# Resources in Cryptography

Ed Blakey

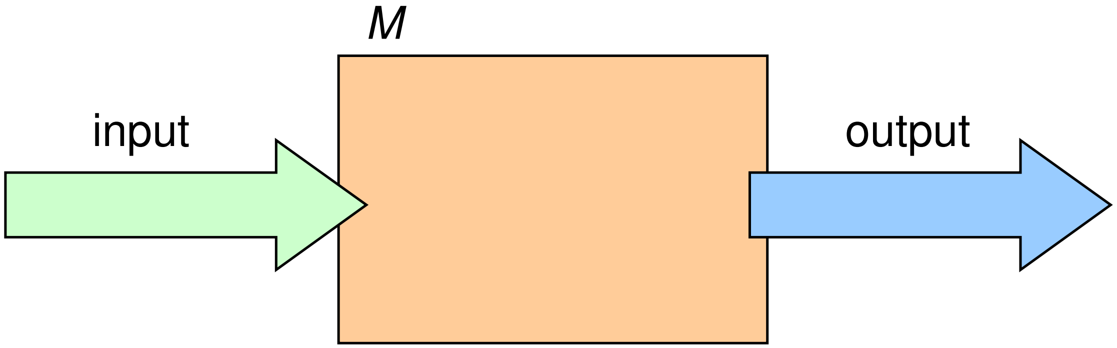
ed.blakey@queens.oxon.org
http://users.ox.ac.uk/~quee1871/

***Information Security as a Resource***
13.x.2011

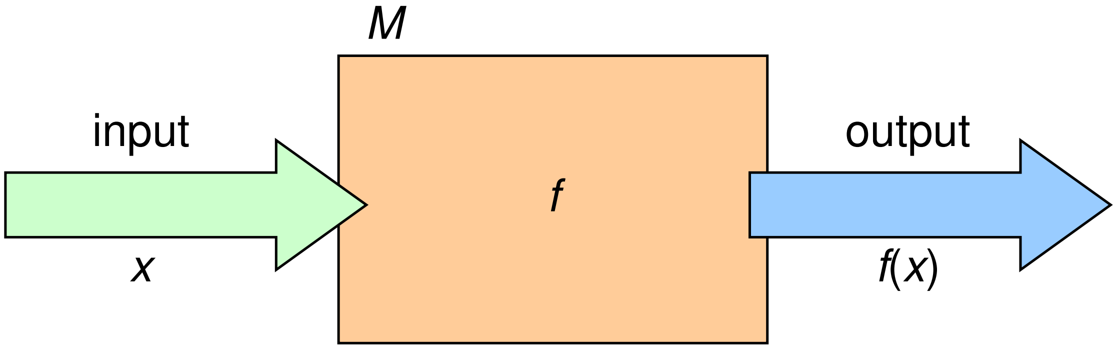
Oxford University Computer Science Department

# Disclaimer!

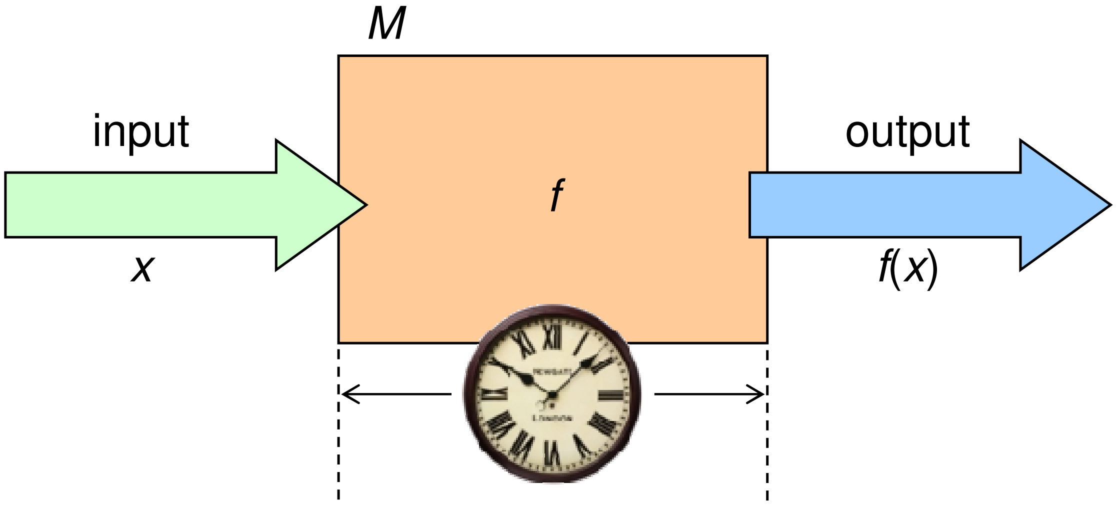
# Computational complexity.



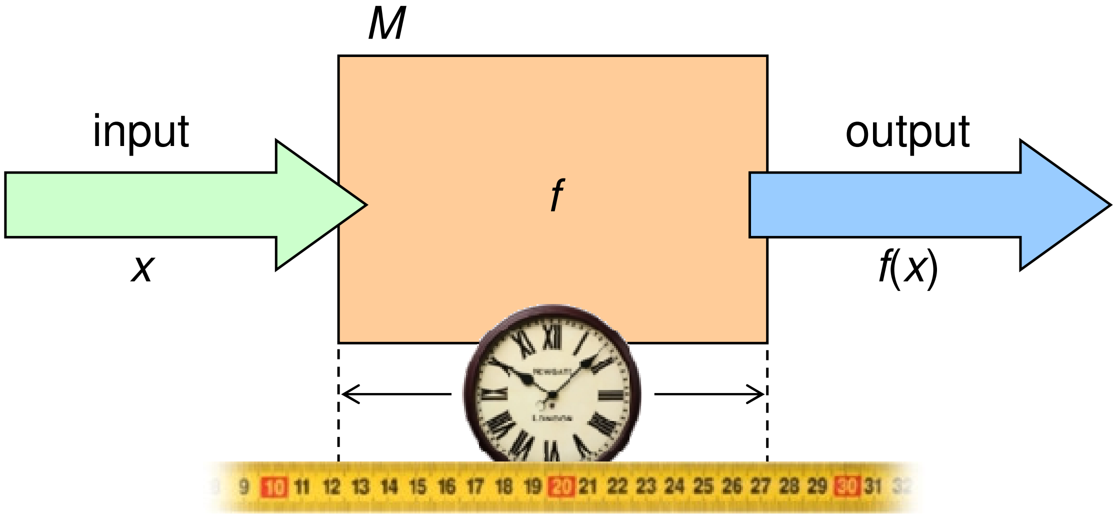
# Computational complexity.



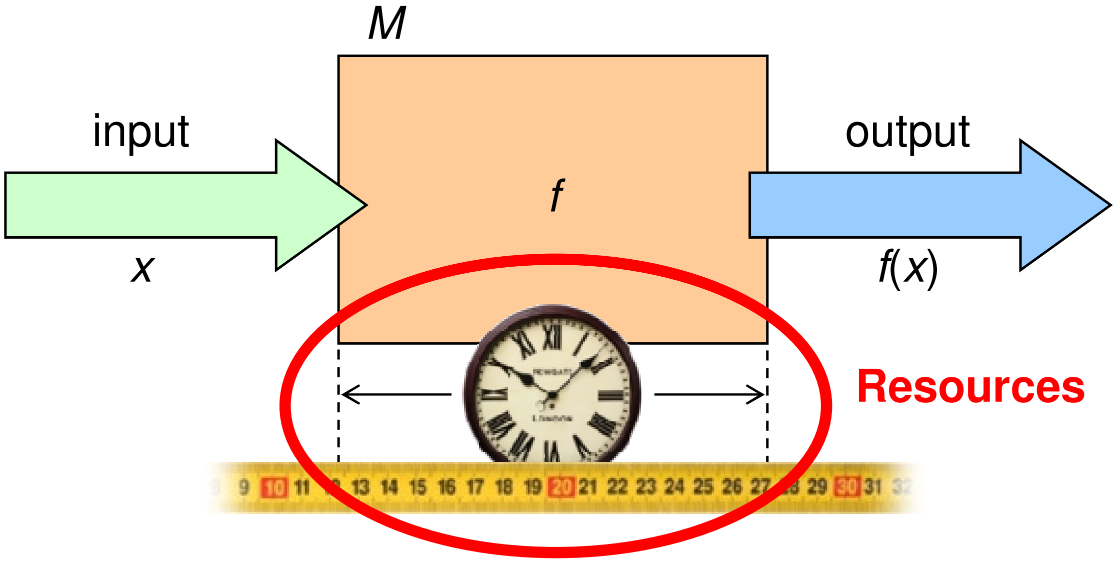
# Computational complexity.



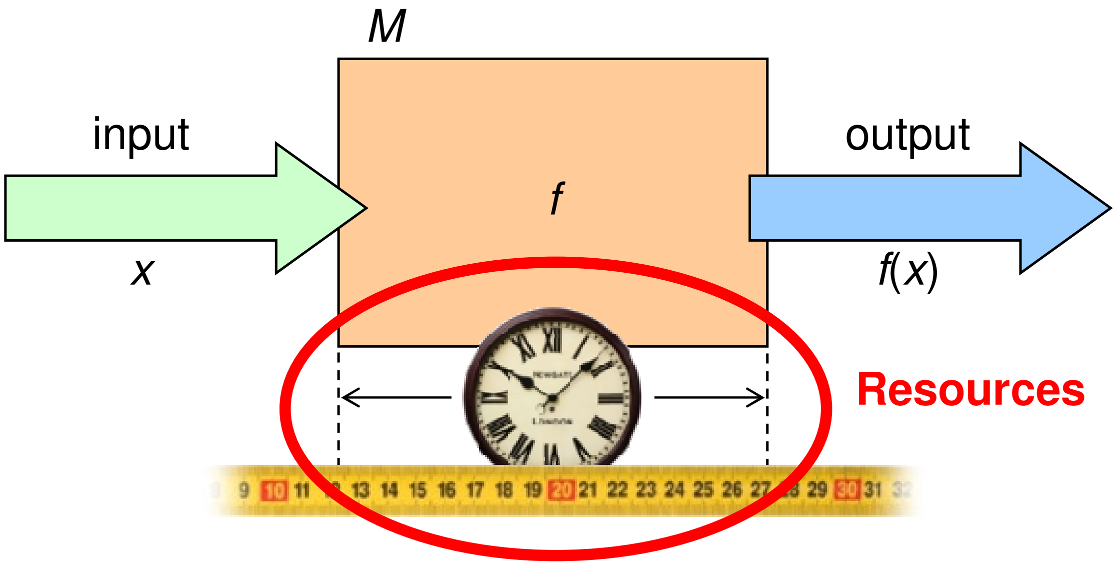
# Computational complexity.



# Computational complexity.

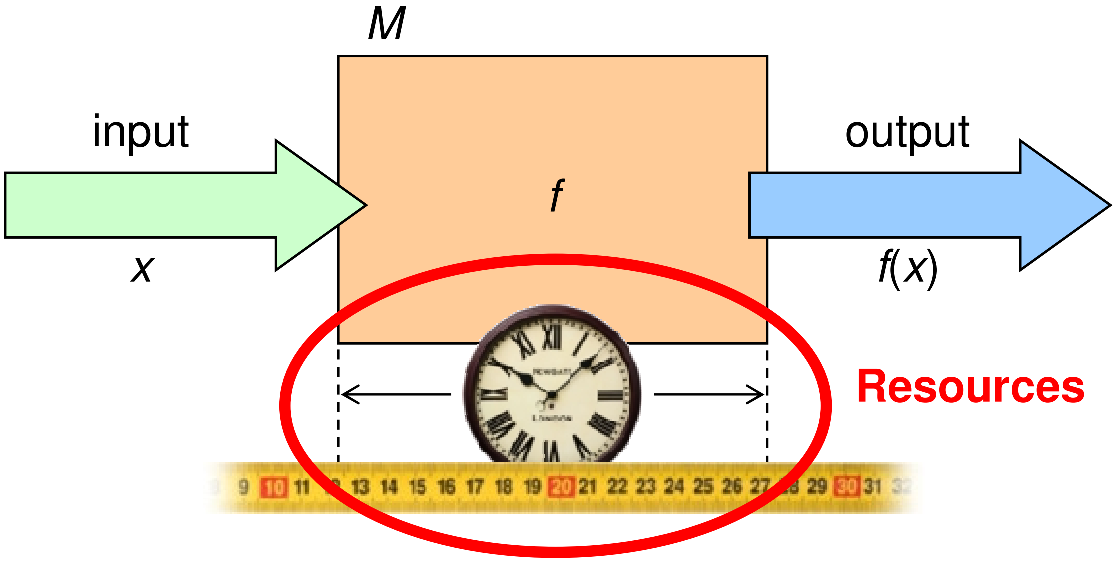


# Computational complexity.



**Complexity**: how resources *scale* with respect to $|x|$.

# Computational complexity.



**Complexity**: how resources *scale* with respect to |*x*|.

Says something: (directly) about *efficiency* of *M*, and
(indirectly) about *difficulty* of computing *f*.

