Redução de Chaves Públicas com Otimização por Colônia de Formigas em Criptografia Homomórfica

Joffre Gavinho Filho[1], Jonice de Oliveira[1], Claudio Micelli[1], Gabriel Pereira da Siva[2]

***Abstract*— The excessive size of public keys generated by the cryptographic algorithms is one of the major problems for the efficient and effective operation of Fully Homomorphic Encryption (FHE). Post-quantum technique that relies on manipulating data in its raw form, this is, the encrypted data are processed and handled without being deciphered. Various methods have been proposed in an attempt to reduce such keys. This article aims to use optimization by Ant Colony in the 2nd variant of Coron, with the aim of reducing public keys produced by this method Fully Homomorphic Encryption method.**

***Resumo* — O tamanho excessivo das chaves públicas produzidas pelos algoritmos criptográficos constitui um dos grandes problemas para a eficiente e eficaz operacionalização da Criptografia Completamente Homomórfica (CCH). Técnica Pós-Quântica que se baseia na manipulação de dados em sua forma bruta, isto é, os dados criptografados são processados e manipulados sem ser decifrados. Vários métodos têm sido propostos na tentativa da redução de tais chaves. Este artigo tem como objetivo a utilização da otimização por colônia de formigas na 2ª Variante de Coron, com a finalidade da redução das chaves públicas produzidas por este método de CCH.**

Palavras-Chave — Criptografia Completamente Homomórfica; Otimização por Colônia de Formigas; Redução de Chaves Públicas.

I. INTRODUÇÃO

A informação é um dos principais e um dos mais importantes "bens" da sociedade moderna. Os avanços na tecnologia da informação, se por um lado trouxeram importantes desenvolvimentos, com novas e amplas perspectivas para diversas aplicações, por outro, trouxeram também, uma série de desafios que ainda devem ser superados. Citamos aqui em especial, a provisão de segurança a este importantíssimo "ativo". A necessidade de se proteger a informação tem sido uma preocupação e um campo de estudo desde a antiguidade, tendo a criptografia como foco e um dos principais métodos utilizados na manutenção da sua segurança [16]. Os sistemas criptográficos [16] não são novos e, mesmo em suas versões computacionais, tem sido utilizadas há bastante tempo. No entanto, o advento de novos avanços, tais como o aumento das capacidades de armazenamento e processamento dos computadores, ameaçam a segurança das conhecidas técnicas criptográficas, determinando com isso, constantes estudos sobre o tema, a fim de não tornar obsoletos os meios de proteção criptográficos da informação [13].

Dentre os vários avanços na tecnologia da informação, destacamos o desenvolvimento de sistemas computacionais que, com base em um ferramental teórico e prático, proporcionou a caracterização da segurança de novos sistemas criptográficos baseados em física quântica, a denominada criptografia quântica [6]. Porém, tal desenvolvimento criou um verdadeiro paradoxo, pois, os mesmos sistemas computacionais quânticos representam também uma ameaça a sistemas criptográficos tradicionais. Sistemas populares como o RSA [15], bem como a criptografia baseada em curvas elípticas [6], tem sua segurança ameaçada pelos algoritmos quânticos de fatoração e de logaritmo discreto de Shor [6]. A possibilidade de se resolver, em tempo polinomial (utilizando um computador quântico), problemas nas quais muitos sistemas criptográficos atuais estão baseados, possibilitou e motivou o desenvolvimento e a criação da chamada criptografia pós-quântica: ramo da criptografia que estuda as classes de algoritmos criptográficos resistentes à criptoanálise quântica [6], dentre os quais destacamos a criptografia homomórfica.

Os sistemas homomórficos caracterizam-se por manipularem os dados processando-os em sua forma crifada, sem a necessidade de acessar a informação em texto descriptografado [4]. O esquema de encriptação homomórfico (EH) é baseado em funções aditivas e multiplicativas, contendo além dos convencionais algoritmos de encriptação e desencriptação [16], um conjunto de algorítmos, que tomam como entrada uma mensagem encriptada m e retornam uma função criptografada $f(m)$, da seguinte forma: dado um esquema criptográfico $E(m)$ e duas entradas criptografadas: m_1 e m_2, tem-se as funções da seguinte forma: função aditiva $f(m) \rightarrow E(m_1 + m_2) = E(m_1) + E(m_2)$, e a função multiplicativa $f(m) \rightarrow E(m_1 * m_2) = E(m_1) * E(m_2)$ [4]. Caso o esquema $E(m)$ processe apenas uma das funções $f(m)$, i.e., a aditiva ou a multiplicativa, o esquema é denominado de parcialmente homomórfico. Porém se ele suporta tanto funções aditivas quanto funções multiplicativas indistintamente, ele também pode avaliar qualquer circuito aritmético em dados criptografados [10] e, portanto, é definido como um esquema de Criptografia Completamente Homomórfica (CCH). Usando esse tipo de regime, qualquer circuito pode receber uma manipulação homomórfica, permitindo a construção de aplicações que podem ser executadas com as suas entradas para produzir uma saída, ambas de forma criptografada. Como exemplo, pode-se multiplicar dois números criptografados e, a menos que se possa descriptografar o resultado, não há como se descobrir o valor dos números originais individualmente [10].

