CMSC/Math 456: Cryptography (Fall 2022)

Lecture 1

Daniel Gottesman

What is This Class About?

Cryptography is about how to protect information against an untrusted "adversary."

We will learn how to make unbreakable codes

… and then we will learn how to break them.

We will learn about what it means for a cryptographic protocol to be secure or insecure and about the advantages and limitations of security proofs.

Cryptography is not just about encryption. We will also learn about other ways to protect information, such as authentication.

We will learn about real-world protocols like AES and RSA

… and why you shouldn't try to make your own cryptographic protocols without a lot more training than this class.

Cryptography is Hard

In cryptography, there is an intelligent opponent who is actively looking for ways to circumvent your cryptographic protocol. This means that even seemingly small mistakes can lead to a complete loss of security.

> Governments spend billions of dollars per year on cryptography, both to make secure codes and to break them.

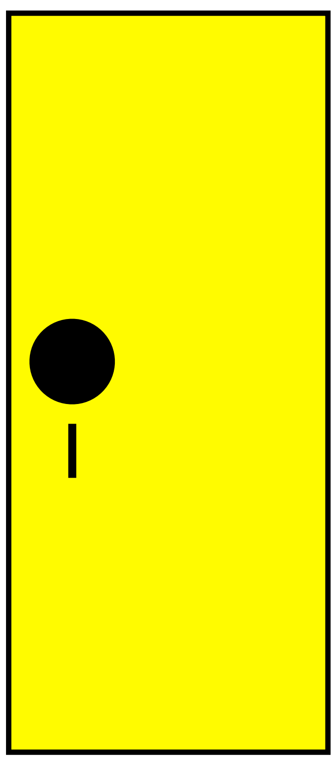
You will need:

- Programming experience (C, C++, Java, Python preferred)
- Analysis of algorithms (e.g., big-O notation)
- Probability and discrete math, particularly modular arithmetic
- Some experience with rigorous proofs

Professional cryptographers need much more number theory and other math (e.g., elliptic curves).

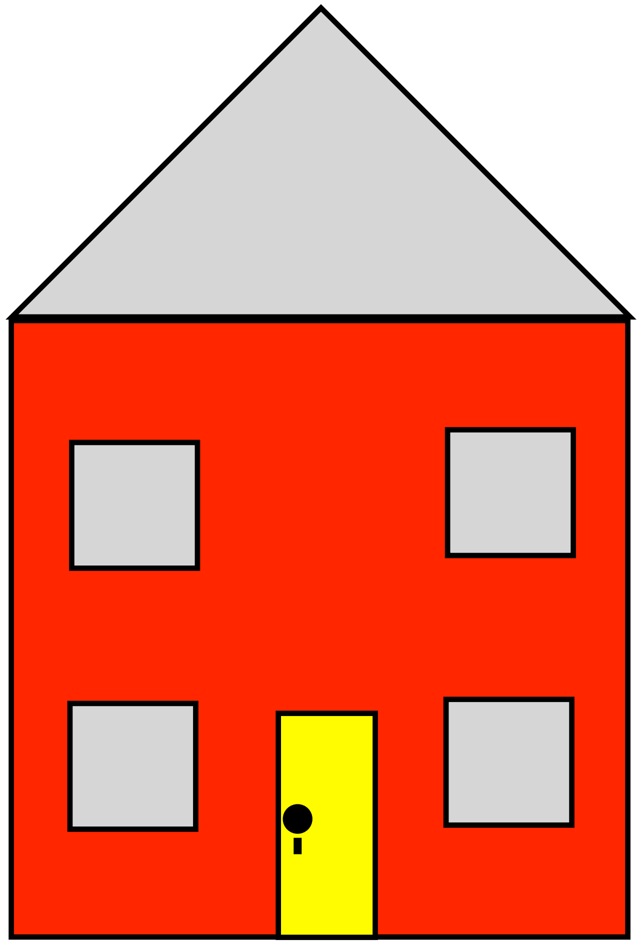
Cryptography vs. Computer Security

Cryptography is the study of concrete protocols to protect information in a specific way against adversaries.



Cryptography is about making secure locks and doors.

