Easy Crypto

Getting in touch with cryptography

Michael Schwarz

September 13, 2016

LosFuzzy's Training Session

Table of contents

1. Introduction
2. Caesar
3. Monoalphabetic substitution cipher

Introduction

Overview

We will look at some entry-level cryptography that is often used in hacklets to disguise flags.

Caesar

Sample text

ILP CRWWV ZOVMQL QOXFKFKD

Sample text

ILP CRWWV ZOVMQL QOXFKFKD

- Classic Caesar cipher

Sample text

ILP CRWWV ZOVMQL QOXFKFKD

- Classic Caesar cipher
- Every letter is shifted

Sample text

ILP CRWWV ZOVMQL QOXFKFKD

- Classic Caesar cipher
- Every letter is shifted
- If 26 letters are used: only 26 possibilities

Sample text

ILP CRWWV ZOVMQL QOXFKFKD
JMQ DSXXW APWNRM RPYGLGLE

- Classic Caesar cipher
- Every letter is shifted
- If 26 letters are used: only 26 possibilities
- Brute force until we have meaningful text

Sample text

ILP CRWWV ZOVMQL QOXFKFKD
JMQ DSXXW APWNRM RPYGLGLE
KNR ETYYX BQXOSN SQZHMHMF

- Classic Caesar cipher
- Every letter is shifted
- If 26 letters are used: only 26 possibilities
- Brute force until we have meaningful text

Sample text

ILP CRWWV ZOVMQL QOXFKFKD
JMQ DSXXW APWNRM RPYGLGLE
KNR ETYYX BQXOSN SQZHMHMF
LOS FUZZY CRYPTO TRAINING

- Classic Caesar cipher
- Every letter is shifted
- If 26 letters are used: only 26 possibilities
- Brute force until we have meaningful text

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*
- Simplest case: monoalphabetic substitution cipher

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*
- Simplest case: monoalphabetic substitution cipher
    - One *symbol* is one letter

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*
- Simplest case: monoalphabetic substitution cipher
  - One *symbol* is one letter
  - Each letter is replaced by a different letter

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*
- Simplest case: monoalphabetic substitution cipher
  - One *symbol* is one letter
  - Each letter is replaced by a different letter
  - Caesar cipher is a "systematic" monoalphabetic substitution cipher

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*
- Simplest case: monoalphabetic substitution cipher
    - One *symbol* is one letter
    - Each letter is replaced by a different letter
    - Caesar cipher is a “systematic” monoalphabetic substitution cipher
- Higher complexity: polyalphabetic substitution cipher

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*
- Simplest case: monoalphabetic substitution cipher
  - One *symbol* is one letter
  - Each letter is replaced by a different letter
  - Caesar cipher is a “systematic” monoalphabetic substitution cipher
- Higher complexity: polyalphabetic substitution cipher
  - One *symbol* is made of multiple letters

Substitution Cipher

Substitution Cipher

- Replace *symbols* in plaintext with different *symbols*
- Simplest case: monoalphabetic substitution cipher
  - One *symbol* is one letter
  - Each letter is replaced by a different letter
  - Caesar cipher is a "systematic" monoalphabetic substitution cipher
- Higher complexity: polyalphabetic substitution cipher
  - One *symbol* is made of multiple letters
  - Vigenere cipher is a "systematic" polyalphabetic substitution cipher

Monoalphabetic substitution cipher

Caesar

AX qgm lZafc UjqhlgYjShZq ak lZW SfkoWj lg qgmj hjgTdWe, lZWf qgm Vgf2l cfgo oZSl qgmj hjgTdWe ak. FWmeSff, usst

Caesar

AX qgm lZafc UjqhlgYjShZq ak lZW SfkoWj lg qgmj hjgTdWe, lZWf qgm Vgf2l cfgo oZSl qgmj hjgTdWe ak. FWmeSff, usst

- More fun if the alphabet size is unknown

Caesar

AX qgm lZafc UjqhlgYjShZq ak lZW SfkoWj lg qgmj hjgTdWe, lZWf qgm Vgf2l cfgo oZSl qgmj hjgTdWe ak. FWmeSff, usst

- More fun if the alphabet size is unknown
- Still easy to brute force

Caesar brute force - Code

```python
upper = "".join([chr(ascii) for ascii in range(65,91)])
lower = "".join([chr(ascii) for ascii in range(97,123)])
digit = "".join([chr(ascii) for ascii in range(48,58)])

# select the alphabet: uppercase, lowercase, digits and apostrophe
ALPHABET = upper + lower + digit + "'"
# try all possible keys
KEY = range(len(ALPHABET))
# message to decode
MSG = "AX qgm lZafc UjqhlgYjShZq ak lZW SfkoWj lg qgmj" +
    "hjgTdWe, lZWf qgm Vgf2l cfgo oZSl qgmj hjgTdWe ak. FWmeSff, usst"

for k in KEY:
    out = ""
    for c in MSG:
        try:
            out += ALPHABET[(ALPHABET.index(c) + k) % len(ALPHABET)]
        except:
            out += c
    print("%d: %s" % (k, out))
```

Caesar brute force

```
0: AX qgm lZafc UjqhlgYjShZq ak lZW SfkoWj lg qgmj hjgTdWe, lZWf qgm Vgf2l cfgo oZSl qgmj hjgTdWe ak. FWmeSff, usst
1: BY rhn mabgd VkrimhZkTiar bl maX TglpXk mh rhnk ikhUeXf, maXg rhn Whg3m dghp paTm rhnk ikhUeXf bl. GXnfTgg, vttu
2: CZ sio nbche WlsjnialUjbs cm nbY UhmqYl ni siol jliVfYg, nbYh sio Xih4n ehiq qbUn siol jliVfYg cm. HYogUhh, wuuv
3: Da tjp ocdif XmtkojbmVkct dn ocZ VinrZm oj tjpm kmjWgZh, ocZi tjp Yji5o fijr rcVo tjpm kmjWgZh dn. IZphVii, xvvw
4: Eb ukq pdejg YnulpkcnWldu eo pda Wjosan pk ukqn lnkXhai, pdaj ukq Zkj6p gjks sdWp ukqn lnkXhai eo. JaqiWjj, ywwx
5: Fc vlr qefkh ZovmqldoXmev fp qeb Xkptbo ql vlro molYibj, qebk vlr alk7q hklt teXq vlro molYibj fp. KbrjXkk, zxxy
6: Gd wms rfgli apwnrmepYnfw gq rfc Ylqucp rm wmsp npmZjck, rfcl wms bml8r ilmu ufYr wmsp npmZjck gq. LcskYll, 0yyz
7: He xnt sghmj bqxosnfqZogx hr sgd Zmrvdq sn xntq oqnakdl, sgdm xnt cnm9s jmnv vgZs xntq oqnakdl hr. MdtlZmm, 1zz0
8: If you think cryptography is the answer to your problem, then you don’t know what your problem is. Neumann, 2001
9: Jg zpv uijol dszquphsbqiz jt uif botxfs up zpvs qspcmfn, uifo zpv epoAu lopx xibu zpvs qspcmfn jt. Ofvnboo, 3112
10: Kh 0qw vjkpm et0rvqitcrj0 ku vjg cpuygt vq 0qwt rtqdngo, vjgp 0qw fqpBv mpqy yjcv 0qwt rtqdngo ku. Pgwocpp, 4223
11: Li 1rx wklqn fu1swrjudsk1 lv wkh dqvzhu wr 1rxu sureohp, wkhq 1rx grqCw nqrz zkdw 1rxu sureohp lv. Qhxpdqq, 5334
12: Mj 2sy xlmro gv2txskvetl2 mw xli erw0iv xs 2syv tvsfpiq, xlir 2sy hsrDx ors0 0lex 2syv tvsfpiq mw. Riyqerr, 6445
13: Nk 3tz ymnsp hw3uytlwfum3 nx ymj fsx1jw yt 3tzw uwtgqjr, ymjs 3tz itsEy pst1 1mfy 3tzw uwtgqjr nx. Sjzrfss, 7556
14: Ol 4u0 znotq ix4vzumxgvn4 oy znk gty2kx zu 4u0x vxuhrks, znkt 4u0 jutFz qtu2 2ngz 4u0x vxuhrks oy. Tk0sgtt, 8667
15: Pm 5v1 0opur jy5w0vnyhwo5 pz 0ol huz3ly 0v 5v1y wyvislt, 0olu 5v1 kvuG0 ruv3 3oh0 5v1y wyvislt pz. Ul1thuu, 9778
16: Qn 6w2 1pqvs kz6x1wozixp6 q0 1pm iv04mz 1w 6w2z xzwjtmu, 1pmv 6w2 lwvH1 svw4 4pi1 6w2z xzwjtmu q0. Vm2uivv, ’889
17: Ro 7x3 2qrwt l07y2xp0jyq7 r1 2qn jw15n0 2x 7x30 y0xkunv, 2qnw 7x3 mxwI2 twx5 5qj2 7x30 y0xkunv r1. Wn3vjww, A99’
18: Sp 8y4 3rsxu m18z3yq1kzr8 s2 3ro kx26o1 3y 8y41 z1ylvow, 3rox 8y4 nyxJ3 uxy6 6rk3 8y41 z1ylvow s2. Xo4wkxx, B’’A
19: Tq 9z5 4styv n2904zr2l0s9 t3 4sp ly37p2 4z 9z52 02zmwpx, 4spy 9z5 ozyK4 vyz7 7sl4 9z52 02zmwpx t3. Yp5xlyy, CAAB
20: Ur ’06 5tuzw o3’150s3m1t’ u4 5tq mz48q3 50 ’063 130nxqy, 5tqz ’06 p0zL5 wz08 8tm5 ’063 130nxqy u4. Zq6ymzz, DBBC
```

