DIFFIE-HELLMAN, PRF E KDF: UMA ALTERNATIVA PARA INICIALIZAR IDVVS

Rodrigo Souza [1]

Diego Luis Kreutz [2]

Rodrigo Machado Bisso [3]

Diego Luis Kreutz [4]

Resumo:

Um recente estudo propôs um método simples e robusto, denominado iDVV (integrated Device Verification Value), para assegurar a segurança da informação entre duas entidades A e B, onde A pode ser um controlador e B um comutador de rede, por exemplo. Mais especificamente, o iDVV representa uma alternativa segura e barata de gerar material criptográfico para dispositivos de baixa capacidade computacional e que demandam alto desempenho (exemplo: baixa latência no processamento de pacotes de dados). Este método, combinado com a biblioteca NaCl e um KDC (Key Distribution Center) tradicional como o Kerberos, representa uma alternativa (para cenários específicos como infraestruturas de rede) ao TLS e PKI. Um dos maiores desafios na proposta do iDVV é o KDC, que é responsável pela inicialização do gerador de iDVVs em ambos os dispositivos A e B, que buscam estabelecer uma comunicação segura entre si. O KDC age como uma terceira parte confiável, mais conhecida como TTP (Trusted Third Party), que é utilizada por todos os dispositivos. Entretanto, esta TTP pode levar a diferentes problemas de segurança, incluindo quebra de confidencialidade e indisponibilidade. Primeiro, o KDC representa um único ponto de falha. Segundo, se o KDC for comprometido, todas as comunicações entre quaisquer duas entidades A e B podem (eventualmente) ser comprometidas. Neste contexto, o objetivo deste trabalho é responder a seguinte pergunta. Existe algum método que pode ser utilizado entre A e B, para inicializar o iDVV de maneira privada e segura, sem depender de um KDC (TTP)? Para conseguir responder a esta pergunta, o desenvolvimento do trabalho foi dividido em cinco etapas. O resultado das duas primeiras etapas foi a seleção dos algoritmos de handshake do TLS e Diffie-Hellman (DH) para inicialização do iDVV. Tanto o desempenho quanto a viabilidade técnica de ambos os métodos foram avaliados no trabalho. Os resultados apontam que o algoritmo de DH é mais simples, tem um desempenho superior (menor latência) e é igualmente seguro para inicializar o mecanismo de geração de iDVVs.

Palavras-chave: integrated Device Verification Value (iDVV), DIFFIE-HELLMAN, Pseudorandom

Function (PRF), Key Derivation Function (KDF), Transport Layer Security (TLS), TLS Handshake

Modalidade de Participação: Iniciação Científica

DIFFIE-HELLMAN, PRF E KDF: UMA ALTERNATIVA PARA INICIALIZAR IDVVS

[1] Aluno de graduação. rodrigomsr2@gmail.com. Autor principal

[2] Mestre em Informática. diego.unipampa@gmail.com. Co-autor

[3] Aluno de graduação. rodrigobissomachado@gmail.com. Co-autor

[4] Docente. diegokreutz@unipampa.edu.br. Orientador

DIFFIE-HELLMAN, PRF E KDF: UMA ALTERNATIVA PARA INICIALIZAR iDVVs

1 INTRODUÇÃO

A solução mais genérica disponível e utilizada na Internet para garantir a segurança da informação entre duas entidades que estão comunicando entre si é o protocolo TLS (*Transport Layer Security*) [E. Rescorla 2008]. Para funcionar, o TLS depende da infraestrutura de chaves públicas, mais conhecida como PKI [Adams, C. e Lloyd, S. 1999]. Na PKI, as chaves privadas e públicas são utilizadas para gerar, assinar e validar certificados. Os certificados podem ser de propriedade de uma pessoa ou de um nome de domínio. O caso mais comum são os certificados de domínio. Como exemplo, a UNIPAMPA gera um certificado para o domínio www.unipampa.edu.br e pede para uma autoridade certificadora assinar este certificado. Após assinado, qualquer entidade terceira, incluindo navegadores *Web*, poderá verificar a autenticidade e validade do certificado, desde que haja uma relação de confiança com a autoridade certificado (exemplos de algumas autoridades certificadoras com reconhecimento global: Let's Encrypt, Comodoro, Symantec, Digicert, GeoTrust).

