Cryptography II

(See lecture notes for CS4830 at https://www.noahsd.com/crypto_lecture_notes.html .)

Noah Stephens-Davidowitz, April 17, 2026

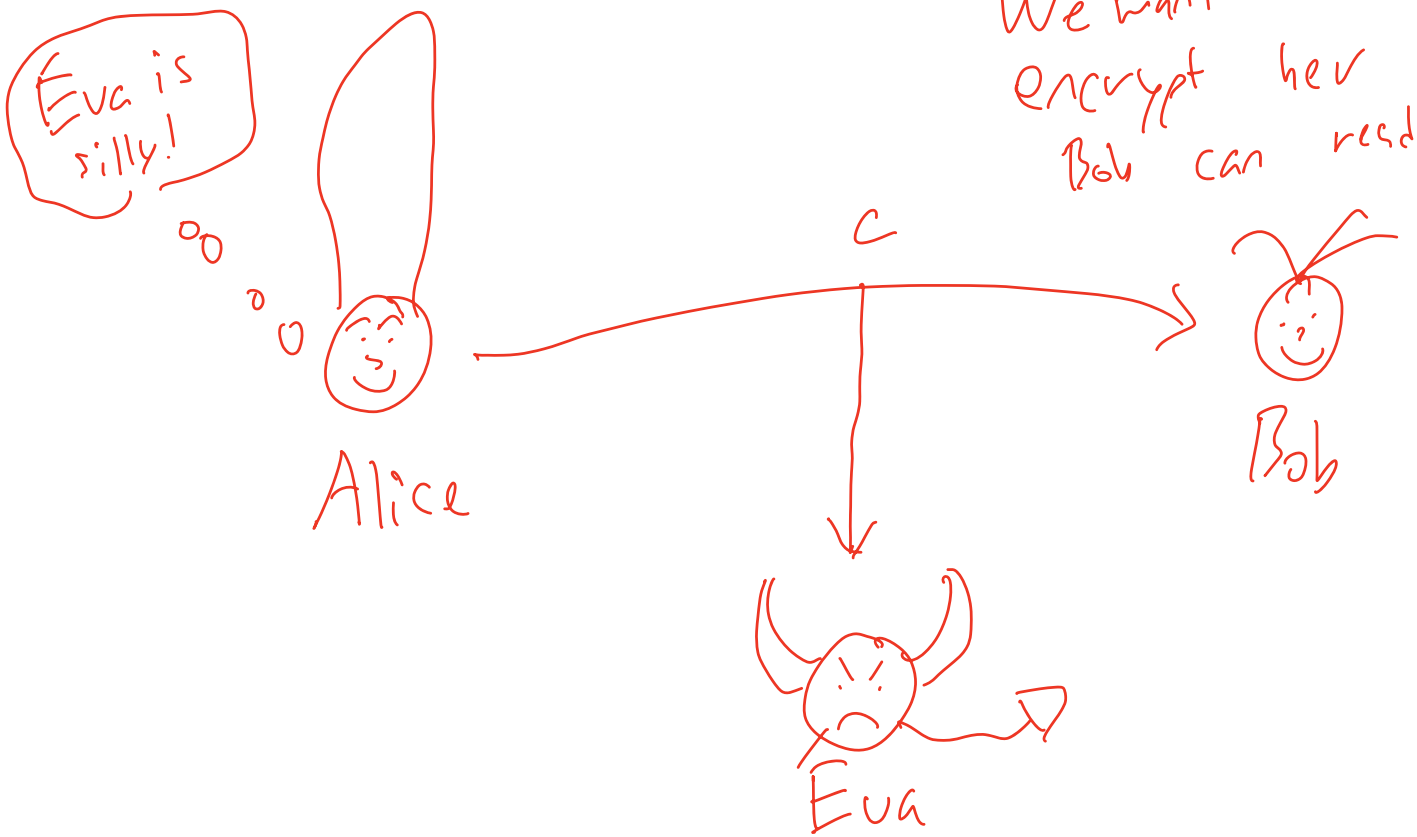
Announcement from Éva

Announcement from Éva

Next homework posted.