Os esquemas homomórfico, foco desse estudo, baseiam-se no parâmetro de segurança λ, e que define também o tamanho, em bits de comprimento, das chaves públicas e privadas geradas. O grande problema para a operacionalização de tais métodos é que os seus elevados tempos de execução, decorrente de excessivos tamanhos dos parâmetros produzidos, especialmente os das chaves públicas, cujo

Joffre Gavinho Filho[1], Jonice Oliveira[1], Claudio Micelli[1], Gabriel Pereira da Silva[2] , [1]Programa de Pós-Graduação em Informática, [2]Departamento de Ciência da Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, Brasil, E-mails: joffreufrj@gmail.com, jonice@gmail.com, cmicelifarias@gmail.com, gabriel.silva@ufrj.br

tamanho são de complexidade na ordem de $O(\lambda^{11})$, o tornam, na prática, impeditivos, [10]. Melhorias dos processos estão sendo propostas, como é o caso de [7], bem como as duas variantes de Coron [4] e [5], com reduções de complexidade na ordem de $O(\lambda^{10})$, $O(\lambda^{7})$ e $O(\lambda^{5})$, respectivamente.

Este trabalho visa reduzir os parâmetros utilizados na 2ª variante de Coron [5], tendo como meta otimizar o desempenho em relação aos tempos de execução das técnicas de compressão, com a consequente redução dos tamanhos das chaves públicas produzidas. Para tal, aplicaremos a heurística e a meta-heurística de Otimização por Colonia de Formigas (*ACO*, do inglês *A*nt *Colony Optimization*), [8], que é baseado no comportamento de formigas reais, relacionada às suas habilidades em encontrar o caminho mais curto entre o ninho e o alimento, por meio da formação de trilhas de feromônio [8], sendo extremamente adequada ao propósito deste estudo.

Este artigo está organizado da seguinte forma: na Seção 2 são descritos os mecanismos de criptografia completamente homomórfica e os métodos de compactação de chaves públicas, bem como os trabalhos relacionados; na Seção 3, a proposta de compactação e otimização é apresentada, bem como os experimentos realizados e as análises dos resultados obtidos; e finalmente, na Seção 4 são tecidas as conclusões finais e as propostas de trabalhos futuros.

II. CONCEITOS BÁSICOS E TRABALHOS RELACIONADOS

Nesta seção são descritos os conceitos básicos, bem como os trabalhos relacionados, que sustentam os temas abordados e são necessários para o entendimento do trabalho proposto. Em especial descrevemos um breve histórico da criação e desenvolvimento das técnicas homomórficas, bem como os métodos e trabalhos propostos para a redução de chaves no processo de criptografia completamente homomórfica.

A. Criptografia Completamente Homomórfica (CCH)

A criptografia homomórfica foi apresentada em sua primeira versão em 1978 por Rivest, Adleman e Dertouzos [14], que o denominaram de homomorfismos secretos (HS) – *privacy homomorphisms.* A manipulação matemática da proposta se faz da seguinte forma: a criptografia da mensagem m é dada por: $E(m) = m^e \bmod p$; onde: $E(m)$ é o esquema criptográfico assimétrico RSA [15]; m a mensagem a ser criptografada; e o elemento calculado pela função totiente de Euler [15]; e, p a chave pública do algoritmo criptográfico. A manipulação homomórfica segue a seguinte forma: dados duas mensagens m_1 e m_2; define-se: $E(m_1)*E(m_2)=m_1^e*m_2^e \bmod p \rightarrow E(m_1)*E(m_2)=(m_1*m_2)^e \bmod p \rightarrow E(m_1) * E(m_2) = E(m_1 * m_2)$, esquema este caracterizando por um homomorfismo da multiplicação da criptografia das duas mensagens. Os autores propuseram tal manipulação matemática, como forma de proteger dados sensíveis que pudessem ser manipulados sem a necessidade de serem descriptografados. O fator limitante dessa proposta é que o esquema criptográfico (HS) é parcialmente homomórfico, pois só realiza manipulações multiplicativas. Além do mais, o RSA [15], base do RAD (HS) [14], é um criptossistema determinístico, portanto não possui segurança semântica [12]. Sendo considerado em criptografia um esquema semanticamente seguro quando dado um texto cifrado de qualquer mensagem m e o tamanho dessa mensagem, não há nenhum algoritmo probabilístico que em tempo polinomial (*Probabilistic Polynomial-Time Algorithm - PPTA*), [11], seja capaz de determinar qualquer informação da mensagem m com significado maior do que a probabilidade de uma escolha aleatória [12]. Isto é, mesmo que um atacante possua uma mensagem cifrada $c(m)$ e o tamanho da mensagem $\#m$, ele não consegue descobrir nenhuma informação sobre esta mensagem em um tempo polinomial [11].