O TLS é o protocolo que tem por objetivo garantir a segurança das comunicações através dos certificados da PKI. Em outras palavras, há um laço de confiança entre o TLS e os certificados e chaves da PKI. Apesar da grande gama de aplicações e uso do TLS e PKI na Internet, há vários desafios em aberto. No caso da PKI, além de apresentar desafios de complexidade e custo operacional, esta infraestrutura tem sido um alvo recorrente de ataques [J. Yu and M. Ryan 2017]. Similarmente, apesar do uso e aceitação do TLS, este protocolo sofre de diferentes causas, como (1) os desafios de implementação devido ao fato de ser um protocolo muito complexo, (2) a necessidade de manter compatibilidade com versões anteriores, (3) a grande dependência no serviço de nome (DNS), (4) vulnerabilidades recorrentes em diferentes níveis de implementação do protocolo, (5) foco na *Web*, deixando a desejar em outros cenários mais específicos, e (6) desempenho [Kreutz, D., et. al. 2018][Kreutz, D. et. al. 2017].

Levando em consideração este cenário, um recente estudo propôs um método simples e robusto, denominado iDVV (***i**ntegrated **D**evice **V**erification **V**alue*), para assegurar a segurança da informação entre duas entidades A e B, onde A pode ser um controlador e B um comutador de rede, por exemplo [Kreutz, D., et al. 2017]. Mais especificamente, o iDVV representa uma alternativa segura e barata de gerar material criptográfico para dispositivos de baixa capacidade computacional e que demandam alto desempenho (exemplo: baixa latência no processamento de pacotes de dados). Este método, combinado com a biblioteca NaCl [D. Bernstein, et. al. 2012] e um KDC (*Key Distribution Center*) tradicional como o Kerberos [Steiner, J. G. et. al. 1988], representa uma alternativa (para cenários específicos como infraestruturas de rede) ao TLS e PKI.

Um dos maiores desafios na proposta do iDVV é o KDC, que é responsável pela inicialização do gerador de iDVVs em ambos os dispositivos A e B, que buscam estabelecer uma comunicação segura entre si. O KDC age como uma terceira parte confiável, mais conhecida como TTP (*Trusted Third Party*), que é utilizada por todos os dispositivos. Entretanto, esta TTP pode levar a diferentes problemas de segurança, incluindo quebra de confidencialidade e indisponibilidade. Primeiro, o KDC representa um único ponto de falha. Segundo, se o KDC for comprometido, todas as comunicações entre quaisquer duas entidades A e B podem (eventualmente) ser comprometidas.

O objetivo principal deste trabalho é responder a seguinte pergunta. *Existe algum método que pode ser utilizado entre A e B, para inicializar o iDVV de maneira privada e segura, sem depender de um KDC (TTP)?* A resposta a esta pergunta passa pela análise de

algoritmos como o *handshake* do TLS e o *Diffie-Hellman* (DH) [Diffie, W., & Hellman, M. 1976]. Ambos são utilizados para gerar material criptográfico (exemplo: chaves mestra, chaves secreta de sessão), que é posteriormente utilizado para fornecer propriedades de segurança essenciais, como autenticidade, integridade e confidencialidade, às comunicações entre duas entidades A e B.

O desenvolvimento do trabalho foi dividido em cinco etapas. As duas primeiras etapas são destinadas ao estudo do estado da arte e potenciais métodos para inicializar o iDVV, respectivamente. A implementação dos métodos selecionados faz parte da terceira etapa. Na sequência, a quarta etapa é destinada a avaliação dos métodos implementados. Finalmente, a quinta etapa é destinada a discussão dos resultados e prospecção de trabalhos futuros.

O resultado das duas primeiras etapas foi a seleção do *handshake* do TLS e *Diffie-Hellman* para inicialização do iDVV. Tanto o desempenho quanto a viabilidade técnica de ambos os métodos são avaliados no trabalho. Os resultados apontam que o DH é mais simples, tem um desempenho superior (menor latência) e é igualmente seguro para inicializar o mecanismo de geração de iDVVs.

As contribuições deste trabalho podem ser resumidas em:

- revisão do estado da arte relativo a mecanismos de geração de material criptográfico para o estabelecimento de comunicação segura entre duas entidade A e B;
- investigação e análise de diferentes métodos para a inicialização do iDVV, sem depender do KDC, entre duas entidades comunicantes A e B;
- implementação e discussão de uma solução, baseada no DH + PRF (*Pseudo-Random Function*) + KDF (*Key Derivation Function*), para inicializar iDVVs.

O restante deste trabalho está organizado como segue. A Seção 2 descreve as etapas de desenvolvimento do trabalho. Na sequência, um resumo do estado da arte é apresentado na Seção 3. Já a implementação dos métodos escolhidos e a discussão dos resultados são discutidos na Seção 4. Por fim, as considerações finais são apresentadas nas Seções 5.

