Armazenamento Seguro de Credenciais e Atributos de Usuários em Federação de *Clouds*

Autores: Luciano Barreto[1], Leomar Scheunemann[1] , Joni da Silva Fraga[1], Frank Siqueira[2]

[1]Departamento de Automação e Sistemas – (DAS)
[2]Departamento de Informática e Estatística – (INE)
Universidade Federal de Santa Catarina

Aluno: André Luís Trigo Fernandes

Federação de *clouds*

- Agrupamento de provedores de serviços de *cloud* em associações;
- Uso de serviços de diversos provedores de clouds para atender a demanda de clientes e os SLAs (Service-Level Agreement) acordados;
- Autoridades independentes e confiáveis de autenticação, chamadas de Provedores de Identidades ou *IdPs* .

Objetivo

- Apresentar uma abordagem que faz uso de provedores de *cloud* para armazenar as informações de usuário (credenciais e atributos) ;

- Flexibilidade do acesso às informações permitindo que usuários possam fazer os seus *logins* a partir de um *IdP* de uma federação de *clouds* e ter acessos a recursos em diferentes partes desta federação.

Caracterização de Federação de *Clouds*

- O agrupamento de provedores de *clouds* através de redes de confiança formadas com o objetivo de atender um vasto número de usuários coloca também grandes desafios na autenticação e autorização destes usuários junto aos provedores nestas federações.

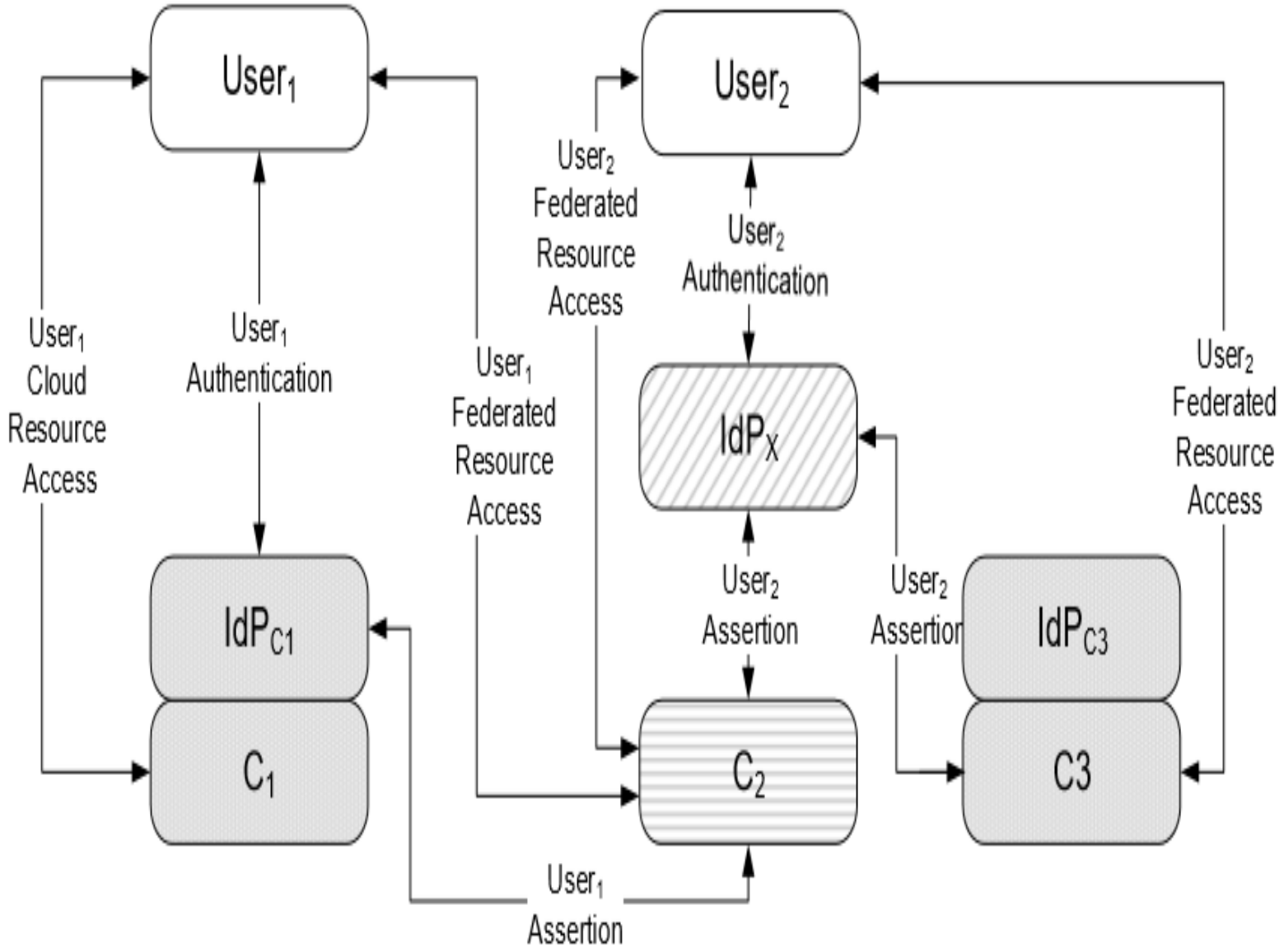


Figura 1 – Autenticação e Autorização em Federação de Clouds

Armazenamento de credenciais a atributos de Usuário

- Existência de diversos *IdPs;*
- Não manter os dados localmente;
- Dispor estas informações, de forma segura em uma base de dados construída sobre recursos distribuídos pela federação.

Armazenamento de credenciais a atributos de Usuário

- As chaves criptográficas usadas nas encriptações de armazenamento seguro em *clouds* são protegidas normalmente com o uso de técnicas de compartilhamento de segredos [Schoenmakers 1999; Shamir 1979];

- (Information Dispersal Algorithms: IDA [Rabin 1989].