Após a construção de homomorfismos secretos [14], iniciou-se, por parte da comunidade científica, uma procura por implementações práticas desta teoria para operações aditivas e multiplicativas, isto é, algoritmos capazes de executar a denominada Criptografia Completamente Homomórfica (CCH).

Apesar de várias tentativas, o problema permaneceu sem solução durante 31 anos, quando, recentemente, em 2009, Craig Gentry [10] o solucionou sugerindo a utilização de reticulados ideais na construção de um sistema de encriptação totalmente homomórfico. Infelizmente, devido à complexidade das avaliações de multiplicações e o tamanho excessivamente grande das chaves públicas geradas no processo, a proposta de Craig Gentry [10], não foi suficientemente eficaz para ser usado na prática. Porém, a sua proposta tornou-se o um marco na criptografia completamente homomórfica, sendo tomada como a referência das propostas subsequentes.

Após o primeiro grande desafio da CCH ter sido vencido, o segundo grande está sendo torná-la eficientemente prática. Enquanto, em virtude de sua complexidade, a construção original de Gentry [10] é vista como impraticável, as últimas propostas e os esforços de implementação, melhoraram efetivamente a eficiência dos algorítmos criptográficos homomórficos. Porém, os tamanhos das chaves públicas criadas, bem como a necessidade de uma estrutura robusta de armazenamento para esses métodos, tornaram-se um dos maiores problemas para a efetiva operacionalização da criptografia homomórfica. Avanços tem sido propostos, como por exemplo, a posterior implementação de [7], que fez uso de recentes desenvolvimentos nos algoritmos criptográficos e de novas técnicas algébricas, conseguindo resultados expressivos na eficiência da execução da CCH. Dijk, Gentry, Halevi e Vaikuntanathan, DGHV [7], possui, para as chaves públicas produzidas, o tamanho na ordem de complexidade de $O(\lambda^{10})$ bits. Coron [4], por sua vez, propuseram uma modificação do DHGV, com a utilização de formas quadráticas na geração de chaves, conseguindo uma redução no tamanho das chaves públicas (*#pk – public keys*), para a ordem de $O(\lambda^{7})$. Uma segunda variante de Coron, proposta em [5], reduz ainda mais o *#pk* para $O(\lambda^{5})$. Esta segunda variante é o foco desse estudo, sendo descrita em detalhes na proxima seção.

B. A 2ª Variante de Coron

O esquema criptográfico DHGV [7] é a base das variantes de Coron [4] e [5], e tem sua estrutura na forma: *E*(*KeyGen*, *Encrypt*, *Decript, Recrypt, Evaluate*), composto de cinco algoritmos, também chamados de primitivas. *Keygen* é a primitiva responsável por gerar o par de chaves públicas e privadas; a criptografia do método é executada pela primitiva *Encrypt*; *Decript* executa a decriptação do texto encriptado; *Recrypt* é utilizada para a criptografia da criptografia, isto é, a criptografia em níveis; e, *Evaluate* faz a avaliação dos circuitos que são manipulados homomórficamente.

Na primeira variante de Coron [4], chamado DGHV com chave reduzida, o autor incluiu aditivamente novos parâmetros de forma quadrática ao esquema de primitivas originais, armazenando assim apenas um pequeno conjunto de

valores que estão relacionados à chave pública (*pk - public key*) e gerando, em seguida, uma *pk* completa em tempo de execução. Por meio desta técnica, Coron demonstrou a redução do #*pk* (tamanho da chave pública) na ordem de $O(\lambda^{10})$ do DGHV para a de $O(\lambda^{7})$, [4].

O regime inicial de inteiros usados por Coron como base para o seu trabalho, bem como para a criação de sua segunda variante, é fundamentada no trabalho de Gentry [10]. Este definiu o DGHV [7] com base em um conjunto de inteiros, $x_i = p.q_i + r_i$, $0 \leq i \leq \tau$, sendo λ o parâmetro de segurança. Os seguintes parâmetros devem ser utilizados para compor o esquema de Criptografia Homomórfica Reduzida (CHR) [2], que, em seguida, deve ser reforçada para gerar o CCH de inteiros [5]:

- γ é o comprimento em bits de x_i.
- η é o comprimento em bits da chave secreta p.
- ρ é o comprimento em bits do ruído r_i.
- τ é o número da chave pública de x_i.
- ρ' é um parâmetro de ruído secundário usado para criptografar.

