Anais do XX Encontro de Iniciação Científica – ISSN 1982-0178
Anais do V Encontro de Iniciação em Desenvolvimento Tecnológico e Inovação – ISSN 2237-0420
22 e 23 de setembro de 2015

Geração de Chave Pública usando Algoritmo Genético

Lucas Carvalho Roncoroni
FEComp
CEATEC
lucas.cr2@puccamp.edu.br

Carlos Miguel Tobar Toledo
Grupo de Sistemas Inteligentes
CEATEC
tobar@puc-campinas.edu.br

Resumo: *O trabalho de IC descrito, trata do uso de um Algoritmo Genético (AG) para a geração de chaves criptográficas no modelo Public Key Criptosystem (PKC). Como resultados são apresentados os requisitos levantados para desenvolver o AG, além de uma análise sobre o uso de AG para geração de chaves criptográficas.*

Palavras-chave: *Criptografia, Algoritmo Genético, Chave Pública.*

Área do Conhecimento: *Ciências Exatas e da Terra, Ciência da Computação.*

1. Introdução

Algoritmos Genéticos (AG) são "algoritmos de busca, baseados no mecanismo de seleção natural e genética natural" [1] [2]. Além disso, pertencem à classe de algoritmos evolucionários (EA), sendo usados para problemas de otimização [1]. Para resolver estes problemas, faz uso de mecanismos baseados na evolução biológica, através de mutação, *crossover*, seleção e herança [1].

A importância da criptografia pode ser atestada pelo fato de ser usada nos lugares onde ocorra armazenamento ou transmissão de dados que sejam cruciais [3], sendo ela então imperativa para a segurança no envio de dados em redes públicas e abertas.

No trabalho relatado, o método de criptografia considerado é o assimétrico de Ron Rivest, Adi Shamir, Leonard Adleman (RSA), os quais são os autores do mesmo. Tal método conta com uma chave pública para criptografar e uma chave privada para descriptografar [3].

Segundo Fanelli [2] e Goyat [4], o fato de encontrar uma chave adequada, que seja útil, é um problema de seleção de várias chaves, com base em uma classificação. Este fato faz com que um AG seja um bom candidato para encontrar boas chaves.

Entre as características de uma boa chave, considerando a dificuldade de ser descoberta, encontra-se seu tamanho, quanto maior melhor; sua constituição, se única melhor; sua diferença em relação a outras chaves utilizáveis no mesmo contexto, quanto mais diferente das outras chaves melhor.

Neste documento é apresentado o conceito de AG. Em seguida abordam-se o uso de AG em criptografia e o método RSA. Os resultados obtidos no trabalho de IC descrito para a montagem de um AG são também apresentados, seguidos de sua análise e de uma conclusão.

2. Algoritmo Genético

A população de um AG é um conjunto de cromossomos ou indivíduos. Um fato, que difere o AG de outros algoritmos de busca, é que a busca é distribuída. Ao invés de explorar apenas um caminho para a solução, o AG explora vários. Pode existir também uma ou mais condições de parada para o algoritmo. Além do limite de rodadas (cada uma para a geração de uma nova população), a busca pode parar quando se encontra um cromossomo com um valor específico estabelecido por uma função de *fitness,* que indica o grau de aptidão do cromossomo (os mais aptos permanecem na população e os não aptos são desconsiderados).

O número de rodadas determina quantas vezes, no máximo, o algoritmo irá repetir o processo evolutivo. A cada evolução é obtida uma geração de novos indivíduos através da aplicação de operações básicas que os indivíduis sofrem.

A função de *fitness* tem um papel muito importante dentro do (AG), pois ela determina quais os melhores candidatos para evoluir, dentro do conjunto de todos os cromossomos existentes (antes e após a aplicação de algumas operações básicas – *crossover* e mutação), e também determina a qualidade de cada um dos cromossomos, ou seja, seu grau de aptidão.

Segundo Fanelli [2], más funções de *fitness* podem fazer o AG ficar preso em soluções ótimas locais e perder sua característica exploratória. Portanto, é necessário um certo cuidado com esta função. A Figura 1 representa o funcionamento de um AG.

Um AG é composto de algumas operações básicas. Existem muitas variações de como elas são realizadas, pois são abstrações de operações que modificam cromossomos. Cada cromossomo, representa uma provável solução para o problema [4].

