Construção de Uma Solução de Criptografia para Gestão Segura de Senhas Administrativas

Rogério de Freitas Ribeiro – rogeriofr@pgsi.uniminas.br e Gilson Marques da Silva – gilson@uniminas.br
Especialização em Segurança da Informação – UNIMINAS – União Educacional de Minas Gerais
Uberlândia – MG – Brasil

***Abstract*- This paper presents a proposal for the safe management of the administrative passwords. The current situation is contextualized showing the commonly risks and problems. The proposal presented is based on a cryptography solution. This proposal was developed through the construction of a real application, so the details of the development and production environments and some considerations about performance and overhead are discussed.**

I. Introdução

Com a crescente valorização das informações no mundo corporativo, a proteção destas é cada vez mais importante. As informações tornaram-se necessárias à continuidade de qualquer negócio, elas geram um fator competitivo que pode determinar o sucesso ou insucesso de uma companhia [1].

Existem muitos controles, muitos sistemas que visam proteger as informações: desde um sistema de criptografia, sistemas de *firewall* [2], sistemas anti-vírus e vários outros, incluindo o plano de aculturação dos usuários que manipulam as informações. No entanto, dentro desta cadeia de proteção, as senhas ainda são muito utilizadas; sejam como a única medida de proteger o acesso a sistemas e informações, ou, como medida complementar a outros elementos de proteção.

A complexidade gerada pelo elevado volume de sistemas, servidores e equipamentos de conectividade aliada à crescente demanda por segurança trazem a necessidade do uso e controle de uma infinidade de senhas.

Este artigo apresenta as dificuldades no gerenciamento de senhas administrativas. Mas também propõe uma solução, com profunda utilização de criptografia, para a questão. Como a solução já foi implementada na prática, informações sobre o ambiente e parâmetros relacionados ao desempenho da solução também são apresentados.

A seção 2 deste artigo apresenta uma visão geral da situação da maioria das empresas em relação ao uso das senhas, esta seção contextualiza o problema a ser resolvido. A seção 3 apresenta os requisitos e funcionalidades necessárias e uma solução baseada em algoritmos. Já a seção 4 expõe o suporte criptográfico utilizado na solução implementada. Na seção 5 é apresentado um estudo com medidas de desempenho e níveis de *overhead* em termos de armazenamento e processamento.

II. Principais Dificuldades no Gerenciamento de Senhas Administrativas

Para o melhor entendimento deste artigo consideram-se duas categorias de senhas. A primeira categoria abrange as senhas de usuários; cada usuário possui uma ou mais senhas, que lhe concedem acessos aos sistemas e informações. A segunda categoria contém as senhas não relacionadas aos usuários, utilizadas para os procedimentos administrativos. São senhas de aplicativos, serviços, conexões e outras; um exemplo é a senha de acesso ao nível privilegiado de um equipamento de rede, por exemplo: um roteador. Cada usuário que administra o equipamento possui senha individual de acesso ao mesmo, mas a senha de nível privilegiado pode ser única. Assim, existe a necessidade de que esta seja compartilhada entre vários administradores.

Embora a boa prática de segurança da informação determine o não compartilhamento de senhas, em casos como estes tal compartilhamento é necessário, e se bem feito, toda a rastreabilidade e segurança pode ser mantida.

As senhas administrativas têm características próprias. Na maioria dos casos são senhas muito importantes, que suportam vários serviços, que possuem alto nível de acesso e privilégio. Além disso, em alguns casos, elas podem apagar e adulterar logs de acessos.

Um grande número de senhas administrativas existe em um ambiente completo, formado por vários servidores, aplicativos, roteadores, *switches* e outros sistemas. Isso pode ser agravado ainda mais, com o aumento do número de senhas diferentes quando é adotada outra boa prática de segurança: que recomenda a utilização de senhas diferentes em cada sistema. Logo, se o ambiente é composto por centenas de equipamentos de rede, centenas de senhas diferentes deverão ser gerenciadas.

Neste cenário é humanamente impossível conhecer todas estas senhas sem o auxílio de mecanismos para o armazenamento destas senhas. É comum que os administradores destes ambientes utilizem mecanismos inseguros como anotações, planilhas eletrônicas e outras bases não protegidas para o armazenamento e consulta as senhas.

O fato de várias pessoas terem a necessidade de acesso a estas senhas cria desafios adicionais. Um exemplo é a consulta as senhas que somente deve ser permitida ao grupo de pessoas autorizadas. Isso leva a necessidade de cuidados especiais com os meios de armazenamento, transmissão e consulta as senhas.

Outro agravante é a alteração da senha de um ativo. Como vários administradores precisam conhecer a nova senha, um novo problema é gerado, pois é necessário um mecanismo que atualize a base consultada por todos os administradores. Se uma anotação impressa é utilizada, uma nova impressão e distribuição seriam necessárias. Nos casos de planilhas e outros documentos eletrônicos o mesmo problema pode ocorrer dependendo se estas bases são centralizadas ou não.

No entanto, o maior problema no armazenamento de senhas em documentos eletrônicos ou impressos ainda não foi apresentado. Trata-se da insegurança nos mecanismos de controle de acesso a estes repositórios, o que permite que as senhas sejam indevidamente consultadas ou roubadas o que pode permitir o comprometimento total do ambiente.

Administradores mais preocupados com a segurança das senhas podem sentir-se seguros ao adotarem documentos eletrônicos que requeiram senhas para a leitura ou alteração do documento. O que não é verdade, pois estes recursos são facilmente atacados com sucesso.

