CMSC/Math 456: Cryptography (Fall 2025)

Lecture 26

Daniel Gottesman

Administrative

Problem set #9 (last problem set) is due today.

Final exam: Friday, Dec. 19, 4-6 PM (in this room).

- Practice materials for the final (including past finals) are now available on ELMS and the course website.
- Will be open book again (textbook, lecture notes)
- Covers all topics from class.
- Slight emphasis on topics after the last midterm.
- Some topics (e.g., quantum and post-quantum, multiparty computation) will be qualitative questions only

Office hours next week:

- TA: Normal (2 PM Thursday, Zoom)
- Mine: Tuesday 3-5 PM, Atlantic 3251 and by appointment

Course evaluations are available to fill out.

This class is being recorded

Review Plan

- Principles and basic tools (Kerckhoff's principle, computational complexity, proof by reduction)
- Cryptographic primitives (Pseudorandom generators, pseudorandom functions, hash functions)
- Cryptographic protocols (Private and public-key encryption, key agreement, KEM, MAC, authenticated encryption, digital signature, identification protocol)
- Some problems from the midterms

This class is being recorded

Principles and Basic Tools

- Kerckhoff's principle
- Computational complexity
- Proof by reduction

This class is being recorded

Kerckhoff's Principle

Assume the protocol is known by the adversary. Only the key is secret.

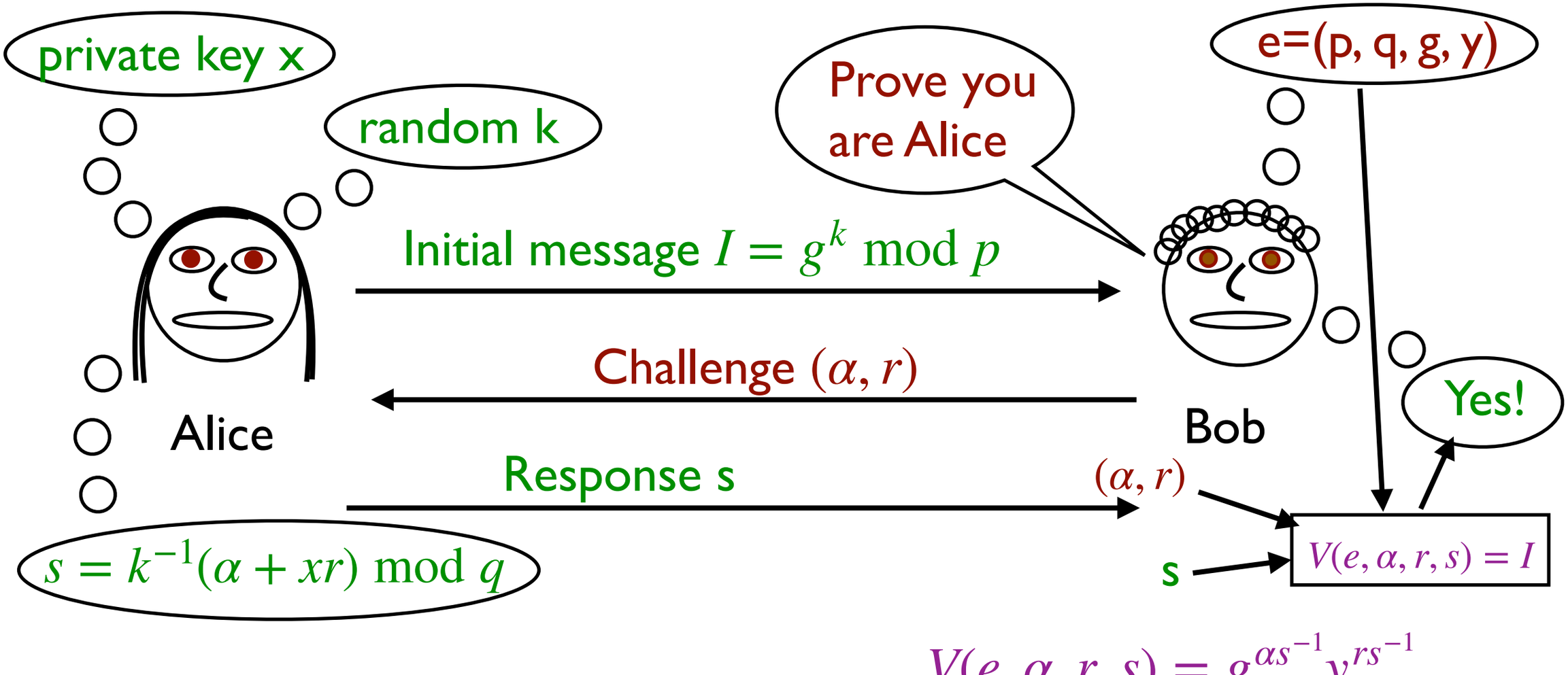
This means that anything that is not explicitly listed as part of the private key (or otherwise is secret) is known to Eve:

- Any parameters of the protocol (e.g., prime q or base g) are known to Eve.
- Any functions involved (e.g., hash function H(x)) are known to Eve.
- Public keys are certainly known to Eve.
- Private keys are not known by Eve.
- Random values picked by a participant and not explicitly announced are not known by Eve.

This class is being recorded

Example: Identification Protocol

The protocol involves the following values: p, q, g, x, y, k, I, α, r, s.
Which are known to Eve and which are not?



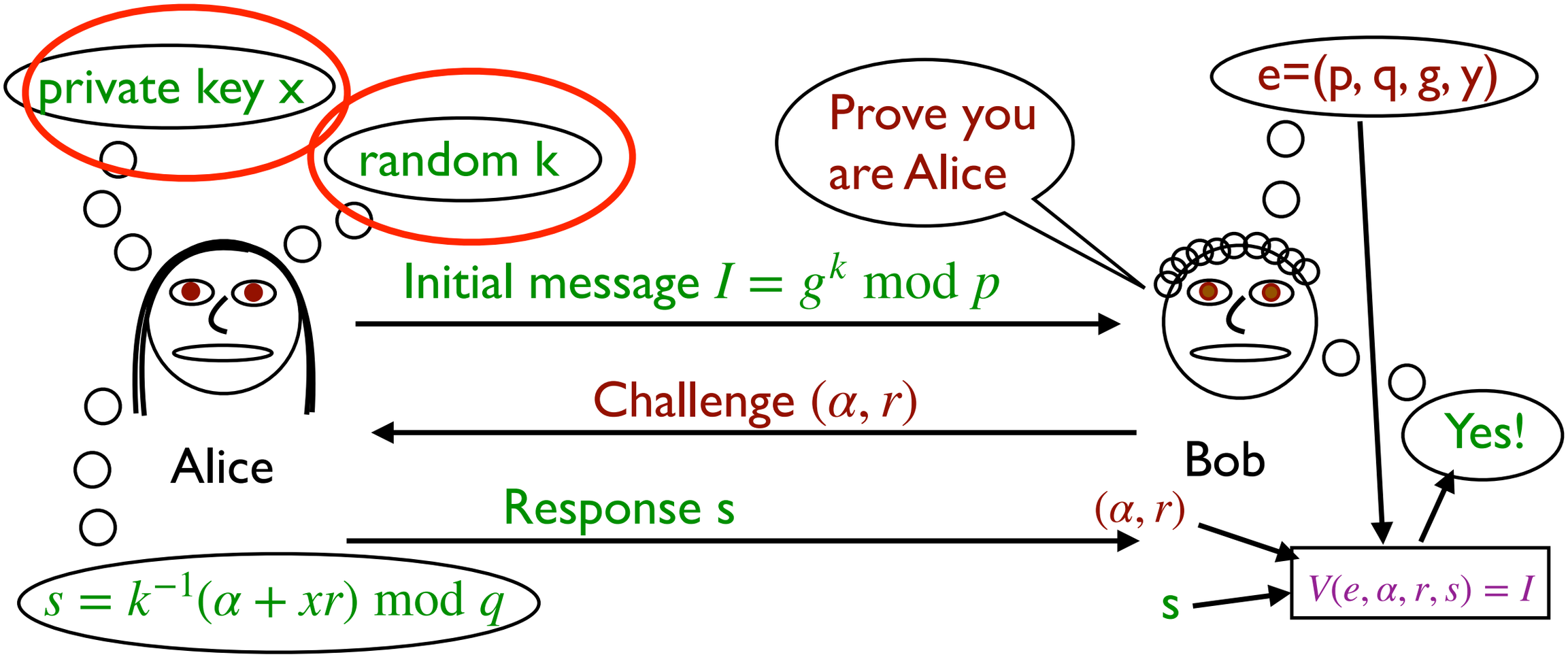
$$V(e, \alpha, r, s) = g^{\alpha s^{-1}} y^{rs^{-1}}$$

This class is being recorded

Example: Identification Protocol

The protocol involves the following values: p, q, g, x, y, k, l, α, r, s.
Which are known to Eve and which are not?

Secret: x, k



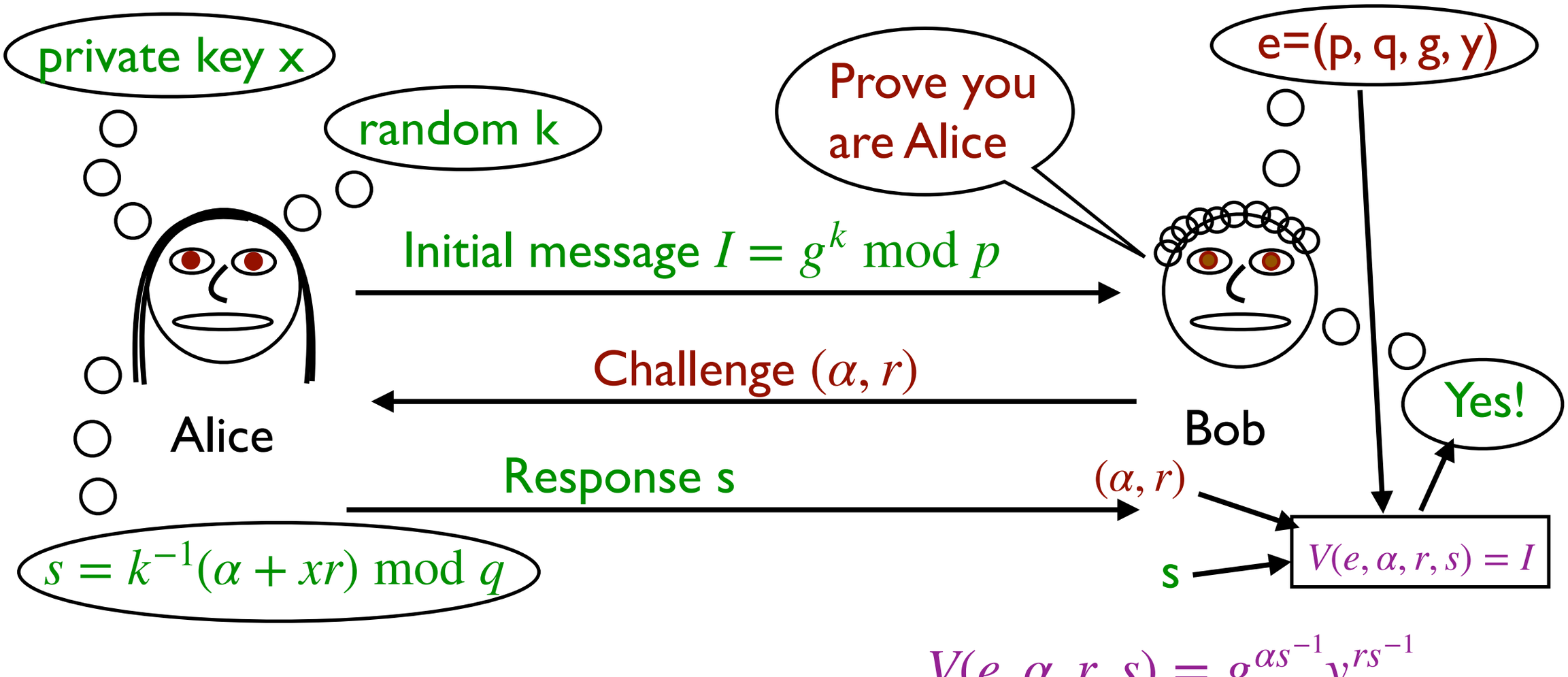
$$V(e, \alpha, r, s) = g^{\alpha s^{-1}} y^{rs^{-1}}$$

This class is being recorded

Example: Identification Protocol

The protocol involves the following values: p, q, g, x, y, k, I, α, r, s.
Which are known to Eve and which are not?