# Standard resources.

**time** and **space**

## Standard resources.

**time** and **space**

NP

P

EXP

AC$^0$

coNP

PH

BPP

PSPACE

L

P/poly

BQP

NC

PP

# Standard resources.

**time** and **space**

non-determinism

NP

P

EXP

$AC^0$

coNP

PH

BPP

PSPACE

L

P/poly

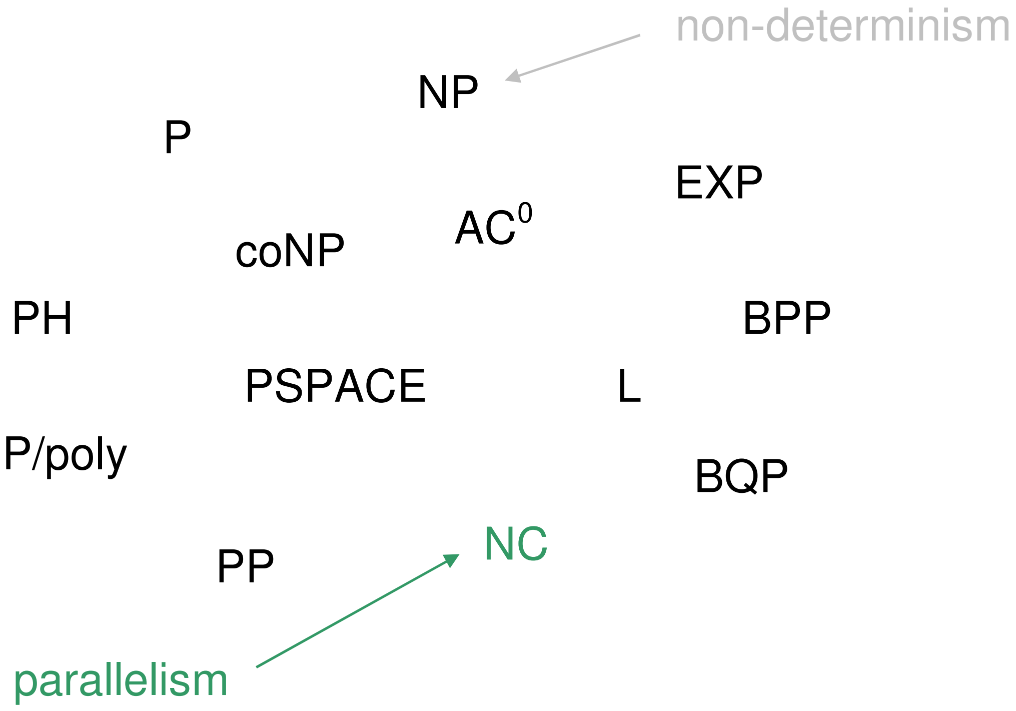
BQP

NC

PP

# Standard resources.

**time** and **space**



non-determinism

NP

P

EXP

$AC^0$

coNP

PH

BPP

PSPACE

L

P/poly

BQP

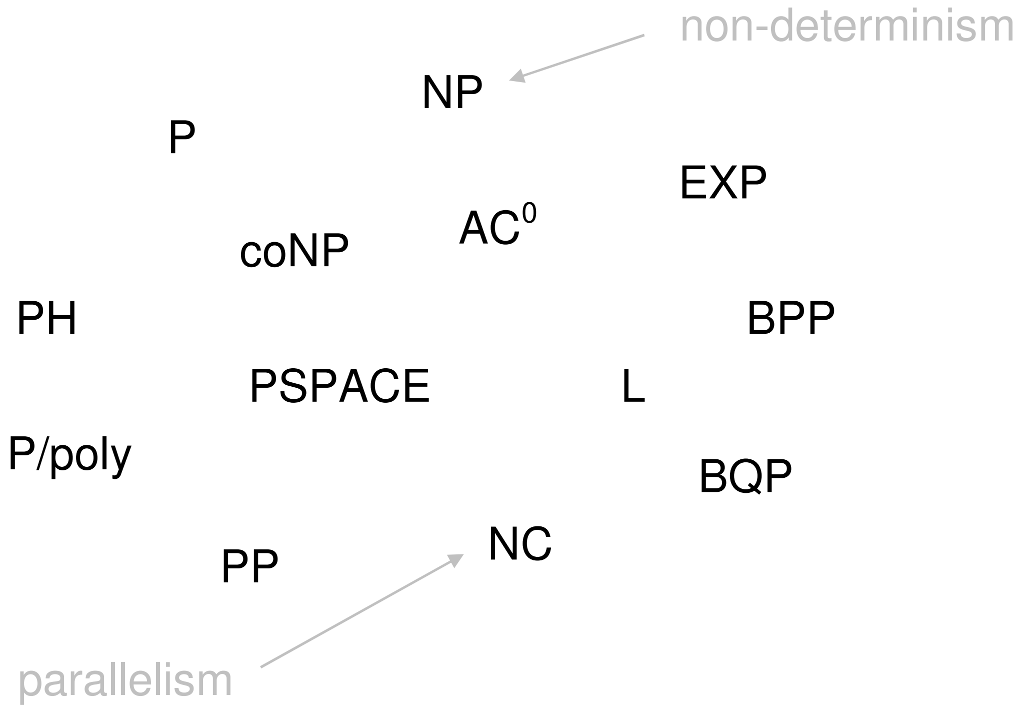
NC

PP

parallelism

## Standard resources.

***Bounds*** in terms of **time** and **space**.



non-determinism

NP

P

EXP

$AC^0$

coNP

PH

BPP

PSPACE

L

P/poly

BQP

NC

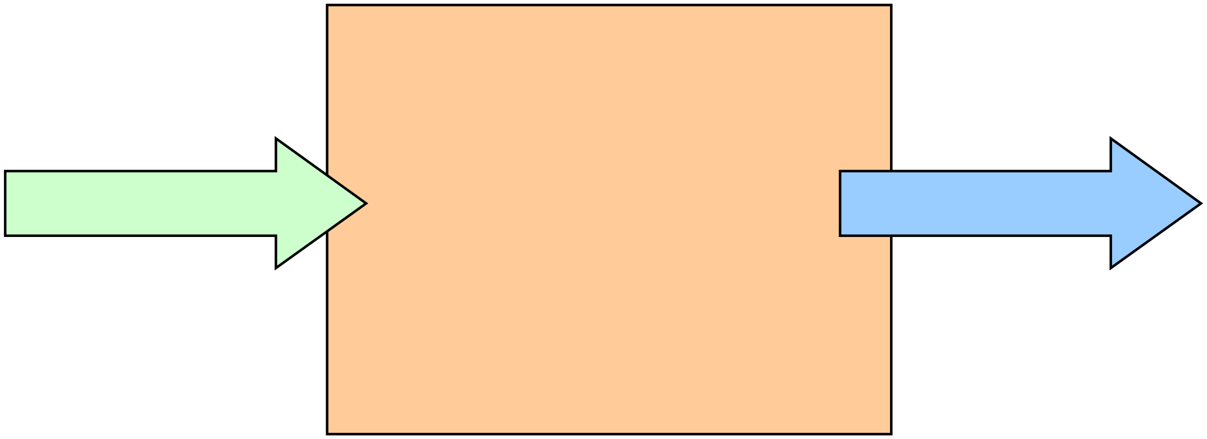
PP

parallelism

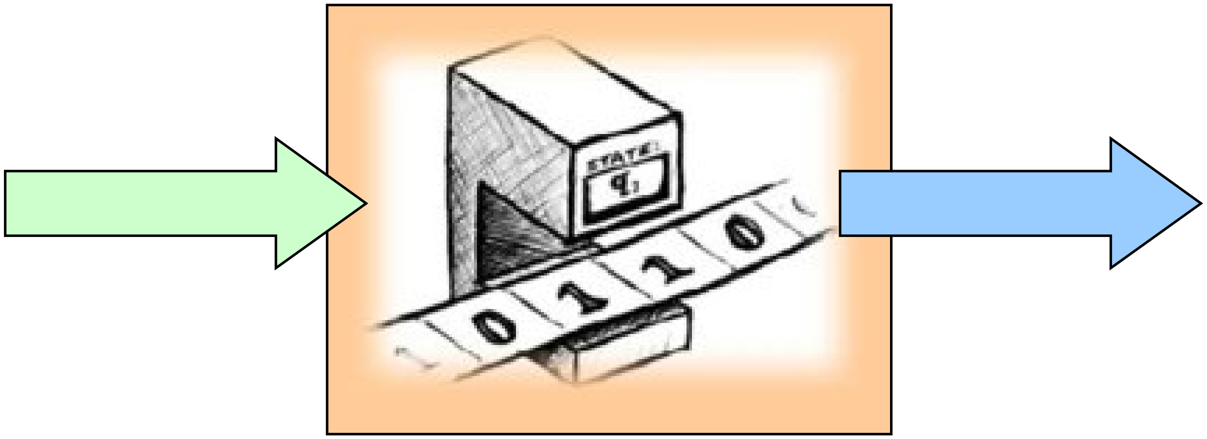
# Non-standard resources

# Non-standard resources
# e.g., *precision*.

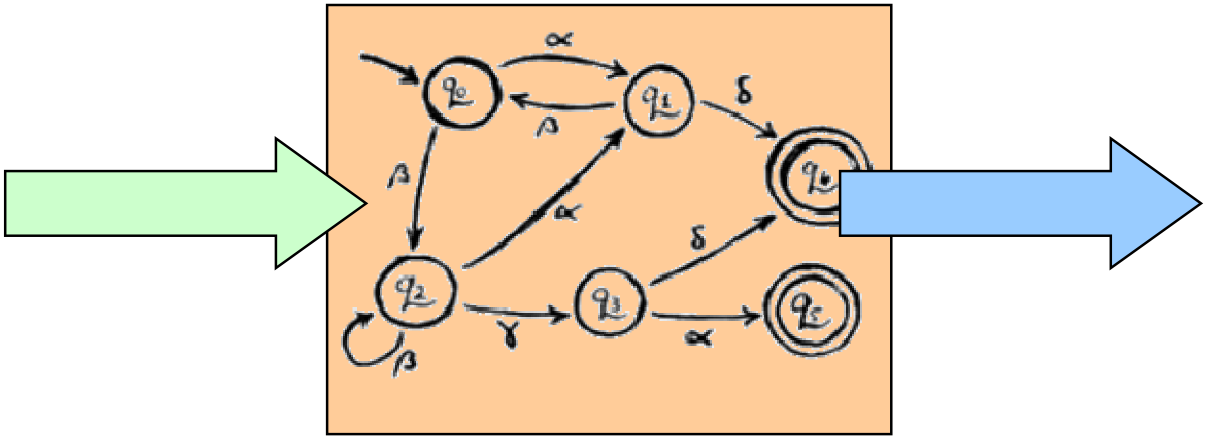
## Non-standard resources
## e.g., *precision*.



## Non-standard resources
## e.g., *precision*.



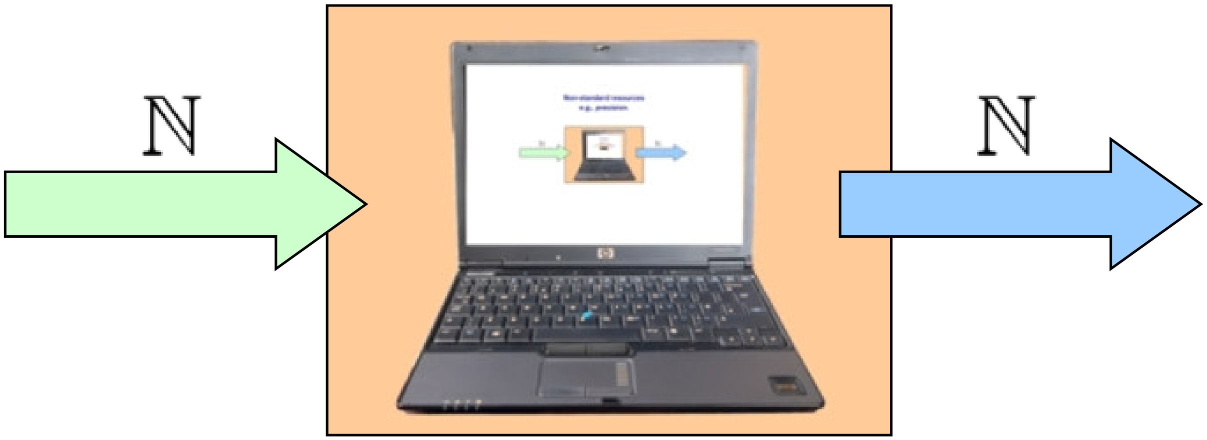
## Non-standard resources
## e.g., *precision*.



## Non-standard resources
## e.g., *precision*.

Non-standard resources
e.g., precision.

## Non-standard resources
## e.g., *precision*.



# Non-standard resources
# e.g., *precision*.

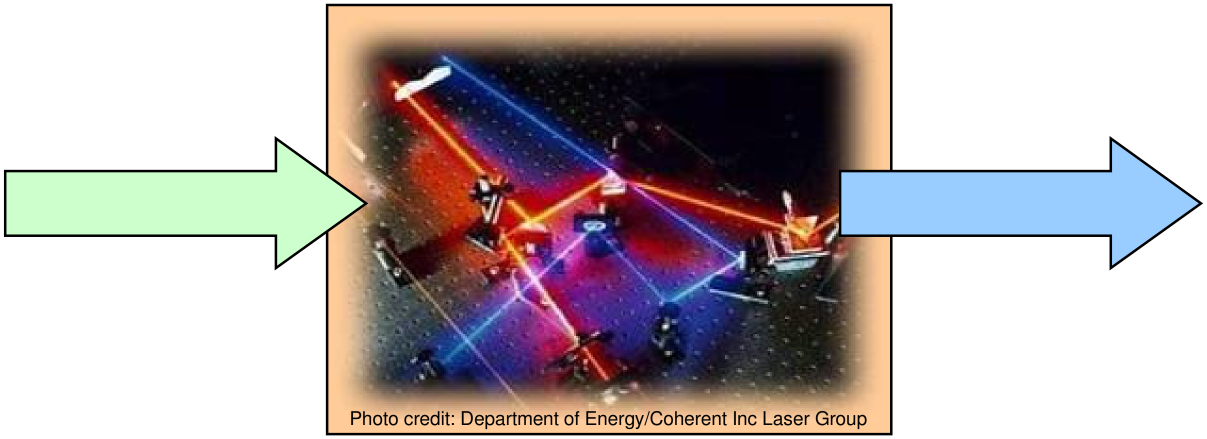
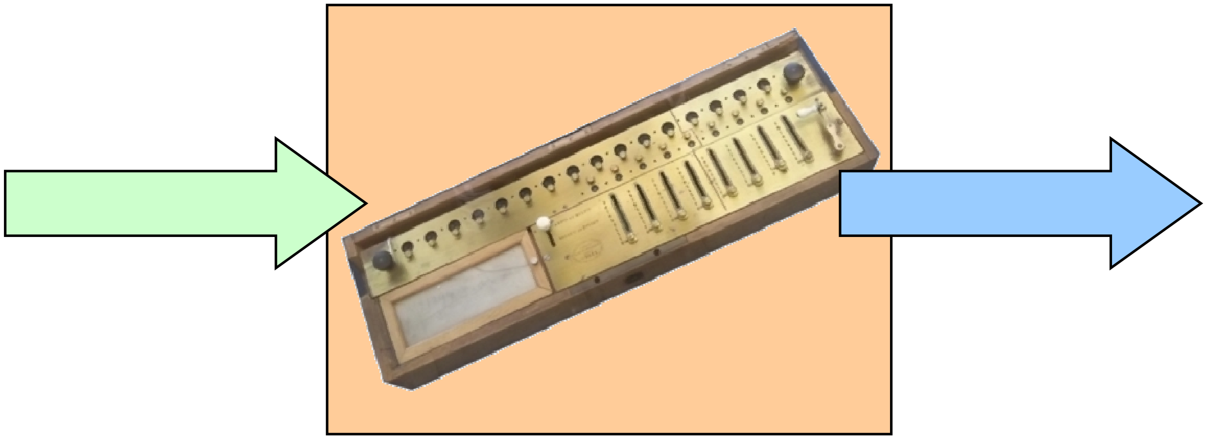


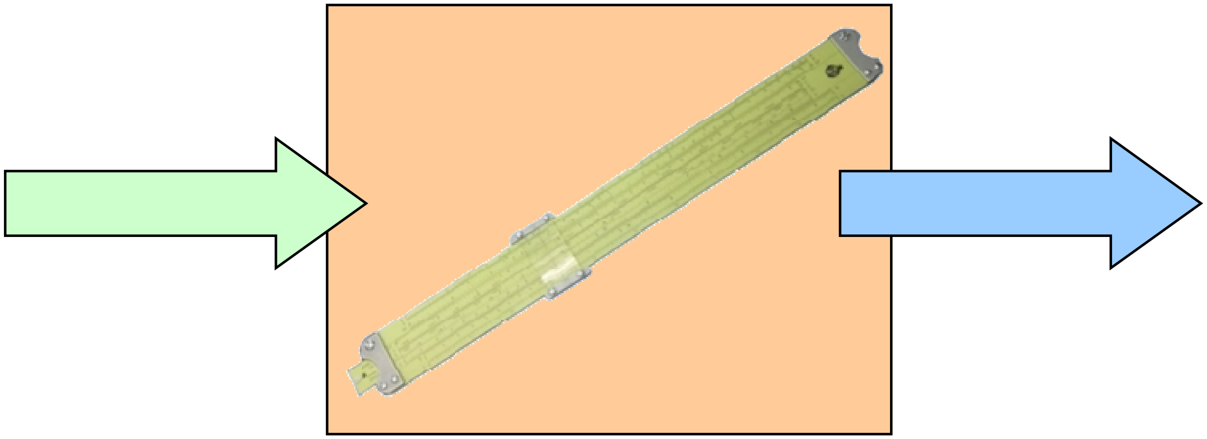
Photo credit: Department of Energy/Coherent Inc Laser Group

## Non-standard resources
## e.g., *precision*.

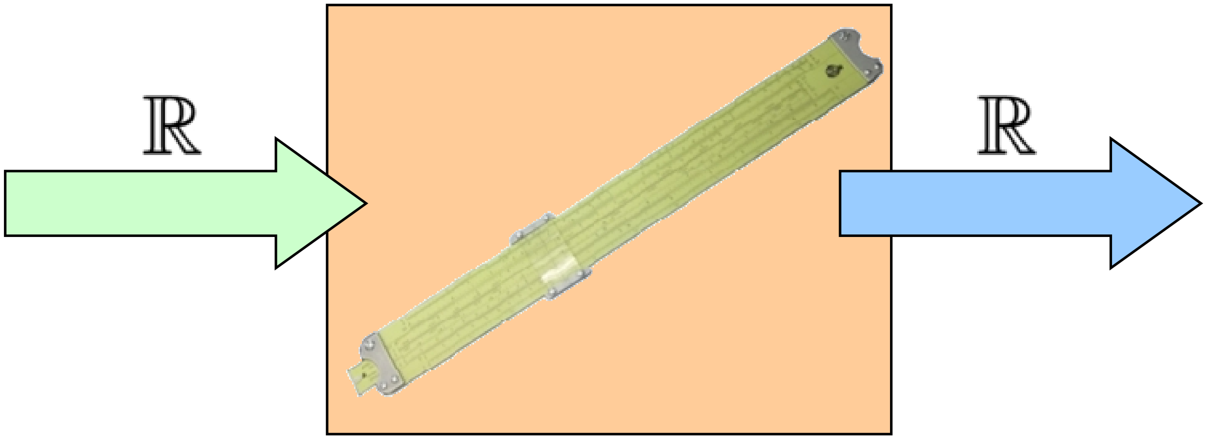


Non-standard resources

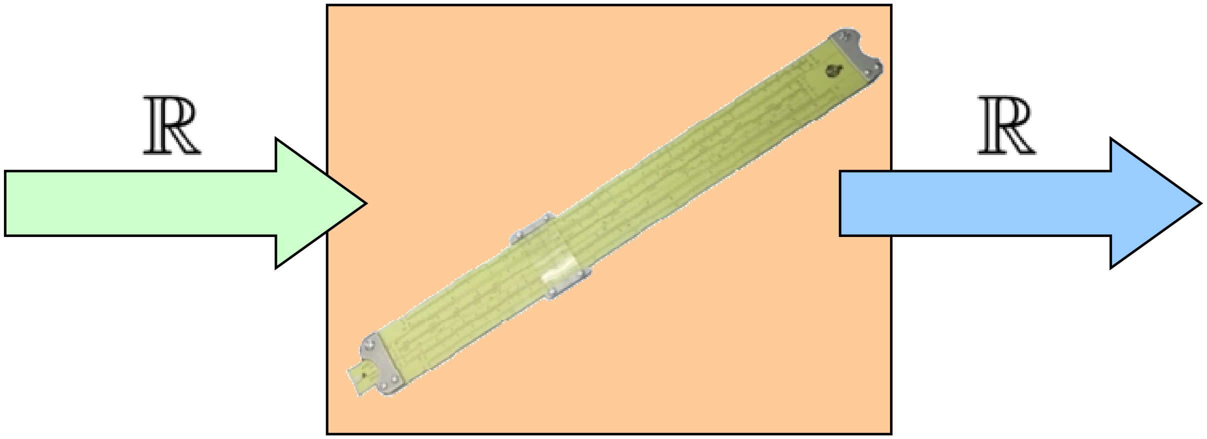
e.g., *precision*.