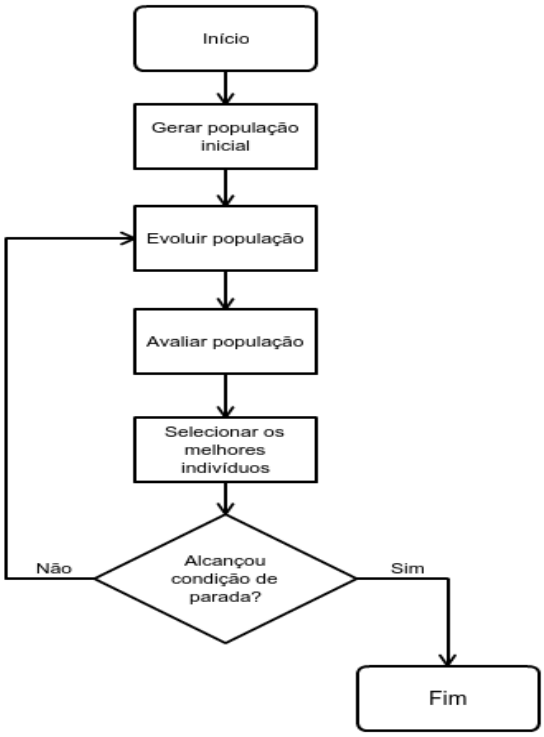

Figura 1. Funcionamento de um AG.
Fonte – o próprio autor.

Os cromossomos são compostos por um conjunto de células [4], sendo elas as menores entidades em um AG. O alfabeto binário {0,1} geralmente é utilizado para representar essas células [4].

As operações básicas de um AG são:

- Mutação – trata-se da modificação de algumas partes de um cromossomo, ou seja, a mudança de uma ou mais células. A Figura 2 mostra um exemplo de mutação em um cromossomo representado por células binárias.

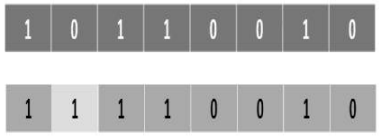

Figura 2. Mutação.

- *Crossover* – esta operação gera um novo cromossomo a partir do cruzamento de outros cromossomos, dois em geral. A Figura 3 mostra um exemplo de *crossover,* em que dois cromossomos (*before*) têm suas celulas separadas em um ponto de corte. As células à esquerda do ponto de corte do primeiro cromossomo são juntadas às celulas à direita do segundo cromossomo. O mesmo ocorrendo com as células à esquerda do segundo com as células à direita do primeiro. Tal procedimento permite gerar dois prováveis novos cromossomos (*after*).

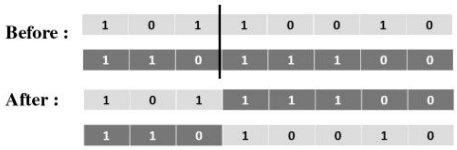

Figura 3. *Crossover*.

- Seleção – esta operação seleciona os melhores cromossomos, para fazerem parte da nova população, que continuará a evoluir nas próximas gerações do algoritmo.
- *Fitness* – esta operação na realidade é uma função usada, geralmente, na geração da população inicial e na operação de seleção. Ela atribui uma nota a um cormossomo, segundo a sua qualidade, permitindo a realização da operação de seleção.

As operações de mutação e de *crossover* devem ser aplicadas antes da seleção.

Pode ocorrer de nem todos os indivíduos de uma população serem cruzados (sofrerem a operação de *crossover*). Para esse caso, determina-se uma taxa de *crossover*.

Pode ocorrer também que nem todos os indivíduos gerados em um *crossover* sofram a operação de mutação. Para esse caso, determina-se uma taxa de mutação.

É possível existir mais de uma forma de divisão dos indivíduos em um *crossover*, com mais de um ponto de corte, por exemplo.

3. AG em Criptografia

Os AG são interessantes, por apresentarem uma característica muito peculiar, a randomicidade da evolução biológica.

Para utilizar um AG na resolução de um problema, basta abstrair indivíduos existentes no problema e representá-los como cromossomos. Além disso, determinar uma função de *fitness*. Tendo encaminhado bem essas duas questões, torna-se possível implementar um AG para buscar uma solução ótima, ou mesmo não ótima, para o problema.

Na área de criptografia, os AG, segundo Khan e Bhatia [4], e Mishra e Bali [1], geralmente são usados para decifrar chaves simples, sendo que o trabalho,

proposto em [4], usa um AG para criar uma chave *one-time pad*, chave escolhida randomicamente usada somente uma vez. Tal chave é teoricamente inquebrável [3].

Em Goyat [3], um AG também foi utilizado para gerar chaves *one-time pad*.

Em Fanelli [2], um AG foi utilizado para encontrar uma chave no modelo RSA. Nesse trabalho, por causa da falta de um método de obtenção de números primos necessários para a definição das chaves RSA, obtêm-se números primos, começando com números parcialmente primos, que têm alta probabilidade de serem primos, gerados por uma *Application User Interface* (API) da linguagem Java e refinam-se estes números, para tentar obter números com maior probabilidade de serem primos. O trabalho descrito neste documento também propõe fazer o uso de um AG para gerar uma chave no modelo RSA, a seguir abordado, e também terá que considerar o problema da geração de números primos.

