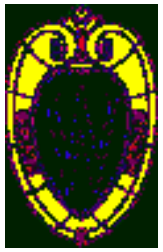


CS/ECE/Math 435
Introduction to Cryptography

Professor Chris DeMarco
Department of Electrical & Computer Engineering
University of Wisconsin-Madison

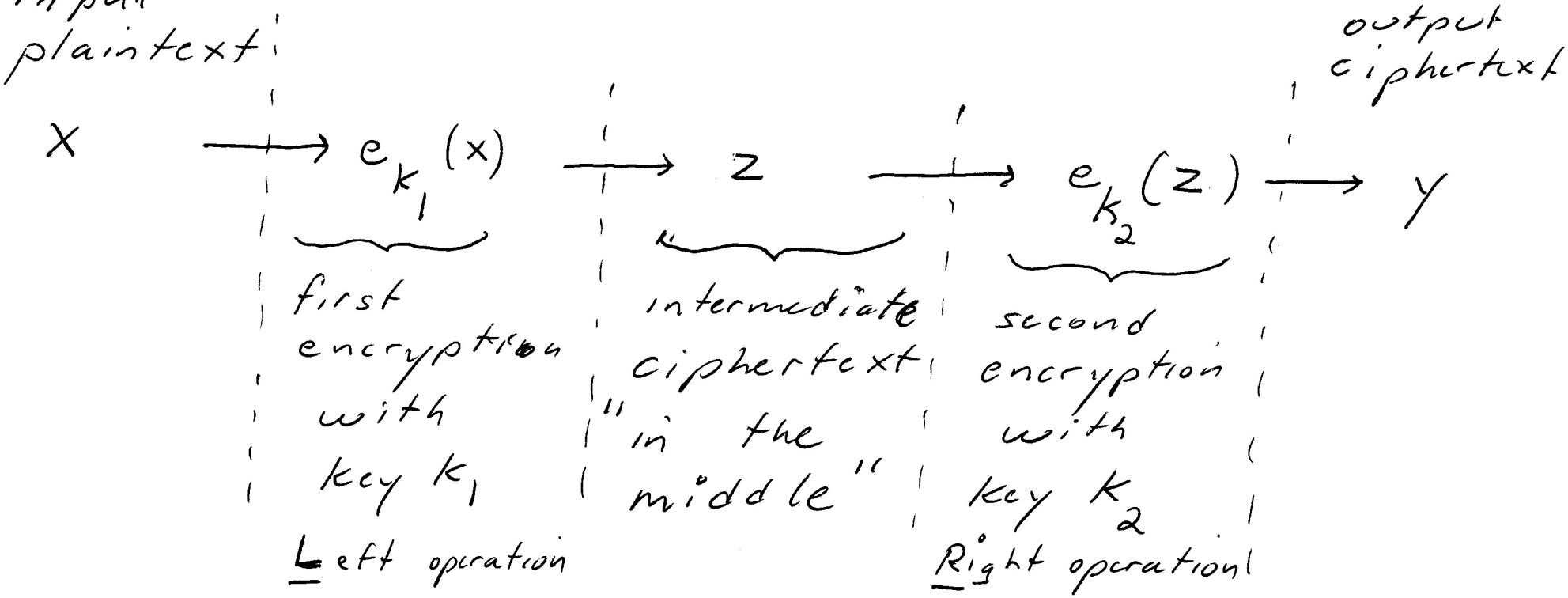
Spring Semester 2021

Return to known plaintext attack on multiple encryption - we'll use case of double encryption to illustrate.

KEY POINT: For any encryption scheme for which the $e_K(\cdot)$ functions are closed under composition, multiple encryption is equivalent to some single encryption in the same class.

So algorithm we are about to examine is only relevant for encryption schemes not closed under composition (e.g. DES).

Consider double encryption structure



Attacker seeks to identify k_1, k_2

given two known plaintext/ciphertext pairs

(x, y)

(x', y')

As always, recall our assumption that the attacker also knows the structure of encryption and decryption, $e_k(\cdot)$ and $d_k(\cdot)$. Note that this now becomes particularly realistic when the encryption scheme is a publically specified standard, such as DES (or AES).

The attack algorithm to follow is often termed a "man in the middle" attack.

Basic idea: Apply encryption function $e_{k_1}(x)$ from "Left," apply decryption operation $d_{k_2}(y)$ from the Right, look for a matching z "in the middle."

Q: Why do we assume attacker knows two pairs (x, y), (x', y'), instead of just one pair (x, y)?

Ans: We'll show that this lowers the probability of getting a false k_1, k_2

$|K|$ denotes size of key space; for DES, with 56 bit key, $|K| = 2^{56}$.

