UM SERVIÇO DISTRIBUÍDO UTILIZANDO REDES DHT PARA CONTROLE DE ACESSO EM ESPAÇOS FÍSICOS

Jean T.Garcia[1], **Lucas Vargas Dias**[1], **André Rech Eichner**[1], **Tiago Antonio Rizzetti**[1]

[1]Curso Superior de Tecnologia em Redes de Computadores –
Universidade Federal de Santa Maria (UFSM)
Caixa Postal 97.105-900 – Santa Maria – RS – Brazil

{jeangarcia, lucas_dias, eichner}@redes.ufsm.br, rizzetti@ctism.ufsm.br

Abstract. *This article describes the design and implementation of a distributed hash table (DHT) based overlay network with the coupling of a public key infrastructure to provide authentication. In addition, the proposed architecture uses the DHT network topology to distribute and synchronize information from a database to provide auditing and access control over physical environments.*

Resumo. *Este artigo descreve o projeto e implementação de uma rede de sobreposição baseada em tabelas de hash distribuídas (DHT) com o acoplamento de uma infraestrutura de chave pública para prover autenticação. Além disso, a arquitetura proposta utiliza da topologia da rede DHT para distribuir e sincronizar informações de um banco de dados para prover auditoria e controle de acesso sobre ambientes físicos.*

1. Introdução

O projeto *Environment Secury Control* (ESC) foi inicialmente desenvolvido em uma arquitetura cliente-servidor devido a facilidade de gerenciamento, entretanto, no decorrer de sua utilização, foi percebido uma grande dependência da comunicação em tempo real entre cliente-servidor e a baixa resiliência em tal arquitetura. Nesse contexto, foi notado que o aprimoramento da arquitetura poderia ser feito através da utilização de uma rede *peer-to-peer*(P2P) na qual tem maior tolerância a falhas, uma vez que não existe a dependência de um nó central e pela sua escalabilidade.

Ainda assim, sabendo que a topologia da rede de sobreposição baseada em tabelas de hash distribuídas(DHT) que segundo [Martinez-Yelmo et al. 2008] possui características particulares desejáveis em um sistema computacional, como tolerância a falhas, resiliência, escalabilidade e alta disponibilidade. O respectivo trabalho utiliza da mesma para prover um serviço de sincronismo de banco de dados que possuem informações de acesso para ambientes físicos da arquitetura ESC, além de prover um mecanismo de autenticação através de certificados digitais utilizando uma infraestrutura de chave pública.

Para validar a arquitetura, foi feita a análise da quantidade em tráfego de rede e tempo de sincronismo das informações, esse segundo se dá ao fato de, por se tratar de um sistema de controle de acesso tem de ser realizado em um tempo razoável.

Com isso, o restante do artigo está organizado na forma que segue: na seção 2 são apresentados os trabalhos relacionados, na seção 3 é descrita a arquitetura proposta. Na seção 4 são apresentados os resultados experimentais da implementação e por fim, considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Conforme [Kubler et al. 2015], a prática de sincronismo de dados tem de levar em consideração a contextualização da aplicação pois existem diferentes formas no momento em que as atualizações precisam ser propagadas. Sendo uma delas de modo síncrono onde tem de ser informado a cada nó que possui a informação, que a mesma será alterada e a outra de modo assíncrona, onde as informações são alteradas sem aviso prévio aos pares que a possuem. Além disso, também tem de ser levado em consideração onde as atualizações devem ser realizadas, podendo adotar um dos seguintes princípios: i) cópia primária que consiste na modificação de informação feita por um nó e então retransmitido aos demais nós, e ii) atualizações em todos os lugares onde todos os nós fazem modificações sobre as informações. Neste contexto, é possível verificar diversas metodologias propostas na literatura que utilizam redes DHT, destacando algumas descritas abaixo.

[Liu and Lai 2017] apresentam um mecanismo de redes móveis P2P em que a rede principal é formada por líderes de grupos diferentes onde o sincronismo de informações se dá em duas ocasiões: i) um nó envia seus dados ao nó vizinho quando se desvincula da rede em um tempo pré-definido denominado tempo de segurança; e ii) quando uma informação é transmitida de um grupo ao outro, cada nó retransmissor que possui a *InfoHsh* referente aos dados, tem as informações atualizadas. Já [Mehta et al. 2011] sugere um cache de banco de dados distribuídos para serviços da Web onde o sincronismo de informações entre o serviço de cache e o banco de dados ocorre em dois momentos: i) quando o buffer de informações do servidor cache estiver cheio e ii) em períodos previamente definidos pela aplicação.

