Uso do algoritmo Diffie-Hellman na geração de *keyfile* para o TrueCrypt

Rayner M. Pires[1], **Vaston G. Costa**[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Goiás (UFG)
campus Catalão
Catalão – GO – Brasil

{raynermp, vaston}@gmail.com

Abstract. *Transporting information often requires environments which guarantee its security against unauthorized access. Several systems and cryptographic techniques require computing resources (memory and processing capabilities) that prevent those from being used in some portable devices. Thus, one option is to use symetric encryption instead of asymmetric encryption to encipher information. Consequently, resources can be saved although the Key Exchange Problem must be dealt with. This paper presents a technique based on Diffie-Hellman Key Agreement, which can be added to the cryptographic system TrueCrypt, and does not require too much processing capability.*

Resumo. *O transporte de informações necessita, muitas vezes, de ambientes que garantam a sua segurança contra acesso não autorizado. Muitas técnicas criptográficas e sistemas que as implementam necessitam de recursos computacionais (memória e capacidade de processamento) que inviabilizam sua aplicação em dispositivos portáteis. Assim, opta-se pela utilização de criptografia simétrica ao invés da assimétrica para cifrar informações. Ganha-se na economia de recursos, mas confronta-se com o problema de troca de chaves. Neste artigo é apresentada uma técnica baseada no Acordo de Chaves de Diffie-Hellman, que pode ser adicionada ao sistema criptográfico TrueCrypt, e que não necessita de grande capacidade de processamento.*

1. Introdução

O sigilo de informações sempre foi uma peça chave no decorrer da evolução humana, remontam relatos em várias documentações que comprovam a utilização de processos para garantir a segurança da informação sob guarda de imperadores, reis e autoridades em geral [da Silva Filho 2004]. Antes a segurança da informação levava em conta as cifras no texto e o local onde essa informação era armazenada e/ou conduzida (baús, cofres e envelopes selados). Atualmente esta preocupação, salvando as devidas analogias, ainda é mantida.

A troca de informação nos dias atuais pode ser feita de várias maneiras, predominando em grandes empresas o uso de redes de computadores, internet ou intranet. E por estes meios muitas são as técnicas de segurança desenvolvidas para garantir a segurança da informação, passando por técnicas de criptografia que utilizam algoritmos baseados em conceitos matemáticos sofisticados, a protocolos de rede assinados digitalmente e criptografados. Em outros ambientes utiliza-se também os mais variados tipos de dispositivos removíveis de armazenamento de dados.

Com a popularização desses dispositivos (*pendrives*, telefones celulares, cartões de memória, etc), novas técnicas para garantir a confidencialidade da informação armazenada neles estão sendo desenvolvidas e são aplicadas de acordo com as limitações que possuem. Evidente que muito do que já foi desenvolvido para dispositivos ditos fixos pode ser adaptado para dispositivos removíveis. Contudo, a adaptação deve levar em conta as limitações de tais dispositivos como, por exemplo, ausência de unidades de processamento ou pouca quantidade de memória.

Alguns sistemas criptográficos que implementam diferentes técnicas de criptografia consagradas lidam com a segurança em dispositivos removíveis, contudo nem todos os possíveis cenários de aplicação e nem todas as técnicas existentes são implementadas.

Neste trabalho, é apresentada uma técnica de criptografia assimétrica que pode ser utilizada para gerar um segredo compartilhado e garantir o acordo de chaves seguro, para que seja mantida a confidencialidade da informação contida em dispositivos removíveis de armazenamento de dados. Tal técnica pode ser aplicada num cenário de repasse de dispositivo entre indivíduos onde a troca de chaves não pode ser feita de modo estável.

Imaginemos um possível cenário: jornalistas, pesquisadores ou policiais (doravante denomindados de Agentes), muitas vezes precisam se deslocar para locais em que não terão acesso à internet e, por questões de segurança pessoal e pela segurança da informação que estão coletando, precisam enviar relatórios periodicamente para suas centrais.

A informação, nestes casos, é armazenada em um dispositivo portátil e transportada por terceiros (doravante Mensageiros) até um local (sub-central) em que possa ser transmitida de alguma maneira para a central.

Prevendo este deslocamento, estes agentes precisam de antemão firmar uma chave que será empregada para cifrar a informação, que será, então, armazenada no dispositivo portátil. Caso contrário, a segurança de toda a informação ficaria dependendo da confiabilidade do mensageiro.