Complexer cipher

Substitution cipher

WSie ueScADSGf
j bHcf heSifSD i hNNB YeScSGfifANG ONe fsS zNc
mHIIkc. uBSicS OAGD Af iffihsSD fN fsAc StiAB. j
SGhekYfSD Af EAfs iSc HcAGR i 128-KAf qSk
RSGSeifSD HcAGR fsS eiGD() OHGhfANG NO fsS h
cfiGDieD BAKeiek. PNfS fsif j hNtYABSD fsS qSk
fNDik if 12:00it.
jG iDDAfANG fN fsAc YeScSGfifANG, j iDDSD iG
iefAhBS iKNHf iG AGfSeScfAGR hekYfN-iBRNeAfst.
LNESQSe, AG NeDSe fN eSiD fsS iefAhBS (Af Ac
iYYeNwAtifSBk 3500 hsieihfSec BNGR), kNH GSSD fN
eSQSecS Esif j DAD fN sADS Af’c hNGfSGf OeNt kNH.
j RAQS kNH i sAGf: ZNH’BB GSSD fsS sSBY NO i
OitNHc eNtiG StYSeNe.
lsSSec, xBAhS

Substitution cipher - Most frequent letter S to e

Weie ueecADeGf

j bHcf heeifeD i hNNB YeeceGfifANG ONe fse zNc mHIIkc. uBeice OAGD Af iffihseD fN fsAc etiAB. j eGhekYfeD Af EAfs iec HcAGR i 128-KAf qek ReGeeifeD HcAGR fse eiGD() OHGhfANG NO fse h cfiGDieD BAKeiek. PNfe fsif j hNtYABeD fse qek fNDik if 12:00it.

jG iDDAfANG fN fsAc YeeceGfifANG, j iDDeD iG iefAhBe iKNHf iG AGfeeecfAGR hekYfN-iBRNeAfst. LNEeQee, AG NeDee fN eeiD fse iefAhBe (Af Ac iYYeNwAtifeBk 3500 hsieihfeec BNGR), kNH GeeD fN eeQeece Esif j DAD fN sADe Af’c hNGfeGf OeNt kNH.

j RAQe kNH i sAGf: ZNH’BB GeeD fse seBY NO i OitNHc eNtiG etYeeNe.

lseeec, xBAhe

Substitution cipher - Guess words

Weie ueecADeGf
j bHcf heeifeD i hNNB YeeceGfifANG ONe fse zNc
mHIIkc. uBeice OAGD Af iffihseD fN fsAc etiAB. j
eGhekYfeD Af EAfs iec HcAGR i 128-KAf qek
ReGeeifeD HcAGR fse eiGD() OHGhfANG NO fse h
cfiGDieD BAKeiek. PNfe fsif j hNtYABeD fse qek
fNDik if 12:00it.
jG iDDAfANG fN fsAc YeeceGfifANG, j iDDeD iG
iefAhBe iKNHf iG AGfeeecfAGR hekYfN-iBRNeAfst.
LNEeQee, AG NeDee fN eeiD fse iefAhBe (Af Ac
iYYeNwAtifeBk 3500 hsieihfeec BNGR), kNH GeeD fN
eeQeece Esif j DAD fN sADe Af’c hNGfeGf OeNt kNH.
j RAQe kNH i sAGf: ZNH’BB GeeD fse seBY NO i
OitNHc eNtiG etYeeNe.
lseeec, xBAhe

Substitution cipher - Guess words

```
Weir uresADeGf
j bHsf hreifeD i hNNB YreseGfifANG ONr fhe zNs
mHIIks.  uBeise OAGD Af iffihheD fN fhAs etiAB. j
eGhrkYfeD Af EAfh ies HsAGR i 128-KAf qek
ReGerifeD HsAGR fhe riGD() OHGhfANG NO fhe h
sfiGDirD BAKrirk.  PNfe fhif j hNtYABeD fhe qek
fNDik if 12:00it.
jG iDDAfANG fN fhAs YreseGfifANG, j iDDeD iG
irfAhBe iKNHf iG AGferesfAGR hrkYfN-iBRNrAfht.
LNEeQer, AG NrDer fN reiD fhe irfAhBe (Af As
iYYrNwAtifeBk 3500 hhirihfers BNGR), kNH GeeD fN
reQerse Ehif j DAD fN hADe Af’s hNGfeGf OrNt kNH.
j RAQe kNH i hAGf:  ZNH’BB GeeD fhe heBY NO i
OitNHs rNtiG etYerNr.
cheers, xBAhe
```

Substitution cipher - Guess words

```
Weir uresADeGf
j bHsf hreifeD i hNNB YreseGfifANG ONr fhe zNs
mHIIks.  uBeise OAGD Af iffihheD fN fhAs etiAB. j
eGhrkYfeD Af EAfh ies HsAGR i 128-KAf qek
ReGerifeD HsAGR fhe riGD() OHGhfANG NO fhe h
sfiGDirD BAKrirk.  PNfe fhif j hNtYABeD fhe qek
fNDik if 12:00it.
jG iDDAfANG fN fhAs YreseGfifANG, j iDDeD iG
irfAhBe iKNHf iG AGferesfAGR hrkYfN-iBRNrAfht.
LNEeQer, AG NrDer fN reiD fhe irfAhBe (Af As
iYYrNwAtifeBk 3500 hhirihfers BNGR), kNH GeeD fN
reQerse Ehif j DAD fN hADe Af's hNGfeGf OrNt kNH.
j RAQe kNH i hAGf:  ZNH'BB GeeD fhe heBY NO i
OitNHs rNtiG etYerNr.
cheers, xBAhe
```

