Lecture 6: October 10, 2001

- Review: DES, Merkle's puzzles
- One-time signatures
- Public key cryptography

- Project proposals due next Monday
- Homework 0: due next Wednesday
  - Anonymous comments gts@dr.com
  - Signed PGP/GPG email gts@dr.com

DES: Modes of Operation

- Electronic code-book (ECB)
  - Yi=E(K,Xi)
- Chained block cipher (CBC)
  - Yi=E(K,Yi-1 XOR Xi)
- Output feedback (OFB)
  - Vi=E(K,Vi-1) Yi=Xi XOR Vi
- Cipher feedback (CFB)
  - Yi=Xi XOR E(K,Yi-1)

  OFB/CFB - encrypt only!
- MAC = X1…Xn, Yn (CBC)

- Local error, permutation attack
- Need IV, error causes avalanche effect
- Stream cipher, local error
- Plaintext dependence, avalanche effect

DES Variants

- 2-DES (double DES) =

  E(K2,E(K1,X)) or D(K2,E(K1,X)) --- weak!
- 3-DES (triple DES) = E(K1,D(K2,E(K1,X)))or E(K3,D(K2,E(K1,X))) --- same security?
- DESx =

  K3 XOR E(K2, K1 XOR X): 2^{184} security proved by Rogaway

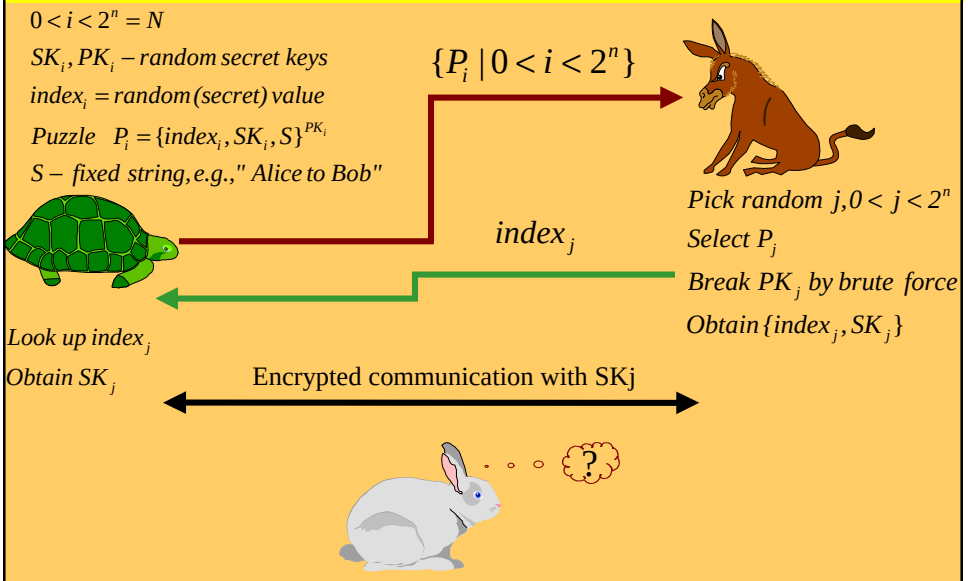
  K1 XOR E(K1,X)?

  E(K1, K1 XOR X)?

DES summary

- Permutation/substitution block cipher
- 64-bit data blocks
- 56-bit keys (8 parity bits)
- 16 rounds (shifts,xors)
- Key schedule
- S-box selection secret…
- DES "aging"
- 2-DES: rendezvous attack
- 3-DES: 112-bit security?
- DESX : 64-bit security?

Merkle's Puzzles (1974)



Merkle's Puzzles (contd.)

- Can we get better than N^2 to N advantage?
- Can we save Alice some memory?
- Can we reduce communication?

Merkle's Puzzles modified

$0 < i < 2^n = N$

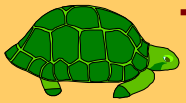
SK_i, PK_i – random secret keys

K_A – Alice' s own $2n$ – bit secret key

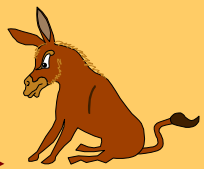
$index_i = \{SK_i, padding\}^{K_A}$

$Puzzle \quad P_i = \{index_i, SK_i, S\}^{PK_i}$

$S = h(index_i, SK_i)$



$\{P_i \mid 0 < i < 2^n\}$



Pick random $j, 0 < j < 2^n$

Break PK_j, P_j

Obtain $\{index_j, SK_j\}$

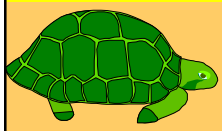
$index_j$

Decrypt $index_j$

Obtain SK_j

Encrypted communication with SKj

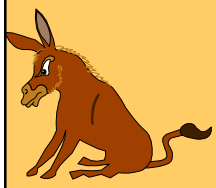
Merkle's Puzzles (contd.)



