# csci54 – discrete math & functional programming
# relations, RSA

# Relations recap

- A (binary) relation on a set A is a subset of AxA
- A relation can be any, or none, of the following:
  - reflexive
  - symmetric
  - transitive
- A relation that is reflexive, symmetric, and transitive is called an equivalence relation
- An equivalence relation on a set A partitions A into a set of equivalence classes

# Practice questions

- For each of the following relations, indicate if it is reflexive, symmetric, and/or transitive. If it's all three and therefore an equivalence relation, describe the equivalence classes.

1. S = all juniors and seniors at Pomona. (x,y) in $R_1$ iff x and y share a major.
2. S = Z. (x,y) in $R_2$ iff x=y.
3. S = {1,2,3,4,5}. $R_3$ = {⟨1, 5⟩, ⟨2, 2⟩, ⟨2, 4⟩, ⟨4, 1⟩, ⟨4, 2⟩}.

- S = all students at Pomona. Define an equivalence relation on S that isn't one of the ones discussed in lecture last time.

# Closures

**Definition 8.11: Reflexive, symmetric, and transitive closures.**

Let $R \subseteq A \times A$ be a relation. Then:

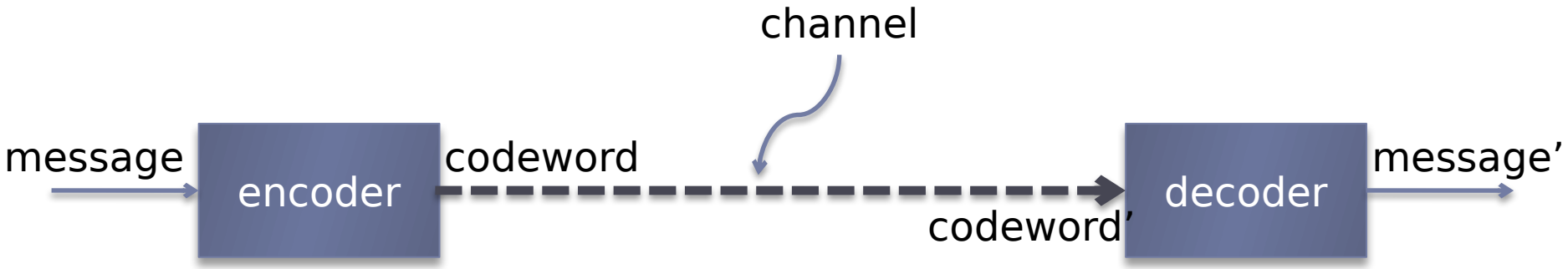
The *reflexive closure of* $R$ is the smallest relation $R' \supseteq R$ such that $R'$ is reflexive.

The *symmetric closure of* $R$ is the smallest relation $R'' \supseteq R$ such that $R''$ is symmetric.

The *transitive closure of* $R$ is the smallest relation $R^+ \supseteq R$ such that $R^+$ is transitive.

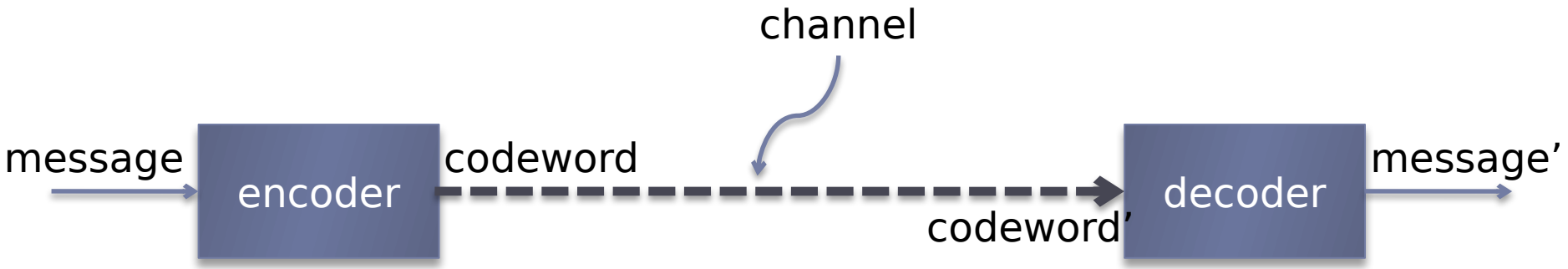
- Consider the relation R = {⟨1, 5⟩, ⟨2, 2⟩, ⟨2, 4⟩, ⟨4, 1⟩, ⟨4, 2⟩} on {1, 2, 3, 4, 5}
  - What is the reflexive closure?
  - What is the symmetric closure?
  - What is the transitive closure?
- S = all students at Pomona. (x,y) in $R_1$ if x and y share a major.