Cybersecurity is the study of security of the computer system as a whole.



Security is about making sure there is not another way into the house.

Important Websites

Course web page: https://www.cs.umd.edu/class/fall2022/cmsc456/

Slides and homeworks will be posted here. Also all this basic information.

Piazza: http://piazza.com/umd/fall2022/cmsc456

Out-of-class discussions and questions should be posted here. This makes it possible for any of us (me, TAs) to answer and lets all students see the answer (but you can ask questions privately or anonymously also).

Gradescope:

Homework will be turned in and graded here.

Course ELMS page: CMSC456-0201/MATH450-0201: Cryptography-Fall 2022 dgottesm

Recorded lectures will be available here.

UMD course policies: https://www.ugst.umd.edu/courserelatedpolicies.html

Instructor, TAs, Office Hours

Instructor: Daniel Gottesman

E-mail: dgottesm@umd.edu
Office hours: Tuesday 10:30-11:30 AM, Atlantic 3251

TAs:

Amadeo David De La Vega Parra

E-mail: adelaveg@umd.edu
Office hours: Thursday 9:15-10:45 AM, AVW 4122

Samira Goudarzi

E-mail: samirag@umd.edu
Office hours: Wednesday 2:00-3:30 PM, AVW 4122

Mahathi Vempati

E-mail: mahathi@umd.edu
Office hours: Monday 1:30-2:30 PM, AVW 4122

It may be possible to Zoom into some of these office hours.
Contact the appropriate person to find out and arrange.

Grading

Problem Sets: 30%

- A mix of theory problems and programming assignments.
- Drop highest and lowest grades and average remaining scores.
- If you collaborate or use external sources (not lectures or textbook), cite your sources.
- Extensions require prior approval from instructor or a TA, plus a good reason. Leave 24 hours to ensure time for a response. Maximum extension 1 week.

Midterm: 30%

Thursday, October 20 (in class)

Final exam: 40%

Monday, December 19, 1:30-3:30 PM

Additional details of midterm and final to be announced later.

Class Materials

Textbook: Katz & Lindell, *Introduction to Modern Cryptography, 3rd ed.*

When I post slides and homeworks, I will also indicate the relevant section of the textbook.

Lectures: Slides will be posted on the course web page following each class. I will also attempt to record the lectures.

However, I strongly encourage you to make a habit of attending class whenever you can.

- Some class recordings I made last year were not usable.
- You will be more engaged with the class if you attend in person.
- You will have the opportunity to ask questions and follow-ups in real time instead of with some delay.
- You will not be tempted to procrastinate watching the recordings.

Course Outline

1. Classical cryptography

   Before the 1970s, cryptography was mostly ad hoc, without too much math or rigorous definitions.

2. Modern private key cryptography

   Central tools and main protocols of modern private-key encryption. Also rigorous definitions and proofs of security.

3. Public key cryptography

   Secure encryption where anyone can send to you.

4. Authentication (message authentication and digital signatures)

   Cryptography is not just about encryption. The next most important class of protocols ensures messages are authentic.

5. Advanced topics, as time allows

   Possibilities include: post-quantum cryptography, quantum key distribution, secure multiparty computation, homomorphic encryption, blockchain