Substitution cipher - Guess words

```
Weir uresADeGt
j bHst hreiteD i hNNB YreseGtitANG ONr the zNs
mHIIks.  uBeise OAGD At ittihheD tN thAs etiAB. j
eGhrkYteD At EAth ies HsAGR i 128-KAt qek
ReGeriteD HsAGR the riGD() OHGhtANG NO the h
stiGDirD BAKrirk.  PNte thit j hNtYABeD the qek
tNDik it 12:00it.
jG iDDAtANG tN thAs YreseGtitANG, j iDDeD iG
irtAhBe iKNHt iG AGterestAGR hrkYtN-iBRNrAtht.
LNEeQer, AG NrDer tN reiD the irtAhBe (At As
iYYrNwAtiteBk 3500 hhirihters BNGR), kNH GeeD tN
reQerse Ehit j DAD tN hADe At’s hNGteGt OrNt kNH.
j RAQe kNH i hAGt:  ZNH’BB GeeD the heBY NO i
OitNHs rNtiG etYerNr.
cheers, xBAhe
```

Substitution cipher - Guess words

```
Weir uresiDeGt
j bHst hreiteD i hNNB YreseGtitiNG ONr the zNs
mHIIks.  uBeise OiGD it ittihheD tN this etiiB. j
eGhrkYteD it Eith ies HsiGR i 128-Kit qek
ReGeriteD HsiGR the riGD() OHGhtiNG NO the h
stiGDirD BiKrirk.  PNte thit j hNtYiBeD the qek
tNDik it 12:00it.
jG iDDitiNG tN this YreseGtitiNG, j iDDeD iG
irtihBe iKNHt iG iGterestiGR hrkYtN-iBRNritht.
LNEeQer, iG NrDer tN reiD the irtihBe (it is
iYYrNwititeBk 3500 hhirihters BNGR), kNH GeeD tN
reQerse Ehit j DiD tN hiDe it’s hNGteGt OrNt kNH.
j RiQe kNH i hiGt:  ZNH’BB GeeD the heBY NO i
OitNHs rNtiG etYerNr.
cheers, xBihe
```

Substitution cipher - Guess words

```
Weir uresiDeGt
j bHst hreiteD i hNNB YreseGtitiNG ONr the zNs
mHIIks.  uBeise OiGD it ittihheD tN this etiiB. j
eGhrkYteD it Eith ies HsiGR i 128-Kit qek
ReGeriteD HsiGR the riGD() OHGhtiNG NO the h
stiGDirD BiKrirk.  PNte thit j hNtYiBeD the qek
tNDik it 12:00it.
jG iDDitiNG tN this YreseGtitiNG, j iDDeD iG
irtihBe iKNHt iG iGterestiGR hrkYtN-iBRNritht.
LNEeQer, iG NrDer tN reiD the irtihBe (it is
iYYrNwititeBk 3500 hhirihters BNGR), kNH GeeD tN
reQerse Ehit j DiD tN hiDe it's hNGteGt OrNt kNH.
j RiQe kNH i hiGt:  ZNH'BB GeeD the heBY NO i
OitNHs rNtiG etYerNr.
cheers, xBihe
```

Substitution cipher - Guess words

```
Weir uresiDent
j bHst hreiteD i hNNB YresentitiNn ONr the zNs
mHIIks.  uBeise OinD it ittihheD tN this etiiB. j
enhrkYteD it Eith ies Hsing i 128-Kit qek
generiteD Hsing the rinD() OHnhtiNn NO the h
stinDirD BiKrirk.  PNte thit j hNtYiBeD the qek
tNDik it 12:00it.
jn iDDitiNn tN this YresentitiNn, j iDDeD in
irtihBe iKNHt in interesting hrkYtN-iBgNritht.
LNEeQer, in NrDer tN reiD the irtihBe (it is
iYYrNwititeBk 3500 hhirihters BNng), kNH neeD tN
reQerse Ehit j DiD tN hiDe it's hNntent OrNt kNH.
j giQe kNH i hint:  ZNH'BB neeD the heBY NO i
OitNHs rNtin etYerNr.
cheers, xBihe
```

Substitution cipher - Guess words

```
Weir uresiDent
j bHst hreiteD i hNNB YresentitiNn ONr the zNs
mHIIks.  uBeise OinD it ittihheD tN this etiiB. j
enhrkYteD it Eith ies Hsing i 128-Kit qek
generiteD Hsing the rinD() OHnhtiNn NO the h
stinDirD BiKrirk.  PNte thit j hNtYiBeD the qek
tNDik it 12:00it.
jn iDDitiNn tN this YresentitiNn, j iDDeD in
irtihBe iKNHt in interesting hrkYtN-iBgNritht.
LNEeQer, in NrDer tN reiD the irtihBe (it is
iYYrNwititeBk 3500 hhirihters BNng), kNH neeD tN
reQerse Ehit j DiD tN hiDe it’s hNntent OrNt kNH.
j giQe kNH i hint:  ZNH’BB neeD the heBY NO i
OitNHs rNtin etYerNr.
cheers, xBihe
```

Substitution cipher - Guess words

Wear uresiDent

j bHst hreateD a hooB presentation Oor the zos mHIIks. uBease OinD it attahheD to this etaiB. j enhrkpteD it Eith aes Hsing a 128-Kit qek generateD Hsing the ranD() OHnhtion oO the h stanDarD BiKrark. Pote that j hotpiBeD the qek toDak at 12:00at.

jn aDDition to this presentation, j aDDeD an artihBe aKoHt an interesting hrkpto-aBgoritht. LoEeQer, in orDer to reaD the artihBe (it is approwitateBk 3500 hharahters Bong), koH neeD to reQerse Ehat j DiD to hiDe it’s hontent Orot koH. j giQe koH a hint: ZoH’BB neeD the heBp oO a OatoHs rotan etperor.

cheers, xBihe

Substitution cipher - Guess words

```
Wear uresiDent
j bHst hreateD a hooB presentation Oor the zos
mHIIks.  uBease OinD it attahheD to this etaiB. j
enhrkpteD it Eith aes Hsing a 128-Kit qek
generateD Hsing the ranD() OHnhtion oO the h
stanDarD BiKrark.  Pote that j hotpiBeD the qek
toDak at 12:00at.
jn aDDition to this presentation, j aDDeD an
artihBe aKoHt an interesting hrkpto-aBgoritht.
LoEeQer, in orDer to reaD the artihBe (it is
approwitateBk 3500 hharahters Bong), koH neeD to
reQerse Ehat j DiD to hiDe it’s hontent Orot koH.
j giQe koH a hint:  ZoH’BB neeD the heBp oO a
OatoHs rotan etperor.
cheers, xBihe
```

Substitution cipher - Guess words

Wear uresident

j bHst hreated a hooB presentation Oor the zos mHIIks. uBease Oind it attahhed to this etaiB. j enhrkpted it Eith aes Hsing a 128-Kit qek generated Hsing the rand() OHnhtion oO the h standard BiKrark. Pote that j hotpiBed the qek todak at 12:00at.

jn addition to this presentation, j added an artihBe aKoHt an interesting hrkpto-aBgoritht. LoEeQer, in order to read the artihBe (it is approwitateBk 3500 hharahters Bong), koH need to reQerse Ehat j did to hide it’s hontent Orot koH.

j giQe koH a hint: ZoH’BB need the heBp oO a OatoHs rotan etperor.

cheers, xBihe

Substitution cipher - Guess words

Wear uresident

j bHst hreated a hooB presentation Oor the zos mHIIks. uBease Oind it attahhed to this etaiB. j enhrkpted it Eith aes Hsing a 128-Kit qek generated Hsing the rand() OHnhtion oO the h standard BiKrark. Pote that j hotpiBed the qek todak at 12:00at.

jn addition to this presentation, j added an artihBe aKoHt an interesting hrkpto-aBgoritht. LoEeQer, in order to read the artihBe (it is approwitateBk 3500 hharahters Bong), koH need to reQerse Ehat j did to hide it’s hontent Orot koH. j giQe koH a hint: ZoH’BB need the heBp oO a OatoHs rotan etperor.

cheers, xBihe

Substitution cipher - Guess words

Dear uresident

I bust created a cool presentation for the zos muIIys. ulease find it attached to this email. I encrypted it with aes using a 128-Kit qey generated using the rand() function of the c standard liKrary. Note that I compiled the qey today at 12:00am.

