Message Authentication

- Content modification
- Sequence modification
- Timing modification

Going to use hash functions + encryption.

(a) $E(K, m)$ - symmetric key
confidentiality, authentication

(b) $E(PU_b, m)$
confidentiality

(c) $E(PR_a, m)$
authentication, signature

(d) $E(PU_b, E(PR_a, m))$
confidentiality, authentication, signature

We don't do this because it's <u>SLOW</u>!

Transmit both the message (maybe encrypted) and MAC (Message Authentication Code).

Use hash function, is short. (faster than reg encryption)

$M \| C(K, m)$ gets sent.

User calculates $C(K, m)$ and sees if it matches.

$$E(K_2, [M \| C(K_1, m)])$$

Same as above with encryption added

$E(K_2, m) \| C(K_1, E(K_2, m))$

authentication tied to ciphertext.

$p = 17$, $3^{16} \equiv 1 \mod 17$

3 is prime root

3^2 is not

3^3 is

3^4 is not

$(3^2)^8 = 3^{16} \equiv 1 \mod 17$

$(3^4)^4 = 3^{16} \equiv 1 \mod 17$

$(3^3) \rightarrow 3^3, 3^6, 3^9, 3^{12}, 3^{15}$

$3^{18}, 3^{21}, 3^{24}, 3^{27}, \ldots$

none exp are perfect multiples of 16

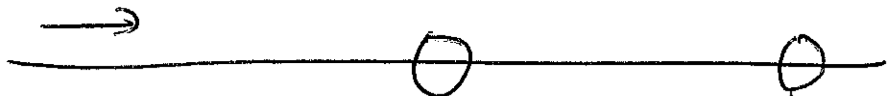
Quantum Cryptography (from Code Book)

0 1

— | \ /   (4 possible settings)

Reader + X

Send particles through fiber-optic cable



Alice

Eve
if she knows which reader is used she can use it and **NOT** corrupt the data.

If eve doesn't know reader and guesses, she'll be wrong $\frac{1}{2}$ the time and of those times will change the bit $\frac{1}{2}$ the time.

Bob
picks a reader

To see if there was tampering, Bob could call Alice, pick some bits at random + tell her what he read.

→ these bits are **unusable**.

Rather than Bob meeting w/ Alice and getting each reader orientation, Bob **GUESSES**.

Alice sends 2^{10} bits (rand reader orientation) to Bob.

Bob to guess the correct reader for 2^9 bits.

→ On phone, share reader guesses so Bob knows when he guessed correctly.

From those 2^9 bits, sample 2^7 of them. (Alice + Bob communicate what was send + what was received.) → Eve will guess correctly for 2^6 of sample bits, incorrectly for 2^6. On average, she'll change 2^5 of these bits.

Prob NONE change is $\left(\frac{3}{4}\right)^{128} \approx 1.02 \times 10^{-16}$

We have left $2^9 - 2^7 \approx 384$ bits transmitted.

Digital Signatures

Requirements

1) bit pattern must be dependent on the msg.
2) info in sig should be unique to the sender to prevent __forgery__.
3) easy to produce + verify
4) computationally infeasible to produce a forgery.
5) practical to retain a copy.

Today: El Gamal Digital Signature

Monday: DSS (Digital Signature Standard)

Global elements (Public) ① prime q

② generator/prim root α.

③ $Y_A = \alpha^{X_A}$

Private element : ① X_A $1 < A_A < q-1$.

SIGNATURE

① Choose random k $1 \leq k \leq q-1$, $\gcd(k, q-1) = 1$.
(diff each msg)

② Calculate $H(m) = m$, the hash function of the message.

③ $S_1 = \alpha^k \bmod q$ (part one of sig)

④ $k^{-1} \bmod (q-1)$

⑤ $S_2 = k^{-1}(m - X_A S_1) \bmod (q-1)$ 11/18/16 ②

I receive $M', (S_1, S_2)$

① I calculate $m' = H(M')$

② $V_1 = \alpha^{m'} \bmod q$

③ $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$

To verify I check to see if $V_1 == V_2$?

$$V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$$
$$= (\alpha^{X_A S_1})\alpha^{k S_2} \bmod q$$
$$= \alpha^{X_A S_1 + k S_2} \bmod q$$
$$= \alpha^{(X_A S_1 + k S_2) \bmod (q-1)} \bmod q$$
$$= \alpha^{X_A S_1 + k \cdot k^{-1}(m - X_A S_1) \bmod q-1} \bmod q$$
$$= \alpha^{X_A S_1 + m - X_A S_1 \bmod q-1} \bmod q$$
$$= \boxed{\alpha^{m} \bmod q}$$

$3^{37} \bmod 17$ is the same as $3^{37 \bmod 16} \bmod 17$ because

$$3^{37} = 3^{32} \times 3^{5} \bmod 17$$
$$\equiv 1 \times 3^{5} \bmod 17$$