O esquema é definido com as seguintes restrições [4]:

- $\rho = \omega(log\lambda)$, para a proteção contra ataques de força bruta.
- $\eta \geq \rho.\Theta(\lambda log^2\lambda)$, para a execução de avaliações homomórficas.
- $\gamma = \omega(\eta^2.log\lambda)$, para a proteção a ataques do problema do MDC.
- $\tau \geq \gamma+\omega(log\lambda)$, para reduzir a abordagem do MDC Aproximado.
- $\rho' = \rho+\omega(log\lambda)$, necessário para o parâmetro de ruído secundário.

Em sua 2ª segunda variante [5], Coron conseguiu reduzir ainda mais o comprimento das chaves públicas geradas, indo de $O(\lambda^{7})$ para $O(\lambda^{5})$. A principal inovação proposta pelo autor neste novo esquema é que, ao invéis de todos os elementos-chaves da criptografia serem armazenados, ele só armazena o valor de correção em relação a um gerador de números aleatórios. Assim, os dados a serem armazenados são menores, e, havendo necessidade dos dados, estes são recuperados "*on-the-fly*" pelas primitivas *Encryipt*, *Recrypt*, *Descript* e *Expand* (nova primitiva criada por Coron para expandir os blocos de dados reduzidos). Além disso, é descrita uma técnica de troca de módulos, que permite este regime, sem usar a estrutura proposta de *boostraping* descrito por Brakerski, Gentry e Vaikuntanathan, (BGV) [2].

Fornecemos aqui uma descrição da primitiva *KeyGen*, responsavel pela geração das chaves, e consequenemente os seus tamanhos . Observe que a técnica de compressão do texto cifrado é aplicada a ambos os elementos: (*i*) os da chave pública x_i do esquema homomorfico restrito; (*ii*) bem como os elementos para a criptografia dos bits δ_i da chave secreta.

KeyGen (1^{λ}): gera randomicamente um inteiro primo p de tamanho h bits. Nesse momento é escolhido aleatoriamente um inteiro ímpar $q_0 \in [0,2^{\lambda}/p)$, sendo $x_0 = q_0 \bullet p$. É inicializado um gerador de números pseudo-aleatórios f_1 com uma semente aleatória $\mathbf{se}_1$. Então $f_1(\mathbf{se}_1)$ é usado para gerar um conjunto de inteiros $X_i \in [0,2^{\lambda}/p)$ para $1 \leq i \leq t$. Sendo que para todo $1 \leq i \leq t$ devemos calcular: $\delta_i = [X_i]_p + \xi_i \bullet p - r_{i,}$ onde: $r_i \leftarrow Z \cap (-2^p, 2^p)$ e $x_i \leftarrow Z \cap [0, 2^{\lambda-h}/p)$, com $pk^* = (\mathbf{se}_{1,} x_{0,} \delta_1,.., \delta_t)$, e a correspondência dos inteiros x_i para $1 \leq i \leq t$ são definidos como $x_i = X_i - \delta_1$.

C. Otimização por Colônia de Formigas.

Uma das linhas de pesquisa inspirada na natureza é a Inteligência de Enxame (do inglês *Swarm Intelligence*), e que tem como objetivo descrever algoritmos e técnicas de resolução de problemas inspirados no comportamento coletivo e auto-organizado dos organismos sociais [3]. Dentre as linhas de pesquisa de inteligência de enxame, destacamos a heurística de Algoritmo Genético [11], e a meta-heurística de Otimização por Colônia de Formigas (*Ant Colony Optimization* - ACO). Esta última inspirada no comportamento social que as formigas apresentam ao buscarem por fontes de alimento para seus ninhos. Formulado na década de 90 por Marco Dorigo [8], o algoritmo têm evoluído desde sua publicação [9], e relacionada às habilidades das formigas em encontrar o caminho mais curto entre o ninho e o alimento.

Com base na estimergia (em inglês *stigmergy*) – que segundo [3], é o método de comunicação em que os indivíduos de um sistema se comunicam entre si pela modificação do ambiente local – no caso proposto, o depósito, a exploração e a evaporação de feromônio – substância química depositada pelas formigas durante seu percurso – há a formação do caminho "ótimo" entre o ninho (colônia) e a fonte de alimento.