A ciphertext

```
WNYTH NGZCZ HNPMN WQZHW NYTHN GZYPE HNPMN WQZHW NYTHN
GZTIZ PMYWH BPQWN YTHNG ZTIZP MMPPO WHGRZ HHWNY THNGZ
ZDPKG PMCZO WZMWN YTHNG ZZDPK GPMWR KEZBL OWNAW NYTHN
GZHZT HPRPM OWIGN WNYTH NGZHZ THPRP MBTEV RZHHW NYTHN
GZHDE WRIPM GPDZW NYTHN GZYWR NZEPM BZHDT WEYZG TBZXZ
EANGW RICZM PEZLH YZGTB RPNGW RICZM PEZLH YZYZE ZTOOI
PWRIB WEZKN NPGZT XZRYZ YZEZT OOIPW RIBWE ZKNNG ZPNGZ
EYTAW RHGPE NNGZD ZEWPB YTHHP MTEOW VZNGZ DEZHZ RNDZE
WPBNG TNHPQ ZPMWN HRPWH WZHNT LNGPE WNWZH WRHWH NZBPR
WNHCZ WRIEZ KZWXZ BMPEI PPBPE MPEZX WOWRN GZHLD ZEOTN
WXZBZ IEZZP MKPQD TEWHP RPROA NGZEZ YZEZT VWRIY WNGTO
TEIZS TYTRB TFLZZ RYWNG TDOTW RMTKZ PRNGZ NGEPR ZPMZR
IOTRB NGZEZ YZEZT VWRIY WNGTO TEIZS TYTRB TFLZZ RYWNG
TMTWE MTKZP RNGZN GEPRZ PMMET RKZWR CPNGK PLRNE WZHWN
YTHKO ZTEZE NGTRK EAHNT ONPNG ZOPEB HPMNG ZHNTN ZDEZH
ZEXZH PMOPT XZHTR BMWHG ZHNGT NNGWR IHWRI ZRZET OYZEZ
HZNNO ZBMPE ZXZE
```

Patterns in the ciphertext create an insecurity in the code

Letter Frequencies

Ciphertext

Letter	# times	%
Z	110	15.0%
N	74	10.0%
W	59	8.0%
P	58	7.9%
T	57	7.8%
H	55	7.5%
E	48	6.5%
G	44	6.0%
R	43	5.9%
Y	31	4.2%
M	28	3.8%
O	22	3.0%
B	20	2.7%
I	20	2.7%
K	13	1.8%
D	12	1.6%
L	8	1.1%
X	8	1.1%
C	6	0.8%
A	5	0.7%
Q	5	0.7%
V	4	0.5%
F	2	0.3%
S	2	0.3%
J	0	0%
U	0	0%

English

Letter	%
e	12.7%
t	9.1%
a	8.2%
o	7.5%
i	7.0%
n	6.7%
s	6.3%
h	6.1%
r	6.0%
d	4.3%
l	4.0%
c	2.8%
u	2.8%
m	2.4%
w	2.4%
f	2.2%
g	2.0%
y	2.0%
p	1.9%
b	1.5%
v	1.0%
k	0.8%
j	0.2%
x	0.2%
q	0.1%
z	0.1%

Distribution of letters in the ciphertext not too far from English with some statistical variation.

Maybe this is a substitution cipher? That is, each English letter is replaced by a corresponding letter, always the same throughout the ciphertext.

Why English and not, say, French? This class is in English, so seems a reasonable guess.

We can use external information to help break the code.

Substitute e for Z

```
WNYTH NGeCe HNPMN WQeHW NYTHN GeYPE HNPMN WQeHW NYTHN
GeTIe PMYWH BPQWN YTHNG eTIeP MMPPO WHGRe HHWNY THNGe
eDPKG PMCeO WeMWN YTHNG eeDPK GPMWR KEeBL OWNAW NYTHN
GeHeT HPRPM OWIGN WNYTH NGeHe THPRP MBTEV ReHHW NYTHN
GeHDE WRIPM GPDeW NYTHN GeYWR NeEPM BeHDT WEYeG TBeXe
EANGW RICeM PEeLH YeGTB RPNGW RICeM PEeLH YeYeE eTOOI
PWRIB WEeKN NPGeT XeRYe YeEeT OOIPW RIBWE eKNNG ePNGe
EYTAW RHGPE NNGeD eEWPB YTHHP MTEOW VeNGe DEeHe RNDeE
WPBNG TNHPQ ePMWN HRPWH WeHNT LNGPE WNWeH WRHWH NeBPR
WNHCe WRIEe KeWXe BMPEI PPBPE MPEeX WOWRN GeHLD eEOTN
WXeBe IEeeP MKPQD TEWHP RPROA NGeEe YeEeT VWRIY WNGTO
TEIeS TYTRB TFLee RYWNG TDOTW RMTKe PRNGe NGEPR ePMeR
IOTRB NGeEe YeEeT VWRIY WNGTO TEIeS TYTRB TFLee RYWNG
TMTWE MTKeP RNGeN GEPRe PMMET RKeWR CPNGK PLRNE WeHWN
YTHKO eTEeE NGTRK EAHNT ONPNG eOPEB HPMNG eHNTN eDEeH
eEXeH PMOPT XeHTR BMWHG eHNGT NNGWR IHWRI eReET OYeEe
HeNNO eBMPE eXeE
```

Lower case will signify plaintext. Also, I have colored the next 5 most common letters, NWPTH, as brown.