Outra atitude que traz a falsa sensação de segurança é a construção de portais, geralmente web, que disponibilizam interface para consulta as senhas, que via de regra, são armazenadas em SGBD (Sistema de Gerenciamento de Banco de Dados) [3] sem qualquer funcionalidade de criptografia, estando a mercê do administrador do banco de dados, do sistema operacional e até de invasões mais elaboradas.

Em sistemas como estes, pelo menos dois riscos são evidentes: o primeiro é relacionado com as informações gravadas no próprio banco de dados, que podem ser lidas por partes não autorizadas, por exemplo, o administrador do banco de dados. O segundo problema se relaciona à transmissão das informações entre o repositório de senhas e o cliente, permitindo que as senhas sejam capturadas na rede.

Uma solução para o gerenciamento seguro de senhas deve contemplar, pelo menos, a garantia da confidencialidade, integridade e autenticidade no meio de armazenamento e no meio de transmissão. A solução não pode permitir acesso irrestrito a nenhum usuário, isso inclui o administrador do banco de dados, do servidor e da rede.

Os requisitos e funcionalidades necessárias para a aplicação de gerenciamento das senhas administrativas e a solução proposta são apresentadas na próxima seção.

III. Requisitos e Solução Implementada

Os seguintes requisitos foram definidos como necessários para uma solução de gerenciamento adequado de senhas administrativas: 1 – armazenamento seguro das senhas utilizando algoritmos criptográficos; 2 – segurança durante o transporte das senhas em qualquer meio de comunicação entre o banco de dados e o cliente; 3 – total rastreabilidade e auditoria das ações realizadas no banco de dados; 4 – a confiança dos demais administradores e operadores dos sistemas de tecnologia para que se sintam confortáveis no armazenamento das senhas administravas.

Sob o aspecto de armazenamento seguro das senhas, é importante que elas estejam armazenadas em um repositório centralizado que propicie a proteção das senhas administrativas. Esta proteção deve ser implementada para que, de forma alguma, qualquer um dos administradores do ambiente (administradores de banco de dados, dos servidores ou qualquer outro equipamento que sustente diretamente a infra-estrutura de operação do sistema), tenham acesso direta ou indiretamente às senhas administrativas dos sistemas de tecnologia, armazenadas no banco de dados. Isto impacta diretamente na definição da solução criptográfica.

Para que a segurança na transmissão das senhas entre o banco de dados e o cliente seja garantida, foram implementados métodos de criptografia que garantem a segurança no transporte destas informações. Isto se faz necessário para que as senhas não sejam capturadas através de técnicas de *sniffing* de rede [2]. Outro aspecto importante é o fato de que as informações não sejam adulteradas ou afetadas com relação à integridade das mesmas.

Quanto aos dois últimos aspectos, relacionados ao processo de rastreabilidade, auditoria e a confiança dos demais administradores e operadores, é imprescindível que o repositório seja capaz de armazenar, de forma segura e íntegra, os logs com todas as ações dos usuários que tenham acesso ao repositório de senhas. Toda esta rastreabilidade confere um grau de confiança que as ações ocorridas com as senhas armazenadas no repositório centralizado possam ser acessadas durante uma auditoria.

Com base na arquitetura definida e na visão geral dos requisitos de segurança, a solução implementada suporta as seguintes funcionalidades: 1 – troca de senha no primeiro login e periódica; 2 – todos os ativos de tecnologia devem estar obrigatoriamente associados a grupos de sistemas; 3 – por padrão, os administradores não têm acesso às senhas administrativas cadastradas pelos usuários do sistema; 4 – todos os usuários pertencentes a um determinado grupo de sistemas possuem permissões de inclusão, remoção, alteração e consulta dos sistemas cadastrados por qualquer um dos demais participantes do mesmo grupo de sistemas; 5 – no cadastramento das informações referentes aos sistemas de tecnologia o sistema deve permitir a escolha de quais campos do cadastro dos sistemas de tecnologia serão criptografados;
6 – para cada um dos grupos de sistemas, a aplicação utiliza uma chave simétrica diferente para criptografar as senhas administrativas; 7 – a aplicação deve ser capaz de realizar, quando necessário, a troca de todas as chaves criptográficas utilizadas no sistema, garantindo a disponibilidade das informações já criptografadas e já armazenadas no repositório centralizado; 8 – todas as chaves criptográficas utilizadas no sistema devem ser obrigatoriamente armazenadas de forma segura utilizando-se de criptografia para o armazenamento das mesmas no repositório; 9 – toda a comunicação entre o banco de dados e o cliente deve garantir a integridade e confidencialidade; 10 – todas as ações realizadas no ambiente deverão ser registradas em log.

Todos estes requisitos serão detalhados com maior ênfase, a luz da solução adotada no decorrer deste artigo. Com base nos requisitos de segurança apresentados, a modelagem dos algoritmos criptográficos se tornou um grande desafio, em virtude de que as possíveis modelagens não pudessem atender todos os requisitos necessários, tornando inviável a construção do sistema.

A solução inicialmente foi modelada utilizando alguns dos algoritmos criptográficos conhecidos, de forma individual. São eles: algoritmos simétricos, assimétricos, de *hash* [4] e o algoritmo de criptografia baseado em senhas ou PBE *(Password-Based Encryption)* [5].

Na primeira modelagem foram considerados apenas os algoritmos de chave simétrica. Esta solução inicialmente apresentou-se inviável comparando com os requisitos necessários. O maior impeditivo para a utilização somente deste algoritmo foi a impossibilidade de resolver o problema de armazenamento da chave simétrica utilizada para criptografar as senhas administrativas.

Isto se torna um problema grave, pois os administradores do banco de dados, onde estão armazenadas as senhas dos sistemas de tecnologia, teriam acesso irrestrito a todas as senhas administrativas, incluindo as chaves simétricas utilizadas no processo de criptografia das senhas.

