FACULDADE DE ENGENHARIA DE COMPUTAÇÃO

PROJETO FINAL I e II

PLANO DE TRABALHO

Geração de chaves criptográficas usando um algoritmo bio-inspirado

Nicholas Bastos Ferreira Fanelli Hugueney

Profº. Dr. Carlos Miguel Tobar Toledo

09/05/2014

INTRODUÇÃO

Atualmente, chaves criptográficas são usadas para cifrar dados sensíveis quando da necessidade de comunicação entre pares em um ambiente que envolve terceiros. Um dos modelos mais utilizados para promover segurança nessa comunicação é a Criptografia de Chave Pública (PKC, na sigla em inglês): uma chave pública para cifrar e uma chave privada para decifrar. No entanto, diferentes chaves criptográficas podem ser classificadas de acordo com sua robustez, característica que promove maior qualidade de segurança ao dado.

Além disso, sabe-se hoje que a segurança de dados por criptografia é refém do poder de processamento das máquinas. Em outras palavras, através da técnica de força bruta, basta dar tempo a uma máquina e qualquer cifra é passível de ser quebrada. Felizmente, no cenário atual, as soluções utilizadas comercialmente ainda são razoavelmente aceitáveis, pois prevêm que o tempo necessário para quebrar suas cifras é praticamente inviável para qualquer propósito prático de um agente mal intencionado.

CARACTERIZAÇÃO DE PROBLEMAS E OBJETIVO(S)

Pela razão explicada anteriormente para a dificuldade de se prover segurança a dados por criptografia, é possível perceber que a escolha de uma chave criptográfica robusta pode não ser uma tarefa tão simples.

Força bruta, no contexto de chaves criptográficas, é um tipo de ataque baseado em um algoritmo determinístico trivial, que consiste em definir todos os possíveis candidatos de uma solução – no caso, uma determinada chave criptográfica - e verificar se ao menos um satisfaz o problema. Este tipo de algoritmo sempre encontrará uma solução, se ela existir. Entretanto, seu custo computacional é proporcional ao número de candidatos a solução do problema.

Sabendo-se disso, uma das alternativas para tornar uma chave mais segura a esse tipo de ataque é fazê-la grande o suficiente para que o tempo necessário para encontrá-la por força bruta seja impraticável. No entanto, isso não é suficiente para uma boa segurança, pois um ataque amplamente distribuído poderia ser capaz de atravessar todo o espaço de candidatos até encontrar a solução em um tempo aceitável.

Portanto, pode-se dizer que uma chave criptográfica é robusta quando tem uma grande quantidade de caracteres (usualmente entre 1024 e 4096 bits), curto prazo de validade – é trocada com alta periodicidade – e é suficientemente distante de outras chaves.

Portanto, o TCC tem como objetivo a geração de chaves criptográficas robustas.

PLANO DE AVALIAÇÃO DO TRABALHO

Para avaliar o trabalho e, consequentemente, a robustez das chaves criptográficas geradas pelo artefato de software desenvolvido, serão basicamente aplicados dois testes, conhecidos como teste de frequência (*frequency test*) e teste de lacuna (*gap test*). O primeiro é um nome genérico para um tipo de teste que pode ser aplicado de várias maneiras, mas no caso do TCC será usado o teste do Qui-quadrado de Pearson (*Chi-square test*), que é um teste estatístico aplicado em conjuntos de dados para se determinar a probabilidade de que qualquer diferença encontrada entre dois conjuntos ocorreu por acaso.

O *gap test*, por sua vez, busca quantificar a significância dos intervalos entre a recorrência de um mesmo caractere em uma dada sequência de caracteres.

A aplicação de ambos esses testes sobre um mesmo conjunto de dados consiste em um método eficaz para se determinar a aleatoriedade das chaves geradas pelo algoritmo genético.

O sucesso do trabalho, por sua vez, poderá ser determinado uma vez que os resultados desses testes sejam satisfatórios.