Digraphs and Trigraphs

```
WNYTH NGeCe HNPMN WQeHW NYTHN GeYPE HNPMN WQeHW NYTHN
GeTIe PMYWH BPQWN YTHNG eTIeP MMPPO WHGRe HHWNY THNGe
eDPKG PMCeO WeMWN YTHNG eeDPK GPMWR KEeBL OWNAW NYTHN
GeHeT HPRPM OWIGN WNYTH NGeHe THPRP MBTEV ReHHW NYTHN
GeHDE WRIPM GPDeW NYTHN GeYWR NeEPM BeHDT WEYeG TBeXe
EANGW RICeM PEeLH YeGTB RPNGW RICeM PEeLH YeYeE eTOOI
PWRIB WEeKN NPGeT XeRYe YeEeT OOIPW RIBWE eKNNG ePNGe
EYTAW RHGPE NNGeD eEWPB YTHHP MTEOW VeNGe DEeHe RNDeE
WPBNG TNHPQ ePMWN HRPWH WeHNT LNGPE WNWeH WRHWH NeBPR
WNHCe WRIEe KeWXe BMPEI PPBPE MPEeX WOWRN GeHLD eEOTN
WXeBe IEeeP MKPQD TEWHP RPROA NGeEe YeEeT VWRIY WNGTO
TEIeS TYTRB TFLee RYWNG TDOTW RMTKe PRNGe NGEPR ePMeR
IOTRB NGeEe YeEeT VWRIY WNGTO TEIeS TYTRB TFLee RYWNG
TMTWE MTKeP RNGeN GEPRe PMMET RKeWR CPNGK PLRNE WeHWN
YTHKO eTEeE NGTRK EAHNT ONPNG eOPEB HPMNG eHNTN eDEeH
eEXeH PMOPT XeHTR BMWHG eHNGT NNGWR IHWRI eReET OYeEe
HeNNO eBMPE eXeE
```

A digraph is a pair of letters; a trigraph is a set of three letters. The most common trigraph in English is "the". In our ciphertext, the most common trigraph ending in "e" is "NGe". Maybe that is it?

N = t, G = h

```
WtYTH theCe HtPMt WQeHW tYTHt heYPE HtPMt WQeHW tYTHt
heTIe PMYWH BPQWt YTHth eTIeP MMPPO WHhRe HHWtY THthe
eDPKh PMCeO WeMWt YTHth eeDPK hPMWR KEeBL OWtAW tYTHt
heHeT HPRPM OWIht WtYTH theHe THPRP MBTEV ReHHW tYTHt
heHDE WRIPM hPDeW tYTHt heYWR teEPM BeHDT WEYeh TBeXe
EAthW RICeM PEeLH YehTB RPthW RICeM PEeLH YeYeE eTOOI
PWRIB WEeKt tPheT XeRYe YeEeT OOIPW RIBWE eKtth ePthe
EYTAW RHhPE ttheD eEWPB YTHHP MTEOW Vethe DEeHe RtDeE
WPBth TtHPQ ePMWt HRPWH WeHtT LthPE WtWeH WRHWH teBPR
WtHCe WRIEe KeWXe BMPEI PPBPE MPEeX WOWRt heHLD eEOTt
WXeBe IEeeP MKPQD TEWHP RPROA theEe YeEeT VWRIY WthTO
TEIeS TYTRB TFLee RYWth TDOTW RMTKe PRthe thEPR ePMeR
IOTRB theEe YeEeT VWRIY WthTO TEIeS TYTRB TFLee RYWth
TMTWE MTKeP Rthet hEPRe PMMET RKeWR CPthK PLRtE WeHWt
YTHKO eTEeE thTRK EAHtT OtPth eOPEB HPMth eHtTt eDEeH
eEXeH PMOPT XeHTR BMWHh eHthT tthWR IHWRI eReET OYeEe
HettO eBMPE eXeE
```