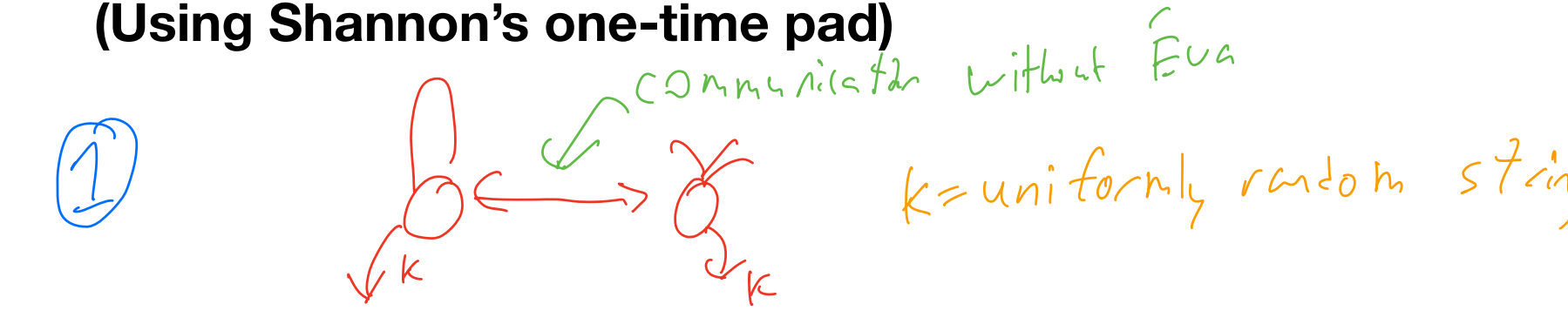
Sending a message privately

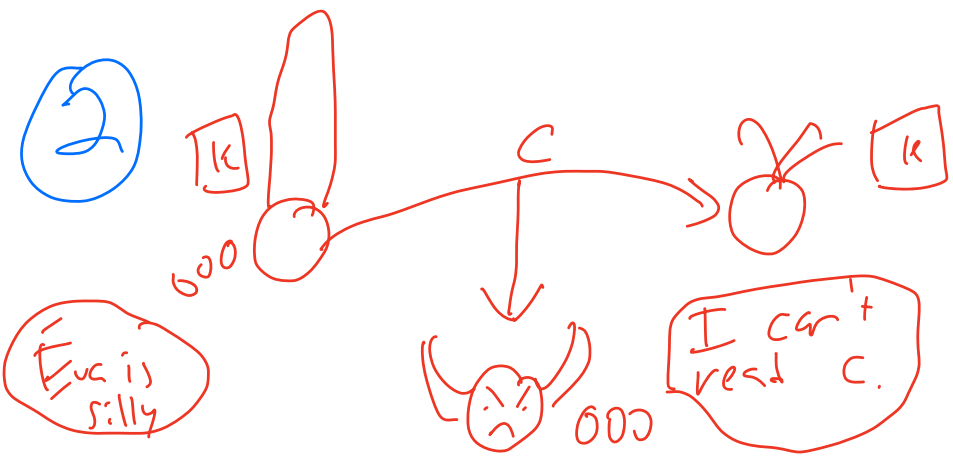
We want method for Alice to encrypt her message so that Bob can read it, but Eva can't.



Sending a message privately with a shared random secret key

(Using Shannon's one-time pad)



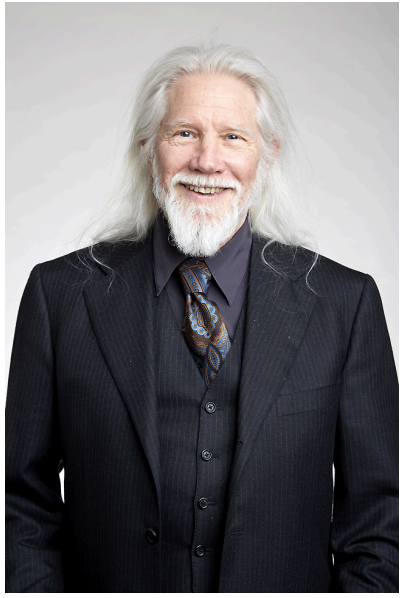


$(c = k \oplus m)$

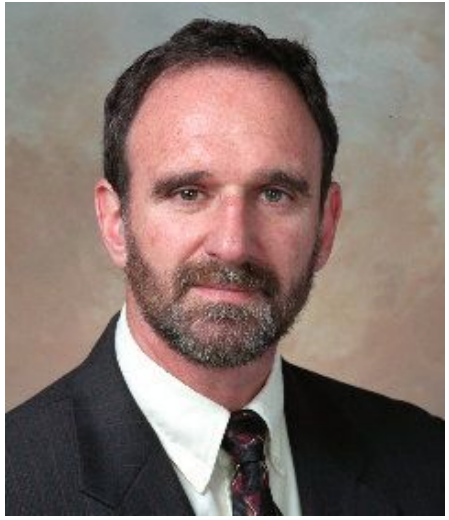
one-time pad

Public-Key Cryptography

[Diffie, Hellman 1976]