In addition to this presentation, I added an article aKout an interesting crypto-algorithm. LoweQer, in order to read the article (it is approwimately 3500 characters long), you need to reQerse what I did to hide it’s content from you. I giQe you a hint: Zou’ll need the help of a famous roman emperor.

cheers, xlice

Substitution cipher - Guess words

Dear President

I just created a cool presentation for the Los Fuzzys. Please find it attached to this email. I encrypted it with aes using a 128-bit key generated using the rand() function of the c standard library. Note that I compiled the key today at 12:00am.

In addition to this presentation, I added an article about an interesting crypto-algorithm. However, in order to read the article (it is approximately 3500 characters long), you need to reverse what I did to hide it’s content from you.

I give you a hint: You’ll need the help of a famous roman emperor.

Cheers, Alice

TOO EASY
WHAT'S NEXT?
memegenerator.net

Vigenère

Vigenère

Vigenère

Vigenère

Vigenère

- Polyalphabetic substitution cipher

Vigenère

Vigenère

- Polyalphabetic substitution cipher
- Composed of multiple Caesar ciphers

Vigenère

Vigenère

- Polyalphabetic substitution cipher
- Composed of multiple Caesar ciphers
- Uses a keyword to select the Caesar shift

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

V__ ______

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VS_ ______

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VSQ ______

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VSQ P_____

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VSQ PY____

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VSQ PYX___

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VSQ PYXJ__

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VSQ PYXJC_

LOS FUZZYS

K
E
Y

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

VSQ PYXJCQ

How to break

- Finding the key length using the Kasiski examination:

How to break

- Finding the key length using the Kasiski examination:
    - Find same words

How to break

- Finding the key length using the Kasiski examination:
    - Find same words
    - Note the difference in their text position

How to break

- Finding the key length using the Kasiski examination:
  - Find same words
  - Note the difference in their text position
  - Factor all differences

How to break

- Finding the key length using the Kasiski examination:
    - Find same words
    - Note the difference in their text position
    - Factor all differences
    - The factor that occurs most often is probably the key length

Kasiski examination

```
myx iyhe,
h gcflc xmz qzmr yi ings or zli gzjc xild kthdx.  rt gzjc xoqd wto zqc
xohsygfd emw ngd htv vd ufhs sm ryds wto.  cn wto jmmb ngd ajhsqyq jzqi
xnzsgth?  zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug?  okcfmd lypy
rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or.
vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld.  emqfnv sx
omnzyltrgaykx!
yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj
cmemwgzsgth:  ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm.  rmy
okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr
yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc.  uj uqd etcmf
rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx.  nmaj qd zpj
cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs.
hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp
xsrscr.  nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql.  rt vqhll ngd
kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.
qd bys xhrazmr dtjlxsfnhf djxy nm kthczw!  ho, anz
```

Kasiski examination

myx iyhe,

h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc
xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi
xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy
rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or.
vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx
omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj
cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy
okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr
yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf
rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj
cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs.
hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp
xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd
kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Kasiski examination

myx iyhe,

h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or. vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs. hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Kasiski examination

myx iyhe,

h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or. vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs. hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Kasiski examination

myx iyhe,

h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or. vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs. hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Kasiski examination

myx iyhe,

h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or. vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs. hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Kasiski examination

myx iyhe,

h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or. vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs. hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Kasiski examination

myx iyhe,

h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or. vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs. hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Kasiski examination

myx iyhe,

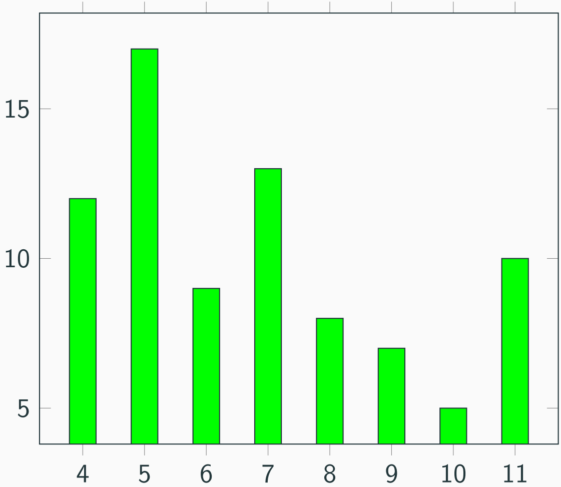
h gcflc xmz qzmr yi ings or zli gzjc xild kthdx. rt gzjc xoqd wto zqc
xohsygfd emw ngd htv vd ufhs sm ryds wto. cn wto jmmb ngd ajhsqyq jzqi
xnzsgth? zqc dit ztfckzzqy nm kthczw scmdrjymsf, xyudl ug? okcfmd lypy
rtpj snt bth’s cpfq zsrjhshms nn xmzlrdjk qghjj snt’pj qzhrnhf emw or.
vynn nm y gymbf fn okyyznqk xyudl, by vhjq zhmb dit sfjld. emqfnv sx
omnzyltrgaykx!

yx nghq jgzhj nm rspthfkw jhbqwundc g hum optphcc dit vgyb lnpj
cmemwgzsgth: ntp yuqfcy qhkj gy sgc sushmsuk ayse ne ufmghllnnm. rmy
okys cr sm fnszap xtqgsa ktlhb shkj vdbyzmd lyss dlnqixdcx qhkj gy ntr
yujhll ngdgw ftmam cm sfj ldrrfoqzlym hm rmy mdglbanpminc. uj uqd etcmf
rt ord rmy azap ymspfhbd ysx adffpd kgpy rtnufhdpx ie emtx. nmaj qd zpj
cm sfj vthjicmf uj qhkj zmd nsw wnucw nn fcy wknqj nn sfj wdmrwuk uyzfs.

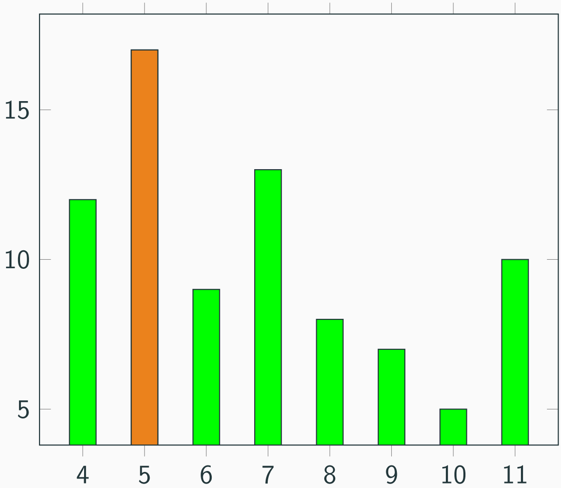
hr nm oqmyybsci vx rcayqzj lozqbx umc y mcfgjd mnofnmshafndc jfmsdp
xsrscr. nn jltwj nsy ngd ezuqcq by vhjq ord amfnqmkiql. rt vqhll ngd
kthdx mzn ne rmy azlp qd vgqf trc f znnb ylnkjd.

qd bys xhrazmr dtjlxsfnhf djxy nm kthczw! ho, anz

Finding the key length



Finding the key length



Continue breaking the cipher

- We (hopefully) know the key length

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers
- In this case: key length of 5 means we have 5 Caesar ciphers

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers
- In this case: key length of 5 means we have 5 Caesar ciphers
- 1., 6., 11., ... letter is encrypted using a Caesar ciphers

