Luke Pepin

[Aidan Mcdonald](mailto:Aidan.McDonald@uconn.edu)

CSE 3400 Problem Set 1

2/12/24

1. Shift Cipher:
   1. n = 2

Consider the following Adversary (A):

Oracle chooses between the following 2 messages equally:

Output: m1 = aa

Output: m2 = ab

Receives: c = cc

If c(0) == c(1), then output m1 else output m2

Then Pr(A"wins") = 1

If c(0) == c(1), given the processes of the Shift Cipher the Adversary (A) knows the same permutation is applied to each character (n) in the message (m). As a result the only possibility is that the encrypted message has the same character twice if it is also original and has the same character twice, making m1 the only appropriate answer.

* 1. y ≥ 2 and n

For the Shift Cipher to be more secure both the size of the alphabet (y) and the size of the message (n) are extremely large to rule out brute force attacks. However, given the nature of the Shift Cipher the key is always smaller than the message space as well as data manipulation such as distribution tables and pattern searching make it so no combination values of y ≥ 2 and n make this scheme secure.

1. CES

CES cannot be perfectly secure, given a encryption scheme that always uses 128-bit keys it fails on one of the requirements that makes One Time Pad to be perfectly secret, that the key space must be equal to the message space. The key has to equal the length of the message. If the message is 129 characters Key is 128 random characters the last element of the message is determined by the cipher at some point where the key was already applied. As a result, that last bit proves a weakness in its security that Adversaries can exploit by comparing the last bit and the point where the key was already applied to gain additional information. For the alternative, if the extension of the key uses other elements of the key they cancel out making neither point of the key secure, breaking the security down and making it no longer secure.

1. CES512

AES works with a flexible key size, while a flat 512-bit key sounds impressive for an algorithm the larger the message size over that 512-bit key the less secure it becomes (see explanation above). On top of the undersized key size for large messages CES512 suffers from an oversized key size for smaller messages, increasing the overhead that is required for the message to send making it inefficient for a small message size.

1. PRG: {0,1}ⁿ → {0,1}ⁿ⁺¹, defined by G(x₁…xₙ)=xₙxₙ₋₁…x₁xₙ

Not = PRG: Consider a distinguisher D:

On input a word w = w₁,...,wₙ

If wₙ = w₁⊕wₙ, output 1

Else output 0

*(output 1 indicating that the input word is likely generated by the PRG)*

Pr(D[G(Uₙ)]=1)=1

The PRG will always result in wₙ = w₁⊕wₙ being true as a result of its deterministic output.

Pr(D(Uₙ)=1)=0.5

The result of wₙ = w₁⊕wₙ in a truly random word is statistically random given each bit are not related.

| 1 - 0.5 | ≠ negl.

This shows candidate PRG is not an actual PRG.

1. PRG: {0,1}ⁿ → {0,1}ⁿ⁺¹, defined by multiple steps

Not = PRG: Consider a distinguisher D:

On input a word w = w₁,...,wₙ

ℓ = 

If w[ℓ+1] = w1⊕w2⊕…⊕wn, output 1

Else output 0

*(output 1 indicating that the input word is likely generated by the PRG)*

Pr(D[G(Uₙ)]=1)=1

The PRG will always result in w[ℓ+1] = w1⊕w2⊕…⊕wn being true as a result of its deterministic output.

Pr(D(Uₙ)=1)=0.5

The result of w[ℓ+1] = w1⊕w2⊕…⊕wn  in a truly random word is statistically random given each bit are not related.

| 1 - 0.5. | ≠ negl.

This shows candidate PRG is not an actual PRG.