ANÁLISE COMPARATIVA ENTRE PROTOCOLOS PARA TROCA DE CERTIFICADOS DIGITAIS

Ricardo C. Branco[1], **Lucas V. Dias**[1], **Tiago A. Rizzetti**[1]

[1]Curso Superior de Técnologia em Redes de Computadores
Colétio Técnico Industrial de Santa Maria (CTISM)
Universidade Federal de Santa Maria (UFSM)
Caixa Postal 97.105 - 900 – Santa Maria – RS – Brasil

{branco, lucas_dias, rizzetti}@redes.ufsm.br

Abstract. *In general, when there is a need to share a digital certificate securely, we find a problem that is often overlooked in the literature, the security involved in the process of exchanging them. Currently several protocols and certificate exchange models are accessible in the literature. This paper aims to compare the Internet Key Exchange (IKE) key exchange protocol with a new protocol proposed by the authors, which has as a premise to ensure message integrity, using digital signatures and data confidentiality, from a Diffie-Hellman agreement, and other key security aspects to be discussed throughout the article.*

Resumo. *De maneira geral, quando há necessidade do compartilhamento de um certificado digital de forma segura, encontramos um problema que muitas vezes é deixado de lado na literatura, a segurança envolvida no processo de troca dos mesmos. Atualmente diversos protocolos e modelos de troca de certificados estão acessíveis na literatura. O presente trabalho tem como objetivo comparar o protocolo para troca de chaves IKE (Internet Key Exchange) com um novo protocolo proposto pelos autores, o qual tem como premissa garantir a integridade da mensagem, utilizando-se de assinaturas digitais e confidencialidade dos dados, a partir de um acordo Diffie-Hellman, e outros aspectos essenciais no âmbito de segurança, a serem discutidos no decorrer do artigo.*

1. Introdução

Public Key Infrastructure (PKI) ou Infraestrutura de chave pública (ICP) é uma alternativa já consolidada para fornecer a capacidade de estabelecimento de relações de confiança entre entidades envolvidas em uma transação em meio digital [Moecke et al. 2010]. Atualmente muitas instituições estão utilizando a infraestrutura de chaves públicas como método de estabelecimento de confiança entre as partes envolvidas na comunicação, seja esta comunicação digital ou para estabelecer uma confiança entre entidades comunicantes.

O funcionamento de uma ICP baseia-se na premissa que uma CA *(Certificatre Authority)* raiz delega permissões para ACs (autoridade certificadora) intermediárias, e estas, por sua vez, proveem assinatura de certificados digitais para clientes finais, formando uma cadeia de permissões que depende da disponibilidade das informações de revogação, assinadas pela AC raiz para revogação de certificados.

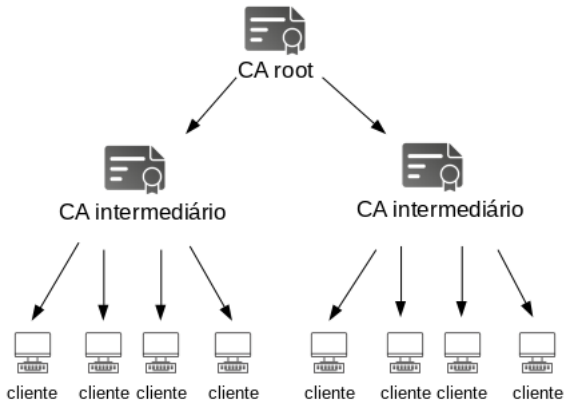

Figura 1. Estrutura de uma PKI.

A estrutura básica de uma ICP, de maneira simplificada pode ser vista na Figura 1, onde AC root é responsável por prover assinatura aos certificados de ACs intermediários, autorizando-os a assinar outros certificados, e também é responsável por divulgar a lista de revogações. AC intermediária é responsável por prover assinatura aos certificados dos clientes finais. Quando um cliente deseja realizar a troca de mensagens de forma segura com outro, este recorre ao certificado assinado pela AC de nível superior e retira do mesmo a chave pública da entidade certificadora e faz a validação do certificado recebido.

