

Escalabilidade, Autonomia e Segurança em Redes Peer-to-Peer: repensado a P2PSL

Giovani Facchini

Orientador: Marinho Pilla Barcellos

Ciência da Computação
Universidade do Vale do Rio dos Sinos (UNISINOS)

Sumário

- Introdução
 - Motivação
 - Contextualização
 - Objetivos
- Fundamentos
 - Redes P2P
 - Gerência de Chaves
 - P2PSL
- Análise da P2PSL e Propostas de Melhorias
 - Arquitetura
 - Escalabilidade
 - Segurança
- Modificações Implementadas
- Conclusões e Trabalhos Futuros

2

Introdução

- Motivação
 - Grande crescimento das redes P2P
 - Vasta gama de aplicações existentes
 - Aumentar a escalabilidade e segurança das aplicações
 - Prover ferramentas que auxiliem o desenvolvimento de aplicações P2P seguras
- Contextualização
 - P2PSL foi criada para diminuir o impacto das decisões de implantação de segurança no código da aplicação
 - Visa promover fácil implantação de mecanismos de segurança em aplicações P2P de maneira modular e gradual

3

Introdução

- Objetivos
 - Analisar a P2PSL:
 - Arquitetura
 - Escalabilidade
 - Aspectos de Segurança e Vulnerabilidades
 - Propor soluções para problemas encontrados
 - Implementar uma prova-de-conceito
 - Avaliar experimentalmente a implementação

4

Sumário

- Introdução
 - Motivação
 - Contextualização
 - Objetivos
- Fundamentos
 - Redes P2P
 - Gerência de Chaves
 - P2PSL
- Análise da P2PSL e Propostas de Melhorias
 - Arquitetura
 - Escalabilidade
 - Segurança
- Modificações Implementadas
- Conclusões e Trabalhos Futuros

5

Redes Peer-to-Peer

- Conceituação
 - Não existe consenso quanto a uma definição formal
 - "Nós são equivalentes em funcionalidade e tarefas que executam"
 - "classe de aplicações que tira vantagens de recursos (armazenamento, ciclos de CPU, conteúdo, presença humana) disponível nas bordas da internet"
 - Rede de sobreposição à rede real (*overlay*)
- Aplicações
 - Compartilhamento de arquivos (mais usada)
 - Comunicação entre usuários (Instant Messaging)
 - Computação distribuída (OurGrid)
 - Armazenamento distribuído (OceanStore)
 - Jogos On-Line (carta e tabuleiro)

6

Redes Peer-to-Peer

- Arquitetura
 - **Não-Estruturada:** nó entra na rede em uma posição aleatória no grafo de conexões. Não existe mapeamento definido entre objetos e nós
 - *Centralizada:* um servidor é responsável pelo bootstrap e pelas tarefas de gerência e busca (Napster)
 - *Parcialmente Centralizada:* existe uma hierarquia de nós da rede, onde alguns atuam como servidor (Kazaa)
 - *Descentralizada:* todas as tarefas de busca e gerência da rede são de responsabilidade dos nós presentes (Gnutella)
 - **Estruturada:** mensagens são roteadas através do *overlay*
 - Um nó tem a posição no grafo determinada por um algoritmo baseado em seu identificador
 - Cada objeto na rede é mapeado para um identificador. Este objeto é associado a um ou mais nós (que ficam com a responsabilidade pelo objeto)

7

Redes Peer-to-Peer

- Fundamentos de Segurança
 - Confidencialidade
 - Autenticação
 - Integridade
 - Não-Repúdio
 - Autorização
 - Auditoria
 - Anonimidade
 - Reputação
 - Disponibilidade
 - Negabilidade

8

Gerência de Chaves

- Distribuição de Chaves Secretas
 - Criptografia por senha
 - Key Encryption Key
 - Centro de Distribuição de Chaves (KDC)
- Distribuição de Chaves Públicas
 - Anuncio Público / Diretório Público
 - Autoridade de Chave Pública
 - Autoridade Certificadoras
 - PGP – Rede de Confianças

9

Gerência de Chaves

- Criptografia assimétrica (chave pública) é lenta se comparada a criptografia simétrica
- Criptografia simétrica tem grande problema na logística de distribuição de chaves e não garante não-repúdio
- Solução:
 - Utilizar as duas técnicas combinadas
 - Utilização a chave pública (assimétrica) para trocar uma chave secreta (simétrica)
 - A chave secreta é denominada chave de sessão e é gerada aleatoriamente
 - A chave pública é sempre a mesma

10

Peer-to-Peer Security Layer (P2PSL)

