Funções *Hash* – Segurança de Sistemas

Rodrigo L. da Silveira[1]

Escola Politécnica – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
90.619-900 – Porto Alegre – RS – Brasil

rodrigo.silveira@edu.pucrs.br

Abstract. *This paper describes the implementation of the third work of the Systems Security discipline, where the problem is to guarantee the authenticity of a large file when being downloaded from a server, and the proposed solution is the application of hash function, where each block of the downloaded file will have the hash calculated for the next block, making it possible to check the blocks received and guarantee that there are no changes to the file.*

Resumo. *Este artigo descreve a implementação do terceiro trabalho da disciplina de Segurança de Sistemas, onde o problema consiste em garantir a autenticidade de um grande arquivo ao ser baixado de um servidor, e a solução proposta é a aplicação da criptografia de função resumo, onde cada bloco do arquivo baixado terá o hash calculado para o próximo bloco, possibilitando a verificação dos blocos recebido e garantido que não ouve alterações no arquivo.*

1. Introdução

O terceiro trabalho da disciplina de Segurança de Sistemas consiste na implementação de um sistema capaz de garantir a autenticidade da transmissão de um vídeo de um servidor à um cliente sem que seja necessário enviar o vídeo inteiro para verificação.

Supondo que um website armazene um grande arquivo de vídeo em que qualquer um possa fazer o download. Os browsers precisam garantir que o video que estão baixando é autêntico antes de mostrar o conteúdo ao usuário. Para isto, uma solução possível é a verificação do arquivo através de uma função *hash*, onde o browser teria acesso ao *hash* do arquivo através de um canal seguro e, após o download do arquivo, ele verificaria a autenticidade calculando o *hash* do arquivo baixado e comparando-o com o *hash* obtido. Porém, esta solução obrigaria o browser a baixar todo o vídeo antes de mostrá-lo ao usuário, o que tornaria o processo todo muito lento.

2. Proposta

Como uma solução viável a este problema, o enunciado do trabalho propõe a criação de um algoritmo que seja capaz de dividir o video em blocos de 1024 bytes e calcular o *hash* de cada bloco concatenando o bloco atual com o *hash* do próximo bloco, onde o browser, com o *hash* do primeiro bloco recebido por um canal seguro, seria capaz de verificar o bloco 0 recebido. Cada bloco contém 1024 bytes de conteúdo e mais o *hash* do próximo bloco, que será utilizado para verificar a autenticidade do bloco subsequente até que todo o arquivo seja recebido. A figura 1 demostra como o algoritmo proposto deverá concatenar os *hashes*.

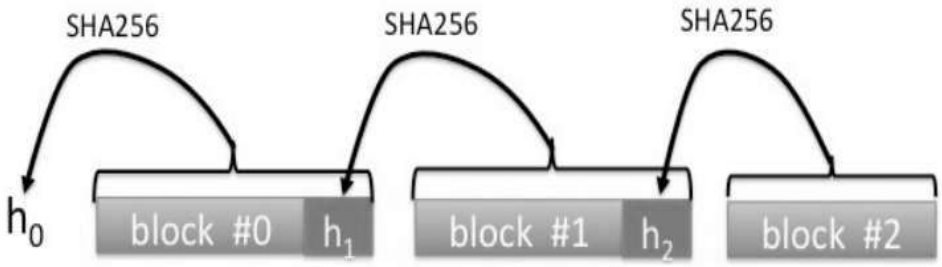

Figura 1. Diagrama da solução proposta.

3. Desenvolvimento

A implementação do algoritmo foi dividida em 3 etapas: 1) a divisão do arquivo em blocos, 2) o cálculo do *hash* para cada blocos e, 3) como saída do algoritmo, os resultados gravados em um arquivo de texto.

Na primeira etapa, é feito o cálculo de quantos blocos será necessário para dividir o arquivo e a efetiva divisão em blocos, foi utilizada uma pilha para armazenar os blocos, visto que, conforme a figura 1 demostra, os *hashes* deverão ser calculados do último bloco para o primeiro.

Na segunda etapa, os blocos são removidos do topo da pilha e, exceto para o primeiro, é concatenado o *hash* do bloco processado anteriormente com os dados do bloco atual do arquivo e gerado um novo *hash*, que ficará armazenado em uma variável temporária para a próxima iteração, cada *hash* gerado é armazenado em uma nova pilha de *hashes*, que ordenará os *hashs* dos blocos para a saída do algoritmo. A função *hash* a ser utilizada é a SHA-256.

Por fim, a etapa 3 irá converter os bytes dos *hashes* para hexadecimal e grava-los em um arquivo texto na pasta output para que possa ser feita a verificação dos *hashes* gerados.

O ambiente de programação utilizado foi o *dotnet core* 3.1 em uma máquina *Windows*.

4. Resultados

Conforme sugerido no enunciado, foi utilizado o arquivo FuncoesResumo - SHA1.mp4 para validação do algoritmo, onde os *hashes* obtidos para o primeiro e o último bloco estão de acordo com os *hashes* previamente informados.

Como resultado da execução do algoritmo para o arquivo FuncoesResumo - Hash Functions.mp4, obtivemos o seguinte *hash* para o primeiro bloco:

```
0: 45013B3A2D5BC5369B90125DA1DC55D2903C47C3852E13B04878DF9942F21B9D
```

O código fonte e os resultados completos podem ser conferidos nos anexos submetidos junto a este relatório no *Moodle* ou no seguinte repositório do *GitHub*.

Referências

Silveira, Rodrigo. (2020) “T3 de Segurança de Sistemas”, https://github.com/RLSilveira/SHA256_T3, 28 de Maio de 2020.