**Non-standard resources**
**e.g., *precision*.**

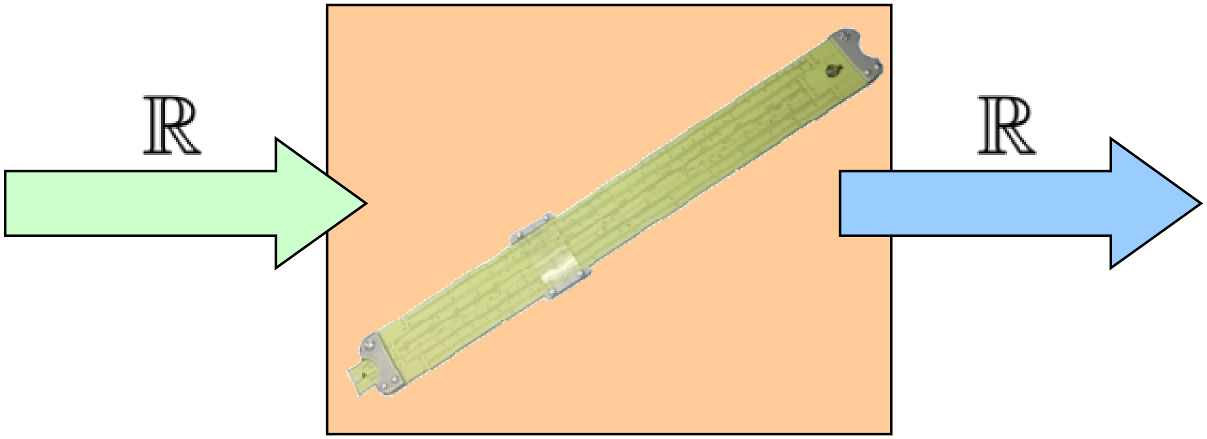


## Non-standard resources
## e.g., *precision*.



***Precision complexity***.

## Non-standard resources
## e.g., *precision*.



***Precision complexity***.

Detail deferred: ***A Model-Independent Theory of Computational Complexity***
`http://users.ox.ac.uk/~quee1871/thesis.pdf`

# Resources…

time

space

# Resources…

time

space

precision

## Resources…

time

space

precision

energy

material cost

thermodynamic cost

mass

## Resources…

time

space

precision

energy

material cost

thermodynamic cost

mass

etc.

# Resources…
# …for *computation*.

time

space

precision

energy

material cost

thermodynamic cost

mass

etc.

## Resources…
## …for *computation*.

time

space

precision

energy

material cost

thermodynamic cost

mass

etc.

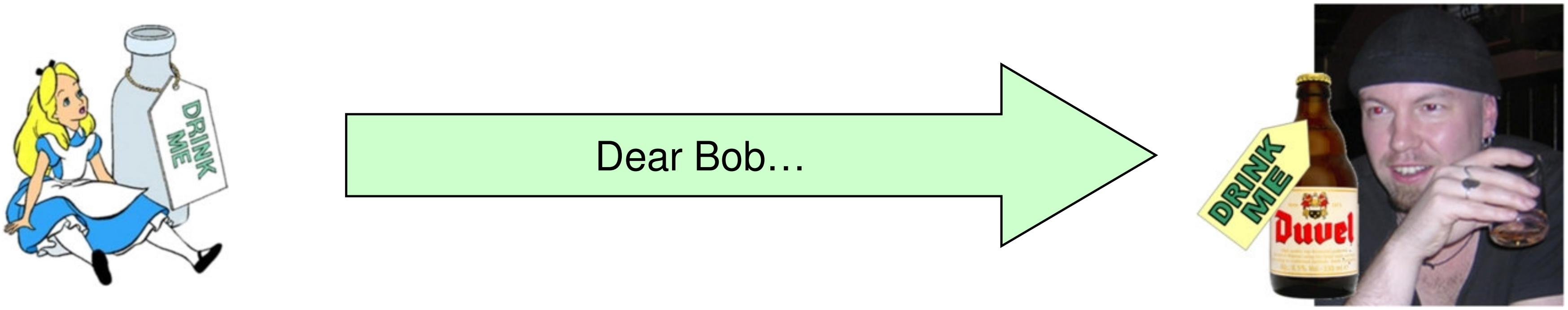
## …for *cryptographic protocols*.

# Communication.

# Communication.

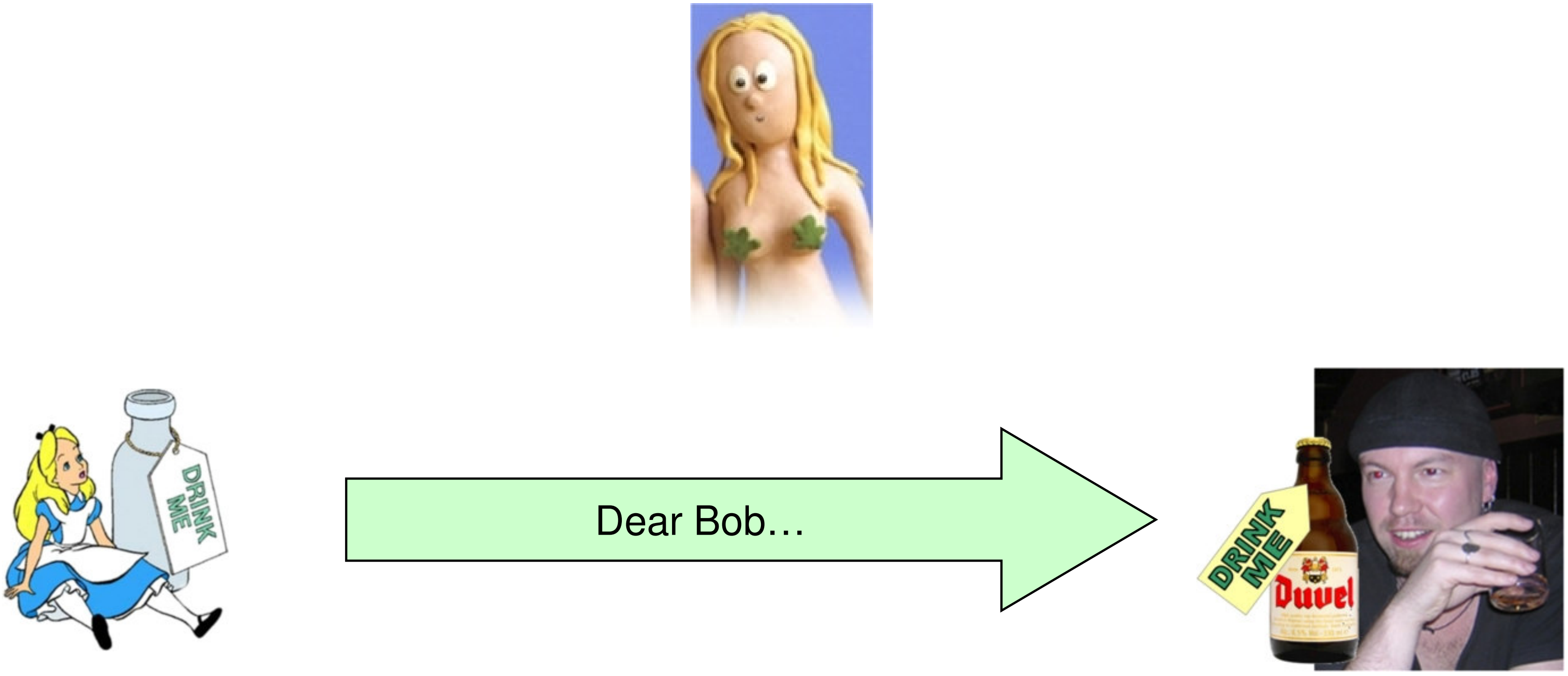
# Communication.

# Communication.



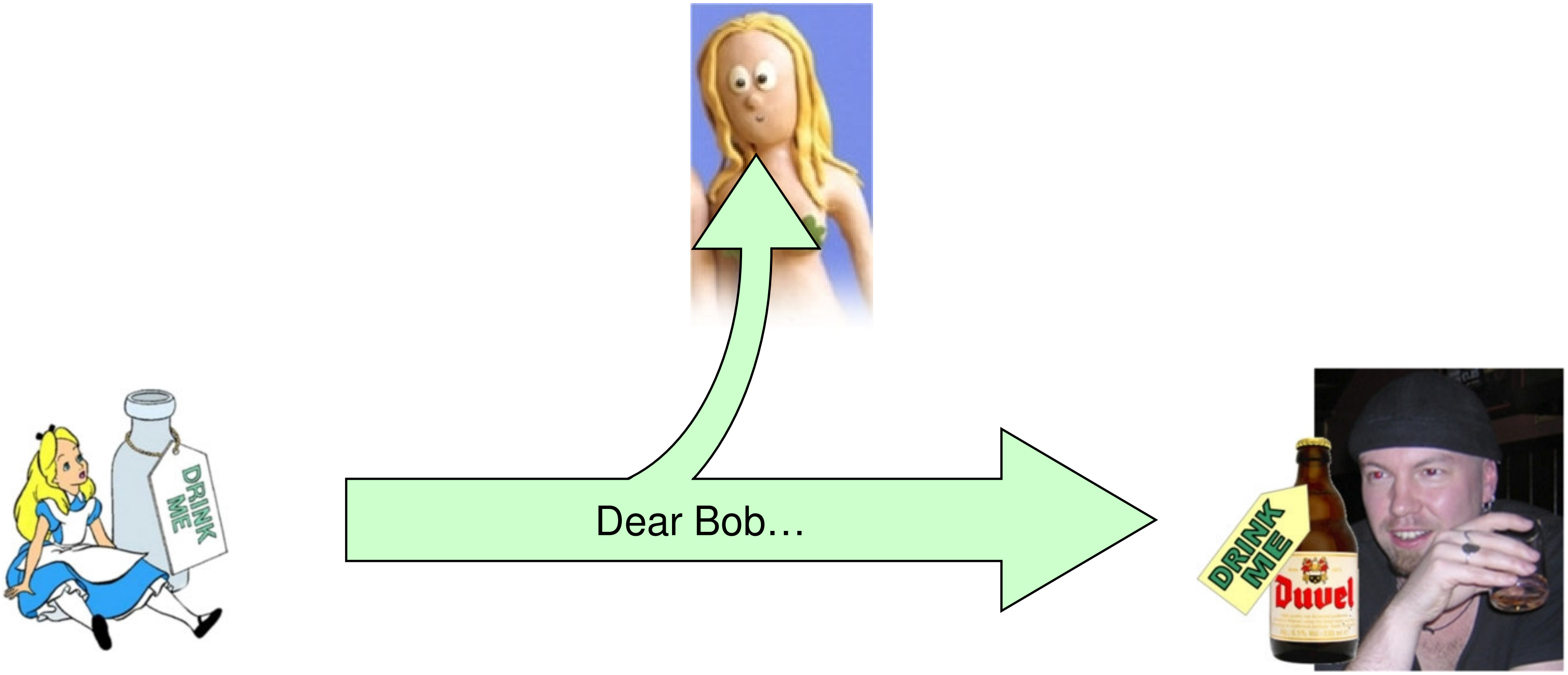
Dear Bob…

# Communication.



Dear Bob…

# Communication.



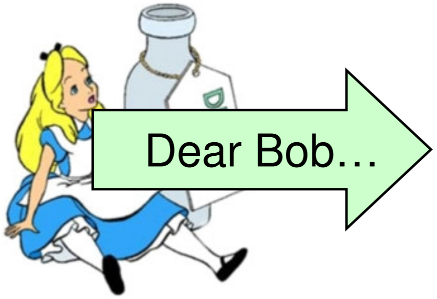
Dear Bob…

~~Communication.~~

Symmetric-key cryptography.

~~Communication.~~

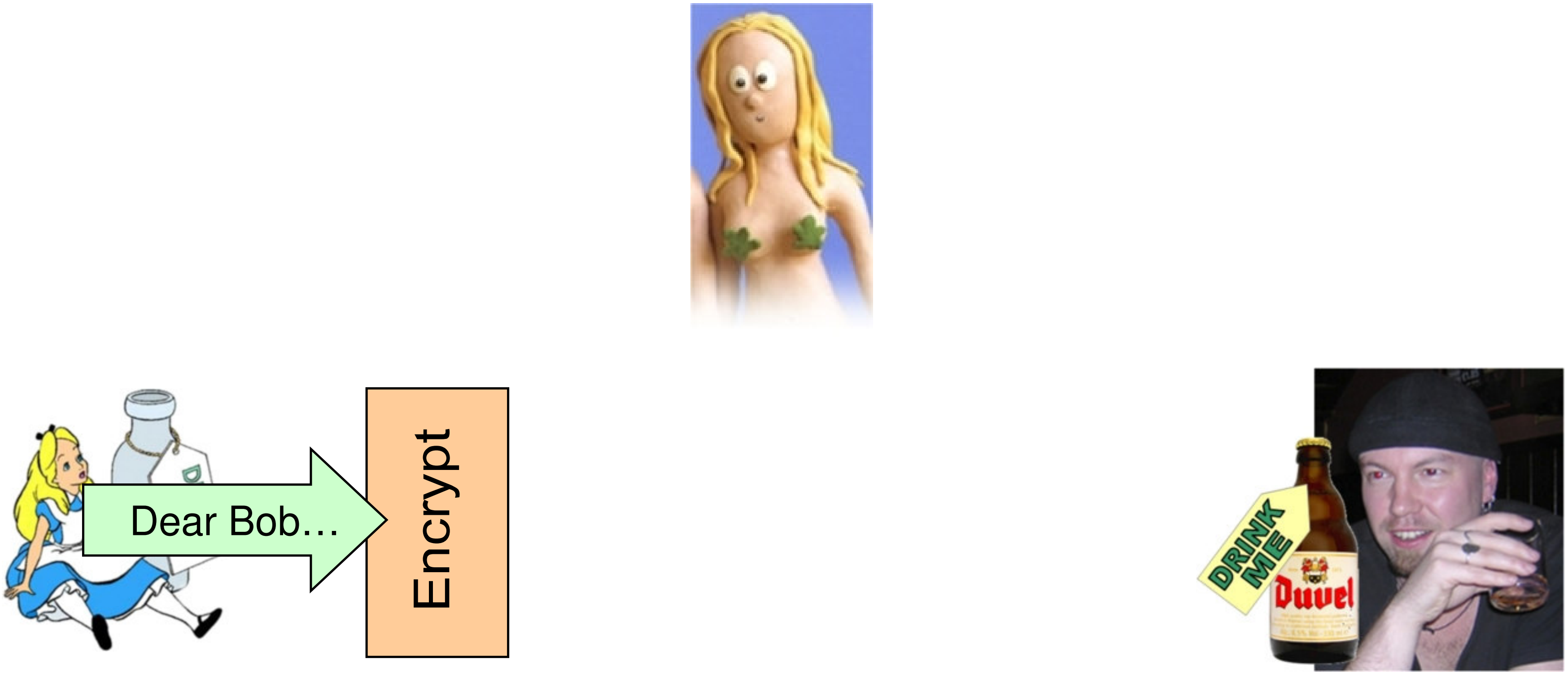
## Symmetric-key cryptography.