2 METODOLOGIA

O desenvolvimento da pesquisa foi estruturado em cinco etapas. Na **primeira etapa** foi realizado o estudo de conceitos básicos e do estado da arte, incluindo métodos, mecanismos e protocolos como PRF, KDF, DH, TLS e iDVV.

A pesquisa e análise preliminar dos métodos candidatos a substituir o KDC na inicialização do iDVV foram realizadas na **segunda etapa** do trabalho. Nesta etapa foram selecionados o *handshake* do TLS e o *Diffie-Hellman*.

Os métodos escolhidos foram implementados, utilizando a linguagem de programação Python, na **terceira etapa**. O motivo principal da escolha do Python foi a experiência com a linguagem, as bibliotecas disponíveis e a produtividade inerente à linguagem. Na implementação do *handshake* TLS foi utilizada a biblioteca pyOpenSSL, na versão 18.0.0. No caso do DH, foram utilizadas apenas bibliotecas padrão da linguagem.

Na **quarta etapa** foram realizados testes de desempenho, para medir a latência, tempo de CPU (tanto no modo usuário, como no modo kernel) e número de chamadas (funções e primitivas) dos dois métodos. Para automatizar o processo, foi implementado um *shell script* Bash. Foram realizadas 10.000 execuções para cada método. Para o teste de latência foi utilizado o método "time()" da biblioteca "time" da linguagem python. Os tempos de CPU foram medidos utilizando a função *time* do Bash. O número de chamadas foi medido utilizando a biblioteca "*cProfile*" do Python. Os resultados, apresentados na Seção 5, representam a média das 10.000 execuções. Os testes foram realizados em um Notebook HP 14-d030br, Intel core i5, segunda geração, memória de 8 GB e 240 GB de SSD e com o sistema operacional Ubuntu 17.10.

Finalmente, a análise e discussão dos resultados e comparações empíricas e analíticas foram realizadas na **quinta etapa** do trabalho, conforme pode ser observado na Seção 5.

3 CONCEITOS E ESTADO DA ARTE

A segurança da comunicação entre duas entidades A e B é ancorada em geradores de material criptográfico de qualidade. Tradicionalmente, algoritmos, funções e protocolos como DH, PRF, KDF e TLS *handshake* são utilizadas para este fim, ou seja, gerar chaves mestra, ou chaves secretas de longa duração, que, por sua vez, serão utilizadas para derivar outras chaves secretas de curta duração, menos críticas, como as chaves de sessão.

O *Diffie-Hellman* [Diffie, W., & Hellman, M. 1976] foi uma das primeiras soluções (e ainda hoje é a mais utilizada) projetadas para a geração de chaves compartilhadas entre duas entidades A e B, sobre uma rede pública. No caso do DH, a chave secreta é construída em conjunto por A e B. De forma resumida, primeiro, são definidos dois valores públicos $\boldsymbol{p}$ e $\boldsymbol{g}$, que são números primos e acordados entre A e B. Esses dois números são utilizados para a geração de um terceiro valor, que será trocado entre A e B. A geração do terceiro valor se dá pelo cálculo $X = \boldsymbol{g}^{\boldsymbol{a}} \; mod \; \boldsymbol{p}$ na entidade A e $\boldsymbol{Y} = \boldsymbol{g}^{\boldsymbol{b}} \; mod \; \boldsymbol{p}$ na entidade B, onde $\boldsymbol{a}$ e $\boldsymbol{b}$ são valores inteiros quaisquer escolhidos por A e B e que não são mantidos em segredo. Na sequência, A e B trocam entre si os valores X e Y e geram a chave secreta compartilhada. A calcula $\boldsymbol{s} = \mathbf{Y}^{\mathbf{a}} \bmod \boldsymbol{p}$, e B calcula $\mathbf{s} = \mathbf{X}^{\mathbf{b}} \bmod \mathbf{p}$, onde $\boldsymbol{s}$ é a chave compartilhada. O valor $\boldsymbol{s}$, calculado por A e B, é o mesmo pois $(\boldsymbol{g}^{\boldsymbol{a}} \; mod \; \boldsymbol{p})^{\mathbf{b}} \bmod \boldsymbol{p} = (\boldsymbol{g}^{\boldsymbol{b}} \; mod \; \boldsymbol{p})^{\boldsymbol{a}} \bmod \boldsymbol{p}$.