"er" and "re" are both common digraphs as well. "E" is the most common undecoded letter that appears before and after "e" in the ciphertext. But a longer ciphertext would help …

E = r

```
WtYTH theCe HtPMt WQeHW tYTHt heYPr HtPMt WQeHW tYTHt
heTIe PMYWH BPQWt YTHth eTIeP MMPPO WHhRe HHWtY THthe
eDPKh PMCeO WeMWt YTHth eeDPK hPMWR KreBL OWtAW tYTHt
heHeT HPRPM OWIht WtYTH theHe THPRP MBTrV ReHHW tYTHt
heHDr WRIPM hPDeW tYTHt heYWR terPM BeHDT WrYeh TBeXe
rAthW RICeM PreLH YehTB RPthW RICeM PreLH YeYer eTOOI
PWRIB WreKt tPheT XeRYe YereT OOIPW RIBWr eKtth ePthe
rYTAW RHhPr ttheD erWPB YTHHP MTrOW Vethe DreHe RtDer
WPBth TtHPQ ePMWt HRPWH WeHtT LthPr WtWeH WRHWH teBPR
WtHCe WRIre KeWXe BMPrI PPBPr MPreX WOWRt heHLD erOTt
WXeBe IreeP MKPQD TrWHP RPROA there YereT VWRIY WthTO
TrIeS TYTRB TFLee RYWth TDOTW RMTKe PRthe thrPR ePMeR
IOTRB there YereT VWRIY WthTO TrIeS TYTRB TFLee RYWth
TMTWr MTKeP Rthet hrPRe PMMrT RKeWR CPthK PLRtr WeHWt
YTHKO eTrer thTRK rAHtT OtPth eOPrB HPMth eHtTt eDreH
erXeH PMOPT XeHTR BMWHh eHthT tthWR IHWRI eRerT OYere
HettO eBMPr eXer
```

"an", "in", and "on" are also very common digraphs and we haven't decoded any of "a", "i", "o", or "n". So let us try to see what "n" could be — maybe "H"? "TH" and "WH" both are common. (No "PH")

Try H = n

WtYTn theCe ntPMt WQenW tYTnt heYPr ntPMt WQenW tYTnt
heTIe PMYWn BPQWt YTnth eTIeP MMPPO WnhRe nnWtY Tnthe
eDPKh PMCeO WeMWt YTnth eeDPK hPMWR KreBL OWtAW tYTnt
heneT nPRPM OWIht WtYTn thene TnPRP MBTrV RennW tYTnt
henDr WRIPM hPDeW tYTnt heYWR terPM BenDT WrYeh TBeXe
rAthW RICeM PreLn YehTB RPthW RICeM PreLH YeYer eTOOI
PWRIB WreKt tPheT XeRYe YereT OOIPW RIBWr eKtth ePthe
rYTAW RnhPr ttheD erWPB YTnnP MTrOW Vethe Drene RtDer
WPBth TtnPQ ePMWt nRPWn WentT LthPr WtWen WRnWn teBPR
WtnCe WRIre KeWXe BMPrI PPBPr MPreX WOWRt henLD erOTt
WXeBe IreeP MKPQD TrWnP RPROA there YereT VWRIY WthTO
TrIeS TYTRB TFLee RYWth TDOTW RMTKe PRthe thrPR ePMeR
IOTRB there YereT VWRIY WthTO TrIeS TYTRB TFLee RYWth
TMTWr MTKeP Rthet hrPRe PMMrT RKeWR CPthK PLRtr WenWt
YTnKO eTrer thTRK rAntT OtPth eOPrB nPMth entTt eDren
erXen PMOPT XenTR BMWnh enthT tthWR InWRI eRerT OYere
nettO eBMPr eXer

Doesn’t seem to work … Maybe “n” is a slightly less frequent letter like “R”? “WR,” “PR,” and “TR” all appear multiple times.

Note: trying different things is a useful code-breaking strategy.

