CS558 Network Security

Lecture 11: SSH

BOSTON UNIVERSITY

Unit 1 Very Brief Recap

ARP, DNS, BGP, DoS

Backbone internet protocols } Some form of DoS

Many to confidentiality

Confidential Communication – Goals

→ No Unintended Recipients can read the message

C I A

Authenticity → right binding to identities.

Integrity → message received = message sent

Replay Attack

timing

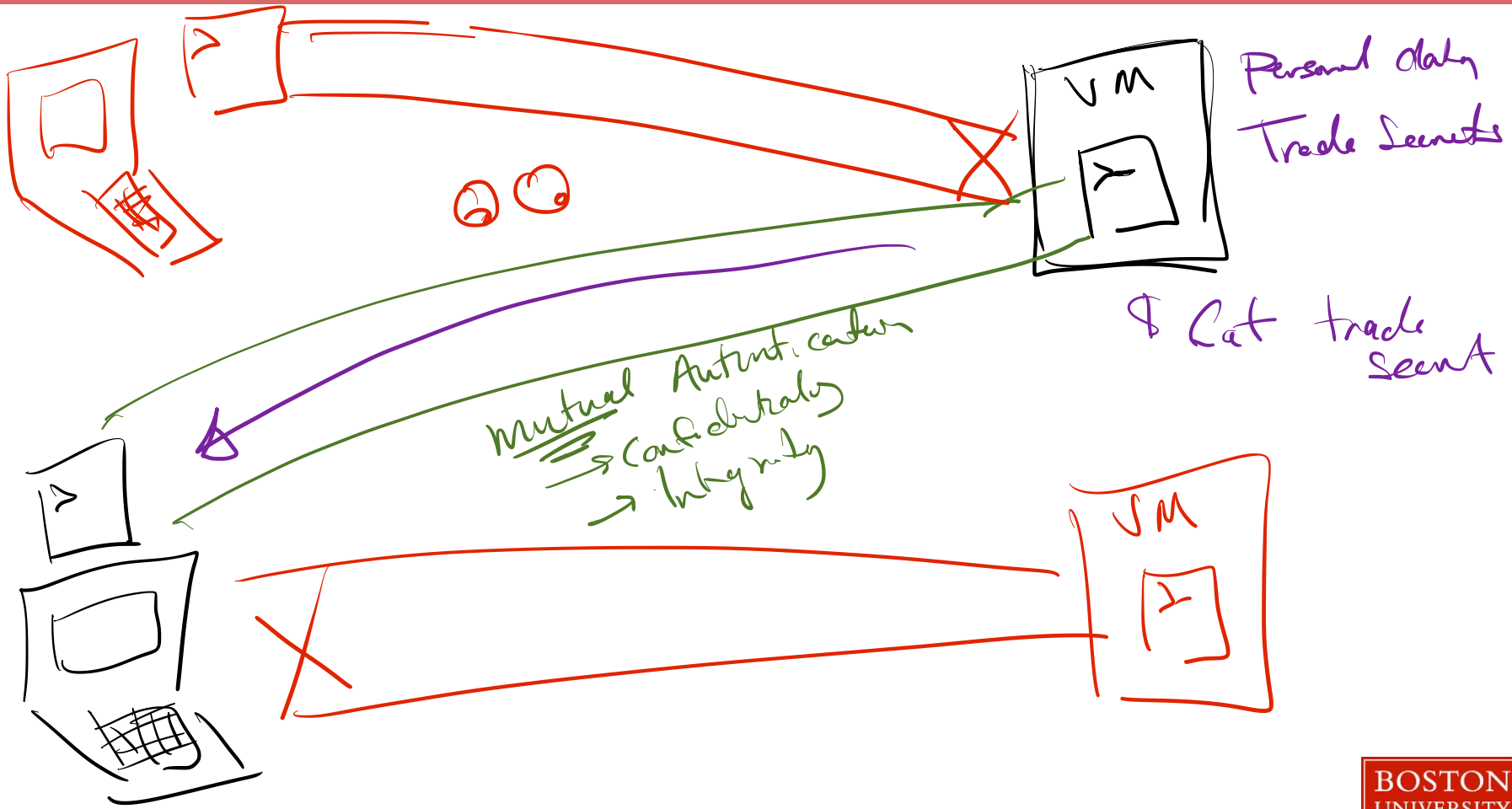
