# Meet in the Middle

March 13, 2019

# Last week's exercise

Solution on whiteboard.

# Last week's exercise (cont.)

- First work on paper, then start coding.
- Try to write code to debug your program. You are working with functions that try to look as random as possible.
- When you get stuck, send me an email or hop by my office. I am there to teach you things.

# SKINNY

- Skinny is a lightweight 'tweakable' block cipher
- See `https://eprint.iacr.org/2016/660.pdf` for the full specification

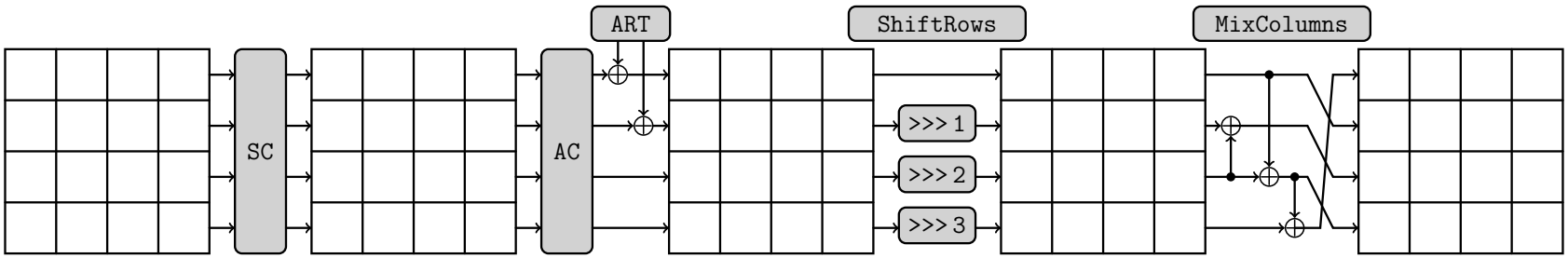
# SKINNY

- Skinny is a lightweight 'tweakable' block cipher
- See `https://eprint.iacr.org/2016/660.pdf` for the full specification
- Skinny-64-128,
  - 64-bit block
  - 128-bit key
  - 36 rounds
  - 23 rounds broken

# SKINNY

- Skinny is a lightweight 'tweakable' block cipher
- See `https://eprint.iacr.org/2016/660.pdf` for the full specification
- Skinny-64-128,
  - 64-bit block
  - 128-bit key
  - 36 rounds
  - 23 rounds broken
- Skinny-128-128, 64-bit block, 128-bit key
  - 128-bit block
  - 128-bit key
  - 40 rounds
  - 19 rounds broken

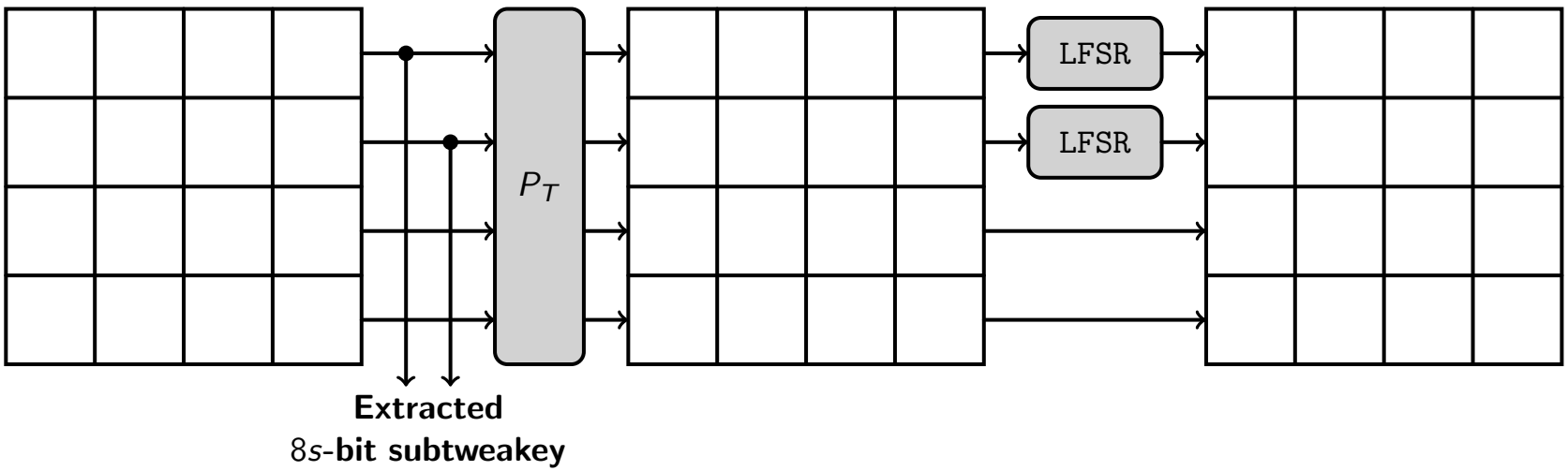
# SKINNY Round Function




$$S_4 = [\texttt{C}\ \texttt{6}\ \texttt{9}\ \texttt{0}\ \texttt{1}\ \texttt{A}\ \texttt{2}\ \texttt{B}\ \texttt{3}\ \texttt{8}\ \texttt{5}\ \texttt{D}\ \texttt{4}\ \texttt{E}\ \texttt{7}\ \texttt{F}]$$

$$M = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

# SKINNY Tweakey Schedule




$$P_T = [9\ \ 15\ \ 8\ \ 13\ \ 10\ \ 14\ \ 12\ \ 11\ \ 0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7]$$

$$\mathrm{LFSR}_{TK2} = (x_3||x_2||x_1||x_0) \rightarrow (x_2||x_1||x_0||x_3 \oplus x_2)$$

# Breaking Skinny

- How far can we go?
- How far can we push our computers?

# Breaking Skinny

- How far can we go?
- How far can we push our computers?
  - CPU - $2^{50}$ time
  - Memory - $2^{40}$ bits

# Let's break SKINNY

Whiteboard

# Summary

- Find the influence of each unknown round key nibble
- Find the output nibble that can be computed with the most unknown key nibbles after $r$ rounds.
- Do the same for the backward direction, but now match on the central nibble.
- Now we need to implement the solution.
  - Mask expansion to compute the key/roundkeys.
  - Filter to match on the inner state.
  - 'Partial' cipher (extra care should be given for decryption).
  - Partial bruteforce.
- **TEST** your code before trying to tackle full exercise
  - Especially test the datastructures you use/implemented.
  - Test if your cipher generates the right ciphertext.
  - Test if for a triple $(m, c, k)$ your program succeeds.
  - Test the attack on fewer rounds.
- Now download the challenge and run your program.
- Celebrate! Or tweak your program.

# For next week

- Finish last weeks exercise.
- For the ones who did not do it yet send me your CPU model and amount of RAM