Para garantir a segurança dos dados criptografados, os agentes não carregam a chave de criptografia, e sim a chave pública de suas centrais. Como os dispositivos que carregam consigo são dispositivos com recursos computacionais limitados, é preferido o uso de criptografia simétrica ao invés da assimétrica. Tendo em mãos estes requisitos, então devem combinar a criptografia simétrica com uso de chaves públicas.

Um outro cenário possível aparece quando temos dispositivos com recursos computacionais restritos, conectados a canais inseguros e instáveis, e precisando enviar informações cifradas para algum destinatário. Nestas condições, a solução mais viável seria enviar a informação usando a menor quantidade de dados possível. Um boa solução permite o envio somente da informação criptografada, sem ser necessário o envio da chave utilizada para cifrá-la.

Nestes dois cenários podemos ver a empregabilidade de um acordo de chaves bem sucedido. É pensando neste problema que desenvolvemos nosso trabalho.

2. A segurança da informação

Informação compreende qualquer conteúdo que possa ser armazenado ou transferido de algum modo, servindo a determinado propósito e sendo de utilidade ao ser humano [da Silva Filho 2004].

A segurança da informação em um sistema computacional ou em um sistema de armazenamento, em seu sentido mais abrangente, envolve requisitos voltados à garantia de origem, uso e trânsito da informação, buscando certificar todas as etapas do seu ciclo de vida. Estes requisitos podem ser resumidos na forma dos três itens a seguir [CERT.BR 2006].

- A **confidencialidade** garante que a informação só esteja disponível para aqueles devidamente autorizados;
- A **integridade** garante que a informação não seja destruída ou corrompida e que o sistema tenha um desempenho correto;
- A **disponibilidade** garante que os serviços/recursos do sistema estejam disponíveis sempre que forem necessários.

3. Criptografia

A criptografia das informações é um mecanismo que garante a **confidencialidade** da informação em diversas camadas [PROMON 2005], através da aplicação de algoritmos de criptografia nessa informação. Esses mecanismos variam desde a criptografia das informações gravadas em dispositivos de memória (ex.: discos rígidos, *pendrives*, *smartphones*) até criptografia das informações em trânsito visando a comunicação segura. A criptografia, como forma de implementação de mecanismos de segurança em sistemas de informação ou dispositivos de armazenamento, é utilizada como prevenção ou solução para falhas em segurança na ampla maioria dos casos [Marciano 2006].

A criptografia resolve o problema de envio de informações sigilosas. Mas uma preocupação ainda recorrente é como compartilhar, com segurança, a chave utilizada para cifrar um conteúdo para a outra parte com quem se troca informação. Esse problema é conhecido como *o problema da distribuição de chaves*: como duas ou mais pessoas podem, de maneira segura, compartilhar chaves por vias inseguras? Pois, se a via fosse segura, então não precisaríamos nem mesmo cifrar o conteúdo.

Existem algumas técnicas que abordam esse problema, e uma delas é o algoritmo Diffie-Hellman.

3.1. O algoritmo Diffie-Hellman

Uma das possíveis soluções para o problema de falta de segurança na troca de chaves é utilizar o algoritmo criado por Whitfield Diffie e Martin Hellman, o algoritmo Diffie-Hellman.

O algoritmo DH, como também é chamado, não executa nenhuma técnica de cifragem de dados. Ele é usado para estabelecer um segredo compartilhado, que geralmente é usado como uma chave simétrica compartilhada [Stamp 2006].

Duas partes podem gerar o mesmo segredo desde que possuam suas próprias chaves privadas e a chave pública de sua contraparte, que está, obviamente, publicamente

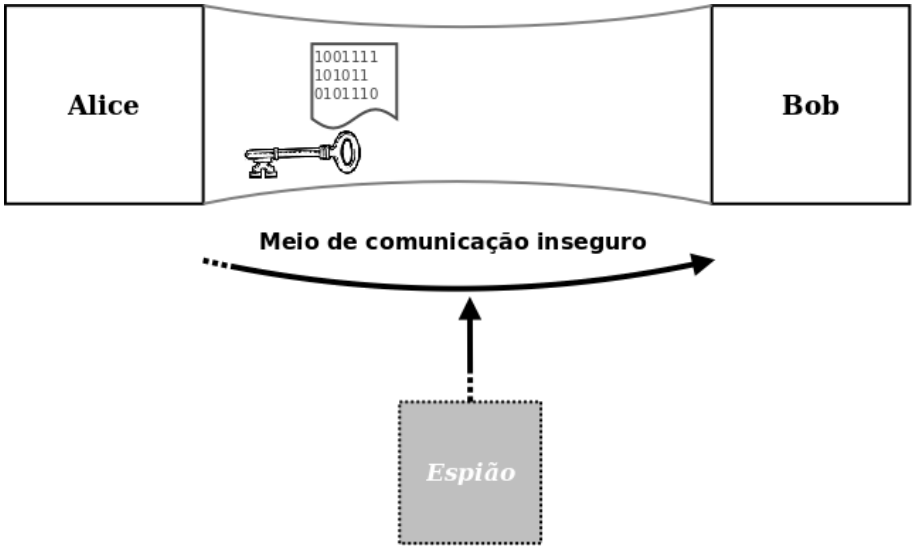

Figura 1. Alice envia a informação cifrada e a chave para Bob, em um canal de comunicação inseguro. Os dados são capturados e lidos por um espião.