A segunda modelagem considerou o uso de algoritmos assimétricos, que também se mostrou inviável ao ser confrontado com os requisitos de segurança estipulados.

O algoritmo assimétrico não seria capaz de atender o requisito que especifica que qualquer usuário pertencente a um grupo de sistemas, deveria ter total acesso às informações cadastradas por qualquer outro participante desse mesmo grupo.

Outra modelagem avaliada foi a implementação do algoritmo baseado em senha ou PBE. O PBE associado ao algoritmo simétrico seria a provável solução, pois, com ele uma chave criptográfica que não tivesse a necessidade de ser armazenada poderia ser gerada. Isto seria possível, pois o PBE utilizaria a senha dos usuários em conjunto com um *salt* gerado pela aplicação. Como resultado, o sistema teria uma chave criptográfica, que poderia ser recriada quantas vezes fossem necessárias. Com esta chave seria possível criptografar a chave de sessão do algoritmo simétrico, resolvendo o problema de armazenamento da chave.

No entanto, a união somente destes dois algoritmos ainda não atenderia todos os requisitos essenciais do sistema. O requisito de troca, ao longo do tempo, de cada chave simétrica dos grupos de sistemas não poderia ser atendido, pois, o sistema necessitaria conhecer a senha de cada um dos usuários do sistema para proteger novamente as novas chaves simétricas dos grupos de sistemas.

A solução desenvolvida utiliza como repositório, um banco de dados relacional qualquer. Os algoritmos e demais complexidades na gestão das informações armazenadas no banco de dados são controladas por uma aplicação, desenvolvida em Java, que tem como função, atuar como a camada servidora da solução. Já no usuário final existe outra aplicação desenvolvida em Java que atua como a camada cliente da solução.

Existe também uma camada intermediária entre o servidor e o cliente que controla os aspectos de acesso às funcionalidades do sistema. Esta camada foi construída utilizando *webservices*. Foi adotado a linguagem Java para o desenvolvimento das camadas clientes, *webservices* e servidora, pois esta linguagem disponibiliza uma extensa biblioteca de criptografia chamada JCE (*Java Cryptography Extension*) [6].

IV. Detalhamento do Esquema de Criptografia da Solução

Para o perfeito entendimento da solução, esta seção apresenta, em detalhes, o esquema de criptografia que faz com que todos os requisitos sejam atendidos implementando todas as funcionalidades desejadas.

Para que este artigo seja facilmente entendido adotou-se uma notação específica para a descrição dos processos e elementos de criptografia, para que posteriormente a solução completa de criptografia seja apresentada.

A. Notação utilizada

O processo de criptografia simétrica é representado pelas notações $CS_K[A]$ e $DS_K[A]$ respectivamente, o processo de criptografia e decriptografia, onde (K) é a chave simétrica e (A), a informação a ser criptografada.

O processo de criptografia assimétrica é representado pelas notações $CA_{KU}[A]$ e $DA_{KR}[A]$ respectivamente, o processo de criptografia e decriptografia, onde (KU) é uma chave pública, (KR) uma chave privada e (A) a informação a ser criptografada.

Os processos de geração de hash e PBE são representados pelas respectivas notações, $(H[A||S])$ e $PBE_S[A]$, onde (S) é o *salt* utilizado e (A) a informação a ser processada.

Em muitos momentos a informação a ser criptografada é uma outra chave criptográfica. Para que isto não crie ambigüidade no entendimento é adotada a identificação através de item gráfico que representa sempre a chave criptográfica utilizada no processo de criptografia.

Todas as informações armazenadas no banco de dados são representadas neste artigo através da simbologia: **{**informação**}** e os próximos tópicos descrevem detalhadamente as funcionalidades do sistema.

B. Primeiro acesso do administrador ao sistema

O sistema já entra em produção com um administrador e senha previamente cadastrados. Estas informações serão alteradas em seu primeiro acesso.

No primeiro acesso do administrador, o nome de usuário (A0) e senha (PA0), previamente cadastrados pelo desenvolvedor, é informado para o processo de autenticação

(AU). O processo de autenticação compara o nome de usuário (A0) e senha (PA0), informados pelo usuário com as informações armazenadas no banco de dados. Toda autenticação no sistema é feita por este mesmo processo, que é representado na Fig. 1.

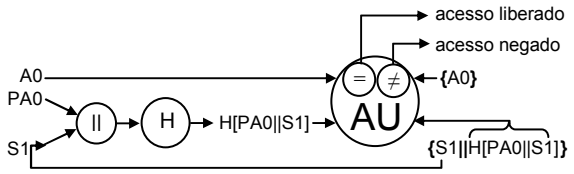


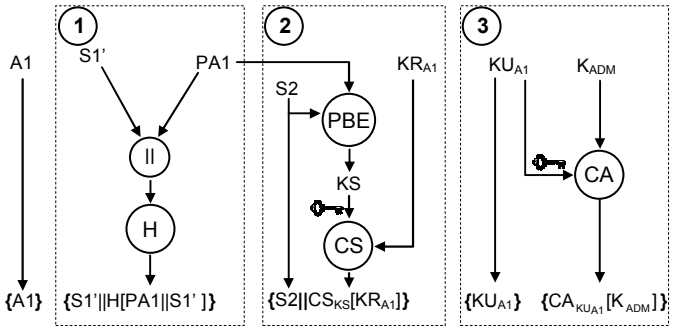
Fig. 1. Autenticação comum a todos os usuários.