Armazenamento de credenciais a atributos de Usuário

- Compartilhamento de segredo (**S**);
- **S** dá origem a n *shadows* (partes);
- Cada parte não revela o segredo e a informação só pode ser reconstruída a partir da obtenção de um conjunto mínimo de t partes ($0 < t \leq n$).

Armazenamento de credenciais a atributos de Usuário

- Para lidar com cenários onde as partes estão sob a ação de entidades maliciosas que agem ativamente contra os protocolos de restauração dos segredos, foram criados os mecanismos de compartilhamento de segredo verificável [Schoenmakers 1999], em que cada fragmento de um segredo fornecido a um combinador pode ter sua validade verificada.

Armazenamento de credenciais a atributos de Usuário

- Compartilhamento de segredo sobre as informações de usuário que decompõe as mesmas em *n* partes;
- Uma parte por cada uma das *n* entre as *m clouds* do sistema;
- Informações recuperadas com no mínimo *t* partes armazenadas em *n* das *m clouds* do sistema, onde $0 < t < n < m$.

Arquitetura Proposta

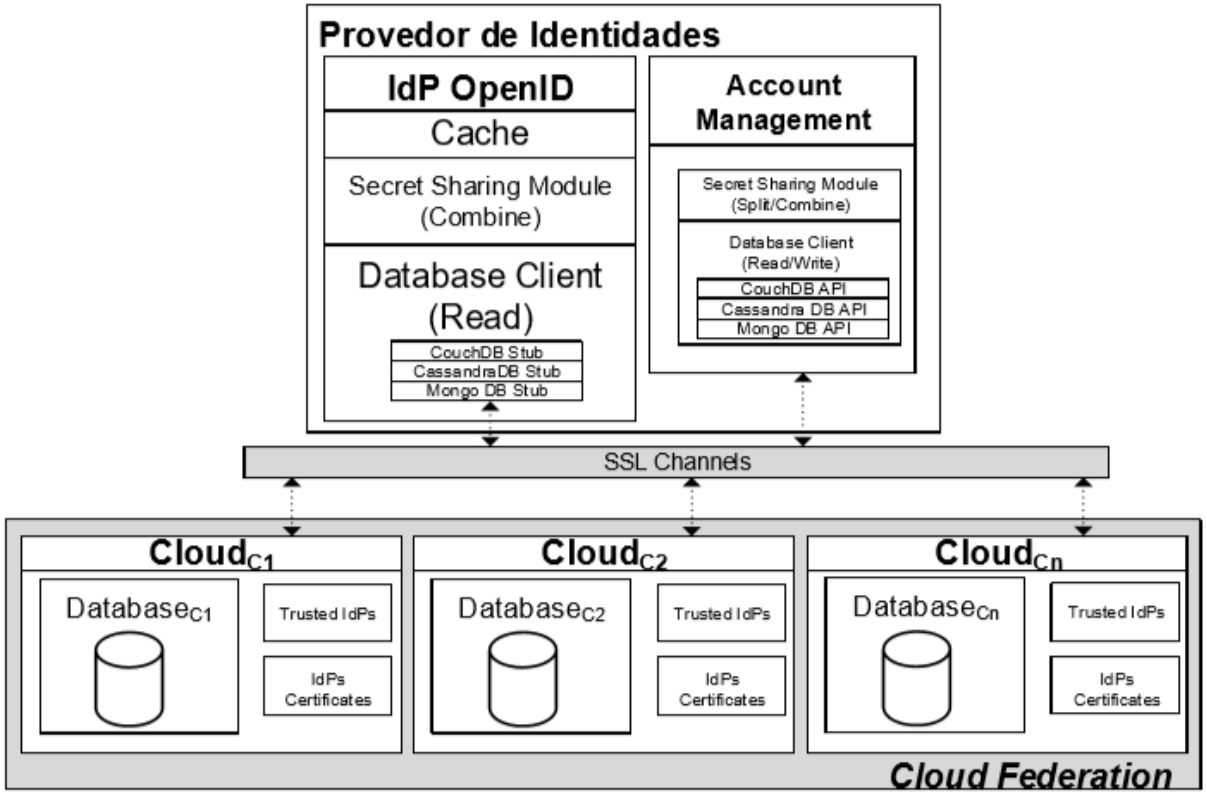


Figura 2 – Organização do Sistema de Autenticação

Protocolos de Abordagem

- Duas possibilidades de criação de contas;
- a interface *Account Management* é acionada por pessoal credenciado da *cloud*, necessária presença física usuário ou;
- Usuário faz o seu registro através da Internet.