disponível. Esse é o *acordo de chaves*. Se combinarmos um valor privado com o outro valor público, cada indivíduo gerará o mesmo *valor secreto* [Burnett and Paine 2002], que pode ser utilizado como a chave simétrica para cifrar conteúdos. Para obter a chave e decifrar esses conteúdos, o destinatário só precisa gerar o mesmo valor secreto novamente.

A segurança do DH, reside no problema de determinar logaritmos discretos para números grandes. O DH funciona da seguinte forma:

Sejam q um número primo e α uma raíz primitiva de q .

1 Alice gera um valor privado X_a e Bob gera um valor privado X_b ;

2 Alice envia $Y_a = \alpha^{X_a} \ mod \ q$ para Bob e Bob envia $Y_b = \alpha^{X_b} \ mod \ q$ para Alice;

3 A chave secreta $K = (Y_b)^{(X_a)} \ mod \ q$ de Alice e a chave secreta $K = (Y_a)^{(X_b)} \ mod \ q$ de Bob são iguais.

X_a e X_b são ditas as chaves privadas de Alice e de Bob, respectivamente. Enquanto, Y_a e Y_b são ditas chaves públicas do acordo.

A Figura 2 ilustra o resultado desse processo.

Uma pessoa que esteja munitorando o canal de comunicação entre Alice e Bob conseguiria obter as chaves públicas geradas, contudo não conseguiria gerar a chave secreta K, visto que para gerá-la é necessário também uma das chaves privadas X_a ou X_b.

4. O software TrueCrypt

As soluções para segurança experimentam uma fase de intensa inovação tecnológica e de crescente sofisticação motivadas, principalmente, pelo aumento constante das atividades maliciosas. O mercado de soluções de segurança é relativamente novo e apresenta uma grande fragmentação em termos de produtos disponíveis. Nele coexistem diversas empresas, oferecendo soluções para diversas áreas.

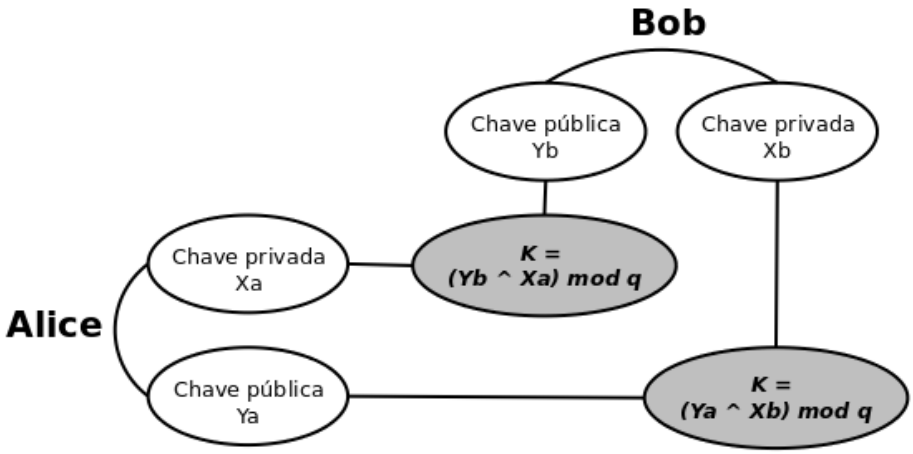

Figura 2. O resultado dos cálculos de DH é idêntico para ambas as partes.

Com relação à criptografia de dispositivos de armazenamento, sejam os móveis ou não, o mercado é amplo, e dentre os softwares livres (*freeware*) mais conceituados hoje estão o EncryptOnClick, o Rohos Mini Drive, o Kruptos e o TrueCrypt. Dos quatro softwares mencionados, o mais utilizado é o TrueCrypt. A justificativa se dá pela sua eficiência, pela facilidade de uso, pela documentação abrangente que possui e também por ser de código aberto (*open source*), possibilitando que novas versões possam ser criadas, ou que derivados dele sejam criados com novos requisitos, mais específicos. Daí a escolha do TrueCrypt como software objeto deste trabalho.

