Secure Authentication

Establishing authentication in a secure communication channel

Trabalho realizado por:
Tomás Costa - 89016
João Marques - 89234

Índice

Introdução

O projeto consiste no desenho e implementação de um protocolo que permita a autenticação mútua entre dois pontos, sendo que o canal segura já é fornecido pelo trabalho prático anterior da criação de um canal seguro de comunicações usando chaves simétricas. Pretende-se que seja possível trocar um ficheiro entre o cliente e o servidor usando o protocolo, caso o servidor considere que o utilizador tem essas permissões. O utente pode-se autenticar com senhas diretas ou com o cartão de cidadão. O servidor deverá conseguir provar a sua identidade, evintando-se ataques de MiTM ou impersonação.

Considerações

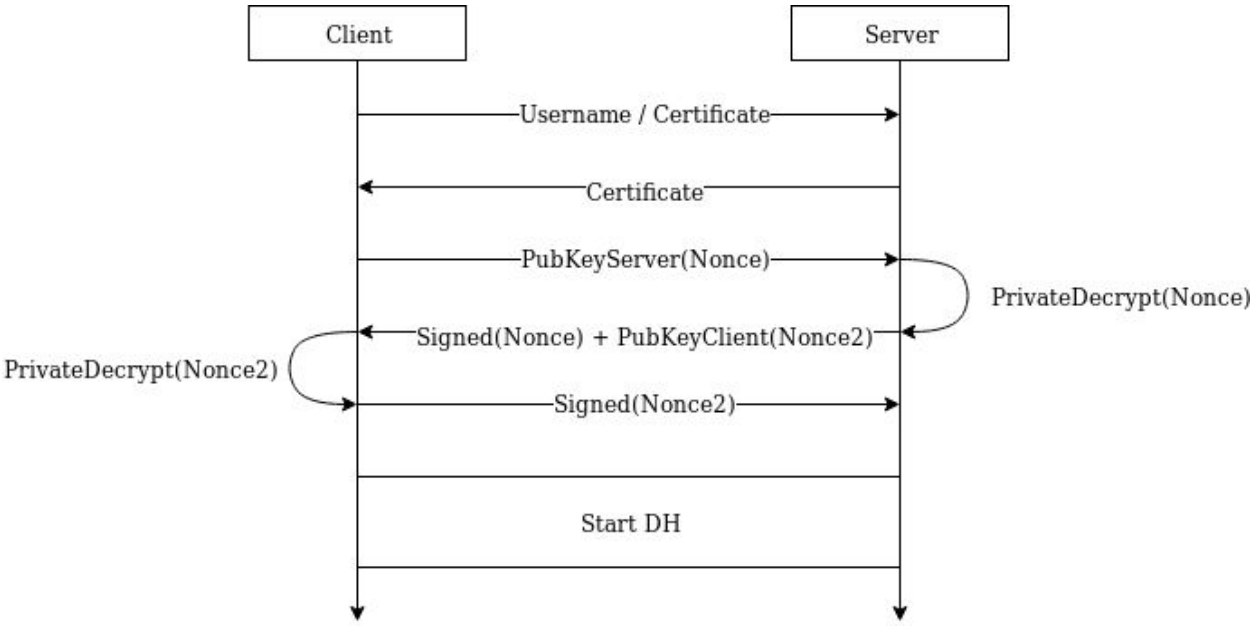
A estrutura inicial já continha o estabelecimento de um canal de comunicações seguro que foi criado para o projeto anterior. Sendo que este foi utilizado para partilhar os certificados e dados de autenticação tanto do cliente como do servidor.

