UNIVERSIDADE FEDERAL DE GOIÁS
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA
E DE COMPUTAÇÃO

CORINDA: QUEBRANDO HASHES DE SENHAS CONCORRENTEMENTE COM A LINGUAGEM DE PROGRAMAÇÃO GO

Bernardo Araujo Rodrigues

[UFG] & [EMC]
[Goiânia - Goiás - Brasil]
24 de janeiro de 2018

UNIVERSIDADE FEDERAL DE GOIÁS
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA
E DE COMPUTAÇÃO

CORINDA: QUEBRANDO HASHES DE SENHAS CONCORRENTEMENTE COM A LINGUAGEM DE PROGRAMAÇÃO GO

Bernardo Araujo Rodrigues

Dissertação apresentada a Banca Examinadora como exigência parcial para a obtenção do título de Mestre em Engenharia Elétrica e de Computação pela Universidade Federal de Goiás (UFG), Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), sob a orientação do Prof. Dr. Wesley Pacheco Calixto.

[UFG] & [EMC]
[Goiânia - Goiás - Brasil]
24 de janeiro de 2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistemas da Bibliotecas da UFG, GO - Brasil

C331s Rodrigues, Bernardo Araujo, 31/07/90.
Corinda: Quebrando *Hashes* de Senhas Concorrentemente com a Linguagem de Programao Go [manuscrito]/ Bernardo Araujo Rodrigues. – [Goiânia - Goiás - Brasil]: [UFG] & [EMC], 24 de janeiro de 2018.
48 f. : il.

Orientador: Wesley Pacheco Calixto - UFG

Dissertação - Universidade Federal de Goiás - UFG, Escola de Engenharia Elétrica, Mecânica e de Computação - EMC

Inclui bibliografia.

1.Senhas - Teses. 2.*Hashes* - Teses. 3.Segurança da Informação - Teses. I. Calixto, Wesley Pacheco; II. Universidade Federal de Goiás. Programa de Pós-Graduação em Engenharia Elétrica e de Computação. III. Corinda: Quebrando *Hashes* de Senhas Concorrentemente com a Linguagem de Programao Go

CDU 000.0.000:000.0

"Escreva aqui a epígrafe."

Escreva aqui o nome do autor da epígrafe

Testando acentuação.

AGRADECIMENTOS

AgradeÃ§o ao orientador e amigo Wesley Pacheco Calixto, por sua dedicaÃ§Ã£o e compreensÃ£o. Por ter acreditado em mim desde o inÃcio, e ter permitido que eu seguisse o caminho onde pude florescer.

Aos colegas do NExT (Nucleo de Estudos Experimentais e Tecnologicos), pelo companheirismo, dedicacao e amizade.

A minha familia, pelo amor e incentivo incondicional, seja em momentos de dificuldade ou de alegria.

Aos colegas de trabalho e amigos da Data Traffic, pela compreensao, incentivo, e risadas.

A CAPES pelo aporte para o desenvolvimento deste trabalho.

RESUMO

Escreva aqui o resumo do seu trabalho... Testando acentuação.

CORINDA: CRACKING PASSWORD HASHES WITH GO PROGRAMMING LANGUAGE

ABSTRACT

Write here the abstract of your work ...

SUMÁRIO

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE TABELAS

Pág.

LISTA DE SÍMBOLOS

Coloque seus símbolos aqui conforme exemplo:

N_m	–	Horizonte do modelo
N_u	–	Horizonte de controle
N_y	–	Horizonte de predição
α	–	Taxa de amortecimento da referência
ω	–	Trajetória referência

LISTA DE ABREVIATURAS E SIGLAS

Coloque suas abreviaturas aqui conforme exemplo.

AG	–	Algoritmo Genético
CA	–	Corrente Alternada
CC	–	Corrente Contínua
CMD	–	Controle por Matriz Dinâmica
CPBM	–	Controle Preditivo Baseado em Modelos
CPBML	–	Controle Preditivo Baseado em Modelo Linear
CPBMNL	–	Controle Preditivo Baseado em Modelo Não Linear
CPPNL	–	Controle Preditivo Prático Não Linear
CPG	–	Controle Preditivo Generalizado
IAE	–	Integral do valor absoluto do erro
ISE	–	Integral do erro quadrático
ITAE	–	Integral do valor absoluto do erro multiplicado pelo tempo
MCC	–	Motor de Corrente Contínua
MIMO	–	Sistemas Multivariáveis
PID	–	Proporcional, Integral e Derivativo
SISO	–	Sistemas Monovariáveis

CAPÍTULO 1

INTRODUÇÃO (xx/02/2018)

Conceitos como Internet das Coisas, Computação na Nuvem e Redes Sociais possuem papel cada vez mais fundamental no desenvolvimento social e econômico da sociedade atual. Isso faz com que o tema da Segurança da Informação seja cada vez mais importante (**??**). Com isto, qualquer vulnerabilidade inerente a uma tecnologia amplamente utilizada torna-se alvo de interesse ao público em geral.

