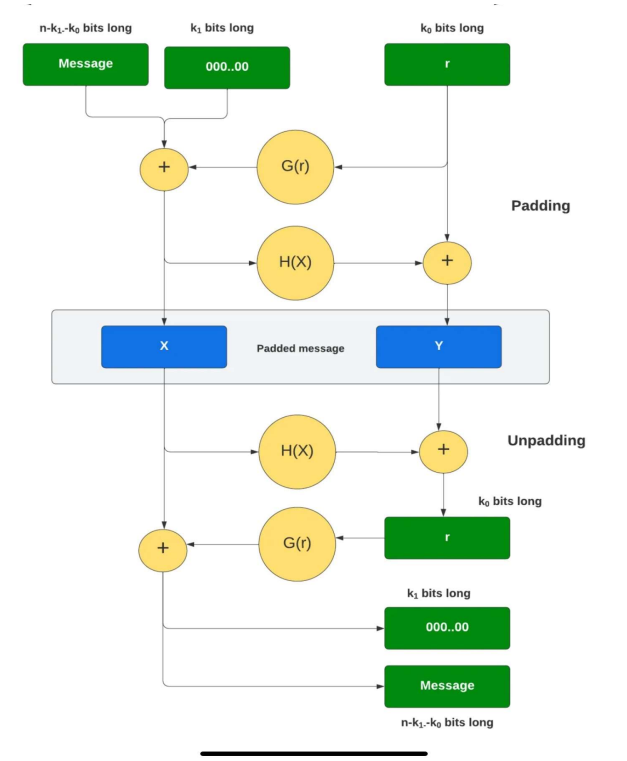
MAC 0336/5723 - Criptografia e Segurança de Dados - Exercício-programa 1 (USP) - 2024

- Prazo de entrega: ver no https://edisciplinas.usp.br/course/view.php?id=117383
- Resolver *individualmente*. Duas soluções **idênticas** receberão **nota zero**
- Utilize **necessariamente** a linguagem Python
- Manuscritos **não** são corrigidos, recebem nota zero.
- Entregue no sistema e-disciplinas um ÚNICO arquivo chamado EP1Python, comprimido (zip, tar, gz), contendo os arquivos seguintes
  - O seu programa em Python, cada uma das saídas, com a sua solução do EP
  - Um arquivo chamado LEIA.ME (em formato TXT) com:
    - seu nome completo, e número USP,
    - os nomes dos arquivos inclusos com uma breve descrição de cada arquivo,
    - qual computador, e qual versão do Python foram usados (modelo, versão, etc..),
- Coloque comentários no seu programa explicando o que cada etapa do programa significa! Isso será levado em conta na sua nota.
- Faça uma saída clara! Isso será levado em conta na sua nota.
- Não deixe para a última hora. Planeje investir 70 porcento do tempo total de dedicação em escrever o seu programa todo ANTES de digitar o programa. Isso economiza muito tempo e energia.
- A nota será diminuída de um ponto a cada dia “corrido” de atraso na entrega.

Este exercício é sobre implementação do algoritmo RSA com OAEP.(Output Added Extension Protocol) Veja a figura.



OAEP (RSA é aplicado aplicado sobre $X||Y$)

Notação:

1. || denota concatenação
2. $+$ no desenho é XOR

Definições:

1. $\boxed{n = 256}$ bits
2. R é pseudo-aleatório tal que $compr(R) = \boxed{k_0 = n/2 = 256/2 = 128}$, que o seu programa pede a um gerador de números com distribuição uniforme
3. Entrada/Message é o seu NUSP com 8 dígitos, $8 \times 4 = \boxed{32}$ bits.
4. Entrada com $compr(Message) = n - k_1 - k_0 = 32$. Logo, $\boxed{k_1 = n - k_0 - 32 = 256 - 128 - 32 = 96}$ bits
5. $G()$ e $H()$ com $compr\ k_0$ são as funções $SHA128()$

O objetivo é implementar RSA com:

1. Primos q, r de 128 bits cada.
2. n de $2 * 128 = 256$ bits

Executar os itens seguintes:

1. Calcular e listar q igual ao primeiro primo $\geq NUSP||NUSP||NUSP||NUSP$ (128 bits) Sugestão: calcular primos com Algoritmo Miller-Rabin (pg. 139)
2. Calcular e listar r igual ao primeiro primo $\geq q + 2$
3. Implementar RSA OAEP: Message, a entrada de 96 bits, é o seu NUSP com 8 dígitos. $X||Y$ é a saída do OAEP, que é a entrada do RSA e:
    - calcular e **listar** as chaves RSA com q, r calculados
    - executar OAEP com seu NUSP como entrada, **listar** os dígitos de $X||Y$
    - calcular e **listar** a saída do RSA com a chave pública (uma das chaves calculadas)
    - calcular e **listar** a inversa do RSA e verificar se obteve o seu NUSP
4. executar como entrada o primeiro bit à esquerda do seu NUSP complementado, e **listar** a saída
    - Calcular o número de bits alterados na saída criptografada e **listar** o resultado (i.e., verificar o Efeito Avalanche na saída)
5. Efetuar o item anterior, (4), com os dois primeiros bits complementados e **listar** o resultado

Exercício teórico opcional: demonstrar que RSA OAEP é seguro contra Chosen Ciphertext Attack — CCA, visto na Lista de Exercícios 2.