Como a necessidade de alternativas mais eficientes para a troca de certificados digitais tem tomado grande destaque mediante avanço tecnológico atual, o presente artigo vem com a proposta de um novo modelo para a troca de certificados de forma segura, o qual foi verificado utilizando-se o software *Scyther* [Casimiro et al.]. A proposta tem como premissa a troca de certificados de forma segura entre ambas as partes comunicantes, com uma autenticação mútua independente de uma terceira entidade para o mesmo. Para garantir uma autenticação bilateral sem a necessidade de uma terceira parte, ou seja, sem a necessidade de uma AC online para validação, a proposta utiliza-se de assinatura digital, criptografia assimétrica e um acordo *Diffie-Hellman* [William Stallings 2014].

2. Trabalho Proposto

A necessidade cada vez maior de uma autenticação e de uma confiança entre os envolvidos fez com que a utilização de infraestruturas de chave pública se disseminasse e trouxesse consigo uma série de limitações relacionadas a seu uso e implementação. Conforme foram sendo descobertos diferentes problemas relacionados a ICP e temas que a cercam, o mercado consumidor passou a migrar para outras alternativas mais eficientes, muitas vezes esquecendo de premissas importantes de segurança, as quais eram providas pelas ICPs [Gutmann 2002].

A estrutura básica do protocolo para autenticação mútua pode ser visto na Figura 2, onde a segurança do protocolo é provida através da assinatura, com a chave privada do remetente, do *hash* do certificado do remetente, juntamente com a *hash* da chave de sessão, escolhida através de um acordo de *Diffie-Hellman*, a fim de garantir a autenticidade da mesma. Com a assinatura deste trecho da mensagem, o resultado é concatenado com o certificado do remetente e encriptado com a chave de sessão do *Diffie-Hellman*, a

fim de garantir a integridade e confidencialidade da mensagem a ser enviada.

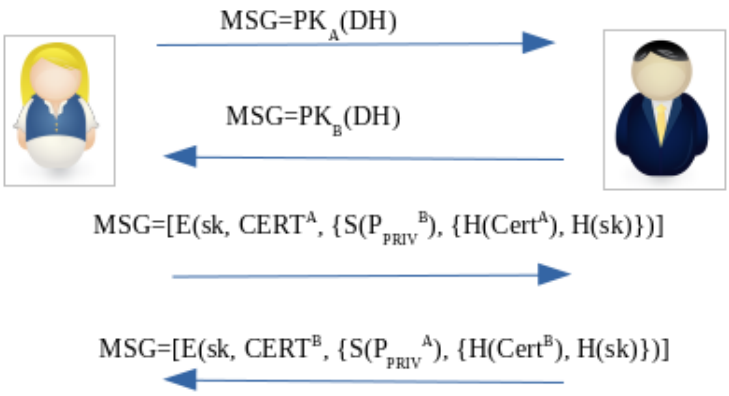

Figura 2. Estrutura do protocolo proposto.

O destinatário da mensagem, ao recebê-la, a decripta com a chave de sessão e verifica se o certificado está dentro da data de validade, caso esteja, o destinatário verifica se o mesmo não está na CRL *Certificate Revocation List*, caso não esteja, o certificado é validado. Quando o certificado não é válido, seja por estar na CRL ou por não estar no período de validade, a comunicação é encerrada.

Para que a autenticação mútua ocorra, basta que sejam enviadas quatro mensagens, duas para que ambas as partes conheçam a chave pública derivada do *Diffie-Hellman* e duas mensagens para que o certificado seja transmitido e forma segura, íntegra e confiável, independentemente da disponibilidade de uma AC no momento da comunicação.

3. Resultados e Discussões

O protocolo IKE pode ser pensado como uma combinação de dois protocolos, o *Internet Security Association e o Key Management Protocol (ISAKMP)* e o *Oakley*. O ISAKMP fornece uma estrutura para estabelecer associações de segurança e chaves criptográficas, mas não prescreve nenhum mecanismo de autenticação específico, [Meadows 1999].