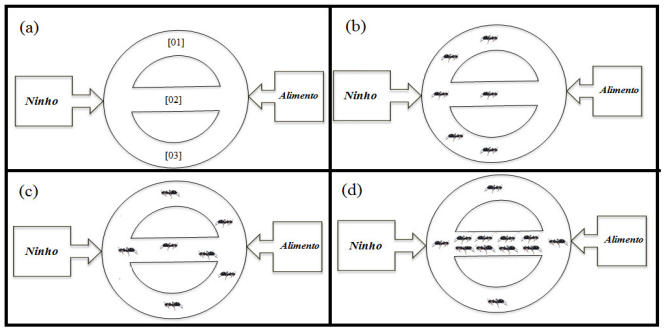

Fig. 1. Busca por alimento: (a) condição inicial; (b) inicio da varredura no terreno; (c) descoberta do alimento; e, (d) formação da trilha de feromônio.

Podemos observar na Fig. 1 uma demonstração com base nos experimentos das pontes binárias de Deneubourg [9]. Em (a), o cenário inicial, formado pelo ninho (colônia de formigas), da fonte de alimentos e de três caminhos alternativos entre o ninho e a fonte de alimento: [01], [02] e [03], com #[01] = #[03] > #[02]. Em (b), as formigas iniciam as suas buscas exploratórias de forma aleatória por alimentos no terreno. Na letra (c), ocorre a detecção do alimento com o transporte para o ninho e início do processo de depósito de feromônio (estimergia); finalmente em (d), observamos a convergência da maior parte das formigas para o menor caminho, [02], onde há a maior concentração de feromônio; apesar da descoberta do menor caminho ninho-alimento-ninho [02], observamos que ainda há formigas procurando caminhos alternativos, evitando com isso que a busca fique estagnada em um mínimo local [9].

Este comportamento foi então mapeado em algoritmos de otimização, conhecidos como ACO, que buscam melhores soluções nas trilhas de feromônio. São realizadas também atualizações locais e globais (depósito e evaporação) de feromônio, melhorando assim a busca de resultados e alternativas por caminhos não trilhados, impedindo com isso, a formação de mínimos locais. As formigas artificiais (*artificial ants*) constroem soluções de forma probabilística utilizando duas informações: (*i*) a trilha de feromônio, que muda dinamicamente durante a execução do programa de modo a refletir a experiência já adquirida durante a busca, e; (*ii*) a informação heurística especifica do problema a ser resolvido. A correspondência entre a natureza e o ACO [9], pode ser observada na Tabela I.

TABELA I. CORRESPONDÊNCIA ENTRE A NATUREZA E O ACO.

Natureza	ACO
Possíveis caminhos entre o ninho e o alimento	Conjunto de possíveis soluções
Caminho mais curto	Solução ótima
Ação via comunicação por feromônio	Procedimento de otimização

A construção do ACO começa com a formiga artificial k, de uma posição inicial n_i, escolhendo probabilisticamente a posição vizinha f_j, por meio de uma aresta A_i, Fig. 2.

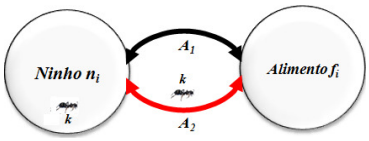

Fig. 2. Ninho n_i, fonte de alimento f_j e caminhos A_i para as formigas k_i.

A probabilidade da formiga k que está no ninho de escolher a aresta ij (A_i) para acessar o alimento f_j é dada pela Fórmula 01 [9]:

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{i \in N_i^k}(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}, \text{ se } j \in N_i^k; \text{ senão } p_{ij}^{k} = 0 \quad (01)$$

onde: τ_{ij}^{k} é a quantidade de feromônio associada à aresta ij; α e β são parâmetros para determinar a influência do feromôneo e da informação heurística, respectivamente; N_i^k são os caminhos (arestas ij) associado à k-ésima formiga; η_{ij} é a informação heurística ou valor heurístico que é definida em função das características do problema.

A atualização do feromônio é dada por: $\tau_{ij}^{k} \leftarrow \tau_{ij}^{k} + \Delta\tau^{k}$, que ocorre no retorno da k-ésima formiga, a qual deposita uma quantidade $\Delta\tau^{k}$ de feromônio nas arestas visitadas. O valor de $\Delta\tau^{k}$ é assumido constante para todas as formigas k. Esta variação de feromônio faz com que os caminhos mais curtos sejam alcançados pela intensificação da quantidade de feromônio nos melhores trechos.

A evaporação do feromônio tende a permitir uma convergência das formigas para um subótimo, ou seja, uma solução nas proximidades do ótimo. A quantidade de feromônio necessariamente diminui, o que leva à exploração de caminhos alternativos durante o processo de busca, e a evaporação limita o nível máximo de feromônio, evitando uma estagnação da solução em um ótimo local. A evaporação respeita a equação: $\tau_{ij}^{k} \leftarrow (1-\rho)\,\tau_{ij}^{k}$, $\forall(i,j) \in A_i$; $\rho \in [0,1]$, sendo ρ o parâmetro da taxa de evaporação de feromônio do ACO.

