Message Authentication

- Content modification
- Sequence modification
- Timing modification

Going to use hash functions + encryption.

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

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

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

(d) $E(PUb, E(PRa, m))$
   confidentiality, authentication, signature

We don't do this because it's slow!

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

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

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

$E(K_2, [M \parallel C(K, m)])$

Same as above with encryption added.
\[ E(k_2, m) \parallel C(k_1, E(k_2, m)) \]

Authentication tied to ciphertext.

\[ p = 17 \quad 3^{16} \equiv 1 \mod 17 \]

3 is prime root

\[ 3^2 \text{ is not} \]

\[ 3^3 \text{ is} \]

\[ 3^{41} \text{ is not} \]

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

\[ (3^{41})^4 = 3^{16} \equiv 1 \mod 17 \]

\[ (3^3) \rightarrow 3, 3^3, 3^6, 3^9, 3^{12}, 3^{15}, \ldots \]

... none are perfect multiples of \( 16 \)
Quantum Cryptography (from Code Book)

0 1
1 \ \ \ (4 possible settings)
Reader + X

Send particles through fiber-optic cable

Alice → Eve
Eve (if she knows which reader is use) picks a reader
she can use it and NOT corrupt the data.

if Eve doesn't know reader and guesses,
she'll be wrong 1/2 the time and
of those times will change the bit 1/2 the time:

To see if there was tampering, Bob
could call Alice, pick some bits at
random and 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 (and reader orientation) to Bob. Bob to guess the correct reader $2^9$ bits.

$\rightarrow$ On phone, share reader guesses so Bob knows when he guessed correctly.

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

$\text{Prob. Now change is } \left( \frac{3}{4} \right)^{128} \approx 2.102 \times 10^{-76}$

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 (1) prime $\mathbb{Z}$
(Public) (2) generator/prim root $\alpha$.
(3) $Y_A = \alpha^X_A$

Private elements: (1) $X_A$, $1 < A < \varphi - 1$.

Signature
(1) Choose random $k_c$, $1 \leq k \leq \varphi - 1$, $\gcd(k, \varphi - 1) = 1$.
(2) Calculate $Y = H(m) = m$, the hash function of the message.
(3) $S_1 = \alpha^k \mod \varphi$ (part one of sig)
(4) $k^{-1} \mod (\varphi - 1)$
I receive $M, (S_1, S_2)$.

1. Calculate $m = H(M)$

2. $V_1 = L^m \mod q$

3. $V_2 = (Y_A)^{S_1} (S_i)^{S_2} \mod q$

To verify: I check to see if $U_1 = U_2$?

$$V_2 = (Y_A)^{S_1} (S_i)^{S_2} \mod q$$

$$= (L^{x_A S_1}) L^{k S_2} \mod q$$

$$= L^{x_A S_1 + k S_2} \mod q$$

$$= L^{(x_A S_1 + k S_2) \mod (q-1)} \mod q$$

$$= L^{x_A S_1 + m - x_A S_1 \mod q + \mod q} \mod q$$

$$= L^m \mod q$$