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers
- In this case: key length of 5 means we have 5 Caesar ciphers
- 1., 6., 11., ... letter is encrypted using a Caesar ciphers
- 2., 7., 12., ... letter is encrypted using a Caesar ciphers

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers
- In this case: key length of 5 means we have 5 Caesar ciphers
- 1., 6., 11., ... letter is encrypted using a Caesar ciphers
- 2., 7., 12., ... letter is encrypted using a Caesar ciphers
- 3., 8., 13., ... letter is encrypted using a Caesar ciphers

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers
- In this case: key length of 5 means we have 5 Caesar ciphers
- 1., 6., 11., ... letter is encrypted using a Caesar ciphers
- 2., 7., 12., ... letter is encrypted using a Caesar ciphers
- 3., 8., 13., ... letter is encrypted using a Caesar ciphers
- 4., 9., 14., ... letter is encrypted using a Caesar ciphers

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers
- In this case: key length of 5 means we have 5 Caesar ciphers
- 1., 6., 11., ... letter is encrypted using a Caesar ciphers
- 2., 7., 12., ... letter is encrypted using a Caesar ciphers
- 3., 8., 13., ... letter is encrypted using a Caesar ciphers
- 4., 9., 14., ... letter is encrypted using a Caesar ciphers
- 5., 10., 15., ... letter is encrypted using a Caesar ciphers

Continue breaking the cipher

- We (hopefully) know the key length
- We can disassemble the cipher text into multiple Caesar ciphers
- In this case: key length of 5 means we have 5 Caesar ciphers
- 1., 6., 11., ... letter is encrypted using a Caesar ciphers
- 2., 7., 12., ... letter is encrypted using a Caesar ciphers
- 3., 8., 13., ... letter is encrypted using a Caesar ciphers
- 4., 9., 14., ... letter is encrypted using a Caesar ciphers
- 5., 10., 15., ... letter is encrypted using a Caesar ciphers
- Use frequency analysis on each Caesar cipher

Frequency analysis of first Caesar (1., 6., 11., ...)

- Most frequent letter: **J**

Frequency analysis of first Caesar (1., 6., 11., ...)

- Most frequent letter: **J**
- Guess that **J** equals **E** in original text

Frequency analysis of first Caesar (1., 6., 11., ...)

- Most frequent letter: **J**
- Guess that **J** equals **E** in original text
- Looking at column **E** until we find **J** gives first letter of key: **F**

Frequency analysis of first Caesar (2., 7., 12., ...)

- Most frequent letter: **Y**

Frequency analysis of first Caesar (2., 7., 12., ...)

- Most frequent letter: **Y**
- Guess that **Y** equals **E** in original text

Frequency analysis of first Caesar (2., 7., 12., ...)

- Most frequent letter: **Y**
- Guess that **Y** equals **E** in original text
- Looking at column **E** until we find **Y** gives second letter of key: **U**

Frequency analysis of first Caesar (3., 8., 13., ...)

- Most frequent letter: **N**

Frequency analysis of first Caesar (3., 8., 13., ...)

- Most frequent letter: **N**
- Guess that **N** equals **E** in original text

Frequency analysis of first Caesar (3., 8., 13., ...)

- Most frequent letter: **N**
- Guess that **N** equals **E** in original text
- Looking at column **E** until we find **N** gives third letter of key: **J**

Frequency analysis of first Caesar (4., 9., 14., ...)

- Most frequent letter: **D**

Frequency analysis of first Caesar (4., 9., 14., ...)

- Most frequent letter: **D**
- Guess that **D** equals **E** in original text

Frequency analysis of first Caesar (4., 9., 14., ...)

- Most frequent letter: **D**
- Guess that **D** equals **E** in original text
- Looking at column **E** until we find **D** gives fourth letter of key: **Z**

Frequency analysis of first Caesar (5., 10., 15., ...)

- Most frequent letters: **C** and **Y**

Frequency analysis of first Caesar (5., 10., 15., ...)

- Most frequent letters: **C** and **Y**
- Guess that **C** or **Y** equals **E** in original text

Frequency analysis of first Caesar (5., 10., 15., ...)

- Most frequent letters: **C** and **Y**
- Guess that **C** or **Y** equals **E** in original text
- Looking at column **E** until we find **C** or **Y** gives fifth letter of key: **Y** or **U**

Decrypting with key FUJZY

```
heo jack,
y heart you wqnt to zoin ui and mqke soce monuy.  to mqke suhe you qre
suytablu for txe job me wanj to meut you.  to you anow txe cenjral pqrk
stqtion?  qre yok avaibable en montay nideteedth, selen pm?  fleasu make
iure yeu don’j draw qttenjion te yourielf wxile yeu’re wqitinw for ui.
wait en a bedch at flatferm selen, we mill fynd yok theru.  follew us
udobtrksiveby!
as txis emqil is itronwly ensryptud i cad provyde yok with core
idformqtion:  eur tahget wyll be jhe najionab bank ef wasxingten.  the
flan ii to atjack dkring bunch jime bucausu many umplooees wyll be eut
taaing txeir lknch id the rustauhants yn the deighrorhoed.  we ahe goidg
to uie the rack edtranse and rehavu like iupplyers ov food.  ence wu are
id the bkildidg we wyll usu our cever te get cbose te the cuntrab vaulj.
it is frotested bo sevehal guqrds add a hiwhly sephisjicatud lasjer
syitem.  te knoca out txe guahds we mill uie chlerofohm.  to bhing txe
monuy out ef the rank wu will kse a feod trelly.
wu can dyscusi everothinw else en montay!  cu, rob
```

Fixing the key

- Looks already kind of readable

Fixing the key

- Looks already kind of readable
- 3. letter seems to be wrong

Fixing the key

- Looks already kind of readable
- 3. letter seems to be wrong
- Using a known word (e.g. wqnt → want) we can calculate the shift

Fixing the key

- Looks already kind of readable
- 3. letter seems to be wrong
- Using a known word (e.g. wqnt → want) we can calculate the shift
- With the difference between **Q** and **A** we can correct the key

Fixing the key

- Looks already kind of readable
- 3. letter seems to be wrong
- Using a known word (e.g. wqnt → want) we can calculate the shift
- With the difference between **Q** and **A** we can correct the key
- **J** becomes **Z**

Fixing the key

- Looks already kind of readable
- 3. letter seems to be wrong
- Using a known word (e.g. wqnt → want) we can calculate the shift
- With the difference between **Q** and **A** we can correct the key
- **J** becomes **Z**
- The final key: **FUZZY**

Decrypted text

```
hey jack,
i heard you want to join us and make some money.  to make sure you are
suitable for the job we want to meet you.  do you know the central park
station?  are you available on monday nineteenth, seven pm?  please make
sure you don’t draw attention to yourself while you’re waiting for us.
wait on a bench at platform seven, we will find you there.  follow us
unobtrusively!
as this email is strongly encrypted i can provide you with more
information:  our target will be the national bank of washington.  the
plan is to attack during lunch time because many employees will be out
taking their lunch in the restaurants in the neighborhood.  we are going
to use the back entrance and behave like suppliers of food.  once we are
in the building we will use our cover to get close to the central vault.
it is protected by several guards and a highly sophisticated laster
system.  to knock out the guards we will use chloroform.  to bring the
money out of the bank we will use a food trolly.
we can discuss everything else on monday!  cu, bob
```

Questions?