O trabalho consiste em quatro grandes objetivos que, englobam tanto o desenho como a implementação, estes são:

- Protocolo para autenticação de utentes através da apresentação de senhas
- Mecanismo para controlo de acesso
- Protocolo para autenticação de utentes através do cartão de cidadão
- Protocolo para autenticação do servidor através de certificados X.509

Arquiteturas Inicias

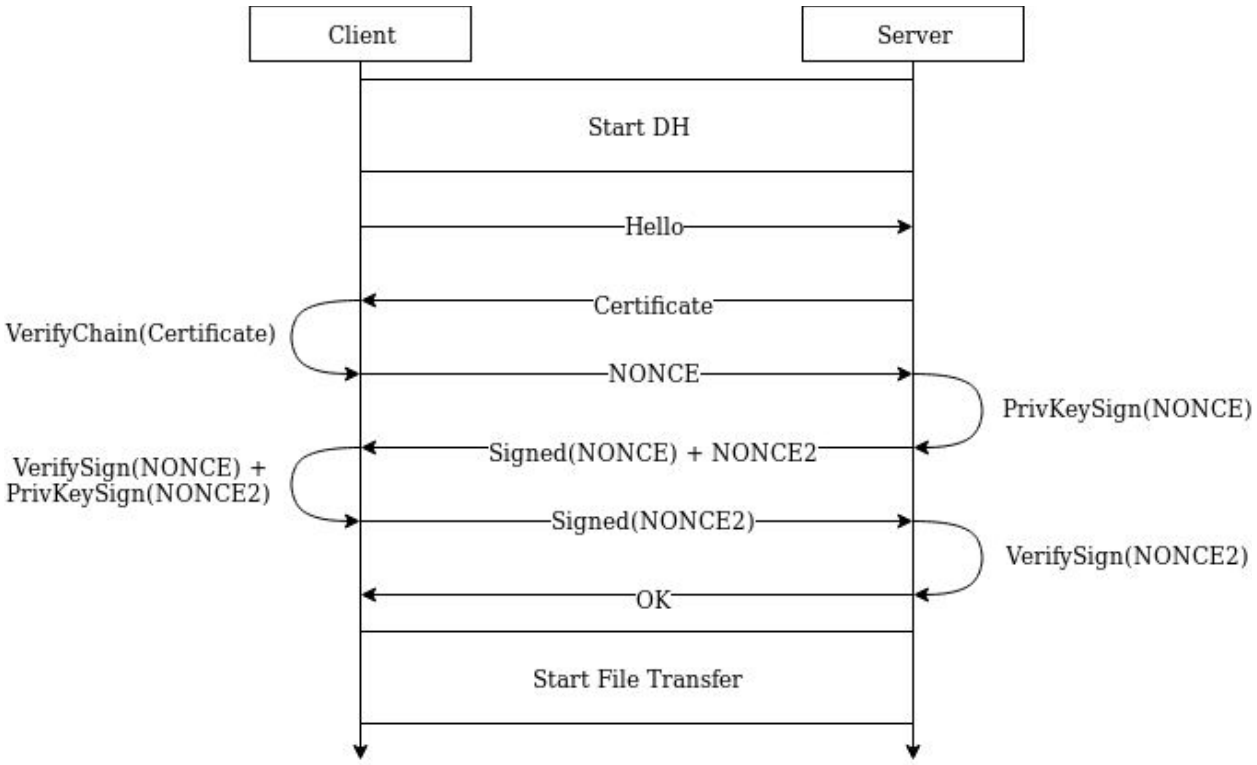
Antes de começarmos a implementar e, visto que havia a liberdade de desenhar qualquer arquitetura, pensamos em várias arquiteturas que foram posteriormente descartadas por vários motivos.
Apesar de termos definido várias arquiteturas, é mais relevante mencionar a penúltima e o porquê de não a termos utilizado.
A arquitetura era a seguinte:



Sendo que existem vários erros na arquitetura, os de maior relevância são: a partilha dos dados do username sem verificação do servidor; uso impróprio do DH (este pode já ser usado no início para os dados serem trocados com chaves simétricas); não há garantia de autenticidade, mas sim de confidencialidade, visto que estamos a encriptar com a pública do remetente.
A correção destes erros levou-nos a nossa arquitetura final, representada no próximo tópico.