Secret: x, k



$$V(e, \alpha, r, s) = g^{\alpha s^{-1}} y^{rs^{-1}}$$

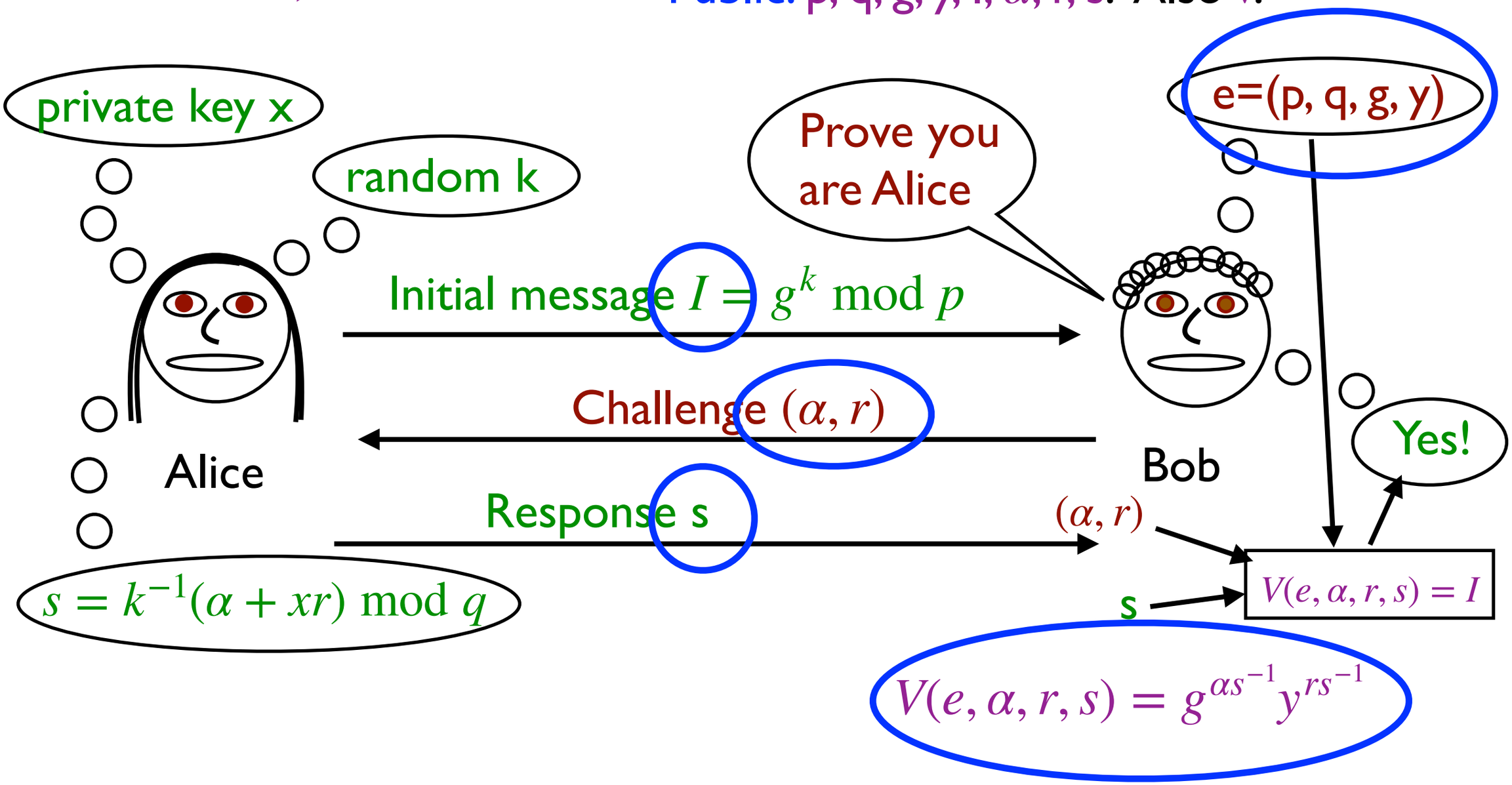
This class is being recorded

Example: Identification Protocol

The protocol involves the following values: p, q, g, x, y, k, I, α, r, s.
Which are known to Eve and which are not?

Secret: x, k

Public: p, q, g, y, I, α, r, s. Also V.



This class is being recorded

Efficient and Negligible

Almost always, we are interested in efficient algorithms, namely ones which run in a time polynomial as a function of the size of the input to the function.

Big-O Notation:

A function $f(x) = O(g(x))$ for the function g(x) if f is *less than or equal to* g in the asymptotic limit of large x. Specifically, $f(x) \leq Cg(x)$ for constant C and sufficiently large x.

E.g.: $f(x) = 1/(100x^3)$ is $O(1)$ and $O(1/x^2)$ but is *not* $O(1/x^5)$.

This class is being recorded

Efficient and Negligible

Almost always, we are interested in efficient algorithms, namely ones which run in a time polynomial as a function of the size of the input to the function.

Big-O Notation:

A function $f(x) = O(g(x))$ for the function g(x) if f is *less than or equal to* g in the asymptotic limit of large x. Specifically, $f(x) \leq Cg(x)$ for constant C and sufficiently large x.

E.g.: $f(x) = 1/(100x^3)$ is $O(1)$ and $O(1/x^2)$ but is *not* $O(1/x^5)$.

A function is negligible if it goes to 0 faster than *any* polynomial. Specifically, $\lim_{x\to\infty} f(x)p(x) = 0$ for all polynomials p(x).

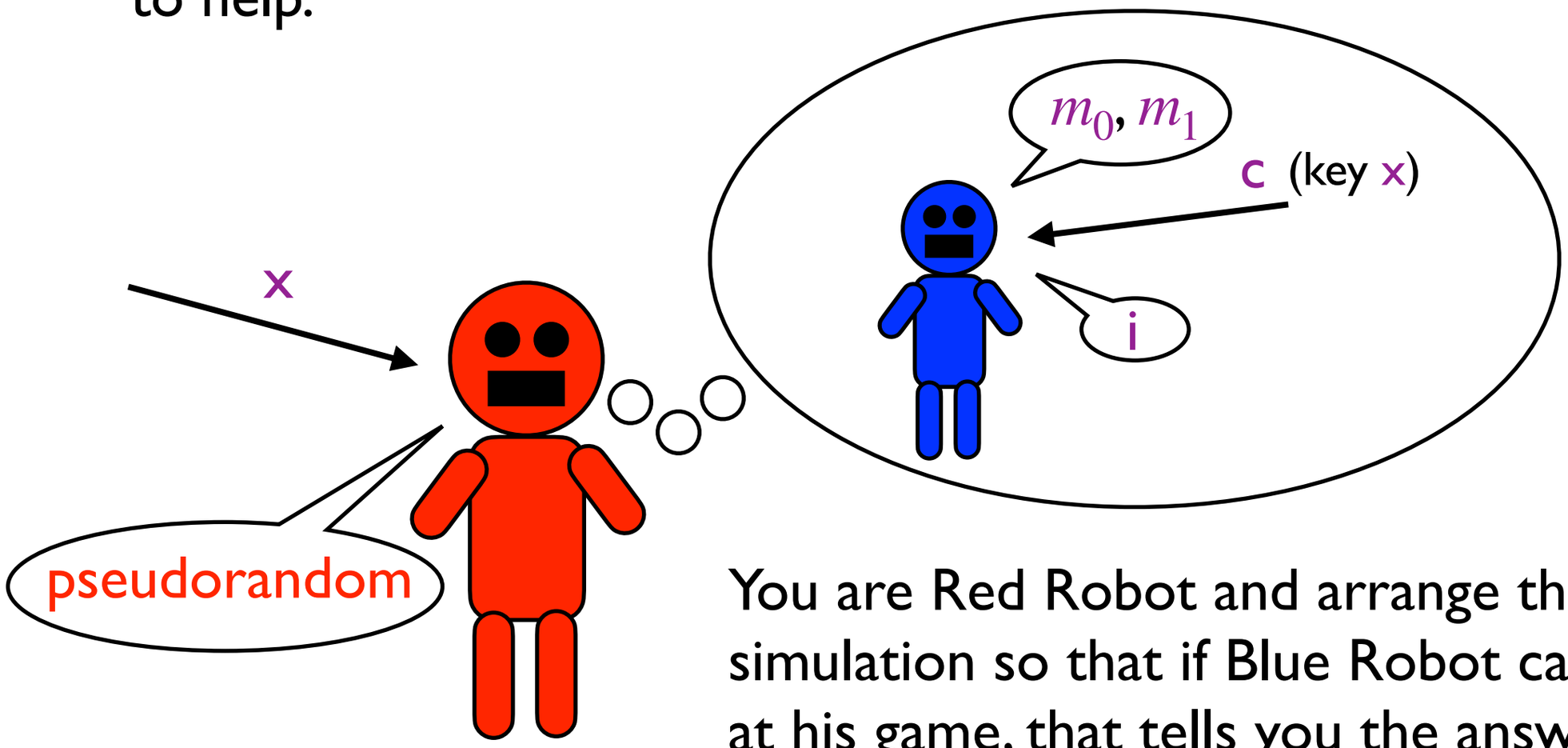
E.g.: $f(x) = 1/(100x^3)$ not negligible

$f(x) = \exp(-\sqrt{x})$ negligible

This class is being recorded

Reductions

Red Robot's job is to play some cryptographic game. He is very bad at his job, so he makes a copy of Blue Robot and runs a simulation of Blue Robot playing a different cryptographic game to help.



You are Red Robot and arrange the simulation so that if Blue Robot can win at his game, that tells you the answer for your own game.
Red Robot's game reduces to Blue Robot's game.

This class is being recorded

Reduction Example

For example, suppose your job is to break the RSA assumption: Given N, e, and random y, find x such that $x^e = y \bmod N$.

> Blue Robot plays the factoring game: Given N, find a factor of N.

When you are given (N, e, y), you start your simulation and your friend's copy thinks they are going to work. You arrange for them to be given the problem N and they answer p. You take this value out of the simulation and calculate q = N/p, then $\varphi(N)$. Then you use Euclid's algorithm to find $d = e^{-1} \bmod \varphi(N)$ and compute $y^d \bmod N$ and use that for your answer x.

> You have reduced breaking RSA to factoring.

Note: The simulation must be identical to Blue Robot's real job or the copy will realize it is a copy.

This class is being recorded

Hardness via Reductions

Unfortunately for you, your friend is also bad at their job. The answers they give are not real factors, or maybe they don't answer at all. Either way, your algorithm will fail.

> Disappointed, you conclude that there is no way to do your job.

Is this a valid conclusion?

This class is being recorded

Hardness via Reductions

Unfortunately for you, your friend is also bad at their job. The answers they give are not real factors, or maybe they don't answer at all. Either way, your algorithm will fail.

> Disappointed, you conclude that there is no way to do your job.

Is this a valid conclusion?

> No. There could be a different robot that is better at factoring, or maybe there is a way to beat RSA that doesn't involve making an unauthorized copy of anyone.

This class is being recorded

Hardness via Reductions

Unfortunately for you, your friend is also bad at their job. The answers they give are not real factors, or maybe they don't answer at all. Either way, your algorithm will fail.

> Disappointed, you conclude that there is no way to do your job.

Is this a valid conclusion?

> No. There could be a different robot that is better at factoring, or maybe there is a way to beat RSA that doesn't involve making an unauthorized copy of anyone.
>
> > But: If your friend finds out about your plan, he might be able to conclude that his job — factoring — is hopeless. If RSA is unbreakable, then factoring must be hard as well.