O TrueCrypt hoje, na sua versão mais atual 6.3, somente oferece algoritmos de encriptação de chave simétrica. Os algoritmos disponíveis são: AES, Serpent, Twofish. Para aumentar ainda mais o nível da complexidade da criptografia aplicada aos seus volumes, o TrueCrypt ainda utiliza de técnicas de iteração de algoritmos, ou aplicação de algoritmos em cascata, como descrito na própria documentação do software [TrueCrypt-Foundation 2009]. Dessa maneira pode-se utilizar combinações de algoritmos para criptografar os dados, acrescentando à lista de algoritmos inicial os seguintes: AES-Twofish, AES-Twofish-Serpent, Serpent-AES, Serpent-Twofish-AES, Twofish-Serpent.

Este trabalho propõe um novo recurso ao TrueCrypt, a fim de permitir a utilização do resultado gerado pelos cálculos de DH como parte da chave simétrica utilizada na cifragem dos volumes criados com esse software. Isso é útil quando o acesso ao volume criptografado precisa ser feito por duas pessoas. Para que a chave de criptografia não seja exposta (publicada) sugerimos que seja usada a técnica Diffie-Hellman para gerar uma *keyfile* que, por sua vez, é utilizada no TrueCrypt.

Mas por que se usar da técnica de criptografia assimétrica para criar uma chave ao invés de usá-la diretamente para cifrar dados em dispositivos? A resposta se dá pela complexidade computacional dos algoritmos de criptografia de chave assimétrica. Estes exigem um esforço computacional muito maior que os algoritmos de chave simétrica, então, em se tratando de dispositivos lentos como *smartphones*, *palm tops*, *tokens*, *smartcards*, etc, o ideal é que se utilize os algoritmos mais rápidos (simétricos) para cifrar

grandes quantidades de dados, e os mais lentos (assimétricos) para cifrar quantidades de dados menores, como 128 bits – o tamanho provável de uma chave simétrica. É essa mesma complexidade computacional dos algoritmos assimétricos que garante a segurança da chave gerada.

5. Aplicação da técnica no TrueCrypt

O TrueCrypt trabalha com elementos chamados *volumes*, os quais são montados pelo software e, então, interpretados pelo sistema operacional como se fossem volumes físicos propriamente ditos. Para criar esses volumes criptografados o TrueCrypt oferece um assistente, onde são selecionadas várias opções, tais como: tipo de volume, local a ser salvo, algoritmo de criptografia, tamanho do volume, tipo de sistema de arquivos e outras. Uma dessas opções pergunta se você deseja inserir alguma(s) *keyfile(s)*. É justamente neste ponto que inserimos o nosso trabalho. Uma *keyfile* é simplesmente um arquivo, que pode ser utilizado para reforçar o mecanismo de criptografia do volume. Segundo [TrueCrypt-Foundation 2009], é recomendável que se use pelo menos uma *keyfile*, e que esta seja de pelo menos 64bytes. O próprio TrueCrypt tem uma opção onde pode-se gerar uma *keyfile* aleatória. Contudo, no cenário apresentado mais acima, este recurso não seria útil, visto que o segundo usuário do volume criptografado, utilizando apenas o TrueCrypt, não poderia gerar a mesma *keyfile* novamente.

Pensando nisso é que desenvolvemos uma aplicação para gerar esse arquivo. Se precisamos de uma *keyfile* eficiente para “abrir"o volume e não podemos enviá-la junto com esse volume, então podemos utilizar o algoritmo Diffie-Hellman para gerar esse arquivo. Deste modo, qualquer um dos dois usuários que compartilharão esse volume poderá gerar a mesma *keyfile*, obedecendo os requisitos do algoritmo: estar de posse da sua própria chave privada e da chave pública de sua contraparte.

Como citado na Seção 3.1, o algoritmo DH necessita de um par de chaves de criptografia pública. Na aplicação desenvolvida é necessário que se insira o arquivo com a sua chave privada e também o arquivo com a chave pública de sua contraparte. Escolhe-se também o local onde o arquivo resultante (o resultado do processamento) deverá ser salvo. A Figura 3 mostra a tela da aplicação.

É este arquivo gerado que será utilizado como a *keyfile* do volume. Qualquer uma das partes gerará o mesmo arquivo, desde que sejam utilizadas as devidas chaves. Então, mesmo que essa *keyfile* não esteja presente, ou seja perdida, ou apagada, ela pode ser gerada novamente.

