stream ciphers 3

+ OTP
+ ATTACKS

CIFRA DE USO ÚNICO

1º ATAQUE: "TWO TIME PAD"

↳ não utilizar a mesma chave 2x

$c_1 \leftarrow m_1 \oplus PRG(K)$

$c_2 \leftarrow m_2 \oplus PRG(K)$

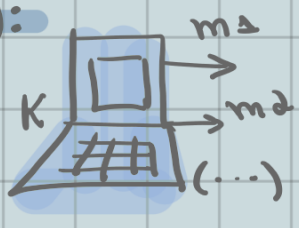
um atacante na escuta: $c_1 \oplus c_2 \rightarrow m_1 \oplus m_2$

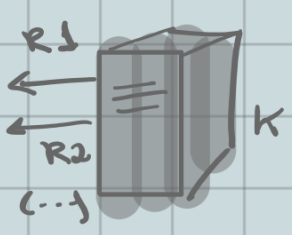
$\rightarrow m_1, m_2$

Exemplos onde esse ataque foi utilizado:

1. PROJETO VENONA: (1941-1946)
   * Programa de contra-inteligência dos EUA durante 2ºG.M.
   * Conseguiram descriptografar mensagens transmitidas pela união soviética

2. ms-pptp (windows nt):
   ponto a ponto: cliente e servidor se comunicam diretamente





$[m_1 || m_2 || m_n] \oplus PRG(K)$

CONCATENAÇÃO

Até aqui sem problemas... mas a resposta do servidor tem também utiliza a mesma chave.

$[R_1 || R_2 || K_n] \oplus PRG(K)$

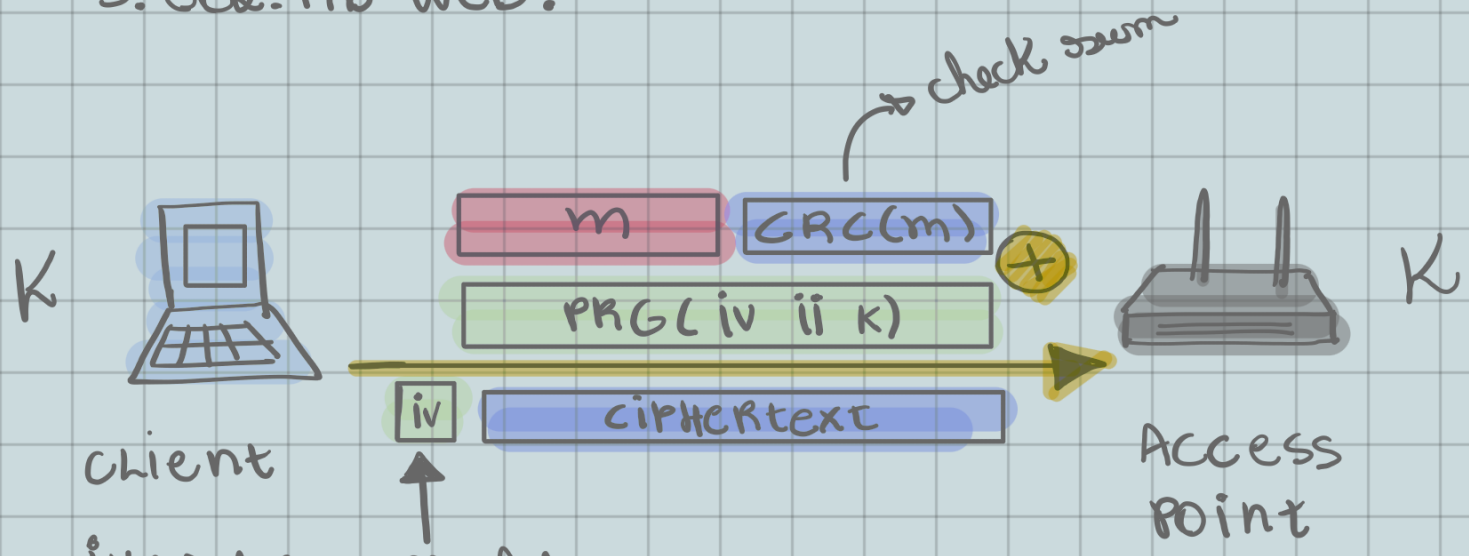
Isso é um problema!