# Transmitting information



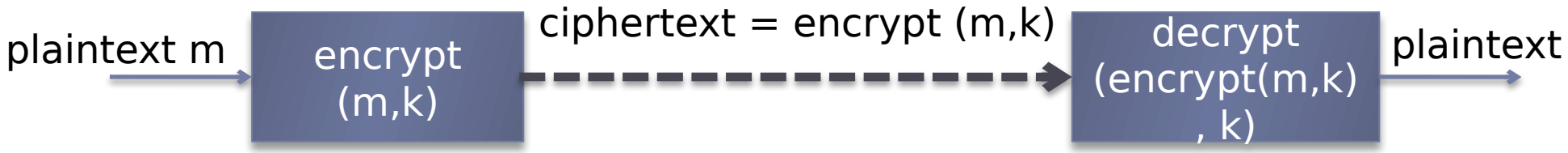

- cryptography
- error correction
- compression

# Transmitting information - cryptography




- goal is to keep someone with access to the channel from finding out information about the message.
- assumptions (for now)
  - message = message'
  - codeword = codeword'
- why?
- how?

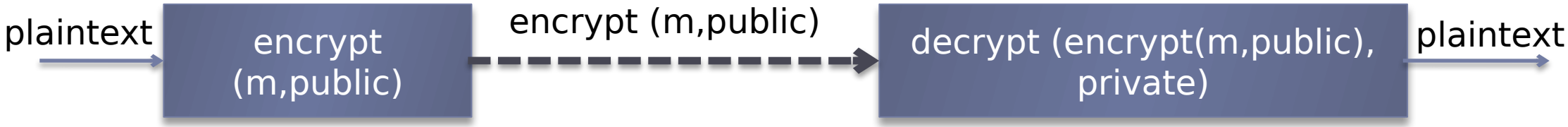
# Private key cryptography




- Symmetric-key algorithms
- The communicating parties share a piece of secret information (the key k)

- Examples?
- Challenges?

# Public key cryptography




- asymmetric-key algorithm

- Everyone publishes their public key.
- If you want to send a message to someone, you encrypt it with their public key.
- When you get a message you can decrypt it with your private key.

# Public key encryption in practice

A

**Richard Adams**

Fingerprint: `3DAF 842A DAAF 190D AB13 D701 DCCE B6EA 2697 15E7`

Email Address: `richard.adams[@]theguardian.com`

PUBLIC KEY

**Esther Addley**

Fingerprint: `081E D0F2 4742 3B23 2C53 B0DA FA39 808F D279 4676`

Email Address: `esther.addley[@]theguardian.com`

PUBLIC KEY

**Jonathan Alden**

Fingerprint: `29C6 09D9 5CF7 1EEE 7549 66F6 CD1B 662A 4686 0F81`

Email Address: `jonathan.alden.freelance[@]theguardian.com`

PUBLIC KEY

**Lorena Allam**

Fingerprint: `36D5 D5B6 0ADF FB83 0528 F6AF 5D00 EECE 797E 0CD7`

Email Address: `Lorena.Allam[@]theguardian.com`

PUBLIC KEY

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.11 (GNU/Linux)