Forward Secrecy

The SSH (User) Experience

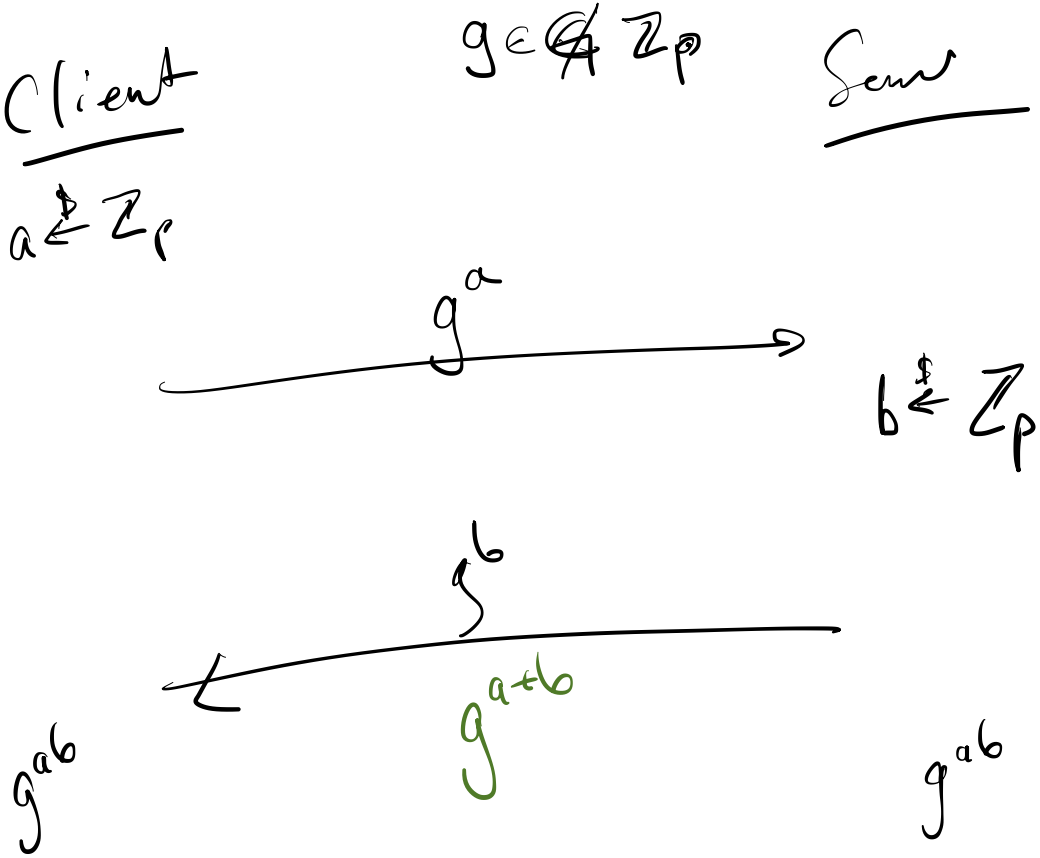
Keys → fulness

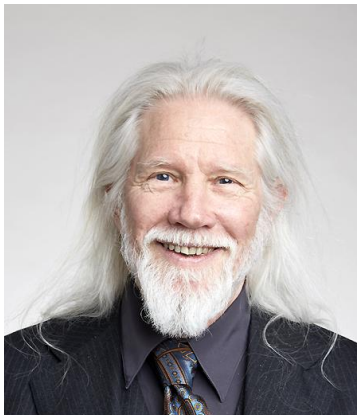
Password → fulness

fingerprints → on first access

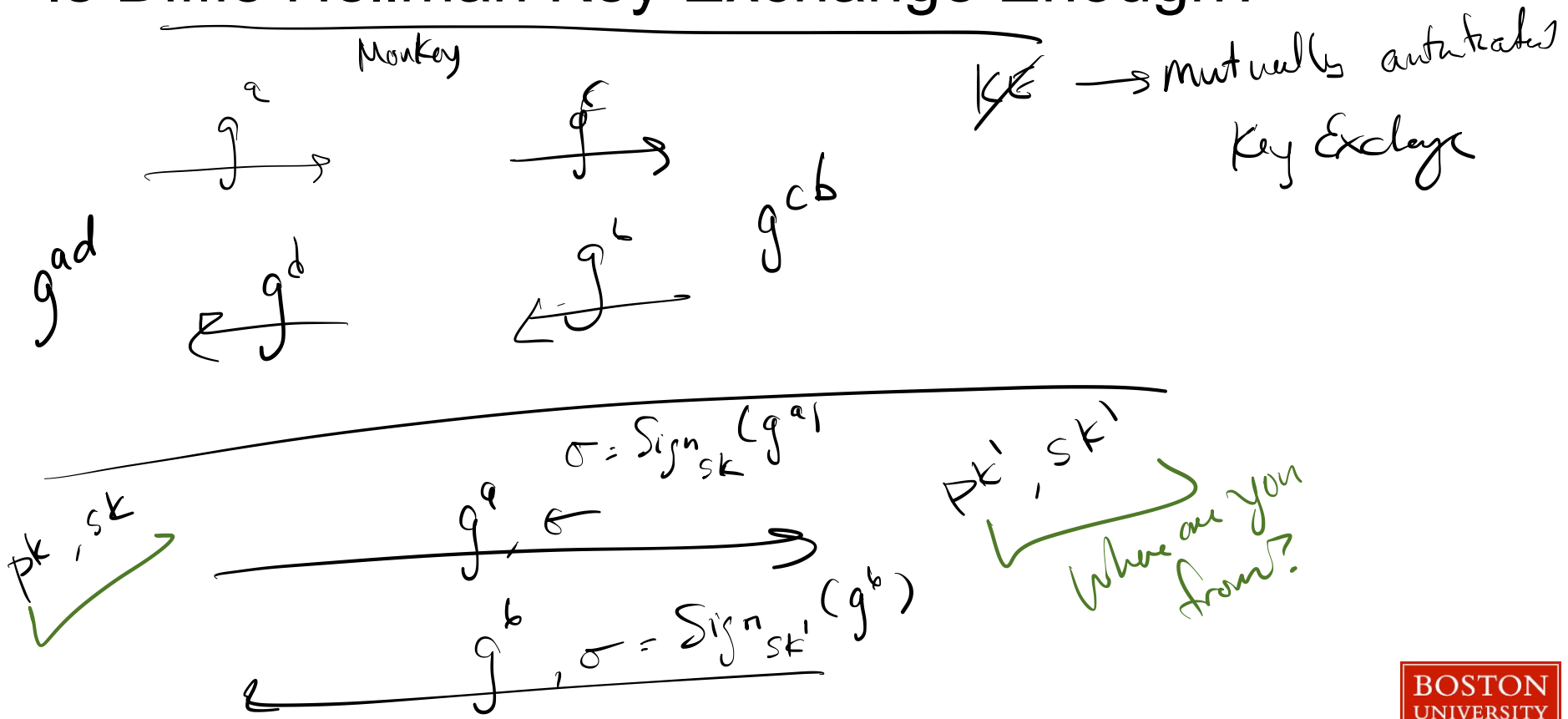


Remember Diffie & Hellman?





Is Diffie Hellman Key Exchange Enough?



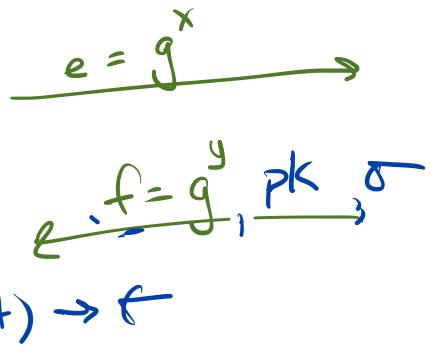
Authenticated Diffie Hellman

The following steps are used to exchange a key. In this, C is the client; S is the server; p is a large safe prime; g is a generator for a subgroup of GF(p); q is the order of the subgroup; V_S is S's identification string; V_C is C's identification string; K_S is S's public host key; I_C is C's SSH_MSG_KEXINIT message and I_S is S's SSH_MSG_KEXINIT message that have been exchanged before this part begins.

