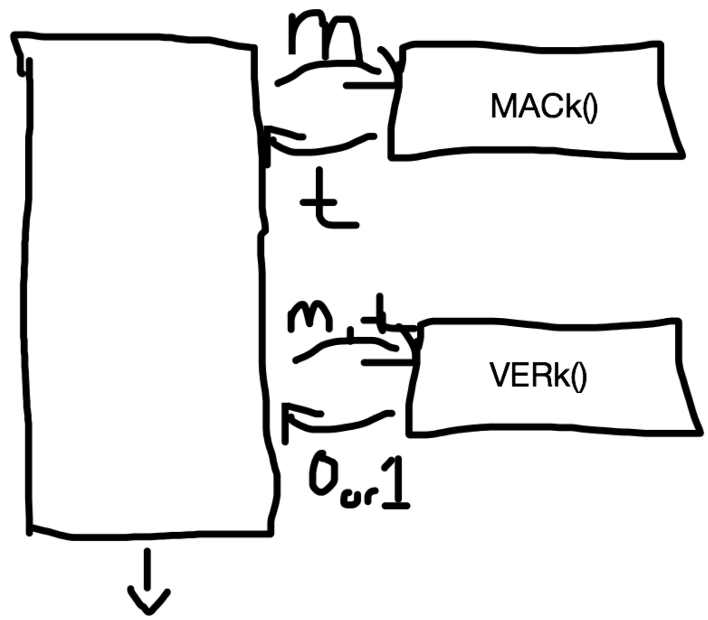
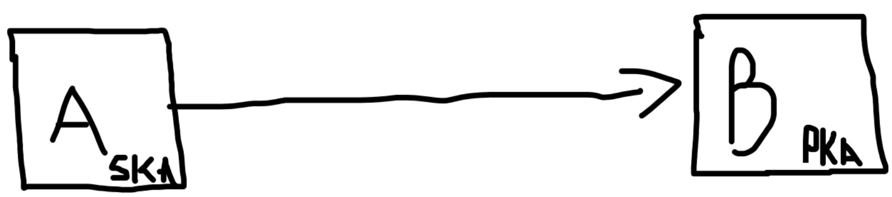
CS558 – 2/16/17

- **Hacking of the 2016 Election Presentation**
    - Timeline
        - June 2015: Cozy Bear Infiltrates DNC
        - April 2016: Second Fancy Bear Attack
        - July 2016: Wikileaks releases DNC emails/chats
    - Cozy Bear: APT 29
        - Emails that establish an encrypted communication with the target
        - Targeted systems will then constantly communicate with the adversary's servers even after termination
    - Fancy Bear: APT 28
        - Spear-phishing through spoofed web domains
        - Installs X-Agent onto the targeted system
    - Does it point to Russia?
        - Cyberstrike/JAR motivation for attribution
        - Software linked back to Russia
- **Public Crypto**
    - If Professor Goldberg wanted to talk with all of us in the class, she would need 60 different keys
        - If each of the classmates wanted to talk to each other, we would each have to have our own separate keys
            - Public crypto methods fixes this problem
    - MAC Security Review

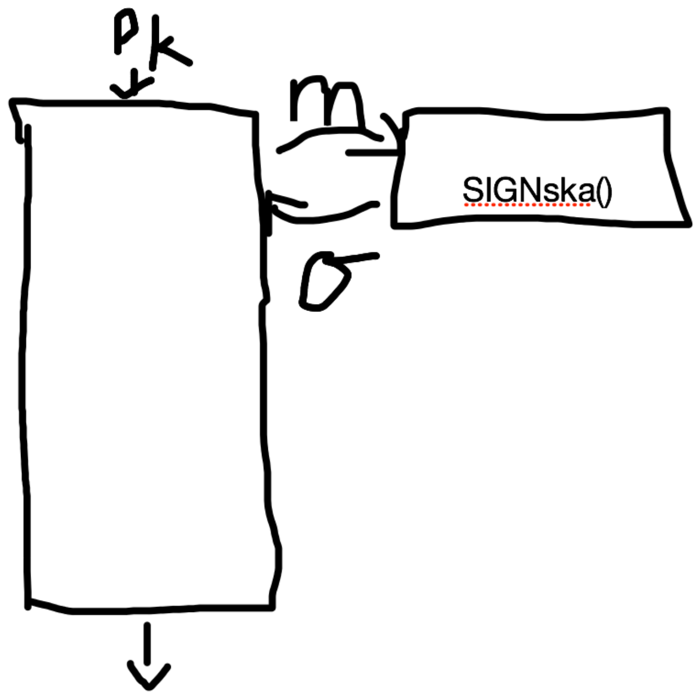


- m*, t* such that VERk(m*, t*) = 1
- Caveat: cannot query m*
- Adversary should not be able to win with probability higher than Pr[Verk(m*, t*) = 1] = 1/2^length
- Alice (mac,tag)→ Bob
    - t = MACk(m) → Verk(m, k) = 1

- Digital Signatures (Public Crypto)

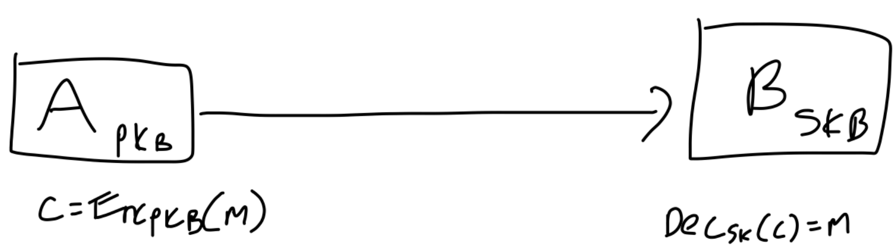


- O = SIGNska(m) Accept if VERpka(m, O) = 1

- 

- Public key held by verifier, secret key held by signer
    - A sends his/her secret key and B receives the message along with a signed sigma
        - Sigma allows the recipient to verify that the message is genuine
- Anyone who has the public key of A can decrypt A’s message and verify that the message came from A
- Correctness: Verpk (m, Sigsk(m)) = 1

o RSA Encryption (Public Crypto)

- 

- Allows a sender to send a message to a specific person or persons

- Sender A encrypts a message with the public key of B and sender B decrypts the message with their own secret key
    - This way, B is the only one who can decrypt the message
    - No way to verify the authenticity of Sender A
- How RSA works:
    - N = pq, where p and q are primes
    - e = encryption exponent
    - easy $(\underbrace{e, N}_{\text{Public key}}, m) \to \boxed{} \to m^e \bmod N$

      hard $(e, N, y) \to \boxed{} \to m$ s.t. $y = m^e \bmod N$
    - Encryption: ENCpkb(m) = m^e mod N
        - = [pad(m)]^e mod n
    - If you know the decryption exponent
        - d = e^(-1) mod ϕN
            - where ϕN = (p-1)(q-1)
            - y^d mod N = m
                - Where m such that y = m^e mod N
    - Public Key / Secret Key
        - Pk (N, e)
        - Sk = (N, d) or (p, q)