Alice:

•O(N) work to generate, compose, send puzzles

• Memory: constant



Bob

•O(N) work to break a single puzzle

•Memory: constant

Eve

•$O(N^2)$ work to break, on the average N/2 puzzles

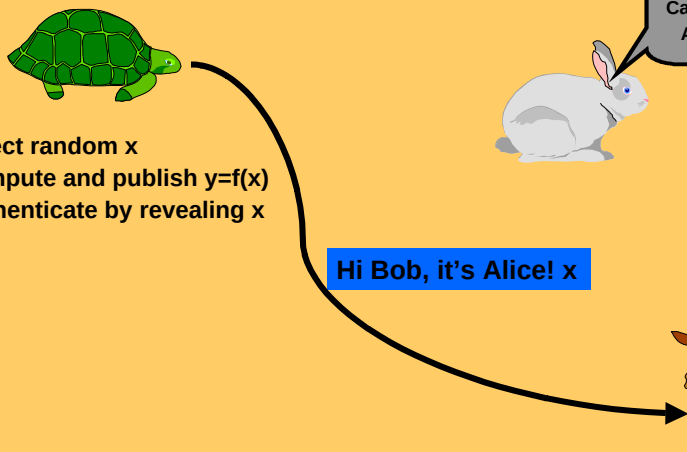
How do Alice and Bob authenticate each other?
Or
Why we need digital signatures…

One-time Authentication and Signatures (Lamport 1978)

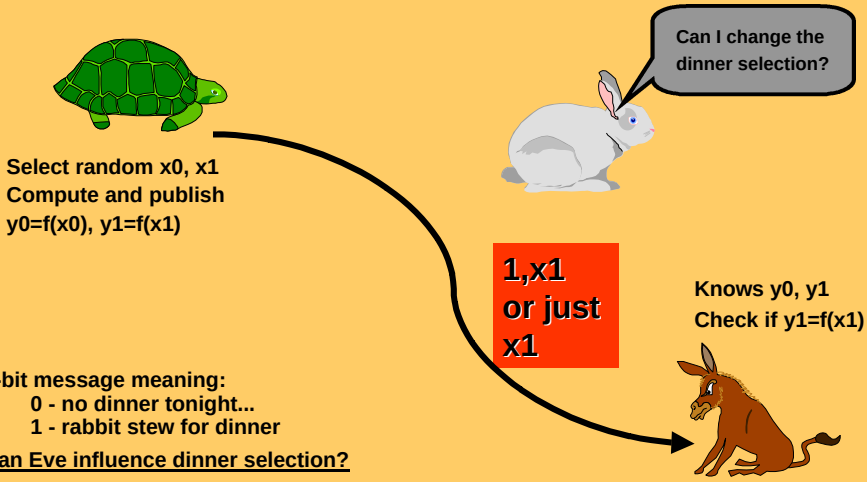
Requires a good One-Way Function (OWF):

- easy to compute: $y=f(x)$
- hard to invert: $x=f^{-1}(y)$
- collision-resistant: hard to find x1,x2 such that $f(x1)=f(x2)$
- e.g., modular exponentiation in prime-order groups

One-time Authentication



One-time Signatures



1-bit message meaning:
0 - no dinner tonight...
1 - rabbit stew for dinner

<u>Can Eve influence dinner selection?</u>

One-time Signatures (contd)

- How to sign arbitrarily long messages?
- How to do so efficiently?
- How to sign multiple messages?

- Two pairs: (x0,y0), (x1,y1) for each bit
- Efficient, hard-to-invert, collision-free hash functions, e.g., MD5 or SHA
  h(message)=fixed-length digest
- Various tricks, e.g., sign only 1-s or only 0-s
- OTS chains and trees

One way to produce a One-time Signature of n-bit message

$m - message$

$h - message\ digest,\ h = hash(m)$

$n = |h| or\ \ log(max(h))$

$SK = x$

$PK = y$

$y = f^{2^n}(x)$

$Signature: m,\ f^{2^n - h}(x)$

One-time Signature of n-bit message

Example:

$h = 01001101$

$n = 8$

$PK = y = f^{256}(x)$

$SK = x$

$SIG(m) = m, f^{179}(x)$

Is it possible to forge a valid message?

"Optimal" One-time Signature of an n-bit message

$m - input$

$h - message\ di\ gest,\ h = hash(m)$

$n = |h| or\ \ log\ (max(h))$

$w = log(n) - 1$

$z = min(\#1bits\ (h), \#0bits\ (h))$

$SK = X_0, X_1, \{x_0, ..., x_{n/2-1}\}, \{x'_{0,0}, x'_{0,1}, ..., x'_{w-1,0}, x'_{w-1,1}\}$

$PK = Y_0, Y_1, \{y_0, ..., y_{n/2-1}\}, \{y'_{0,0}, y'_{0,1}, ..., y'_{w-1,0}, y'_{w-1,1}\}$

$f(x) = y$ (for all x, y)

$Signature : [X_0 \mid X_1, \{x_{i_0}, ..., x_{i_{z-1}}\}, \{x'_{0,0}, x'_{0,1}, ..., x'_{w-1,0}, x'_{w-1,1}\}]$

One-time Signature of n-bit message

Example:

$h = 0100110100011000$

$n = 16, w = 3, z = 6 = 110$

$SIG(m) = m, X_1, (x_1, x_4, x_5, x_7, x_{11}, x_{12}), (x_{0,1}, x_{1,1}, x_{2,0})$

Is it possible to forge a valid message?