O algoritmo ACO, basicamente se resume em três rotinas: [9]: (*i*) *construção das soluções com as formigas*; (*ii*) *atualização de feromônio*, e; (*iii*) *ações daemon*. A primeira rotina é a construção das soluções pelas formigas com base em um grafo (estrutura formada por vértices e arestas) [9]. As formigas movem-se entre os componentes do grafo (variáveis do problema), estabelecendo uma conexão entre eles. Este movimento é determinado estocasticamente e localmente pelas informações das trilhas de feromônio, e pela informação heurística η, a qual é utilizada para melhorar a eficiência do algoritmo, sendo definida em função das características do problema. O segundo procedimento está relacionado ao depósito ou à evaporação do feromônio durante a construção na busca da solução. Quanto maior a quantidade de feromônio maior a probabilidade de uma mesma conexão ou componente ser usado, reforçando sua trilha. Por outro lado, a diminuição faz com que se busquem novas regiões ainda não consideradas, podendo ser regiões próximas do ótimo. O terceiro procedimento, ações *daemon*, diz respeito a rotinas que venham a melhorar a busca em determinado local, ações de busca local, ou um conjunto de ações globais que possibilitem tomar decisões positivas. O termo *daemon*, que é originário da linguagem computacional [3], significa uma rotina ou um programa criado para realizar determinada tarefa padrão, com fins específicos, a ser executado.

III. PROPOSTA DE REDUÇÃO DE CHAVES PÚBLICAS.

Essa proposta tem como base a aplicação da ACO (*Ant Colony Optimization*), no método denominado de 2ª variante de Coron [5], especificamente, na escolha pelas formigas artificiais, por meio do ACO, do valor fixo da semente $\mathbf{se}_1$ da função pseudoaleatória $f_1(\mathbf{se}_1)$ – escolhida de modo aleatório no método original – com a finalidade da redução do tamanho das chaves públicas produzidas pelo esquema de Coron.

Em nossa proposta o esquema homomórfico de Coron [5] e os três procedimentos artificiais (rotinas) de ACO, foram implementados na ferramenta matemática Matlab/Simulink©.

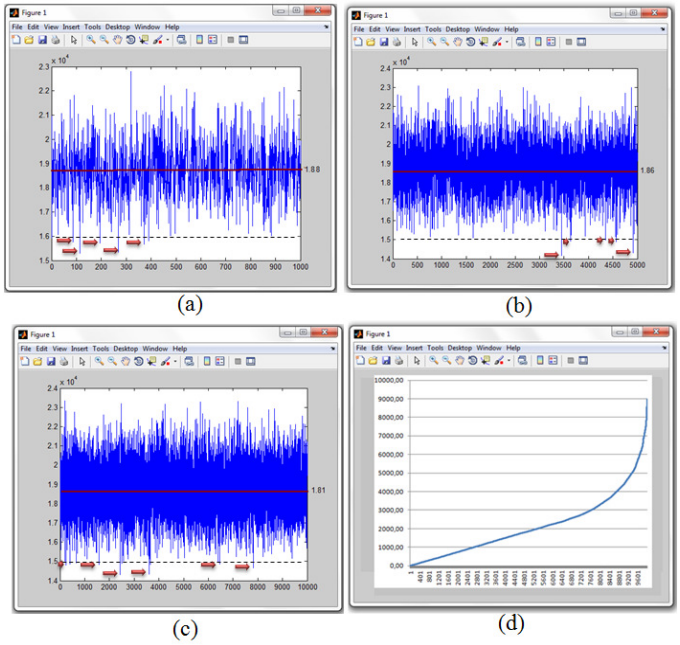

Fig. 3. Simulação: (a) 1.000; (b) 5.000; (c) 10.000 rodadas; e, (d) distribuição dos valores da semente $\mathbf{se}_1$, escolhidas aleatoriamente.

Incialmente, os parâmetros, métricas e equações definidas por Coron são simulados com o intuito de calibração e ajuste do simulador. A partir desse ponto, as simulações ocorrem de forma a reproduzir a criptografia de Coron, em sua segunda variante. Podemos observar na Fig. 3, três rodadas de simulação, com seus respectivos tamanhos de chaves públicas produzidas: com (a)1000; (b) 5000; (c) 10.000 execuções; e, (d) o gráfico da curva de distribuição dos valores da semente $\mathbf{se}_1$, que são escolhidas aleatoriamente pelos algorítimo de Coron, representando a simulçao de 10.000 execuções. Podemos também observar a média de 18 MB, Coron [5], para os tamanho das chaves, bem como a marcação das chaves produzidas com tamanhos inferiores a 16 MB em (a) e 15 MB em (b) e (c), observação esta, que nos orientou para a fixação da semente $\mathbf{se}_1$.