This class is being recorded

Cryptographic Primitives

- Pseudorandom generators
- Pseudorandom functions
- Hash functions

This class is being recorded

Pseudorandomness

Pseudorandom generator G(s)

- One input s, "seed"
- Output looks like a random string when s unknown
- Output should be longer than the seed
- Stream cipher is a more flexible version

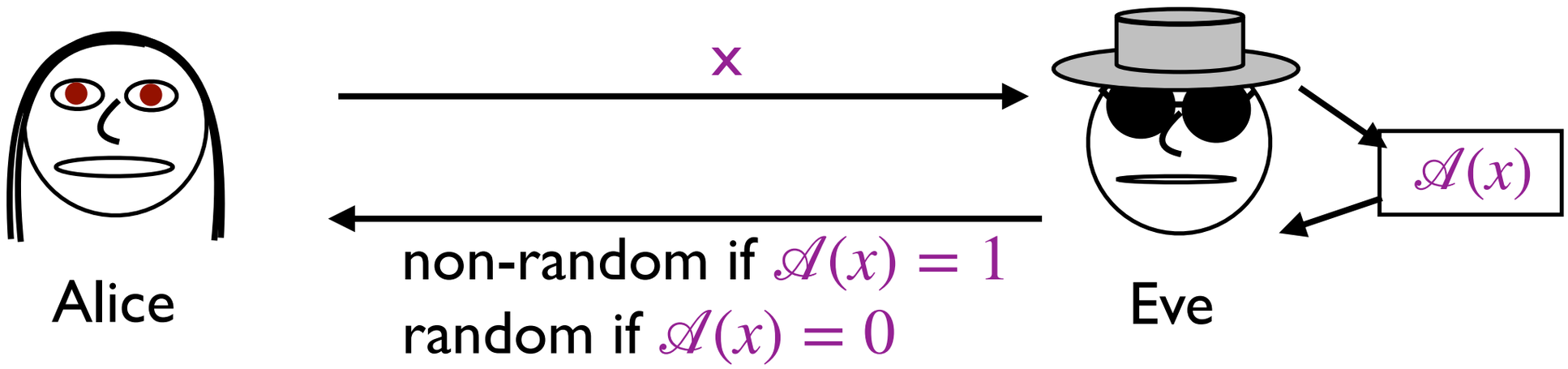
Pseudorandom function $F_k(r)$

- Two inputs: k (key) and r
- For fixed but unknown k, looks like a random function of r
- Output can be the same size or smaller than the input
- Block cipher is a fixed-size version, but must be a permutation (with computable inverse for known k)

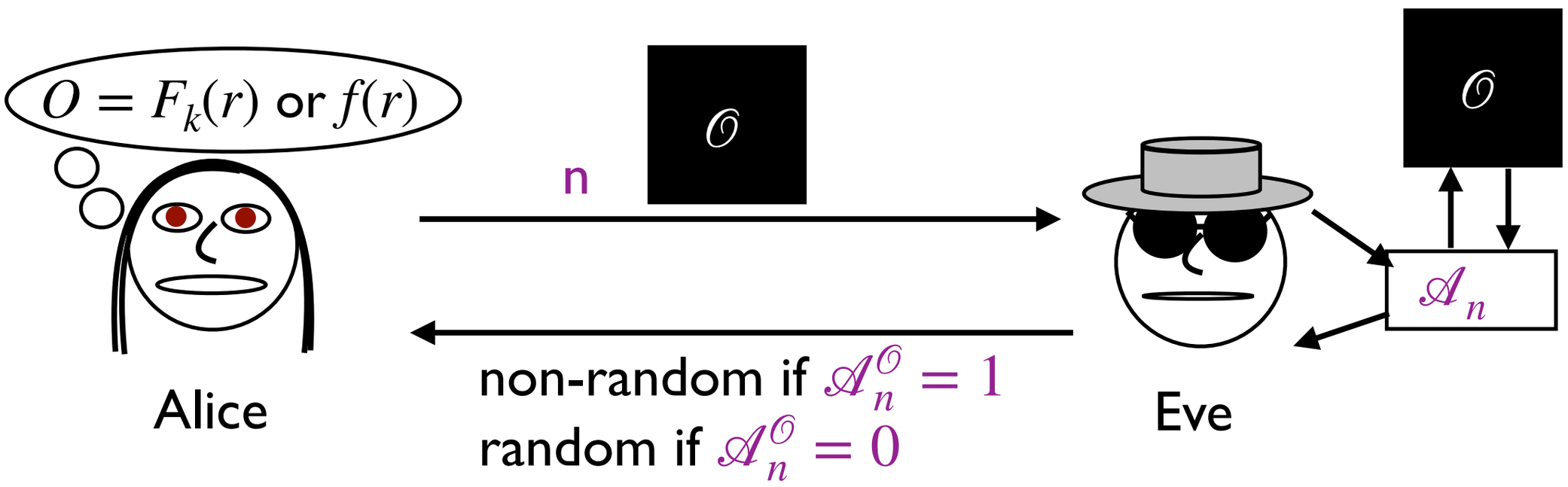
This class is being recorded

Pseudorandomness Games

Pseudorandom generator:



Pseudorandom function:



This class is being recorded

Hash Functions

Hash function H(x)

- Unlike pseudorandom functions and generators, function and input are often both known (i.e., no key)
- Output must be shorter than the input
- Main cryptographic property is collision resistance, meaning it is hard to find two inputs x_1, x_2 with the same output $H(x_1) = H(x_2)$
- Does not need to look like a random function
- But is often modeled as a random oracle anyway
- Note: but (unlike a pseudorandom function) it is always easy to distinguish from a truly random function since we can just test it on specific inputs
- Often take arbitrary-length inputs

This class is being recorded

Cryptographic Protocols

- Private-key encryption
- Public-key encryption
- Key agreement
- Key encapsulation mechanism (KEM)
- MAC
- Authenticated encryption
- Digital signature
- Identification protocol
- (Also: various multiparty computation-related protocols, but we will not be reviewing them.)

This class is being recorded

Cryptographic Protocols Comparison

Protocol	Purpose	Pub./Priv.	Interactive?
Private-key encryption	Encryption	Private	No
Public-key encryption	Encryption	Public	No
Key agreement	Gen. key	None	Yes
KEM	Gen. key	Public	No
MAC	Authenticate	Private	No
Authenticated encrypt.	Enc. + Auth.	Private	No
Digital signature	Authenticate	Public	No
Identification protocol	Authenticate	Public	Yes

This class is being recorded

Public Key vs. Private Key

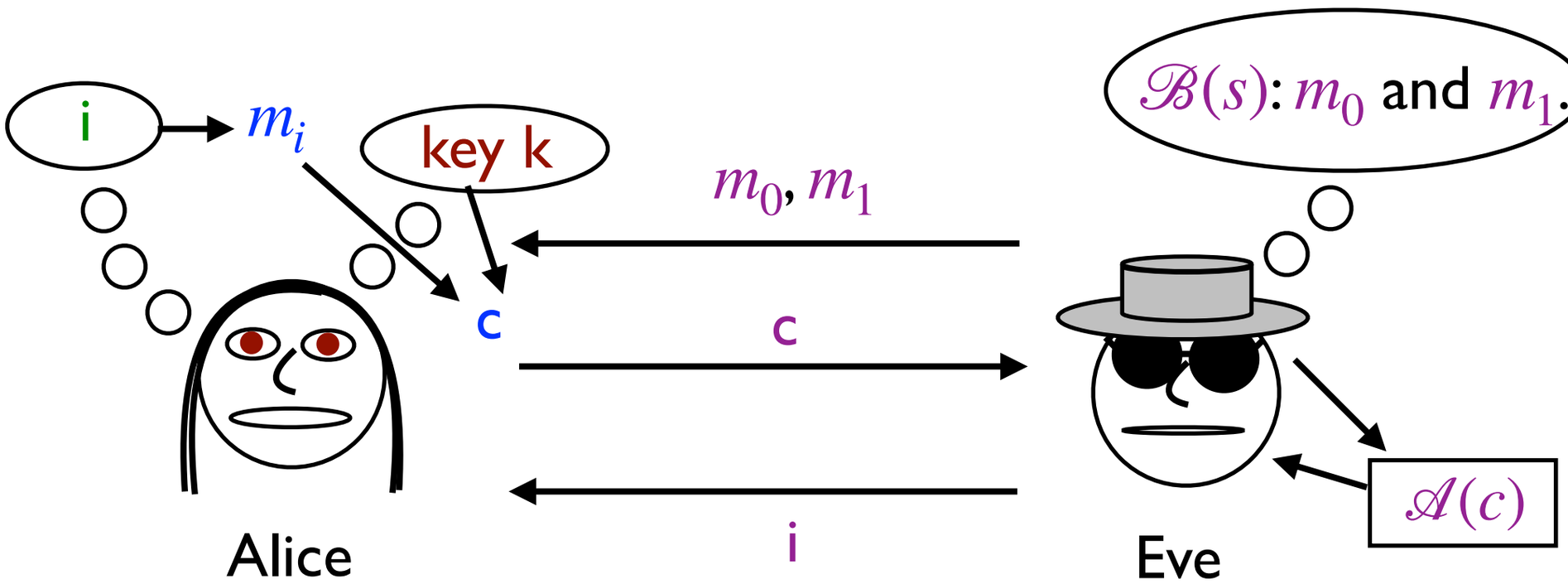
Private Key	Public Key
Highly secure (short keys sufficient)	Less secure (long keys needed)
Very efficient	Slow
Symmetric (same key)	Asymmetric (public and private key pair)
Need different key for each pair of people	Public key can be distributed to many people
	May have additional properties

This class is being recorded

Encryption Security Definitions

Eve chooses two messages m_0 and m_1 and must identify an encryption of one of them.

EAV security:

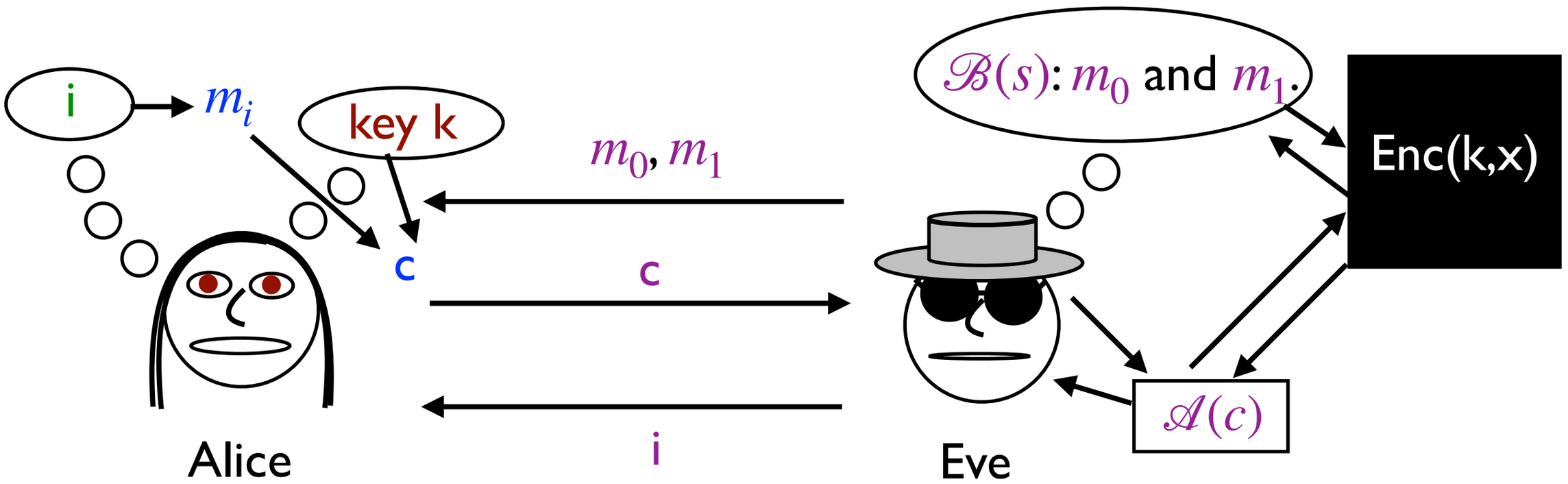


This class is being recorded

Encryption Security Definitions

Eve chooses two messages m_0 and m_1 and must identify an encryption of one of them.