Try R = n

```
WtYTH theCe HtPMt WQeHW tYTHt heYPr HtPMt WQeHW tYTHt
heTIe PMYWH BPQWt YTHth eTIeP MMPPO WHhne HHWtY THthe
eDPKh PMCeO WeMWt YTHth eeDPK hPMWn KreBL OWtAW tYTHt
heHeT HPnPM OWIht WtYTH theHe THPnP MBTrV neHHW tYTHt
heHDr WnIPM hPDeW tYTHt heYWn terPM BeHDT WrYeh TBeXe
rAthW nICeM PreLH YehTB nPthW nICeM PreLH YeYer eTOOI
PWnIB WreKt tPheT XenYe YereT OOIPW nIBWr eKtth ePthe
rYTAW nHhPr ttheD erWPB YTHHP MTrOW Vethe DreHe ntDer
WPBth TtHPQ ePMWt HnPWH WeHtT LthPr WtWeH WnHWH teBPn
WtHCe WnIre KeWXe BMPrI PPBPr MPreX WOWnt heHLD erOTt
WXeBe IreeP MKPQD TrWHP nPnOA there YereT VWnIY WthTO
TrIeS TYTnB TFLee nYWth TDOTW nMTKe Pnthe thrPn ePMen
IOTnB there YereT VWnIY WthTO TrIeS TYTnB TFLee nYWth
TMTWr MTKeP nthet hrPne PMMrT nKeWn CPthK PLntr WeHWt
YTHKO eTrer thTnK rAHtT OtPth eOPrB HPMth eHtTt eDreH
erXeH PMOPT XeHTR BMWHh eHthT tthWn IHWRI enerT OYere
HettO eBMPr eXer
```

If "W", "P", and "T" are "a", "i", and "o", which is which? This circled part doesn't seem to work except for "P" = "o", so let's try that too. And then maybe our other common letter "H" is "s".

P = o and H = s

```
WtYTs theCe stoMt WQesW tYTst heYor stoMt WQesW tYTst
heTIe oMYWs BoQWt YTsth eTIeo MMooO Wshne ssWtY Tsthe
eDoKh oMCeO WeMWt YTsth eeDoK hoMWn KreBL OWtAW tYTst
heseT sonoM OWIht WtYTs these Tsono MBTrV nessW tYTst
hesDr WnIoM hoDeW tYTst heYWn teroM BesDT WrYeh TBeXe
rAthW nICeM oreLs YehTB nothW nICeM oreLs YeYer eTOOI
oWnIB WreKt toheT XenYe YereT OOIoW nIBWr eKtth eothe
rYTAW nshor ttheD erWoB YTsso MTrOW Vethe Drese ntDer
WoBth TtsoQ eoMWt snoWs WestT Lthor WtWes WnsWs teBon
WtsCe WnIre KeWXe BMorI ooBor MoreX WOWnt hesLD erOTt
WXeBe Ireeo MKoQD TrWso nonOA there YereT VWnIY WthTO
TrIeS TYTnB TFLee nYWth TDOTW nMTKe onthe thron eoMen
IOTnB there YereT VWnIY WthTO TrIeS TYTnB TFLee nYWth
TMTWr MTKeo nthet hrone oMMrT nKeWn CothK oLntr WesWt
YTsKO eTrer thTnK rAstT Ototh eOorB soMth estTt eDres
erXes oMOoT XesTn BMWsh esthT tthWn IsWnI enerT OYere
settO eBMor eXer
```

We need more text to continue with frequency analysis, but at this point we can start to look for sensible words and phrases to complete. E.g., "thereYere" = "there were"? "thTtthWn" = "that thin…"? Then probably "Y" = "w", "W" = "i" and "T" = "a".