Whitfield (“Whit”) Diffie

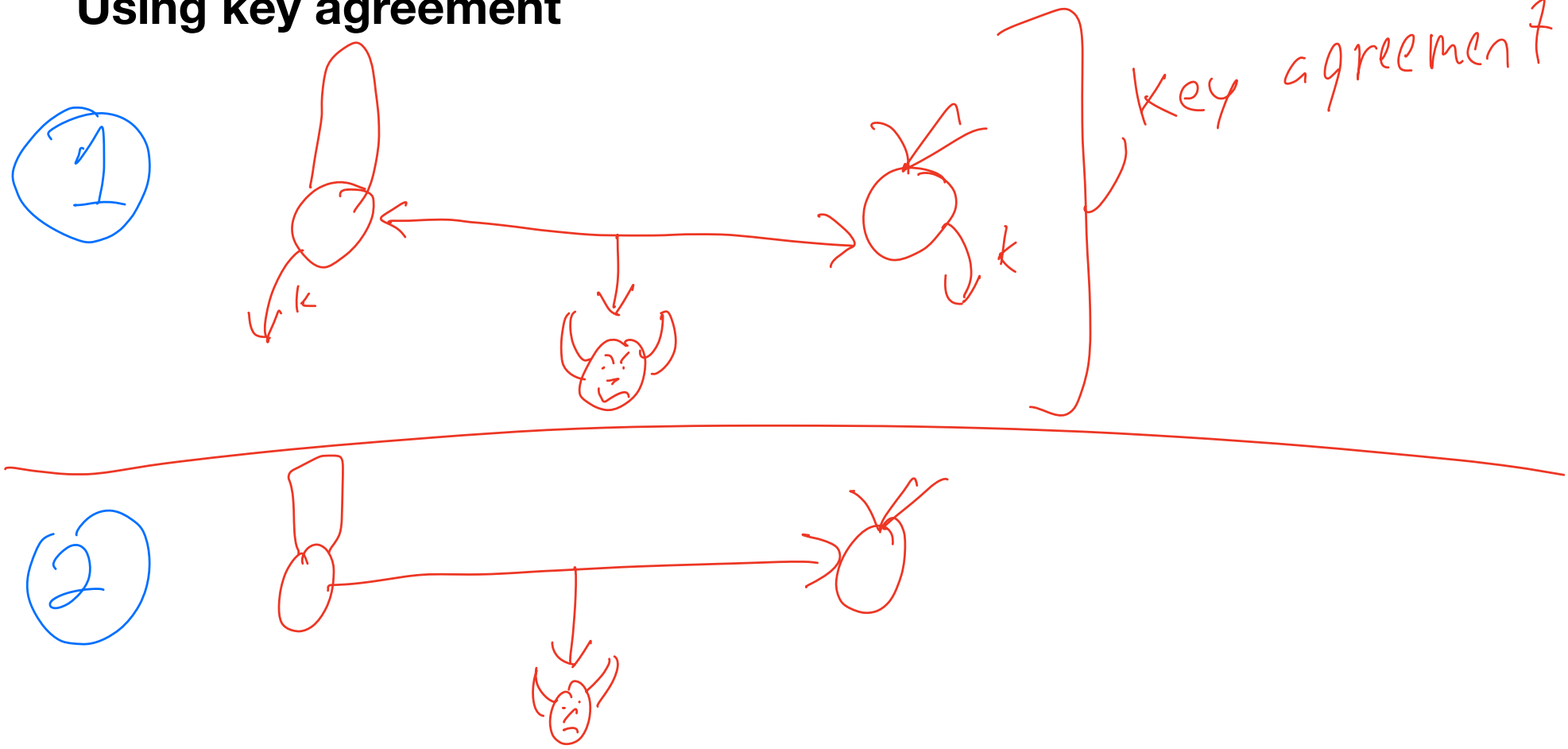


Martin Hellman

“New Directions in Cryptography”, 1976.

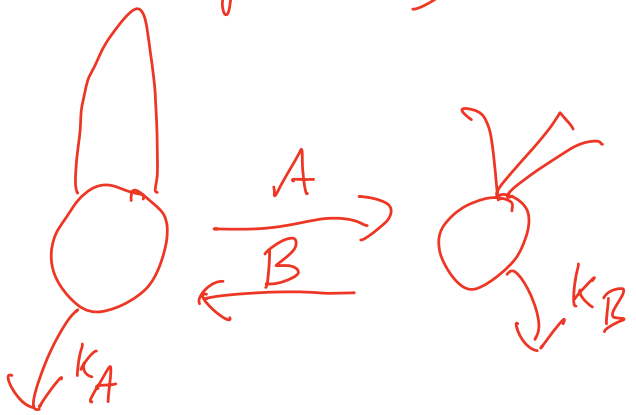
Sending a message privately with a shared “random" secret key

Using key agreement



Key agreement (aka, Key Exchange)

In a key agreement protocol, Alice and Bob exchange messages



and each output a key k_A, k_B.

Correctness: $k_A = k_B$

Security: For any efficient (poly time) algorithm $\mathcal{E}$, $\mathcal{E}(A, B, k_A) \approx \mathcal{E}(A, B, R)$

($\mathcal{E}$ = Eve; R is uniform and independent of A and B.)