mQENBFIfN90BCADXc0hKEUKuZjLajLygK6unrB6mQlFXVcMLO+LE/FIAyJRdTXP2
PqxpDkSsK7H6DcOZaoUO3jCxAgmjF9MMMztFDWyAZHqVLpwJw+ZqgX94pMsRgmTL
SLFAYSiSa/yPukEkmc1OCwEKUppJ2s1MvRsmr+t0aqlp0QWl1haCmv0hbEiYsXGc
nbW98hnvQGucXoW+9Y7Xcg6xGvCOGNRduaYCmRjro2io9k5C8Qb2SzoSvWGwkVtD
1wx2lcIqhA4AlHXWDQJRcrTpVebAYzSnijPbXyZAOKbPkvwnApI2pU/6qWAF/3yX
2Ld3MWAOTSIzhgdBDlZsmaSeLNRvt9ATpO1ZABEBAAG0LUVzdGhlciBBZGRsZXkg
PGVzdGhlci5hZGRsZXlAdGhlZ3VhcmRpYW4uY29tPokBOAQTAQIAIgUCUh833QIb
LwYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQ+jmAj9J5RnbKXwgAgB2sTnnB
n+pUPQmf9tEMa7y7/X5tRdFOPTekmRTczV1q1o+GfATdIQpCVtlMm+nKKMM6QVka
2nFCs/wzOWAneqp+iw9YzFJ10LV0J6f9YJFXldS2EzP4gJ44T+7d/y30HW3qdecm
v73mxp8YcsCruhk2UEZQn7igL1OqgdNfqgpk9h41KvFV3TPWbz8xTP1HgJ1U8qAL
j5AH8OYEdFwN1BMiu7QsKeY5srD3MqwvQ1v3BxbBCN5rx+fBVZ7iE9guCXoMc5mx
qKk+92IzmLE7gDhX0qcvBGsROBFKQlP5RPCeIxljuS+GS3JU5QrRtTxhg8Zh88jz
IAp2d/tzPBoKUriNBFIfN90BBACp7yZA1Mgj+rYViDK7OVJJzbrfx1weBJtUMvvK
l6E5KJr2VoxNMNk5+YkZi0GSxwP7U2urrtWXe3yJlxCoJp916jvaM92ImlD8FkP4
tmJxuIvmdnHH0+S7eqq286eti18IuHFosFN2tUs9xS8WWYB1aBmbAzujrrkLIvnB
3LzvqwARAQABiQG9BBgBAgAJBQJSHzfdAhsuAKgJEPo5gI/SeUZ2nSAEGQECAAYF
AlIfN90ACgkQt0F2F+VPJGBSmAP/d6q4czB+kbaGR15dWHjPO/Si93YzJjYAhyhq
w7Bf8Qv2LXiput1XQRk9EK7Pl7NnRMxb2oTv7yURIXG+O4qrrvS7lpO9ygNYULLP
OE2NEcSySfs2ayur1MHdzD+RsyaGLMfPBh4D4w5SSptc5xCzEYie5h8L6APMWGAr
PIcFUqe1fwf/ascMLMyHOLeWSq9lQaFJNl7WiwXBQeXmXFwH2U+R4tjPLwBGceQ+
Z8NsTMCQ7t0s0dn5YvbLFhfuozQs2U2QlIS5UQgQwf+xChaw3eJ9ICdaZRNX9c8f
C17LI3Z7aqbZrDgUU1ox6lTO7FZCrSNFSFUpNXhKVXlrTygcdfnbPiqRrsGQr85I
dMvTvFKfQglxQSITF9YkKD8yX3Opy5KR9mlOlg5olV0z9KEpg2vNpucDc2Nn3DKY
E6oV5JBlQpYh2GQgESY/I/bxz8ZHvoZ7+SH3DZ+ufrXEzBcAtGHYFhnNgikS63EX
zk5k0nDQnf1qOiZ3mncYU8hNntu3SgumSw==
=O+zV
-----END PGP PUBLIC KEY BLOCK-----
```

