

Permutation Ciphers

- Instead of substituting different characters, scramble up the existing characters
- Use algorithm based on the key to control how they're scrambled
- Decryption uses key to unscramble

Characteristics of Permutation Ciphers

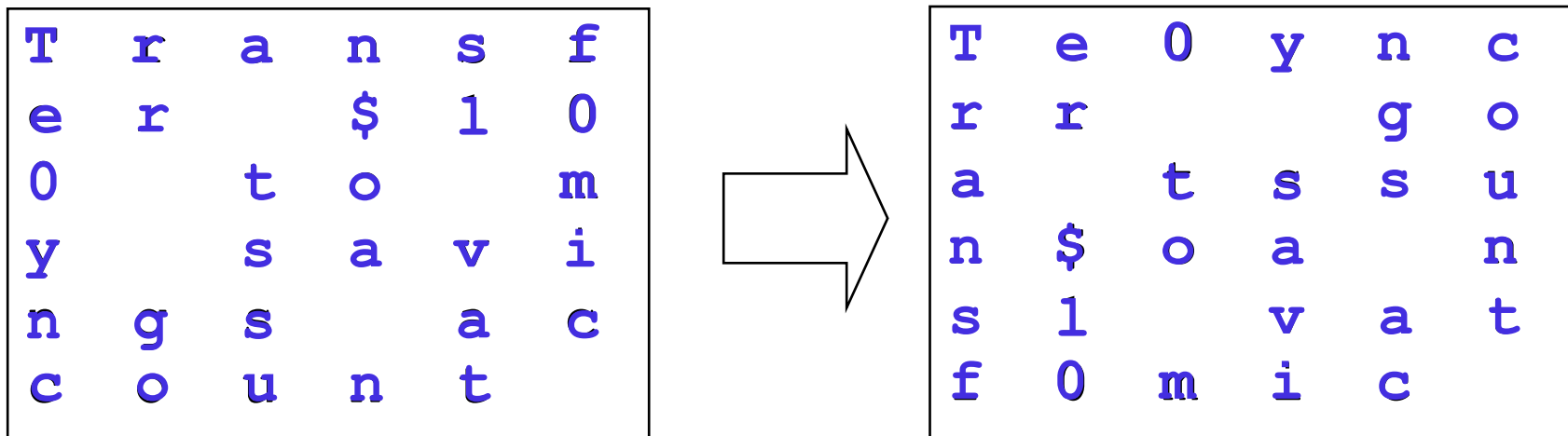
- Doesn't change the characters in the message
 - Just where they occur
- Thus, character frequency analysis doesn't help cryptanalyst

Columnar Transpositions

- Write the message characters in a series of columns
- Copy from top to bottom of first column, then second, etc.

Example of Columnar Substitution

How did this transformation happen?



Looks a lot more cryptic written this way:

Te0yncrr goa tssun\$oa ns1 vatf0mic

Attacking Columnar Transformations

- The trick is figuring out how many columns were used
- Use information about digrams, trigrams, and other patterns
- Digrams are pairs of letters that frequently occur together (“re”, “th”, “en”, e.g.)
- For each possibility, check digram frequency

For Example,

^{4 5 6} Te0yncrr ^{1 2 3 4 5 6 1 2 3 4 5 6 1 2 3} goa tssun\$oa nsl vatf0mic

\$ 1 0 0

In our case, the presence of dollar signs and numerals in the text is suspicious

Maybe they belong together?

Umm, maybe there's 6 columns?

Double Transpositions

- Do it twice
- Using different numbers of columns
- How do you break it?
 - Find pairs of letters that probably appeared together in the plaintext
 - Figure out what transformations would put them in their positions in the ciphertext
- Can transform more than twice, if you want

Generalized Transpositions

- Any algorithm can be used to scramble the text
- Usually somehow controlled by a key
- Generality of possible transpositions makes cryptanalysis harder

Which Is Better, Transposition or Substitution?

- Well, neither, really
- Strong modern ciphers tend to use both
- Transposition scrambles text patterns
- Substitution hides underlying text characters/bits
- Combining them can achieve both effects
 - If you do it right . . .

Quantum Cryptography

- Using quantum mechanics to perform crypto
 - Mostly for key exchange
- Rely on quantum indeterminacy or quantum entanglement
- Existing implementations rely on assumptions
 - Quantum hacks have attacked those assumptions
- Not ready for real-world use, yet
- Quantum computing (to attack crypto) even further off