Dear Bob…

~~Communication.~~

## Symmetric-key cryptography.

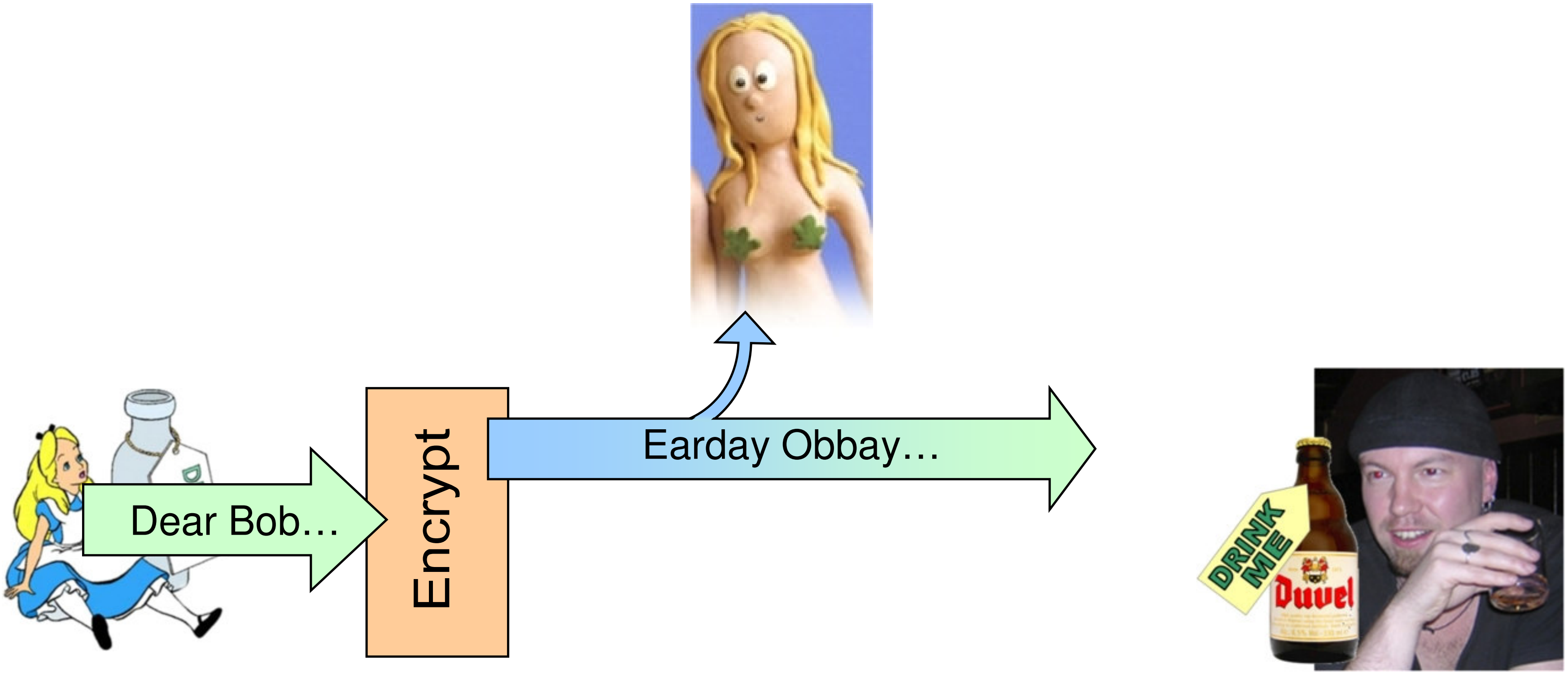


Dear Bob…

Encrypt

~~Communication.~~

# Symmetric-key cryptography.



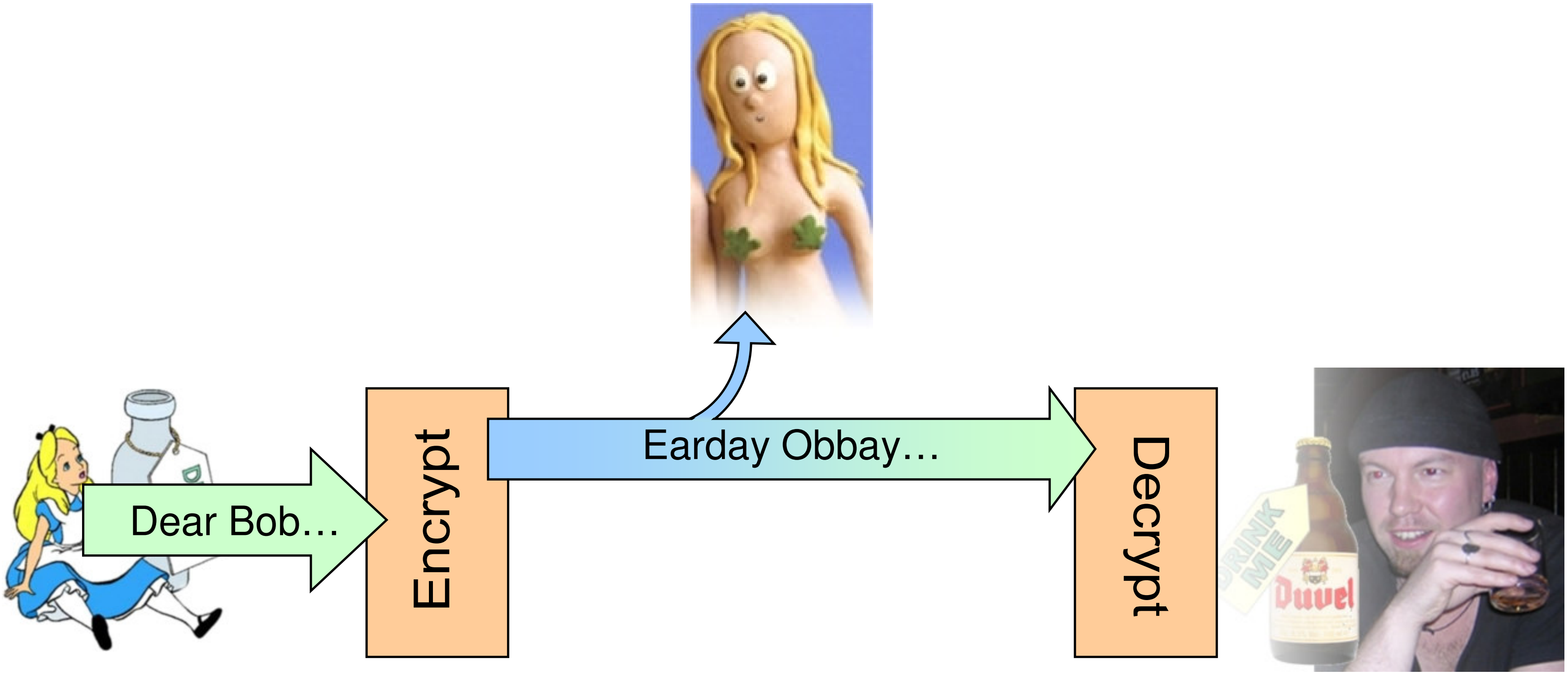
Dear Bob…

Encrypt

Earday Obbay…

## ~~Communication.~~

## Symmetric-key cryptography.



Dear Bob…

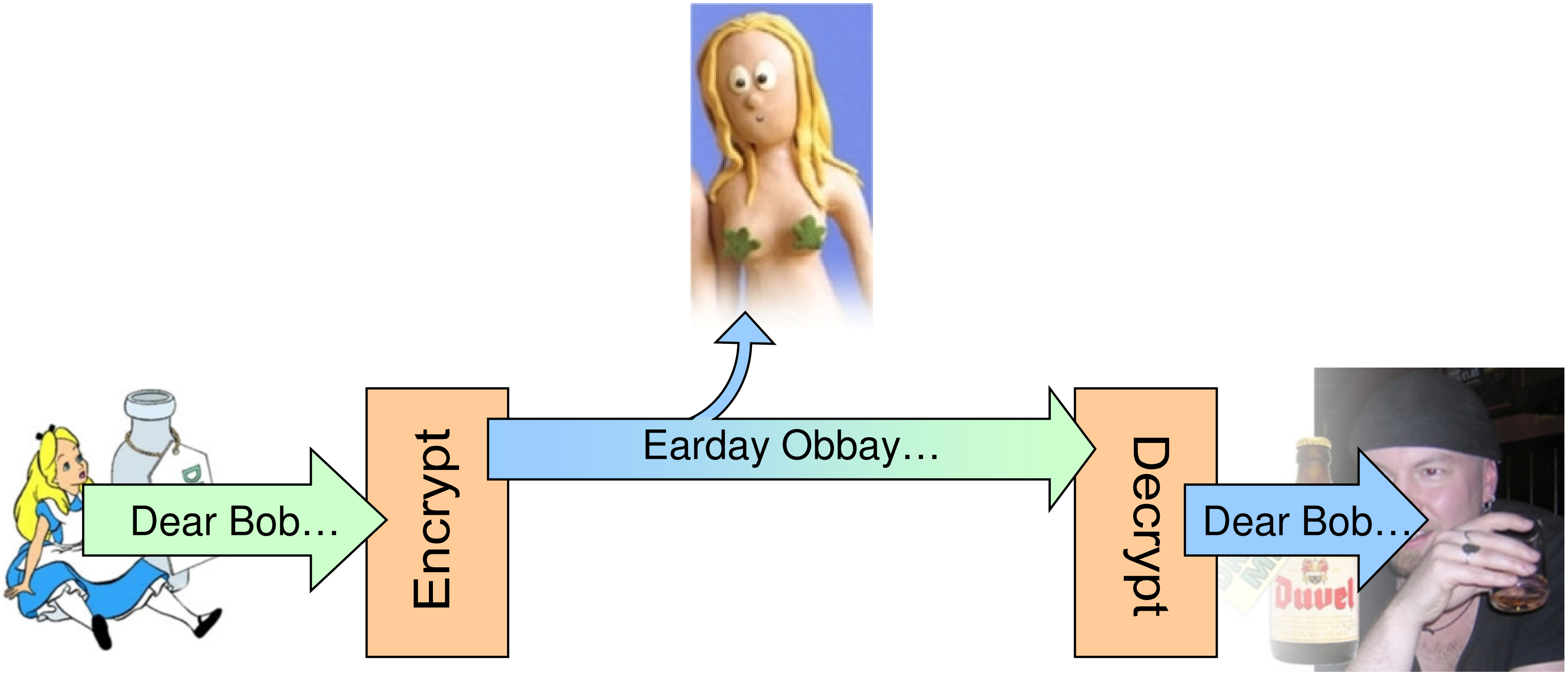
Encrypt

Earday Obbay…

Decrypt

~~Communication.~~

## Symmetric-key cryptography.



Dear Bob…

Encrypt

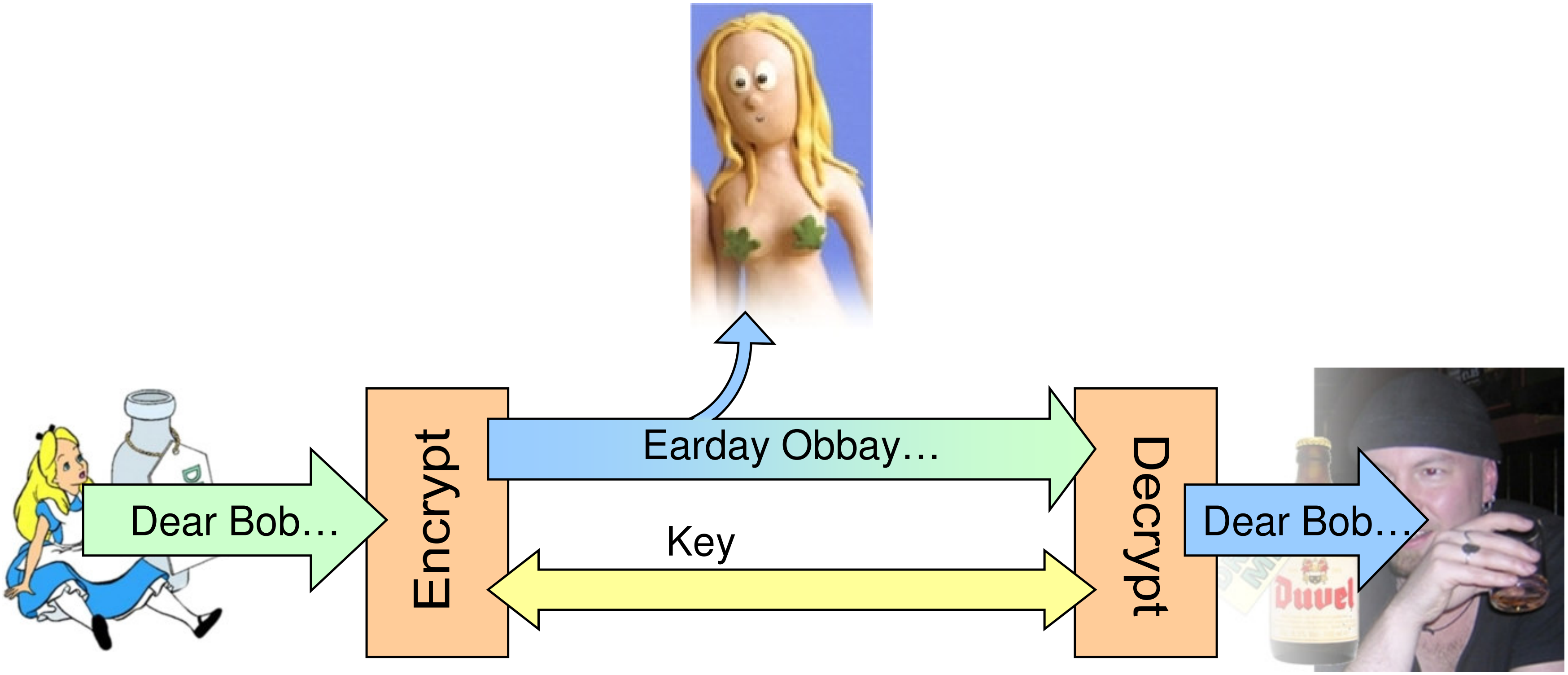
Earday Obbay…

Decrypt

Dear Bob…

~~Communication.~~

## Symmetric-key cryptography.



Dear Bob…

Encrypt

Earday Obbay…

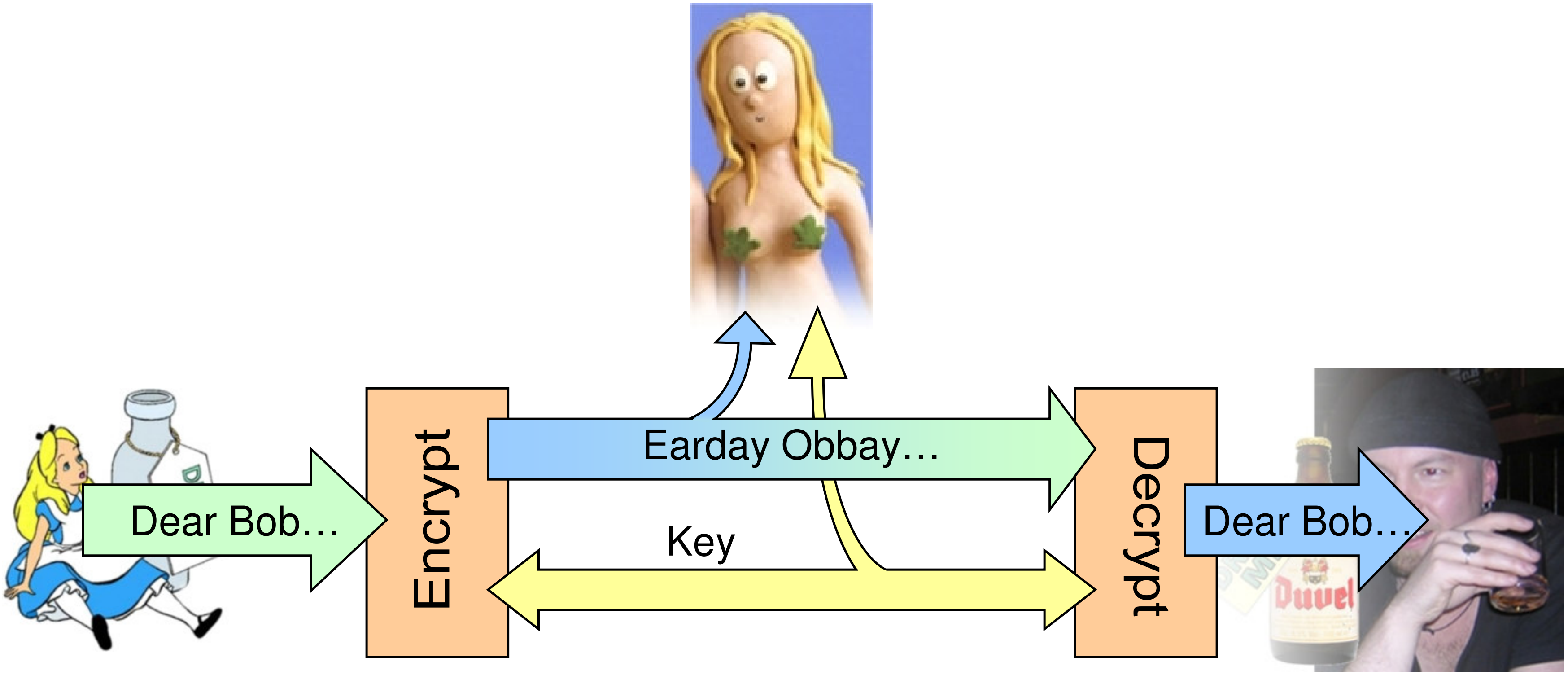
Key

Decrypt

Dear Bob…

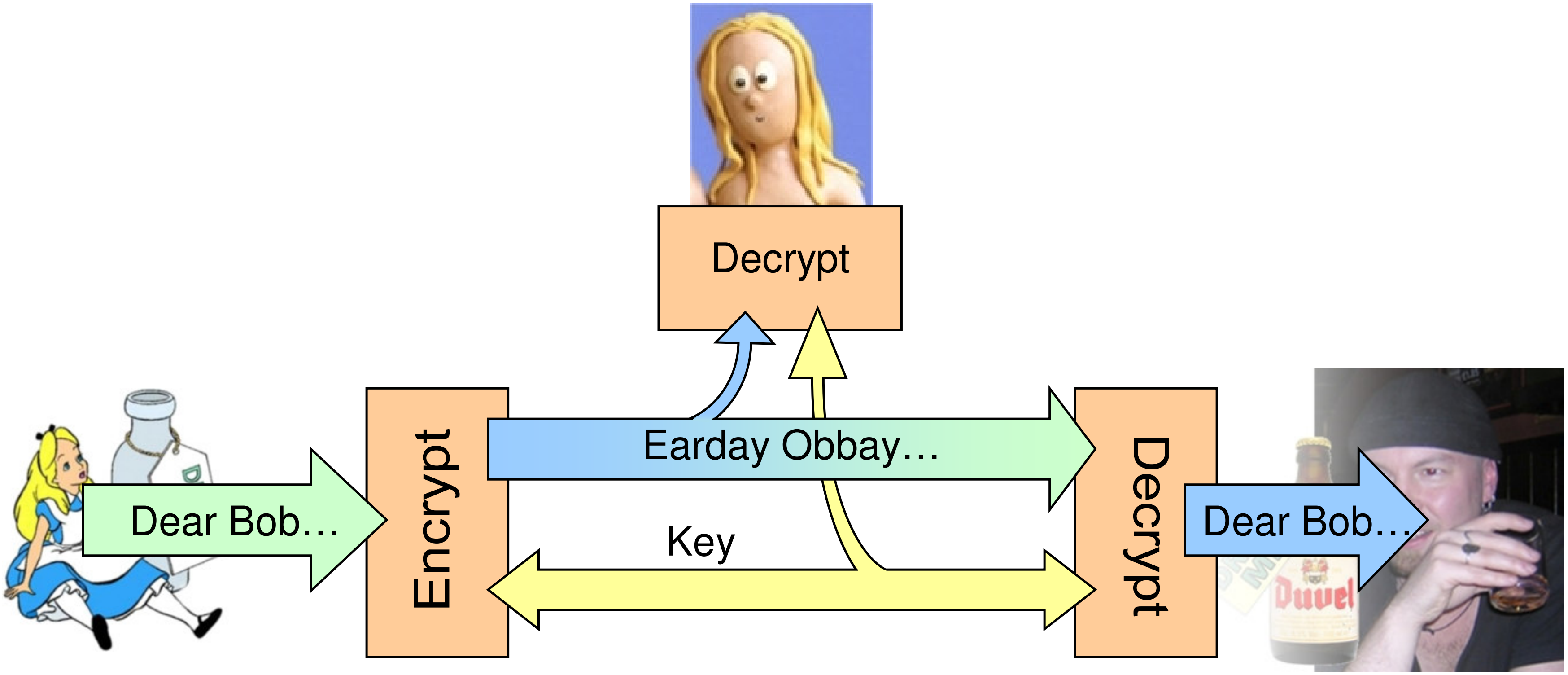
~~Communication.~~