## Encrypted Email

If you use PGP encryption, here is our fingerprint and link to our public key. If you use our public key with a mail encryption plugin, for example Mailvelope or Enigmail, this encrypts the contents of your message but not the subject line or the name of the sender.

Fingerprint:
**EC6C 2905 F0F9 3C03 7394 6CA1 0642 427A 5FF7 80BE**

Email: **lockbox@washpost.com**

The Post's public key

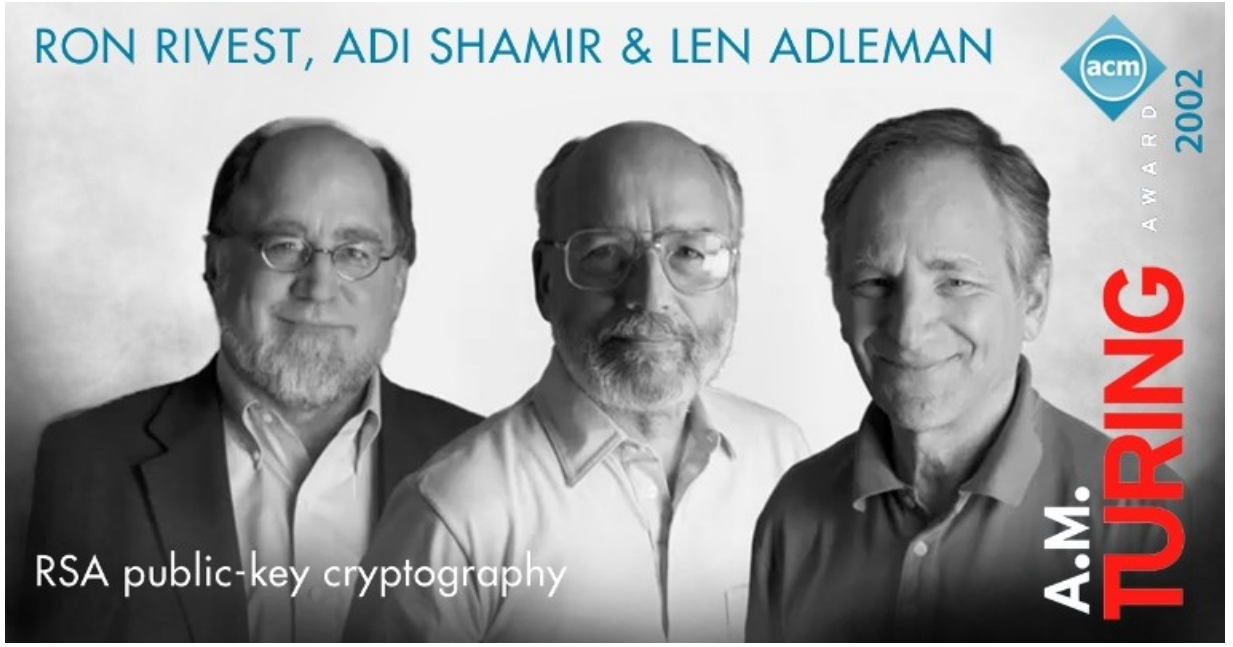
Copy


—BEGIN PGP PUBLIC KEY BLOCK—
mQGNBGLr60kBDAC7/dyy27fxfbaE1Ss13QI9II93YePYFNjLW1JonvNcsmN+ncuA
5u8HZJQFo9ICtytfMIzEwW6JwcTVFY5TvZcjDi/8FtNzpCCFmnzkCZP1TVXo5xGL
V7HC3rzpJSP8n3vcHO7xCPbBsBdzVrzA6QQZCDTnlCITBYHdFZYb7qT9NGD34mP
b +gmhzBNxZf8YfJ3jj7H+Bq3dz2laDi/IHg7+TnfvOGwHJuA4uMMPxWTXhZFZv2to
YpuYPgj+pfwG0m4fTQEEjc8BK2xpCl3oOsgg+IhHKtpyJ2GF43ee8IBBFMlcZNSK
xGo7676QYM8bp9TuBB6qGiNeML08EIB5OLYYFnIlAWxHyx5DbSdSYFGEAnaJnH
3K WrvPI5/YvlVsa8ulYK3gcyLIJI3VW3PBHvU7lsH/oOrl6fprTERuaBfkd5xgJlvF
FG+VLBOZnTnQ4ap7wXVY5Omje4BACBqWzKuyVtuCyKdP3j3fYaMymdrwomFIAb
hl q5LcZATSTdApSSsAEQEAAbQlVFdQIExvY2tib3ggPGxvY2tib3hAd2FzaHBvc3Qu
Y29tPokB6wQTAQgAPhYhBOxsKQXw+TwDc5RsoQZCQnpf94C+BQJl6+tJAhsDBQ
kD
w7o3BQsJCAcCBhUKCQgLAgQWAgMBAh4BAheAACEJEAZCQnpf94C+FlEE7GwpBf
D5
PANzlGyhBkJCel/3gL5TgQv+P3OaInPOlYz2sTLVninPd8s9guhBKvoR1b2k0oA4
iS2g/sONY109CC4SWIUJVxqaVLFhDi3x5g/tgWOzv51pKGuKZuzlmS456Z0ofIvw
bJuHHc9BypTA7GNqFEp7yITL3H1BTeYXcWqzTIqAwYrvkDzbfjIRd4nDgfoJffHI
HDEwOun/UFaUK6TpBS5HSzSrthxQxRQ2Gq05plvA9QWmaN7U1et9eZoy2q76bv
6T
IJ2yAse/VN6BE4txcbPmFBF9ZLWhDs+gtzpMWeaLqK11tlyGvWZ1j64ncVs3K7O/
NYfnaaYIjuKlpF+fvzrIiS8yoe8FX3AFOSWYe6hk13GFgceF7AbhIAlDyRSJSsVQ
yY4yrtuTlsSP8m5bQI71VvX1Mw7n9eEwc5XeZ77ndbVOFDBz3OesOXd7e/RcGBLz
fuIldKJVuMW8T78X34Ide6Ow2/6rIk41tebQMaCgcK4dEOu1JIG4fChEZh09usLc
nLxn6PGUqDcAZIrIBWonb2H5uQGNBGLr60kBDACnIsLKmY1Hu15iEWcfUOIeArFf
4saw/gTBYcne2uKQRFflmq7I6W7I3aIEqCaezkUZF3soikng6h1PqE7DW+9uzOWr9
rpfIF2+PakFaTLUCbcIYdh/mxltXeAjadAkVtcGVJKOEb5OtvSOpK39dFlsnhm63