O protocolo IKE foi projetado para troca de chaves e negociação de associações de segurança para comunicações seguras, [Zhou 2000], premissas as quais compartilha com a nova proposta dos autores. Utiliza-se do protocolo IPSec para prover segurança. Pode-se dizer que, para que uma comunicação segura seja estabelecida, o IKE necessita de pelo menos seis mensagens, número menor que na versão anterior (*IKEv1*), onde eram necessárias nove mensagens para a sessão, para que então, alcance o estado de *START_IPSEC_SA_PROTECTION* e estabeleça sua SA.

O esquema de mensagens trocadas pelo protocolo pode ser visto em [Abdel Hakeem et al. 2017] como mostrado na Figura 3. A primeira fase na troca de mensagens (*IKE_INIT*) define uma associação de segurança *IKE* (*IKE_SA*), que tem como função a definição de parâmetros que posteriormente servirão como parâmetros de segurança para o IKE e para a definição do algoritmo *Diffie-Hellman*. A segunda fase da troca de mensagens consiste na troca de informações referentes a autenticação, identificação e chaves secretas, definida como (*IKE_AUTH*).

A segunda fase na troca de mensagens é iniciada após o encerramento da primeira fase, nesta segunda etapa são criadas as SAs opcionais, denominadas CHILD-SAs, que será utilizado para a geração de novas chaves *Diffie-Hellman* para proteção IPSec. Todas as trocas de mensagens da segunda etapa são enviadas de forma segura, uma vez que na primeira fase são definidos as chaves e algoritmos a serem utilizados.