A arquitetura proposta se diferencia pelo fato de utilizar o princípio de cópia primária para modificação de dados e modo síncrono para a atualização de informações. Ainda assim, redes de sobreposição não são seguras, a propriedade distribuída de uma rede de sobreposição torna o problema de segurança desafiador. A existência de nós mal intencionados é comum na maior parte do tempo, dessa forma um sistema deve ser resiliente perante a presença de nós maliciosos [Avramidis et al. 2012]. Buscando tratar dos problemas descritos, alguns mecanismos são propostos, destacando-os a seguir:

Segundo [Takeda et al. 2008], métodos de autenticação em redes P2P podem ser divididos em duas categorias, a primeira autêntica identificadores de nó, a segunda refere-se às permissões do usuário. O trabalho foca na primeira categoria, onde é proposto um método de autenticação de usuário, no entanto não é tratado a autenticação inicial de nós. Por outro lado, [Haowei and Yubo 2010] apresentam um modelo de autenticação baseado em confiança dos nós, onde é proposto um método de cálculo de confiança entre os nós para realizar a autenticação de certificados e também uma estrutura própria de revogação de certificados. No entanto, [Haowei and Yubo 2010] também não tratam da autenticação inicial. O respectivo trabalho se difere pela utilização de uma infra-estrutura de chave pública para prover autenticação abordando a autenticação inicial e permissões de cada nó.

3. Arquitetura proposta

A arquitetura ESC utilizada atualmente é dependente de um serviço denominado *ESC Manager*(ESCMA) que faz o tratamento de eventos em tempo real ocorridos no *ESC Hardware*(ESCHA) que por sua vez, faz sensoriamento e atuação no ambiente em que

se encontra, a comunicação entre eles se dá em tempo real em uma arquitetura cliente-servidor, as operações nesse sentido são denominadas síncronas, a arquitetura também possui uma interface Web chamada ESC *Interface*(ESCI) que permite a interação entre usuário e ESCHA [Pedrozo et al. 2018]. Com isso, no momento que há falha na comunicação, o ESCHA tem de permancer fazendo sensoriamento e atuação sobre seu ambiente, para isso, a arquitetura proposta utiliza da topologia da rede DHT para diminuir a dependência de um nó centralizador e para o compartilhamento de informações de controle de acesso entre os nós.

Sendo assim, a arquitetura proposta vista na Figura 1 na qual migração da arquitetura ESC de cliente-servidor para P2P é formada por 5 componentes principais: i) Banco de dados: através das informações contidos nesta base que tem início o sistema de autenticação e controle de acesso, ii) ESCMA: software gerente responsável pela identificação dos dispositivos na rede, iii) DHT ESCMA: módulo DHT do ESCMA que faz a publicação e atualização de informações sobre o banco de dados e sobre a rede DHT, iv) ESCHA: sensor ou atuador do ambiente em que se encontra através de mecanismo de controle de acesso seguros por meio de informações obtidas na rede DHT; e v) DHT ESCHA: módulo DHT do ESCHA responsável pela busca de informações sobre o controle de acesso do mesmo, além de publicar informações referentes aos eventos ocorridos no ESCHA.

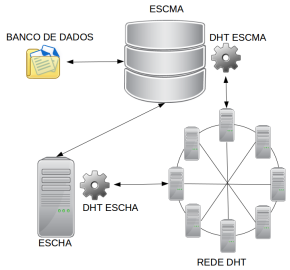


Figura 1. Arquitetura de sincronismo

A arquitetura tem finalidade de eliminar a dependência de um único nó, para isso, utiliza da topologia da rede de sobreposição P2P estruturadas baseadas em DHT, além de usá-lo como mecanismo para prover o sincronismo de informações para controle de acesso. Optou-se pelo uso da rede DHT por fornecer uma estrutura auto organizada para aplicações ponto-a-ponto que exigem escalabilidade, resiliência, tolerância a falhas e alta disponibilidade [Kwon et al. 2008]. Também, pelo fato de que cada nó mantém as informações em cache de cada mensagem, garantindo a indisponibilidade de n-1 nós, característica que torna a rede DHT descentralizada e distribuída [Guangmin 2009].