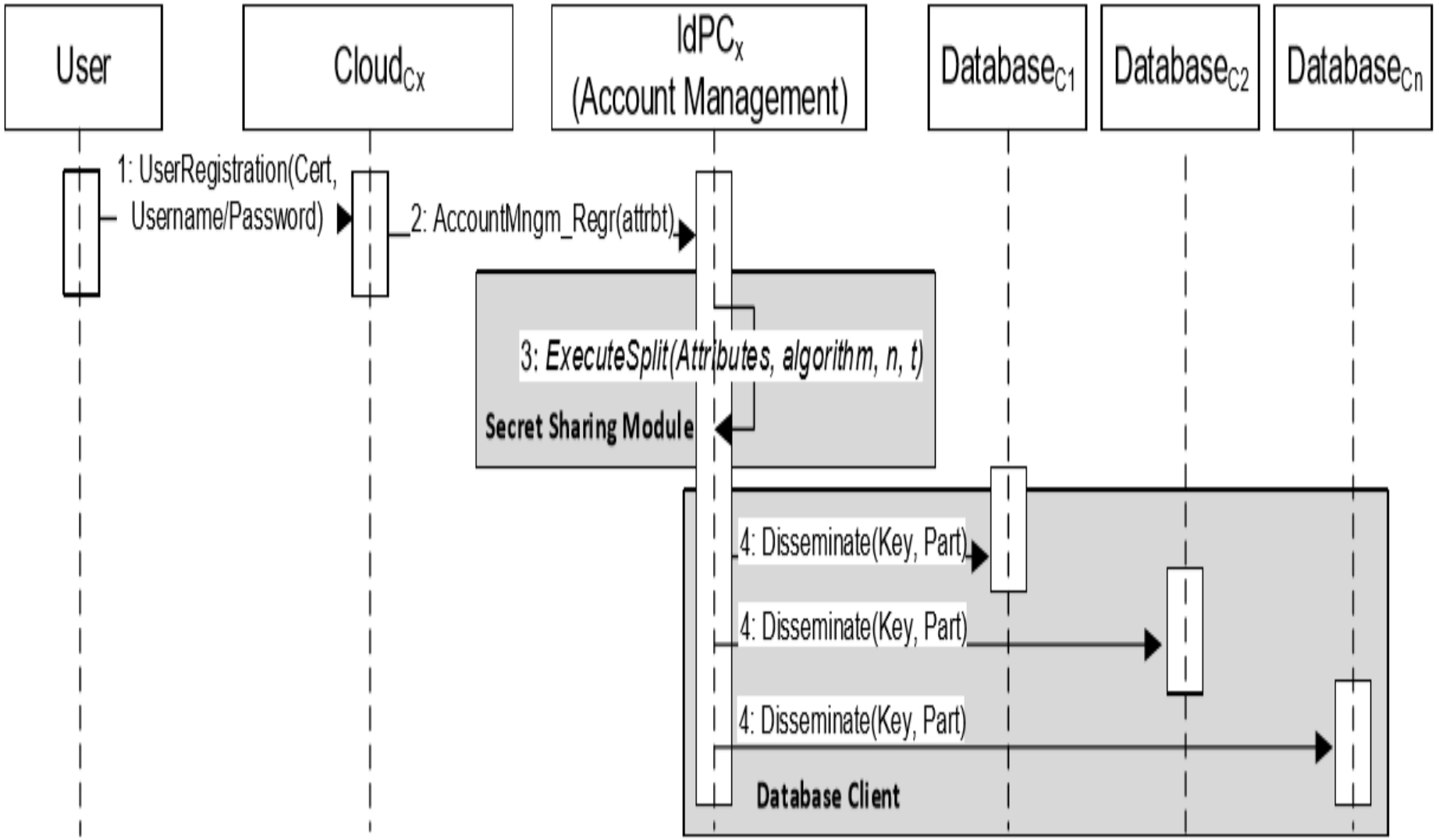


Figura 3 – Registro de Usuários

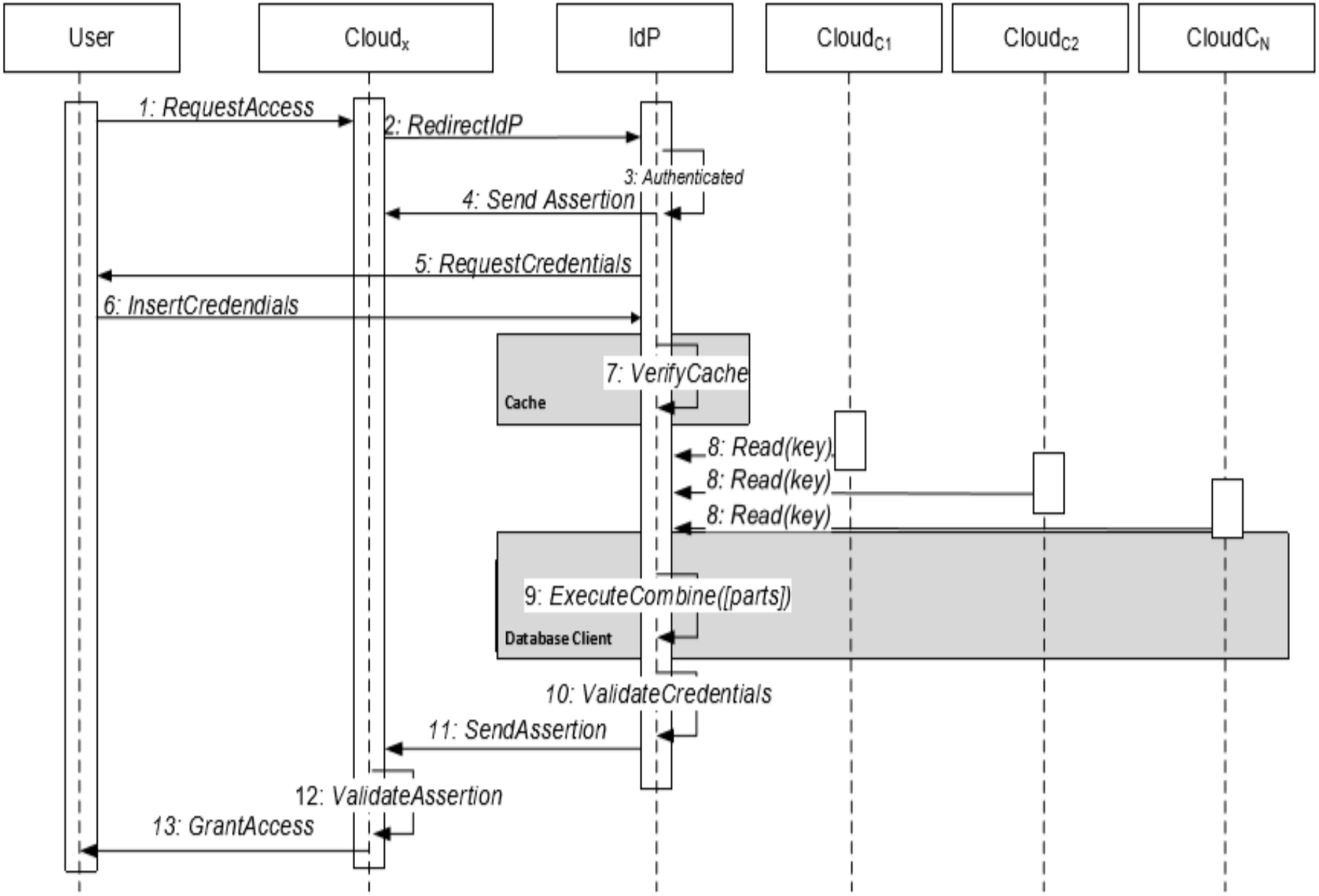


Figura 4 – Autenticação de Usuário

Protótipo e Resultados

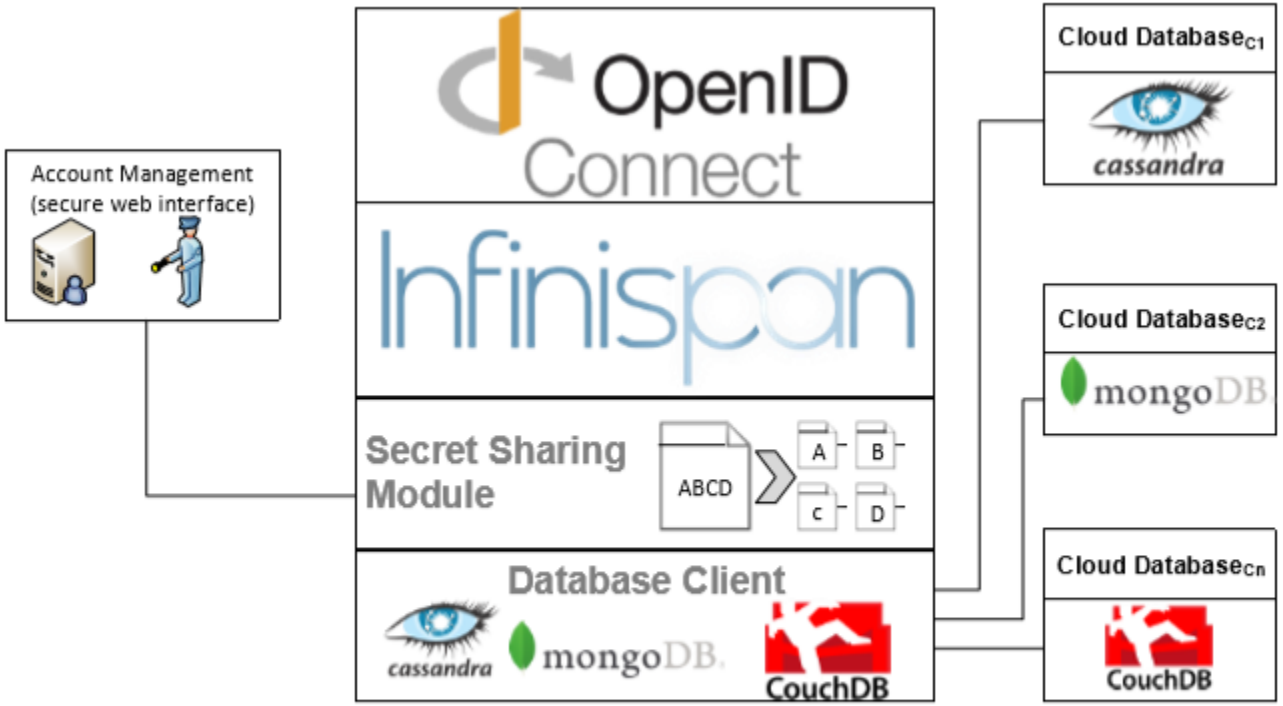


Figura 5 – Tecnologias utilizadas

