5. Considerando o protocolo da seção 5, um usuário A esqueceu-se de se desconectar com o sistema ("logout"). O oponente pode então se fazer passar pelo sistema perante o terminal, mesmo que a precaução de fazer r:= CS(n) for tomada, onde r é o valor aleatório e n a informação a ser digitada pelo usuário (fórmula (1) da seção 5).

6. Dada uma função unidirecional f, cada elemento da tabela de senhas de usuários contém $f^i(s)$ e i, onde s é a senha e i é inteiro (inicialmente i=1000, digamos). A cada conexão, o usuário fornece s; o terminal, de posse de i, computa $f^{i-1}(s)$ e envia o resultado ao sistema. Este computa $f^i(s)$ e substitui $f^i(s)$,i por $f^{i-1}(s)$,i-1, na tabela.

7. *Notas*

É um fato bem conhecido que a maioria dos usuários prefere nomes, datas de nascimento e outros dados de entes queridos. Segundo Morris e Thompson [MOTH], um oponente pode tentar também palavras de um dicionário, nomes em livros de nomes próprios, tudo isso soletrado de trás para frente, números de CPF, etc. Não há criptografia que resolva esse problema. O máximo que se pode fazer é deixar o ciframento das senhas muito lento, de forma a demorar o processo ao máximo, ou mesmo "congelar" o usuário e/ou o terminal depois de algumas tentativas.

A idéia de compressão de senha foi tirada da compressão de chave de Konheim e outros [KOAL], usada no IPS da IBM.