Arquitetura Final

A arquitetura final é a seguinte:



Sendo que começamos por negociar as chaves simétricas e, posteriormente, passamos à autenticação encriptada com essas mesmas chaves.
O cliente contacta o servidor com um HELLO, o servidor envia o seu certificado e o cliente envia ou o certificado do seu Cartão de Cidadão ou os dados de login no servidor, depois é feito um desafio-resposta para cada entidade e após estas verificações todas (assumindo que o cliente tem acesso de escrita), o cliente pode começar a transferir o ficheiro.

Protocolo para autenticação de utentes através de senhas

O mecanismo de autenticação por senhas, passa por o utilizador inserir dois tipos de dados: username e password.
Caso estes dados não se encontrem na base de dados de utilizadores, o processo vai ser terminado (explicado mais detalhadamente na secção de controlo de acesso). Um exemplo teste é:

- USERNAME: tomas
- PASSWORD: 123

INSERIR AQUI FOTO DE INSERIR DADOS USER

Visto que não temos par chaves associado ao username, o que fazemos e enviar um NONCE do cliente ao servidor, o servidor responde assinando com a sua privada e manda também um NONCE (NONCE2).

INSERIR FOTO DA RESPOSTA AO NONCE

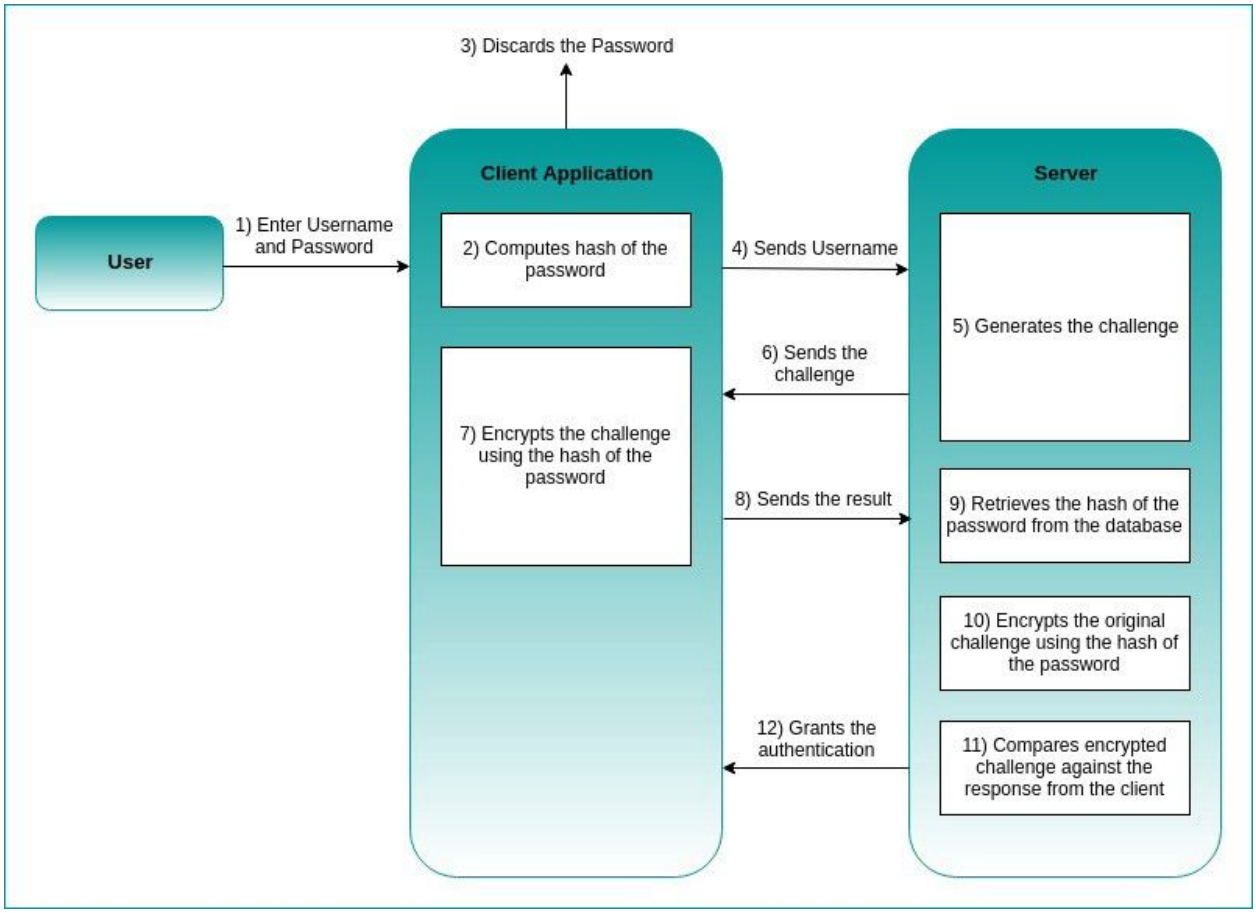
O cliente, por sua vez, valida a resposta ao NONCE vinda do server e responde ao NONCE2 da seguinte forma:

1. Hash da password inserida pelo utilizador
2. HMAC do hash da password em 1.
3. Assina NONCE com HMAC anterior

O servidor quando recebe a resposta, vai repetir o processo de hashing com a password que tem armazenada no sistema e, vai comparar a sua resposta ao desafio com a resposta do user ao desafio, caso sejam iguais e porque o cliente está a tentar autenticar com a password corresponde a sua conta e portanto pode ser validada a autenticação.

O seguinte processo foi implementado conforme descrito em: https://medium.com/@nipunadilhara/challenge-response-authentication-protocol-850925f50813

Segue o diagrama do desafio:



Mecanismo para controlo de acesso

Para o mecanismo de controlo de acesso, está implementado um ficheiro que serve como base de dados de utilizador, denominado de 'userdb' que segue o seguinte esquema:

```
tomas:BI158122680:123:AUTH_WRITE
tom1k:BI158122690:123hiperseguro:AUTH
joao:BI300281420:passsupersegura1234:AUTH_WRITE
```

A notação do ficheiro é a seguinte: um utilizador por linha com as suas informações delimitadas por ':' .
O seu username na posiçao 0, o seu número do BI na posiçao 1 e as suas permissões na posição 2.
Existem 2 tipos de permissões: AUTH (permissão para fazer login) e AUTH_WRITE (permissão para login e escrita).
E apesar de o sistema estar construido apenas com a funcionalidade da escrita de ficheiros, esta notação permite escalabilidade e adaptabilidade futura.

Protocolo para autenticação de utentes através do cartão de cidadão

Descreve que é semelhante ao de senha supracitado, mas que difere pq já nao há a marosca de verificar hashes, simplesmente mandamos certificado e vemos essa publ_key do lado do serv

Protocolo para autenticação do servidor através de certificados X.509

Mano aqui explica aquela cena que faz cenas quando verificas uma cena das cenas.

Conclusão

Com a conclusão deste trabalho, estamos bastante satisfeitos com o que realizamos, pois, permitiu-nos desenvolver: o raciocínio, através da refutação das diferentes arquiteturas iniciais, criatividade da solução, visto que a liberdade era grande e o esquema final não tinha apenas uma solução, e, mais importante, permitiu-nos aprofundar conceitos lecionados nas aulas de Segurança Informática em Organizações.

Bibliografia

Fontes mais relevantes à realização deste trabalho:

```
* [Slides 5 and 6](https://joao.barraca.pt/teaching/sio/2019/)
* [Cryptography.io](https://cryptography.io)
* [StackOverflow (Several Doubts)](https://stackoverflow.com/)
```