É preciso utilizar novas chaves para c → s e s → c

$K = (K_{s \rightarrow c}, K_{c \rightarrow s})$

PAR de CHAVES

3. 802.11b WEB:



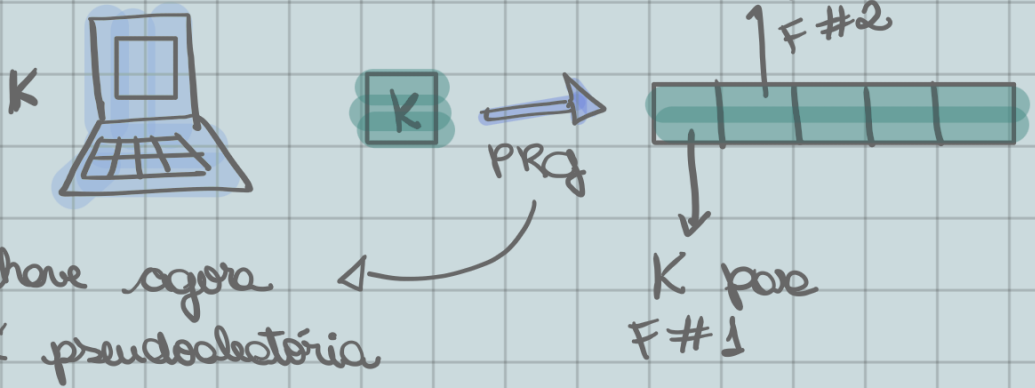
iv: string 24 bits

I- Repete após $2^{24} \cong 16M$ frames → Problema!

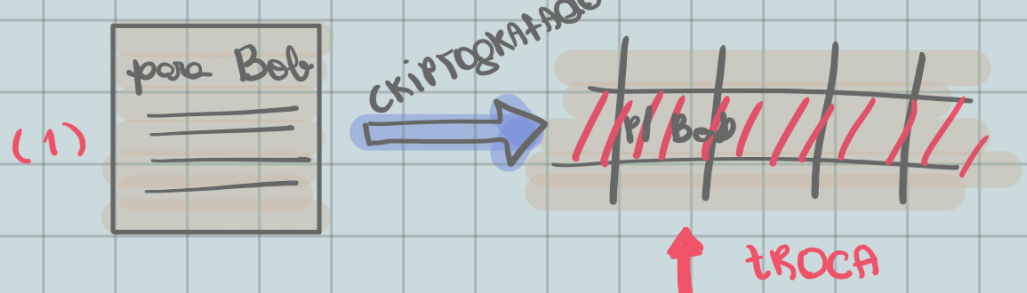
II- Em algumas implementações: IV reseta para 0 após um ciclo. → Problema!

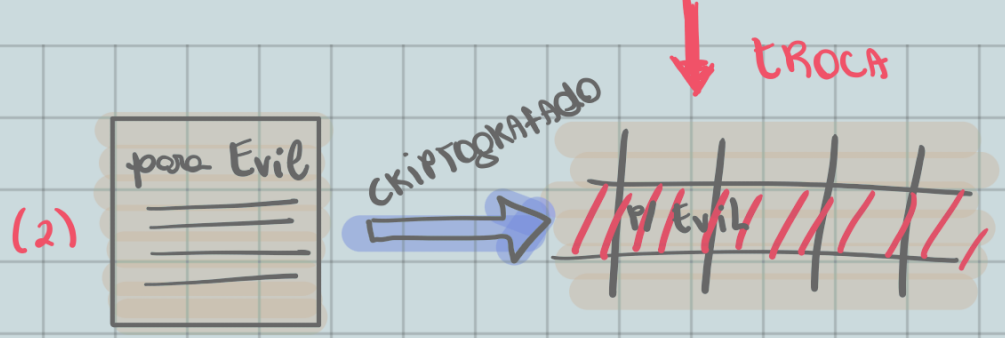
III- Chave frame #1 : (1||K)
" " #2 : (2||K)
(...)
#n : (n||K)
} Problema! Tem lógica, muito relacionados.

OPÇÃO MELHOR:



4. CRIPTOGRAFIA DE DISCO





2 arquivos criptografados c/ mesma chave, se uma terceira parte pegar esse chave terá +1 arquivo com essa chave e não saberemos se este novo arquivo é de um atacante.

2° ATAQUE: "no integrity" → OTP maleável