Após a autenticação do administrador, a aplicação obriga a alteração do nome de usuário (A0) e senha (PA0) para A1 e PA1 informados pelo administrador. A nova senha PA1 é armazenada no banco de dados como H[PA1||S1'] utilizando um novo *salt* (S1') gerado pela aplicação que também é armazenado no banco de dados. Este *salt* é utilizado para impedir ataques de força bruta utilizando *hashs* pré-processados a partir de palavras de dicionários. Outra função do *salt* é produzir resultados diferentes mesmo que dois usuários escolham por coincidência a mesma senha. Isso impede que um administrador de banco de dados realize inferências na tentativa de descobrir usuários que utilizem à mesma senha.

Neste primeiro acesso, a aplicação gera uma chave simétrica (K_{ADM}) para o grupo administrador, e um par de chaves pública (KU_{A1}) e privada (KR_{A1}) para o administrador do sistema.

A chave simétrica é compartilhada com todos os usuários administradores do sistema, pois ela é responsável por criptografar as demais chaves simétricas utilizadas nos grupos de sistemas, conforme as funcionalidades já descritas. Em nenhum momento, o administrador conhece a composição dessas chaves simétricas, pois elas são geradas e controladas pela aplicação.

A chave simétrica (K_{ADM}) e a chave privada (KR_{A1}) devem ser protegidas. A chave pública do administrador é armazenada no banco de dados de forma não criptografada. A chave simétrica (K_{ADM}) é criptografada através de um algoritmo assimétrico (CA) utilizando a chave pública (KU_{A1}) do administrador. A chave privada (KR_{A1}) do administrador é criptografada através de um algoritmo simétrico (CS) utilizando uma chave de sessão (KS) gerada pelo algoritmo PBE. O PBE gera esta chave de sessão (KS), utilizando a nova senha (PA1) e um *salt* (S2), gerado pela aplicação. O armazenamento do par de chaves pública e privada dos usuários no banco de dados é uma implementação necessária para atender o requisito de troca das chaves simétricas dos grupos de sistemas ao longo do tempo. Isto será melhor entendido no tópico G. A Fig. 2 representa o primeiro acesso do administrador.



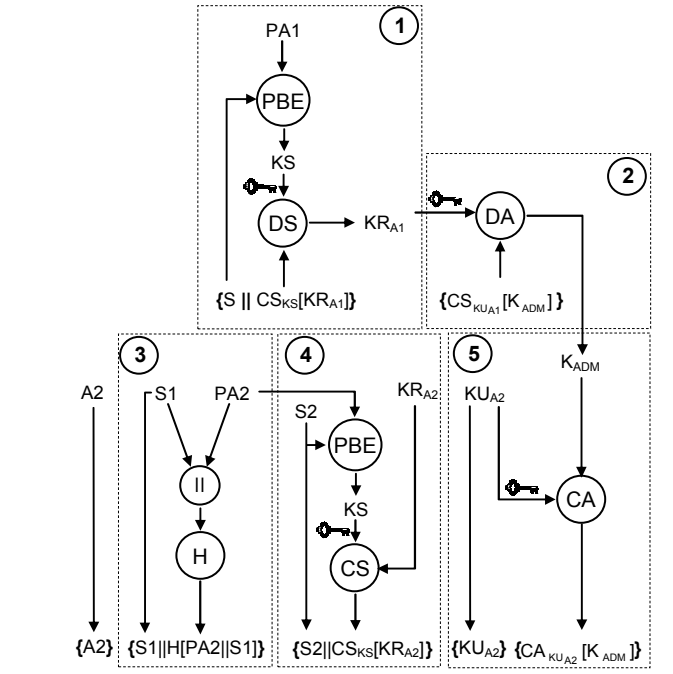
1 – armazenamento da nova senha do usuário administrador
2 – armazenamento da chave privada do usuário administrador através de criptografia simétrica
3 – armazenamento da chave K_{ADM} através de criptografia assimétrica, e armazenamento da chave pública KU_{A1} de forma não criptografada

Fig. 2. Primeiro acesso do administrador.

Os administradores usam a chave simétrica K_{ADM}, já os usuários usam as chaves simétricas dos grupos de sistemas para criptografar as senhas administrativas. Esta chave será disponibilizada no perfil do usuário quando este for alocado no referido grupo de sistemas, e isso será melhor entendido no tópico H.

C. Criação do segundo administrador (A2)

O administrador (A1), já autenticado no sistema, pode criar um novo administrador. Para isto é necessário informar um nome de usuário (A2) e uma senha inicial (PA2). A aplicação gera dois *salts* (S1 e S2) e um par de chaves, pública (KU_{A2}) e privada (KR_{A2}) conforme representado na Fig. 3.



1 – obtenção da chave privada através de decriptografia simétrica.
2 – obtenção da chave K_{ADM} através de decriptografia assimétrica
3 – armazenamento da senha do usuário A2
4 – armazenamento da chave privada do usuário A2 através de criptografia simétrica
5 – armazenamento da chave K_{ADM} através de criptografia assimétrica, e armazenamento da chave pública KU_{A2} de forma não criptografada

Fig. 3. Criação do administrador A2.

Como o novo administrador (A2) precisa da chave simétrica (K_{ADM}) do grupo de administradores, ela será copiada para o seu perfil, e será protegida utilizando sua chave pública (KU_{A2}). A Fig. 3 representa a criação do novo administrador e a criptografia das chaves.

D. Primeiro acesso do administrador A2

O novo administrador, de posse do nome de usuário (A2) e senha inicial (PA2), fornecidos pelo administrador A1, realiza o acesso ao sistema da mesma maneira que foi apresentado na Fig. 1.