É importante enfatizar que no TrueCrypt a *keyfile* não funciona como a chave de criptografia, mas é uma parte do conjunto de todos os itens que compõem o segredo que cifra os volumes. Caso essa parte esteja ausente (ou seja incorreta), bem como outras partes (a senha, a semente e o *hash*), os volumes não podem ser abertos, mantendo, assim, o sigilo dos dados.

6. Conclusão

Neste trabalho foi apresentada a técnica de Diffie-Hellman para criação de uma chave simétrica que pode ser adicionada ao TrueCrypt.

A técnica pode ser utilizada em vários cenários, mas idealmente se propõe a sua

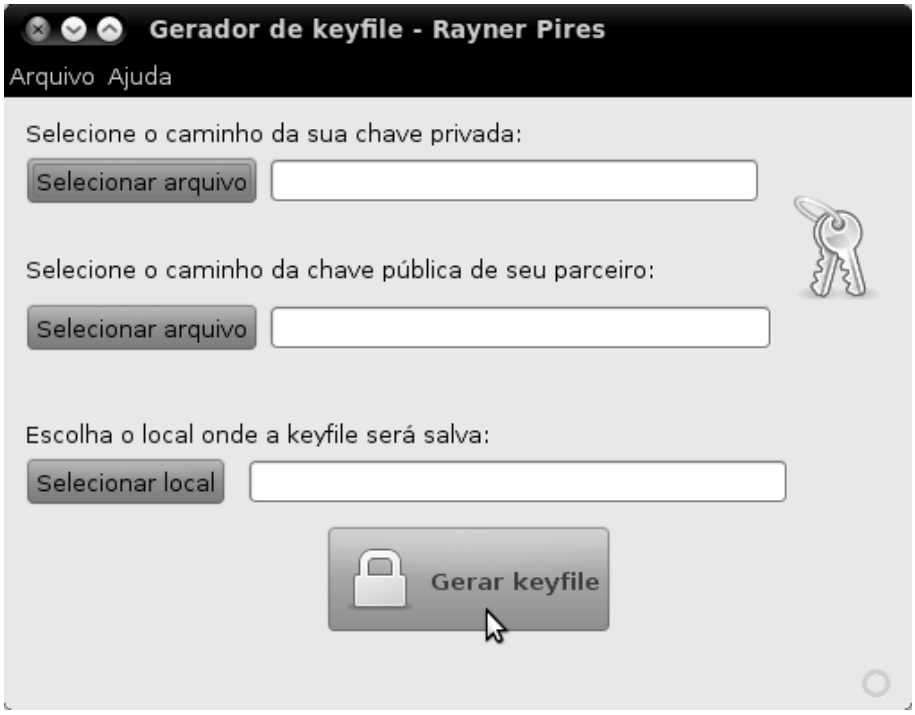

Figura 3. Tela onde as chaves assimétricas são inseridas e é gerada a *keyfile*.

aplicação quando não se pode fazer uso diretamente de criptografia assimétrica na informação que vai ser encaminhada.

Este cenário ideal envolve dispositivos que possuam pouca capacidade de processamento e armazenamento, tais como *pendrives*, *smart-cards* e telefones celulares.

Como trabalhos futuros, há a necessidade de implementar a ferramenta diretamente no TrueCrypt bem como estudar outras técnicas de geração de chaves simétricas como, por exemplo, as que são baseadas em curvas elípticas.

Como continuação deste trabalho sugerimos que essa aplicação seja acrescida ao TrueCrypt, visto que este software é muito utilizado nesta área, e isso ajudaria bastante as pessoas que o utilizam no cenário citado.

Referências

Burnett, S. and Paine, S. (2002). *Criptografia e Segurança - O Guia Oficial RSA*. Elsevier.

CERT.BR (2006). Cartilha de segurança. Acessado em 2009, disponível em http://cartilha.cert.br/.

da Silva Filho, A. M. (2004). Segurança da informação: Sobre a necessidade de proteção de sistemas de informações. Online. Disponível em: http://www.espacoacademico.com.br/042/42amsf.htm. Acessado em setembro de 2010.

Marciano, J. L. P. (2006). *Segurança da Informação - uma abordagem social.* PhD thesis, UnB.

PROMON (2005). *Segurança da informação, Um diferencial determinante na competitividade das corporações.*

Stamp, M. (2006). *Information Security - Principles and Practice*. Wiley.

TrueCrypt-Foundation (2009). Documentation. Online. Acessado em Agosto de 2009, disponível em TrueCrypt Free Open-source on-the-fly encryption: http://www.truecrypt.org/docs.