CPA security (private key):

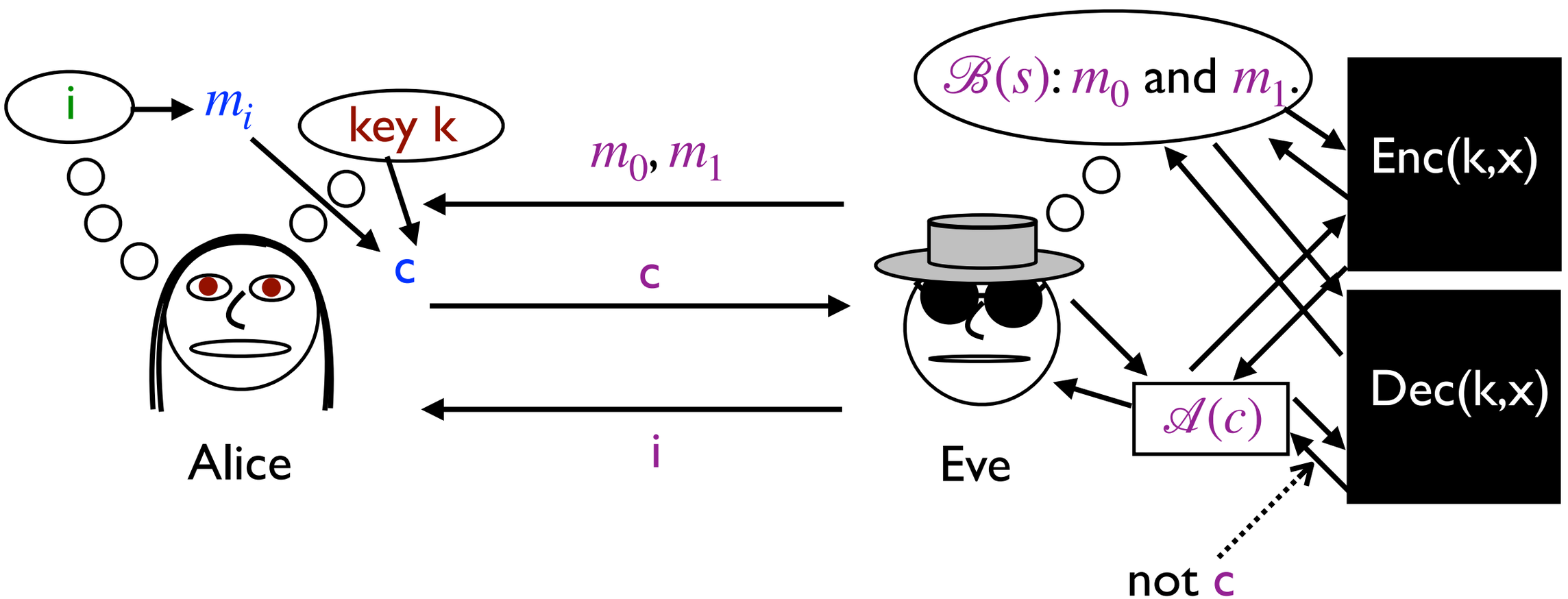


This class is being recorded

Encryption Security Definitions

Eve chooses two messages m_0 and m_1 and must identify an encryption of one of them.

CCA security (private key):

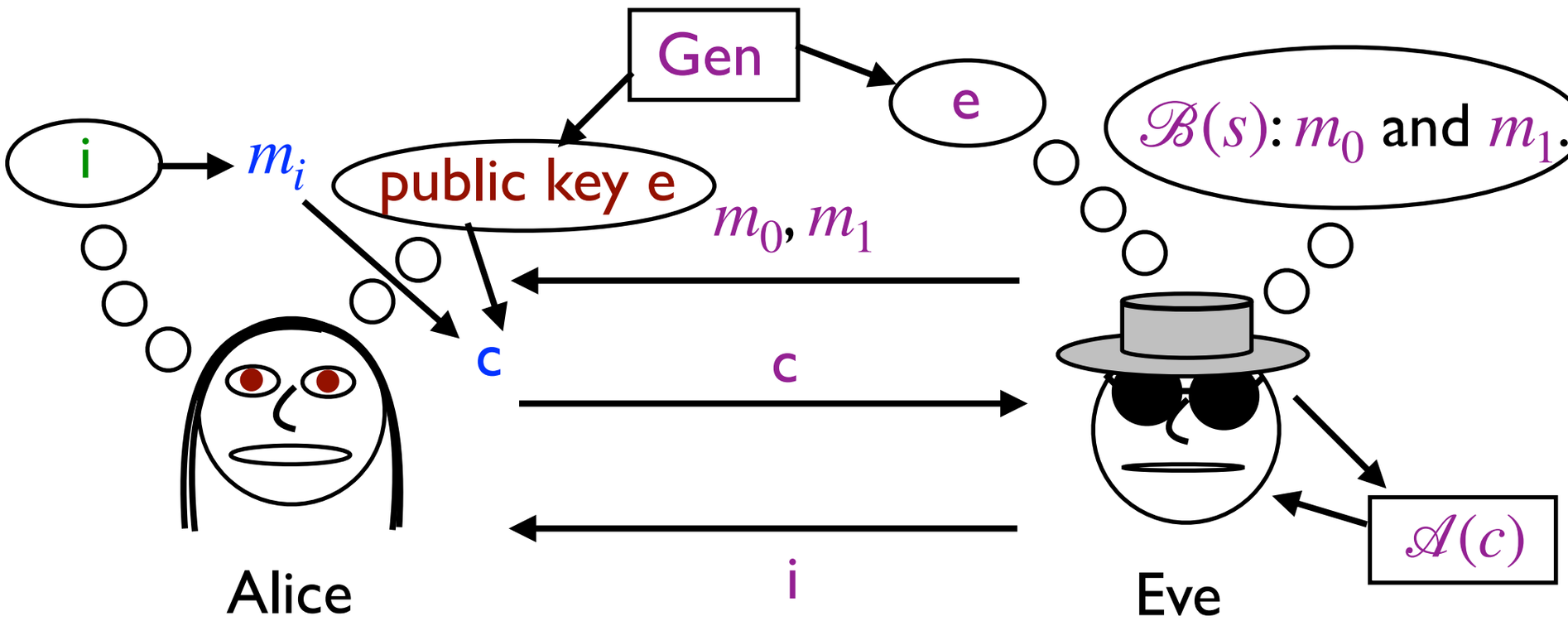


This class is being recorded

Encryption Security Definitions

Eve chooses two messages m_0 and m_1 and must identify an encryption of one of them.

CPA security (public key):

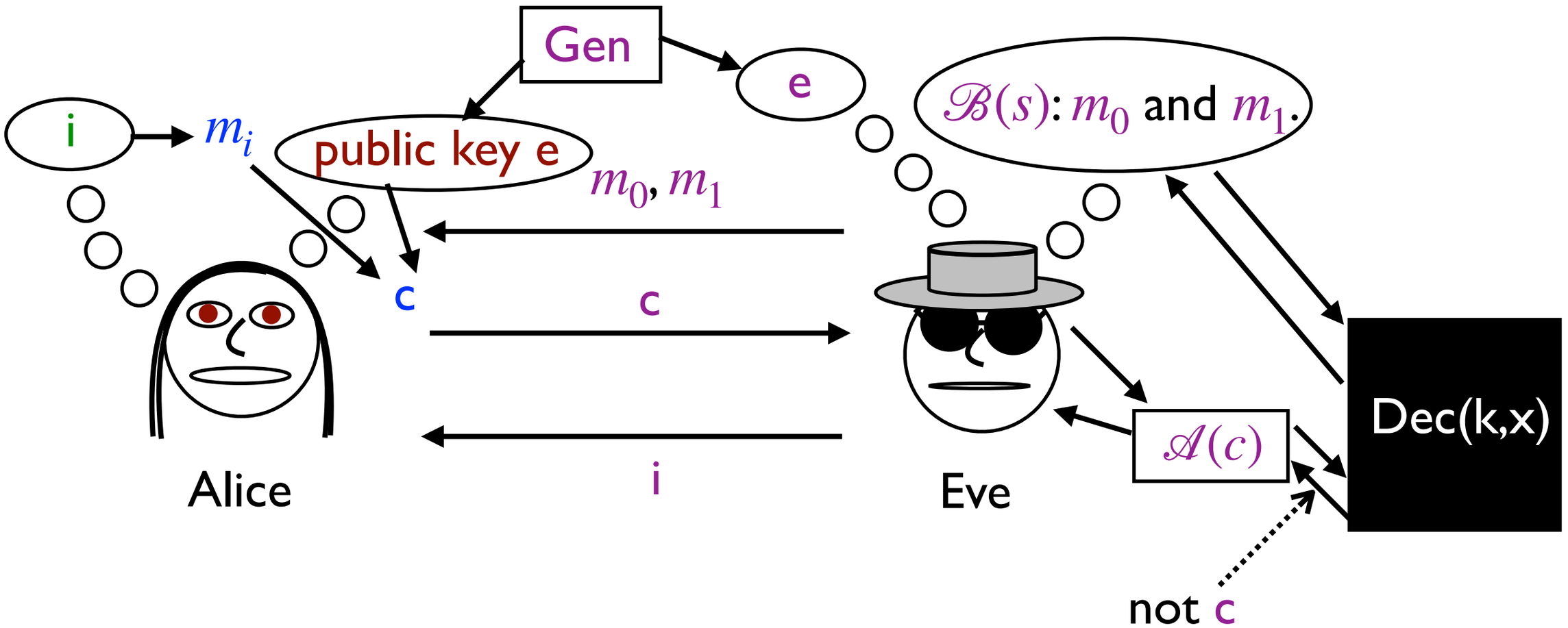


This class is being recorded

Encryption Security Definitions

Eve chooses two messages m_0 and m_1 and must identify an encryption of one of them.

CCA security (public key):

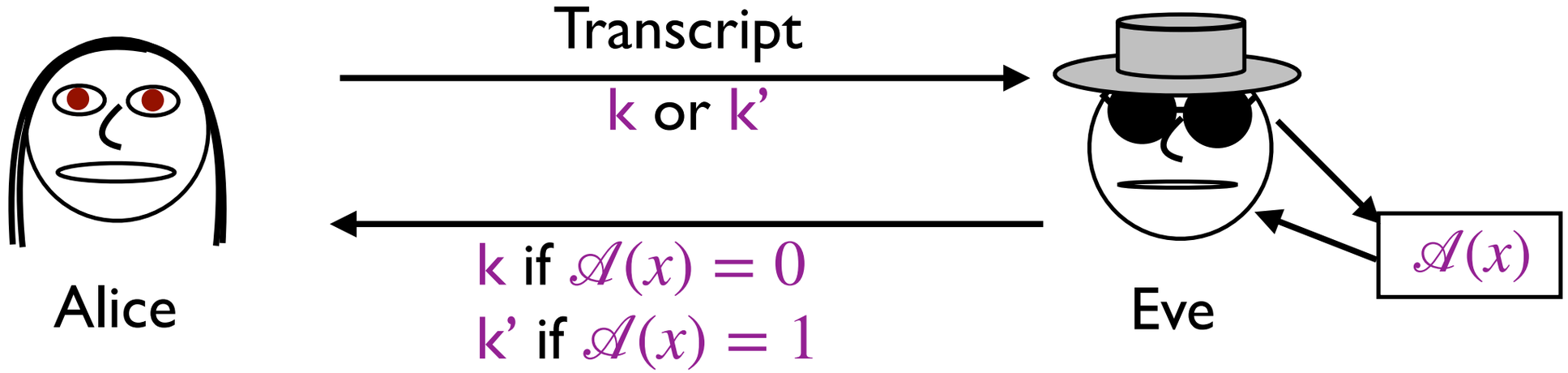


This class is being recorded

Security Definitions for Key Gen.

Eve must distinguish between k generated by the protocol and a uniformly random k'.