Y = w, W = i, T = a

```
itwas theCe stoMt iQesi twast hewor stoMt iQesi twast
heaIe oMwis BoQit wasth eaIeo MMooO ishne ssitw asthe
eDoKh oMCeO ieMit wasth eeDoK hoMin KreBL OitAi twast
hesea sonoM OiIht itwas these asono MBarV nessi twast
hesDr inIoM hoDei twast hewin teroM BesDa irweh aBeXe
rAthi nICeM oreLs wehaB nothi nICeM oreLs wewer eaOOI
oinIB ireKt tohea Xenwe werea OOIoi nIBir eKtth eothe
rwaAi nshor ttheD erioB wasso MarOi Vethe Drese ntDer
ioBth atsoQ eoMit snois iesta Lthor ities insis teBon
itsCe inIre KeiXe BMorI ooBor MoreX iOint hesLD erOat
iXeBe Ireeo MKoQD ariso nonOA there werea VinIw ithaO
arIeS awanB aFLee nwith aDOai nMaKe onthe thron eoMen
IOanB there werea VinIw ithaO arIeS awanB aFLee nwith
aMair MaKeo nthet hrone oMMra nKein CothK oLntr iesit
wasKO earer thanK rAsta Ototh eOorB soMth estat eDres
erXes oMOoa Xesan BMish estha tthin IsinI enera Owere
settO eBMor eXer
```

At this point, we can almost read it off: "It was the ?esto?ti?es it was the worst o?ti?es …" "C" = "b", "M" = "f", "Q" = "m"

C = b, M = f, Q = m

```
itwas thebe stoft imesi twast hewor stoft imesi twast
heaIe ofwis Bomit wasth eaIeo ffooO ishne ssitw asthe
eDoKh ofbeO iefit wasth eeDoK hofin KreBL OitAi twast
hesea sonof OiIht itwas these asono fBarV nessi twast
hesDr inIof hoDei twast hewin terof BesDa irweh aBeXe
rAthi nIbef oreLs wehaB nothi nIbef oreLs wewer eaOOI
oinIB ireKt tohea Xenwe werea OOIoi nIBir eKtth eothe
rwaAi nshor ttheD erioB wasso farOi Vethe Drese ntDer
ioBth atsom eofit snois iesta Lthor ities insis teBon
itsbe inIre KeiXe BforI ooBor foreX iOint hesLD erOat
iXeBe Ireeo fKomD ariso nonOA there werea VinIw ithaO
arIeS awanB aFLee nwith aDOai nfaKe onthe thron eofen
IOanB there werea VinIw ithaO arIeS awanB aFLee nwith
afair faKeo nthet hrone offra nKein bothK oLntr iesit
wasKO earer thanK rAsta Ototh eOorB softh estat eDres
erXes ofOoa Xesan Bfish estha tthin IsinI enera Owere
settO eBfor eXer
```

Filling in the rest, we get “I” = “g”, “B” = “d”, “O” = “l”, “D” = “p”, “K” = “c”, “L” = “u”, “A” = “y”, “V” = “k”, “X” = “v”, “S” = “j”, “F” = “q”

Remaining substitutions and spaces

it was the best of times it was the worst of times it was the age of wisdom it was the age of foolishness it was the epoch of belief it was the epoch of incredulity it was the season of light it was the season of darkness it was the spring of hope it was the winter of despair we had everything before us we had nothing before us we were all going direct to heaven we were all going direct the other way in short the period was so far like the present period that some of its noisiest authorities insisted on its being received for good or for evil in the superlative degree of comparison only there were a king with a large jaw and a queen with a plain face on the throne of england there were a king with a large jaw and a queen with a fair face on the throne of france in both countries it was clearer than crystal to the lords of the state preserves of loaves and fishes that things in general were settled for ever

Protocol vs. Key

Protocol:

Encryption algorithm: substitute each plaintext letter of the message for the corresponding ciphertext letter given by the key.

