Bacharelado em Ciência da Computação

CCMP3079 Segurança de Redes de Computadores

Atividade Cap. 07

Prof. Sérgio Mendonça

Nome Completo: Luan Valentino Sampaio Marques

Data: 11/12/2023

Universidade Federal do Agreste de Pernambuco
Av. Bom Pastor s/n - Boa Vista
55292-270 Garanhuns/PE
+55 (87) 3764-5500
http://www.ufape.edu.br

Questões retiradas do livro-texto da disciplina

Conforme conversamos em sala de aula, as atividades devem ser realizadas para apresentação e discussão em sala, sempre nas aulas das quintas-feiras, atribuindo ao estudante uma nota de 0 ou 1 por cada atividade realizada e apresentada.

Questão 1

Qual é a diferença entre aleatoriedades estatísticas e imprevisibilidade?

Solução:

Aleatoriedade Estatística: Refere-se a uma sequência de números que segue critérios estatísticos bem definidos, como a distribuição uniforme e independência. A distribuição uniforme significa que a frequência de "0"s e "1"s na sequência é aproximadamente a mesma, e a independência significa que nenhum valor da sequência pode ser previsto a partir dos outros. A aleatoriedade estatística é importante para garantir que a sequência de números pareça aleatória e não exiba padrões óbvios.

Imprevisibilidade: Refere-se à incapacidade de prever elementos futuros de uma sequência com base nos elementos anteriores. Em criptografia, a imprevisibilidade é essencial para garantir que, mesmo que um oponente conheça parte da sequência, ele não possa prever os números subsequentes. Isso é especialmente importante em aplicações como autenticação e geração de chaves, onde a segurança depende de sequências que não podem ser antecipadas.

Enquanto a aleatoriedade estatística se preocupa com padrões detectáveis, a imprevisibilidade se foca na segurança contra previsões de adversários.

Questão 2

Liste considerações de projeto importantes para uma cifra de fluxo.

Solução:

Período Grande da Sequência de Encriptação: O gerador de número pseudoaleatório deve produzir um fluxo determinístico de bits com um longo período antes de repetir. Um período grande dificulta a criptoanálise, pois torna mais complicado identificar padrões repetitivos no fluxo de chaves.

Aparência Aleatória do Fluxo de Chaves: O fluxo de chaves deve se aproximar ao máximo das propriedades de um fluxo de número aleatório verdadeiro, com uma distribuição equilibrada de "1s" e "0s". Se o fluxo for tratado como um conjunto de bytes, todos os 256 valores possíveis devem ter frequências semelhantes. Quanto mais aleatório parecer o fluxo de chaves, mais seguro será o texto cifrado.

Tamanho da Chave Adequado: A chave usada no gerador pseudoaleatório deve ser suficientemente longa para resistir a ataques de força bruta. O material sugere um tamanho de chave de pelo menos 128 bits para garantir a segurança adequada, em linha com as considerações para cifras de bloco.

Esses elementos são essenciais para garantir a segurança e a eficácia de uma cifra de fluxo em criptografia.

Questão 3

Por que não é desejável reutilizar uma chave de cifra de fluxo?

Solução:

Não é desejável reutilizar uma chave de cifra de fluxo porque, se dois textos claros diferentes forem encriptados com a mesma chave, a criptoanálise se torna muito mais fácil. Quando dois textos cifrados gerados pela mesma chave são submetidos a uma operação XOR, o resultado é o XOR dos textos claros originais.

Se os textos claros tiverem propriedades conhecidas, como serem strings de texto, números de cartão de crédito, ou outros padrões previsíveis, um atacante pode usar essa informação para revelar os textos originais ou descobrir a chave. Isso compromete completamente a segurança da cifra de fluxo. Portanto, a reutilização de chaves em cifras de fluxo deve ser evitada para prevenir vulnerabilidades graves.

Questão 4

Que operações primitivas são usadas no RC4?

Solução:

Permutação Aleatória: O RC4 utiliza uma permutação aleatória dos números de 0 a 255, armazenados em um vetor de estado S de 256 bytes. Essa permutação é continuamente modificada durante a geração dos bytes de saída.

Troca de Valores (Swap): Durante o processo de inicialização e geração de fluxo de chaves, o RC4 realiza trocas (swaps) entre elementos do vetor S. Isso é feito para misturar a permutação inicial e para atualizar S a cada byte de saída gerado.

