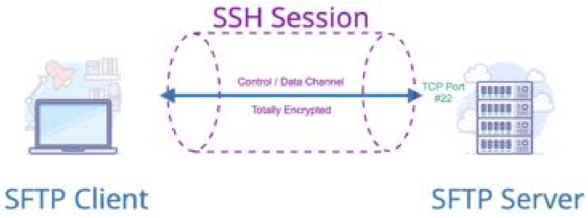
SSH, SSL, Single Sign-On, Authorization, Token, etc.

- **Authentication** – prove identity of the user or server (user name + password, cards, retina scans, voice recognition, fingerprints, secure certificates, tokens, etc.)
- **Authorization** – check if user/server is allowed (authorized) to do something
- **Encryption** - transforming the data so that it is unreadable without decryption key.
- **One-way encryption** - hash function, hard to reverse, different for different strings
- **Symmetric encryption** (symmetric cipher - uses same key to encrypt and decrypt).
- **Asymmetric encryption** (public key / private key).
- **PGP** = Pretty Good Privacy, since 1991, uses both symmetric an assymmetric encryption, the de facto standard for email security.
- **AES** = Advanced Encryption Standard, symmetric encryption, fast, used in databases
- **SSH** = Secure Shell protocol (SSH-2 : terminal, sending files, SecureFTP, etc.)
- **SSL** = Secure Socket Layer protocol (https://...), 1995 – 2015, deprecated
- **TLS** = Transport Layer Security, a successor of SSL 3.0, 1999, https://www.ssl2buy.com/wiki/ssl-vs-tls
- **SSL certificate** (a.k.a. "digital certificate") is installed on a web server and has two functions: It authenticates the identity of the website to visitors, and it is used for data encryption.
- **ssh-keygen** = command to manually generate a pair of keys as files in **.ssh** hidden directory
- **ssh.com** = SSH Communications Security, Inc. , proprietary SSH solutions
- ssh UKM (Universal Key Management) - https://www.ssh.com/products/universal-ssh-key-manager -
- **OpenSSH** – Open-source SSH implementation - https://www.openssh.com/
- **Single Sign-On** - https://en.wikipedia.org/wiki/Single_sign-on - authentication scheme that allows a user to log in once - and access multiple services without re-authenticating. User authenticates with an **Identity Server** which issues a **token**. User uses this token to access services. A service verifies the token with **Identity Server** before giving access.
- **Token** = a collection of data about the user which is passed between systems for getting access. Tokens must be **digitally signed** for the token receiver to verify that the token is coming from a trusted source. The **certificate** that is used for this **digital signature** is exchanged during the initial configuration process.
- https://jwt.io/ - JSON Web Tokens , http://www.passportjs.org/ - authentication middleware
- https://firebase.google.com/ - platform from Google for creating mobile and web applications. Includes authentication and tokens.
- **Oauth 2.0** - https://oauth.net/2/ - open standard for **access delegation**, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.
- **YubiKey** - a small USB device used to authenticate logins - https://www.yubico.com/products/ . On touch it sends a string containing it's public_id and AES-encrypted OTP (One-Time Password).

Taher Elgamal
"Father of SSL"
Netscape 1995-98

Tatu Ylönen,
in 1995 invented the
SSH protocol and
founded ssh.com
(SSH Communications
Security, Inc.)

HTTP + SSL = HTTPS
Hypertext Transfer Protocol
Secure Sockets Layer
Hypertext Transfer Protocol Secure


SSH Session
Control / Data Channel
Totally Encrypted
TCP Port #22
SFTP Client
SFTP Server