# Project 2:
# Extending the file system to support smartcard authentication

Sistemas de Elevada Confiabilidade

Grupo 10

Rui Peres 73831
Carlos Macedo 73919
João Costa 85652

## Arquitetura

Para esta entrega mantive-mos a estrutura de blocos já existente na primeira parte, ou seja, um cliente só tem um ficheiro e existe um bloco por ficheiro. Agora os ficheiros (blocos) são identificados pela chave pública do proprietário do ficheiro.

Esta nova parte do projeto, é caracterizada pela utilização do cartão de cidadão como meio de autenticação, para tal adicionámos duas novas função remotas ao servidor:

- **boolean storePubKey(X509Certificate cert)**: recebe e guarda o certificado do cliente devolvendo true, caso o certificado já exista no servidor, devolve false.

**- List<PublicKey> readPubKeys()**: devolve todas as chaves públicas guardadas no servidor.

Também modificámos a interface da biblioteca do sistema de ficheiros (FSLibrary):

**- byte[] FS_read(PublicKey id, int pos, int nbytes)**: de forma a identificar o ficheiro que se pretende ler, esta função agora recebe a chave pública como identificador desse ficheiro.

**- List<PublicKey> FS_list()**: devolve a lista de chaves públicas contidas no servidor.

**- void FS_exit()**: termina a utilização do cartão de cidadão.

## Confiabilidade

O nosso sistema de ficheiros incorpora nesta segunda parte, a autenticação de clientes através do respetivo cartão de cidadão (smart card) e permite também assinar os seus próprios ficheiros. O facto de se utilizar o smart card como token criptográfico garante uma maior segurança pois a chave privada do cliente não está guardada em nenhum servidor! A chave "anda" sempre com ele usufruindo da ubiquidade do cartão de cidadão.

A autenticação do cliente é então verificada quando o servidor antes de inserir o conteúdo no ficheiro verifica a assinatura recebida usando os argumentos da função *put_k*. Esta verificação garante igualmente a não repudiação do conteúdo inserido no ficheiro por parte do cliente.

O sistema de ficheiros também garante a integridade dos seus ficheiros ao verificar a assinatura do bloco correspondente ao ficheiro pretendido, usando a função *get*, (antes de enviar o ficheiro para ser lido pelo cliente).

No entanto, o nosso sistema de ficheiros não verifica a validade dos certificados guardados no servidor, nem verifica se o emissor ou o *certification path* dos certificados são de confiança.