Seleção de Byte (Indexação): O algoritmo seleciona um byte específico do vetor S usando um índice calculado com base em valores que mudam a cada passo do processo. Esse byte selecionado é usado como parte do fluxo de chaves.

Operação XOR: O byte gerado pelo vetor S é combinado com um byte do texto claro ou cifrado utilizando a operação bit a bit XOR, tanto na encriptação quanto na decriptação.

Essas operações primitivas simples tornam o RC4 rápido e fácil de implementar em software.

Questão 5

Se apanharmos um algoritmo de congruência linear com um componente aditivo de 0:

$$X_{n+1} = (aX_n) \mod m$$

então, podemos mostrar que, se m é primo, e se determinado valor de a produz o período máximo de m 1, então ak também produzirá o período máximo, desde que k seja menor que m e que m 1 não seja divisível por k. Demonstre isso usando X0 = 1 e m = 31, e produzindo as sequências para ak = 3, 32, 33 e 34.

Solução:

Questão 6

(a) Qual é o período máximo que pode ser obtido do seguinte gerador?

$$X_{n+1} = (aX_n) \mod 24$$

Solução:
O período máximo é 8, pois 24 tem os fatores primos 2 e 3, e $\phi(24) = 8$, onde ϕ é a função totiente de Euler.

(b) Qual deverá ser o valor de 'a'?

Solução:
O valor de 'a' deve ser 5, 7, 11, 13, 17, 19 ou 23, todos primos relativos a 24 para garantir o período máximo de 8.

(c) Que restrições são exigidas na semente?

Solução:
A semente X_0 deve ser um número ímpar e não deve ser divisível por 3 para garantir que a sequência atinja o período máximo de 8.

Questão 7

Que valor de chave RC4 deixará S inalterado durante a inicialização? Ou seja, após a permutação inicial de S, as entradas de S serão quais aos valores de 0 a 255 na ordem crescente.

Solução:
A chave composta por todos os bytes iguais a 0 (por exemplo, uma chave de 256 bytes com valor 0) deixará S inalterado, mantendo $S[i] = i$ para i de 0 a 255 na ordem crescente.

Questão 8

O algoritmo Blum Blum Shub é baseado na teoria dos resíduos quadráticos e utiliza três números inteiros para realizar os cálculos: p, q e s.

(a) Escolha dois números primos grandes p e q, onde p e q sejam congruentes a 3 mod 4 e não tenham fatores primos comuns. Por exemplo, você pode escolher $p = 499$ e $q = 503$.

Solução:
Os números $p = 499$ e $q = 503$ são primos, ambos congruentes a $3 \bmod 4$, e não possuem fatores primos comuns, pois $\text{mdc}(499, 503) = 1$.

(b) Calcule $n = p \times q$. Neste caso, n seria igual a $499 \times 503 = 250997$.

Solução:
O valor de n é calculado como $499 \times 503 = 250997$.

(c) Escolha um número inteiro s entre 1 e $n-1$ que seja co-primo com n. Por exemplo, você pode escolher $s = 17$.

Solução:
O número $s = 17$ é escolhido, sendo co-primo com $n = 250997$.

(d) Calcule o valor inicial $x_0 = (s^2) \mod n$. Neste caso, x_0 seria igual a $17^2 \mod 250997 = 289$.

Solução:
O valor inicial x_0 é calculado como $17^2 \mod 250997 = 289$.

(e) Agora, vamos gerar uma sequência de números aleatórios usando o algoritmo Blum Blum Shub. Para gerar cada número da sequência, use a seguinte fórmula: $x_i = (x_{i-1}^2) \mod n$.

(f) Execute a fórmula várias vezes para gerar uma sequência de números aleatórios. Por exemplo, você pode executar a fórmula 10 vezes para obter 10 números aleatórios.
Aqui está a sequência de números aleatórios gerados usando o algoritmo Blum Blum Shub com os valores do exemplo:
$289, 253306, 14107, 23546, 67740, 144593, 79829, 46219, 132936, 9863$.

Qual foi a sua sequência?

Solução:
A sequência gerada seria baseada na aplicação repetida da fórmula $x_i = (x_{i-1}^2) \mod n$, iniciando com $x_0 = 289$ e seguindo o mesmo processo descrito.