t7/G/aFICIAWRmmvMzsXeKdH+GVXF4NbKH+q6d9hPuxIBP92wYOeo/630JJTXlq
J
OmuqM2BYyodb9RqXkYOZcgkTm0XuXUoHseIPhlrReWzoZtsa16zL1aCgoz5BeqG
w
rBoE9EatsexexpAJP7Jt7VzZOJyF4tGXQRkmfkWCwOxnTQWAave1xvdwk3VYB6cH
NkN4WaF4TD4Wx+xBadMAOnKV2vOZbNNPMHYsUsLKNy1Lv15FK5nAYN+o26u
OAiFF
o3lMNBwI9QqLTeRhgvwxMeIO9UdrV+bxziGIFMDkyrd62b6qw4evTLi6QzT9f9/5
1vNfmTpW1E44iChMQB64hrDJ7TWstSV+4JDje+EAEQEAAYkBOwQYAQgAJhYhBO
xs
KQXw+TwDc5RsoQZCQnpf94C+BQJl6+tJAhsMBQkDw7o3ACEJEAZCQnpf94C+FIE
E 7GwpBfD5PANzlGyhBkJCel/3gL7uXwwAhxceVQGfug5U7ZmKIzBjgCcF4VhITaFt
iGMKP1WJO7jPXkX9qReYWARKpcW2u16crg8fndAeKHguOt1KRnCJaTnxWUHMqX
+z uOX6l4GSdxIvrkv3arqVz48doxNW2ph1u7dV46j3MFTquJjZnkI77rUfaoCnb2OY
UOdR+1LAPLqf4U9fdndWd0DJwoK6pulALFZfHlx1PqDGaO5gRcBPNaCPlGhjI3Uv


https://www.theguardian.com/pgp
https://www.washingtonpost.com/anonymous-news-tips/

# RSA algorithm

- A very widely used public key encryption algorithm
- Three algorithmic components
  - key generation
  - encryption
  - decryption
- Our plan
  - What is the algorithm?
  - Why does it work?
  - How to implement it efficiently?



# Modular arithmetic – definitions and properties

$a \equiv b \pmod m$ if and only if $a \bmod m = b \bmod m$

i.e., exists x, y, z in Z: $a = x * m + z \wedge b = y * m + z$

Some useful facts from Wikipedia:

(if $a1 \equiv a2 \pmod m$ and $b1 \equiv b2 \pmod m$):

- $a + k \equiv b + k \pmod m$ for any integer k (compatibility with translation)
- $k\,a \equiv k\,b \pmod m$ for any integer k (compatibility with scaling)
- $k\,a \equiv k\,b \pmod{k * m}$ for any integer k
- $a1 + a2 \equiv b1 + b2 \pmod m$ (compatibility with addition)
- $a1 - a2 \equiv b1 - b2 \pmod m$ (compatibility with subtraction)
- $a1 * a2 \equiv b1 * b2 \pmod m$ (compatibility with multiplication)
- $a^k \equiv b^k \pmod m$ for any non-negative integer k (compatibility with exponentiation)