Senhas são objetos importantes em sistemas computacionais, sendo essenciais para garantir a segurança de informações pessoais ou sigilosas, bem como acesso a variedade de dispositivos inteligentes (**??**). Enquanto diversos tipos alternativos de autenticação têm sido implementados, senhas baseadas em sequências de caracteres ainda são o tipo de autenticação mais comum (**????????**).

Combinações frágeis de *username* e senhas fazem com que dispositivos e contas sejam facilmente comprometidas por agentes maliciosos (**??????**). Além de senhas fracas, a reutilização de senhas é prática comum entre usuários (**??**). Tais práticas abrem espaço para técnicas de engenharia social e escalada de privilégios, tais como as utilizadas no vazamentos de dados do serviço de armazenamento de arquivos *Dropbox* em 2016, onde agentes maliciosos se aproveitaram de senhas reutilizadas para obter acesso ao sistema da empresa (**??**).

Funções de dispersão criptográficas (FDC) geram longa e complexa sequência de caracteres (chamada *hash*) a partir de informação inicial, sendo matematicamente impossível recuperar a informação original a partir do *hash* calculado, permitindo que tal informação seja armazenada secretamente. FDC são funções unidirecionais, o que significa que a única forma de obter determinado *hash* é fornecendo a informação original como entrada da função (**????**). Atualmente, sistemas computacionais armazenam apenas o *hash* da senha do usuário. A autenticação ocorre quando o *hash* calculado a partir da senha digitada pelo usuário é comparada com o *hash* armazenado no banco de dados, permitindo o acesso caso os valores sejam iguais (**??**).

No contexto de modelagem de ameaças, o indivíduo malicioso tentando obter a senha em questão é chamado de atacante. O administrador do sistema ou especialista que tenta previnir que a senha seja comprometida é chamado de defensor (**??**). De forma a obter a senha, o atacante deve encontrar a sequência de caracteres cuja FDC

corresponde ao *hash* roubado do banco de dados. Tal processo de adivinhação é chamado de quebra da senha, e existem ferramentas específicas capazes de realizar milhões de palpites por segundo, tais como *John The Ripper* e *Hashcat* (**????**). *Hashcat* é a ferramenta mais avançada e popular atualmente, tornando possível o uso de técnicas de paralelização baseadas no uso de *Open Computing Language* (*OpenCL*), descrita por **??**). O *Hashcat* pode ser utilizado em diversas plataformas de *hardware*, tais como Unidades de Processamento Central (*CPU*), Unidades de Processamento Gráfico (*GPU*), Processadores Digitais de Sinais (*DSP*), e Arranjos de Portas Programáveis em Campo (*FPGA*).

Técnicas de quebra de senhas tem sido amplamente pesquisadas, não apenas pela comunidade científica, mas também por comunidades online e fóruns (**??????**). Em 2016, mais de 96% dos *hashes* vazados dos bancos de dados da rede social de negócios *LinkedIn*, descritos por **??**), foram quebrados menos de cinco meses depois de terem sido disponibilizados online. Se as senhas forem simples o bastante, mesmo FDC robustas não resitem a técnicas modernas de quebra de senha.

Quando usuários criam suas senhas, a maioria tende a utilizar padrões que possam ser facilmente lembrados. Isto faz com que seja possível que o atacante consiga adivinhá-la. Alguns autores trabalharam com a tentativa de encontrar padrões na forma que os usuários criam suas senhas. **??**) utilizam gramáticas probabilísticas livre de contexto como entrada em *software* de quebra de senha. **??**) procuram por padrões comuns tais como números de caracteres e palavras de dicionários. Outros pesquisadores também tentaram elaborar métricas confiáveis capazes de quantificar o quão robusta determinada senha é frente a tentativas de quebra. **??**) utilizam Modelos de Markov para modelar a força de senhas. O padrão definido por **????**) nas diretrizes de autenticação online do *National Institute of Standards and Technology* (*NIST*) propõe a entropia de caractere como métrica de força. **??**) avaliam a métrica do *NIST* como útil, porém limitada na maioria dos casos. **??**) propõem como métrica a entropia empírica baseada em senhas previamente coletadas, também com resultados limitados.