1. C generates a random number x (1 < x < q) and computes e = g^x mod p. C sends e to S.

2. S generates a random number y (0 < y < q) and computes f = g^y mod p. S receives e. It computes K = e^y mod p, H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K) (these elements are encoded according to their types; see below), and signature s on H with its private host key. S sends (K_S || f || s) to C. The signing operation may involve a second hashing operation.

3. C verifies that K_S really is the host key for S (e.g., using certificates or a local database). C is also allowed to accept the key without verification; however, doing so will render the protocol insecure against active attacks (but may be desirable for practical reasons in the short term in many environments). C then computes K = f^x mod p, H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K), and verifies the signature s on H.



To WIRESHARK!

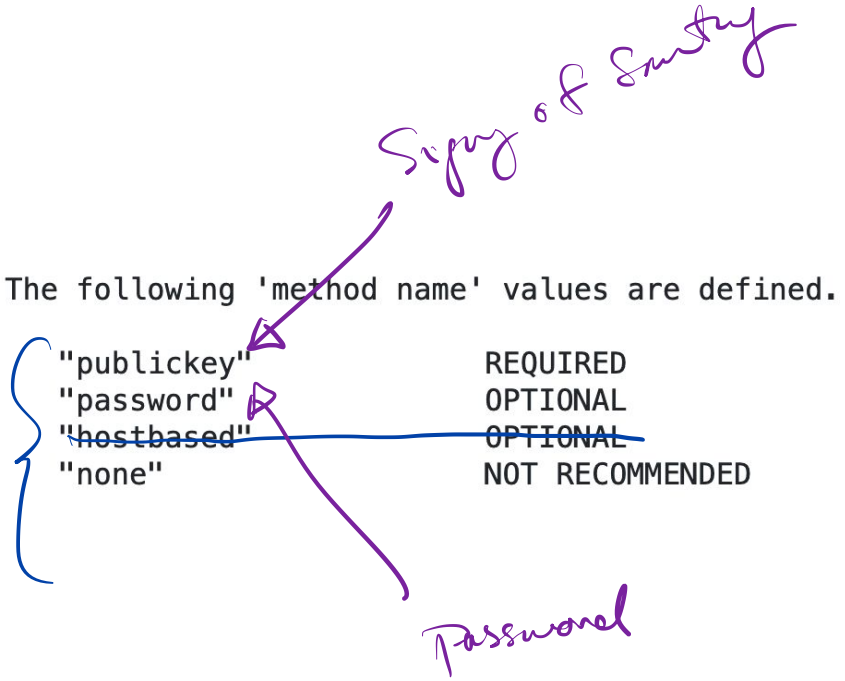
(Mutually) Authenticated Diffie Hellman

The following steps are used to exchange a key. In this, C is the client; S is the server; p is a large safe prime; g is a generator for a subgroup of GF(p); q is the order of the subgroup; V_S is S's identification string; V_C is C's identification string; K_S is S's public host key; I_C is C's SSH_MSG_KEXINIT message and I_S is S's SSH_MSG_KEXINIT message that have been exchanged before this part begins.

1. C generates a random number x (1 < x < q) and computes e = g^x mod p. C sends e to S.

2. S generates a random number y (0 < y < q) and computes f = g^y mod p. S receives e. It computes K = e^y mod p, H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K) (these elements are encoded according to their types; see below), and signature s on H with its private host key. S sends (K_S || f || s) to C. The signing operation may involve a second hashing operation.

