# Pontifícia Universidade Católica de Campinas

Raissa Furlan Davinha RA: 15032006

José Carlos Clausen Neto RA: 15055825

Luiz Felipe Zerbetto Masson RA: 15166804

Janaina Sanches RA: 07270085

Yessica Melaine P. Castillo RA: 13054895

Danilo Luís Lopes Raymundo Paixão RA: 15051659

**MÓDULO DE CRIPTOGRAFIA KERNEL LINUX**

Projeto Sistemas Operacionais B

Campinas

2017

**INTRODUÇÃO**

O objetivo do experimento é a familiarização com a implementação de um módulo do Kernel fazendo uso de uma API criptográfica, já incluso no kernel desde a versão 2.5.4. O módulo, acompanhado de um programa em nível usuário, é capaz de cifrar, decifrar e calcular resumo criptográfico de dados que são fornecidos pelo usuário.

**DESENVOLVIMENTO**

O primeiro passo foi buscar a compreensão do uso da API de criptografia, como usar e como funciona. O modelo para o módulo foi tirado do projeto anterior, onde quando iniciado, o mesmo recebe uma string como parâmetro, que no projeto atual, será usado como chave para criptografar os dados.

Primeiramente, deve-se saber que o objetivo da criptografia é garantir que os intervenientes na troca de informação tenham garantias de que os requisitos de segurança foram satisfeitos. Também deve-se saber a definição de uma cifra que é um algoritmo criptográfico que faz as transformações entre o texto limpo e o criptograma. Além disso, reconhecer que a chave é quem controla o comportamento do funcionamento interno quando é de conhecimento geral.

As funções de hash criptográficas (também chamadas funções de message digests, etc.) transformam um input de tamanho variável num output de tamanho (menor) fixo. Para garantir integridade interessa extrair uma impressão digital não invertível de uma mensagem: obter um valor que identifique o conteúdo essa mensagem. Como as funções de hash não são injetivas, a identificação não pode ser unívoca: várias mensagens serão mapeadas no mesmo valor de hash.

O SHA foi desenvolvido pelas agências governamentais americanas NIST e NSA para incluir no standard de assinaturas digitais DSS. Atualmente utiliza-se uma versão melhorada: o SHA-1. O valor de hash produzido pelo algoritmo é de 160 bits.

O algoritmo SHA1, no seu funcionamento lógico acrescenta bits de enchimento e acrescenta um bloco de 64 bits a mensagem; inicia um buffer de 160 bits para manter os resultados intermediários e final da função hash, representado por cinco registradores que são iniciados com valores em valores hexadecimal;processamento da mensagem em blocos de 512 bits em quatro interações, após a última interação tem a saída do hash de 160 bits.

**Referências**

The Linux Kernel Module Programming Guide. Capítulo 2.6, Passing Command Line Arguments to a Module. Disponível em: <http://www.tldp.org/LDP/lkmpg/2.6/html/x323.html>. Acesso em: 10 de Setembro de 2017.

UBUNTU Kernel Crypto API Cryptographic Module. Version 1.0. Disponível em: <https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp2962.pdf>. Acesso em: 20 de Setembro de 2017.

Linux Documentation Crypto API.FreeEletrons. Dsiponível em: < http://elixir.free-electrons.com/linux/latest/source/Documentation/crypto/api-intro.txt>. Acesso em: 28 de setembro de 2017.

Crypto API (Linux). Disponível em:

<https://en.wikipedia.org/wiki/Crypto\_API\_(Linux)> Acesso em: 01 de Outubro de 2017.

How to use CryptoAPI in the Linux Kernel 2.6. Disponível em: <https://stackoverflow.com/questions/3869028/how-to-use-cryptoapi-in-the-linux-kernel-2-6>. Acesso em: 04 de Outubro de 2017.

Criptografia Módulo I – Terminologia Disponível em:

<http://twiki.di.uminho.pt/twiki/pub/Education/Criptografia/CriptografiaBiomedica0607/Cripto-Mod1.pdf> Acesso em: 09 de Outubro de 2017.