De forma a estabelecer *framework* generalizado para a quantificação da resistência de determinada senha contra técnicas de quebra, **??**) formaliza os conceitos de **complexidade** e **força** com sólidas definições matemáticas que enfatizam suas diferenças. Estes conceitos possuem importância fundamental no contexto de quebra de senhas, uma vez que eles derivam da compreensão fundamental de que:

> *...o sucesso do atacante ao quebrar uma senha deve ser definido pelos*

> *seus recursos computacionais disponíveis, tempo disponível, FDC utilizada, bem como a topologia que define o espaço de busca (??).*

Com o objetivo de avaliar os conceitos de complexidade e força estabelecidos por **??**), **??**) realizam série de experimentos. Entre as conclusões, observa-se que existem padrões estatísticos distintos na distribuição das senhas analisadas. A possibilidade de criação de métrica que reflita tais padrões estatísticos, bem como o conceito de complexidade estabelecido por **??**), justificam este estudo.

- Hipótese
- Objetivos
- Estrutura do trabalho

CAPÍTULO 2

CONTROLE MODERNO (xx/xx/2017)

Este capítulo descreve o fenômeno de vazamento de dados de autenticação. Algumas listas conhecidas são apresentadas, bem como a ferramenta de avaliação de complexidade de senhas chamada Passfault.

2.1 Informação Sigilosa

Na maioria das organizações, as informações relativas à autenticação de usuários ficam armazenados em servidores. Tais servidores encontram-se instalados em complexas infraestruturas de rede, com diversos pontos que podem servir de entrada para agentes maliciosos, caso sejam mal configurados ou estejam desatualizados. Por razões que passam por ativismo político, crime organizado e até guerras cibernéticas entre nações, tais servidores são invadidos e informações de autenticação de usuários são roubadas.

De forma a mitigar os danos no caso do vazamento dos dados de autenticação, a utilização de função de dispersão criptográfica (FDC) para criptografar senhas armazenadas é prática comum. Apenas o *hash* da senha é armazenado, e a autenticação ocorre quando o *hash* calculado a partir da senha digitada pelo usuário é comparada com aquele armazenado no banco de dados, permitindo o acesso caso os valores sejam idênticos.

Quando as listas de *usernames* e *hashes* são roubadas, várias vezes seus destinos são mercados negros *online*. Eventualmente, estas listas acabam emergindo para a superfície da internet. Comunidades inteiras se formam para compartilhar técnicas e resultados de quebra de *hashes*. Exemplo disto é a ferramenta *Hashcat*, projeto de código aberto resultado de esforço puramente comunitário. Quando ocorre o vazamento público dos dados de alguma corporação, curto tempo se decorre até que elevada porcentagem dos *hashes* seja quebrada.

Tabelas de consulta com *hashes* previamente computados são utilizados para reduzir o tempo de quebra dos mesmos. Caso considerável número de usuários utilizem a mesma senha em determinada lista, basta que a primeira senha seja quebrada para que todas iguais a ela sejam comprometidas. De forma a diminuir a eficácia das tabelas das *hashes* pré computados, a técnica de *salting* é prática de segurança recomendada por especialistas. A técnica consiste em concatenar caracteres aleatórios (*salt*) à senha antes do cálculo do *hash* via FDC. O *salt* deve ser armazenado de

forma segura. Durante a autenticação, a senha fornecida pelo usuário mais o *salt* armazenado são usadas no cálculo do *hash*. Contudo, caso os *salts* também sejam roubados junto aos *hashes*, tal técnica não impede que os *salts* sejam usados para computar a nova tabela de consulta.

Existem algumas listas de *hashes* roubadas que são conhecidas, como: i) RockYou, ii) LinkedIn, e iii) Anti Public.

RAINBOW TABLE

2.2 RockYou

O RockYou fornece serviços de jogos *on-line*. Em Dezembro de 2009, a empresa de segurança da informação emitiu notificação sobre séria falha de segurança do tipo SQL Injection nos servidores da RockYou, que alegou ter resolvido o problema. Este tipo de falha já havia sido amplamente documentada e estudada por cerca de uma década. Apesar das alegações da RockYou sobre terem corrigido o problema, agentes maliciosos foram capazes de explorar tal falha e roubar cerca de 32 milhões de senhas na forma *plaintext*, ou seja, sem nenhuma proteção por FDC.

Este acontecimento foi amplamente divulgado na mídia logo após o roubo. Especialistas de segurança aconselharam aos usuários do *site* que trocassem todas suas senhas imediatamente, uma vez que as senhas roubadas poderiam ser utilizadas para roubar contas em outros *sites* (**??**).