Key agreement security:

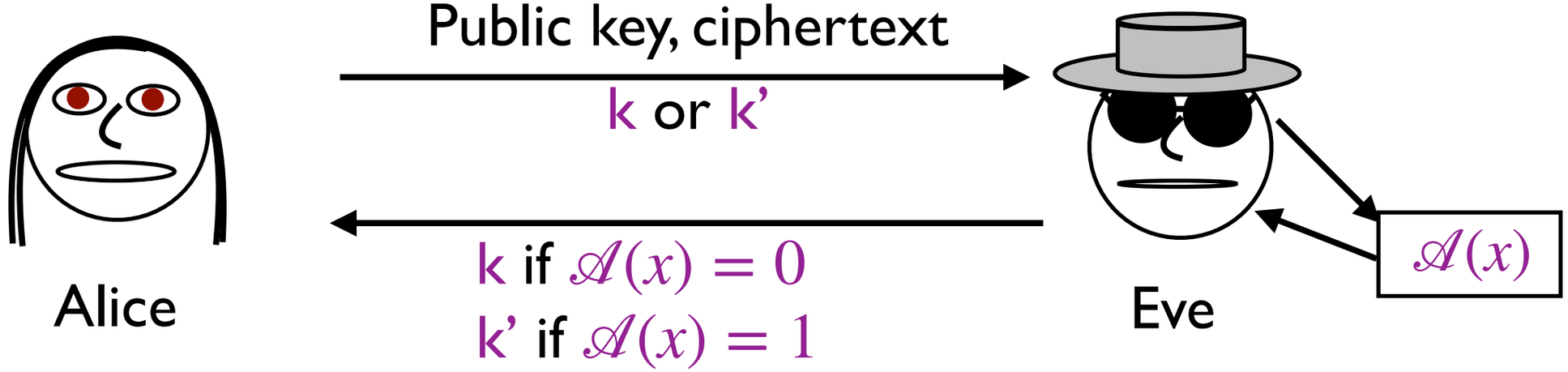


This class is being recorded

Security Definitions for Key Gen.

Eve must distinguish between k generated by the protocol and a uniformly random k'.

KEM CPA security:

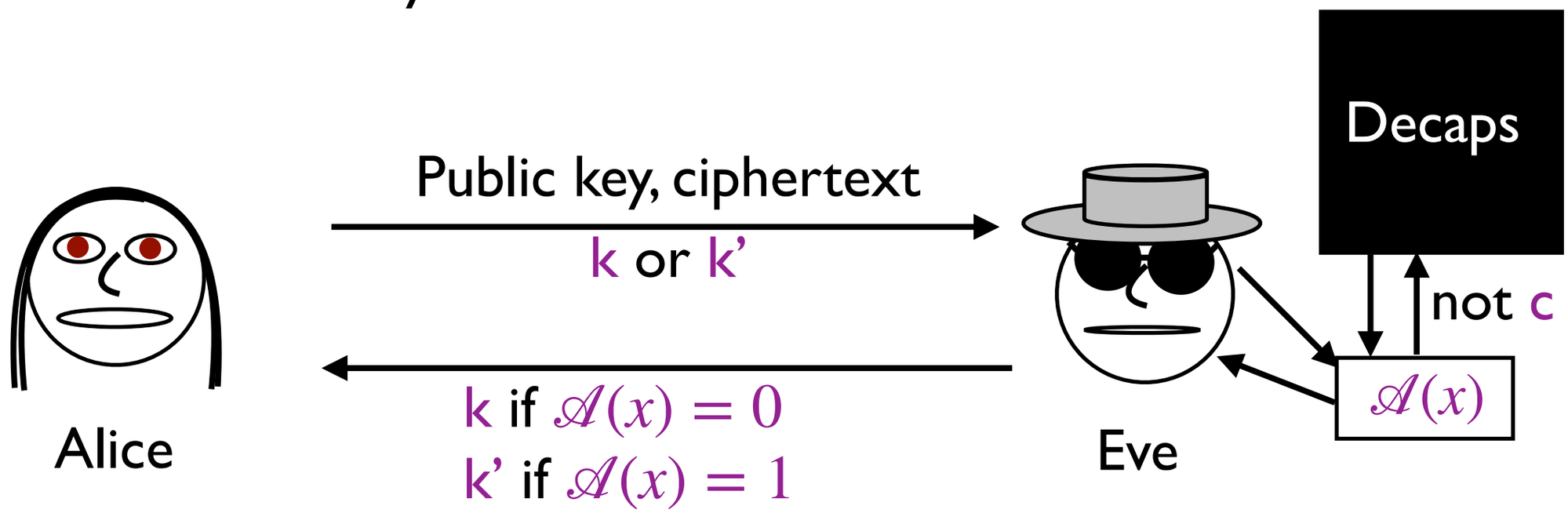


This class is being recorded

Security Definitions for Key Gen.

Eve must distinguish between k generated by the protocol and a uniformly random k'.

KEM CCA security:

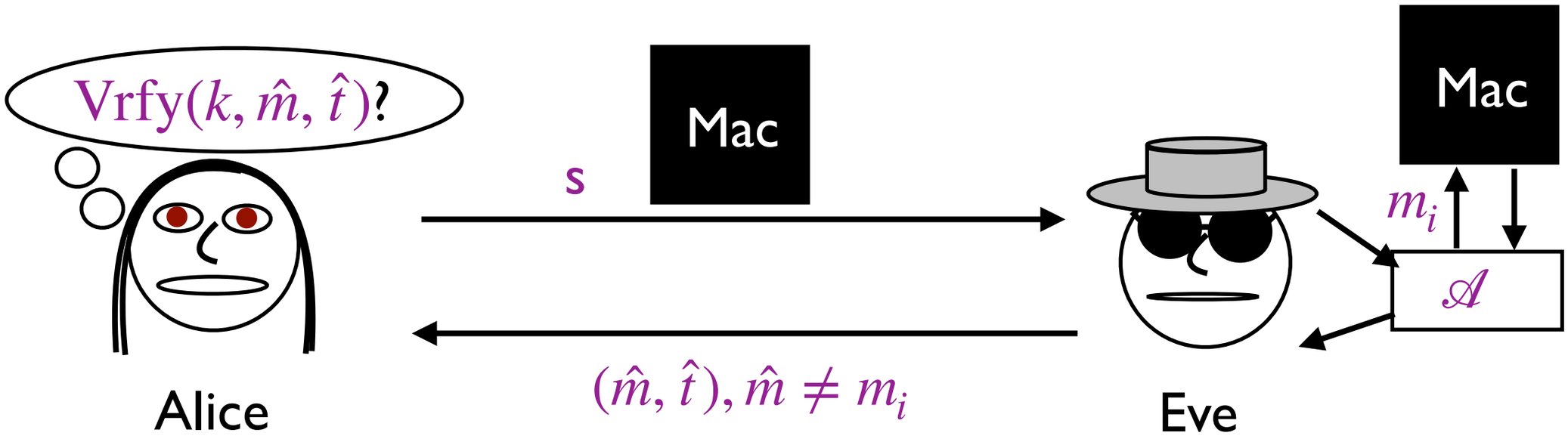


This class is being recorded

Authentication Security Definitions

Eve must forge a message which she hasn't queried to her oracle.

MAC security:

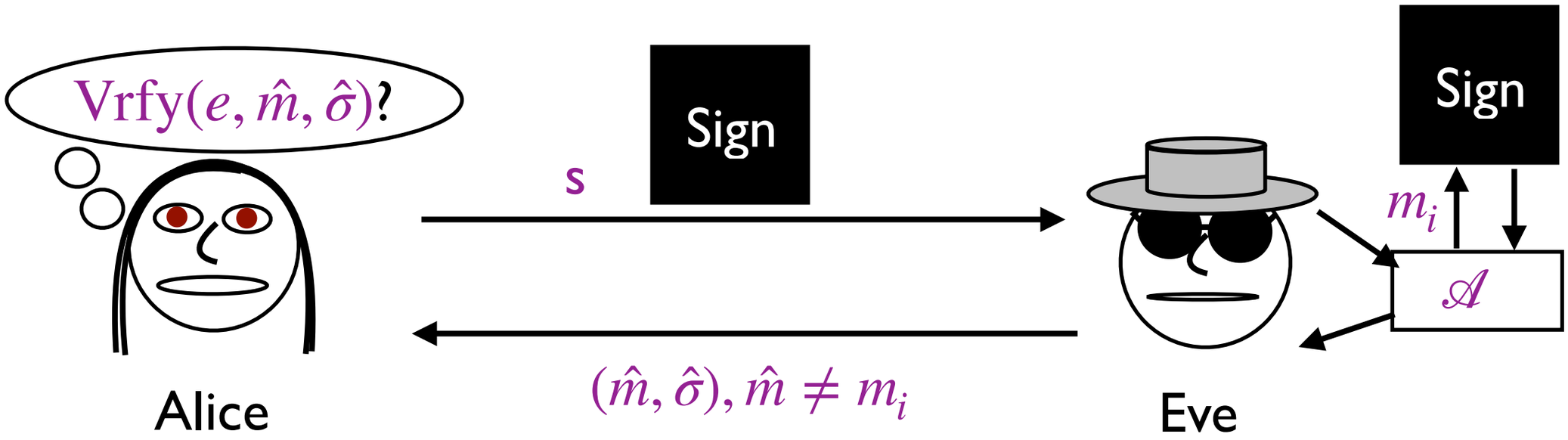


This class is being recorded

Authentication Security Definitions

Eve must forge a message which she hasn't queried to her oracle.

Digital signature security:

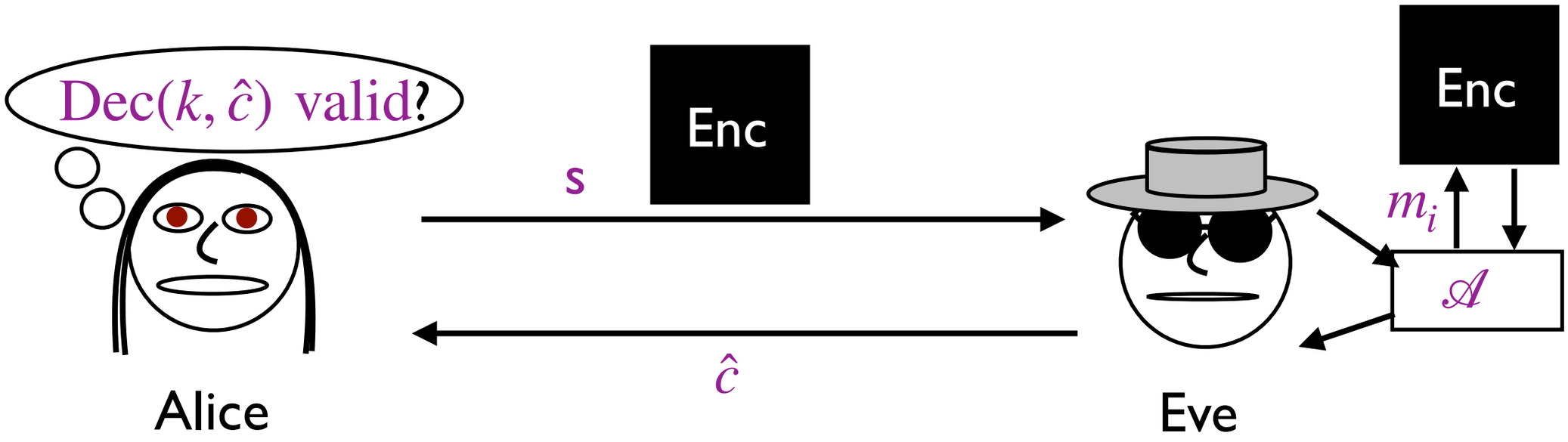


This class is being recorded

Authentication Security Definitions

Eve must forge a message which she hasn't queried to her oracle.

Unforgeability (for encryption):



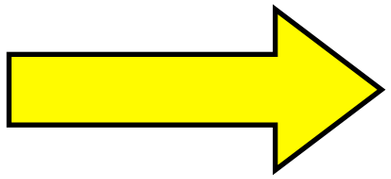
Recall that authenticated encryption is CCA security plus unforgeability.