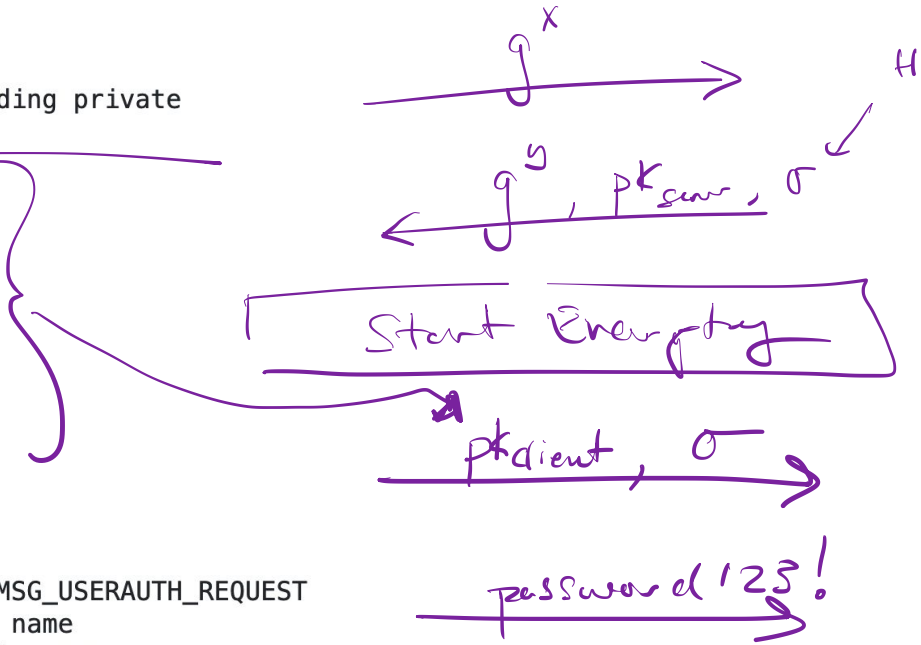
3. C verifies that K_S really is the host key for S (e.g., using certificates or a local database). C is also allowed to accept the key without verification; however, doing so will render the protocol insecure against active attacks (but may be desirable for practical reasons in the short term in many environments). C then computes K = f^x mod p, H = hash(V_C || V_S || I_C || I_S || K_S || e || f || K), and verifies the signature s on H.



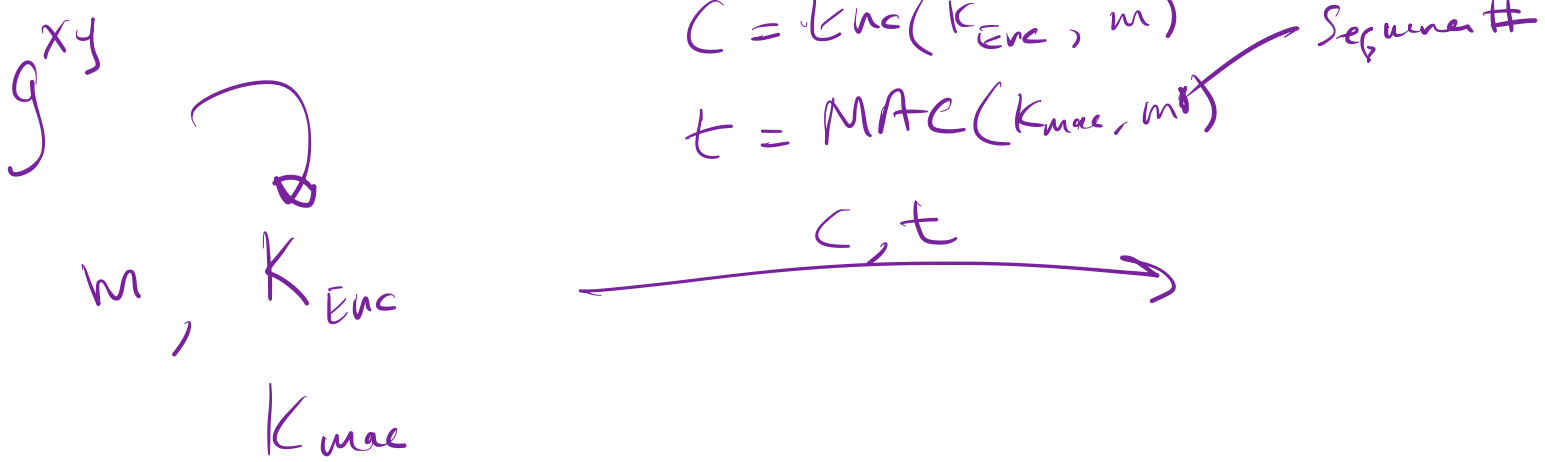
Mutual Auth Options

The value of 'signature' is a signature by the corresponding private key over the following data, in the following order:

```
string    session identifier
byte      SSH_MSG_USERAUTH_REQUEST
string    user name
string    service name
string    "publickey"
boolean   TRUE
string    public key algorithm name
string    public key to be used for authentication
```



