IA012 - Tarefa 6

Rodrigo Seiji Piubeli Hirao (186837)

16 de dezembro de 2021

01) Na primeira forma alguém pode se passar por A gerando um nounce novo já que não há mais nenhuma informação além do nounce, no segundo caso é possível mandar um conteúdo antigo por um terceiro para assim ser recebido ele decriptografado. A terceira forma é a única que não expõe a criptografia pelo nounce.

02) Caso algum terceiro consiga uma mensagem mais antiga, mas com a mesma chave de sessão, é possível enviar o texte como um nounce e receber o texto decriptografado.

03)

- Esquema I
  - I - Ana envia a Beto seu ID e um nounce criptografado com sua chave simétrica com a entidade de distribuição de chaves (KDC)
  - II - Beto envia à KDC os mesmos dados enviados por Ana mais seus dados
  - III - KDC retorna a Beto seus dados e a chave de sessão criptografados com a chave simétrica só conhecido por Beto e por KDC e também os dados de Ana no mesmo formato, mas com a chave simétrica dela com KDC
  - IV - Beto envia a Ana os dados enviados criptografados com a chave dela
- Esquema II
  - I - Ana envia a KDC um nounce com o pedido de iniciar a sessão
  - II - KDC responde com os mesmos dados, a chave de sessão e, criptografados com a chave de Beto, a chave de sessão e o id de Ana, tudo criptografaco com a chave de Ana
  - III - Ana envia os criptografados com a chave de Beto a Beto
  - IV - Beto confirma que é Ana enviando um novo nounce criptografado com a chave de sessão
  - V - Ana responde decriptografando o nounce, aplicando uma função simples, e criptografando o nounce novamente

Pode ser visto que que os esquemas usam da KDC para gerar uma chave de sessão de sendo que no primeiro esquema ele também é usado para enviar o nounce para garantir o instante de envio, enquanto no segundo esquema é feito após a troca de chaves com um segundo nounce. Assim deve haver uma comunicação dos 2 antes de ser pedido a chave a KDC no primeiro esquema.

04) No modo interno seria inviável, pois apenas é possível verificar que o hash do código corretor é diferente, mas não onde é diferente, pois o hash deve ser imprevisível e de mão única, no modo externo é possível arrumar o hash, mas não a mensagem original, o que também deixa inviável.

05) Invertendo a assinatura de A com a criptografia de B, assim só é preciso assinar 1 vez, mas deverá ser enviado toda a assinatura de A à B depois.

$$A \to X : ID_A || E_{K_{prA}}(ID_A || EK_{puB}(M))$$

06) É possível fazer um hash da mensagem e enviar o hash junto da mensagem, assim, se a mensagem for alterada, o hash também será.

07)

$$A \rightarrow B : ID_A||ID_B||EK_{puKDC}(N_A)$$
$$B \rightarrow KDC : ID_A||ID_B||EK_{puKDC}(N_A)||EK_{puKDC}(N_B)$$
$$KDC \rightarrow B : E_{K_{puB}}(K_{puA}||ID_A||ID_B||K_S)||E_{K_{puA}}(K_{puB}||N_A||N_B||ID_A||ID_B||K_S)$$
$$B \rightarrow A : E_{K_{puA}}(N_B||EK_{prKDC}(K_{puB}||N_A||ID_A||ID_B||K_S))$$
$$A \rightarrow B : E_{K_s}(N_B)$$

08) Pois o problema se assemelha ao problema do aniversário, que tem como solução de quantidade **k** de aniversariantes para ter **p** de probabilidade dos inteiros **d** é:

$$p = 1 - \frac{d}{d} \times \frac{d-1}{d} \times \cdots \times \frac{d-k}{d}$$
$$p = 1 - \prod_{i=1}^{\infty}(\frac{d-i}{d})$$
$$p = 1 - \prod_{i=1}^{\infty}(d - \frac{i}{d})$$
$$p \approx 1 - (\frac{d-1}{d})^{\frac{k(k-1)}{2}}$$

No caso do hash:

$$p \approx 1 - (\frac{2^n - 1}{2^n})^{\frac{k(k-1)}{2}}$$
$$k \approx \sqrt{2 \cdot 2^n \cdot \ln \frac{1}{1-p}}$$
$$k \approx \sqrt{2^{n+1} \cdot \ln (\frac{1}{1-p})}$$

Assim k se aproxima de $2^{\frac{n}{2}}$

09)

1. O sal é um número pseudo-aleatório, no caso no Unix são 12 bits baseados no relógio do sistema.
2. O sal faz com que mensagens iguais tenham hashes diferentes, assim não se pode fazer uma análise num grupo de hashs para procurar por padrões.
3. Pois o sal não é usado para aumentar a segurança do algoritmo de hash, mas sim de mensagens iguais terem diferentes hashs, logo a chance de ter 2 mensagens iguais e ainda com hashes iguais é muito pequena já com 12 bits.
4. Não irá fazer muita diferença, pois inicialmente o número aleatório com poucos bits já deve comportar a quantidade de usuários.
5. Vai sendo percebido uma melhora na segurança até um ponto que não é mais percebido uma diferença, pelo mesmo motivo explicado anteriormente

10) É passada a senha por **r** funções de hash $h_i = H_i(r), 0 < i \leq r$, e então é ligado o bit h_i do vetor de hash. Ele é melhor pois o vetor de hash pode ser armazenado para um dicionário inteiro e ocupar menor espaço.