Uma vez autenticado, o sistema obriga a troca da senha inicial (PA2) por uma nova senha (PA2') informada pelo novo administrador. Entretanto, a chave privada (KR_{A2}) do administrador A2 foi criptografada simetricamente utilizando como chave de sessão o resultado de um PBE baseado na senha inicial (PA2) cadastrada pelo primeiro administrador e um *salt* gerado pela aplicação. Dessa forma, é necessário que a aplicação decriptografe a chave privada deste administrador e a criptografe novamente com a nova senha (PA2') e um outro *salt* (S2') gerado pela aplicação. A Fig. 4 representa o processo de troca de senha, que será utilizado também por todos os outros usuários do sistema.

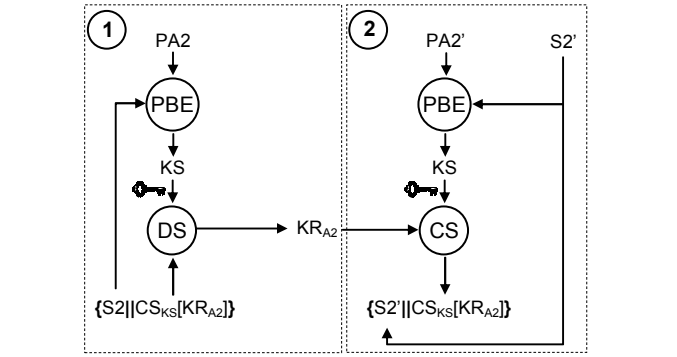


Fig. 4. Primeiro acesso do administrador A2.

E. Criação de novos usuários (U1, U2, Un) por um administrador qualquer (An)

O processo de criação de qualquer usuário no sistema segue um fluxo único. As etapas a seguir são realizadas sempre que um novo usuário for criado, exceto para usuários administradores, já apresentado no tópico C.

Um administrador qualquer, já autenticado no sistema, pode criar novos usuários. Para isto ele informa um nome de usuário (U1) e uma senha inicial (PU1). A senha PU1 em conjunto com um *salt* (S1) gerado pela aplicação será armazenada no banco de dados como resultado de uma função *hash* (H[PU1||S1]).

A aplicação gera um par de chaves sendo uma pública (KU_{U1}) e uma privada (KR_{U1}). A chave privada do usuário U1 é criptografada através de um algoritmo simétrico (CS) utilizando uma chave de sessão (KS) gerada por um algoritmo PBE. O PBE gera essa chave de sessão (KS), utilizando a senha (PU1) e um *salt* (S2), gerado pela aplicação. A Fig. 5 representa a criação do usuário U1. A mesma representação é válida para a criação de qualquer usuário.

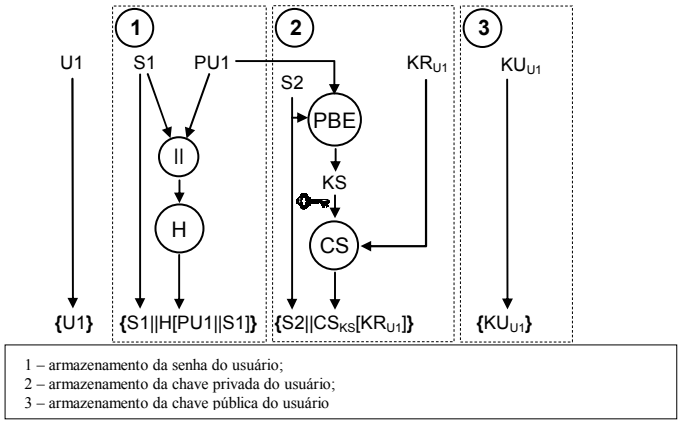


Fig. 5. Criação de um usuário.

F. Primeiro acesso do usuário U1

O novo usuário, de posse do nome de usuário (U1) e senha (PU1), fornecidos pelo administrador, realiza o acesso ao sistema da mesma maneira que foi apresentado na Fig. 1.

Uma vez autenticado, o sistema obrigará a troca de sua senha. O usuário informa uma nova senha PU1'. O processo de alteração da senha do usuário U1 é o mesmo apresentado na Fig. 4.

G. Criação de grupos de sistemas (GSn) por um administrador qualquer (An)

Qualquer administrador devidamente autenticado pode criar grupos de sistemas (GSn).

Para a criação do grupo a aplicação gera uma chave simétrica (K_{GSn}), que será utilizada pelos usuários que participarem desse grupo. Esta chave é utilizada para criptografar as senhas administrativas dos sistemas de tecnologia pertencentes a este grupo.

Todos os administradores precisam armazenar esta chave simétrica (K_{GSn}) de forma criptografada, para que em um momento posterior a mesma seja copiada para o perfil do usuário que participará deste grupo de sistema. Para isto as chaves simétricas (K_{GSn}) dos grupos de sistemas (GSn) são armazenadas de forma criptografada nos perfis dos administradores. Essas chaves simétricas são criptografadas por um algoritmo de criptografia assimétrico (CA), utilizando como chave de criptografia a chave pública (KU_{An}) dos administradores. A Fig. 6 representa o processo de criação de dois grupos de sistemas GS1 e GS2 e suas respectivas chaves simétricas K_{GS1} e K_{GS2}, bem como o processo de armazenamento dessas chaves simétricas no perfil do administrador A1. O mesmo processo é realizado para armazenar as chaves K_{GS1} e K_{GS2} para qualquer outro administrador.

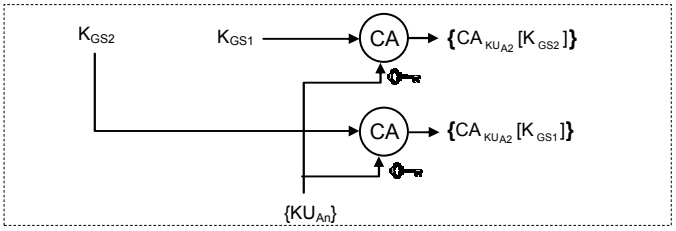


Fig. 6. Criação de grupos de sistemas.

