

Tala Abujbara U90689026
Evangelos Petropoulos U75564437
Group 5

Extra-Credit Question [10 points]

0.2 Analysis and Comparison of the Toy Stream Cipher Concept

A. Potential Vulnerabilities: In the above toy stream cipher application, assume that Alice uses your software as is, and after encrypting n blocks of data, her computer completely crashes and then re-start just code to encrypt the rest of the data. Does this create a specific vulnerability in the encryption process? Describe an attack that can exploit the current state of the implementation in this event (the answer should have ideally 5-6 sentences, backed up with some basic equations to support the arguments). [5 pt]

This will create a vulnerability, specifically featuring the PRNG function. This is because PRNG is a deterministic function, but is still used because of its efficiency over TRNG (<https://www.geeksforgeeks.org/dsa/pseudo-random-number-generator-prng/>). Upon restart of the system, the function will encrypt with the same keystream starting from the beginning from before the system shut down, since the shared seed is still the same. Since the keystream is now reused, the system loses security, making it vulnerable to an Attack on Two Time Pad(<https://www.crypto-it.net/eng/attacks/two-time-pad.html>). The attacker gathers two ciphertexts utilizing the same keystreams and is able to extract the original messages by adding them together and XORing.

c1,c2 = ciphertexts m1,m2= messages k=keystream

c1 <- m1 XOR PRNG(k)
c2 <- m2 XOR PRNG(k)
to
c1 XOR c2 = m1 XOR PRNG(k) XOR m2 XOR PRNG(k) = m1 XOR m2
to
m1 XOR m2 -> m1, m2

Messages are extracted(<https://www.crypto-it.net/eng/attacks/two-time-pad.html>).

B. Potential Remedies: Can you suggest at least two ways to mitigate the negative impacts of these potential risks? Your solution should address the root cause of this vulnerability (remember in-class discussions). You can consider an approach only local to the machine, and another that gets benefits from the internet connection.

[5 pt]

Local: As discussed in class, a counter function tacked on to the PRNG function input that tracks how many keys have been generated so far and saves every iteration in case of system shut down is possible for local systems.

Internet Connection: Perhaps a function that combines the server time with the shared seed, so that way the PRNG function creates a unique keystream each session regardless of its deterministic function.

Works Cited

<https://www.geeksforgeeks.org/dsa/pseudo-random-number-generator-prng/>

<https://www.crypto-it.net/eng/attacks/two-time-pad.html>