Desde então, esta lista de senhas tem sido amplamente utilizada para fins de pesquisa.

2.3 LinkedIn

LinkedIn consiste de rede social de contatos profissionais. Em Junho de 2012, os *hashes* das senhas de cerca de 6,5 milhões de usuários são roubadas por criminosos russos. As senhas eram protegidas pela FDC *Secure Hashing Algorithm 1* (*SHA1*), sem *salting*. Estes usuários não puderam mais acessar suas contas, e a empresa encorajou que todos seus usuários mudassem suas senhas. No mesmo dia, os *hashes* foram quebrados e postados em fóruns *online* (**??**).

Em Maio de 2016, mais 117 milhões de *hashes* aparecem na internet. Especula-se que esta lista foi roubada no mesmo incidente de 2012, contudo permanecendo em mercados negros *online* antes de ser descoberta. Como não havia proteção por

salting, mais de 90% da lista de *hashes* foi quebrada em menos de 72 horas. Como resposta, o LinkedIn invalidou as senhas de todos os usuários que não mudaram a senha desde 2012 (**??**).

2.4 *Anti Public*

A lista conhecida como *Anti Public* surgiu na internet em Dezembro de 2016 por meio de fóruns russos. A lista *Anti Public* foi amplamente comercializada em mercados negros. Quase nada se sabe sobre sua verdadeira origem, mas especula-se que ela seja o compilado geral de diversos vazamentos. Tais listas são comumente chamadas de listas *combo*.

A lista contém cerca de 458 milhões de *e-mails* diferentes, alguns com múltiplas senhas. A lista também é utilizada em prática conhecida como *credential stuffing*, que consiste em automatizar a autenticação em diversos *websites* utilizando diferentes credenciais da lista (**??**).

2.5 Vieses Psicológicos

Comumente estudados na psicologia comportamental, vieses psicológicos (ou cognitivos) consistem de tendências a pensamentos e decisões que desviam sistematicamente da racionalidade padrão (**??**). Vieses psicológicos podem ser vistos como heurísticas que o cérebro utiliza como atalhos no processamento de informação.

No contexto deste trabalho, tais vieses se manifestam estatisticamente, na tendência dos usuários a criarem senhas seguindo padrões frequentes. As reais causas e mecanismos por trás destes vieses psicológicos fogem o escopo deste trabalho, mas especula-se que a facilidade de armazenamento na memória do indivíduo, bem como os mecanismos linguísticos do cérebro, possuam papel importante na manifestação deste fenômeno.

Por exemplo, a Tabela (2.1) lista as dez senhas mais frequentes na lista roubada da rede social LinkedIn. Nota-se que sequências numéricas são um padrão recorrente. Palavras associadas ao contexto da senha (**linkedin** e **password**) também mostram-se bastante populares. Estes são exemplos de vieses psicológicos que delimitam o espaço de busca para senhas prováveis, reduzindo sua ordem de grandeza de trilhões de senhas para algumas dezenas de milhares. Isso faz com que o *hash* da senha seja facilmente quebrado por agentes maliciosos experientes.

Tabela 2.1 - Dez senhas mais frequentes da lista LinkedIn

Senha	**Frequência**
123456	753.305
linkedin	172.523
password	144.458
123456789	94.314
12345678	63.769
111111	57.210
1234567	49.652
sunshine	39.118
qwerty	37.538
654321	33.854

2.6 Passfault

Criada em 2001, a *Open Web Application Security Project* (*OWASP*) é a comunidade *online* com o objetivo de produzir artigos, metodologias, documentação, ferramentas e tecnologias no campo da segurança de aplicações *web*.

Mantido pela OWASP, o Passfault é ferramenta de código aberto, implementada em *Java* e distribuída sob *Apache Licence 2.0*. Seu desenvolvimento começou em 2011, com o objetivo de fazer com que a complexidade e força das senhas sejam facilmente entendidas pela população. O Passfault recebe como entrada uma senha, uma opção de FDC e uma opção de *hardware* adversário. Ele analisa a estrutura da senha, tentando encontrar o modelo que melhor descreve a mesma. Então, o Passfault informa o usuário da complexidade da senha (tamanho do espaço de busca), bem como o tempo necessário para quebrá-la, baseado na FDC e no *hardware* escolhido (**??**).

O código do Passfault pode ser encontrado em repositório aberto em https://github.com/OWASP/passfault.

