CIn–UFPE
Introdução à Criptografia Moderna (Graduação)
Fundamentos da Criptografia Moderna (Pós-Graduação)
2009.1
Lista de Exercícios 8
Entrega: 3ª feira, 26/05/2009

Exercício 1 (Katz & Lindell (2007), 7.18 (2,0)) Determine se o seguinte problema é difícil: seja p um primo, e fixe um $x \in \mathbb{Z}_{p-1}^*$. Dados p, x, e $y := [g^x \bmod p]$ (onde g é um valor aleatório entre 1 e $p-1$), encontre g; ou seja, compute $y^{1/x} \bmod p$. Se você disser que o problema é difícil, mostre uma redução para uma das suposições introduzidas no capítulo 7 do livro texto. Se você disser que o problema é fácil, apresente um algoritmo, justifique sua corretude, e analise sua complexidade.

Exercício 2 (Katz & Lindell (2007), 7.20 (2,0)) Seja $\mathcal{G}_1$ um algoritmo que, sobre a entrada 1^n, escolhe um primo p de n-bits aleatório, e dá como saída p e a ordem do grupo $q = p - 1$ juntamente com um gerador g de $\mathbb{Z}_p^*$. Conjectura-se que o problema do logaritmo discreto é difícil relativo a $\mathcal{G}_1$.

Mostre que o fato do problema do logaritmo discreto ser difícil relativo a $\mathcal{G}_1$ implica na existência de uma família de permutações unidirecionais. (**Dica:** Defina uma permutação sobre os elementos de $\mathbb{Z}_p^*$.)

Exercício 3 (Boneh (2009) (2,0)) Neste problema, veremos por que é uma má idéia escolher um primo $p = 2^k + 1$ para protocolos baseados no logaritmo discreto: o logaritmo discreto pode ser eficientemente calculado para primos como p.

Suponha que g seja um gerador para $\mathbb{Z}_p^*$.
(a) Mostre como se pode computar o bit menos significativo do logaritmo discreto. Ou seja, dado $y = g^x$ (com x desconhecido), mostre como determinar se x é par ou ímpar computando $y^{(p-1)/2} \bmod p$.
(b) Se x for par, mostre como computar o 2º bit mais significativo de x. (Dica: considere $y^{(p-1)/4} \bmod p$.
(c) Generalize a parte (b) e mostre como computar todos os bits de x. (Dica: seja $b \in \{0, 1\}$ bit menos significativo x obtido usando a parte (a). Tente inicializar $y_1 \leftarrow y/g^b$ e observe that y_1 é uma potência par g. Depois use a parte (b) para deduzir o segundo bit menos significativo de x. Mostre como iterar esse procedimento para recuperar todos os bits x.

(d) Explique brevemente por que seu algoritmo não funciona para um primo aleatório p.

Exercício 4 (Katz & Lindell (2007), 9.2 (2,0)) Descreva em detalhe um ataque de homem-no-meio sobre o protocolo de troca de chaves de Diffie–Hellman através do qual o adversário termina compartilhando uma chave k_A com Alice e uma chave (diferente) k_B com Bob, e Alice e Bob não conseguem detectar que algo andou errado.

O que acontece se Alice e Bob tentarem detectar a presença de um adversário homem-no-meio enviando um ao outro perguntas (encriptadas) que somente a outra parte saberia como responder?

Exercício 5 (Katz & Lindell (2007), 9.3 (2,0)) Considere o seguinte protocolo de troca de chaves:

(a) Alice escolhe $k, r \leftarrow \{0,1\}^n$ aleatoriamente, e envia $s := k \oplus r$ a Bob.

(b) Bob escolhe $t \leftarrow \{0,1\}^n$ aleatoriamente, e envia $u := s \oplus t$ a Alice.

(c) Alice computa $w := u \oplus r$ e envia w a Bob.

(d) Alice dá como saída k e Bob computa $w \oplus t$.

Mostre que Alice e Bob dão como saída a mesma chave. Analise a segurança do esquema (i.e., prove que é seguro, ou mostre um ataque concreto).