PROPOSTA DO ARTEFATO

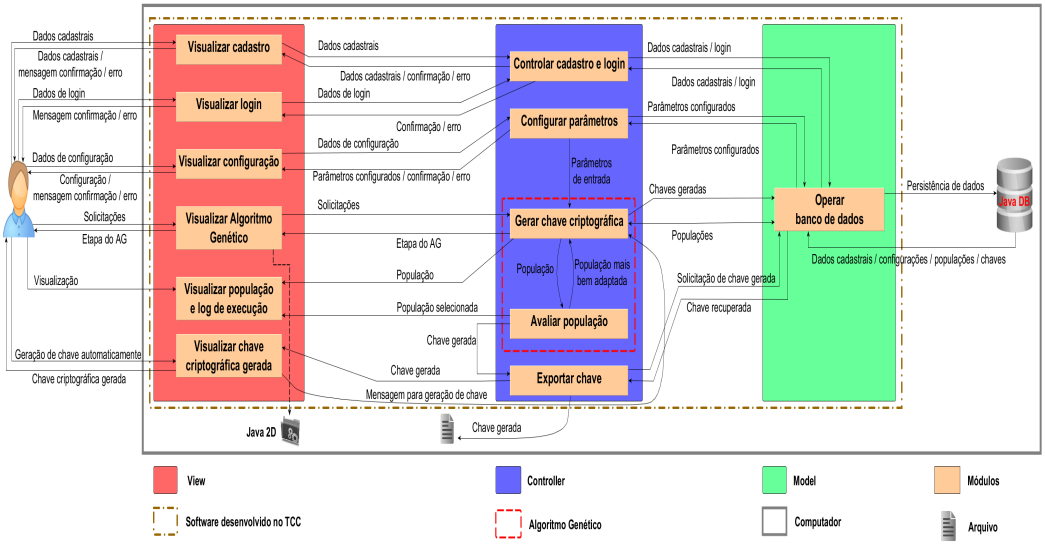
O artefato a ser desenvolvido consistirá principalmente de um Algoritmo Genético, um tipo de algoritmo de busca heurística adaptável, baseado no funcionamento da genética e seleção natural, e que pertence à classe de Algoritmos Evolutivos (AEs). Esses algoritmos são usados para encontrar soluções para problemas de otimização usando-se mecanismos baseados na evolução biológica, tais como mutação, *crossover* (cruzamento), seleção e herança.

O artefato possuirá também um módulo de configuração, através do qual o usuário será capaz de definir valores essenciais para a execução do AG, tais como o tamanho das chaves, periodicidade de geração, o número de gerações a serem geradas, entre outros, além de um módulo responsável por exibir graficamente as etapas realizadas pelo algoritmo implementado.

Outra característica fundamental do artefato será uma função de *fitness*, a qual servirá dois propósitos: primeiramente selecionar as chaves produzidas em cada geração do algoritmo que atendam ao requisito de robustez procurado; num segundo momento, avaliará as chaves geradas de modo a justificar o sucesso ou fracasso do trabalho.

O cliente do TCC, que também atua como orientador do mesmo, é doutor em Engenharia de Computação (UNICAMP) e possui ampla experiência com Sistemas Inteligentes e Software Básico. Além dele, o aluno autor do trabalho conta também com um coorientador, bacharel em Análise de Sistemas e Tecnologia da Informação (FATec – Americana) e mestrando em Engenharia Elétrica (UNICAMP), com bons conhecimentos em Inteligência Artificial.

Diagrama de arquitetura



TRABALHOS RELACIONADOS

Na tabela Comparação de aspectos entre o TCC e outros trabalhos publicados são usadas siglas para identificar os sistemas desenvolvidos nos trabalhos em questão, da seguinte forma:

- O software desenvolvido em MISHRA, S.; BALI, S (2013) é denominado “Sistema 1”.
- O software desenvolvido em JHAJHARIA, S.; MISHRA, S.; BALI, S (2013) é denominado “Sistema 2”.
- O software desenvolvido em GOYAT, S. (2012) é denominado “Sistema 3”.

O TCC o qual este plano descreve, por sua vez, é denominado “GCCABI”.

Comparação entre aspectos aspectos do TCC e outros trabalhos publicados				
	Algoritmos utilizados	**Parâmetros do algoritmo**	**Tamanho de chave**	**Método de avaliação**
GCCABI	AG	Configuráveis	Configurável	Coeficiente de autocorrelação, testes de frequência e lacuna
Sistema 1	AG	Fixos	192 bits	Coeficiente de autocorrelação, testes de frequência e lacuna
Sistema 2	AG + ANN	Fixos	192 bits	Coeficiente de autocorrelação, testes de frequência e lacuna
Sistema 3	AG + Vernam Cipher	Fixos	Não informado	Coeficiente de autocorrelação, testes de frequência e lacuna

MÉTODO DE DESENVOLVIMENTO

O método escolhido para o desenvolvimento do projeto é o Scrum solo, uma adaptação do Scrum (Schwabber, 2013) para uso individual. Essa escolha se deu pelo fato de que esse método visa entregas rápidas e frequentes de módulos funcionais do produto final e colaboração contínua com o cliente, características as quais combinam bem com o escopo do trabalho, uma vez que a interação desenvolvedor-orientador é fundamental para o bom progresso e sucesso final do projeto.