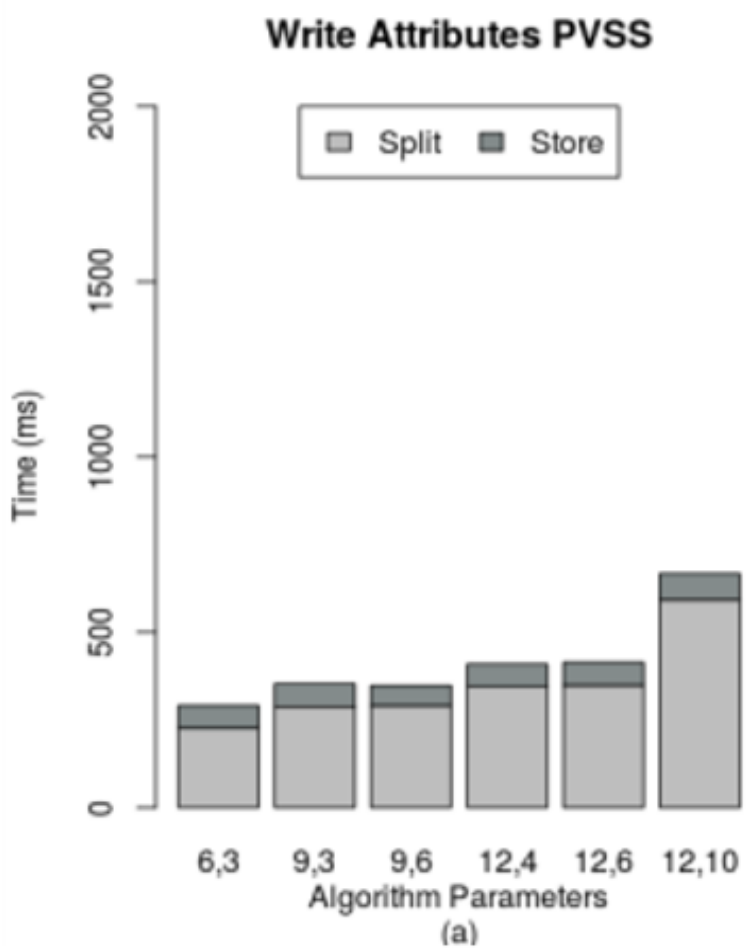
Protótipo e Resultados

- *Secret sharing;*
- *Shamir;*
- *PVSS;*

Testes e Resultados

- 1000 registros usuários;
- Credenciais: Usuário e Senha;
- Atributos: nome, sobrenome, endereço, data de nascimento, profissão, etc.;
- Diferentes Paramêtros *n,t.*