H. Um administrador qualquer (An) concedendo permissões para um usuário a um grupo de sistemas

Para que um usuário do sistema, seja capaz de criar, consultar, alterar e apagar as senhas administrativas dos sistemas de tecnologia relacionados a um grupo, ele terá que participar deste grupo de sistemas. Para que este usuário seja capaz de realizar estas ações ele precisa ter a chave simétrica desse grupo de sistemas armazenada em seu perfil no banco de dados. O usuário deve possuir esta chave simétrica, pois é com ela que as informações são criptografadas e decriptografadas.

Qualquer um dos administradores do sistema está apto a conceder acesso para um usuário em um grupo de sistema. Isto é possível, pois, no momento da criação do grupo de sistemas, a chave simétrica do grupo foi armazenada no perfil de cada um dos administradores do sistema. No momento em que um administrador concede acesso a um usuário em um ou mais grupos de sistemas, a aplicação precisa copiar a chave simétrica destes grupos para o perfil do usuário. Para isto, a aplicação decriptografa a chave privada do administrador através de um algoritmo simétrico (CS).

Com a chave privada do administrador decriptografada, a aplicação a utiliza, para decriptografar a chave simétrica K_{GSn} do sistema que o administrador deseja incluir o usuário. Após a chave K_{GSn} ser decriptografa, a aplicação copia esta chave para o perfil do usuário, criptografando-a através de um algoritmo assimétrico (CA) utilizando como chave de criptografia a chave pública (KU_{Un}) do usuário. A Fig. 7 representa o processo de inclusão do usuário U1 no grupo de sistema GS1, bem como o armazenamento da chave simétrica K_{GS1} desse grupo em seu perfil no banco de dados.

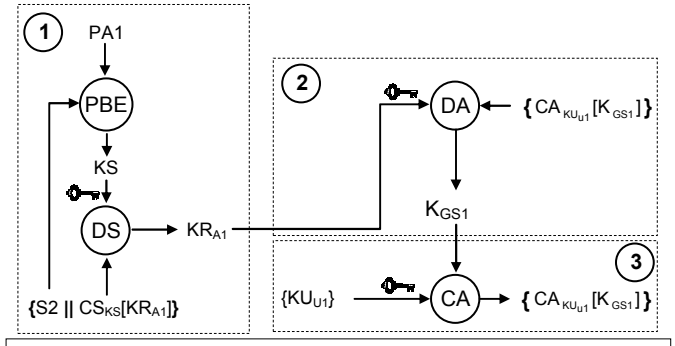


Fig. 7. Alocando um usuário a um grupo de sistemas.

I. O usuário (U1), criando um sistema de tecnologia

Um usuário do sistema devidamente autenticado e pertencendo a um ou mais grupos de sistemas, está apto a criar um sistema de tecnologia e armazenar a senha ou senhas administrativas deste sistema de tecnologia. Para a criação do sistema de tecnologia o usuário informa em qual dos grupos de sistemas que o sistema de tecnologia será armazenado. O usuário poderá armazenar este sistema somente nos grupos de sistemas que o mesmo possui acesso.

Após o usuário preencher as informações do sistema de tecnologia, como usuário e senhas administrativas, a aplicação procede com o armazenamento das senhas administrativas no banco de dados.

As senhas administravas (SA) são criptografadas utilizando um algoritmo simétrico (CS), utilizando como chave de sessão a chave simétrica do grupo K_{GSn}. Como a chave de sessão K_{GSn} está criptografada no perfil do usuário U1, a aplicação precisa decriptografá-la para utilizá-la posteriormente no algoritmo simétrico (CS). Para isso, a aplicação utiliza a chave privada (KR_{U1}) do usuário para decriptografar a chave simétrica K_{GSn} . Como a chave privada do usuário também está criptografada, a aplicação utiliza a senha do usuário e o *salt* armazenado no banco de dados para decriptografar a chave privada do usuário e completar o processo. A Fig. 8 representa o processo de inclusão de um sistema de tecnologia no grupo GS1.

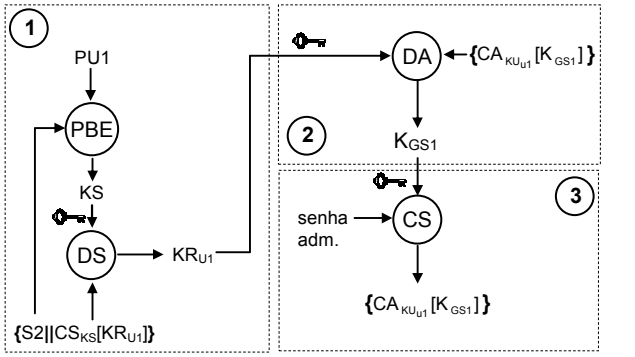


Fig. 8. Inclusão de um sistema de tecnologia

J. O usuário (U2), acessando o sistema de tecnologia criado pelo usuário (U1)

Qualquer outro usuário que faça parte deste grupo de sistemas está apto a realizar a consulta das senhas administrativas bem como toda a gerência deste sistema de tecnologia (alterar, atualizar e apagar).

O usuário U2 devidamente autenticado solicita a consulta da senha administrava (SA) do sistema de tecnologia. Para isso a aplicação precisa obter a chave simétrica do grupo de sistemas K_{GS1} que está armazenada no perfil do usuário U2. Para obter essa chave simétrica a aplicação realiza as mesmas etapas 1 e 2 representadas na Fig. 8, utilizando a senha e chave

privada do usuário U2.