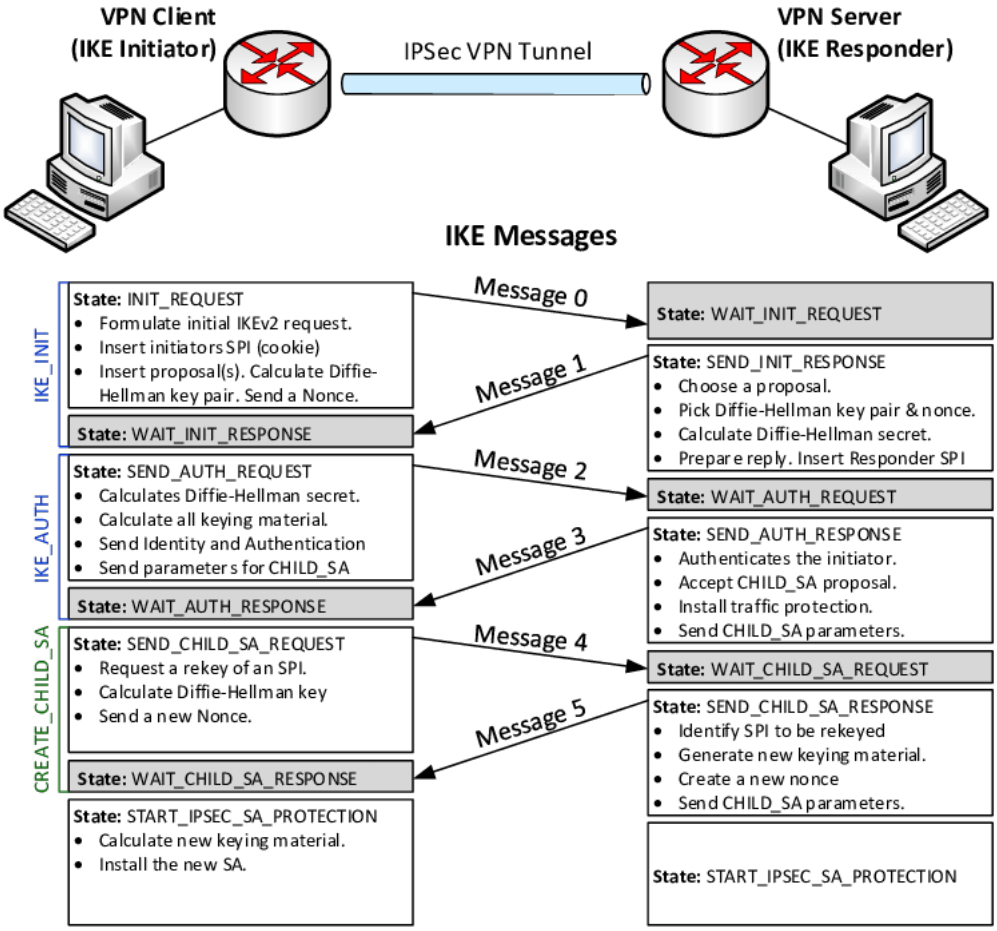

Figura 3. Troca de mensagens do protocolo IKEv2.

Como pode ser visto na proposta do novo protocolo, o número de mensagens enviadas para que ambos usuários, remetente e destinatário, confiem um no outro, é de apenas quatro, sendo duas delas para o compartilhamento de chaves públicas e duas para a autenticação mútua, enquanto que, para termos uma comunicação segura utilizando-nos do protocolo *IKE*, necessitamos de seis mensagens, além de uma entidade que garantirá a identidade dos nós participantes na comunicação.

Os resultados obtidos utilizando-se a ferramenta para análise de protocolos podem ser vistos na Figura 4 e são promissores, uma vez que foram testados pelo *software* e não apresentaram vulnerabilidades aparentes que possam ser exploradas por atacantes e levar a uma falta de confiança das partes durante a comunicação. Futuramente será desenvolvida uma aplicação com intuito de testar os diversos tipos de ataques aos quais uma comunicação está sujeita, bem como ataques de reflexão, temporização, replicação e ataques de interceptação destes dados, seja para alteração dos mesmos ou para a tentativa de extração de informações das mensagens.

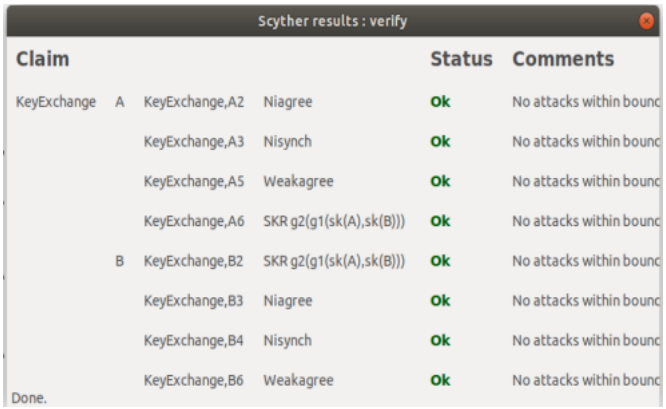

Figura 4. Troca de mensagens do protocolo proposto e possíveis ataques.

Para realizar o envio e recebimento das mensagens na origem da comunicação no *software Scyther* foi utilizada a sintaxe demonstrada no quadro abaixo, onde realizamos a assinatura com sua chave privada, e encriptação com a chave de sessão, como já foi mencionado anteriormente, além de outras operações, a fim de garantir os demais aspectos de segurança.

```
 //Send its own cert on plain text and the encripted sign of own cert and the hash of
DHE session key with DHE session key

 send_1(A, B, {certA, {h(certA), h(SessionKeyDHE), h(A)} sk(A)} k(SessionKeyDHE));

 //Receive its own cert on plain text and the encripted sign of own cert and the hash of
DHE session key with DHE session key


 recv_2(B, A, {certBrcv, {h(certBrcv), h(SessionKeyDHE), h(B)} sk(B)}
k(SessionKeyDHE));
```