Algorithm will build (and the sort) an array, "T", of $2 \times |K|$ rows:

For all possible keys, k_i, $i = 1, 2, \ldots, |K|$

- row $(2i-1)$ of T is assigned:

$$[\, e_{k_i}(x),\ e_{k_i}(x'),\ k_i,\ L \,]$$

- row $(2i)$ of T is assigned:

$$[\, d_{k_i}(y),\ d_{k_i}(y'),\ k_i,\ R \,]$$

Note: these are all in the form of intermediate ciphertext, Z

just a binary label for Left or Right

- So each row of T has four components, with first two components of form (z, z').
- Search rows of T for pair of rows, say, row indices i, j such that: $z_i = z_j$ AND $z'_i = z'_j$

  AND rows i, j are of complementary types with respect to L and R (i.e. row i is L, row j is R, or vis-versa)
- In other words, a (z, z') match between a type L row and type R row.

- If a <u>single</u> pair of rows meeting these conditions, algorithm succeeds, and output is

$$\left.\begin{aligned} k_1 &= k_i^\circ \\ k_2 &= k_j^\circ \end{aligned}\right\} \text{ if row } i \text{ is type L, row } j \text{ is type R}$$

<u>OR</u>

$$\left.\begin{aligned} k_1 &= k_j^\circ \\ k_2 &= k_i^\circ \end{aligned}\right\} \text{ if row } i \text{ is type R, row } j \text{ is type L}$$

- Otherwise, algorithm fails (implying that attacker needs additional (x'', y'') to solve k_1, k_2).

Analogy to two-group birthday problem:

Two-group birthday problem:

Group L, of size n_1,

Group R, " " n_2.

Every member of each group has an associated integer value $k \in \mathbb{Z}_{365}$ (their birthday), randomly, uniformly distributed.

Classic result in probability (see Bach notes p. 28-1):

- Define a "Birthday pair" as a match in birthday between a person in L, a person in R.

[Expected number of birthday pairs between group L and group R]:

$$= \frac{n_1 \cdot n_2}{N}$$

Application to our "man in the middle" attack algorithm:

As we build rows of T, we construct 2^ℓ (z, z') samples of type "L" $\Rightarrow$ this plays role of Group L, of size $n_1 = 2^\ell$

We also construct 2^l (w, w') samples of type "R" $\Rightarrow$ this plays role of Group R, of size $n_2 = 2^l$.

The number of possible (w, w') values is $2^m \times 2^m = 2^{2m}$; this plays the role of N.

So the expected number of of (w, w') matched pairs between L and R $= \frac{n_1 \cdot n_2}{N} = \frac{(2^l)^2}{2^{2m}} = 4^{l-m}$

So for DES, $4^{l-m} = 4^{-8} = 1.53 \times 10^{-5}$

= expected number of L/R matches, "at random," without the structure of

$$e_{k_1}(x) = z = d_{k_2}(y)$$

$$e_{k_1}(x') = z' = d_{k_2}(y')$$

Next: the Advanced Encryption Standard.

Required Background: Finite Fields

We've danced around concepts of algebraic rings and fields thus far in the semester.

For arbitrary positive integer N, our sets $(\mathbb{Z}_N, +, \times)$ have structure of an Algebraic Ring, that we outlined in first lectures of the semester (Bach notes section 2).

But we've seen that for some special $\tilde{N}$, notably $\tilde{N} = 2$, we have the "nice" added structure that

for all $x \neq 0$, $x \in \mathbb{Z}_{\tilde{N}}$, there exists $y \in \mathbb{Z}_{\tilde{N}}$ such that $x \cdot y = 1$.

Stated in other ways:

- all nonzero x have a multiplicative inverse
- $\mathbb{Z}_{\tilde{N}}^{*} = \mathbb{Z}_{\tilde{N}}$ less $\{0\}$

Systems of form $(\mathbb{F}, +, \times)$ with this nice added property that all non-zero elements have multiplicative inverse are <u>fields</u>. Familiar examples of fields with infinite numbers of elements are real numbers ($\mathbb{R}$) complex numbers ($\mathbb{C}$), rational numbers ($\mathbb{Q}$).

Our interest lies with fields having finite numbers of elements: "Finite Fields" or equivalently "Galois Fields."

Commonly used notation (as you'll find, for example, in MATLAB documentation):

$GF(q)$, $q \geq 2$, denotes a Galois field of q elements.

So our use of $\mathbb{Z}_2$ has been an example in $GF(2)$. Indeed, by our definition of $\mathbb{Z}_N^*$, it is apparent that $\mathbb{Z}_N^*$ does $= \mathbb{Z}_N$ less $\{0\}$ when N is prime.

Hence, with appropriate $+$, $\times$ operations modulo N, $\mathbb{Z}_N$ forms a finite field when N is prime.

Constructing finite fields

Fact: For every $n \geq 1$, there exists a finite field F with 2^n elements.