A. *ACO aplicada para a escolha e fixação da semente* $\mathbf{se}_1$

É proposto então a determinação e validação do tamanho fixo da semente $\mathbf{se}_1$ por meio do ACO da seguinte forma:

(*i*) *Construção das soluções com as formigas* → O esquema de solução proposto, seguindo o definido por [9] inicia-se com varredura aleatória de m formigas artificiais k, em um primeiro estágio (E_1), ($E_i \rightarrow 1 \leq i \leq 65.536$) partindo do ninho n_i, escolhendo, com mesma probabilidade → p_{ij}^{k}, entre uma das n

arestas A_i, observados na Fig. 4. Cada aresta A_i é definida com o valor do parâmetro (semente) $\mathbf{se_1}$.

(*ii*) *atualização de feromônio* → igualmente, todas as arestas A_i possuem a mesma quantidade de feromônio $\tau_{ij}=\tau_{Ai}=0{,}01$. A atualização de feromônio, é dada por: $\tau_{ij}^{k} \leftarrow \tau_{ij}^{k} + \Delta\tau^{k}$, sendo $\tau^{k}=0{,}001$ [9]. A evaporação respeita a equação $\tau_{ij}^{k} \leftarrow (1-\rho)\ \tau_{ij}^{k}$ com $\rho[0,1]$, que varia randomicamente adaptando-se as perdas "naturais" como "chuva" e "calor" (artificiais) [3]. A valoração α da taxa de depósito de feromônio é a de evaporação são diretamente proporcionais à redução final do tamanho de *pk*, Fig.4 (c) e (d).

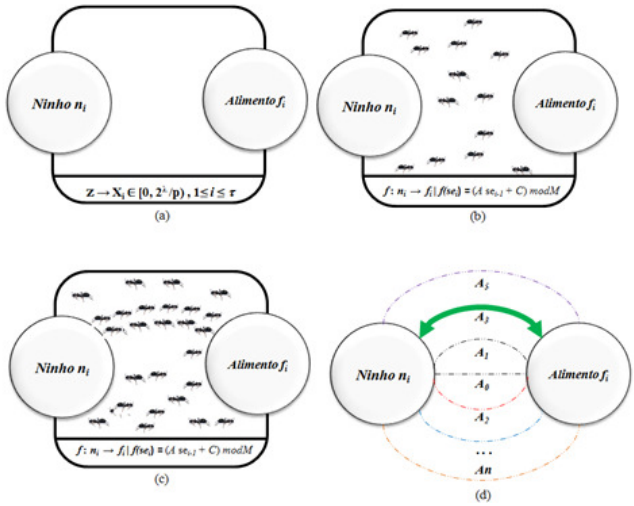

Fig. 4. Formação da Trilha de Feromônio: (a) condição inicial; (b) inicio da varredura no terreno; (c) e (d) formação da trilha de feromônio.

(*iii*) *Ações daemon* → os valores da base heurística são associados à redução final do tamanho de *pk*, e da mesma forma que o processo para a atualização do feromônio, é realizado o processamento a cada iteração do método.

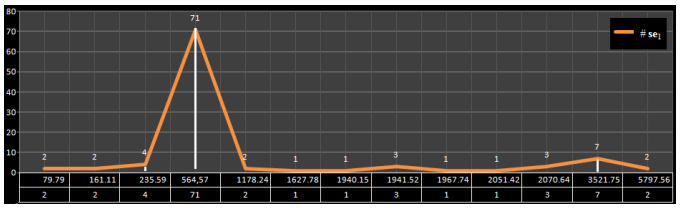

Fig. 5. Convergencia do ACO para o valor de $\mathbf{se_1}$ = 564.57.

A evolução na busca da solução ótima de "nossas formigas", definidas pelas configurações descritas em (*i*), (*ii*) e (*iii*), convergiram para o valor de $\mathbf{se_1}$ = 564.57, como pode ser observado no gráfico da Fig.5.

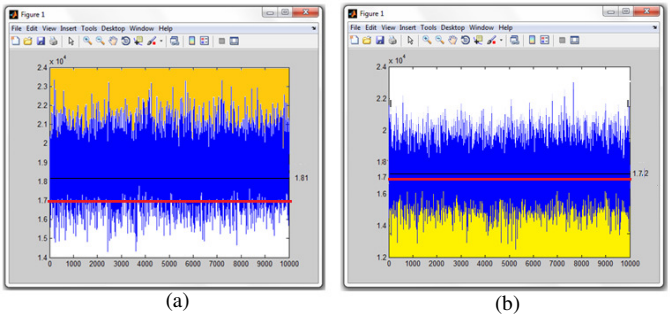

Fig. 6. Comparação das Simulações: (a) $\mathbf{se_1}$ aleatória, (b) $\mathbf{se_1}$= 564.57.