## Symmetric-key cryptography.



## ~~Communication.~~

## Symmetric-key cryptography.



Decrypt

Dear Bob…

Encrypt

Earday Obbay…

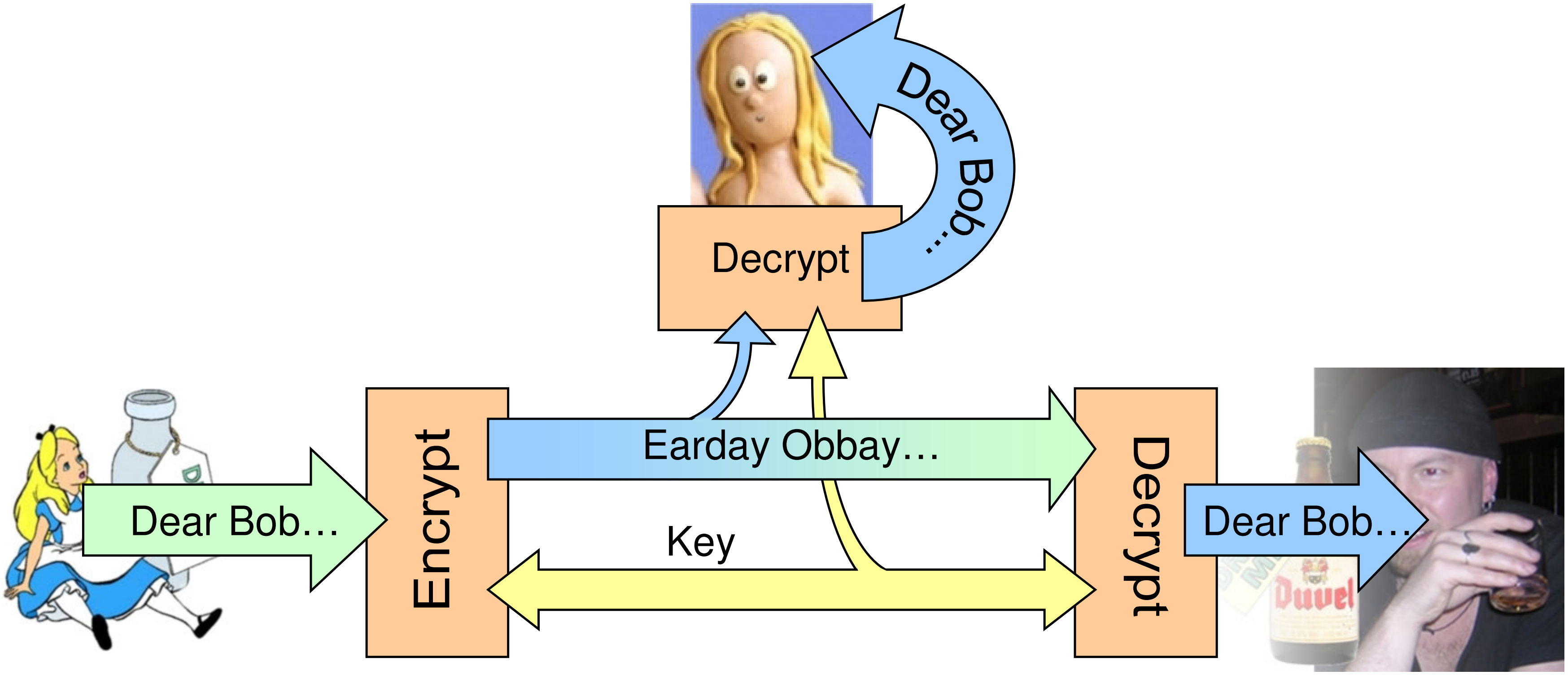
Key

Decrypt

Dear Bob…

~~Communication.~~

## Symmetric-key cryptography.



~~Communication.~~

~~Symmetric-key cryptography.~~

Public-key cryptography.

~~Communication.~~

~~Symmetric-key cryptography.~~

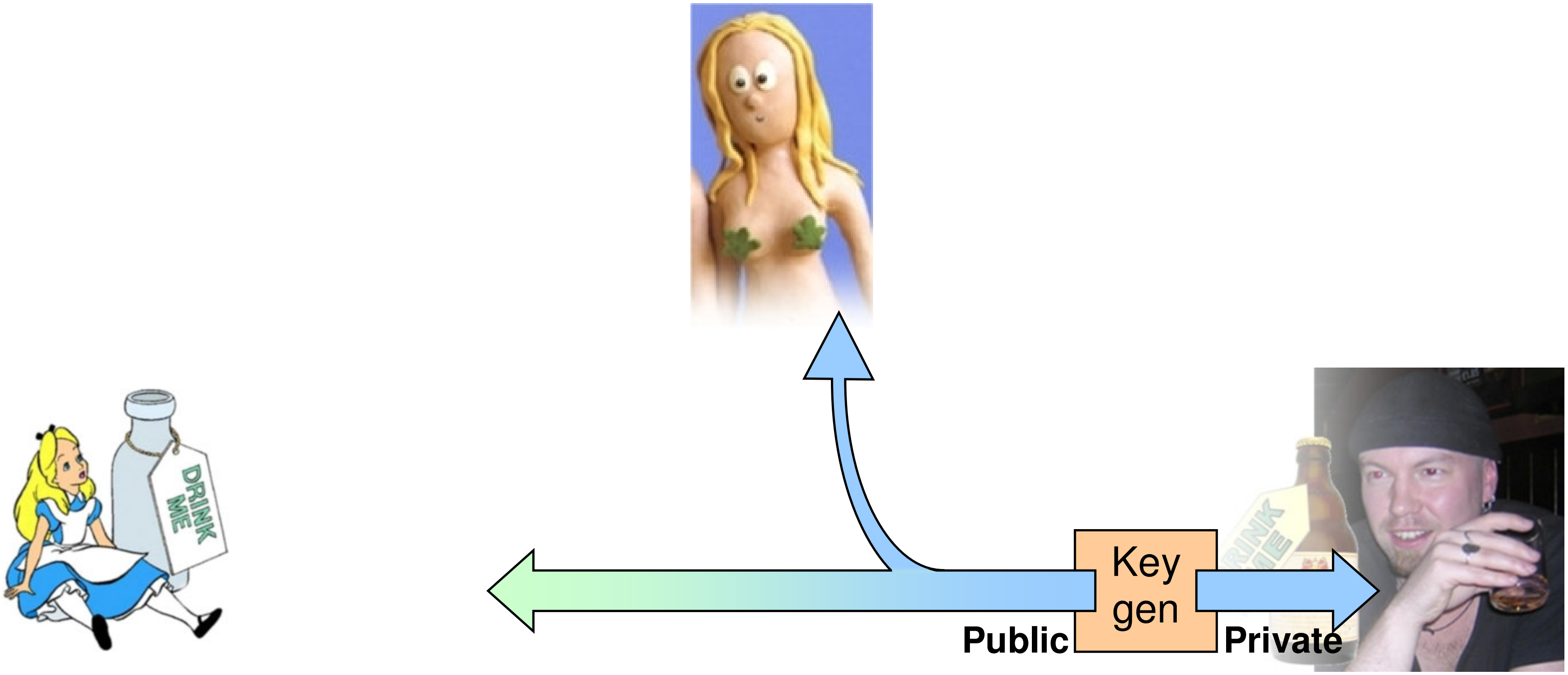
# Public-key cryptography.

Key
gen

~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.



Key gen

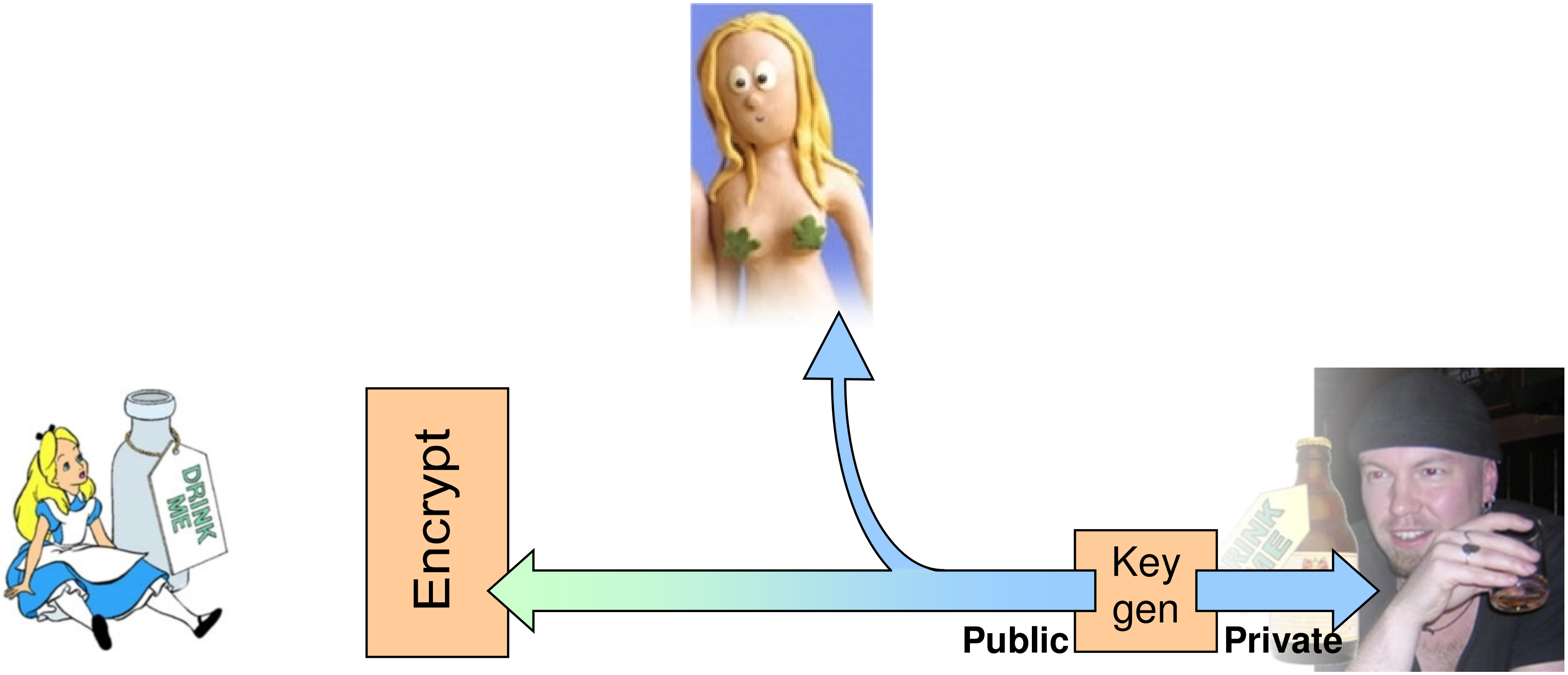
Public

Private

~~Communication.~~

~~Symmetric-key cryptography.~~

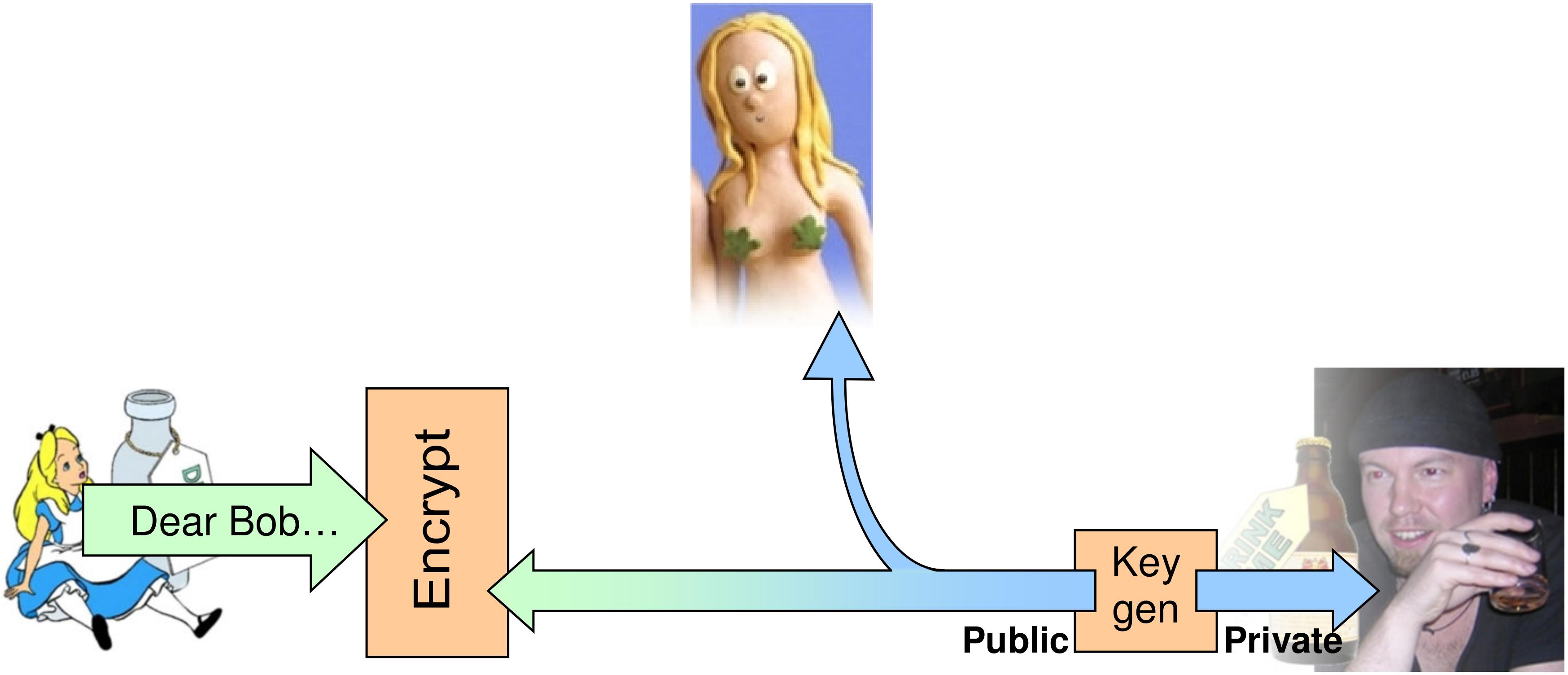
# Public-key cryptography.



~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.