Ainda assim, existe a comunicação entre ESCMA e ESCHA para que o segundo tenha como fazer busca de informações na rede através de uma *InfoHash* com identificação fornecida pelo primeiro. Uma *InfoHash* é um *hash* consistente que representa um conjunto de informações determinado por um identificador [Klampanos and Jose 2012]. De maneira resumida, um dispositivo ESCHA ao ingressar na rede busca por seu identificador que por sua vez é passado ao seu módulo DHT, denominado DHT ESCHA. A comunicação também existe para a realização de tarefas assíncronas solicitadas pelo ESCI. Entretanto, para que o DHT ESCHA consiga obter as informações de controle de

acesso do seu respectivo identificador, as informações tem de serem publicadas na rede DHT, isso é feito através do módulo DHT do ESCMA por possuir comunicação direta com o banco de dados.

Uma vez lançada tais informações na rede, os ESCHAs que obtiveram suas informações também podem disponibilizá-las na rede para evitar a dependência de um único nó. Além disso, quando informações no banco de dados são alteradas, o módulo DHT do ESCMA, atualiza a *infohash* referente aos mesmos, fazendo com que todos os nós possuidores da mesma, atualizem as informações obtidas. Por outro lado, quando um usuário tenta fazer um acesso ao ESCHA, suas credenciais são verificadas em um banco de dados local formado pelos dados obtidos na rede DHT. Esse evento é publicado na rede DHT para que o módulo DHT do ESCMA obtenha e registre a informação no banco de dados para prover auditoria sobre o sistema.

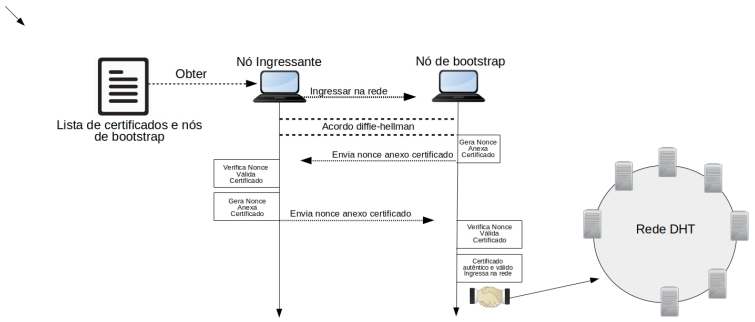


Figura 2. Arquitetura de autenticação

A arquitetura proposta é acoplada a uma infraestrutura de chave pública para prover autenticação na rede DHT por meio de certificados digitais. Para que um nó que deseje ingressar na rede DHT tenha conhecimento de nós que podem servir de nó de bootstrap, é proposta uma estrutura vista na Figura 2 que contém as seguintes informações: Uma lista de nós que podem servir de nó de bootstrap, uma lista de certificados que foram revogados pela autoridade certificadora(CA) e a data de validade da estrutura. Cada nó que quiser ingressar na rede, primeiramente buscará uma estrutura que contém informações de nó de bootstrap e lista de certificados revogados, assume-se que há uma CA previamente configurada. Após tomar conhecimento de nós de bootstrap válidos, o nó ingressante realizará o acordo de chaves *diffie-hellman* visto em [Stallings and Brown 2013] com o nó de bootstrap. Feito isso e verificado a assinatura e validade de ambos os certificados, o nó ingressa na rede. A cada interação entre os nós é verificado a validade dos certificados com base na estrutura descrita anteriormente. Vale ressaltar que todas as comunicações entre os nós é feita com a verificação de desafios.

4. Resultados e Discussões

Para colocar a rede DHT em funcionamento foi utilizado o OpenDHT [Béraud 2015], as simulações foram realizados no ambiente do utilitário de emulação de redes core emulator[Ahrenholz 2010] com o ambiente apresentado na Figura 3. Para validar a arquitetura foram realizados 10 rodadas de testes com arquivos que de 5, 20 e 40 registros

para contrapor o tempo médio de cada um, além da comparação feita sobre o tráfego de rede gerado por eles.

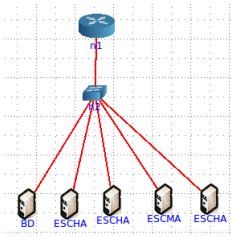
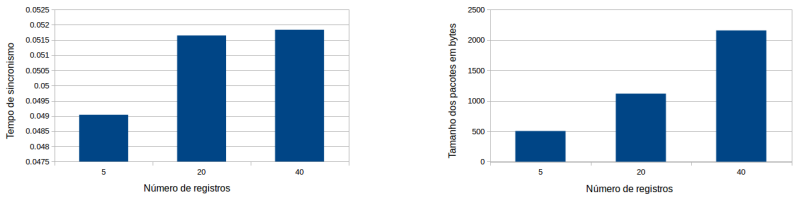


Figura 3. Topologia da simulação

Para obter o tempo de processamento de cada um dos pacotes com o diferentes números de registros, foi adicionado junto à função de pesquisa da informação, um temporizador que iniciava no momento em que encontrou a informação até o tempo de deserialização dos dados. O resultado visto abaixo na Figura 4(a) demonstra que a diferença de tempo de sincronismo de arquivos com 5 e 20 registros é de apenas 0.025 segundos. Entretanto, para os arquivos com 20 e 40 registros a diferença no tempo de sincronismo é pouco perceptível, ainda assim, o tempo de sincronismo em todos os casos é consideravelmente baixo.