This class is being recorded

General Encryption Constructions

EAV security:

Pseudorandom generator G(s)

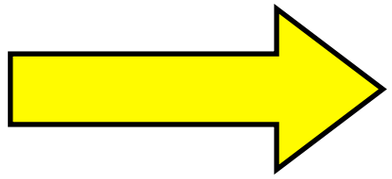


Pseudo one-time pad

$c = G(k) \oplus m$

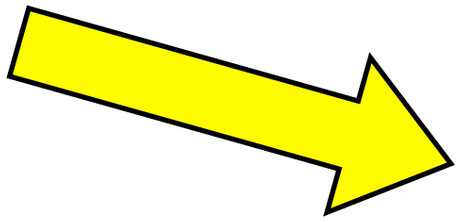
CPA security:

Pseudorandom function $F_k(r)$

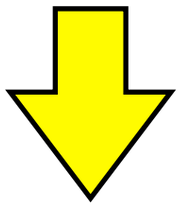


$(r, F_k(r) \oplus m)$

Need random IV r to avoid repeating ciphertext

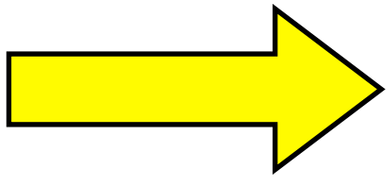


CBC mode or CTR mode for longer messages



CCA security/AE:

MAC

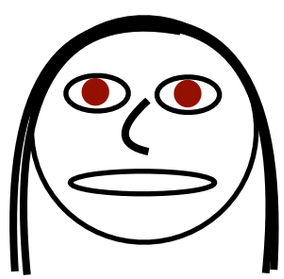


Encrypt then authenticate

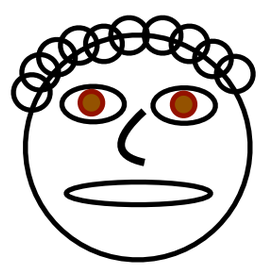
This class is being recorded

KEM/DEM

Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:



Alice

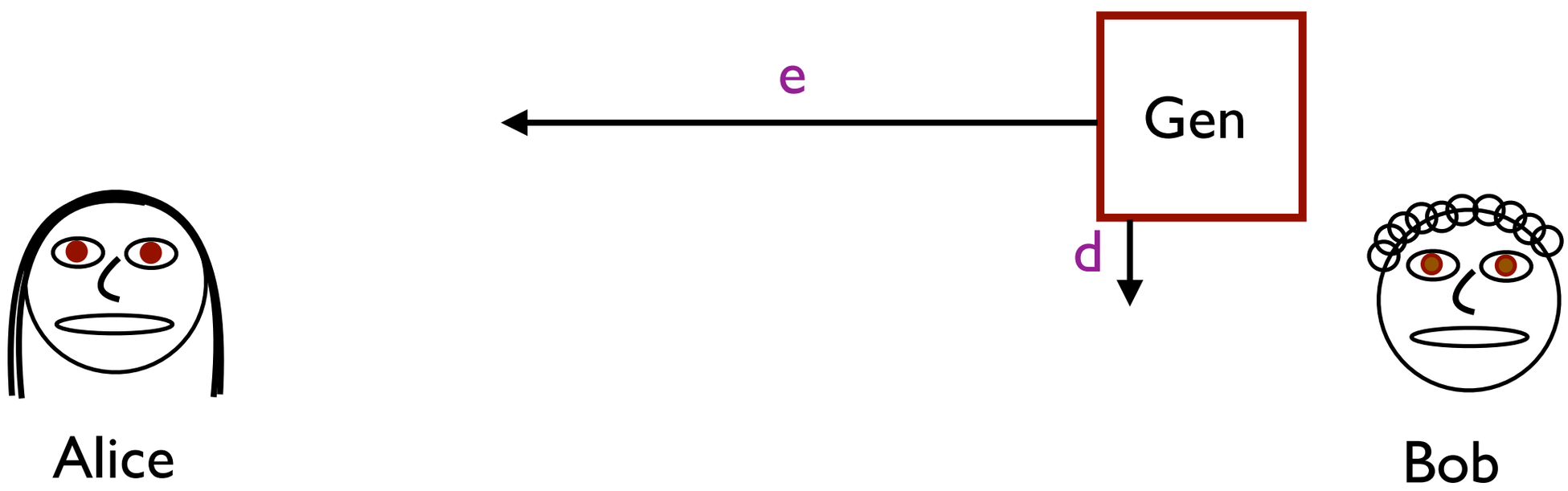


Bob

This class is being recorded

KEM/DEM

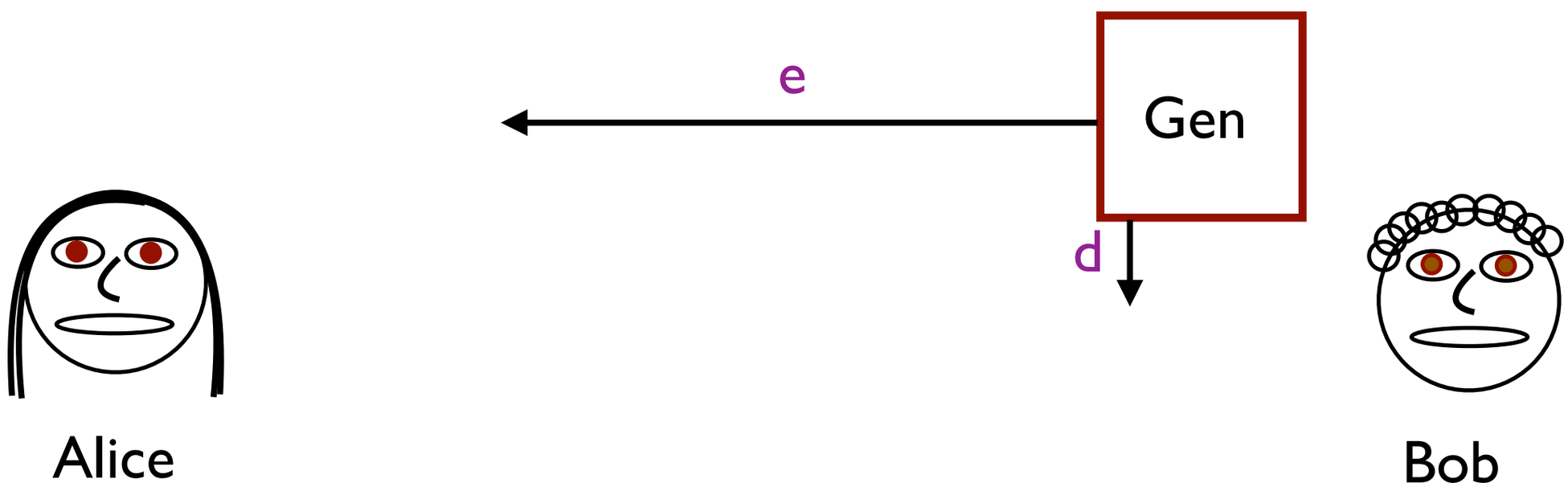
Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:



This class is being recorded

KEM/DEM

Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:

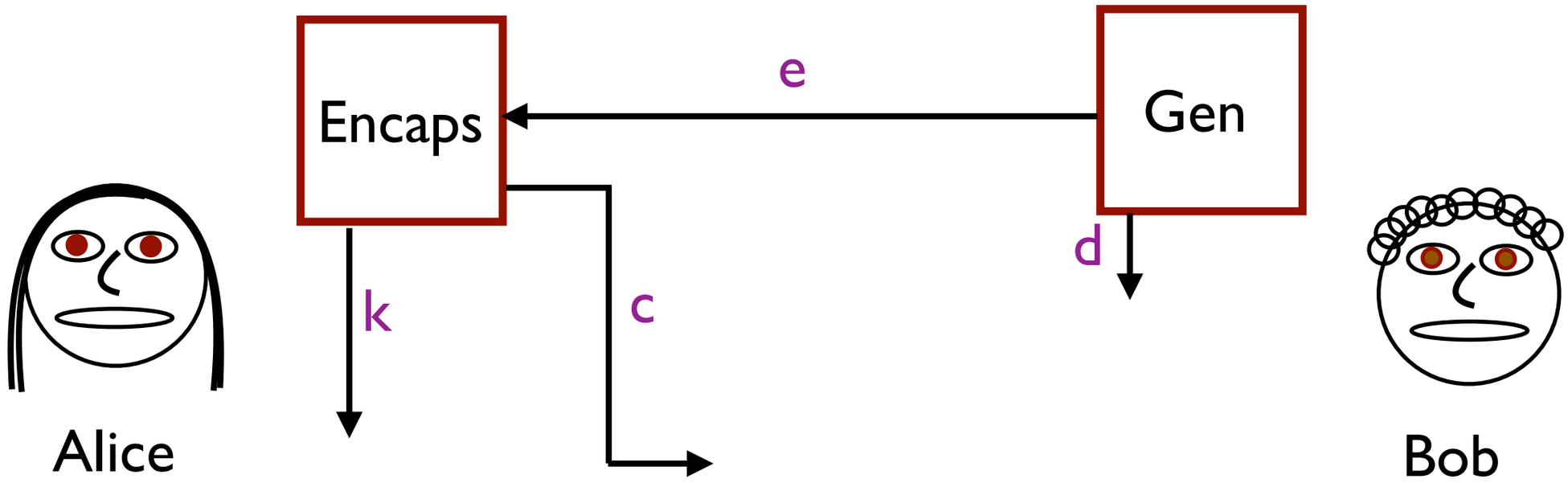


message m

This class is being recorded

KEM/DEM

Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:

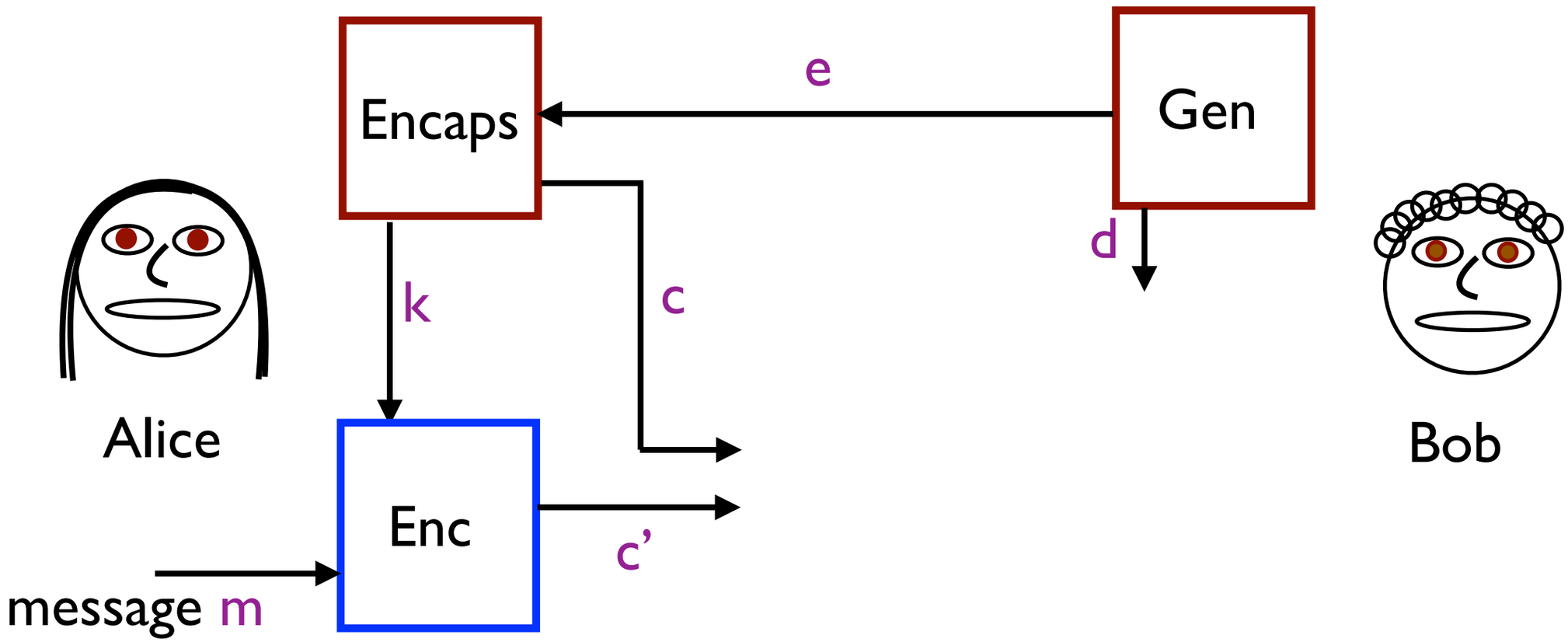


message m

This class is being recorded

KEM/DEM

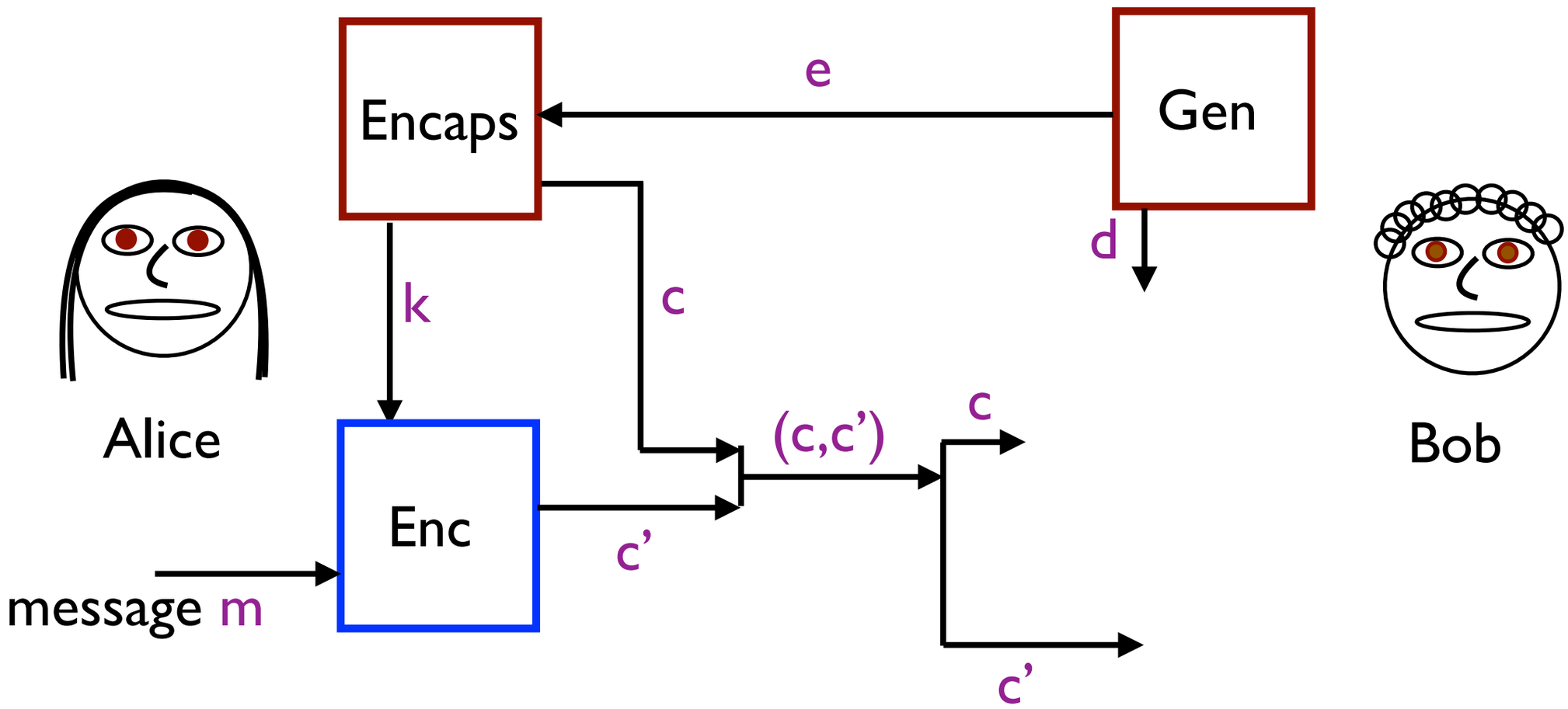
Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:



This class is being recorded

KEM/DEM

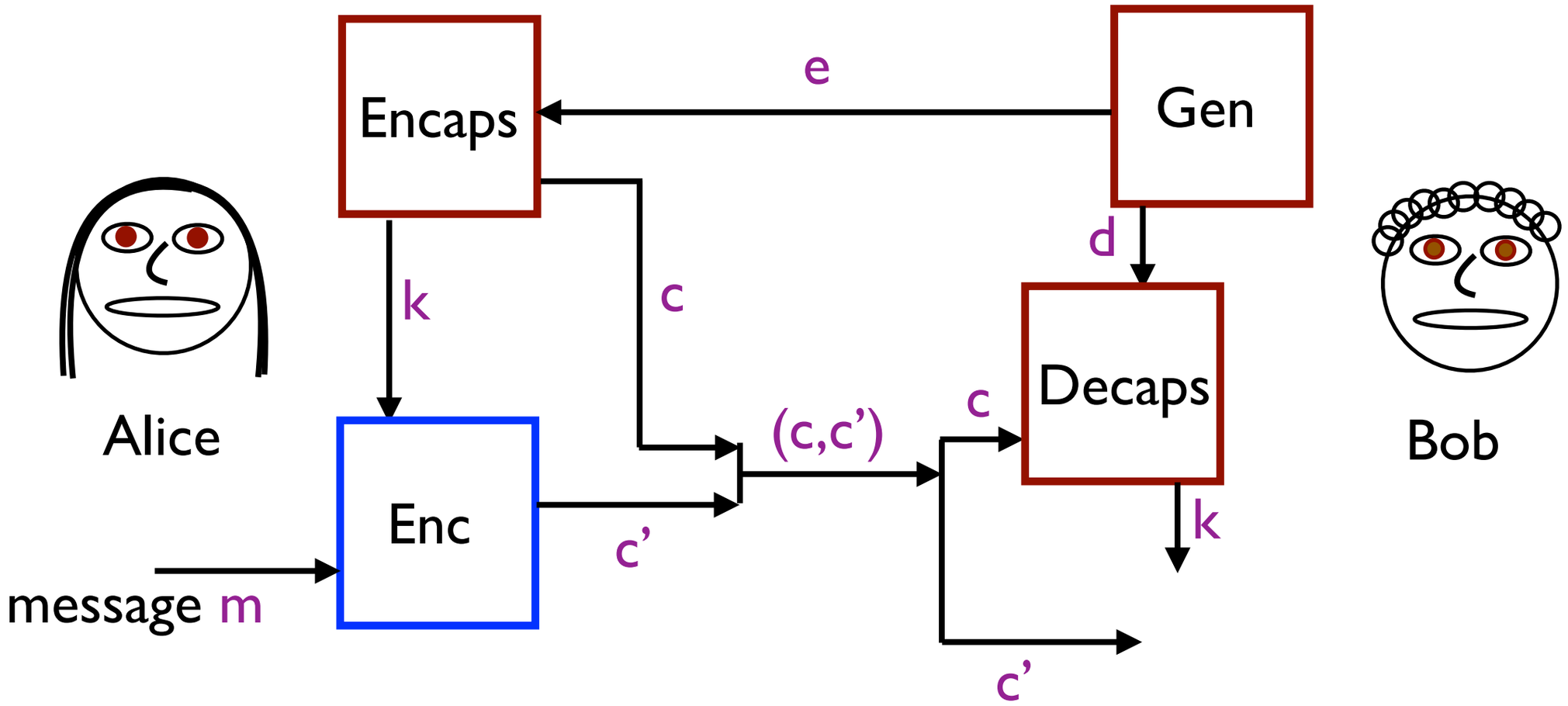
Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:



This class is being recorded

KEM/DEM

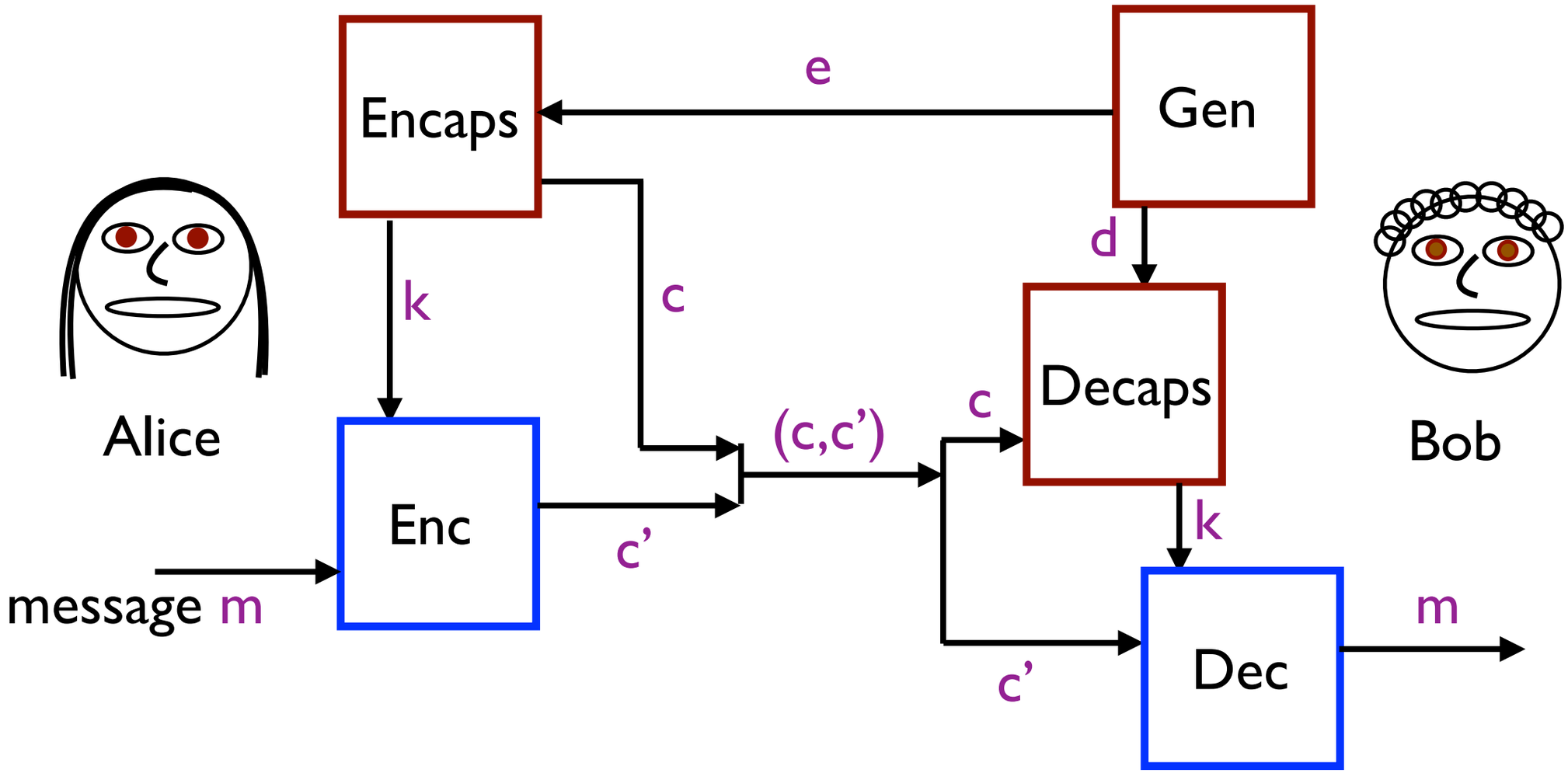
Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:



This class is being recorded

KEM/DEM

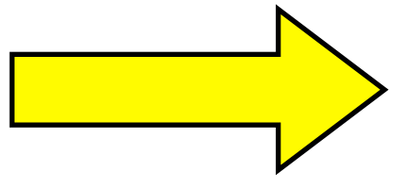
Using a KEM, we can effectively upgrade these private-key encryption protocols into public-key encryption protocols:



This class is being recorded

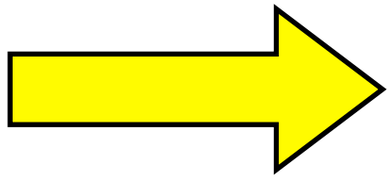
Generic MAC Constructions

Pseudorandom function $F_k(r)$



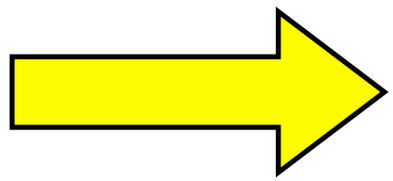
$\mathrm{Mac}(k, m) = F_k(m)$

CBC-Mac



For longer messages; no IV, include length as first input, tag is only output of last block