No destinatário da comunicação, ocorre uma inversão na ordem, o mesmo recebe a mensagem e, realiza encima desta, as operações necessárias para garantir que o certificado é válido, está dentro do prazo de validade e que a assinatura da AC está em vigor, além de verificar se o mesmo não está na CRL. Caso todas informações estejam corretas, este envia a mensagem para que o outro lado da comunicação realize as mesmas operações, fazendo, com isso, que a troca de certificados ocorra de maneira íntegra e segura.

```
 //Receive its own cert on plain text and the encripted sign of own cert and the hash of
DHE session key with DHE session key

 recv_1(A, B, {certArcv, {h(certArcv), h(SessionKeyDHE), h(A)} sk(A)}
k(SessionKeyDHE));

 //Send its own cert on plain text and the encripted sign of own cert and the hash of
DHE session key with DHE session key


 send_2(B, A, {certB, {h(certB), h(SessionKeyDHE), h(B)} sk(B)} k(SessionKeyDHE));
```

4. Considerações Finais

Mesmo com um número menor de mensagens, garantimos integridade, confidencialidade e autenticidade das mensagens, pilares essenciais na área de segurança, e que, muitas vezes, acabam sendo deixados de lado em algumas implementações.

O protocolo proposto pelos autores não vem com o intuito de substituição ao IKE, mas sim uma alternativa, uma vez que, ambos os comunicantes detenham certificados assinados e validados pela CA, não há a necessidade da disponibilidade da mesma, como ocorre com o protocolo para troca de chaves na *internet*. A grande vantagem em relação ao IKE é a independência de uma AC online para a autenticação dos nós, uma vez que a nova proposta baseia-se em métodos isolados da autoridade certificadora para garantir a identidade dos nós e a autenticidade da comunicação.

Para garantir a autenticidade dos comunicantes basta que o nó solicite mediante uma autoridade certificadora seu certificado válido e assinado e mantenha consigo uma URL (*Uniform Resource Locator*) para consulta dos certificados revogados. Tendo posse destas informações, em ambos os lados os quais desejam comunicar-se, é possível realizar uma autenticação mútua entre os nós da comunicação.

O protocolo proposto passou por testes realizados com o *Scyther* e apresentou um ótimo resultado tendo em vista sua proposta simples e de baixa complexidade, pois não apresentou nenhuma vulnerabilidade com o *software*. Sua segurança baseia-se nas propriedades matemáticas oferecidas pelo acordo *Diffie-Hellman*, que, embora não possa ser totalmente representado no *software* escolhido devido à falta de suporte da ferramenta a suas características, foi amplamente testado na literatura e concluiu-se que este é matematicamente seguro, uma vez que não é viável a quebra de seu algoritmo de cifração em tempo hábil.

Outra característica que leva o protocolo proposto a ser considerado seguro é sua flexibilidade de suporte a diferentes algoritmos de *hash* e criptografia assimétrica, que, conforme a implementação podem ser variados.

Com a utilização do modelo proposto, torna-se menos complexo e mais rápido a troca de certificados de forma segura e *online*, sem a necessidade de métodos *offline* para que o compartilhamento destes certificados ocorra de forma totalmente segura e rápida.

Referências

Abdel Hakeem, S., Arslan, S., and Kim, H. (2017). Ike hardware engine based on cam for concurrent processing of massive user sessions. pages 154–159.

Casimiro, A., d Lemos, R., and Gacek, C. Operational semantics and verification of security protocols.

Gutmann, P. (2002). Pki: it's not dead, just resting. *Computer*, 35(8):41–49.

Meadows, C. (1999). Analysis of the internet key exchange protocol using the nrl protocol analyzer. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No. 99CB36344)*, pages 216–231. IEEE.

Moecke, C. T., Custódio, R. F., Kohler, J. G., and Carlos, M. C. (2010). Uma icp baseada em certificados digitais autoassinados. *Simpósio Brasileiro em Segurança da Informaç ao e de Sistemas Computacionais*, pages 91–104.

William Stallings, L. B. (2014). *Segurança de Computadores: Princípios e Práticas*. Rio de Janeiro.

Zhou, J. (2000). Further analysis of the internet key exchange protocol. *Computer Communications*, 23(17):1606–1612.