A aplicação após obter a chave simétrica K_{GS1}, decriptografa as senhas administrativas (SA) do sistema de tecnologia utilizando como chave simétrica de decriptografia a chave K_{GS1}. A Fig. 9 representa a decriptografia da senha administrativa (SA) do sistema de tecnologia.

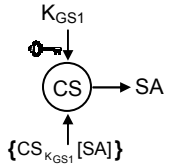


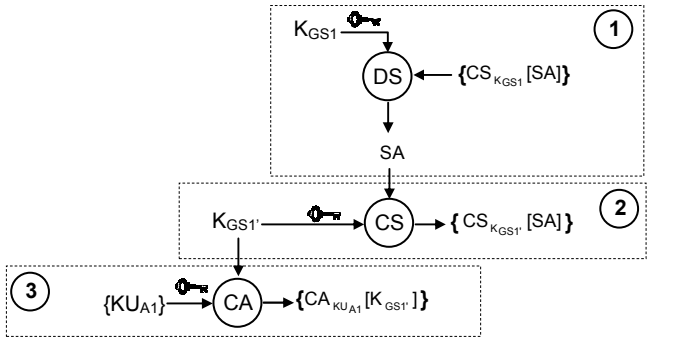
Fig. 9. Consulta da senha administrativa de um sistema de tecnologia

K. Troca da chave simétrica dos grupos

Por questões de segurança a aplicação está preparada para a troca de todas as chaves simétricas dos grupos de sistemas, mantendo possível a decriptografia das informações já armazenadas. Esta ação será realizada por qualquer um dos administradores.

Para isto a aplicação obtém a chave simétrica K_{GS1} do grupo de sistemas GS1 conforme etapas 1 e 2 da Fig. 8. Após a obtenção da chave simétrica do grupo de sistemas GS1, a aplicação decriptografa todas as senhas administrativas de todos os sistemas de tecnologia associados a este grupo de sistemas.

A aplicação após decriptografar as senhas administrativas dos sistemas de tecnologia gera uma nova chave simétrica $K_{GS1'}$. Com esta nova chave simétrica, a aplicação criptografa novamente as senhas administrativas. A aplicação também precisa armazenar a nova chave $K_{GS1'}$, no perfil de todos os administradores, e no perfil de todos os usuários do sistema que estão associados a este grupo de sistemas. A aplicação realiza estes passos para todos os demais grupos de sistemas existentes. A Fig. 10 representa o processo de troca da chave K_{GS1}.



1 – obtenção da senha administrativa
2 – armazenamento da senha administrativa utilizando como chave de criptografia a nova chave simétrica $K_{GS1'}$
3 – armazenamento da nova chave simétrica $K_{GS1'}$ no perfil do usuário utilizando como chave de criptografia a chave pública do usuário

Fig. 10. Troca da chave simétrica de um grupo.

V. DESEMPENHO E *OVERHEAD*

Uma premissa para a solução proposta é oferecer o acesso às senhas administrativas de forma simples, rápida e transparente ao usuário final. Esta solução em um primeiro momento pode induzir, ao pensamento de que toda a camada de criptografia implementada gera um *overhead* tornando o sistema lento para a sua operação no dia-a-dia, contradizendo esta premissa.

Se o sistema desenvolvido tivesse a sua performance afetada pelos processos de criptografia, a solução poderia se tornar inviável pois o tempo de acesso maior do que o tolerável na consulta das senhas administrativas poderia onerar as atividades dos profissionais que dependem das mesmas para suas atividades cotidianas.

No entanto a implementação prática do sistema demonstrou que o *overhead* ocasionado pela camada de criptografia não afeta significativamente o tempo total no acesso das senhas administrativas. Uma versão sem a camada de criptografia do mesmo sistema implementado foi utilizada como base comparativa. Quando as versões com criptografia e sem criptografia foram comparadas, identificou-se que a diferença nos tempos de cada uma das versões está na casa de milesegundos. Desta forma, os processos criptográficos são imperceptíveis para o usuário final, não inviabilizando a sua utilização no dia-a-dia.

A tabela 1, mostra os tempos, em milesegundos, gastos nas operações de armazenamento e consulta das senhas administrativas na versão do sistema sem a camada de criptografia. Foram realizadas 200 interações seqüenciais em cada uma das operações. Lembrando que nesta versão o sistema se comporta como qualquer outro sistema de gerenciamento de informações em um banco de dados relacional. Nesta situação as senhas foram gravadas de forma não protegida no banco de dados estando sujeitas a todos os problemas apresentados no início deste artigo.

TABELA 1
TEMPOS PARA CONSULTA E ARMAZENAMENTO DAS SENHAS SEM A CAMADA DE CRIPTOGRAFIA.

valores	armazenamento	consulta
Mínimo	4 ms	3 ms
Máximo	7 ms	5 ms
Média	4,85 ms	4,15 ms

Nos tempos apresentados não foi contabilizado o tempo gasto para a abertura e fechamento da conexão com o banco de dados, pois o servidor de aplicação implementa o recurso de *pool* de conexões, disponibilizando para a aplicação conexões previamente abertas. Para a coleta de todos os tempos foi utilizado um processador Xeon 3.2 GHz 2MB *cache* e 1GB RAM.

Da mesma forma, a tabela 2, mostra os tempos na versão do sistema com a camada de criptografia implementada. Deve ser observado que estes tempos não contemplam a geração das chaves envolvidos no processo pois ambas já estão armazenadas no banco de dados.

