cryptology

symmetric encryption

- $E(x)$: ENCRYPTION FUNCTION
- $D(x)$: DECRYPTION FUNCTION
- m : message
- $E(m) = c$: ENCODED MESSAGE : WHATS SENT

$$m = D(c) = D(E(m))$$

- TO ENCODE AND DECODE MESSAGES, YOU NEED A SHARED KEY

ONE TIME PAD

$m = CCI$

m in ascii = 01000011 01000011 01001001

key = k = 00110000 00100110 10000101

XOR

p	q	p⊕q
0	0	0
0	1	1
1	0	1
1	1	0

$E(m) = m_i \oplus k_i$

$E(m) =$ 01110011 11100101 11001100

$c = E(m)$

$D(c) = c_i \oplus k_i = m$

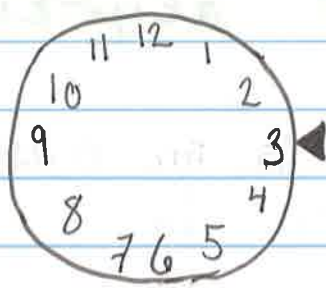
$D(c) =$ 01000011 / 01000011 / 01001001

$m = CCI$

Encryption Standards

- **DES:** Data Encryption Standards
- **AES:** American Encryption Standards
- **IDEA:** International Data Encryption Algorithm
- RC4 used in:
  - **SSL:** Secure Sockets Layer
  - **WEP:** Wired Equivalent Privacy

Modular Arithmetic



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
G H I J K L M N O P Q R S T U V W X Y Z A B C D E F

$$E(n) = (n + 6) \% 26$$

RELATIVE PRIMENESS

- **PRIME:** n is prime if it has no integral divisors other than 1 and itself
- **RELATIVELY PRIME:** n + m are relatively PRIME IF THEY SHARE NO OTHER INTEGRAL DIVISORS OTHER THAN 1 (DON'T NEED TO BE PRIME)

EXAMPLES

PRIME	7, 13, 17, 19, 43, 2017, 4969
RELATIVE PRIME	12+13, 2+3, 17 and 19

- $\gcd(m, n) = 1$

GCD ALGORITHM

1) COMPUTE r AS THE REMAINDER OF m divided by n. $r = m \% n$

2) IF $r = 0$: STOP AND OUTPUT n as THE GCD

3) ELSE
   a) REPLACE m WITH n
   b) REPLACE n WITH r
   c) STEP 1

Diffie-Hellman Key Exchange

- Public key technique to establish a shared secret without transmitting the secret
- Two numbers g and p where p is prime + publically known

1) Alice generates a random number a and Bob generates a random number b
2) Alice transmits $g^a \% p$ to Bob
3) Bob transmits $g^b \% p$ to Alice
4) Bob and Alice compute

$$(g^b)^a \% p = g^{ab} \% p = (g^a)^b \% p$$

Public Key Encryption

Assymmetric: Different Encryption + Decryption key
- No shared secret
- Encryption key is public
- Decryption key is private

* Anyone can encrypt a message for anyone else. Only the intended recipient can decrypt it *

RSA Key Generation

1) Pick 2 large random numbers p and q
2) Let $n = pq$
3) Compute $\phi(n) = (p-1)(q-1)$
4) Pick e relatively prime to $\phi(n)$
5) Find d such that $ed = 1 \% \phi(n)$
6) Publish e and n; d is kept private
7) $E(x) = x^e \% n$
8) $D(x) = x^d \% n$
9) $x = E(D(x)) = D(E(x)) = x^{ed} \% n$

Signatures

Question: In PKC, how do we know the sender is real?
Answer: Append a signature that can only come from the purposed sender

1) Alice (a) is sending to Bob (b)
2) Alice computes $c = E_b(m)$
3) Alice computes $s = H(m)$ where H is a cryptographic hash function
4) Alice sends $(c, D_a(s))$
5) Bob verifys that $H(D_b(c)) = E_a(D_a(s))$
6) Only Bob can read c and only Alice can send $D_a(s)$

CERTIFICATES

QUESTION: HOW DOES A SENDER KNOW IT HAS THE RIGHT PUBLIC KEY FOR A RECIPIENT?

ANSWER: A CERTIFICATE FROM A MUTUALLY TRUSTED PARTY

CERTIFYING AUTHORITY (CA): THE MUTUALLY TRUSTED PARTY

- CA ANSWERS QUERIES WITH A CERTIFICATE CONTAINING THE PUBLIC KEY CONTAINING THE PUBLIC KEY IN QUESTION AND A SIGNATURE FROM THE CA