- Camada de segurança
- Provê:
 - **Encapsulamento:** expõe uma interface simples sem que o desenvolvedor tenha que inserir complexidade de implementação para lidar com segurança
 - **Modularidade:** É possível escolher quais requisitos serão utilizados sem obrigação de usar todos
 - **Implantação Gradativa:** Permite que o sistema possa ir se modificando aos poucos sem parar tudo para uma migração
 - **Re-configurabilidade:** Permite que os requisitos de segurança possam ser trocados a qualquer momento

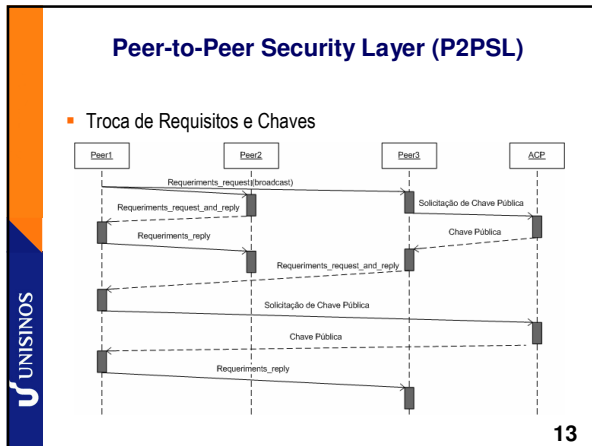
11

Peer-to-Peer Security Layer (P2PSL)

- Funcionamento

The diagram illustrates the P2PSL architecture. At the top is the **JXTA Application**, which has methods `receiveMessage()` and `broadcastOnPropagatePeer()`. Below it is the **SecurePeer** component, which has methods `send(message())` and `broadcastOnPropagatePeer()`. To the right are **PgpSignature** and **PgpEncryption** components, with methods `verifyIncomingMessage()` and `adjustOutgoingMessage()`. At the bottom is the **EZMinimalPeer (JXTA/JAL)** component, which has methods `receiveMessage()` and `broadcastOnPropagatePeer()`. A **XML File** is shown interacting with the **SecurePeer** component. On the far right, a box labeled **Peer 1** is shown with a **PGPS** icon.
- Módulos Implementados
 - Assinatura e Criptografia PGP
 - Controle de Acesso (RBAC)
 - Auditoria (LOG)

12

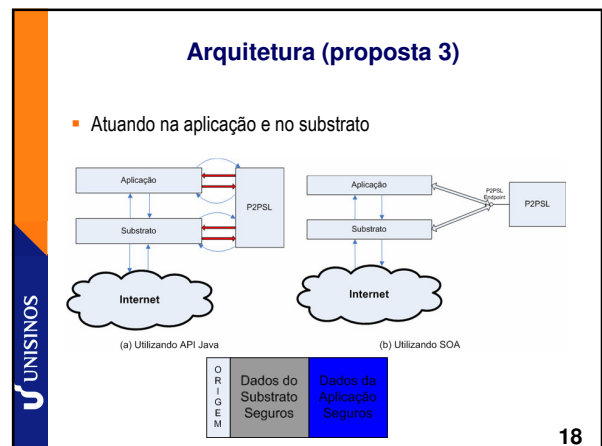
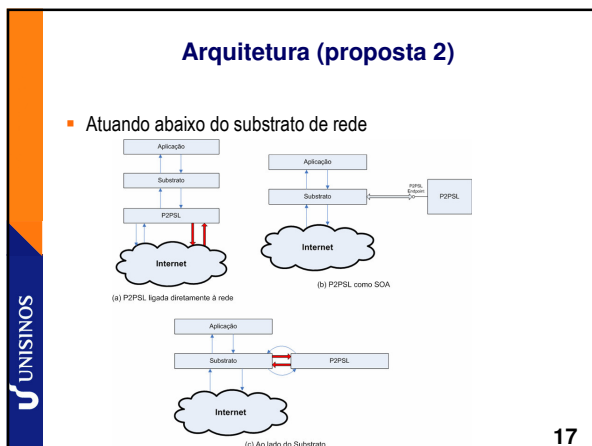
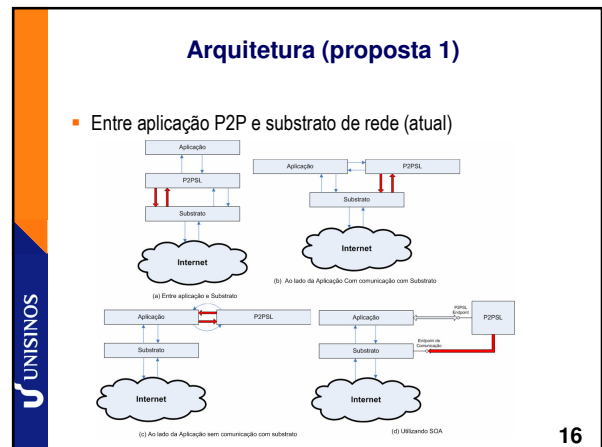