Hash-and-Mac

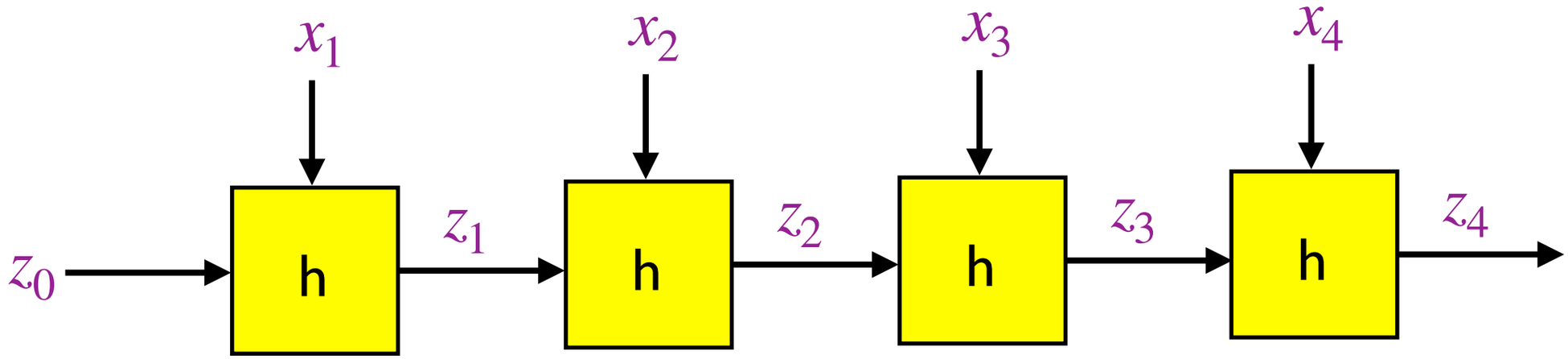


$\mathrm{Mac}(k, H(m))$

This class is being recorded

Hash function construction

Merkle-Damgard construction makes hash functions for arbitrary input out of a compression function of fixed size.



Need to pad the input appropriately.

This class is being recorded

DH Encryption and Signature

Diffie-Hellman: Alice sends $A = g^a \bmod p$, Bob sends $B = g^b \bmod p$, key is $A^b = B^a = g^{ab} \bmod p$.

El Gamal: Public key is $y = g^b \bmod p$, private key is b, encryption is $my^a \bmod p$ (for secret random a).

DH KEM: Public key is $y = g^b \bmod p$, private key is b, ciphertext is $A = g^a \bmod p$, key is $H(A^b) = H(y^a) = H(g^{ab}) \bmod p$.

DSA: Public key is $y = g^b \bmod p$, private key is b. Signature is (r,s): $s = k^{-1}(H(m) + br) \bmod q$ for random k, $r = g^k \bmod p$.

(Verify by checking that $r = g^{H(m)s^{-1}} y^{rs^{-1}} \bmod p$)

This class is being recorded

RSA encryption and signatures

RSA encryption: Public key is (N, e), private key is d such that $de = 1 \bmod \varphi(N)$. Encryption is $\tilde{m}^e \bmod N$ (with m appropriately padded to $\tilde{m}$).

RSA KEM: Public key is (N, e), private key is d such that $de = 1 \bmod \varphi(N)$. Encryption is $x^e \bmod N$, key is $H(x)$.

RSA signature: Public key is (N, e), private key is d such that $de = 1 \bmod \varphi(N)$. Signature is $\sigma = H(m)^d \bmod N$.

This class is being recorded

Look for Similarities

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

This class is being recorded

Look for Similarities

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

This looks like RSA!

This class is being recorded

Look for Similarities

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

This looks like RSA!

But the difference is that the encryption exponent a is chosen randomly each time and hidden (sort of) in the ciphertext.

This class is being recorded

Look for Similarities

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

This looks like RSA!

But the difference is that the encryption exponent a is chosen randomly each time and hidden (sort of) in the ciphertext.

There are also these extra parameters x and y in the public key.

This class is being recorded

Look for Similarities

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

b) Once Bob knows a, what should he do to find m? This decryption procedure can fail for certain values of a. When does it fail?

This class is being recorded

Look for Similarities

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

b) Once Bob knows a, what should he do to find m? This decryption procedure can fail for certain values of a. When does it fail?

Since Gen and Enc are similar to RSA, it makes sense Dec should be similar as well. a plays the role of the encryption key e in RSA. But then he can determine d as in RSA, by using Euclid's algorithm so that $ad - k\varphi(N) = 1$, but only if $\gcd(a, \varphi(N)) = 1$.

This class is being recorded

Look for Differences

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

c) This protocol is also not secure because Eve can learn the private key. How can she do that?

This class is being recorded

Look for Differences

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

c) This protocol is also not secure because Eve can learn the private key. How can she do that?

RSA is secure (with padding) but this is not. It must be because of one of the differences from RSA. Maybe it is getting x and y?

This class is being recorded

Look for Differences

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

c) This protocol is also not secure because Eve can learn the private key. How can she do that?

RSA is secure (with padding) but this is not. It must be because of one of the differences from RSA. Maybe it is getting x and y?

Indeed, x and y have a common factor with N. For example, $\gcd(N, x) = p$. This is something Eve can calculate to factor N.

This class is being recorded

Remember Restrictions

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

a) How can Bob find a as part of his Dec algorithm?

This class is being recorded

Remember Restrictions

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

a) How can Bob find a as part of his Dec algorithm?

x and y are part of the public key, so the "obvious" way to find a is just to divide ax by x. But remember, division mod N is only defined if $\gcd(x, N) = 1$, which is not true here.

This class is being recorded

Remember Restrictions

From midterm #2:

Gen: Bob picks 2 random large primes p and q, two random integers r and s. Let $N = pq$, $x = rp \bmod N$, and $y = sq \bmod N$. The public key is then (N, x,y) and the private key is (p, q).

Enc: Alice chooses a random integer a with $0 < a < N$. Then she pads the message m as in RSA to get $\tilde{m}$ and sends the ciphertext $c = (ax \bmod N, ay \bmod N, \tilde{m}^a \bmod N)$.

a) How can Bob find a as part of his Dec algorithm?

x and y are part of the public key, so the "obvious" way to find a is just to divide ax by x. But remember, division mod N is only defined if $\gcd(x, N) = 1$, which is not true here.

We need to find another way, and either Euclid's algorithm or the Chinese remainder theorem will work here.

This class is being recorded

Read Questions Carefully

From midterm #1 (True/False):

> Given that there are about 30,000 undergraduates at UMD and assuming that each student has exactly 1 best friend who is also a UMD undergraduate (and the feeling is mutual), then the probability that a particular party with 60 students contains a pair of best friends is small.

This class is being recorded

Read Questions Carefully

From midterm #1 (True/False):

> Given that there are about 30,000 undergraduates at UMD and assuming that each student has exactly 1 best friend who is also a UMD undergraduate (and the feeling is mutual), then the probability that a particular party with 60 students contains a pair of best friends is small.

Notice: This question does *not* say that the students go to parties independently.

This class is being recorded

Read Questions Carefully

From midterm #1 (True/False):

> Given that there are about 30,000 undergraduates at UMD and assuming that each student has exactly 1 best friend who is also a UMD undergraduate (and the feeling is mutual), then the probability that a particular party with 60 students contains a pair of best friends is small.

Notice: This question does *not* say that the students go to parties independently.

> If they are independent, the birthday paradox would say $60^2 = 3600 < 30000$, so being at the same party is unlikely.

This class is being recorded

Read Questions Carefully

From midterm #1 (True/False):

> Given that there are about 30,000 undergraduates at UMD and assuming that each student has exactly 1 best friend who is also a UMD undergraduate (and the feeling is mutual), then the probability that a particular party with 60 students contains a pair of best friends is small.

Notice: This question does *not* say that the students go to parties independently.

> If they are independent, the birthday paradox would say $60^2 = 3600 < 30000$, so being at the same party is unlikely.

> But it is pretty likely that best friends would go to the same party! Party attendance is correlated.

This class is being recorded

Read Questions Carefully

From midterm #2:

> Find possible values of N and give an algorithm to compute the quantity: The unique value of $1 < x < N$ such that $x^{31} = 1 \bmod N$, assuming $N > 31$. (Note that 31 is prime.)

This class is being recorded

Read Questions Carefully

From midterm #2:

> Find possible values of N and give an algorithm to compute the quantity: The unique value of $1 < x < N$ such that $x^{31} = 1 \bmod N$, assuming $N > 31$. (Note that 31 is prime.)

When can we have $x^{31} = 1 \bmod N$?

This class is being recorded

Read Questions Carefully

From midterm #2:

> Find possible values of N and give an algorithm to compute the quantity: The unique value of $1 < x < N$ such that $x^{31} = 1 \bmod N$, assuming $N > 31$. (Note that 31 is prime.)

When can we have $x^{31} = 1 \bmod N$?

> Euler-Fermat theorem says that $x^{\varphi(N)} = 1 \bmod N$, so we need $31 \mid \varphi(N)$.

This class is being recorded

Read Questions Carefully

From midterm #2:

> Find possible values of N and give an algorithm to compute the quantity: The unique value of $1 < x < N$ such that $x^{31} = 1 \bmod N$, assuming $N > 31$. (Note that 31 is prime.)

When can we have $x^{31} = 1 \bmod N$?

> Euler-Fermat theorem says that $x^{\varphi(N)} = 1 \bmod N$, so we need $31 \mid \varphi(N)$.

Note: It is *necessary* that $31 \mid \varphi(N)$ but it may not be *sufficient*. That is, it might be that there are some values of N such that $31 \mid \varphi(N)$ but there are no elements of order 31. As it happens, there *is* always an element of order 31.

This class is being recorded

Read Questions Carefully

From midterm #2:

> Find possible values of N and give an algorithm to compute the quantity: The unique value of $1 < x < N$ such that $x^{31} = 1 \bmod N$, assuming $N > 31$. (Note that 31 is prime.)

When can we have $x^{31} = 1 \bmod N$?

> Euler-Fermat theorem says that $x^{\varphi(N)} = 1 \bmod N$, so we need $31 \mid \varphi(N)$.

Note: It is *necessary* that $31 \mid \varphi(N)$ but it may not be *sufficient*. That is, it might be that there are some values of N such that $31 \mid \varphi(N)$ but there are no elements of order 31. As it happens, there *is* always an element of order 31.

This class is being recorded

Read Questions Carefully

From midterm #2:

> Find possible values of N and give an algorithm to compute the quantity: The unique value of $1 < x < N$ such that $x^{31} = 1 \bmod N$, assuming $N > 31$. (Note that 31 is prime.)

When can we have $x^{31} = 1 \bmod N$?

> Euler-Fermat theorem says that $x^{\varphi(N)} = 1 \bmod N$, so we need $31 \mid \varphi(N)$.
>
> Note: It is *necessary* that $31 \mid \varphi(N)$ but it may not be *sufficient*. That is, it might be that there are some values of N such that $31 \mid \varphi(N)$ but there are no elements of order 31. As it happens, there *is* always an element of order 31.

But any element of order 31 *cannot* be unique!

This class is being recorded

Read Questions Carefully

From midterm #2:

> Find possible values of N and give an algorithm to compute the quantity: The unique value of $1 < x < N$ such that $x^{31} = 1 \bmod N$, assuming $N > 31$. (Note that 31 is prime.)

When can we have $x^{31} = 1 \bmod N$?

> Euler-Fermat theorem says that $x^{\varphi(N)} = 1 \bmod N$, so we need $31 \mid \varphi(N)$.
>
> Note: It is *necessary* that $31 \mid \varphi(N)$ but it may not be *sufficient*. That is, it might be that there are some values of N such that $31 \mid \varphi(N)$ but there are no elements of order 31. As it happens, there *is* always an element of order 31.

But any element of order 31 *cannot* be unique!

> Because $y = x^a \bmod N$ *also* has order 31 unless $31 \mid a$.

This class is being recorded

Class Evaluations

Please fill out your course evaluations.

I am continuing to tweak the class, so your feedback is helpful.

I am particularly interested in understanding how the new grading method (less emphasis on problem sets, 2 midterm exams) worked for you.

This class is being recorded