Uma PRF [M. Bellare et. al. 1996], ou função pseudo-randômica, é um método que recebe um certo valor como entrada e, a partir dele, produz um outro valor computacionalmente indistinguível de uma sequência verdadeiramente aleatória. Na prática, protocolos e mecanismos de segurança, como é o caso do TLS, utilizam PRFs para gerar chaves mestras a partir de um determinado conjunto de dados de entrada. Em trabalhos recentes, PRFs foram adaptadas para gerar material criptográfico, de maneira determinística, em algoritmos de replicação de máquinas de estado [Kreutz, D. et. al. 2014].

Uma KDF [Carlisle Adams et. al. 2004], ou função de derivação de chave, tem como objetivo obter uma ou mais chaves baseadas em um segredo, este podendo ser uma senha, palavra-chave ou até mesmo uma outra chave secreta. KDFs são comumentemente utilizadas para "esticar" uma chave curta em chaves mais longas, ou ainda para "converter" chaves de um formato para outro (exemplo: chave resultante do DH em uma chave simétrica para usar em algoritmos como o AES). Uma chave derivada, "esticada", pode ser representada pelo protótipo de função DK = KDF(*chave*, *salt*, *iterações*), onde DK é a chave derivada, KDF é a função de derivação, geralmente utilizando funções de hash criptográficas, *chave* é a chave de entrada, *salt* é um dado aleatório e *iterações* representa o número iterações que será realizadas sobre a *chave* e o *salt* (exemplo: N iterações da função hash criptográfica H).

O *handshake* do TLS é a fase inicial do protocolo, onde as duas entidades A e B da comunicação estabelecem um processo de troca de informação que resulta em uma chave mestra. Desta chave resultam chaves secretas, derivadas, que são utilizadas para garantir a segurança das comunicações. O processo é complexo e pode ser dividido em 6 fases [Rescorla, E. 2018]. De forma resumida, o cliente e o servidor trocam mensagens em si, enviando números pseudo-aleatórios, certificados, informações sobre algoritmos de cifra suportados, entre outras coisas. Os números pseudo-aleatórios são utilizados, entre outras coisas, para gerar as chaves denominadas pré-mestra. Os certificados podem ser utilizados para autenticação mútua em cliente e servidor. Por fim, o cliente e o servidor irão negociar o algoritmo de cifra que será utilizado para garantir a confidencialidade da comunicação.

O iDVV é um mecanismo criado para gerar valores determinísticos, locais e não-reconhecíveis, que podem ser utilizados como material criptográfico forte (exemplo:

chave simétrica secreta). Um iDVV é inicializado a partir de uma chave secreta pré-definida e de uma semente (valor pseudo-aleatório), fornecidos por um KDC. Depois de inicializado, o iDVV pode ser utilizado de forma síncrona entre A e B, sem nenhuma troca de informação pela rede sobre os dados utilizados para inicializar ou atualizar o iDVV.

O iDVV pode, então, ser utilizado de diferentes formas: 1) por mensagem; 2) por sequência de mensagens; 3) por intervalo de tempo e 4) por sessão de comunicação. Um dos motivos do iDVV ser considerado uma melhor opção em alguns cenários, em relação aos algoritmos tradicionais, se explica por ele ser gerado localmente na máquina, ao invés de ser acordado por mensagens. Outro motivo é o fato de a geração de um iDVV ser de baixíssimo custo, sendo uma boa opção para cenários que demandam uma chave secreta por mensagem e alto desempenho, por exemplo.

4 IMPLEMENTAÇÃO E RESULTADOS

Método 1: DH + PRF + KDF. O DH é utilizado para criar uma chave pré-mestra, entre as duas entidades A e B. Na sequência, ambas as entidades utilizam uma PRF para gerar a chave mestra. A partir dela são derivadas a chave secreta e a seed (semente) utilizando uma KDF. Por fim, esses valores são utilizados para inicializar o iDVV. A chave mestra pode ser utilizada, também, para a sincronização do iDVV, conforme métodos descritos em [Kreutz, D. et. al. 2017]. Para garantir PFS (*Perfect Forward Secrecy*) [Krawczyk H. 2011], ou seja, mesmo descobrindo a chave mestra atual, o atacante não conseguirá decifrar mensagens que utilizam as versões anteriores da chave, a chave mestra é evoluída toda vez que for utilizada para inicializar um iDVV. A evolução ocorre através de uma função hash criptográfica na forma de *chave mestra = HASH(chave mestra)*.

Método 2: *handshake* do TLS. O handshake do TLS é utilizado como método para inicializar o iDVV. Como mencionado anteriormente, o TLS depende da PKI.