Testes Tempo de Registro de Usuários



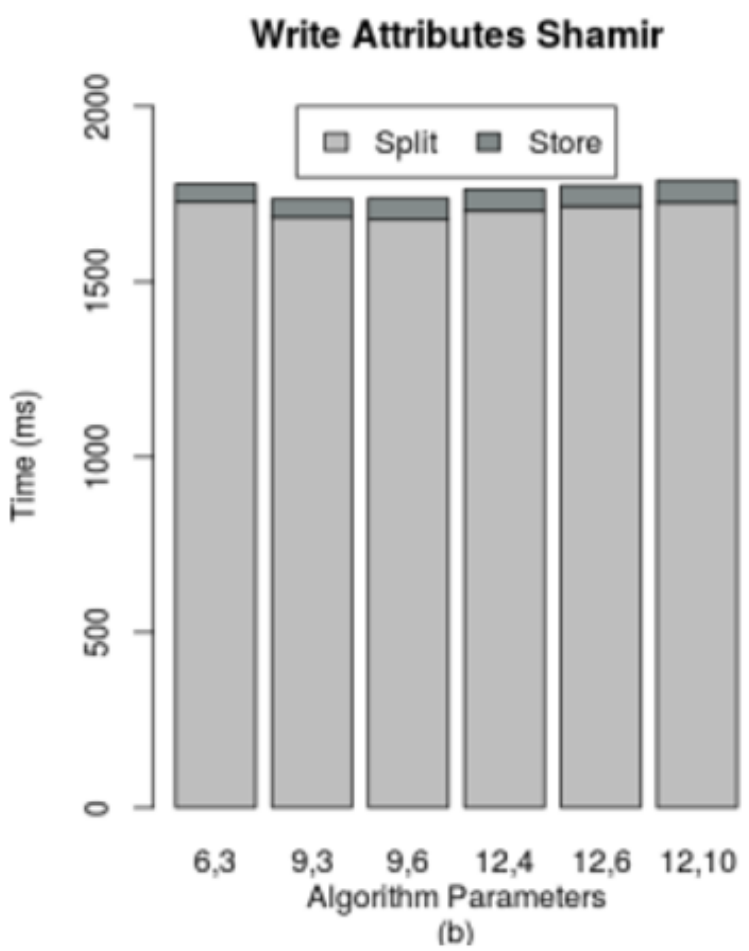


Figura 6 – Tempos de Registro

Teste Tempo de Autenticação

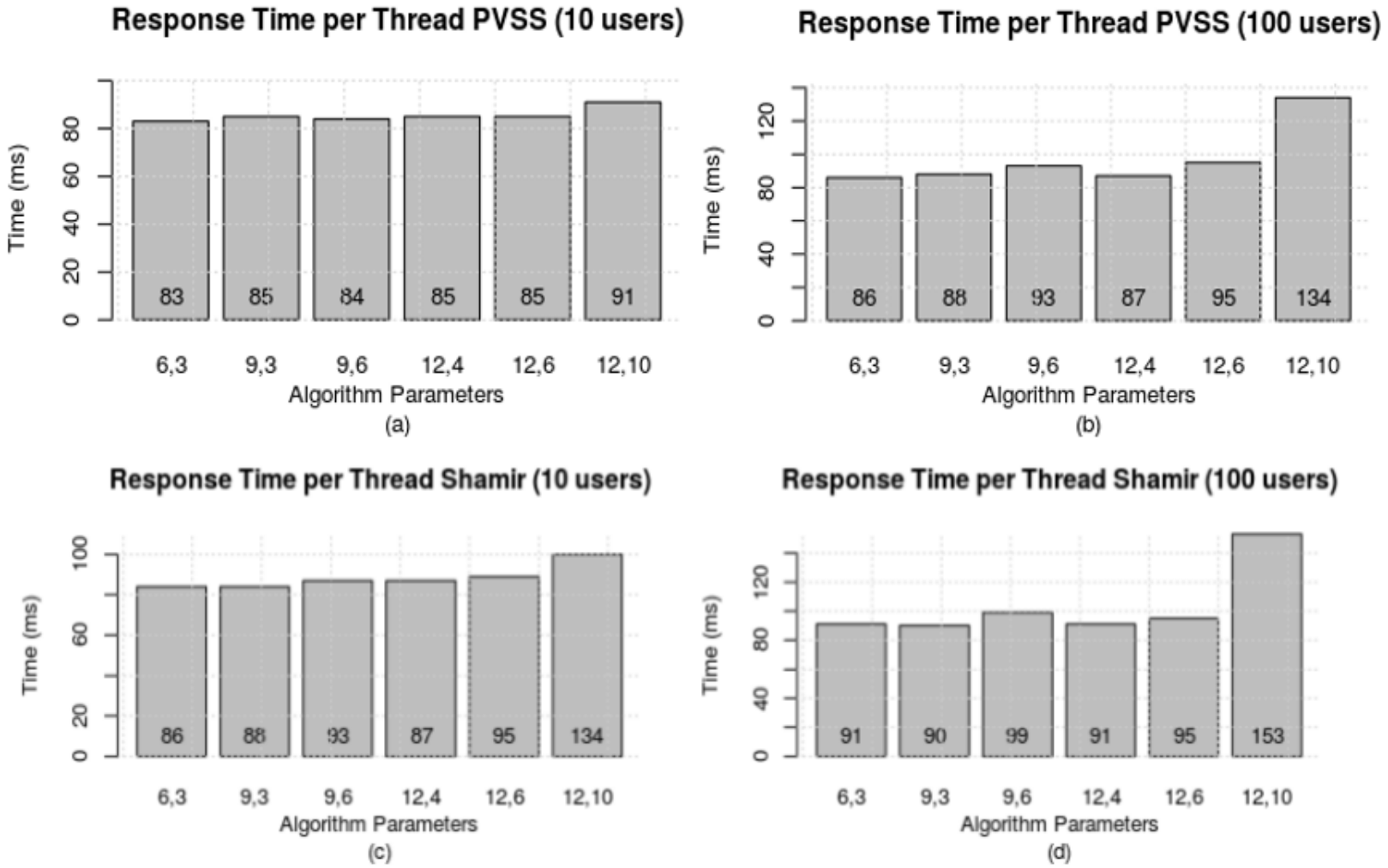