Dear Bob…

Encrypt

Key gen

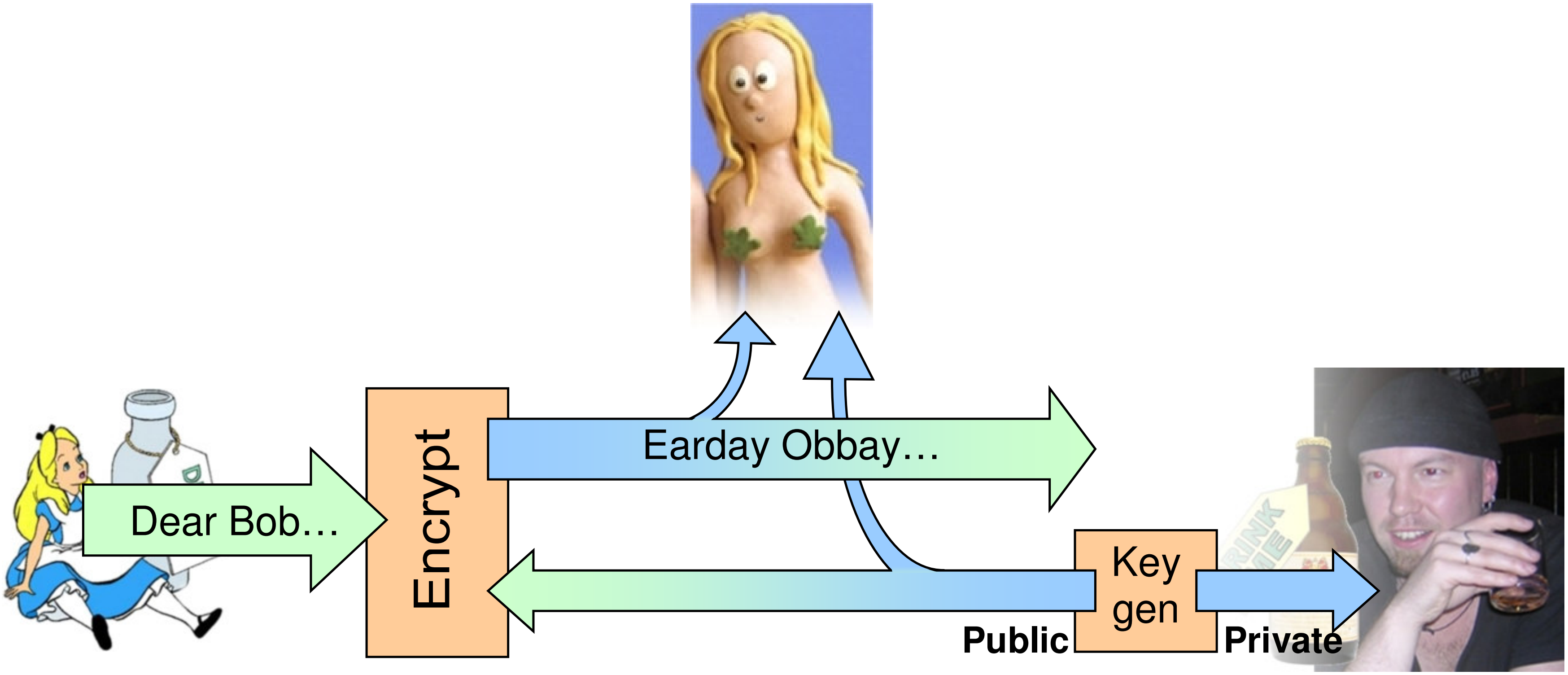
Public

Private

~~Communication.~~

~~Symmetric-key cryptography.~~

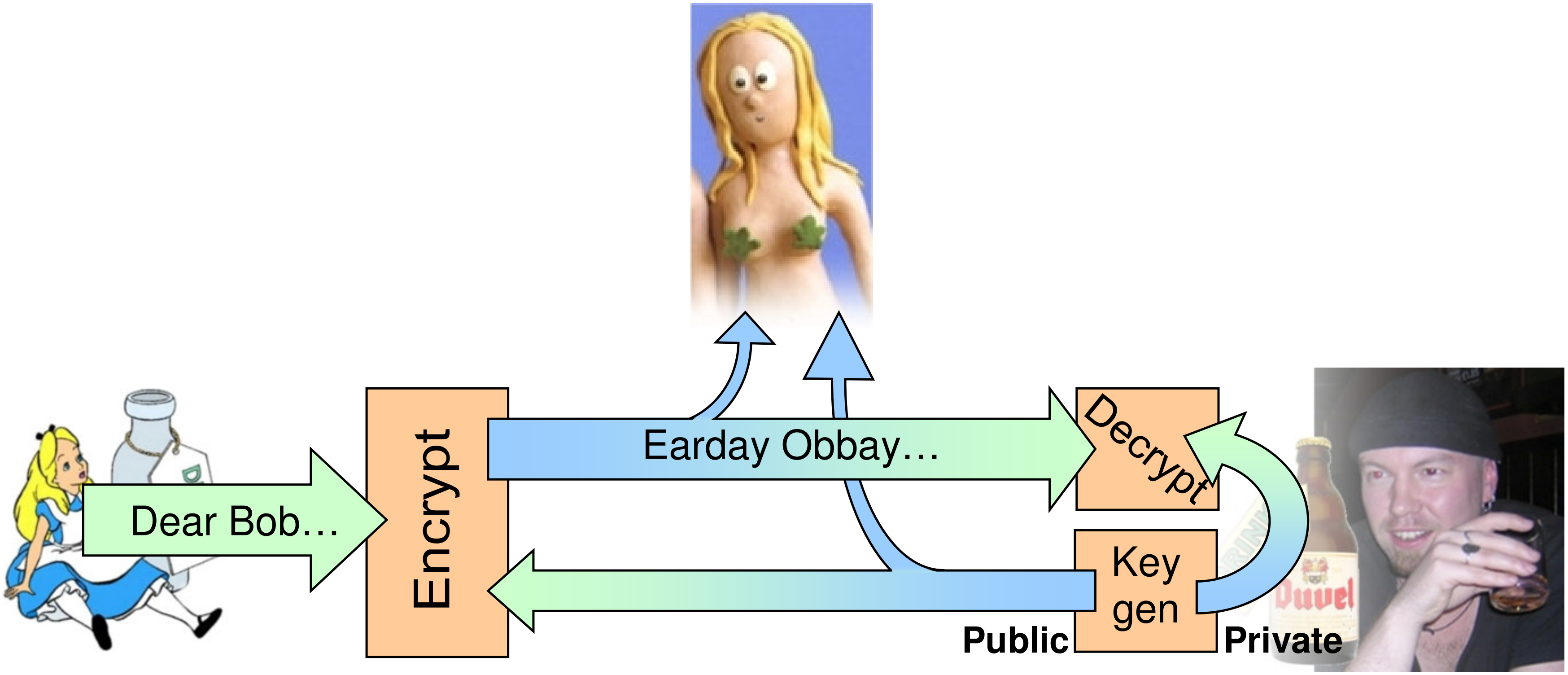
# Public-key cryptography.



~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.



Dear Bob…

Encrypt

Earday Obbay…

Decrypt

Key gen

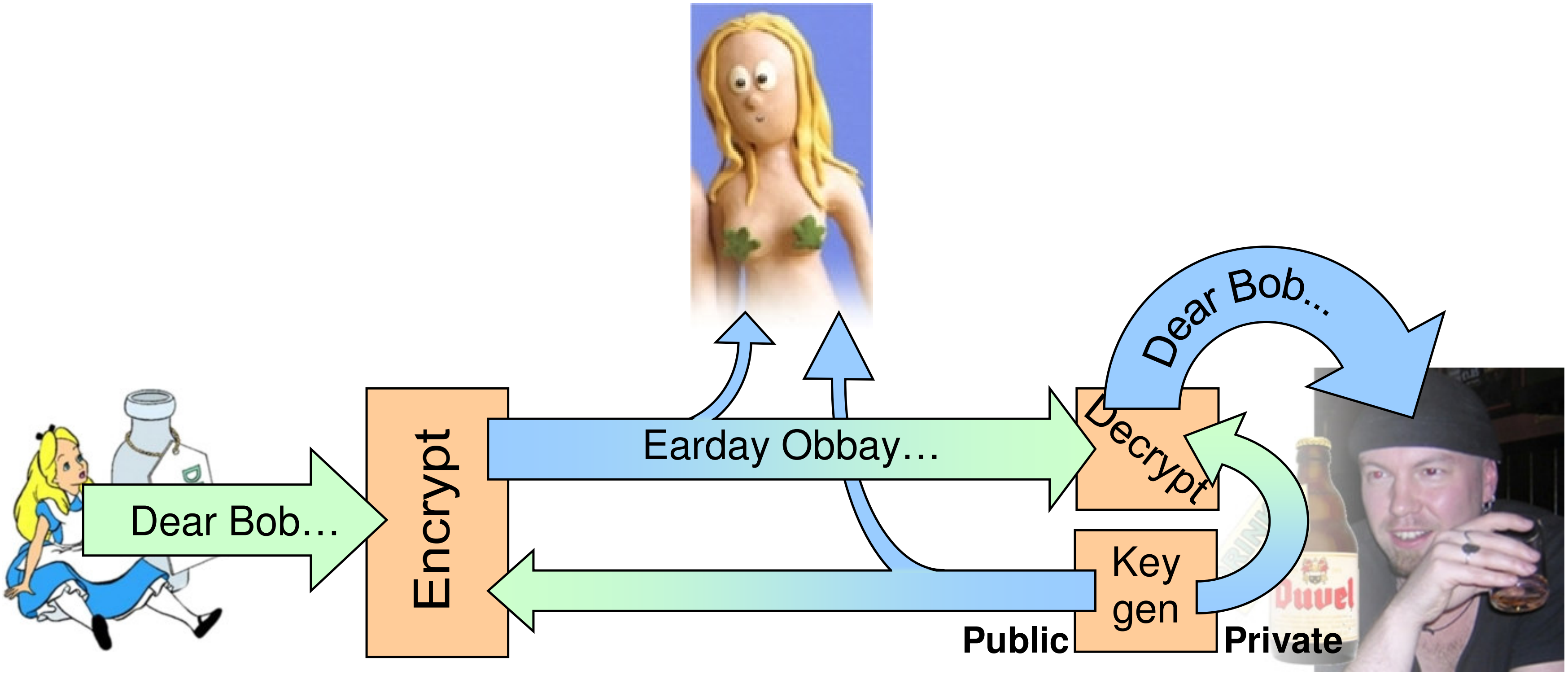
Public

Private

~~Communication.~~

~~Symmetric-key cryptography.~~

# Public-key cryptography.



Dear Bob…

Encrypt

Earday Obbay…

Decrypt

Dear Bob...

Key gen

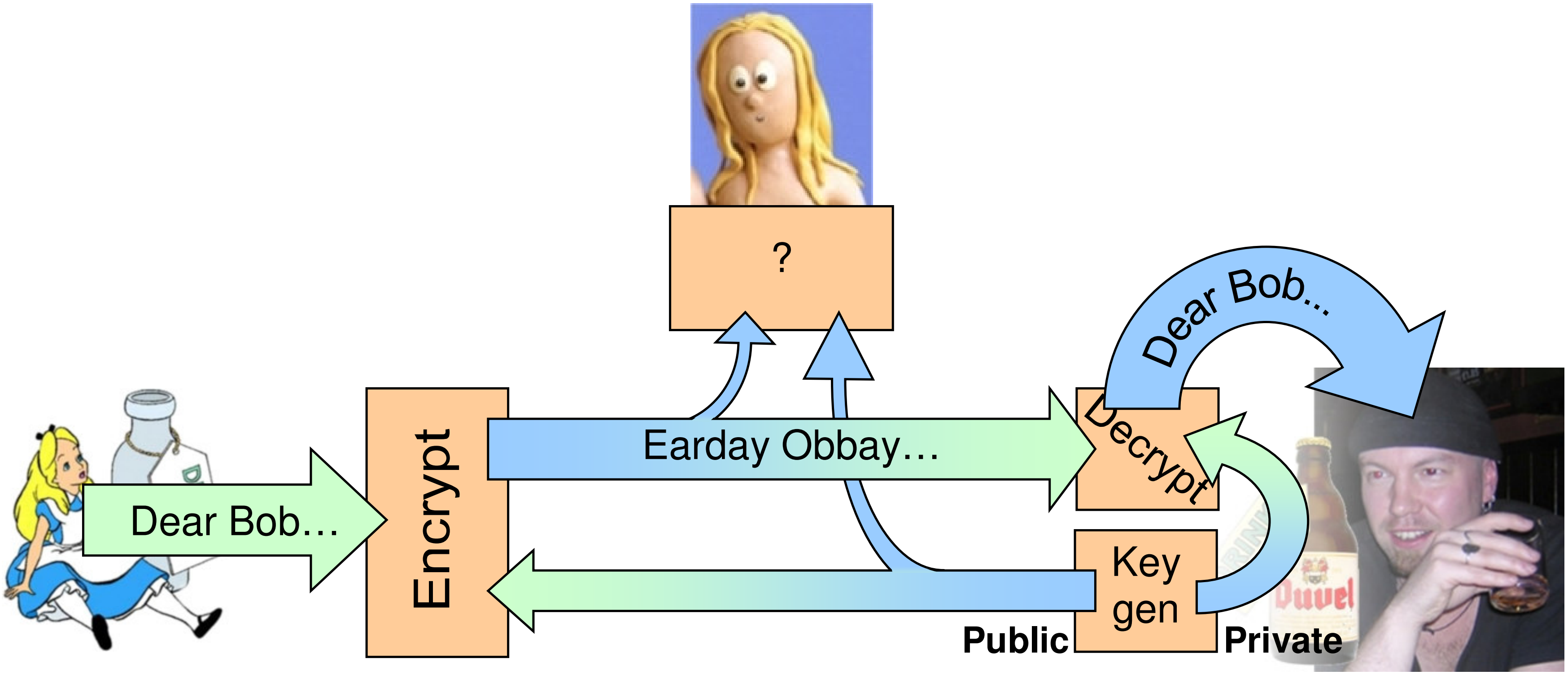
Public

Private

~~Communication.~~

~~Symmetric-key cryptography.~~

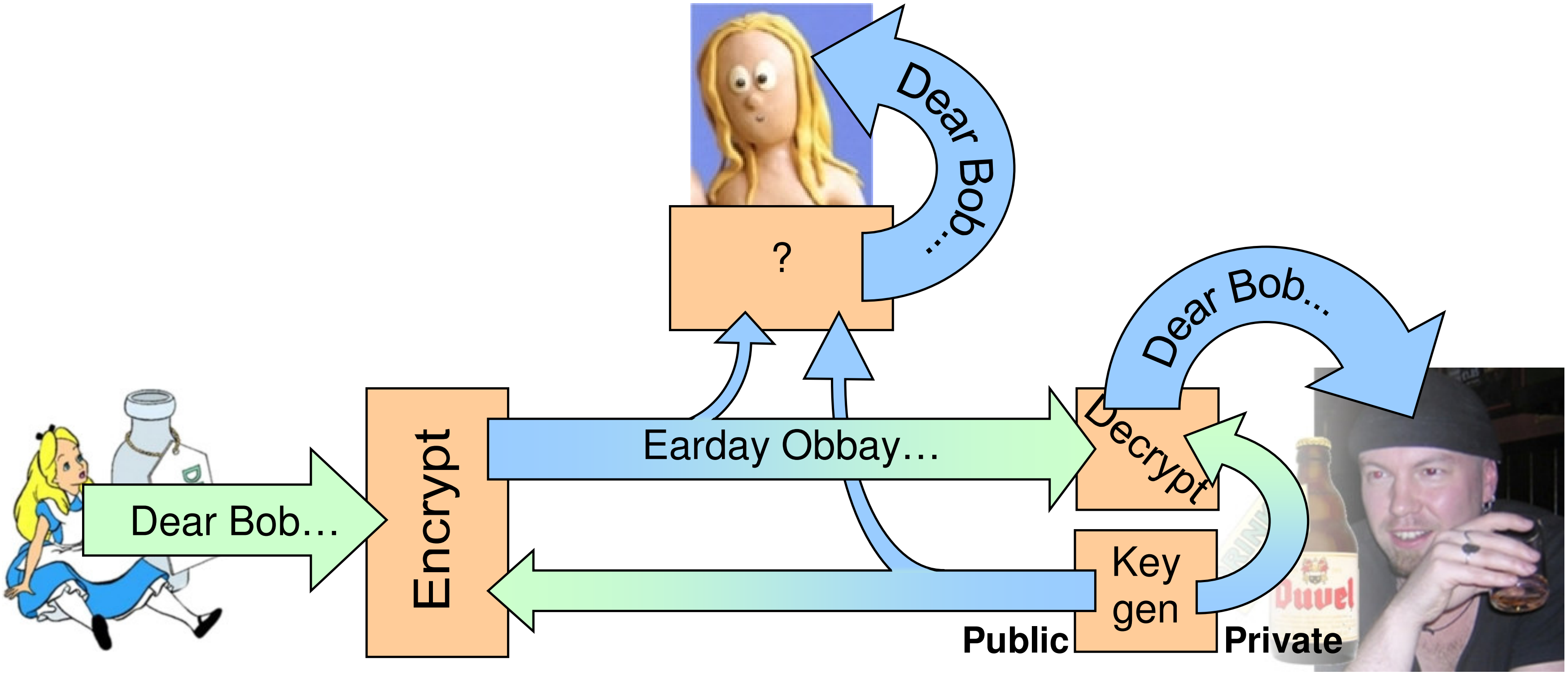
## Public-key cryptography.



~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.

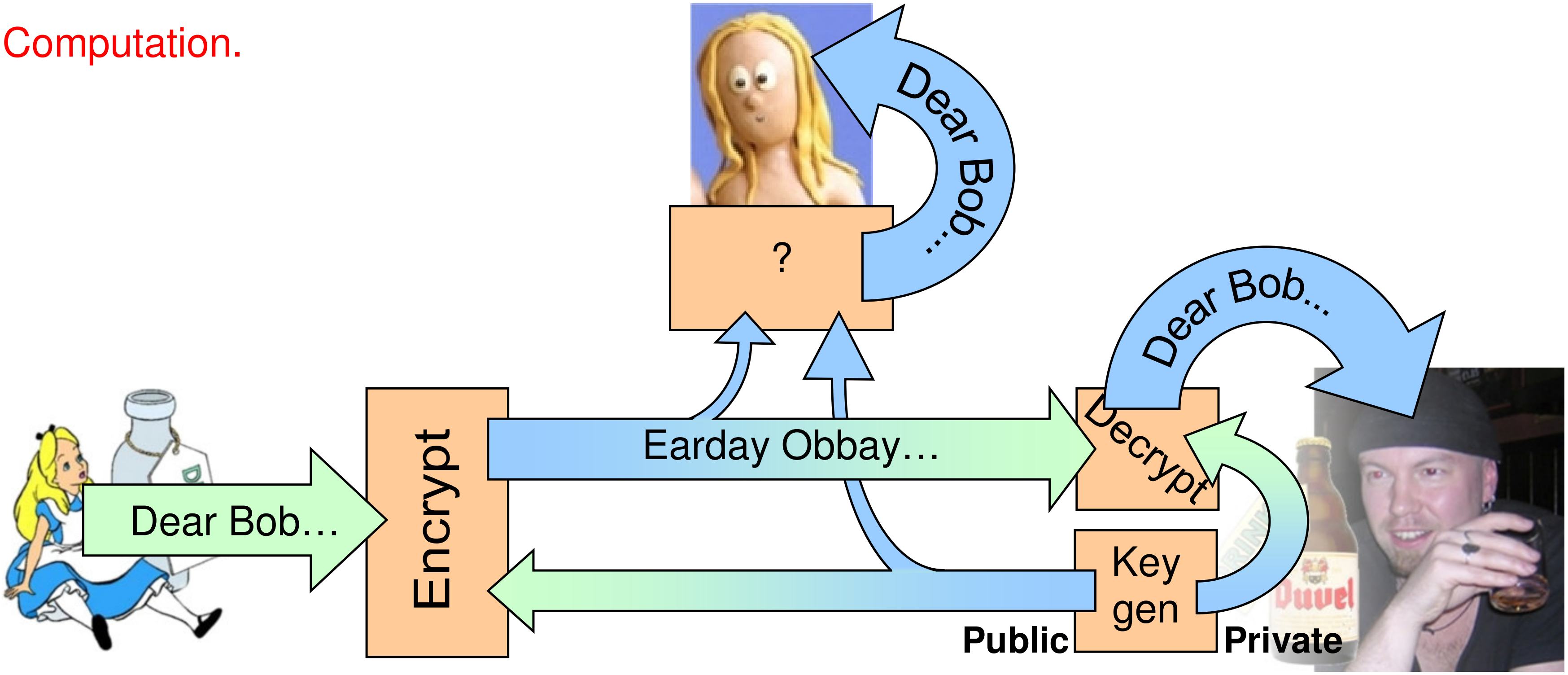


~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.