Sumário

- Introdução
 - Motivação
 - Contextualização
 - Objetivos
- Fundamentos
 - Redes P2P
 - Gerência de Chaves
 - P2PSL
- Análise da P2PSL e Propostas de Melhorias
 - Arquitetura
 - Escalabilidade
 - Segurança
- Modificações Implementadas
- Conclusões e Trabalhos Futuros

14

- ### Análise da Arquitetura
- Vantagens
 - Permite implementação simples sem necessidade de preocupação com comunicação
 - Integra-se facilmente em ambientes usando JXTA
 - Garante os requisitos de segurança para os dados da aplicação P2P
 - Desvantagens
 - A camada de comunicação PRECISA ser JXTA
 - A utilização da camada de abstração JAL limita a criação de algoritmos (como roteamento)
 - Problemas advindos da JXTA serão difíceis de debuggar
 - Requisitos de segurança não podem ser aplicados sobre o substrato de rede (impede a adoção de requisitos de segurança para mensagens de atualização de rotas, mensagens roteadas, etc.)
- 15



Análise de Escalabilidade

- Problemas
 - Quando um nó não conhece a chave de outro a Autoridade de Chave Pública é consultada (gargalo)
 - Na ACP, as chaves públicas dos usuários podem ser sabotadas
 - Quando um nó entra no sistema, um *broadcast* é feito requisitando as necessidades de segurança dos outros nós presentes (não escala)
- Proposta de Melhoria
 - Mudar o modelo de troca de chaves para utilização de Autoridade Certificadora, pois a necessidade de interação dos nós com a AC é menor que no modelo atual
 - Fazer a requisição das necessidades de segurança somente para os nós de interesse e não para todos os nós (sob demanda)

19

Análise de Segurança

- Negação de serviço
 - *DoS (nível de rede)*: Não existe contra-medida para P2PSL
 - *DoS (nível de aplicação) [Grande quantidade de consultas]*: Criação de um módulo que monitorasse atividade dos nós limitando, de forma parametrizada, a quantidade de recursos consumidos
 - *Nó responde a busca, mas não entrega conteúdo*: Colocar manualmente o nó em um perfil "suspeito". Para solução "automática", incorporar um sistema de reputação
 - *Ataque de nó lento*: Módulo de autenticidade evita a modificação de mensagens

20

Análise de Segurança

- Roteamento
 - *Eclipse*: Utilizar informação do certificado para gerar o identificador. Dessa forma, atacantes não podem escolher qual intervalo de identificadores podem utilizar para ataque
 - *Falsificação de mensagens de atualização do overlay / Desvio de rota e envenenamento das tabelas*: Para não sofrer ataque de falsificação, o módulo de autenticidade pode ser usado. O substrato de rede, antes de atualizar uma tabela, precisa verificar se o caminho é atingível. Caso contrário, coloca o nó que tentou atualizar a rota em um perfil de lista negra
 - Obs: Essas propostas ficam mais a cargo do substrato, não da P2PSL. Entretanto o substrato irá utilizar funcionalidades providas pela P2PSL para implementar as soluções

21

Análise de Segurança

- Autenticidade
 - *Apesar de garantir a origem digital de uma mensagem, não se tem informação sobre quem detém aquela identidade*: Certificados digitais diminuem esse problema, pois atrela uma identidade a um e-mail (mas e-mails podem ser forjados)
 - *Um nó pode gerar várias chaves e entrar na rede com várias identidades*: Utilizar certificados de alta confiabilidade
 - Essa solução pode tornar a adoção da rede complicada, pois necessita comprovação de identidade fisicamente na AC (*tradeoff*)

22

Análise de Segurança

- Reputação e Confiança
 - *Não existem métodos automáticos de reputação para o sistema, mas um nó pode reputar outro localmente através dos perfis*: Implementar um módulo que fizesse a coleta e raqueamento de um nó baseado em suas informações locais e as informações dos nós conhecidos
 - *Utilizar Sybil para enganar um nó*: Pode-se utilizar certificados de alta confiabilidade
 - *Conluio*
 - Obs: As propostas já apresentadas na literatura poderiam ser incorporadas na P2PSL na tentativa de minimizar o problema de reputação