Figura 7 – Tempo de Autenticação

Teste de Vazão de Autenticações

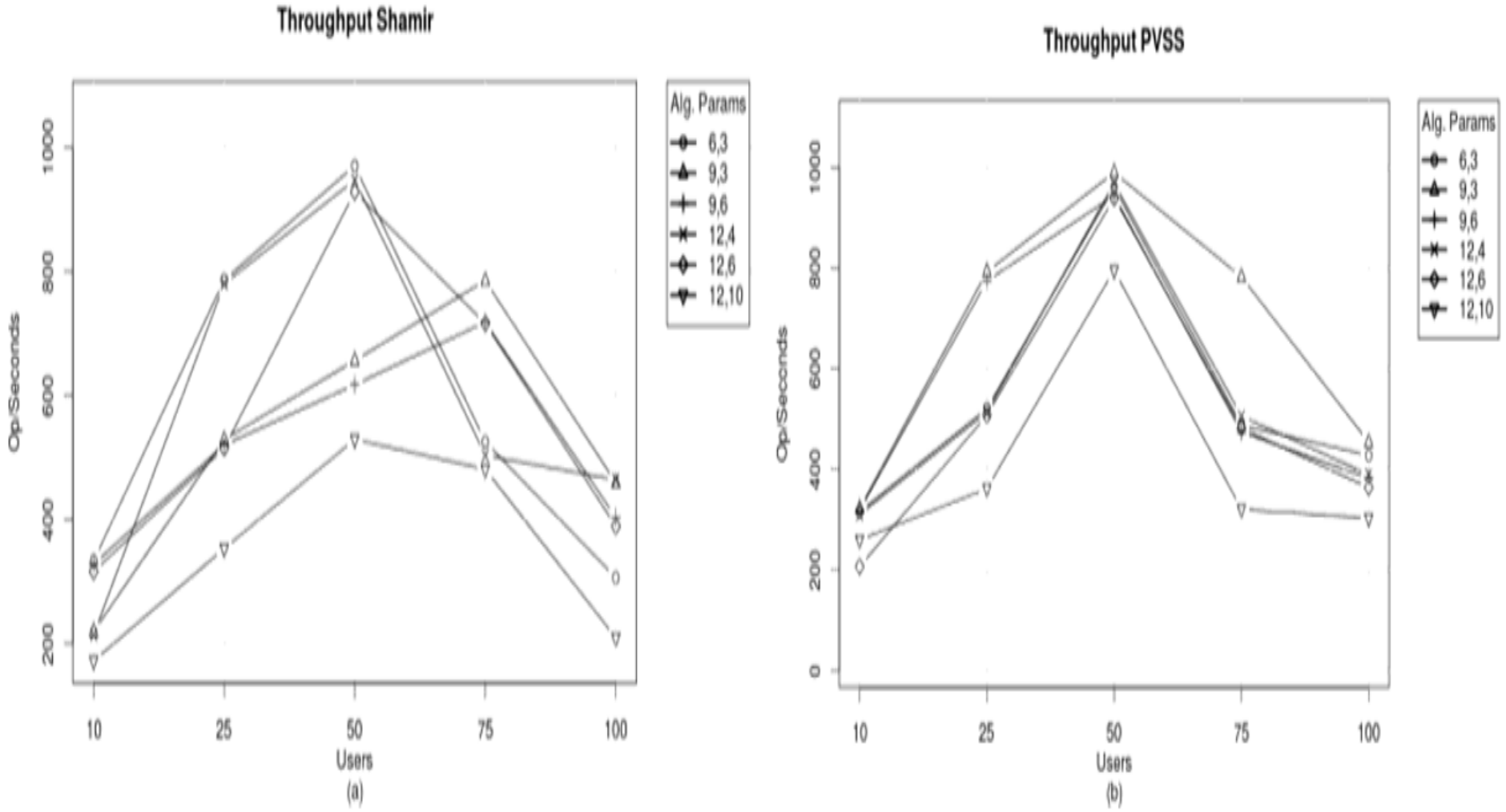


Figura 8 – Vazão de autenticações (operações por segundo)