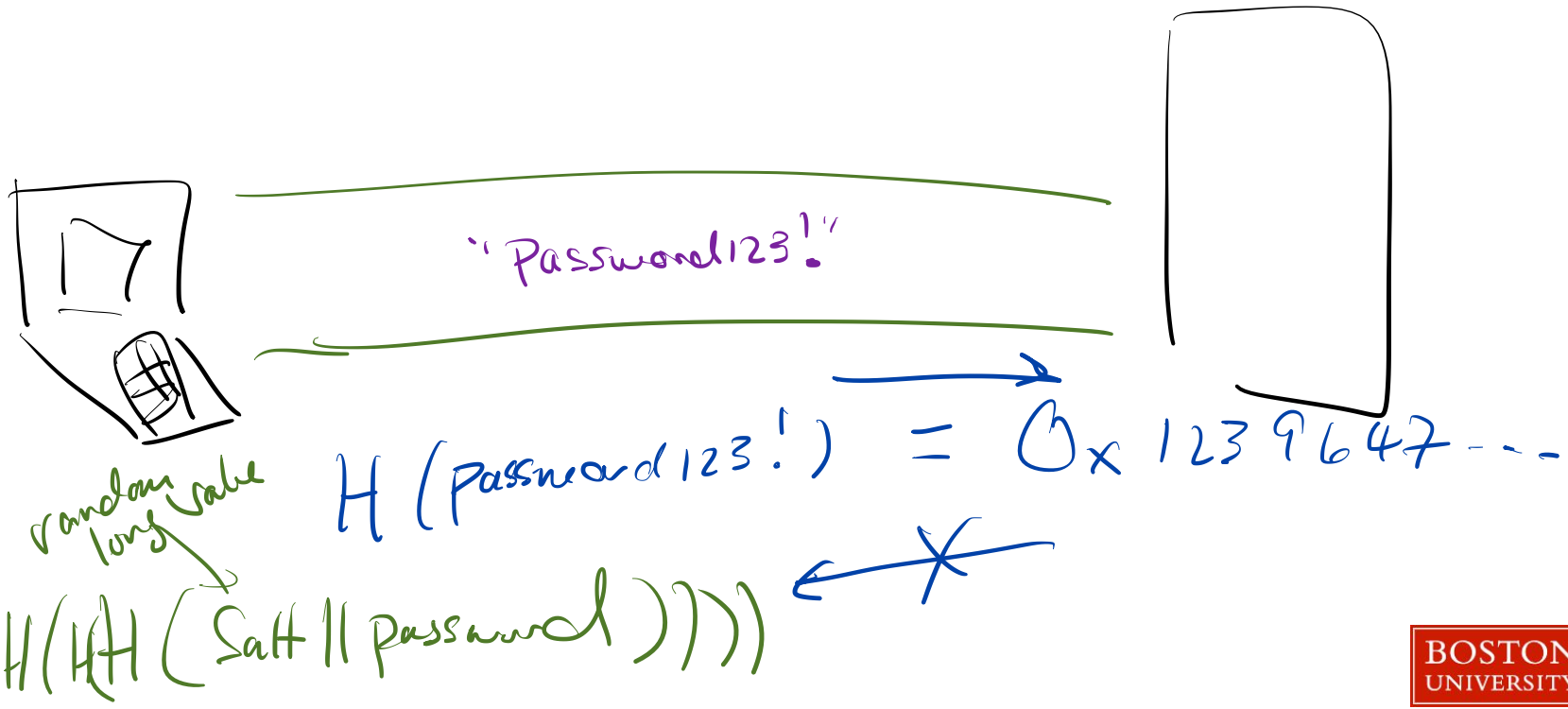

```
byte      SSH_MSG_USERAUTH_REQUEST
string    user name
string    service name
string    "password"
boolean   FALSE
string    plaintext password in ISO-10646 UTF-8 encoding [RFC3629]
```



So let's talk about Passwords (again)

Salt, Hash

Gabe: ~~Password123!~~



SCP/SFTP

file transfer

1. do an SSH Handshake