Na Fig.6 constatamos que, executando as simulações – como exemplo, em 10.000 execuções – com a semente $\mathbf{se_1}$ escolhida de forma aleatória, letra (a), com média de 18.1 MB, obtivemos 27% apenas de chaves públicas com tamanho inferiores a 17 MB. Em contrapartida ao utilizarmos, em (b), a semente $\mathbf{se_1}$ fixada em 564.57, com valores médios de 17.2 MB, os valores obtidos com chaves menores que 17 MB alcançaram o percentual de 58%, demonstrando com isso que, há a redução real do tamanho das chaves públicas em 1 MB ao fixarmos o valor da semente $\mathbf{se_1}$.

Na Tabela II encontram-se os valores comparativos dos tamanhos das chaves públicas (*#PK*), dos métodos descritos nesse trabalho, bem como a diferença na redução propostas por eles.

TABELA II. REDUÇÃO DAS CHAVES PÚBLICAS DOS MÉTODOS.

Método	***# Pk***	**Redução**	**Tempo de Execução *KeyGen***
Gentry 2009 [10]	25.000		02:20:00 [13]
DHGV 2010 [7]	2.250	22,5 GB	00:43:00 [13]
Coron 2011 [4]	0.802	21,7 GB	00:10:00 [4]
Coron 2012 [5]	0.018	784 MB	00:06:18 [11]
Essa Proposta	**0.017**	**1 MB**	00:05:01

IV CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi utilizada a Otimização por Colônia de Formigas que obteve como resultado a redução em 1 MB no tamanho das chaves públicas geradas pela proposta de Coron em sua segunda variante, por meio da definição e fixação do valor da semente $\mathbf{se_1}$ em 564.57, que no trabalho original de Coron é escolhida por meio de um gerador de números pseudo-aleatórios $f(\mathbf{se_1})$. Quanto a trabalhos futuros, procuraremos realizar os experimentos e calibrações por meio de outras heurísticas com base em inteligência de enxames, como por exemplo: o vôo dos pássaros, ou o algoritmo de busca de alimentos dos sapos.

REFERÊNCIAS

[1] Boneh, D., Halevi, S., Hamburg, M., et al. "Circular-secure encryption from decision diffie-hellman". In: Advances in Cryptology–(2008).
[2] Brakerski, Z., Gentry, C., Vaikuntanathan, V. "Fully homomorphic encryption without bootstrapping", ITCS 2012, (2012).
[3] Castro, L. N. de. Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications. Chapman & Hall/CRC, (2006).
[4] Coron, J., Mandal, A., Naccache, D., et al. "Fully homomorphic encryption over the integers with shorter public keys", AC (2011).
[5] Coron, J., Naccache, D., Tibouchi, M. Optimization of Fully homomorphic Encryption. Cryptology ePrint Archive, Report , (2012).
[6] Daniel J. Bernstein. Introduction to post-quantum cryptography. In *Post-Quantum Cryptography*. 2009.
[7] DHGV - Dijk., M. Van, Gentry, C., Halevi, S. e Vaikuntanathan, V., Fully homomorphic encryption over the integers. EUROCRYPT.
[8] Dorigo, M.; Gambardella, L. M., Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE 1997.
[9] Dorigo, M.; Stutzle, T. Ant Colony Optimization. Massachusetts Institute of Technology. Cambridge, (2004).
[10] Gentry, C. "Fully homomorphic encryption using ideal lattices". In: Proceedings of the 41st annual ACM, (2009).
[11] Gavinho, Joffre Filho; Micelli, C; Pereira, G. "A Public Key Compression Method for Fully Homomorphic Encryption using Genetic Algorithms" 19th International Conference on Information Fusion–Heildeberg-Alemanha 2016.
[12] Michael O. Rabin. Probabilistic algorithm for testing primality. Journal of Number Theory, 12(1):128 – 138, (1980).
[13] Morais, E.; Dahab, R. Encriptação homomórfica, Simpósio Brasileiro Minicursos do XII — SBSeg – Campinas -2012.
[14] NIST- National institute of standards and technology. Cyber security Framework Development Overview.NIST's Role in Implementing Executive Order 7213636, Presentation to ISPAB, (2013).
[15] RAD - R L Rivest, L Adleman, and M L Dertouzos. On data banks and privacy homomorphisms, in r. a. demillo et al. In Eds.), FSC (1978).
[16] RSA - Rivest, R. L. "Remarks on a Proposed Cryptanalytic Attack on the MIT Public-Key Cryptosystem." *Cryptologia* **2**, 62-65, 1978
[17] Stalling, Willian, Criptografia e Segurança de Redes: Princípios e Práticas 4. Ed. Prentice Hall Brasil, pag 17-36, (2007).