2.7 Considerações

Muitas organizações ainda não utilizam boas práticas de segurança como o *salting*. Ainda assim, técnicas de quebra de *hashes* continuam em constante evolução, e os vazamentos de credenciais continuam acontecendo com frequência cada vez maior. Assim, é importante a conscientização dos usuários para que utilizem senhas fortes de forma a dificultar a quebra do *hash*. O capítulo a seguir descreve a métrica de

força de senha.

CAPÍTULO 3

SISTEMA, MODELO E OTIMIZAÇÃO (xx/xx/2017)

Este capítulo formaliza as principais definições da métrica de força. O embasamento teórico dos conceitos como: i) *string*, ii) modelo, iii) frequência relativa, iv) função de dispersão criptográfica, v) processo de quebra e vi) força do modelo são apresentados, consistindo da união das teorias dos conjuntos e dos modelos de primeira ordem e da inferência estatística.

3.1 String

Seja c o símbolo pertencente ao conjunto universo de símbolos A. Chama-se c de **caractere**. Seja A o conjunto universo de todos os Caracteres possíveis. Por exemplo, A pode ser o conjunto de caracteres ASCII. Chama-se A de **alfabeto**.

$$A = \{c_{i=1}, \cdots, c_N\} \tag{3.1}$$

Seja s o multiconjunto bem ordenado de Caracteres. Chama-se s de **string**.

$$s = \{c_{j=1}, \cdots, c_M | c_j \in A\} \tag{3.2}$$

Seja $\mathbb{X}(A)$ o conjunto de todas strings possíveis sobre A.

$$\mathbb{X}(A) = \wp(A^n), n \to \infty \tag{3.3}$$

Aqui, $\wp(x)$ representa o conjunto potência de x, e A^n representa o n-ésimo produto cartesiano de A.

Seja $\tilde{s}$ o multiconjunto bem ordenado de strings. Chama-se $\tilde{s}$ de **string composta**.

$$\tilde{s} = \{s_{k=1}, \cdots, s_L\} \tag{3.4}$$

Como qualquer string com mais de um caractere pode ser considerada string composta, utiliza-se os termos string e string composta indistintamente.

Seja Γ o multiconjunto de strings resultado do processo de amostragem. Chama-se Γ de **multiconjunto de amostras**.

3.1.1 Operações com Strings

A partir das operações de partição e concatenação, é possível transformar strings compostas em strings elementares e vice-versa. Define-se a operação de **partição** de string composta $\Psi(\tilde{s})$, como a partição do multiconjunto $\tilde{s}$ em subconjuntos s_i, dada por:

$$\Psi(\tilde{s}) = s_{i=1}|\cdots|s_N \tag{3.5}$$

Por exemplo, seja $\tilde{s} =$ "$psword1$". Possíveis partições de $\tilde{s}$ são:

$$\Psi_1(\tilde{s}) = \{\{\text{“}p\text{”}\}, \{\text{“}s\text{”}\}, \{\text{“}word\text{”}\}, \{\text{“1”}\}\} \tag{3.6}$$

$$\Psi_2(\tilde{s}) = \{\{\text{“}ps\text{”}\}, \{\text{“}word\text{”}\}, \{\text{“1”}\}\} \tag{3.7}$$

$$\Psi_3(\tilde{s}) = \{\{\text{“}psword\text{”}\}, \{\text{“1”}\}\} \tag{3.8}$$

Define-se a operação de **concatenação** de strings $\Psi^{-1}(s_{i=1}|\cdots|s_N)$ como a operação inversa da partição, tal que:

$$\Psi^{-1}(s_{i=1}|\cdots|s_N) = \tilde{s} \tag{3.9}$$

As concatenações dos subconjuntos em (3.6), (3.7), (3.8) são dados por:

$$\Psi_1^{-1}(\{\{\text{“}p\text{”}\}, \{\text{“}s\text{”}\}, \{\text{“}word\text{”}\}, \{\text{“1”}\}\}) = \text{“}psword1\text{”} \tag{3.10}$$

$$\Psi_2^{-1}(\{\{\text{“}ps\text{”}\}, \{\text{“}word\text{”}\}, \{\text{“1”}\}\}) = \text{“}psword1\text{”} \tag{3.11}$$

$$\Psi_3^{-1}(\{\{\text{“}psword\text{”}\},\{\text{“}1\text{”}\}\}) = \text{“}psword1\text{”} \tag{3.12}$$