References i

Caesar Cipher https://en.wikipedia.org/wiki/Caesar_cipher

Substitution Cipher https://en.wikipedia.org/wiki/Substitution_cipher

Frequency Analysis https://en.wikipedia.org/wiki/Frequency_analysis

Vigenère Cipher https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher

Kasiski examination https://en.wikipedia.org/wiki/Kasiski_examination

ECDSA Hacklet

Breaking a secure signature scheme

Michael Schwarz

September 11, 2016

LosFuzzy's Training Session

Table of contents

1. Introduction
2. The target
3. Elliptic Curves
4. ECDSA
5. Breaking into the server

Introduction

Overview

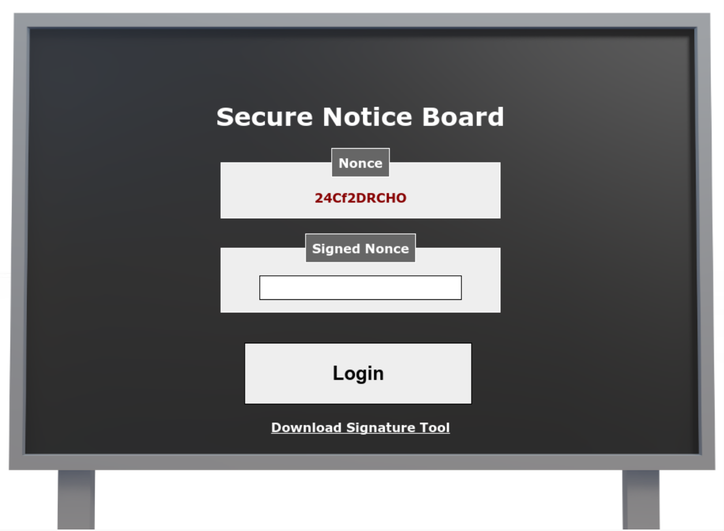
We have to deal with a server storing notes where logins are based on strong elliptic curve cryptography (ECC): http://fuzzys.attacking.systems

We will analyze the used scheme, give an (ultra) short introduction to elliptic curves and then see how to break the scheme.

The target

The Server (i)

http://fuzzys.attacking.systems



Secure Notice Board

Nonce

24Cf2DRCHO

Signed Nonce

The Server (ii)

- The server has a list of users

The Server (ii)

- The server has a list of users
- Each user has a public key on the server

The Server (ii)

- The server has a list of users
- Each user has a public key on the server
- Each user has a note on the server

The Server (ii)

- The server has a list of users
- Each user has a public key on the server
- Each user has a note on the server
- We have an account on the server too

The Server (ii)

- The server has a list of users
- Each user has a public key on the server
- Each user has a note on the server
- We have an account on the server too
  http://fuzzys.attacking.systems/user2_priv.pem

The Scheme

The scheme is based on ECDSA (Ellipctic Curve Digital Signature Algorithm)

The Scheme

The scheme is based on ECDSA (Ellipctic Curve Digital Signature Algorithm)

1. Server displays a random nonce

The Scheme

The scheme is based on ECDSA (Ellipctic Curve Digital Signature Algorithm)

1. Server displays a random nonce
2. User signs the nonce using his private key

The Scheme

The scheme is based on ECDSA (Ellipctic Curve Digital Signature Algorithm)

1. Server displays a random nonce
2. User signs the nonce using his private key
3. User sends signed nonce (signature) to server

The Scheme

The scheme is based on ECDSA (Ellipctic Curve Digital Signature Algorithm)

1. Server displays a random nonce
2. User signs the nonce using his private key
3. User sends signed nonce (signature) to server
4. Server checks which public key verifies the signature

The Scheme

The scheme is based on ECDSA (Ellipctic Curve Digital Signature Algorithm)

1. Server displays a random nonce
2. User signs the nonce using his private key
3. User sends signed nonce (signature) to server
4. Server checks which public key verifies the signature
5. If there is a public key matching the signature, the server displays the notes of the corresponding user

Network capture

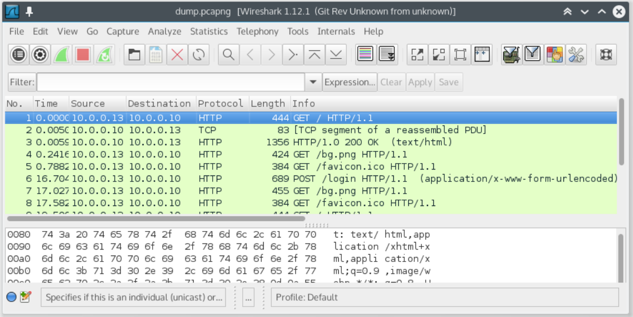
- The server is not using an ecnrypted connection

Network capture

- The server is not using an ecnrypted connection
- We managed to capture the network traffic of two login of our victim

Network capture

- The server is not using an ecnrypted connection
- We managed to capture the network traffic of two login of our victim



dump.pcapng [Wireshark 1.12.1 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply
Save

```
No. Time     Source    Destination Protocol Length Info
  1 0.0000  10.0.0.13 10.0.0.10   HTTP      444 GET / HTTP/1.1
  2 0.0050  10.0.0.10 10.0.0.13   TCP        83 [TCP segment of a reassembled PDU]
  3 0.0059  10.0.0.10 10.0.0.13   HTTP     1356 HTTP/1.0 200 OK  (text/html)
  4 0.2416  10.0.0.13 10.0.0.10   HTTP      424 GET /bg.png HTTP/1.1
  5 0.7882  10.0.0.13 10.0.0.10   HTTP      384 GET /favicon.ico HTTP/1.1
  6 16.704  10.0.0.13 10.0.0.10   HTTP      689 POST /login HTTP/1.1  (application/x-www-form-urlencoded)
  7 17.027  10.0.0.13 10.0.0.10   HTTP      455 GET /bg.png HTTP/1.1
  8 17.582  10.0.0.13 10.0.0.10   HTTP      384 GET /favicon.ico HTTP/1.1
```

```
0080  74 3a 20 74 65 78 74 2f  68 74 6d 6c 2c 61 70 70   t: text/ html,app
0090  6c 69 63 61 74 69 6f 6e  2f 78 68 74 6d 6c 2b 78   lication /xhtml+x
00a0  6d 6c 2c 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   ml,appli cation/x
00b0  6d 6c 3b 71 3d 30 2e 39  2c 69 6d 61 67 65 2f 77   ml;q=0.9 ,image/w
```

http://fuzzys.attacking.systems/dump.pcapng

Elliptic Curves

Elliptic Curves

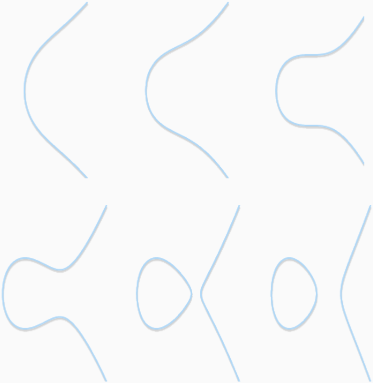
The set of points described by the following equation (Weierstrass normal form):

$$y^2 = x^3 + ax + b$$

Elliptic Curves

The set of points described by the following equation (Weierstrass normal form):

$$y^2 = x^3 + ax + b$$

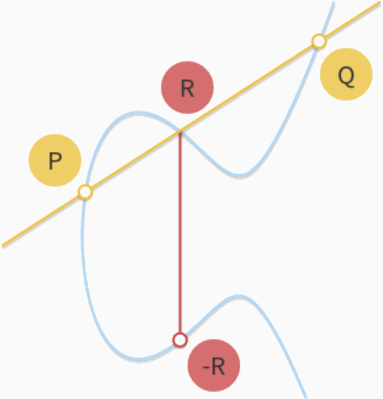


Elliptic Curves - Addition

We can add points two points P and Q on the curve

Elliptic Curves - Addition

We can add points two points P and Q on the curve



Note

This is a simplified representation - we will not cover any (mathematical) details and corner cases as this is not required to understand the signature algorithm.