23

Análise de Segurança

- Autorização
 - *Regras de autorização estão mapeadas para os nomes dos nós (nomes pode ser falsificados)*: O módulo RBAC poderia atuar em conjunto com módulo de autenticidade, indicando para o módulo de autorização qual a chave pública do nó que tem acesso a um recurso

24

Análise de Segurança

- **Anonimidade**
 - P2PSL não endereça essas características:
 - Deixando a P2PSL com controle total sobre a rede, seria possível adicionar a funcionalidade de roteador cebola a ela
 - A P2PSL gerenciaria quais nós fariam o papel de roteadores intermediários e proxies

25

Análise de Segurança

- **Poluição**
 - *Conteúdo é íntegro, mas é incorreto, ou seja, não corresponde ao que um usuário espera:*
 - De maneira similar à proposta de módulos de reputação, um módulo que coletasse informação sobre conteúdo poderia ser implementado
 - Esse módulo precisaria interagir com o usuário da aplicação para que pudesse identificar conteúdo já visto como poluído ou correto (repassando essa informação para os nós conhecidos).

26

Sumário

- **Introdução**
 - Motivação
 - Contextualização
 - Objetivos
- **Fundamentos**
 - Redes P2P
 - Gerência de Chaves
 - P2PSL
- **Análise da P2PSL e Propostas de Melhorias**
 - Arquitetura
 - Escalabilidade
 - Segurança
- **Modificações Implementadas**
- **Conclusões e Trabalhos Futuros**

27

Modificações Implementadas

- **Arquitetura**
 - Foi implementado o modelo em que a P2PSL pode atuar aplicando requisitos de segurança tanto para a aplicação P2P quanto para o substrato de rede
 - Assinatura dos métodos da classe SecurePeer mudaram para:
 - *Serializable prepareSendMessage(String, Serializable):*
 - *Serializable obtainReceivedMessage(String, Serializable):*

28

Modificações Implementadas

- **Troca de Chaves**
 - Foi implementado gerenciamento de certificados utilizando a *keystore* JKS da SUN.
 - Os arquivos de configuração foram modificados para incluir os certificados

29

Modificações Implementadas

- **Troca de Requisitos**
 - O protocolo de troca de requisitos foi modificado
 - Um nó deve contactar apenas aquele no qual possui interesse para troca de requisitos

30

Modificações Implementadas

- **Módulos**
 - O módulo de criptografia assimétrica foi modificado para suportar certificados
 - Para cada mensagem, uma chave de sessão secreta é gerada
 - O módulo de assinatura suporta o uso de certificado para realizar o procedimento.
- **Aplicação exemplo**
 - Faz a troca de certificados digitais usando um substrato de rede simplificado para demonstrar o funcionamento do protocolo de troca de chaves

UNISINOS **31**

Modificações Implementadas

- **Análise de Desempenho no Recebimento**

UNISINOS **32**

Modificações Implementadas

- **Análise de Desempenho no Envio**

UNISINOS **33**

Sumário

- **Introdução**
 - Motivação
 - Contextualização
 - Objetivos
- **Fundamentos**
 - Redes P2P
 - Gerência de Chaves
 - P2PSL
- **Análise da P2PSL e Propostas de Melhorias**
 - Arquitetura
 - Escalabilidade
 - Segurança
- **Modificações Implementadas**
- **Conclusões e Trabalhos Futuros**

UNISINOS **34**

Conclusões

- Esse trabalho identificou vulnerabilidades da P2PSL
- Propôs alternativas para melhoria de:
 - Arquitetura
 - Escalabilidade
 - Segurança
- Implementou como prova-de-conceito modificações para melhorar:
 - Arquitetura
 - Flexibilidade de Uso
 - Possibilidade de inserir segurança no substrato
 - Escalabilidade
 - Troca de chaves
 - Troca de Requisitos
- Analisou o desempenho dos módulos

UNISINOS **35**

Trabalhos Futuros

- Implantar a P2PSL juntamente com algum aplicativo P2P existente
- Investigar profundamente os aspectos de segurança e as sugestões especuladas neste trabalho
- Colocar em prática mecanismos que diminuam as vulnerabilidades expostas neste trabalho
- Analisar experimentalmente a P2PSL utilizando tráfego elástico e não-elástico

UNISINOS **36**

Escalabilidade, Autonomia e Segurança em
Redes Peer-to-Peer: repensado a P2PSL

Obrigado!

Giovani Facchini
Perguntas???

Orientador: Marinho Pilla Barcellos

Ciência da Computação
Universidade do Vale do Rio dos Sinos (UNISINOS)