Testes com o uso de *cache*

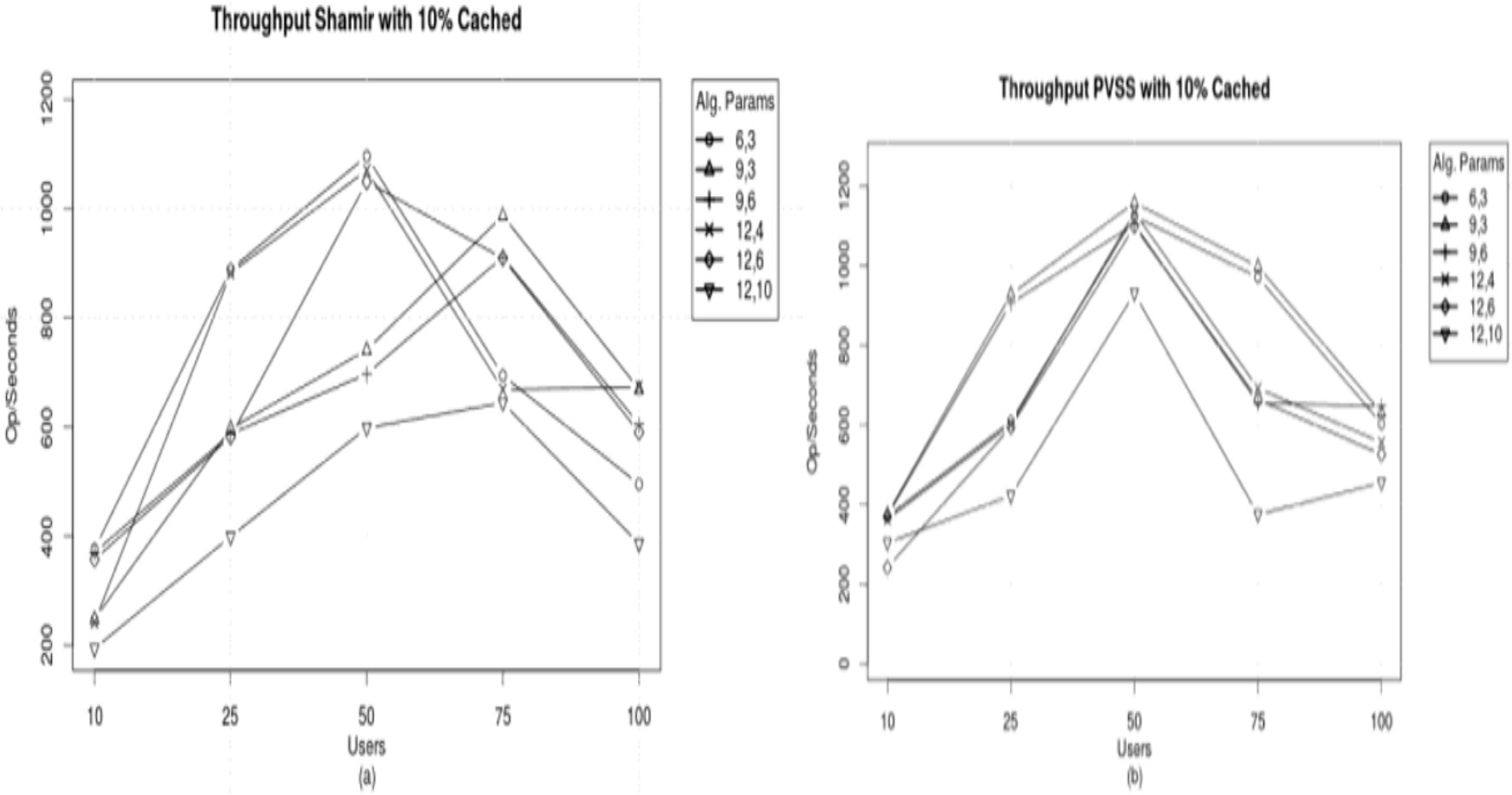


Figura 9 – Vazão com o uso de *cache* (operações por segundo)