Elliptic Curves - Double

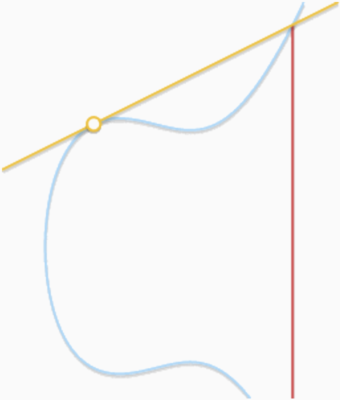
- We can double a point P to get $2P$

Elliptic Curves - Double

- We can double a point P to get $2P$
- It is the special case of the addition, where $P = Q$.

Elliptic Curves - Double

- We can double a point P to get $2P$
- It is the special case of the addition, where $P = Q$.
- The line between the two points is now the tangent to P



Elliptic Curves - Scalar Multiplication ($n \cdot P$)

- We can multiply a point P with a scalar ("normal number") n

Elliptic Curves - Scalar Multiplication ($n \cdot P$)

- We can multiply a point P with a scalar ("normal number") n
- The multiplication is based on addition and doubling (double and add) and is very efficient

Elliptic Curves - Scalar Multiplication ($n \cdot P$)

- We can multiply a point P with a scalar ("normal number") n
- The multiplication is based on addition and doubling (double and add) and is very efficient
- What about the other way round? We know P and $Q = n \cdot P$ and want to calculate n

Elliptic Curves - Scalar Multiplication ($n \cdot P$)

- We can multiply a point P with a scalar ("normal number") n
- The multiplication is based on addition and doubling (double and add) and is very efficient
- What about the other way round? We know P and $Q = n \cdot P$ and want to calculate n
- This is the *discrete logarithm problem (DLP)* for which no efficient algorithm is known

Elliptic Curves - Scalar Multiplication ($n \cdot P$)

- We can multiply a point P with a scalar ("normal number") n
- The multiplication is based on addition and doubling (double and add) and is very efficient
- What about the other way round? We know P and $Q = n \cdot P$ and want to calculate n
- This is the *discrete logarithm problem (DLP)* for which no efficient algorithm is known
- These properties are the base of elliptic curve cryptography

ECDSA

ECDSA

ECDSA

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the Digital Signature Algorithm (DSA). It was accepted in 1999 as an ANSI standard, and was accepted in 2000 as IEEE and NIST standards. [...]

ECDSA

ECDSA

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the Digital Signature Algorithm (DSA). It was accepted in 1999 as an ANSI standard, and was accepted in 2000 as IEEE and NIST standards. [...] no subexponential-time algorithm is known for the elliptic curve discrete logarithm problem. [...]

ECDSA

ECDSA

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the Digital Signature Algorithm (DSA). It was accepted in 1999 as an ANSI standard, and was accepted in 2000 as IEEE and NIST standards. [...] no subexponential-time algorithm is known for the elliptic curve discrete logarithm problem. [...] the strength-per-key-bit is substantially greater in an algorithm that uses elliptic curves.

Source: http://cs.ucsb.edu/ koc/ccs130h/notes/ecdsa-cert.pdf

How it works - Preparation

- Alice wants to send a signed message to Bob

How it works - Preparation

- Alice wants to send a signed message to Bob
- They agree on the curve parameters and a base point G on the curve

How it works - Preparation

- Alice wants to send a signed message to Bob
- They agree on the curve parameters and a base point G on the curve
- Alice creates a random private key integer d_A and a public key curve point $Q_A = d_A \times G$

How it works - Preparation

- Alice wants to send a signed message to Bob
- They agree on the curve parameters and a base point G on the curve
- Alice creates a random private key integer d_A and a public key curve point $Q_A = d_A \times G$
- Alice sends Bob her public key Q_A

How it works - Sign

- Alice wants to sign a message m

How it works - Sign

- Alice wants to sign a message m
- She...
  - calculates $z = hash(m)$

How it works - Sign

- Alice wants to sign a message m
- She...
  - calculates $z = hash(m)$
  - chooses a *random* integer k

How it works - Sign

- Alice wants to sign a message m
- She...
  - calculates $z = hash(m)$
  - chooses a *random* integer k
  - calculates the curve point $(r, y) = k \times G$

How it works - Sign

- Alice wants to sign a message m
- She...
  - calculates $z = hash(m)$
  - chooses a *random* integer k
  - calculates the curve point $(r, y) = k \times G$
  - calculates $s = k^{-1}(z + r\, d_A)$

How it works - Sign

- Alice wants to sign a message m
- She...
  - calculates $z = hash(m)$
  - chooses a *random* integer k
  - calculates the curve point $(r, y) = k \times G$
  - calculates $s = k^{-1}(z + r\, d_A)$
- The signature is (r, s)

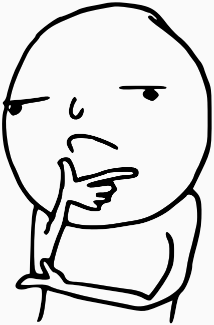
How it works - Sign - Code

```python
import hashlib, random, base64
from ecdsa import SigningKey
from ecdsa.util import string_to_number, sigencode_der

m = "Hi Bob!"

# read private key and curve parameters (d_A, G, ...)
sk = SigningKey.from_pem(open("signkey.pem").read())

# calculate z = hash(m)
z = string_to_number(hashlib.sha1(m).digest())

# random k
k = random.randint(0, sk.privkey.order)

# calculate r, s
r, s = sk.sign_number(z, None, k)

# encode signature
sig = sigencode_der(r, s, sk.privkey.order)
print(base64.b64encode(sig))
```

Breaking into the server

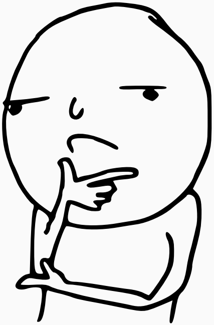
Where is the vulnerability?

- The authentication scheme seems to be secure



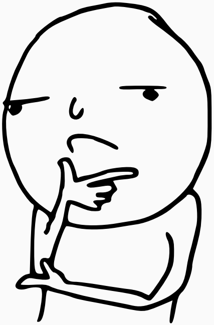
Where is the vulnerability?

- The authentication scheme seems to be secure
- ECDSA is probably secure



Where is the vulnerability?

- The authentication scheme seems to be secure
- ECDSA is probably secure
- This is a crypto session, so maybe there is still something wrong with ECDSA...



ECDSA implementation

- Maybe some implementations do not comply to the standard

ECDSA implementation

- Maybe some implementations do not comply to the standard

Choosing k

[...] it is crucial to select *different* k for *different signatures*, otherwise the equation can be solved for d_A, the private key.

ECDSA implementation

- Maybe some implementations do not comply to the standard

Choosing k

[...] it is crucial to select *different k* for *different signatures*, otherwise the equation can be solved for d_A, the private key.

- But nobody would use the following random number generator, right?

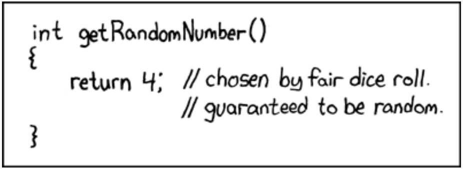
ECDSA implementation

- Maybe some implementations do not comply to the standard

Choosing k

[...] it is crucial to select *different* k for *different signatures*, otherwise the equation can be solved for d_A, the private key.