Modular Exponentiation

Why are we talking about this? (For good reason!)

mod operator (%)

Given positive integers g and q,

$g \bmod q$ = remainder when we divide g by $q \in \{0, 1, \ldots, q-1\}$

Ex. $17 \bmod 6 = 5$ $\quad 17 = 6 * 2 + 5$

modular exponentiation

Given positive ints g, q, and a

$g^a \bmod q = \overbrace{g \cdot g \cdot g \cdot g \cdots g}^{a \text{ times}} \bmod q$ | $g^0 \bmod q = 1$, $g^1 \bmod q$, $g^2 \bmod q$, $g^3 \bmod q$, ...

this sequence must repeat

We call the minimal $a > 0$ s.t. $g^a \bmod q = 1$, the order of g.

The repeated squaring algorithm for modular exponentiation

Why are we talking about this? (For good reason!)

Computing $g^a \bmod q$ naively: Compute g, then $g^2 \bmod q$, then $g^3 \bmod q$, then $g^4 \bmod q$, ..., $g^{a-1} \bmod q$, $g^a \bmod q$.

This requires $a-1$ multiplications.

If a is an 100-bit integer, this takes $\approx 2^{100}$ time.

Repeated squaring

$h_0 = g$, $h_1 = g^2 \bmod q = h_0^2 \bmod q$, $h_2 = g^4 \bmod q = h_1^2 \bmod q$,

$$h_\ell = g^{2^\ell} \bmod q = h_{\ell-1}^2 \bmod q$$

Can compute h_ℓ using ℓ multiplications.

$$a = b_0 + 2b_1 + 4b_2 + 8b_3 + \cdots + 2^\ell b_\ell \qquad b_i \in \{0,1\}$$

$$g^a \bmod q = g^{b_0} \cdot (g^2)^{b_1} \cdot (g^4)^{b_2} \cdots (g^{2^\ell})^{b_\ell} \bmod q$$

$$= \underbrace{h_0^{b_0} \cdot h_1^{b_1} \cdots h_\ell^{b_\ell}} \bmod q$$

Can compute $h_0, \ldots, h_\ell$ using ℓ mults, and then this product using ℓ more multiplications.

$O(\ell)$ multiplications

$= O(\log a)$

The Discrete Logarithm

Why are we talking about this? (For good reason!)

If $A = g^a \bmod q$, we say that a is the discrete logarithm of A (mod q in base g).

The discrete logarithm problem asks us to find a s.t. $A = g^a \bmod q$, given g, q, A.

There is no known efficient algorithm for this problem!

One-way function: Easy to compute $A = g^a_{\bmod q}$ given a

but given A, hard to find a.

Diffie-Hellman Key Agreement

Public modulus q and public element g with order p (with appropriate number-theoretic properties).

Alice	Bob
$a \sim \mathbb{Z}_p$	$b \sim \mathbb{Z}_p$

Diffie-Hellman Key Agreement

Public modulus q and public element g with order p (with appropriate number-theoretic properties).

Alice		Bob
$a \sim \mathbb{Z}_p$		$b \sim \mathbb{Z}_p$
	$A := g^a \bmod q$ →	

Diffie-Hellman Key Agreement

Public modulus q and public element g with order p (with appropriate number-theoretic properties).

Alice		Bob
$a \sim \mathbb{Z}_p$		$b \sim \mathbb{Z}_p$
	$A := g^a \bmod q \longrightarrow$	
	$\longleftarrow B := g^b \bmod q$	

Diffie-Hellman Key Agreement

Public modulus q and public element g with order p (with appropriate number-theoretic properties).

Alice		Bob
$a \sim \mathbb{Z}_p$		$b \sim \mathbb{Z}_p$
	$A := g^a \bmod q \rightarrow$	
	$\leftarrow B := g^b \bmod q$	
Output $B^a \bmod q = g^{ab} \bmod q$		Output $A^b \bmod q = g^{ab} \bmod q$

Diffie-Hellman Key Agreement

Public modulus q and public element g with order p (with appropriate number-theoretic properties).

Alice		Bob
$a \sim \mathbb{Z}_p$		$b \sim \mathbb{Z}_p$
	$A := g^a \bmod q \rightarrow$	
	$\leftarrow B := g^b \bmod q$	
Output $B^a \bmod q = g^{ab} \bmod q$		Output $A^b \bmod q = g^{ab} \bmod q$

The Decisional Diffie-Hellman Assumption (DDH): No efficient algorithm can tell the difference between $(g, g^a \bmod q, g^b \bmod q, g^{ab} \bmod q)$ and $(g, g^a \bmod q, g^b \bmod q, g^c \bmod q)$ for random c.