TABELA 2
TEMPOS PARA CONSULTA E ARMAZENAMENTO DAS SENHAS COM A CAMADA DE CRIPTOGRAFIA.

valores	armazenamento	consulta
mínimo	29 ms	28 ms
máximo	34 ms	36 ms
média	31,75 ms	31,25 ms

Nesta solução de criptografia é utilizado o algoritmo de chave simétrica *Triple* DES *(Data Encryption Standard)*, 3k3DES, [4] com chave de 168 bits, e o algoritmo de chave assimétrica RSA [5], com chave de 1024 bits.

A tabela 3, mostra o *overhead* identificado pela camada de criptografia no sistema implementado. Os valores foram obtidos pela subtração das médias dos tempos obtidos entre as versões com criptografia e sem criptografia.

TABELA 3
OVERHEAD COM A IMPLEMENTAÇÃO DA CAMADA DE CRIPTOGRAFIA.

	sem criptografia	com criptografia	*overhead*	% de *overhead*
armazenamento	4,85 ms	31,75 ms	26,9 ms	655%
consulta	4,15 ms	31,25 ms	27,1 ms	753%

A tabela 4, mostra a quantidade de armazenamentos e consultas das senhas administrativas baseando-se no comportamento de 10 usuários distintos na utilização do sistema. É uma tendência normal no decorrer da utilização do sistema que exista uma quantidade maior de consultas do que armazenamentos.

TABELA 4
QUANTIDADE DE ARMAZENAMENTO OU ATUALIZAÇÕES(A) E CONSULTAS(C) DE SENHAS ADMINISTRATIVAS.

Usuá-rio	1º dia		2º dia		3º dia		4º dia		5º dia		Total	
	A	C	A	C	A	C	A	C	A	C	A	C
U1	5	32	4	21	6	14	3	31	5	26	23	124
U2	7	30	6	18	5	24	5	29	4	18	27	119
U3	6	25	5	26	7	23	8	36	3	24	29	134
U4	7	15	6	19	6	19	5	29	2	16	26	98
U5	3	14	4	23	8	21	5	30	6	28	26	116
U6	8	16	6	19	4	23	7	20	5	29	30	107
U7	4	20	4	23	5	23	5	23	6	25	24	114
U8	6	16	7	17	5	21	3	19	8	19	29	92
U9	3	32	4	23	3	25	5	14	6	17	21	111
U10	5	24	6	26	7	26	3	24	5	18	26	118
Total	54	224	52	215	56	219	49	255	50	220	261	1133

Em um dia de trabalho comum de 8 horas um profissional dispõe no total de 28.800.000 milesegundos (1000ms * 60s * 60min * 8h), este pode realizar aproximadamente 5.938.144 armazenamentos, (28.800.000/4,85ms), e 6.939.759 consultas, (28.800.000/4,15ms), às senhas administrativas no sistema sem a camada de criptografia.

Considerando a mesma carga diária, um profissional pode realizar aproximadamente 907.086 armazenamentos, (28.800.000/4,84), e 921.600 consultas, (28.800.000/4,15), às senhas administrativas no sistema com a camada de criptografia.

Como o *overhead* nos tempos de armazenamento e consulta às senhas administrativas estão na ordem de milesegundos, a implementação da solução descrita neste artigo apenas seria inviável com relação aos tempos na operação do sistema, caso os usuários necessitem realizar mais de 907.086 operações de armazenamento ou atualizações e 921.600 operações de consultas diariamente, o que não é verdade quando analisado a tabela 4, que mostra o comportamento de 10 usuários distintos durante uma semana de trabalho.

VI. CONCLUSÃO E TRABALHOS FUTUROS

Como contribuição acadêmica, este artigo apresentou a viabilidade de uma especificação arquitetural de criptografia para a construção de um sistema seguro, considerando os atuais padrões de segurança, para o armazenamento e gerenciamento de senhas administrativas. Esta mesma arquitetura de criptografia proposta e implementada pode ser re-implementada em qualquer linguagem de desenvolvimento de sistemas tornando-se uma arquitetura de referência.

A utilização de um sistema para tal propósito, reduz significativamente os riscos à segurança das informações dos sistemas de tecnologia, pois todos os controles especificados e modelados nesta arquitetura de criptografia, visam mitigar os atuais riscos nos métodos de armazenamento e gerenciamento de senhas apresentados no início deste artigo.

Mesmo com toda a camada de criptografia implementada e que é necessária para o atendimento dos requisitos de segurança, o *overhead* identificado no tempo de resposta do sistema construído sobre esta arquitetura de referência, é considerado desprezível, pois não afeta significativamente o usuário final.

Trabalhos futuros podem considerar a análise do sistema utilizando tamanhos de chaves criptográficas maiores em comparação aos ganhos em segurança da solução face a um trabalho de criptologia. Outra opção e realizar uma análise com relação ao *overhead* na utilização deste sistema no que diz respeito ao: espaço necessário em disco para o armazenamento, bem como o *overhead* no tráfego da rede entre o cliente e o servidor.

VII. REFERÊNCIAS

[1] MAMEDE, Henrique São. "**Segurança Informática nas Organizações":** São Paulo, FCA, 2006, 512p

[2] ZUQUETE, André. "**Segurança em Redes Informáticas",** São Paulo, Editora Informática, 2006, 288p

[3] KORT, Henry F. et all, "**Sistema de Bancos de Dados",** São Paulo, Makron Books, 1999, 806p

[4] STALLINGS, William. "**Cryptography and Network Security: Principles and Practice",** New Jersey, Prentice Hall, 2003, 681p

[5] BURNETT, Steve; PAINE, Stephen, "**Criptografia e Segurança – O Guia Oficial RSA",** São Paulo, Campus, 2002, 392p

[6] TAYLOR, Art; BUEGE, Brian; LAYMAN, Randy. "**Segurança contra Hackers J2EE e Java",** São Paulo, Futura, 2003, 456p