Vazão dos algoritmos com intrusões

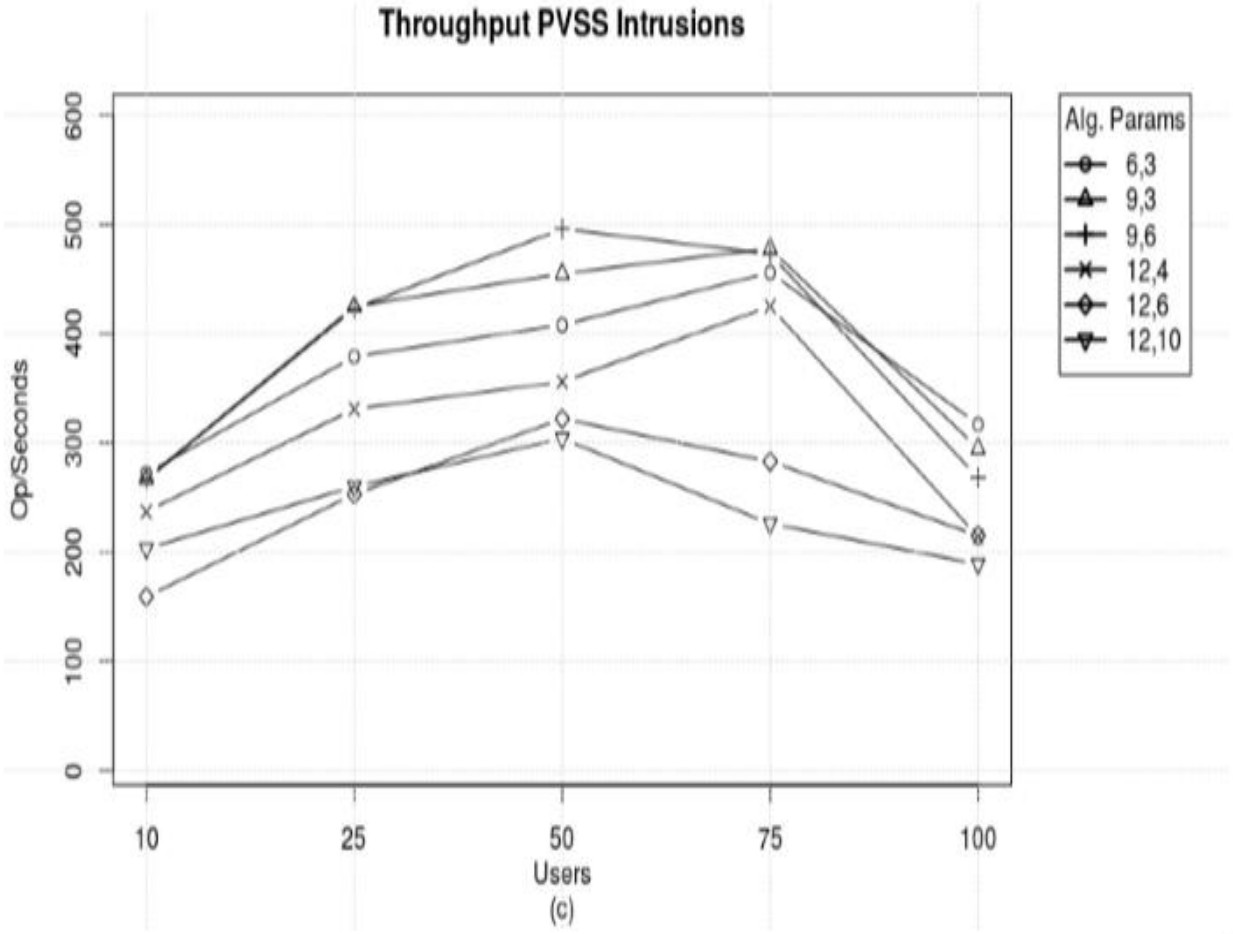


Figura 10 – Vazão com intrusões utilizando PVSS (operações por segundo)

Vazão dos algoritmos com intrusões

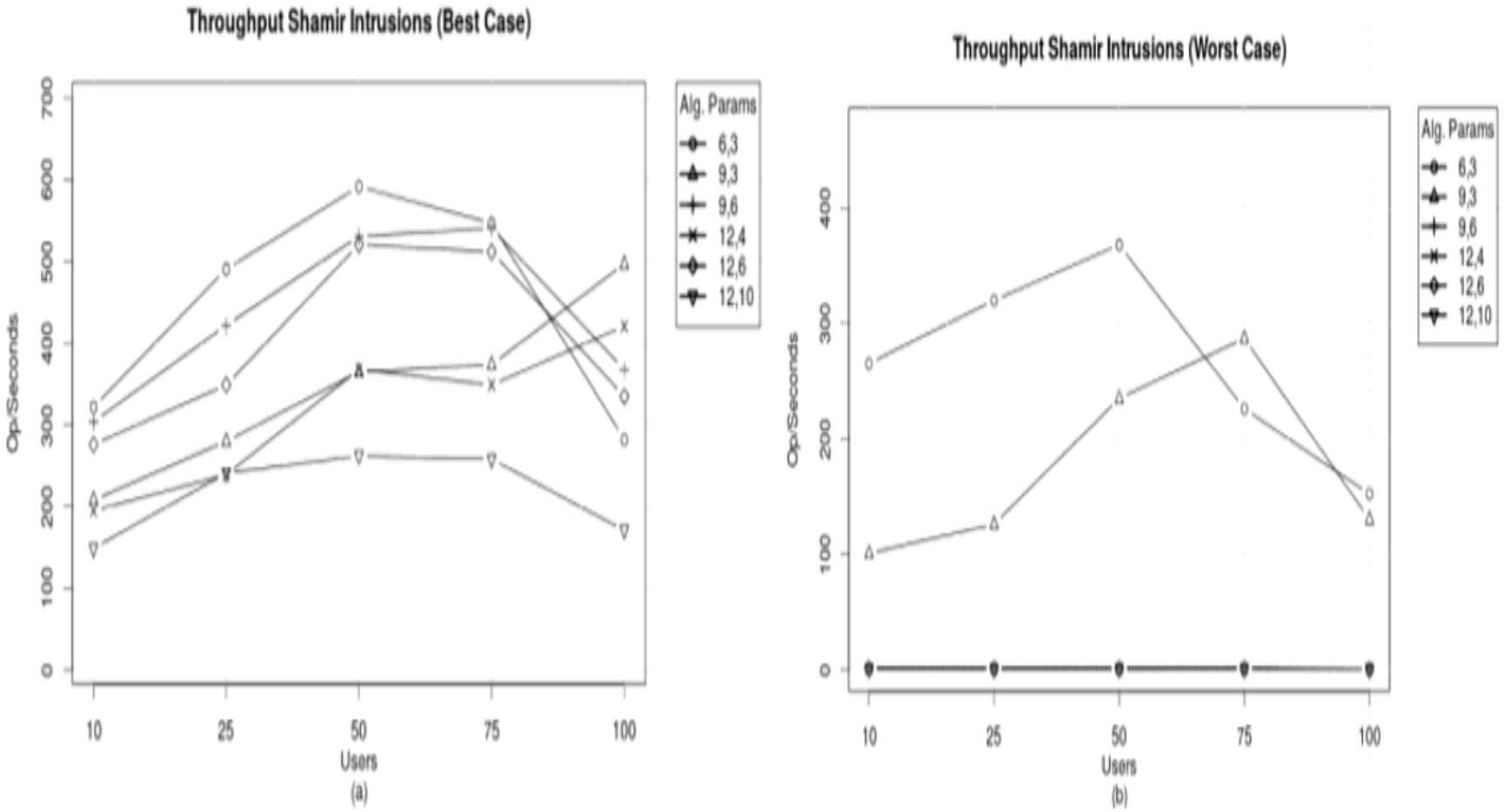


Figura 11 – Vazão com intrusões utilizando Shamir (operações por segundo)

Conclusão