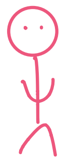
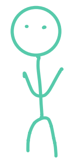
ASYMMETRIC KEY ENCRYPTION



PROBLEM: Alice doesn't have a way of verifying that Bob is the one sending her messages!

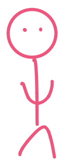


PK_A, SK_A

PK_B, SK_B

DIGITAL SIGNATURES

we add a PK/SK key pair on each side that we call the "verify" (public) and "sign" keys.



SCHEMA:

verify key

sign key

- KEYGEN() → (V, S)
- $\text{SIGN}_S(M)$ ← ONLY AUTHOR CAN SIGN
- $\text{VERIFY}_V(M, \text{SIG})$ ← ANYONE MAY VERIFY

Encryption Keypair: E_{PK}, E_{SK} (or, E_A + D_A)

Authentication Keypair: V_A, S_A (verify + sign keys)

Sending a message

Let E_B be Bob's public encryption key.

Alice sends $E_B(M) \mid H(S_A, E_B(M))$

Bob computes $\text{Verify}(V_A, H(S_A, E_B(M)))$

If checks out, then Bob computes $D_B(E_B(M)) = M$

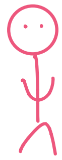
MESSAGE AUTHENTICATION CODES

MAC's are the symmetric-key alternative to Digital Signatures.

These are "KEYED CHECKSUMS" that only those w/ the shared key may compute.

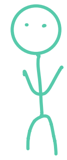
SCHEMA

- KEYGEN() → K
- SIGN(K, M) → T ("Tag")
- VERIFY: compute tag, check if match



K_E: encryption key

K_M: MAC key



$M =$ "Hi There"

$C = E_{K_E}(M)$

$T = \text{SIGN}_{K_M}(C)$

(C, T) →

$M = D_{K_E}(C)$

$T' = \text{SIGN}_{K_M}(C)$

If $T' = T$, then we're good!

EXAMPLE: RSA ENCRYPTION

SCHEMA

- KEYGEN() → Pick a random pair of large primes p, q

  Let $N = pq$

  Let e = any number relatively prime to $(p-1)(q-1)$

  Bob's Public Key: (N, e)

  Bob's Secret Key: d = Inverse of $e \bmod (p-1)(q-1)$

- $\text{ENCRYPT}_{PK}(M)$: $C = M^e \bmod N$

- $\text{DECRYPT}_{SK}(C)$: $M = C^d \bmod N$

EXTENSION: RSA SIGNATURES

For this class, we fix $e = 3$.

SCHEMA

- KEYGEN() → Same as Above

- $\text{SIGN}_{SK}(M) \rightarrow H(M)^d \bmod n$

- $\text{VERIFY}_{PK}(M, SIG) \rightarrow \begin{cases} \text{TRUE} & \text{if } H(M) = SIG^3 \bmod n \\ \text{FALSE} & \text{otherwise} \end{cases}$