A principal atividade deste método é a *sprint*, que neste projeto será de três semanas. Vinculadas a ela estão alguns *time boxes* de reunião, sendo uma delas a *daily meeting*. No Scrum convencional, essa é composta por todo o time de desenvolvimento. Mas como o time de desenvolvimento deste trabalho é composto por apenas uma pessoa, o aluno, a *daily meeting* será substituída por uma rápida auto avaliação a ser realizada ao início de cada dia de trabalho, com as mesmas questões das reuniões diárias definidas para o Scrum convencional: “O quê fiz até o momento?”, “O quê farei hoje?” e “O quê está me impedindo de progredir?”.

Além disso, para auxiliar o gerencimento das atividades de cada *sprint* será usada a ferramenta Trello.

CRONOGRAMA

Identificação da Atividade	Descrição	Duração	
		Início	Fim
A1	Gerenciar o TCC	17/02/14	27/10/14
A2	Preparar ambiente de desenvolvimento	27/03/14	27/04/14
A3	Definir *product backlog*	10/04/14	16/04/14
A4	Executar *sprint* 1	28/04/14	18/05/14
A5	Executar *sprint* 2	19/05/14	08/06/14
A6	Executar *sprint* 3	09/06/14	29/06/14
A7	Executar *sprint* 4	15/07/14	04/08/14
A8	Executar *sprint* 5	05/08/14	25/08/14
A9	Executar *sprint* 6	26/08/14	15/09/14
A10	Executar *sprint* 7	16/09/14	06/10/14
A11	Executar *sprint* 8	07/10/14	26/10/14
A12	Preparar defesa do TCC	27/10/14	16/11/14

DISTRIBUIÇÃO DE ATIVIDADES

Identificação da Atividade	Primeiro Semestre Mês/Semana																				
	Fev				Mar					Abr				Mai					Jun			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A1			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
A2								X	X	X	X	X	X									
A3											X	X	X									
A4													X	X	X							
A5																X	X	X				
A6																			X	X	X	

Identificação da Atividade	Segundo Semestre Mês/Semana																		
	Jul				Ago				Set				Out					Nov	
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
A1			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A7			X	X	X														
A8					X	X	X												
A9								X	X	X									
A10											X	X	X						
A11														X	X	X			
A12																	X	X	X

RESULTADOS ESPERADOS

Identificação do Resultado	Descrição	Identificação da Atividade
R1	Plano de trabalho	A1
R2	*Scripts* de configuração para *backup*	A2
R3	*Product backlog*	A3
R4	*Burndown sprint* 1	A4
R5	Módulo de execução do AG	A4
R6	*Burndown sprint* 2	A5
R7	Módulo de configuração	A5
R8	*Burndown sprint* 3	A6
R9	Módulo de *login* e cadastro e módulo de exportação de chave	A6
R10	Documentos de projeto	A1

R11	*Burndown sprint* 4	A7
R12	Interface de visualização da operação de *crossover*	A7
R13	*Burndown sprint* 5	A8
R14	Interface de visualização da operação de mutação	A8
R15	*Burndown sprint* 6	A9
R16	Interface de visualização da operação de seleção	A9
R17	*Burndown sprint* 7	A10
R18	Interface funcional de interação com o AG	A10
R19	*Burndown sprint* 8	A11
R20	Interfaces de visualização de populações, chaves e *log*	A11
R21	Monografia do trabalho	A1
R22	Mídias com resultados	A1

RECURSOS MATERIAIS

- Computador *desktop*
  - Máquina principal para o desenvolvimento do artefato computacional. Será utilizada em casa.
- Computador *laptop*
  - Máquina secundária para o desenvolvimento do artefato computacional, a ser utilizada nos ambientes da universidade, trabalho e possíveis viagens, de modo a garantir a mobilidade do desenvolvimento e apresentações (parciais e final) do projeto;
- Programa de criação de diagramas (yEd 3.12.2)
  - Artefato de software *third party* auxiliar, utilizado para a criação de diagramas essencias ao desenvolvimento e documentação do projeto;
- Ambiente de controle de versão (Git 1.9.0)
  - Artefato de software *third party* essencial para o controle de versões do projeto. Terá uso contínuo ao longo do desenvolvimento;
- Pen drive
  - Artefato de hardware que será utilizado para a eventual transferência de arquivos entre os computadores de desenvolvimento e para o computador da universidade, quando necessário, a partir de onde serão realizadas as apresentações semanais.