Computation.

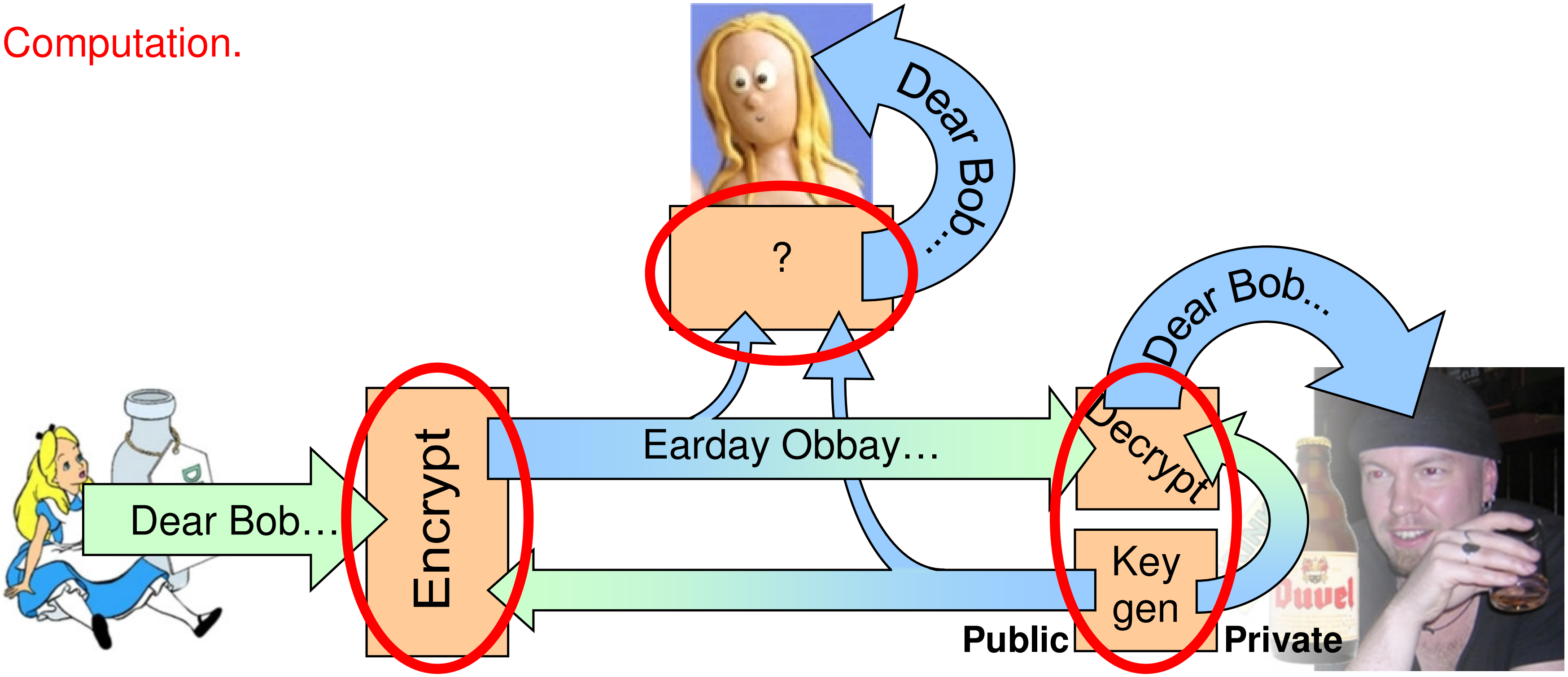


~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.

Computation.



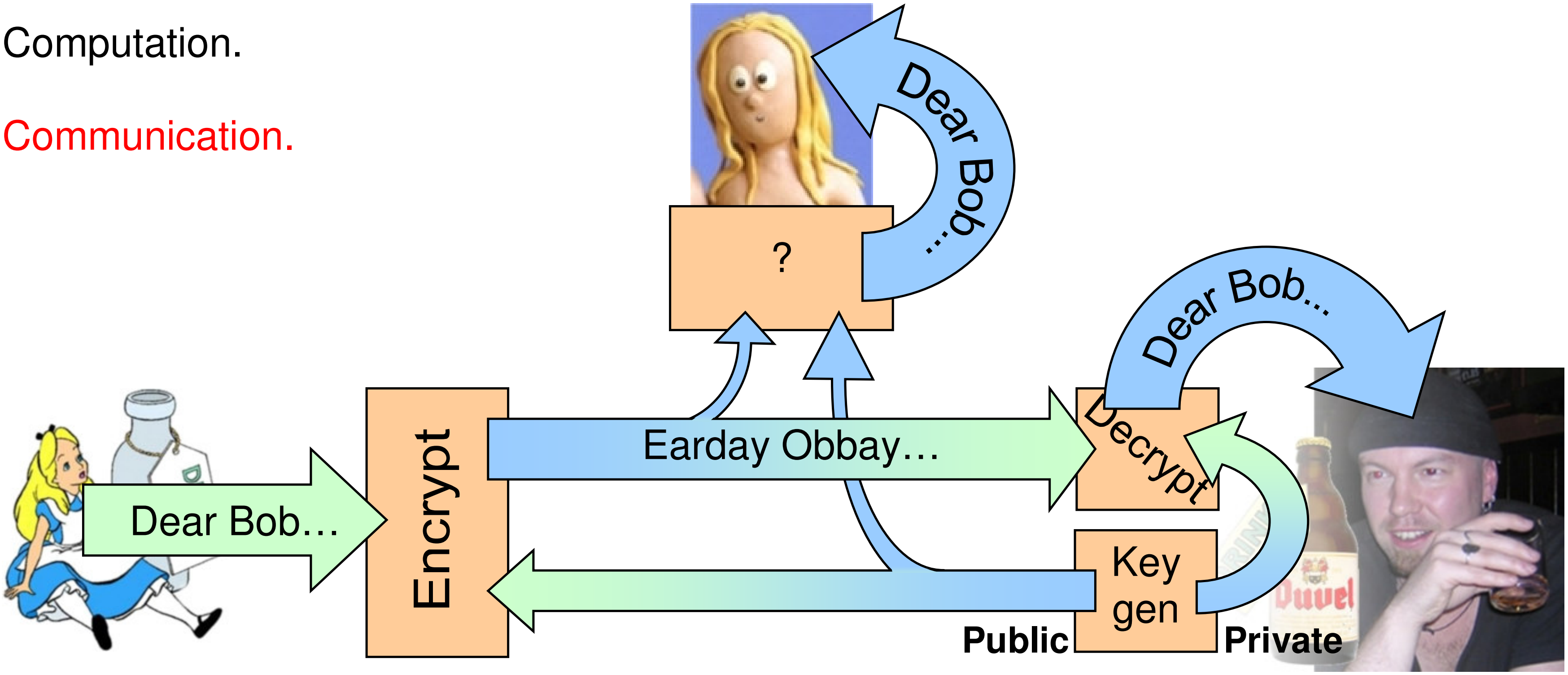
~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.

Computation.

Communication.



?

Dear Bob…

Dear Bob…

Dear Bob…

Encrypt

Earday Obbay…

Decrypt

Key gen

Public

Private

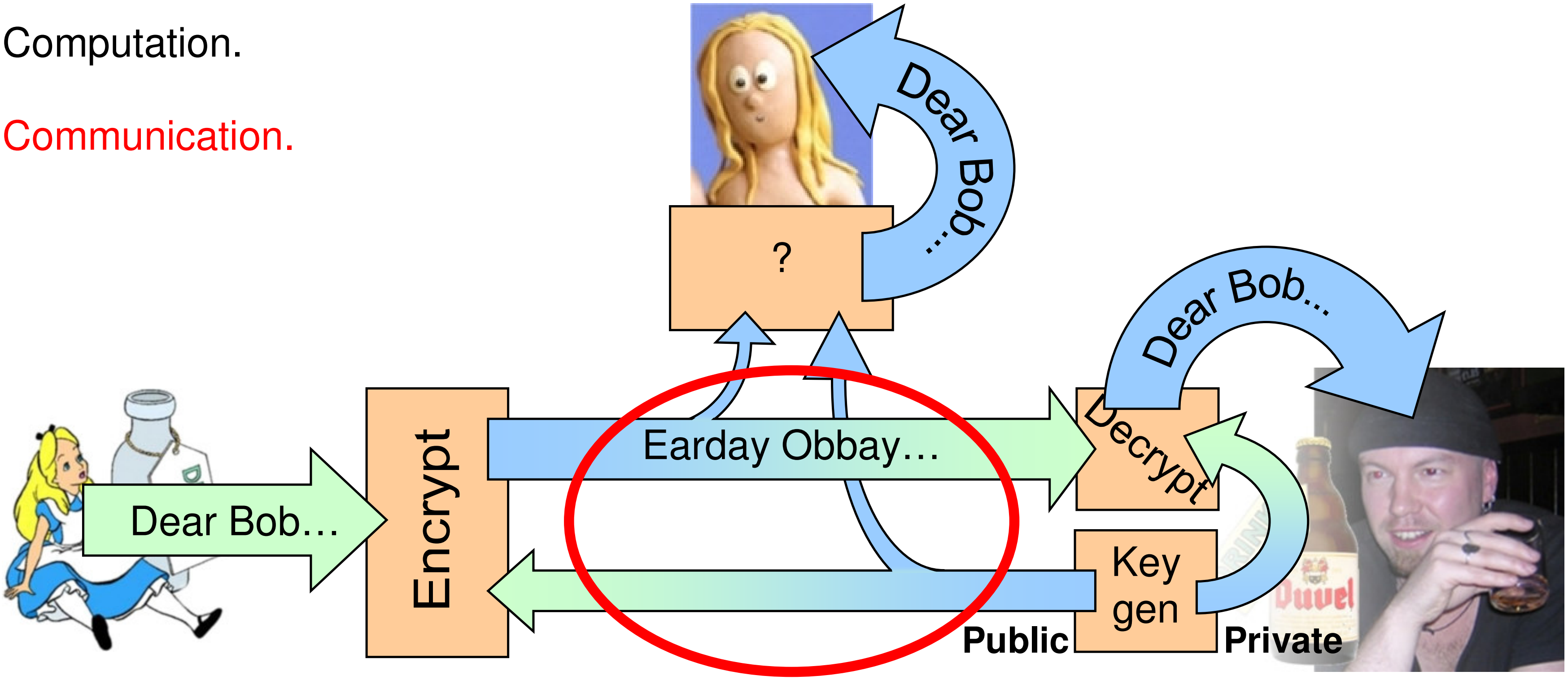
~~Communication.~~

~~Symmetric-key cryptography.~~

## Public-key cryptography.

Computation.

Communication.



~~Communication.~~

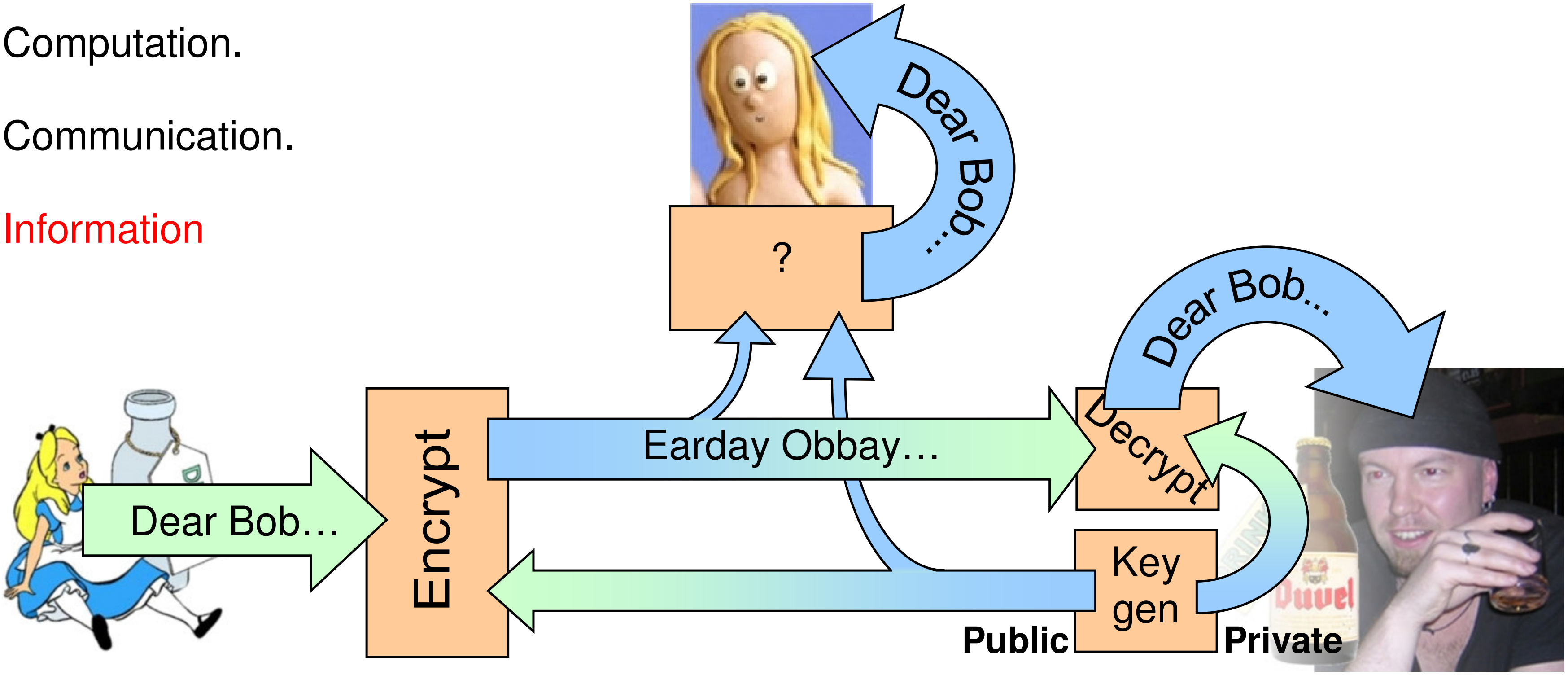
~~Symmetric-key cryptography.~~

# Public-key cryptography.

Computation.

Communication.

Information



~~Communication.~~

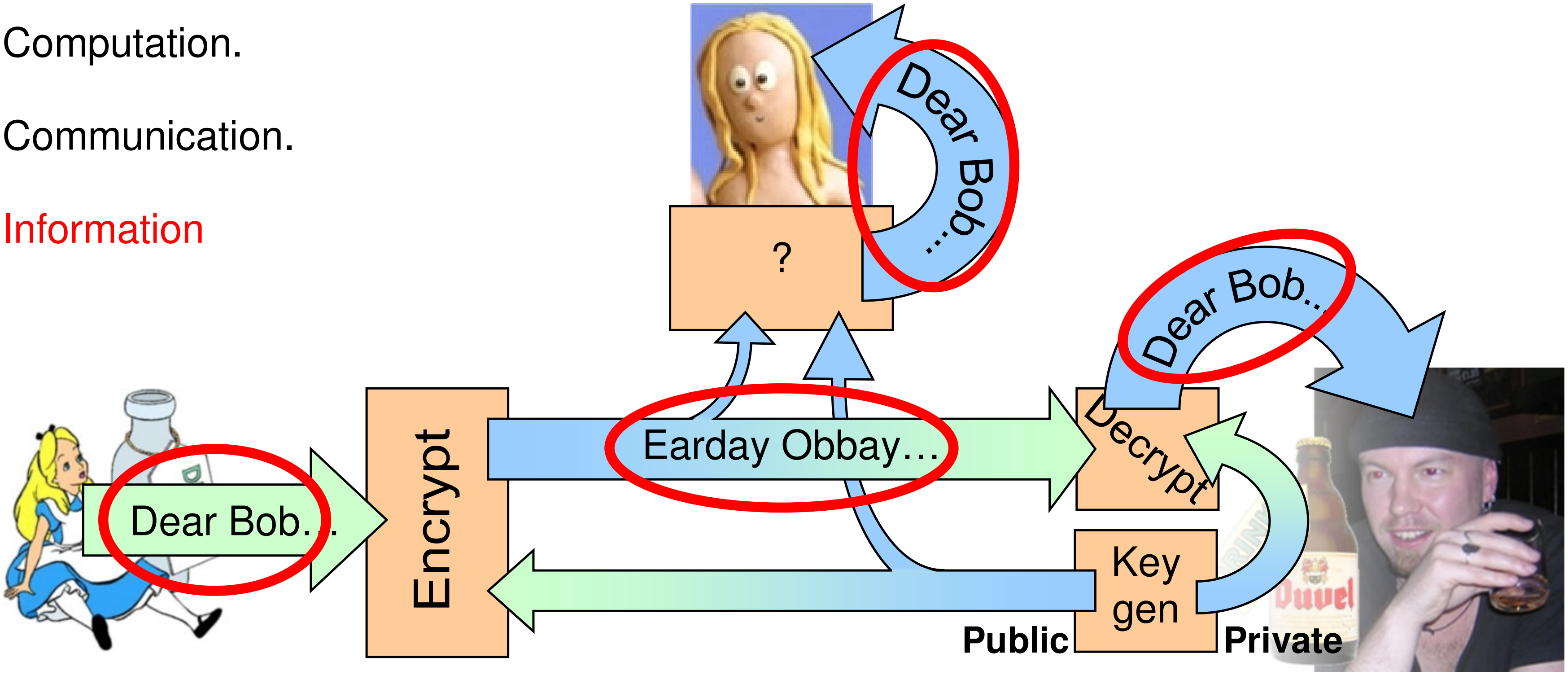
~~Symmetric-key cryptography.~~

## Public-key cryptography.

Computation.