4. RSA

O modelo RSA é um modelo de criptografia assimétrico, o que significa que existem duas chaves criptográficas, uma para criptografar e outra para descriptografar. Este modelo de criptografia assimétrica é conhecido como *Public Key Cryptosystem* (PKC). Pública porque não é necessário às pessoas saberem qual chave foi utilizada para cifrar uma mensagem confidencial. A única chave que deve ser mantida guardada é a chave de decifragem, também chamada de chave privada. Por causa disso, este modelo possibilita a comunicação sem nenhuma necessidade de um canal seguro para a transmissão da chave.

Segundo Menezes e colegas [5], o RSA produz chaves criptográficas difíceis de serem descobertas, pois baseia-se na dificuldade de se fatorar grandes números inteiros compostos de números primos, um problema matemático considerado difícil.

5. Resultados

Como resultados do trabalho de IC relatado, foram especificados os requisitos para um AG para a geração de boas chaves RSA. Entre eles estão:

(1) O usuário escolhe o tamanho da chave.

(2) A função de *fitness* atribui duas notas, de 0 a 1, a uma chave (cromossomo). Uma avalia a sua randomicidade e a outra a relação que as chaves têm umas com as outras (existência de algum padrão entre elas).

(3) O usuário escolhe o tamanho da população inicial.

(4) O usuário escolhe a taxa de mutação, esta é a porcentagem da população que sofre mutação.

(5) O usuário escolhe a taxa de *crossover*, esta corresponde ao número de indivíduos que são envolvidos em *crossover*.

(6) O usuário determina a condição de parada, entre parar após um número de rodadas máximo e / ou até encontrar uma chave com um valor específico de *fitness*.

(7) O usuário determina o valor de *fitness* desejável.

São vantagens dos requisitos especificados:

(1) Deixar o algoritmo configurável.

(2) O usuário poder modificar a maneira que a geração das chaves vai ocorrer, dependendo da circunstância.

(3) Por ser configurável, pode ser usado em outras pesquisas, que desejem estudar mais características sobre a geração de chaves usando um AG.

6. Análise

Por causa de suas características únicas, o conceito do AG apresenta perspectivas promissoras no campo de geração de chaves assimétricas, sendo que a maioria das referências deste artigo envolve este assunto.

O AG torna-se um método interessante, pois, segundo Fanelli [2] e Goyat [3], este replica a randomicidade da natureza, além do fato da sobrevivência dos mais aptos, ou seja, de quem apresenta maior valor de *fitness* [3]. Além disso, segundo Kan e Bhatia [4], as chaves geradas, considerando populações do AG, mostraram ser melhores do que quando definidas por um *Pseudo Randomic Number Generator* (PRNG), mesmo que trate de Vernam Cipher, um método mais simples do que aquele definido para o trabalho relatado.

Kan e Bhatia [4] relatam que as chaves geradas eram bem “aleatórias”, o que também é o objetivo do trabalho relatado: gerar chaves com pouca relação umas com as outras, o que mostra que o AG é promissor para encontrar boas chaves.

7. Conclusão

Este artigo relata o uso de um AG para a geração de chaves RSA, considerando que este caminho é promissor.

Os AG podem sim ser usados para encontrar chaves RSA (pública e privada) e, por ser uma abordagem

extremamente nova, um AG configurável que gere este tipo de chaves será muito útil para outras pesquisas, além de ter aplicação prática para a criptografia de dados, garantindo o sigilo de mensagens em redes públicas.

Agradecimento

A Deus, por tudo que tem feito por mim. À Pontifícia Universidade Católica de Campinas, pela oportunidade da Iniciação Científica. A minha família, por sempre ter estado comigo. Ao professor Tobar, pela oportunidade de realizar o trabalho.

Referências

[1] Mishra, S. Bali S. (2013), Public Key Cryptography Using Genetic Algorithm**,** *International Journal of Recent Technology and Engineering (IJRTE)*,. Disponível em: <http://www.ijrte.org/attachments/File/v2i2/B0634052213.pdf>. Acesso em: 06 ago. 2015, 23:03:27.

[2] Fanelli, N. (2014), *Geração de Chaves Criptográficas Usando Um Algoritmo Bio-Inspirado*, Monografia de Trabalho de Conclusão de Curso, Engenharia de Computação, Pontifícia Universidade Católica de Campinas, 2014.

[3] Goyat, S. (2013), Genetic Key Generation For PublicKey Cryptography, *International Journal of Recent Technology and Engineering (IJRTE)*. Disponível em: <http://www.ijsce.org/attachments/File/v2i3/C0814062312.pdf>. Acesso em: 06 ago. 2015, 23:05:32.

[4] Khan, F.;Bhatia, S. (2012), A Novel Approach to GeneticAlgorithm Based Cryptography, *International Journal of Research in Computer Science*, 2012. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.391.324&rep=rep1&type=pdf>. Acesso em: 06 ago. 2015, 23:05:32.

[5] Menezes, A.; VanOorschot, P.; Vanstone, S. (1997), *Handbook of Applied Cryptography*. Boca Raton: CRC Press.

.