(a) Tempo de processamento (b) Tamanho dos pacotes em bytes

Figura 4. Resultados do tempo de sincronismo e quantidade de tráfego gerado para diferentes quantidades de registros

Já na Figura 4(b) é apresentado a quantidade de tráfego gerado em bytes para arquivos com diferentes tamanhos obtido através do monitoramento do tráfego de rede com a ferramenta *wireshark*. Existe uma diferença considerável da quantidade de tráfego entre os arquivos com 5 e 20 registros, entretanto, a diferença entre a quantidade de tráfego gerado de 20 e 40 registros quase se duplica do segundo em relação ao primeiro. Isso demonstra que não mantém uma linearidade da quantidade de registro em relação a quantidade de tráfego.

5. Considerações Finais e Trabalhos Futuros

Conforme os resultados analisados na seção anterior, a arquitetura demonstrou um período curto para o sincronismo, portanto, é viável a utilização do mesmo em um sistema para controle de acesso como o ESC onde informações têm de serem atualizadas em um intevalo de tempo razoável. Vale ressaltar que o tráfego gerado é aceitável pelo fato de que o tamanho máximo de *InfoHash* é de 512 kilobytes. Com isso, muitas informações podem ser adicionadas aos registros, fortalecendo o uso de tal arquitetura. Por fim, como trabalho futuro fica o prosseguimento da validação dos mecanismo de segurança aclopado a arquitetura proposta.

Referências

Ahrenholz, J. (2010). Comparison of core network emulation platforms. In *Military Communications Conference, 2010-MILCOM 2010*, pages 166–171. IEEE.

Avramidis, A., Kotzanikolaou, P., Douligeris, C., and Burmester, M. (2012). Chord-pki: A distributed trust infrastructure based on p2p networks. *Computer Networks*, 56(1):378 – 398.

Béraud, A. (2015). Welcome to the opendht wiki! Disponível em `https://github.com/savoirfairelinux/opendht/wiki`, acessado em 28/09/2018.

Guangmin, L. (2009). An improved kademlia routing algorithm for p2p network. In *2009 International Conference on New Trends in Information and Service Science*, pages 63–66.

Haowei, J. and Yubo, T. (2010). Research in p2p-pki trust model. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 5, pages 114–117.

Klampanos, I. A. and Jose, J. M. (2012). Searching in peer-to-peer networks. *Computer Science Review*, 6(4):161–183.

Kubler, S., Främling, K., and Derigent, W. (2015). P2p data synchronization for product lifecycle management. *Computers in Industry*, 66:82–98.

Kwon, H., Koh, S., Nah, J., and Jang, J. (2008). The secure routing mechanism for dht-based overlay network. 2:1300–1303.

Liu, C.-M. and Lai, C.-C. (2017). A heuristic data update mechanism in unstructured mobile p2p systems. *Ad Hoc Networks*, 58:138–149.

Martinez-Yelmo, I., Cuevas, R., Guerrero, C., and Mauthe, A. (2008). Routing performance in a hierarchical dht-based overlay network. In *Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on*, pages 508–515. IEEE.

Mehta, H. K., Kanungo, P., and Chandwani, M. (2011). Distributed database caching for web applications and web services. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, pages 510–515. ACM.

Pedrozo, W. F., Rodrigues, A. S., Alves, B. D. S., Del Rio, L. S., Rosa, C. L., and Rizzetti, T. A. (2018). Uma arquitetura para sensoriamento e tratamento de eventos voltada à área de segurança para controle e rastreio de usuários em ambientes físicos. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, 2(1).

Stallings, W. and Brown, L. (2013). *Segurança de Computadores: Princípios e Práticas*. ELSEVIER, 2a edition.

Takeda, A., Hashimoto, K., Kitagata, G., Zabir, S. M. S., Kinoshita, T., and Shiratori, N. (2008). A new authentication method with distributed hash table for p2p network. In *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, pages 483–488.