Communication.

Information



~~Communication.~~

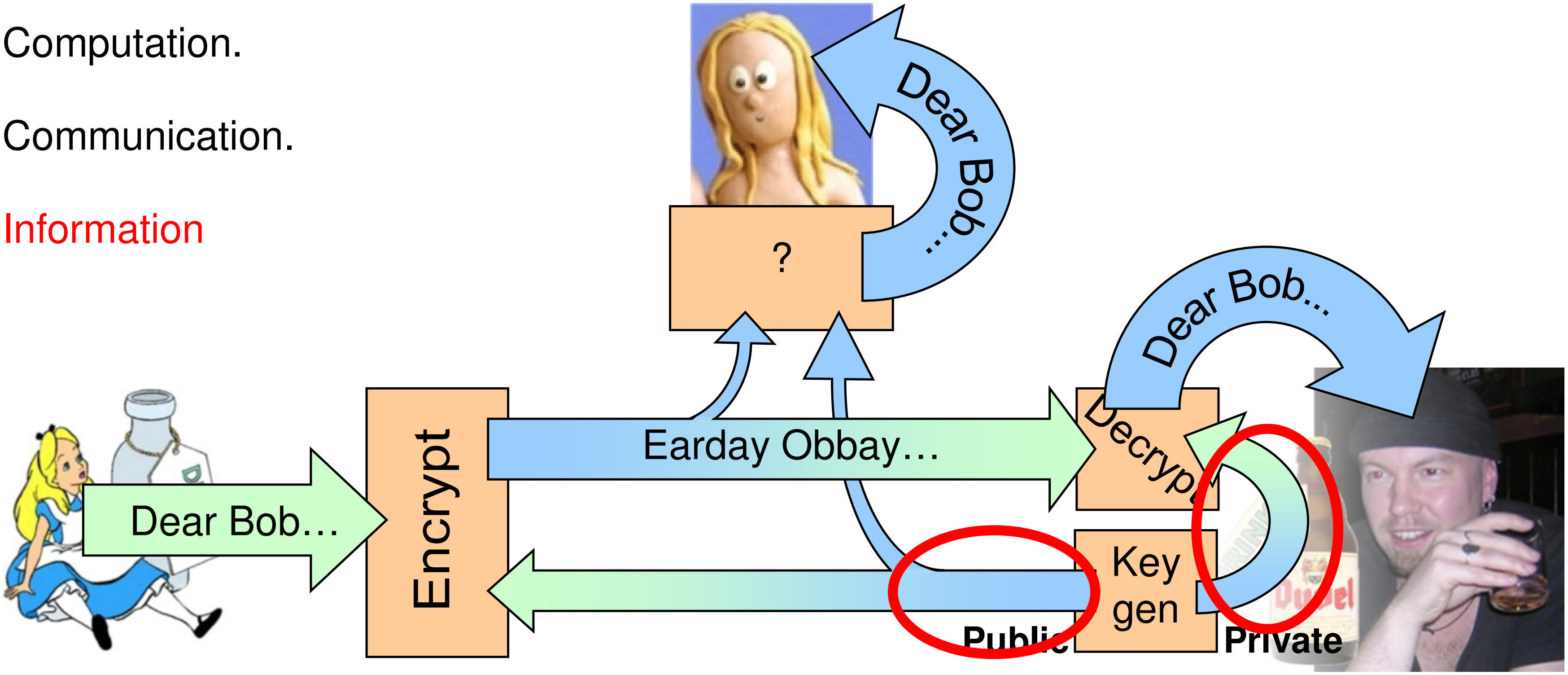
~~Symmetric-key cryptography.~~

## Public-key cryptography.

Computation.

Communication.

Information



~~Communication.~~

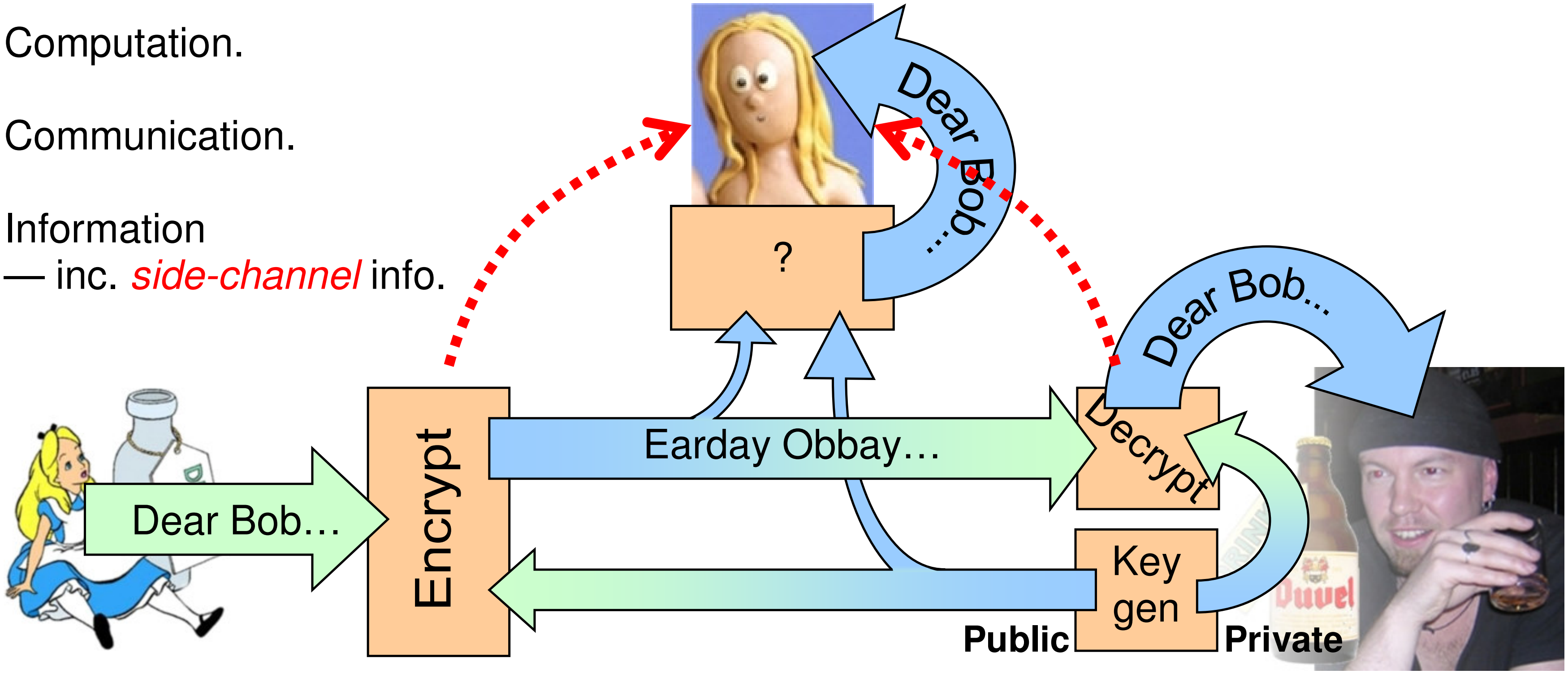
~~Symmetric-key cryptography.~~

## Public-key cryptography.

Computation.

Communication.

Information
— inc. *side-channel* info.



~~Communication.~~

~~Symmetric-key cryptography.~~

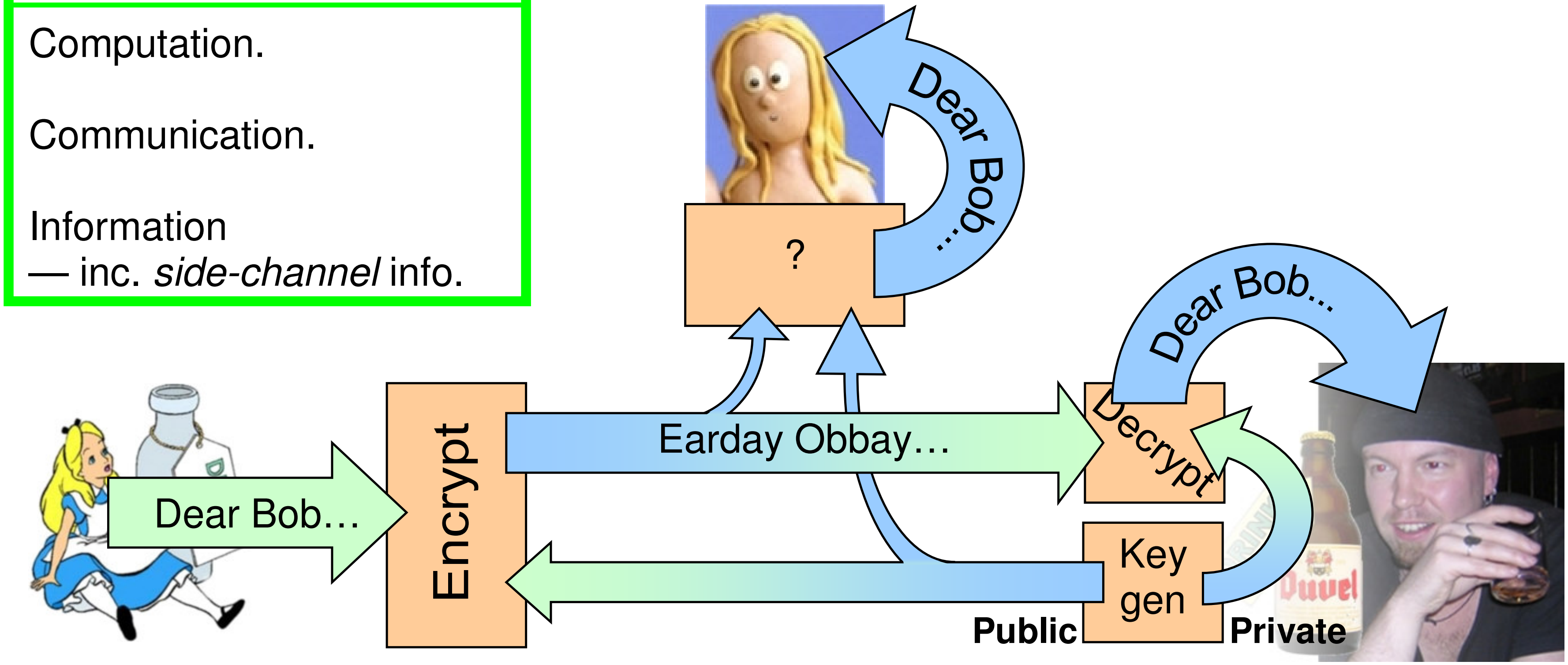
## Public-key cryptography.

**What we want to capture**

Computation.

Communication.

Information
— inc. *side-channel* info.



## Resource of ‘security’.

**Temptation**: to produce some (1-D) quantity (that depends on |key|, say) that’s

- *large* when things are difficult for Eve but easy for Alice and Bob, and
- *small* otherwise.

## Resource of ‘security’.

**Temptation**: to produce some (1-D) quantity (that depends on |key|, say) that’s

- *large* when things are difficult for Eve but easy for Alice and Bob, and
- *small* otherwise.

**However**, boils down to standard **comp. complexity** of Eve’s decryption computation.

# Resource of ‘security’.

**Temptation**: to produce some (1-D) quantity (that depends on |key|, say) that’s

- *large* when things are difficult for Eve but easy for Alice and Bob, and
- *small* otherwise.

e.g. factorization

**However**, boils down to standard **comp. complexity** of Eve’s decryption computation.

## Resource of ‘security’.

**Temptation**: to produce some (1-D) quantity (that depends on |key|, say) that’s

- *large* when things are difficult for Eve but easy for Alice and Bob, and
- *small* otherwise.

e.g. factorization

**However**, boils down to standard **comp. complexity** of Eve’s decryption computation.

**Instead**, maybe want a (multi-D) quantity that reflects

- computational difficulty for Eve,
- computational ease for Alice and Bob,
- information aspects of protocol,
- etc.

# Resource of ‘security’.

**Temptation**: to produce some (1-D) quantity (that depends on |key|, say) that’s

- *large* when things are difficult for Eve but easy for Alice and Bob, and
- *small* otherwise.

**However**, boils down to standard **comp. complexity** of Eve’s decryption computation.

e.g. factorization

**Instead**, maybe want a (multi-D) quantity that reflects

- computational difficulty for Eve,
- computational ease for Alice and Bob,
- information aspects of protocol,
- etc.

i.e. ‘what we want to capture’ (prev. slide)

## Maintaining generality.

**Problem** with using concepts like ‘difficulty for Eve’:

— assumes rigid goody/baddy roles seen in cryptographic protocols, *but not necessarily seen in wider information-theory setting*.

## Maintaining generality.

**Problem** with using concepts like ‘difficulty for Eve’:

— assumes rigid goody/baddy roles seen in cryptographic protocols,

*but not necessarily seen in wider information-theory setting*.

(complexity) (information)

**Instead**, consider how hard agents must compute, what they know, etc. ***without*** using a priori goody/baddy labels.

## Maintaining generality.

**Problem** with using concepts like ‘difficulty for Eve’:

— assumes rigid goody/baddy roles seen in cryptographic protocols,

*but not necessarily seen in wider information-theory setting*.

(complexity) (information)

**Instead**, consider how hard agents must compute, what they know, etc. ***without*** using a priori goody/baddy labels.

Then ***work out*** which agent is Alice, which is Bob, which is Eve based on difficulty, etc.

## Primitives.

Goody/baddy-free approach ⇒ dealing at level of ***primitives***

# Primitives.

One-way fn. Trapdoor fn. Pseudorandom no. gen. etc.

Goody/baddy-free approach $\Rightarrow$ dealing at level of ***primitives***

## Primitives.

One-way fn. Trapdoor fn. Pseudorandom no. gen. etc.

Goody/baddy-free approach ⇒ dealing at level of ***primitives*** rather than
dealing with full-blown protocols with predefined roles.

## Primitives.

One-way fn.    Trapdoor fn.    Pseudorandom no. gen.    etc.

Goody/baddy-free approach ⇒ dealing at level of ***primitives*** rather than
dealing with full-blown protocols with predefined roles.

So, want to consider trade-offs between security and not only *resources*, but also *primitives*.

## Idea.

Want a framework that accommodates such things as

- ***computational resources*** (⇒ complexity),
- ***communication resources***,
- ***primitives*** and
- availability of ***information***.

## Idea.

Want a framework that accommodates such things as

- ***computational resources*** (⇒ complexity),
- ***communication resources***,
- ***primitives*** and
- availability of ***information***.

Gives us a better chance of spotting (e.g.) side-channel attacks than (say) a complexity-only view.

## Idea.

Want a framework that accommodates such things as

- ***computational resources*** (⇒ complexity),
- ***communication resources***,
- ***primitives*** and
- availability of ***information***.

Gives us a better chance of spotting (e.g.) side-channel attacks than (say) a complexity-only view.

Can view a process (computation, comm., etc.) as having ***costs*** in these ‘dimensions’.

# Idea.

Want a framework that accommodates such things as

- ***computational resources*** (⇒ complexity),
- ***communication resources***,
- ***primitives*** and
- availability of ***information***.

Gives us a better chance of spotting (e.g.) side-channel attacks than (say) a complexity-only view.

Can view a process (computation, comm., etc.) as having ***costs*** in these ‘dimensions’.

Many such processes/entities have ‘thickness’ in only one dimension;
if this were true of *all* entities, then framework would decompose and give nothing new.

# Idea.

Want a framework that accommodates such things as

- ***computational resources*** (⇒ complexity),
- ***communication resources***,
- ***primitives*** and
- availability of ***information***.

Gives us a better chance of spotting (e.g.) side-channel attacks than (say) a complexity-only view.

Can view a process (computation, comm., etc.) as having ***costs*** in these ‘dimensions’.

Many such processes/entities have ‘thickness’ in only one dimension;
if this were true of *all* entities, then framework would decompose and give nothing new.

But some special entities—like ***security***—straddle more than one dimension,
and make the structure non-trivial and useful.

## Questions?

*Precision complexity* reference:

***A Model-Independent Theory of Computational Complexity***
`http://users.ox.ac.uk/~quee1871/thesis.pdf`

This research was funded by the EPSRC grant:

***Complexity and Decidability in Unconventional Computational Models*** (EP/G003017/1)

**Ed Blakey**
`http://users.ox.ac.uk/~quee1871/`
`ed.blakey@queens.oxon.org`

## ~~Questions?~~
## Discussion.

*Precision complexity* reference:

***A Model-Independent Theory of Computational Complexity***
`http://users.ox.ac.uk/~quee1871/thesis.pdf`

This research was funded by the EPSRC grant:

***Complexity and Decidability in Unconventional Computational Models*** (EP/G003017/1)

**Ed Blakey**
`http://users.ox.ac.uk/~quee1871/`
`ed.blakey@queens.oxon.org`