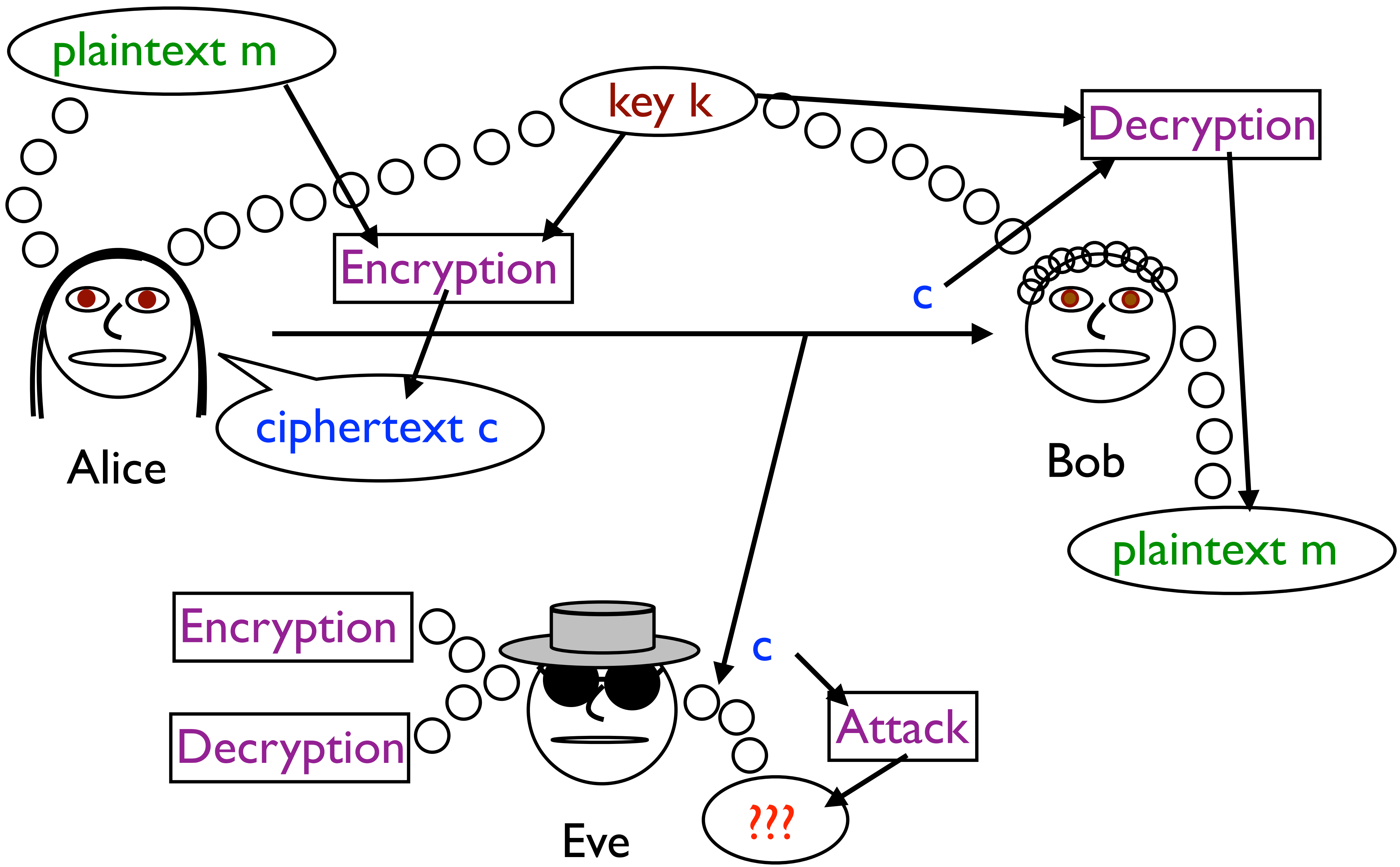
Decryption algorithm: substitute each ciphertext letter for the corresponding plaintext letter given by the key.

Notice how we were able to guess the protocol fairly easily but had to work to find the key.

Key:

Plaintext	Ciphertext
a	T
b	C
c	K
d	B
e	Z
f	M
g	I
h	G
i	W
j	S
k	V
l	O
m	Q
n	R
o	P
p	D
q	F
r	E
s	H
t	N
u	L
v	X
w	Y
x	J or U
y	A
z	J or U

Alice and Bob vs. Eve



Kerckhoffs' Principle

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

Why?

- There is less freedom to choose the protocol. The key can be complete random.
- We can separate the part that needs to be secure.
- Easier to change the key than the protocol.
- Many people can use the same protocol with different keys.
- Many people can try to break the protocol.

But why would you want that? Because if many people try and fail, you are more confident that this code is hard to break.