3.2 Frequência Relativa

Como $\mathcal{M}_\Gamma$ é multiconjunto, podem ocorrer membros $\hat{m}_i$ repetidos. Seja n_i o número de ocorrências de cada $\hat{m}_i$ em $\mathcal{M}_\Gamma$, e seja $|\mathcal{M}_\Gamma|$ o número total de membros de $\mathcal{M}_\Gamma$, incluindo repetições. Define-se então θ_i como a **frequência relativa** de $\hat{m}_i$ em $\mathcal{M}_\Gamma$ (**??**):

$$\theta_i = \frac{n_i}{|\mathcal{M}_\Gamma|} \tag{3.13}$$

3.3 Função de Dispersão Criptográfica

A função de dispersão criptográfica (FDC) é uma função cuja inversão é considerada praticamente irrealizável, ou seja, é quase impossível recriar os valores de entrada, utilizando somente o valor resultante da dispersão. Define-se a FDC $H(\tilde{s})$ como função unidirecional que toma como entrada string de tamanho arbitrário $\tilde{s}$ e retorna como saída string de tamanho fixo h, chamada ***hash*** (**????**). A FDC possui as seguintes características: i) dado $\tilde{s}$, é trivial computar $h = H(\tilde{s})$; ii) dado h, é computacionalmente difícil encontrar $\tilde{s}$ tal que $h = H(\tilde{s})$ e iii) dado $\tilde{s}$, é computacionalmente difícil encontrar $\tilde{s}'$ tal que $H(\tilde{s}) = H(\tilde{s}')$.

3.4 Processo de Quebra

O **processo de quebra** do *hash* h, consiste da tarefa de encontrar string $\tilde{s}$ tal que $H(\tilde{s}) = h$. Para isto, o Processo consiste em calcular a FDC $H(\tilde{s}_i)$ de cada string $\tilde{s}_i$ pertencente ao domínio $\hat{\lambda}$ do modelo crítico $\hat{m}$, até que seja encontrada $H(\tilde{s}_i) = h$.

Dependendo da FDC escolhida e da capacidade computacional utilizada no processo de quebra, caso o domínio $\hat{\lambda}$ seja grande o suficiente, o processo pode estender-se por período de tempo extremamente longo, fazendo com que o mesmo torne-se inviável.

3.5 Força do Modelo

A **força do modelo** $\hat{m}$ é definida quando alguma string de seu domínio é submetida a processos de quebra. Denota-se a força do modelo $\hat{m}$ por $\mathfrak{S}(\mathcal{C}, \theta)$, onde $\mathcal{C}$ é a cardinalidade de $\hat{m}$ e θ é a frequência relativa de $\hat{m}$ em $\mathcal{M}_\Gamma$.

$$\mathfrak{S}(\mathcal{C},\theta) = (1-\theta)^a, a = \frac{1}{\log_{10}(\mathcal{C})} \tag{3.14}$$

Observa-se que apesar de $\mathfrak{S}(\mathcal{C},\theta)$ estar relacionada ao modelo crítico $\hat{m}$, chama-se de força do modelo por simplicidade. Observa-se também que modelos fortes geram $\mathfrak{S}(\mathcal{C},\theta) \approx 1$, enquanto modelos fracos geram $\mathfrak{S}(\mathcal{C},\theta) \approx 0$.

Esta métrica de força, diferentemente de outras estabelecidas na literatura (**??????**), leva em conta a frequência relativa θ do modelo em conjunto de amostras. Se o conjunto de amostras Γ utilizado para obter $\mathcal{M}_\Gamma$ e θ, possuir desejada representatividade estatística da distribuição de modelos críticos na vida real, pode-se afirmar que a probabilidade de que o domínio $\hat{\lambda}$ do modelo $\hat{m}$ seja usado em processos de quebra é igual a θ, e portanto, a probabilidade de que $\hat{\lambda}$ **não** será utilizado é $(1-\theta)$. Isto é ponto de partida favorável para a métrica de força, uma vez que modelos populares terão alta probabilidade de ser bem sucedidos em processos de quebra.

O termo a em (3.14), denominado de **fator de cardinalidade**, é utilizado para parametrizar a cardinalidade do modelo. Assim, a métrica recompensa modelos com altas cardinalidades e baixas frequências relativas, classificando-os como fortes. Por outro lado, modelos com baixas cardinalidades e altas frequências relativas são penalizados, sendo classificados como fracos.

A Figura **??** ilustra o mapa de $\mathfrak{S}(\mathcal{C},\theta)$, que foi gerado na linguagem *Python 3.6*, com auxílio dos pacotes *NumPy* e *Matplotlib*. Observa-se que $\mathcal{C}_{min} = 1$, e portanto, $\log_{10}(\mathcal{C}_{min}) = 0$, teoricamente a pode resultar em divisão por *zero*. Contudo, o interpretador *IPython* trata tal caso associando $a = \infty$ e gerando *warning*, conforme ilustrado na Figura **??**.