UTILIZAÇÃO DOS RECURSOS MATERIAIS

- O computador *desktop* será a principal máquina de desenvolvimento, logo deverá ser utilizado em todos os blocos “Desenvolvimento TCC” alocados na grade de organização semanal (documento anexo);
- O computador *laptop* será utilizado quando o aluno não puder utilizar o computador *desktop*, seja por problema de funcionamento neste ou por não poder estar na sua residência, onde está o equipamento. Além disso, o computador *laptop* também será utilizado para realizar as apresentações semanais e eventualmente a defesa final do trabalho. Este computador deverá ser adquirido utilizando-se de recursos financeiros familiares, até, no máximo, final de Março de 2014;
- O repositório Git será usado nos momentos de desenvolvimento de código quando houver a necessidade de se fazer ou atualizar uma versão do trabalho.

GRAU DE DIFICULDADE – ASPECTOS DE INOVAÇÃO E APRIMORAMENTO

Não é esperado que o Trabalho de Conclusão de Curso ao qual esse documento se refere apresente algum aspecto de inovação, pois as pesquisas nessa área de estudo que levam a eles costumam ser mais

aprofundadas e demandam mais tempo. Por outro lado, existem algumas dificuldades inerentes ao projeto que, uma vez superadas, causarão aprimoramento técnico do aluno.

A primeira dificuldade identificada é o uso de Algoritmos Genéticos (GA, na sigla em inglês), pelo fato do aluno não ter experiências passadas com esse tipo de técnica, além de não ter fortes conhecimentos na área de Inteligência Artificial.

Além disso, o conceito de chaves criptográficas também não é ampla e profundamente dominado pelo aluno e, portanto, implica uma dificuldade que deverá ser rapidamente superada para o bom andamento do TCC. O trabalho de pesquisa e levantamento bibliográfico, aliado aos encontros com o professor orientador do projeto, no entanto, devem ser suficientes para promover o bom entendimento do conceito por parte do aluno.

Outro aspecto que representa uma dificuldade prevista é a criação do módulo de representação gráfica do algoritmo genético, pois o aluno não tem experiência com a implementação de interfaces graficamente interativas.

É esperada também uma dificuldade com a avaliação do resultado final do projeto, pelo fato de envolver uma função de *fitness*, que também é desconhecida do aluno.

Por fim, o aluno já fez uso de sistemas de controle de versão, porém nunca de forma intensa, seja em ambiente acadêmico ou de trabalho (estágio), por razão de não ter havido necessidade. O mesmo se aplica para o uso de ferramentas de *backup*. Logo, este será mais um aspecto que agregará alguma dificuldade, mas também terá por consequência o aprimoramento dos conhecimentos do aluno.

ANÁLISE DE RISCOS

Má gerência do cronograma de atividades – risco médio

- Havendo problemas com a gerência do cronograma, o aluno prevê buscar opiniões e conselhos de colegas de turma e/ou veteranos e ainda, quando necessário, promover mais encontros com o professor orientador e com o coorientador para auxílio na solução de impasses;

Impossibilidade de compra de um computador *laptop* – risco grave

- Apesar de ser um risco que será ao máximo evitado, não pode ser desconsiderado. Na sua ocorrência, os problemas decorrentes do mesmo devem ser mitigados com ajustes na grade de atividades semanais para destinação de mais horas de trabalho em casa, além de configuração de uma máquina virtual adequada em um *pen drive* para ser utilizada nos momentos de apresentação na universidade;
- Se for necessário, será pedida ajuda do professor orientador para a solicitação de instalação de *software* de virtualização na(s) máquina(s) da(s) sala(s) onde devem ocorrer as apresentações;
- Em última instância, o computador desktop de propriedade do aluno será levado à Universidade sempre que necessário para as apresentações.

OUTRAS OBSERVAÇÕES

Para evitar o problema de perda acidental de dados do projeto, uma política de *backup* será aplicada. Esta consistirá do uso da aplicação rsync (versão 3.0.9), para a cópia automática de arquivos, atrelada ao utilitário cron dos sistemas Unix-*like* para o agendamento das tarefas de cópia. Os dados serão copiados para o diretório da aplicação Dropbox (versão 2.6.31) do aluno em *snapshots* diários, semanais e mensais, que serão sobrescritos com o tempo, produzindo portanto cópias suficientes para que se possa, se necessário, ter acesso a qualquer arquivo que tenha sido perdido desde o momento inicial do desenvolvimento do trabalho.