- But nobody would use the following random number generator, right?



```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

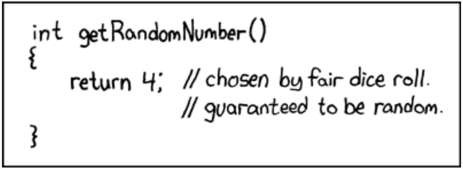
ECDSA implementation

- Maybe some implementations do not comply to the standard

Choosing *k*

[...] it is crucial to select *different k* for *different signatures*, otherwise the equation can be solved for d_A, the private key.

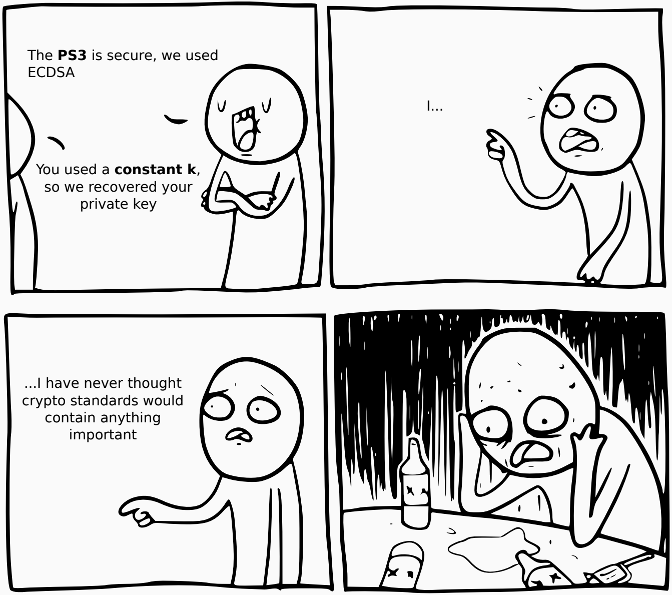
- But nobody would use the following random number generator, right?



```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

Except for...

Sony's ECDSA fail



The **PS3** is secure, we used ECDSA
You used a **constant k**, so we recovered your private key
I...
...I have never thought crypto standards would contain anything important

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.
- We calculate $z = hash(m)$ and $z' = hash(m')$.

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.
- We calculate $z = hash(m)$ and $z' = hash(m')$.
- Remember ECDSA signing: $s = k\ ^{-1}(\ z + r\ d_A\ )$

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.
- We calculate $z = hash(m)$ and $z' = hash(m')$.
- Remember ECDSA signing: $s = k^{-1}(z + r\ d_A)$
- The difference $s - s' = k^{-1}(z + r\ d_A - (z' + r\ d_A))$

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.
- We calculate $z = hash(m)$ and $z' = hash(m')$.
- Remember ECDSA signing: $s = k^{-1}(z + r\ d_A)$
- The difference $s - s' = k^{-1}(z + \cancel{r\ d_A} - (z' + \cancel{r\ d_A}))$

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.
- We calculate $z = hash(m)$ and $z' = hash(m')$.
- Remember ECDSA signing: $s = k^{-1}(z + r\ d_A)$
- The difference $s - s' = k^{-1}(z - z')$

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.
- We calculate $z = hash(m)$ and $z' = hash(m')$.
- Remember ECDSA signing: $s = k\ ^{-1}(\ z + r\ d_A\ )$
- The difference $s - s' = k\ ^{-1}(\ z - z')$
- We get $k = \dfrac{z - z'}{s - s'}$

Recover the private key

- We only need two signatures (r, s) and (r, s') for different messages m and m' and same (unknown) k.
- We calculate $z = hash(m)$ and $z' = hash(m')$.
- Remember ECDSA signing: $s = k^{-1}(z + r\ d_A)$
- The difference $s - s' = k^{-1}(z - z')$
- We get $k = \dfrac{z - z'}{s - s'}$
- And finally the private key $d_A = \dfrac{sk - z}{r}$

Recover the private key - Code (1/2)

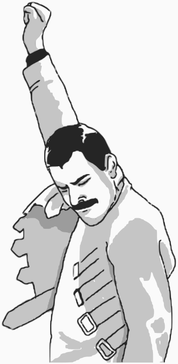
```
from ecdsa import SigningKey
from ecdsa.util import number_to_string, sigdecode_der
import base64, hashlib

# extended euclidean algorithm
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

# we cannot divide - have to multiply with modular inverse
def modinv(a, m):
    g, x, y = egcd(a, m)
    return x % m

# ensure values are not negative and in range [0, n)
def modn(val, n):
    while val <  0: val += n
    return val % n

# read our key for the curve parameters
sk = SigningKey.from_pem(open("user2_priv.pem").read())
```

Recover the private key - Code (2/2)

```
# captured nonces and signed nonces
nonce1 = b'iBW0oioDnq'
nonce2 = b'0CX9nTSLt2'
sign1 = base64.b64decode(
    b'MDUCGQCPnWHAQc9Yiu0i0ZwYT6qe6YBVn0unZM4CGFQc2gBrmO4NInldykH1' +
    b'PUUCPB53WutyBg==')
sign2 = base64.b64decode(
    b'MDUCGQCPnWHAQc9Yiu0i0ZwYT6qe6YBVn0unZM4CGAYzHMi8TrvCJQxtMSmQ' +
    b'/u9+MVnZ+Jf8iQ==')

# get (r,s) and (r,s')
r1, s1 = sigdecode_der(sign1, sk.privkey.order)
r2, s2 = sigdecode_der(sign2, sk.privkey.order)
# calculate z and z'
z1 = int(hashlib.sha1(nonce1).hexdigest(), 16)
z2 = int(hashlib.sha1(nonce2).hexdigest(), 16)
# recover private key
n = sk.privkey.order
k = (modn(z1 - z2, n) * modinv(modn(s1 - s2, n), n)) % n
d = (modn(s1 * k - z1, n) * modinv(r1, n)) % n

# show private key
sk_secret = SigningKey.from_string(number_to_string(d, n))
print(sk_secret.to_pem())
```

Recover the private key

```
-----BEGIN EC PRIVATE KEY-----
MF8CAQEEGKjIYQURchpR7x7DGXSmTgQCv5PAJO+116AKBggqhkjOPQMBAaE0AzIA
BEP05vGvGFRofftZj5zbIteDdqKGlt9KGDvQrxTQAT9X6/G++5h3AMyV3/M1Qv7r
Qg==
-----END EC PRIVATE KEY-----
```



Questions?

How it works - Verify

- Bob wants to verify a signature (r, s) of message m

How it works - Verify

- Bob wants to verify a signature (r, s) of message m
- He...
  - calculates $z = hash(m)$

How it works - Verify

- Bob wants to verify a signature (r, s) of message m
- He...
  - calculates $z = hash(m)$
  - calculates $w = s^{-1}$

How it works - Verify

- Bob wants to verify a signature (r, s) of message m
- He...
  - calculates $z = hash(m)$
  - calculates $w = s^{-1}$
  - calculates $u_1 = zw$ and $u_2 = rw$

How it works - Verify

- Bob wants to verify a signature (r, s) of message m
- He...
  - calculates $z = hash(m)$
  - calculates $w = s^{-1}$
  - calculates $u_1 = zw$ and $u_2 = rw$
  - calculates the curve point $(x_1, y_1) = u_1 \times G + u_2 \times Q_A$

How it works - Verify

- Bob wants to verify a signature (r, s) of message m
- He...
  - calculates $z = hash(m)$
  - calculates $w = s^{-1}$
  - calculates $u_1 = zw$ and $u_2 = rw$
  - calculates the curve point $(x_1, y_1) = u_1 \times G + u_2 \times Q_A$
- The signature is valid if $x_1 \equiv r$

How it works - Verify - Code

```python
from ecdsa import VerifyingKey
from ecdsa.util import sigdecode_der

# read public key and curve parameters (Q_A, G, ...)
vk = VerifyingKey.from_pem(open("verifykey.pem").read())

# sig = signature (DER encoded), m = message
print(vk.verify(sig, m, hashfunc=hashlib.sha1,
                        sigdecode=sigdecode_der))
```

References i

ECC	http://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/
ECDSA	https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm
Python ECDSA	https://github.com/warner/python-ecdsa
Random Number	https://xkcd.com/221/
PS3 Hack	https://events.ccc.de/congress/2010/Fahrplan/attachments/1780_27c3_console_hacking_2010.pdf