
  - Choose random $0 < d < q$, compute $\beta \equiv \alpha^d \mod p$
  - Publish (p, q, α, β)
- **Alice – Sign message** x
  - Choose ephemeral $0 < k_E < q$
  - Compute

$$r \equiv \left(\alpha^{k_E} \mod p\right) \mod q$$

$$s \equiv (\mathrm{SHA}(x) + dr)\, k_E^{-1} \mod q$$

  - Send $(x, (r, s))$

- **Bob – Verify Signature**
  - Compute

$$w \equiv s^{-1} \mod q$$

$$u_1 \equiv w \cdot \mathrm{SHA}(x) \mod q$$

$$u_2 \equiv w \cdot r \mod q$$

$$v \equiv (\alpha^{u_1}\beta^{u_2} \mod p) \mod q$$

  - If $v = r$ then valid
    If $v \neq r$ then invalid

Specifications in FIPS PUB 186-4

https://csrc.nist.gov/publications/detail/fips/186/4/final

1. Pick a random 160-bit number q, check if prime, if not pick another q

2. Loop up to 4096 times (*Not sure why exactly 4096*)
   - Pick a random 1024-bit number m
   - Let $p = m - (m \mod 2q) + 1$ *(Say what?)*
   - If p prime, then we're done, if not repeat loop

3. If don't have prime pair (q, p), then go to Step 1 and try another q