- Rsync backup bash script:

```
#!/bin/bash

# Snapshots diários em diretórios do tipo "daily-4-Thu", "daily-5-Fri", e assim por diante.
if [[ "$1" == "daily" ]]
then
    path=daily-`date +%u-%a`
fi

# Snapshots semanais em diretórios do tipo "weekly-1", onde 1 é o dia do mês
if [[ "$1" == "weekly" ]]
then
    path=weekly-`date +%d`
fi

# Snapshots mensais em diretórios do tipo "monthly-04-Apr"
if [[ "$1" == "monthly" ]]
then
    path=monthly-`date +%m-%b`
fi


# Executa o script com o comando "go" como segundo parâmetro para executar o rsync,
# caso contrário imprime o comando que teria sido executado.
# -a, --archive : archive (resumo de -rlptgoD, que usa recursão e preserva quase tudo)
# -v, --verbose : verbosity (mais informação nos logs)
# -z, -- compress : compressão de dados
# --delete : remove os arquivos presentes no diretório de destino que não estão presentes no diretório fonte
if [[ "$2" == "go" ]]
then
        rsync -avz --delete /home/nicholas/TCC /home/nicholas/Dropbox/TCC_backups/$path
else
        echo rsync -avz --delete /home/nicholas/TCC /home/nicholas/Dropbox/TCC_backups/$path
fi
```

- Entradas no Crontab:

```
40 21 * * 1,3,4 /home/nicholas/TCC/rsync_backup_script.sh daily go
00 22 4,12,20,28 * * /home/nicholas/TCC/rsync_backup_script.sh weekly go
15 22 30 * * /home/nicholas/TCC/rsync_backup_script.sh monthly go
```

REFERÊNCIAS

GOYAT, S. Genetic Key Generation For Public Key Cryptography. *International Journal of Soft Computing and Engineering (IJSCE)*, p. 231-233, July 2012

JHAJHARIA, S.; MISHRA, S.; BALI, S. Public Key Cryptography using Neural Networks and Genetic Algorithms, *Contemporary Computing (IC3)*, p. 137-142, Aug 2013.

MISHRA, S.; BALI, S. Public Key Cryptography Using Genetic Algorithm. *International Journal of Recent Technology and Engineering (IJRTE)*, p. 150-154, May 2013.

SCHWABER, K.; SUTHERLAND, J. The Scrum Guide, p. 3-16, July 2013. Available at: <http://www.scrum.org>. Cited 14 mar. 2014.

DEFINIÇÕES E ABREVIATURAS

AG – algoritmo genético.

ANN – *Artificial neural network.*

Artefato Computacional – sistema de *software* ou de *hardware*, ou ainda uma combinação dos dois, que será desenvolvido com vistas à solução de um ou mais problemas identificados em um ambiente de interesse.

GA – *genetic algorithm*.

GCCABI – Geração de chaves criptográficas usando um algoritmo bio-inspirado. Sigla para o título do TCC.

PKC – Public Key Cryptography (Criptografia de Chave Pública): modelo assimétrico de troca de chaves (pública e privada) para a cifra de dados.

PO – *product owner*.

Relatório de Atividades – conjunto de lançamentos de eventos que ocorrem no decorrer do TCC, sempre que ocorrer: término previsto, atraso, antecipação ou cancelamento, considerando o início e o fim de uma atividade. Um lançamento é constituído: da identificação da atividade, sua descrição, sua data de início e sua data de fim, conforme proposto no Cronograma. Segue o status (término conforme cronograma, atraso, antecipação ou cancelamento). Caso o término não seja o esperado, devem ser incluídos: justificativa (o porquê do evento); encaminhamento (alteração do cronograma – pode ser apenas a proposta de uma nova data de fim, por conta de um atraso, ou o cancelamento da atividade); e consequência (análise e alteração das atividades ainda não encerradas por conta do encaminhamento decidido). Esses lançamentos serão úteis para a escrita da monografia.

Sistema 1 – software desenvolvido em MISHRA, S.; BALI, S. (2013).

Sistema 2 – software desenvolvido em JHAJHARIA, S.; MISHRA, S.; BALI, S. (2013).

Sistema 3 – software desenvolvido em GOYAT, S. (2012).