Na exceção de $\mathfrak{S}(1,0) = (1)^\infty = 1$, o interpretador *Python* atribui $\mathfrak{S}(1,\theta) = (1-\theta)^\infty = 0$, como ilustrado na Figura **??**. Ou seja, independente da frequência relativa θ, se a cardinalidade do modelo é mínima, a força também é mínima. Observa-se contudo, que $\mathfrak{S}(1,0) = 1$ não é problema, uma vez que o modelo que nunca ocorre em $\mathcal{M}_\Gamma$, de fato é forte, não importa quão ínfima seja sua cardinalidade $\mathcal{C}$.

3.6 Entropia do Modelo

- Transmissão de uma mensagem: simbolos, alfabeto de signos, mensagem, esquema de codificação
- codificacao: tradução de signos para simbolos

- Entropia: quantidade de informação contida em cada simbolo de determinada codificação. Por isso pode se entender como a quantidade média de perguntas que precisam ser feitas até que se descubra qual o signo codificado.

- Quanto mais signos possiveis, maior a entropia

- no contexto da avaliacao de força de senhas, dado determinado modelo, quanto mais perguntas forem necessárias para adivinhar a senha, melhor. logo, é desejável que o modelo possua entropia alta. aqui, a senha faz o papel de uma mensagem de apenas um signo a ser codificado. o modelo faz o papel da codificação a ser utilizada. o domínio faz o papel do alfabeto de signos

  TI TC TM Corinda signos elemento senha simbolo hash todos os símbolos do código conjunto domínio do modelo dicionário

  teoria bayseiana pode ser usada para obter p(s), e consequentemente H(m) mais precisos... contudo, dificultaria a implementacao, por causa dos dicionarios a serem varridos pelo hashcat entao assumo p(s) -> 1/C(m)

  DIAGRAMA DE DECISOES COM MORSE ABCD E DICIONARIO

  $H(\hat{m}) = -\sum P(\tilde{s}) \log_{10} P(\tilde{s})$

  se simplificar, fica $H(\hat{m}) = -\log_{10}(1/\mathcal{C}(\hat{m}))$

  One ban or hartley is the information content of an event if the probability of that event occurring is 1/10

  FORÇA = ENTROPIA DO MODELO * FREQUENCIA DO MODELO

3.7 Considerações

A definição de string estabelece representação formal de senhas. As definições de modelo e seus derivados formalizam os conceitos que permitem a representação dos vieses psicológicos envolvidos no processo de criação das senhas por seres humanos. A definição de FDC estabelece a noção de *hash*. Por fim, a formalização do conceito de força do modelo permite quantificar quão robusta é determinada senha quando submetida a processos de quebra. Com base nestas definições, o próximo capítulo estabelece a metologia utilizada na implementação do **Corinda**.

CAPÍTULO 4

METODOLOGIA (xx/xx/2017)

4.1 Construção e configuração do sistema

4.2 Modelagem e simulação

4.3 Aplicação do processo de otimização na sintonia dos controladores

4.4 Análise de desempenho entre os controladores

4.5 Considerações

CAPÍTULO 5

RESULTADOS (xx/xx/2018)

Se qualificação, **RESULTADOS PRELIMINARES**. Se defesa final, apenas **RESULTADOS**.

Na mesma sequência da metodologia.

5.1 Construção da bancada e do modelo

5.2 Resultado de simulação

5.2.1 Otimização aplicada na sintonia dos controladores

5.3 Resultado de bancada

5.4 Comentários

Veja as Tabela 5.4 e Tabela 5.2, que servem de exemplo de como inserir tabela. Caso necessário, baixe o aplicativa *La Table* para lhe auxiliar na formatação de tabelas.

Tabela 5.1 - Parâmetros da expressão NARMAX do modelo do sistema.

Parâmetro	**Valores**
n_y	$\begin{bmatrix}1\end{bmatrix}$
n_u	$\begin{bmatrix}1\end{bmatrix}$
t_d	$\begin{bmatrix}1\end{bmatrix}$
P	$\begin{bmatrix} 0.0011 & -0.0032 \\ 3.97 \cdot 10^{-6} & 0.0872 \end{bmatrix}$
L'	$\begin{bmatrix}928.9889 & 14.3086\end{bmatrix}$
d	$\begin{bmatrix}1338.6041\end{bmatrix}$
Q	$\begin{bmatrix} 0.0011 & -0.0032 \\ 3.97 \cdot 10^{-6} & 0.0872 \end{bmatrix}$
A	$\begin{bmatrix}-248.3902 & -39.8336 & 27.1611 & 12.6842 & 28.3081\end{bmatrix}$
B	$\begin{bmatrix} -2.0417 & 4.9036 & 3.2205 & 4.0505 & 5.7625 \\ 0.9853 & 1.0024 & -0.9176 & -0.8418 & 0.4071 \end{bmatrix}$
C	$\begin{bmatrix}8.9914 & -8.2400 & -6.0503 & -4.3234 & -1.7206\end{bmatrix}$

Tabela 5.2 - Parâmetros otimizados para o controlador PID.

Parâmetro	K_P	K_I	K_D
Valores	$0,01241$	$0,000002$	$0,00641$

CAPÍTULO 6

CONCLUSÃO (xx/xx/2018)

Se qualificação, **CONCLUSÃO PARCIAL**. Se defesa final, apenas **CONCLUSÃO**.

6.1 Contribuições do Trabalho

As contribuições podem assim ser descritas:

Artigos em revista:

Artigos em congresso:

Se qualificação:

6.2 Continuação do trabalho

Se defesa final:

6.3 Sugestões para Trabalhos Futuros

REFERÊNCIAS BIBLIOGRÁFICAS

GLOSSÁRIO

Condutivímetro - é um medidor digital portátil que mensura a condutividade elétrica do solo diretamente "*in loco*".

Data Logger - é um coletor de dados também chamado de datalogger ou gravador de dados. É um dispositivo eletrônico que registra os dados ao longo do tempo ou em relação a uma localização, construído com sensores externos. São baseados em um processador digital com memórias internas para armazenamento de dados. São de uso geral para uma gama de aplicações em dispositivos de medição específicos, podem ser programáveis.

GPS - é um sistema de navegação por satélite que fornece a um aparelho receptor móvel a posição do mesmo, assim como informação horária, sob todas quaisquer condições atmosféricas, a qualquer momento e em qualquer lugar na Terra, desde que o receptor se encontre no campo de visão de quatro satélites GPS.

Neossolo Regolítico - são tipos de solos que apresentam textura arenosa e baixa capacidade de adsorção de nutrientes, quando comparado com solos argilosos, possui baixo teor de matéria orgânica e nitrogênio que diminuem, após alguns anos de uso agrícola.

Nitossolo Vermelho - são solos minerais, não-hidromórficos, apresentando cor vermelho-escura tendendo à arroxeada. São derivados do intemperismo de rochas básicas e ultrabásicas, ricas em minerais ferromagnesianos. Uma característica peculiar é que esses solos, como os Latossolos Roxos, apresentam materiais que são atraídos pelo imã. Seus teores de ferro ($Fe2O3$) são elevados (superiores a 15%).

Plintossolo Pétrico Concrecionário - são solos que ocorrem em áreas baixas e nas bordas das chapadas, constituindo geralmente por solos pobres em nutrientes. A origem de concreções ferruginosas nos solos tem sido atribuída, de forma generalizada, às condições de variações sazonais do lençol freático. Este, inicialmente elevado, propicia a redução do ferro com a sua retirada parcial do sistema, mobilização, transporte e concentração. Posteriormente, em épocas secas, a oxidação forma plintitas constituídas por mistura de argila pobre em C orgânico e rica em ferro e alumínio, segregada sob a forma de manchas vermelhas, que com a retirada do lençol freático, apresentam endurecimento constituindo concreções ferruginosas ou petroplintitas.

PVC - é feito a partir de repetidos processos de polimerização que convertem hidrocarbonetos, contidos em materiais como o petróleo, em um único composto chamado polímero. O vinil é formado basicamente por etileno e cloro. Por

uma reação química, o etileno e o cloro combinam-se formando o dicloreto de etileno, que por sua vez é transformado em um gás chamado *VCM* (Vinyl chloride monomer, em português cloreto de vinila). O passo final é a polimerização, que converte o monómero num polímero de vinil, que é o *PVC*, ou simplesmente vinil, contém, em peso, 57% de cloro (derivado do cloreto de sódio - sal de cozinha) e 43% de eteno (derivado do petróleo).

TC scan - é uma tomografia computadorizada (*TC*), originalmente apelidada tomografia axial computadorizada (*TAC*), é um exame complementar de diagnóstico por imagens tridimensionais, que consiste numa representação de uma secção ou fatia do estudo. É obtida através do processamento por computador de informação recolhida após expor o objeto estudado a uma sucessão de raios X. Seu método principal é estudar a atenuação de um feixe de raios X durante seu trajeto através de um segmento do objeto estudado; no entanto, ela se distingue da radiografia convencional em diversos elementos.