A **Tabela 1** apresenta uma comparação entre os dois métodos implementados. Como pode ser observado, são avaliadas a complexidade, o custo e a latência (desempenho).

Tabela 1- Comparação entre os dois métodos implementados

Métrica	Diffie-Hellman/ iDVV	Handshake TL
Complexidade	baixa	alta
Custo	baixo	alto
Latência(µs)	524	2749

Com relação à complexidade dos dois métodos, o DH apresenta uma complexidade baixa quando comparada ao *handshake* do TLS. O DH é um algoritmo simples, de fácil implementação. Por outro lado, o *handshake* do TLS é um protocolo complexo, que depende de várias variáveis, requer a troca de várias mensagens entre A e B, e possui uma implementação complexa (integrada aos sub-protocolos do TLS) e grande (muitas linhas de código). Esse argumento é reforçado quando observamos a quantidade de funções e primitivas chamadas por cada um dos métodos. O *handshake* do TLS realiza aproximadamente 6x mais chamadas de função que o DH.

O segundo ponto de avaliação é o custo. Enquanto o custo da implementação e computação do DH + PRF + KDF é baixo, o custo do *handshake* do TLS é bastante elevado.

Finalmente, a análise de desempenho, observando a latência e o tempo de execução na CPU dos dois métodos, não deixa dúvida sobre o método mais adequado para inicializar o iDVV. Enquanto o método do *handshake* do TLS obteve uma média de 2749µs por requisição e um tempo de CPU (modo kernel + modo usuário) de 17.23m, o DH + PRF + KDF obteve uma média de apenas 524µs por requisição e um tempo de CPU de 8.17m, ou seja, aproximadamente 5x mais eficiente em termos de latência e 2x menos dispendioso em termos de tempo de utilização da CPU.

5 CONSIDERAÇÕES FINAIS

Os dois métodos investigados, DH+PRF+KDF e *handshake* TLS, podem ser utilizados para substituir o KDC na inicialização do iDVV entre duas entidades A e B. Entretanto, como discutido na Seção 4, o método DH+PRF+KDF é mais eficiente em todos os parâmetros de avaliação considerados (complexidade, custo e desempenho/latência).

Trabalhos futuros. Implementação de uma versão do KDC utilizando o método proposto neste trabalho. Aplicação de tecnologias como SGX/Intel no contexto do método proposto. Avaliação de cenários de aplicação prática do iDVV, como *Internet of Things* (IoT).

REFERÊNCIAS

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory*, *22*(6), 644-654.

Kreutz, D., Yu, J., Esteves-Verissimo, P., Magalhaes, C., & Ramos, F. (2017). The KISS principle in software-defined networking: an architecture for keeping it simple and secure. *arXiv preprint*.

Rescorla, E. (2018). *The transport layer security (TLS) protocol version 1.3* (No. RFC 8446)

Carlisle Adams, Guenther Kramer, Serge Mister and Robert Zuccherato, "On The Security of Key Derivation Functions", ISC'2004, LNCS 3225, 134-145.

M. Bellare, R. Canetti, and H. Krawczyk. "Pseudorandom Functions Revisited: The Cascade Construction and Its Concrete Security", Proc. 37th FOCS, pages 514–523. IEEE, 1996.

Kreutz, D., Bessani, A., Feitosa, E., & Cunha, H. (2014, November). Towards secure and dependable authentication and authorization infrastructures. In IEEE PRDC. IEEE, 2014.

Steiner, J. G., Neuman, B. C., & Schiller, J. I. (1988, February). Kerberos: An Authentication Service for Open Network Systems. In *Usenix Winter* (pp. 191-202).

J. Yu and M. Ryan, "Evaluating web PKIs," in Software Architecture for Big Data and the Cloud. Elsevier, June 2017, ch. 7.

Kreutz, D., Yu, J., Ramos, F. & Esteves-Verissimo, P., (2018). ANCHOR: logically-centralized security for Software-Defined Networks. *arXiv preprint*.

D. Bernstein, T. Lange, and P. Schwabe, "The security impact of a new cryptographic library," in Progress in Cryptology - LATINCRYPT, ser. L. N. in CS. Springer, 2012, vol. 7533.

Adams, C. e Lloyd, S. Understanding Public-Key Infrastrucuture: Concepts, standards and deployment considerations. Macmillan Technical Publishing, 1999;

Krawczyk H. (2011) Perfect Forward Secrecy. In: van Tilborg H.C.A., Jajodia S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA