Cryptography Engineering

Jerry J. R. Shieh, PhD

March 9 2023 Week 3

Dan Boneh

1

Outline

- Vigener Cipher
- Symmetric Ciphers
- Perfect Secrecy
- Stream Cipher
- Semantic security

Dan Boneh

RECAP

Dan Boneh

3

Polyalphabetic cipher

- A polyalphabetic cipher is any cipher based on **substitution**, using multiple substitution alphabets.
- The Vigener cipher is probably the best-known example of a polyalphabetic cipher, though it is a simplified special case.
- The **Enigma machine and** some **modern electric encryption device are** more complex but is still fundamentally a **polyalphabetic substitution cipher**.

Dan Boneh

4

Vigener cipher

A polyalphabetic cipher

k = C R Y P T O C R Y P T O C R Y P T

(+ mod 26)

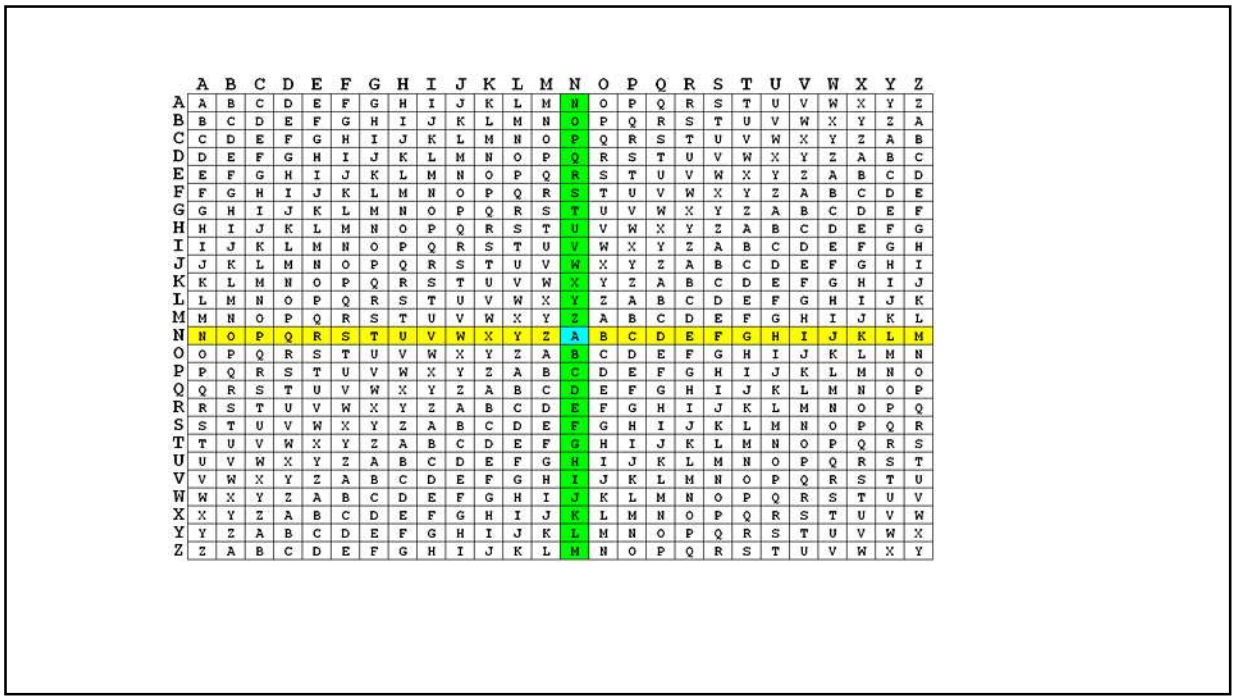
m = W H A T A N I C E D A Y T O D A Y

c = Z Z Z J U C L U D T U N W G C Q S

suppose most common = "H" ⟹ first letter of key = "H" – "E" = "C"

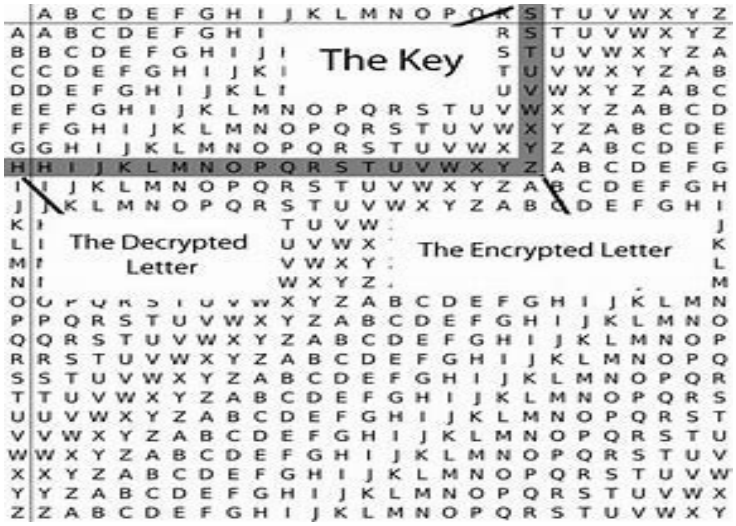
Dan Boneh

5



	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
A	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

6



https://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/Vig-Base.html

Vigener cipher

A polyalphabetic cipher

k = C R Y P T O C R Y P T O C R Y P T

(+ mod 26)

m = W H A T A N I C E D A Y T O D A Y

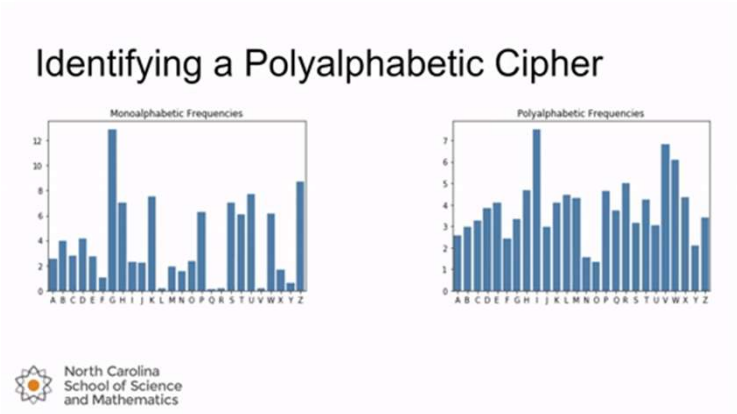
c = Z Z Z J U C L U D T U N W G C Q S

suppose most common = "H" ⟹ first letter of key = "H" – "E" = "C"

Dan Boneh

8

Identify a Polyalphabetic Cipher



Dan Boneh

Index of Coincidence I.C.

Calculating Probabilities

For the monoalphabetic ciphertext, the probability of picking two letters at random from the text and having them both be A's (using the distribution above) would be:

$$M_{\text{both A's}} = \frac{73}{1000} \cdot \frac{72}{999} \approx 0.00526$$

The probability of picking two letters at random from the text and having them both be B's

$$M_{\text{both B's}} = \frac{9}{1000} \cdot \frac{8}{999} \approx 0.0000721$$

and in general, where n_i denotes the number of character i and n_{text} denotes the total number of characters in the text:

$$M_{\text{both i's}} = \frac{n_i}{n_{text}} \cdot \frac{n_i - 1}{n_{text} - 1}$$

For long messages, $\frac{n_i}{n_{text}} \approx \frac{n_i - 1}{n_{text} - 1}$, so the formula can be simplified to:

$$M_i^2 \approx \left(\frac{n_i}{n_{text}}\right)^2$$

The sum of all these probabilities, which is equivalent to asking "What's the probability of picking two letters at random from the text, and having them be the same letter?" works out to be:

$$\sum_{i=A}^{Z} M_i^2 \approx 0.066$$

Doing the same calculation on our very evenly distributed polyalphabetic ciphertext yields:

$$\sum_{i=A}^{Z} P_i^2 \approx 0.038$$

This score, calculated by summing the squares of the letter frequencies is called the **Index of Coincidence**. When presented with an unknown ciphertext, it's index of coincidence will suggest if it was enciphered with a polyalphabetic cipher (if the score is close to 0.038) or a monoalphabetic cipher (if the score is close to 0.066). This gives a quick and easy way to determine with a single number the most likely type of cipher used to create a ciphertext.

Dan Boneh

Identifying a Polyalphabetic Cipher

Monoalphabetic

A	73	J	2	S	63
B	9	K	3	T	93
C	30	L	35	U	130
D	44	M	25	V	13
E	27	N	5	W	16
F	28	O	74	X	78
G	16	P	27	Y	19
H	35	Q	3	Z	1
I	74	R	77	.....	

Polyalphabetic

A	38	J	38	S	39
B	39	K	39	T	38
C	38	L	38	U	39
D	38	M	39	V	39
E	39	N	39	W	38
F	38	O	38	X	38
G	38	P	39	Y	39
H	39	Q	38	Z	38
I	38	R	39	.....	

North Carolina School of Science and Mathematics

Dan Boneh

11

Identifying a Polyalphabetic Cipher

Monoalphabetic

A	73	J	2	S	63
B	9	K	3	T	93
C	30	L	35	U	130
D	44	M	25	V	13
E	27	N	5	W	16
F	28	O	74	X	78
G	16	P	27	Y	19
H	35	Q	3	Z	1
I	74	R	77	.....	

`character_frequency( message )`

`[ 0.073, 0.009, 0.030, ..., 0.019, 0.001]`
`M_a M_b M_c M_y M_z`

$$M_{\text{both i's}} = \frac{n_i}{n_{text}} \cdot \frac{n_i - 1}{n_{text} - 1}$$

`So for long messages...`

$$M_i^2 \approx \left(\frac{n_i}{n_{text}}\right)^2$$

North Carolina School of Science and Mathematics

North Carolina School of Science and Mathematics

Dan Boneh

12

Identifying a Polyalphabetic Cipher

Monoalphabetic

A	73	J	2	S	63
B	9	K	3	T	93
C	30	L	35	U	130
D	44	M	25	V	13
E	27	N	5	W	16
F	28	O	74	X	78
G	16	P	27	Y	19
H	35	Q	3	Z	1
I	74	R	77	.....	

This number is around 0.066 for monoalphabetic ciphertexts

North Carolina School of Science and Mathematics

```
character_frequency( message )

[ 0.073, 0.009, 0.030, ..., 0.019, 0.001]
   M_A     M_B     M_C          M_Y     M_Z
```

Index of Coincidence

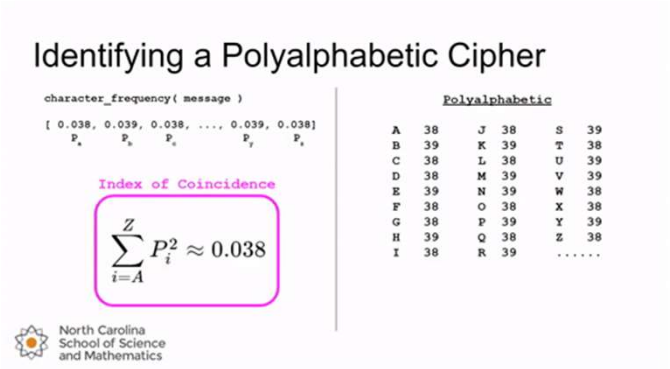
$$\sum_{i=A}^{Z} M_i^2 \approx 0.066$$

Probability of picking 2 A OR 2 B OR 2 C ...

Dan Boneh

13

To identify a Polyalphabetic Cipher



Dan Boneh

Identifying a Polyalphabetic Cipher

Ciphertext:

```
RHVST TEYSJ KMHUM BBCLC GLKBM HBSJH HDAYC PFWHD UUTAP STJAI YMXKA OKARN NATNG CVRCH BNGJU
EMXWH UERZE RLDMX MASKT LAHRJ KIILJ BQCTI BVFZW TKBQE OPKEQ OEBMU NUTAK ZOSLD MKXVO YELLX
SGHTT PNROY MORBW BWZKX FFIQJ HVDZZ JGJZY IGYAT KWVIB VDBBM BNVFC MAXAM CALZE AYAZK HAOAA
ETSGZ AAJFX HUEKZ IAKPM FWXTO EBUGN THMYH FCEKY VRGZA QWAXB RSMSI IWHQM HXRNR XMOEU ALYHN
ACLHF AYDPP JBAHV MXPNF LNWQB WUGOU LGFMO BJGJB PEYVR GZAQW ANZCL XZSVF BISMB KUOTZ TUWUO
WHFIC EBAHR JPCWG CVVEO LSSGN EPGCC SWHYK BJHMF ONHUE BYDRS NVFMR JRCHB NGJUB TYRUU TYVRG
ZAXWX CSADX YIAKL INGXF FEEST UWIAJ EESFT HAHRT WZGTM CRS
```

Index of Coincidence ≈ 0.041709 --> Most Likely Polyalphabetic

North Carolina School of Science and Mathematics

Dan Boneh

keyword length	Index of Coincidence
1	0.066
2	0.052
5	0.044
10	0.041
large	0.038

Dan Boneh

16

Example

```
JAKXQ SWECW MMJBK TQMCM LWCXJ BNEWS XKRBO IAOBI NOMLJ GUIMH
YTACF ICVOE BGOVC WYRCV KXJZV SMRXY VPOVB UBIJH OVCVK RXBOE
ASZVR AOXQS WECVO QJHSG ROXWJ MCXQF OIRGZ VRAOJ RJOMB DBMVS
CIESX MBDBM VSKRM GYFHA KXQSW ECWME UWXHD QDMXB KPUCN HWIWF
NFCKA SKXNF DLJBY RNOBI YFSQN HRIYV IWRQS WCGKC BHRVN SSWYF
SQNTS ZNWCT AWWIB SFIWW CTAWW IWWXI RGKRN LZIAW WIWHK PNFBS
ASVIE SXMBD BMVSK RMGYC NGKPU CNHWI WFNFC KASKX NFDLJ BYRNO
BIYFS QNHRI NBQMW SOVBO IWCVB INWCT AWWIO WFIRG ZVRAO WNJOR
RGZVR AORRB OMBDB MVSOP NJORR GZVRA OXQWB XNSXM BDBMV SPMOH
OIWWC TAWWI
```

Dan Boneh

17

Example

```
ABABA BABAB ABABA BABAB ABABA BABAB ABABA BABAB ABABA BABAB
JAKXQ SWECW MMJBK TQMCM LWCXJ BNEWS XKRBO IAOBI NOMLJ GUIMH


Group 1 (A): JKQWCMJKQCLCJNWXROABNMJUM...   I.C. = 0.06060
Group 2 (B): AXSEWMBTMMWXBESKBIOIOLGIH...   I.C. = 0.05624

                                            Avg. = 0.05842
```

Dan Boneh

18

Example

ABCAB CABCA BCABC ABCAB CABCA BCABC ABCAB CABCA BCABC ABCAB
JAKXQ SWECW MMJBK TQMCM LWCXJ BNEWS XKRBO IAOBI NOMLJ GUIMH

Group 1 (A): JXWWJTCWJEXBAIMGM... I.C. = 0.04405
Group 2 (B): AQEMBQMCBWKOONLUH... I.C. = 0.05108
Group 3 (C): KSCMKMLXNSRIBOJI ... I.C. = 0.04782

Avg. = 0.04765

North Carolina School of Science and Mathematics

Dan Boneh

Example

Key Length	Average I.C.
2	0.05842
3	0.04765
4	0.08340
5	0.04539
6	0.04539
7	0.04814
8	0.08125

A little higher than expected, but most likely candidate for the correct key length

This is also a likely candidate, but so are lengths of 12, 16, 20, etc. Why does that make sense if the key has length 4?

North Carolina School of Science and Mathematics

Dan Boneh

20

The Algorithm

1. Assume key length, *n*, starting with a value of 2
2. Split ciphertext into *n* groups so that characters in the same group would have been enciphered using the same character of the keyword
3. Calculate the index of coincidence of each group
4. Calculate the average index of coincidence of all groups
5. If the average index of coincidence is "close" to the English value of ≈ 0.068 then assume *n* is the correct length
6. If not, increase *n* by 1 and start the process over

North Carolina School of Science and Mathematics

Dan Boneh

How can we crack a Vigener cipher!!

Ciphertext

```
JAKXQ SWECW MMJBK TQMCM LWCXJ BNEWS XKRBO IAOBI NOMLJ GUIMH
YTACF ICVOE BGOVC WYRCV KXJZV SMRXY VPOVB UBIJH OVCVK RXBOE
ASZVR AOXQS WECVO QJHSG ROXWJ MCXQF OIRGZ VRAOJ RJOMB DBMVS
CIESX MBDBM VSKRM GYFHA KXQSW ECWME UWXHD QDMXB KPUCN HWIWF
NFCKA SKXNF DLJBY RNOBI YFSQN HRIYV IWRQS WCGKC BHRVN SSWYF
SQNTS ZNWCT AWWIB SFIWW CTAWW IWWXI RGKRN LZIAW WIWHK PNFBS
ASVIE SXMBD BMVSK RMGYC NGKPU CNHWI WFNFC KASKX NFDLJ BYRNO
BIYFS QNHRI NBQMW SOVBO IWCVB INWCT AWWIO WFIRG ZVRAO WNJOR
RGZVR AORRB OMBDB MVSOP NJORR GZVRA OXQWB XNSXM BDBMV SPMOH
OIWWC TAWWI
```

North Carolina School of Science and Mathematics

Dan Boneh

Find out key length is 4 by I.C.

Ciphertext

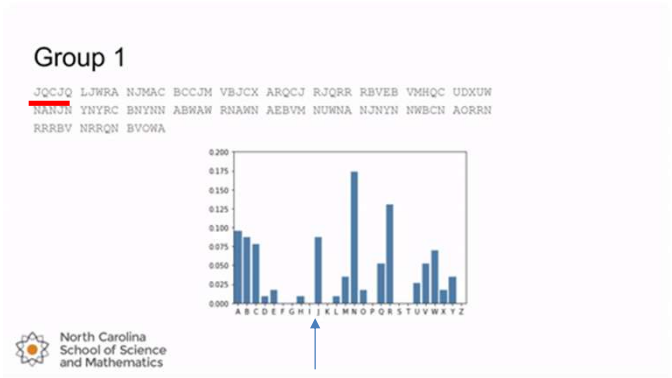
1234

JAKXQ SWECW MMJBK TQMCM LWCXJ BNEWS XKRBO IAOBI NOMLJ GUIMH
YTACF ICVOE BGOVC WYRCV KXJZV SMRXY VPOVB UBIJH OVCVK RXBOE
ASZVR AOXQS WECVO QJHSG ROXWJ MCXQF OIRGZ VRAOJ RJOMB DBMVS
CIESX MBDBM VSKRM GYFHA KXQSW ECWME UWXHD QDMXB KPUCN HWIWF
NFCKA SKXNF DLJBY RNOBI YFSQN HRIYV IWRQS WCGKC BHRVN SSWYF
SQNTS ZNWCT AWWIB SFIWW CTAWW IWWXI RGKRN LZIAW WIWHK PNFBS
ASVIE SXMBD BMVSK RMGYC NGKPU CNHWI WFNFC KASKX NFDLJ BYRNO
BIYFS QNHRI NBQMW SOVBO IWCVB INWCT AWWIO WFIRG ZVRAO WNJOR
RGZVR AORRB OMBDB MVSOP NJORR GZVRA OXQWB XNSXM BDBMV SPMOH
OIWWC TAWWI

North Carolina School of Science and Mathematics

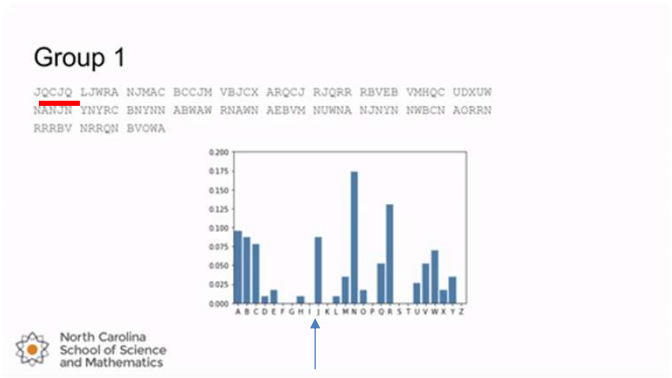
and Mathematics

Dan Boneh


Group 1
JQCJQ LJWRA NJMAC BCCJM VBJCX ARQCJ RJQRR RBVEB VMHQC UDXUW
NANJN YNYRC BNYNN ABWAW RNAWN AEBVM NUWNA NJNYN NWBCN AORRN
RRRBV NRRQN BVOWA
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
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
North Carolina
School of Science
and Mathematics

A

Dan Boneh

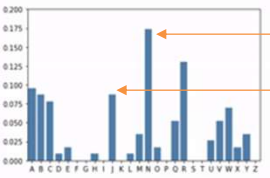

Group 1
JQCJQ LJWRA NJMAC BCCJM VBJCX ARQCJ RJQRR RBVEB VMHQC UDXUW
NANJN YNYRC BNYNN ABWAW RNAWN AEBVM NUWNA NJNYN NWBCN AORRN
RRRBV NRRQN BVOWA
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
North Carolina
School of Science
and Mathematics

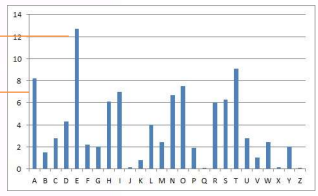
A

Dan Boneh

Group 1

JQCJQ LJWRA NJMAC BCCJM VBJCX ARQCJ RJQRR RBVEB VMHQC UDXUW
NANJN YNYRC BNYNN ABWAW RNAWN AEBVM NUWNA NJNYN NWBCN AORRN
RRRBV NRRQN BVOWA





North Carolina School of Science and Mathematics

Shit 9 like a Caesar cipher

Dan Boneh

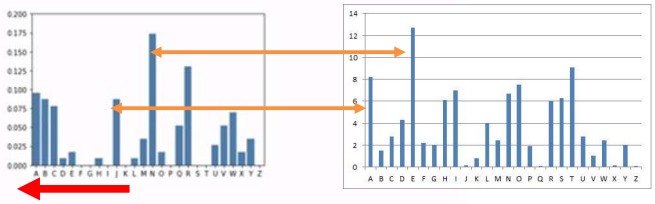
26

How can we find out the right shift

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

Group 1

JQCJQ LJWRA NJMAC BCCJM VBJCX ARQCJ RJQRR RBVEB VMHQC UDXUW
NANJN YNYRC BNYNN ABWAW RNAWN AEBVM NUWNA NJNYN NWBCN AORRN
RRRBV NRRQN BVOWA



North Carolina School of Science and Mathematics

Standard A-Z plaintext frequency distribution in English

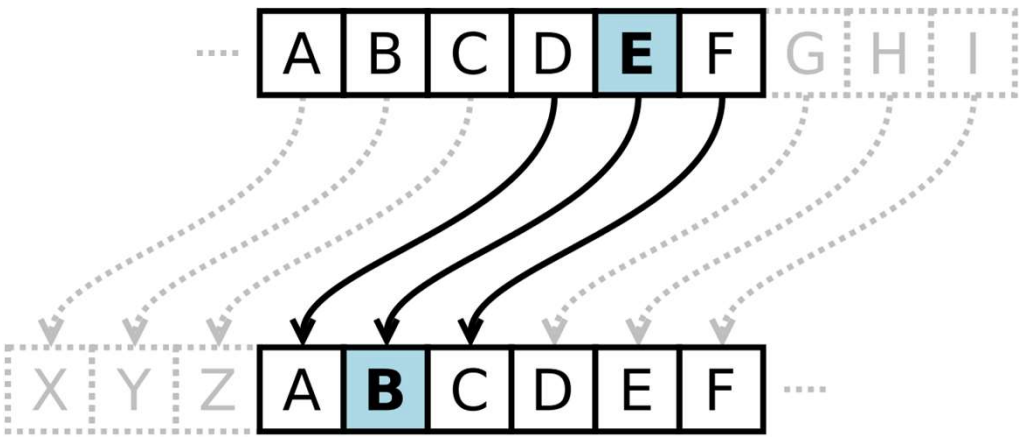
A=f(J)=J+b

Shift 9 like a Caesar Cipher

Dan Boneh

27

Caesar cipher



Dan Boneh

28

I. C.

I.C. = $\frac{\sum_{i=A}^{i=Z}(f_i+f_{i+k'})(f_i+f_{i+k'}-1)}{(N+N')(N+N'-1)}$ $where\ 0 \le k < 26$

$$\sum_{i=A}^{i=Z}\left(f_i^2 + f_i'^2 + 2f_i\, f'_{i+k} - f_i - f_{i+k}'\right)$$
$$= \sum_{i=A}^{i=Z} f_i^2 + \sum_{i=A}^{i=Z} f_{i+k}'^2 + \mathbf{2}\sum_{i=A}^{i=Z} \boldsymbol{f_i f'_{i+k}} - \sum_{i=A}^{i=Z} f_i - \sum_{i=A}^{i=Z} f'_{i+k}$$

$$\sum_{i=A}^{i=Z} f_i f'_{i+k} = 3(2) + 4(2) + 5(1) + \cdots + 0(2) + 5(6)$$

Dan Boneh

The Chi-Square Test

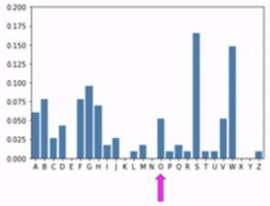
$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

Dan Boneh

30

Group 2

ASWBM WBSBO OGHCV GWVZR PUHVB SASVH OMFGA JDSSD SGASW WQBCI
FSFBO FHVQG HSFTW WSWWW GLWHF SSDSG GCIFS FBOFH BSOVW WWGAJ
GABDS JGAWS DSHWW



North Carolina School of Science and Mathematics

```
 0 (A):   325.7668
 1 (B):   883.3711
 2 (C):  4317.3209
 3 (D):   611.5731
 4 (E):   893.7738
 5 (F):   751.9344
 6 (G):  3896.8727
 7 (H):  1589.9641
 8 (I):  1479.7620
 9 (J):  2843.9011
10 (K):  1107.8685
11 (L):   991.7172
12 (M):   751.9465
13 (N):  2039.9947
14 (O):    50.3009
15 (P):  1269.0717
16 (Q):  1414.7607
17 (R):  1465.7353
18 (S):   553.8066
19 (T):  4314.7729
20 (U):   459.8573
21 (V):  2343.4717
22 (W):  1096.8136
23 (X):  4275.2646
24 (Y):  1108.1107
25 (Z):  1788.0592
```

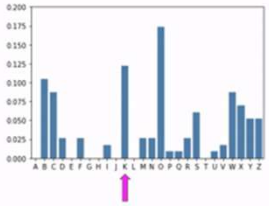
A=f(O)=O+b

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

Dan Boneh

Group 3

KWMKC CNXOB MUYFO OYKVX OBOKO ZOWOS XCOZO OBCXB KYKWM XDKNW
CKDYB SRISK RSSSC WFCWX KZWKB VXBKY KNWCK DYBSR QOIBC WFZOO
ZOOBO OZOBX BPOCW



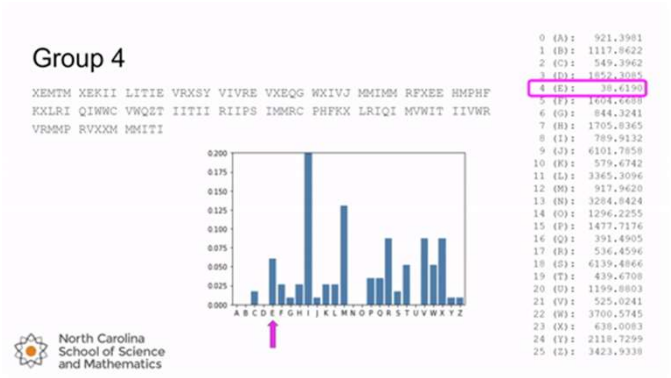

```
 0 (A):  1046.2090
 1 (B):  1561.0377
 2 (C):  2470.9050
 3 (D):  1353.3253
 4 (E):  1357.3504
 5 (F):  2957.6687
 6 (G):  1232.2920
 7 (H):   797.8171
 8 (I):   705.6488
 9 (J):   886.3809
10 (K):    58.9651
11 (L):  3622.6714
12 (M):  1265.4848
13 (N):  2159.6061
14 (O):   552.4254
15 (P):  5049.3125
16 (Q):   549.0219
17 (R):  2475.0480
18 (S):  1151.4112
19 (T):  1490.7523
20 (U):  2040.6587
21 (V):   491.1330
22 (W):   340.3210
23 (X):  1335.5712
24 (Y):  4497.8480
25 (Z):  1250.2771
```

North Carolina School of Science and Mathematics

$A=f(K)=K+b$

Dan Boneh

32



$A=f(E)=E+b$

Dan Boneh

Ciphertext: Keyword = JOKE

```
JAKXQ SWECW MMJBK TQMCM LWCXJ BNEWS XKRBO IAOBI NOMLJ GUIMH
YTACF ICVOE BGOVC WYRCV KXJZV SMRXY VPOVB UBIJH OVCVK RXBOE
ASZVR AOXQS WECVO QJHSG ROXWJ MCXQF OIRGZ VRAOJ RJOMB DBMVS
CIESX MBDBM VSKRM GYFHA KXQSW ECWME UWXHD QDMXB KPUCN HWIWF
NFCKA SKXNF DLJBY RNOBI YFSQN HRIYV IWRQS WCGKC BHRVN SSWYF
SQNTS ZNWCT AWWIB SFIWW CTAWW IWWXI RGKRN LZIAW WIWHK PNFBS
ASVIE SXMBD BMVSK RMGYC NGKPU CNHWI WFNFC KASKX NFDLJ BYRNO
BIYFS QNHRI NBQMW SOVBO IWCVB INWCT AWWIO WFIRG ZVRAO WNJOR
RGZVR AORRB OMBDB MVSOP NJORR GZVRA OXQWB XNSXM BDBMV SPMOH
OIWWC TAWWI
```

North Carolina School of Science and Mathematics

Dan Boneh

Plaintext

```
amath emati ciana physi cista ndane ngine erare eacha skedt
oprov ethea ssert ionth atall oddnu mbers great ertha nonea
repri methe mathe matic iansa ysthr eeisp rimef iveis prime
seven ispri meand sobym athem atica lindu ction allod dnumb
ersgr eater thano neare prime theph ysici stsay sthre eispr
imefi veisp rimes eveni sprim enine isane xperi menta lerro
relev enisp rimea ndsoy esall oddnu mbers great ertha nonea
repri methe engin eersa ysthr eeisp rimef iveis prime seven
ispri menin eispr imeel eveni sprim ethir teeni sprim efift
eenis prime
```

North Carolina School of Science and Mathematics

Dan Boneh

35

Exercise

1. Please write a program to determine the keyword length of these two encrypted messages using I.C.
2. Then write a program to solve the encryption keyword letters
3. Finally, break this ciphertext and recover the plaintext.

Reference https://macs4200.org/chapters/08/5/determining-keyword.html

Dan Boneh

Encrypted message 1

- ZQQTK PQUWD PGMWD BQTXY LFQWL SHAJB UCIPV KUQEJ RBAAC LRSIZ ZCRWT LDFMT PGYXF ISOSE ASZXN PHTAY HHIIR ADDIJ LBFO E VKUWW VFFLV TCEXG HFFXF ZVGXF BFQEI ZOSEZ UGFGF UJUGK PCZWZ UQQJI VAFLV CSDCX YOPYR SQTEI HQFII VTAYI LRGGR AWAR N LAGWK JCZXZ UIMPC FTAVX LHMRU LAMRT PDMXV VIDWV SJQWW YCYOE VKXIU NSBVV CWAYJ SMMGH BWDIU DSYYJ AGQXR ZWP IF SRZSK PCZWR URQQS YOOIW YSELF USEEE KOEAV SSMVE DSYYJ APQHR PZKYE SSMVE PBSWF TSFLZ UUILZ JVUXY HGOSJ AIERF ZAMP C SONSL YOZHR ULUIK FHAET XIUVV HBPXY PGPMW MWOYC AMMXK HQTIJ PHEIC MAAVV JZAWV SMFSR UOSIZ UKTMT ODDSX YSEW Y HGSEZ USPEJ AFARX HGOIE KSZGP VJQVG YSVYU PQQEE KWZAY PQTTV YGARJ HBPXY PBSWR YSPEP IMPEP MWZHZ UUFLV PFDIR SZQ ZV SWZPZ LIAJK OSUVT VBHIE AWARR SJMPL LHTIJ HAQTI PBOMG SSEAY PQTLR CSEAV WHMAR FHDEU PHUSE HZMFL ZSEEE KKTMT O ODID HYURX YOBMU OOHST HAARX AVQVV CSZYV ZCRWZ USOYI PGFWR UREXI PDBME NHTIK OWZXR DRDCM LWXJI VAMXK YOOXZ C SEYG LFEXZ AWARJ HFQAF YYURX HGMGK PJQPP PBXMK LFMXL YSMWZ UGAGZ LHKXY LQDIU BZUXP VTARV DFUXV YCDXY LDMVK POX MK FCREE VHTII MWZHJ HGBSN LFRYC HHAYT OGFSE LOZHR ZKTSC LGAQV HQTEJ AWEID LBFME AVQLV HZFLP ZQQTK PQUWD VTMXV TDQVR ASOPR ZGAJR UHMKF UWEXJ HGFLV KFQED ZCRGF UGQVM HHUWD VFFLV PABSJ AIDIJ VTBPL YOXMJ AGURV JIDIJ PBFLV JVGVT OVUWK VFKEE KHDEU PHUSE DVQXY LFAJR UQUIE ACDGF TDMVR AWHIC FFQGV UHFMD LGMVV ZINNV JHQHK VJQVP KWRJV YSZXY HBPPZ UURVF THTEK DVUGY AVQME KIXKV UQQSI JFQHL SWFCF MTAVD LFMKV ZQAYC KOXPF DAQVV ZHMXV TSZXJ HFQNV HZAYJ SM IEK JVQHR URFLV TCFMM LGAJK OSIVZ ASDJF YAMWZ TDAVK HBFEE PBSVV KWQRK PBFLV HBMPP ZWESW OWELZ ZHAVP HGFLV MOO XJ OSDIT VFPWG YCNES PZUXP PGMTF DSDJL SOZHK YCGFC LGAQV ASEXR URUXZ ZPKXY PGFVF BPXIJ VAQWK HBPEI KHTEK HZMVX LD AVK PCZSW OWEXF YWOEC LJUHV UQQMJ ZWRXV KQARJ PGFIE JMUWE VZQWJ WSDXZ UOOMF BGMRU LLMGK PBSME PHEHV TOZHJ PBNVZ LTFSN YWFIR OWEXF YMIID BGFOE VKYSI LHTEE TSDIW HQFWY BAMRE HHGVV CWQAV KIZHV YOZME KIOXZ VBAJV EHQRU LRQ BG LFUIE JSUWK OSNIJ AVQPG ACFLV JFUXZ JWEQF MVGQR UVUWK VFKLZ ZHAVZ JOXGY HFMGK LFEGR UCZPP ISQWK PAMXV KPKXY L GFEE KODHN OWOLY BAMRV EDQVZ LBOIN OSFLV YOOXL HZAVK YOPMK PCZEI FVMWW BFZMJ OSPXF MCDQT VFDIT AJUIN ZCRME K WHMU BOXWN LAGWK YSSEI KHTID HGRSI TWZKG HFFWF MOSVV HHILF SSIID BGFQV HGGVV AVQQS FHTIZ YFQPR AWARK VHTID HGE SW ISURX ZPKAY VAFLV FODIJ BFDSL URQHR URURT VBFID WZMXZ UUFLV PBOMU LBFWZ UHTIZ YZUZV ZCDGF URUXZ VBILZ JVFVR KW FMF UVMWY HBPIU KCIRK VIEAV TIEXI HHTII JCZWZ KSDXY LUQRV YOXFV HFURX VTFLV DVAPV UODVR AWHIK OOZXY LFQWG LQFMM LDDSS HPUPZ AMAJZ AGPIK HWXW

Dan Boneh

37

Encrypted message 2 **HINT key length could be 5 or 6**

- IVIKDKDQMJGLPWLZGMPFBJIIDBBYSLJDXFGBIWWEHAPHEYSGNCCYOOTSTZABCOBVRTAZEYWVWWAZAIDGAZ
  PETHPVBPWOBVJXGFMDOBCGPFKXKSZZAIGCJRPETACJHUTHPVHKJHPZHFPMEVZEQSBYOMHSDVFTASFGZTC
  OBZCGHFMDOBCWVNVBRVKRGXDBMKFBTGBVGMPTBVFMTGBLBMXZWESHGCBYSKDTBYSFWOARQHCJQEQBC
  UIDCNCHWWGNEDWIHPTKQCZGDKIGDENHPZGIGWVTWIASBFHATQIJSBCDWZBMPGQKKTHTQIGMEFMJSGISLK
  CFTHPVFXLSZVHAGSMGCLHWJCSXMDTRBTIWWEGHUHPVGXRZCJWHCCZZBVPFKVFTIWWECYIVQJUXCHTVAT
  CWVRBHJHPFILTCNYWLUOBYSKHAIEGBDBBYSKTKIJHATSFGZTCOBZCGIVIKVXLOAZBAXRQEUYDFITFBBSWIHA
  PHPVKTHAIUOGSHPRHMWSGNWLWSLKCTKCQUOGPGGCIFDFBYOMWSPRRLDAMUWLTOAVKAXQPTONHSLYWL
  HSOISZPHQFBBRCCCRMWWVBCYCCWKVXGOLVENPHMJCEJHQFBLIVMJSMWSVYOWICJVGBUHMUOGSPICOGR
  SLRUTXBAKSTRVWKVXG

Dan Boneh

End of Segment

39

Online Cryptography Course — Dan Boneh

See also: http://en.wikibooks.org/High_School_Mathematics_Extensions/Discrete_Probability

Introduction

Discrete Probability (crash course, cont.)

Recap

U: finite set (e.g. $U = \{0,1\}^n$)

Prob. distr. P over U is a function $P: U \longrightarrow [0,1]$ s.t. $\sum_{x \in U} P(x) = 1$

$A \subseteq U$ is called an **event** and $\Pr[A] = \sum_{x \in A} P(x) \in [0,1]$

A **random variable** is a function $X: U \longrightarrow V$.

X takes values in V and defines a distribution on V

Dan Boneh

Recap

A **Random Process or Stochastic Process** defined as a family of random variables.

Dan Boneh

42

Independence

Def: events A and B are **independent** if $\Pr[A \text{ and } B] = \Pr[A] \cdot \Pr[B]$

random variables X,Y taking values in V are **independent** if

$\forall a,b \in V$: $\Pr[X=a \text{ and } Y=b] = \Pr[X=a] \cdot \Pr[Y=b]$

Example: $U = \{0,1\}^2 = \{00, 01, 10, 11\}$ and $r \xleftarrow{R} U$

Define r.v. X and Y as: $X = \text{lsb}(r)$, $Y = \text{msb}(r)$

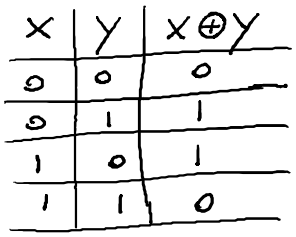
$\Pr[X=0 \text{ and } Y=0] = \Pr[r=00] = ¼ = \Pr[X=0] \cdot \Pr[Y=0]$

Dan Boneh

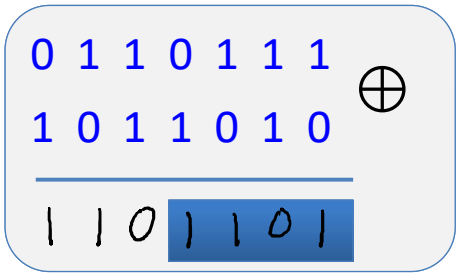
43

Review: XOR

XOR of two strings in $\{0,1\}^n$ is their bit-wise addition mod 2



x	y	x⊕y
0	0	0
0	1	1
1	0	1
1	1	0



Dan Boneh

An important property of XOR

Thm: Y a rand. var. over $\{0,1\}^n$, X an indep. uniform var. on $\{0,1\}^n$

Then $Z := Y \oplus X$ is uniform var. on $\{0,1\}^n$

Proof: (for n=1)

$\Pr[Z=0] = \Pr[(x,y)=(0,0) \text{ or } (x,y)=(1,1)] =$

$= \Pr[(x,y)=(0,0)] + \Pr[(x,y)=(1,1)] =$

$= \frac{p_0}{2} + \frac{p_1}{2} = \frac{1}{2}$

Y	Pr
0	p_0
1	p_1

X	Pr
0	1/2
1	1/2

X	Y	Pr
0	0	$p_0/2$ ⇐
0	1	$p_1/2$
1	0	$p_0/2$
1	1	$p_1/2$ ⇐

Dan Boneh

假設你有一個源源不絕的亂數流，要怎麼和明文作用達到加密的效果呢

假設你有一個源源不絕的亂數流，要怎麼和明文作用達到加密的效果呢？

當然，我們希望密文也是呈隨機亂數

但是明文通常都有不均勻的統計現象，怎麼辦？

XOR就是一個方法

不管你的明文有多明顯的現象，XOR亂數流之後都會像亂數流

問題：難道沒有別的方法嗎？

有，譬如各種block cipher ，但是XOR是運算最簡單的一種

可用於加解密效率需求很高的地方

此外，即使是block cipher ，也是由基本運算組成，XOR的這個特性使得只要產生亂數隨機，那XOR之後就會隨機，充分達到confusion 注意沒有Diffusion)

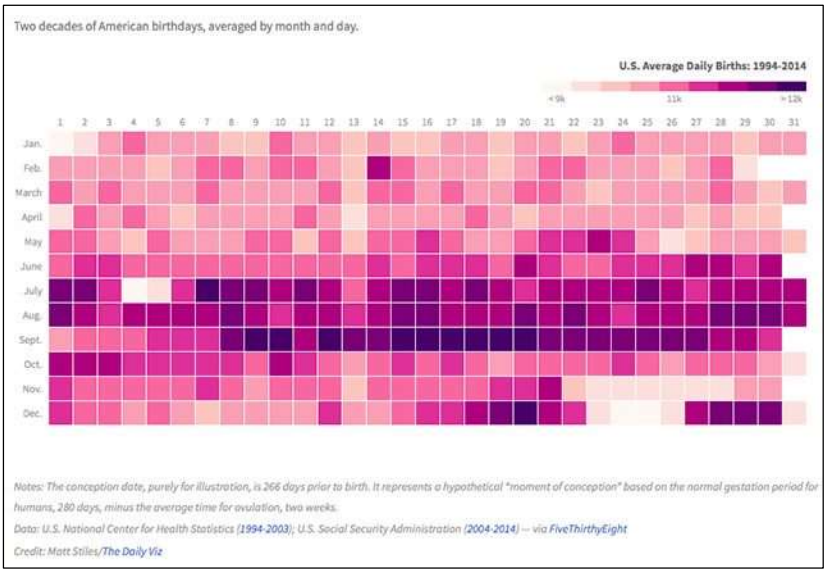
Dan Boneh

46

- A paradox is a statement or concept that contains conflicting ideas.

Dan Boneh

47


Two decades of American birthdays, averaged by month and day.
U.S. Average Daily Births: 1994-2014
< 9k
11k
> 12k
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Jan.
Feb.
March
April
May
June
July
Aug.
Sept.
Oct.
Nov.
Dec.
Notes: The conception date, purely for illustration, is 266 days prior to birth. It represents a hypothetical "moment of conception" based on the normal gestation period for humans, 280 days, minus the average time for ovulation, two weeks.
Data: U.S. National Center for Health Statistics (1994-2003); U.S. Social Security Administration (2004-2014) -- via FiveThirtyEight
Credit: Matt Stiles/The Daily Viz

Dan Boneh

48

- There are n people in a room, what is the probability that at least two people have the same birthday?
- For n=2: P(2) = 1 - $\frac{364}{365}$
- For n=3: P(3) = 1 – ($\frac{364}{365} \times \frac{363}{365}$)
- For n persons: P(n) = 1 – ($\frac{364}{365} \times \frac{363}{365} \times \ldots \times \frac{365-n-1}{365}$)

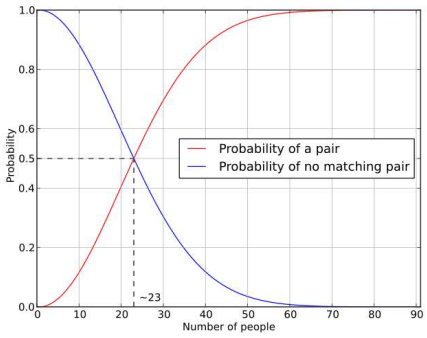
不同生日的機率

- With 22 people in a room, there is better than 50% chance that two people have a common birthday.
- With 40 people in a room there is almost 90% chance that two people have a common birthday.

Dan Boneh

49

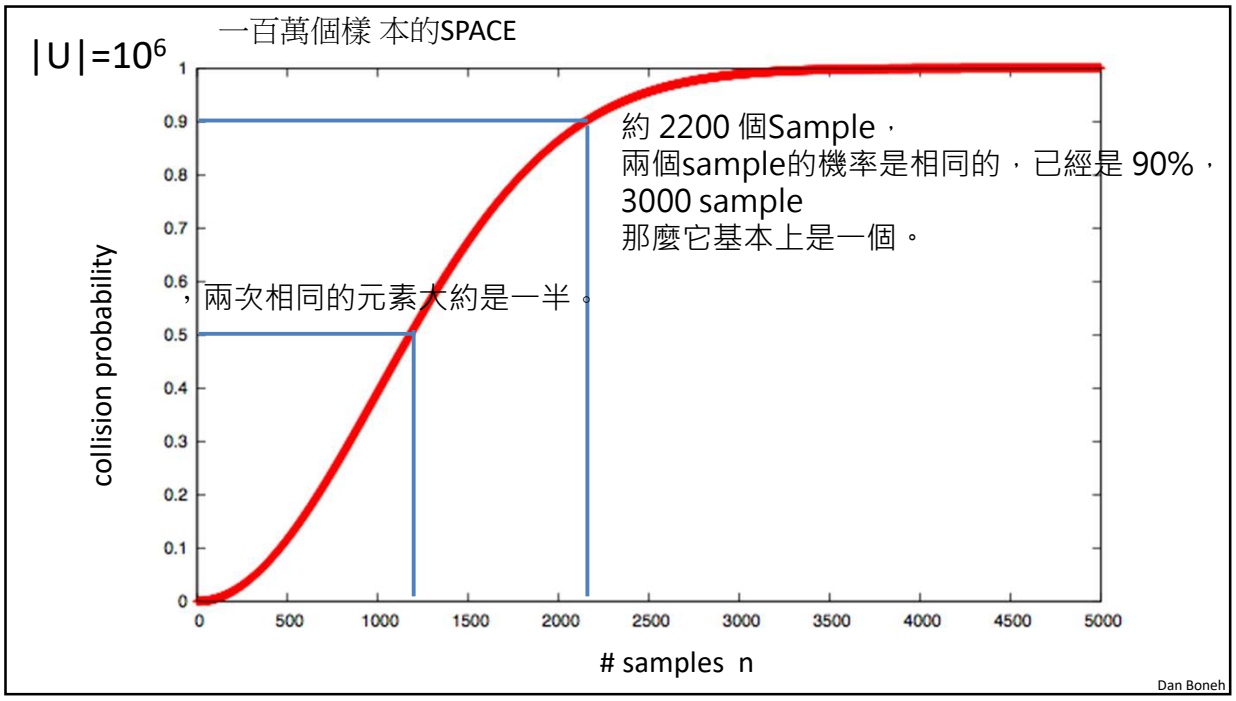
- If n $\geq \sqrt{365}$ then this probability is more than half.
- In general, if there are k possibilities then on average $\sqrt{k}$ trials are required to find a collision.



n	$p(n)$
1	0.0%
5	2.7%
10	11.7%
20	41.1%
23	50.7%
30	70.6%
40	89.1%
50	97.0%
60	99.4%
70	99.9%
100	99.99997%
200	99.99999999999999999999999998%
300	$(100 - (6\times10^{-80}))\%$
350	$(100 - (3\times10^{-129}))\%$
365	$(100 - (1.45\times10^{-155}))\%$
366	100%
367	100%

Dan Boneh

50


|U|=10^6
一百萬個樣 本的SPACE
collision probability
samples n
約 2200 個Sample，
兩個sample的機率是相同的，已經是 90%，
3000 sample
那麼它基本上是一個。
，兩次相同的元素大約是一半。
Dan Boneh

51

The birthday paradox

Let $r_1, \ldots, r_n \in U$ be indep. identically distributed random vars.

Thm: when $n = 1.2 \times |U|^{1/2}$ then $\Pr[\exists i \neq j: r_i = r_j] \geq ½$

notation: $|U|$ is the size of U

Example: Let $U = \{0,1\}^{128}$

After sampling about 2^{64} random messages from U,

some two sampled messages will likely be the same

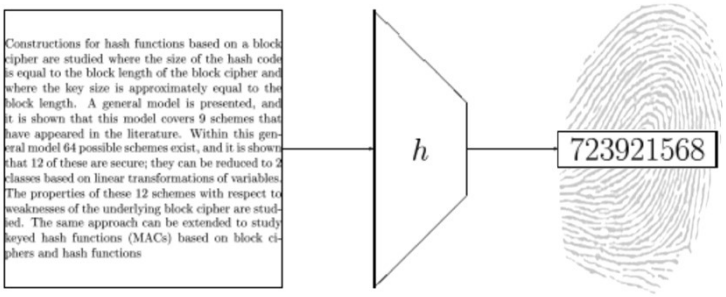
Dan Boneh

Hash Functions

- A hash function takes a variable length message M and produces a fixed length message digest.
- If the length of the digest is m then there are 2^m possible message digests.
- More than one message will be mapped to the same digest.

Dan Boneh

- An efficient function mapping binary strings of **arbitrary length** to binary strings of **fixed length**, called the **hash-value** or **hash-code** (**fingerprint**, **checksum**)

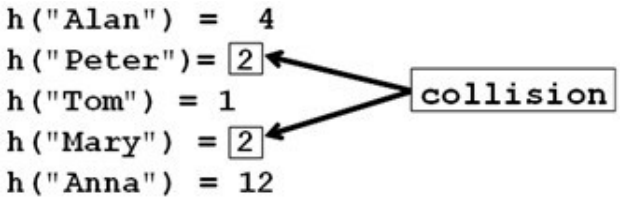


54

Dan Boneh

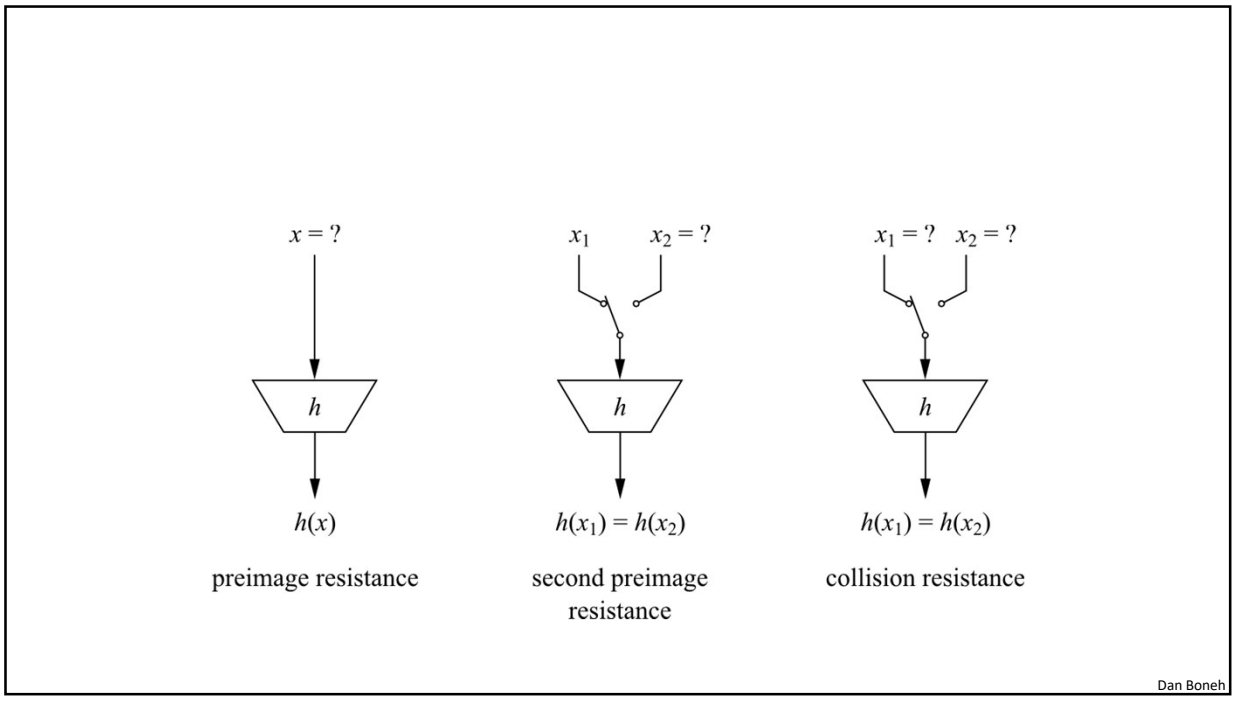
Probability of Hash Collisions

- If we apply k random messages to our hash code what must the value of k to have probability of 0.5 that at least one duplicate?



Using previous equation, we have k = $\sqrt{2^m} = 2^{m/2}$

Dan Boneh


x = ?
h
h(x)
preimage resistance
x1
x2 = ?
h
h(x1) = h(x2)
second preimage resistance
x1 = ?
x2 = ?
h
h(x1) = h(x2)
collision resistance
Dan Boneh

56

Birthday Attack

- Consider a hash function that gets an arbitrary message and outputs a n-bit digest.
- There are 2^n possible digests.
- **Then we need to try an average of $2^{n/2}$ messages to find two with the same digest.**
- For a 64-bit digest, this requires 2^{32} tries.
- For a 128-bit digest, this requires 2^{64} $(\sim 10^{19})$ tries. (That is computationally infeasible.)

Dan Boneh

57

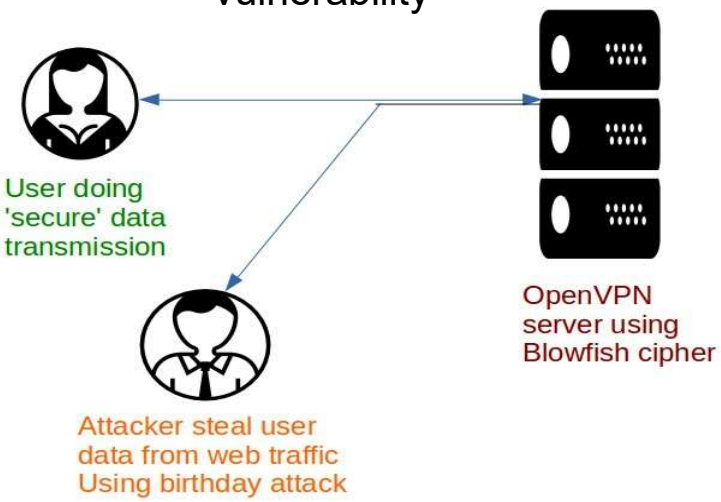
Birthday Attack…

- The two sets of messages are compared to find a pair of messages that produce the same hash code. The probability of success is greater than 0.5. If no match is found, additional valid and fraudulent messages are generated until a match is made.
- **The attacker offers the valid variation to A for signature**. This signature can then be attached to the fraudulent variation for transmission to the intended recipient. Because the two variations have the same hash code, they will produce the same signature; the attacker is assured of success even though the encryption key is not known.

Dan Boneh

58

Sweet32 birthday attacks OpenVpn blowfish cipher vulnerability



https://bobcares.com/blog/how-to-prevent-sweet32-birthday-attacks-openvpn-blowfish-cipher-vulnerability/

Dan Boneh

59

Bonus Exercise

1. Building an elementary MD5 hash cracker program via Python Programming to recover hash value.

   5f4dcc3b5aa765d61d8327deb882cf99

Reference

https://nicholasmordecai.co.uk/programming/creating-simple-python-md5-cracker/

https://s1.nordcdn.com/nord/misc/0.55.0/nordpass/200-most-common-passwords-en.pdf

Dan Boneh

End of Segment

Symmetric Ciphers: definition

Def: a **cipher** defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

is a pair of "efficient" algs (***E***, ***D***) where

$E: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, $D: \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

s.t. $\forall m \in \mathcal{M}, k \in \mathcal{K}: \; D(k, E(k, m)) = m$

- ***E*** is often randomized. ***D*** is always deterministic.

Dan Boneh

62

The One Time Pad (Vernam 1917)

First example of a "secure" cipher

$M = C = \{0,1\}^n$, $\mathcal{K} = \{0,1\}^n$

key = (random bit string as long the message)

Dan Boneh

The One Time Pad (Vernam 1917)

$C := E(K,m) = K \oplus m$

$D(K,C) = K \oplus C$

msg:	0	1	1	0	1	1	1
key:	1	0	1	1	0	1	0
CT:							

$\oplus$

Indeed:

$D(K, E(K,m)) = D(K, K \oplus m) = K \oplus (K \oplus m) = (K \oplus K) \oplus m = 0 \oplus m = m$

Dan Boneh

You are given a message (m) and its OTP encryption (c).

Can you compute the OTP key from m and c ?

No, I cannot compute the key.

Yes, the key is $k = m \oplus c$.

I can only compute half the bits of the key.

Yes, the key is $k = m \oplus m$.

Dan Boneh

65

The One Time Pad (Vernam 1917)

Very fast enc/dec !!

… but long keys (as long as plaintext)

Is the OTP secure? What is a secure cipher?

Dan Boneh

66

What is a secure cipher?

Attacker's abilities: **CT only attack** (for now)

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k, m) = m$ would be secure

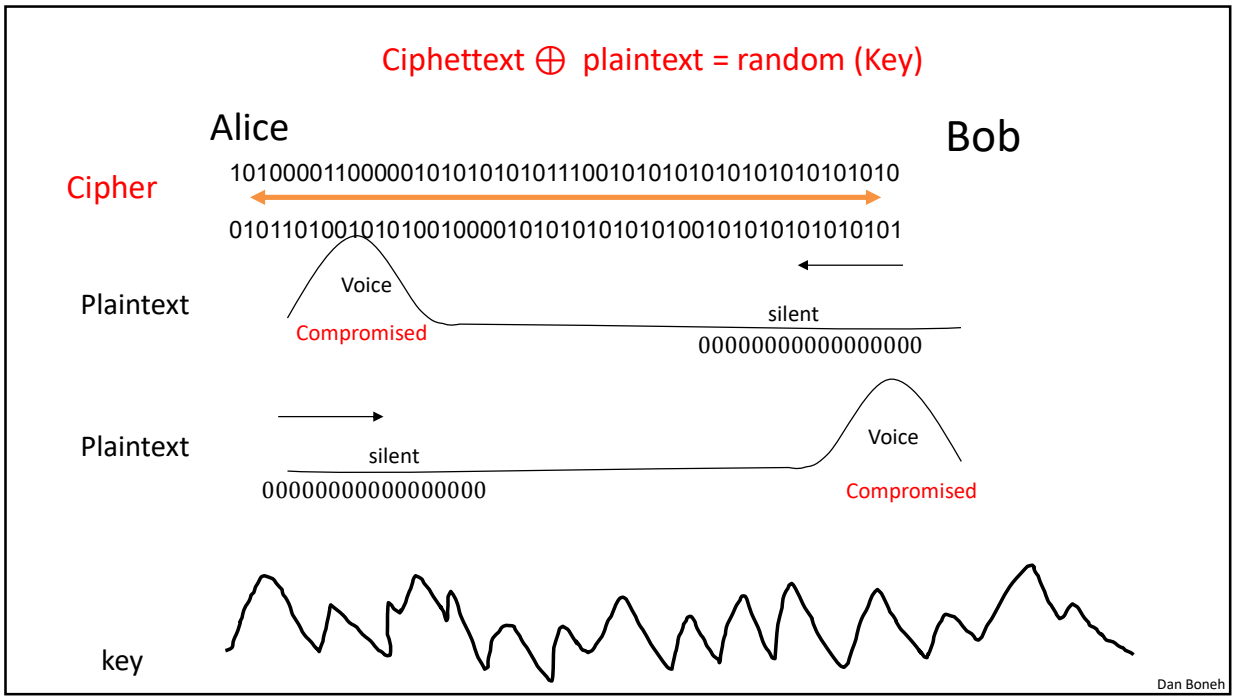
attempt #2: **attacker cannot recover all of plaintext**

$E(k, m_0 \| m_1) = m_0 \| k \oplus m_1$ would be secure

Shannon's idea:

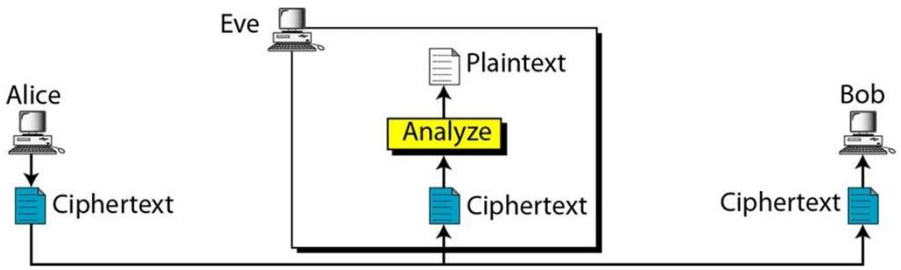
CipherText should reveal no "info" about Plaintext

Dan Boneh


Ciphettext ⊕ plaintext = random (Key)
Alice
Bob
Cipher
10100001100000101010101011100101010101010101010101010
01011010010101001000010101010101010010101010101010101
Plaintext
Voice
Compromised
silent
00000000000000000
Plaintext
silent
00000000000000000
Voice
Compromised
key
Dan Boneh

68

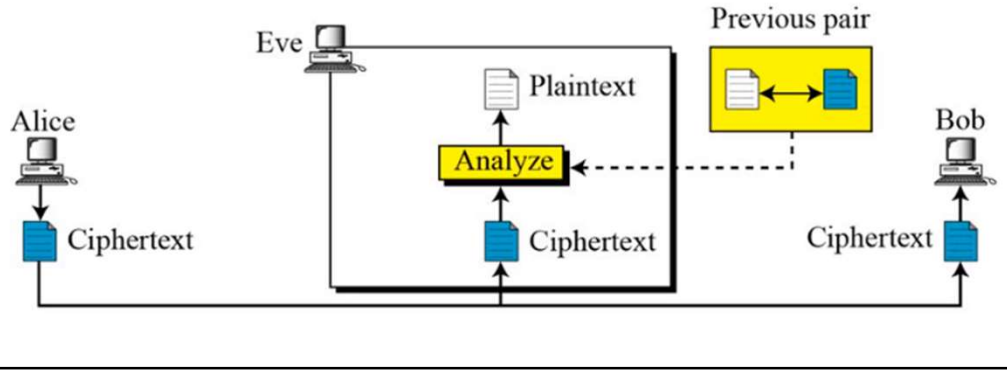
Ciphertext-Only Attack



Dan Boneh

69

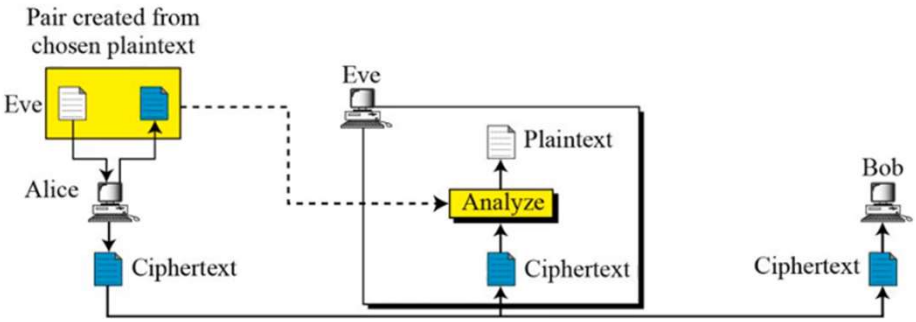
Known-Plaintext Attack



Dan Boneh

70

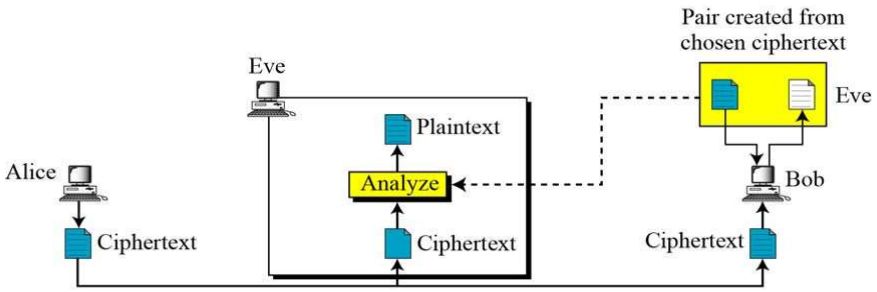
Chosen-Plaintext Attack



Dan Boneh

71

Chosen-Ciphertext Attack



Dan Boneh

72

Chosen Ciphertext Attack

- As a concrete example, suppose General A is sending messages to General B using a Vigenère cipher with an unknown key. The enemy is somehow able to intercept a message and replace it with some completely random letters of his own choosing, say NLLCJOVFXXHMLY. General B decrypts this and gets AKRUWNBXKWNEYX which is nonsense. Bemused, and not thinking this nonsense is worth keeping secret, he picks up a non-secure phone and calls General A: "What the hell do you mean with AKRUWNBXKWNEYX? Did they change the key without telling me?" But the enemy is eavesdropping on the line and now knows that NLLCJOVFXXHMLY decrypts to AKRUWNBXKWNEYX. He can then subtract the two sets of nonsense to get MATHMATHMATHMA, and now he knows the key

Dan Boneh

Chosen plaintext attack

- The difference is how the plaintext-ciphertext pairs that the attacker has access to are generated.
- In a *chosen plaintext attack*, the attacker *chooses* some *plaintext* and is handed the corresponding ciphertext. In other words, the attacker may ***en*crypt arbitrary messages**.
- In a *chosen ciphertext attack*, the attacker can *additionally* (a chosen ciphertext attack is usually understood to subsume a chosen plaintext attack) *choose* some *ciphertext* and is handed the corresponding plaintext. In other words, the attacker may ***en*crypt and *de*crypt arbitrary messages**.

Dan Boneh

74

The Binary Version of One-Time Pad

Plaintext space = Ciphtertext space =

Keyspace = $\{0,1\}^n$

Key is chosen randomly

For example: **Alice encrypted**

- Plaintext is 11011011 m
- Random is 01101001 k
- Then ciphertext is 10110010 c

The Binary Version of One-Time Pad

Plaintext space = Ciphtertext space =

Keyspace = $\{0,1\}^n$

Key is chosen randomly

For example: **Bob decrypt**

- Ciphertext is 10110010
- Random is 01101001
- Plaintext is 11011011

What is a secure cipher?

Attacker's abilities: **Cipher only attack** (for now)

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k, m) = m$ would be secure

attempt #2: **attacker cannot recover all of plaintext**

$E(k, m_0 \| m_1) = m_0 \| k \oplus m_1$ would be secure

Shannon's idea:

CT should reveal no "info" about PT

Dan Boneh

Information Theoretic Security

(Shannon 1949)

Def: A cipher (***E***, ***D***) over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ has **perfect secrecy** if

$\forall m_0, m_1 \in \mathcal{M} \quad (|m_0| = |m_1|)$ and $\forall c \in \mathcal{C}$

$$\Pr[E(k, m_0) = c] = \Pr[E(k, m_1) = c]$$

where k is uniform in $\mathcal{K}$ $(k \xleftarrow{R} \mathcal{K})$

Dan Boneh

```
Plantext  M  m  1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 .....
KEY       K  k  1 0 1 0 1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 .....
Ciphertext C c  0 1 0 1 0 1 1 0 0 1 0 0 1 0 1 1 1 0 1.....
```

Def: A cipher ***(E,D)*** over (K,M,C) has **perfect secrecy** if

$\forall m_0, m_1 \in M \quad (|m_0| = |m_1|) \quad \text{and} \quad \forall c \in C$

$$Pr[\, E(k,m_0)=c \,] \;=\; Pr[\, E(k,m_1)=c \,] \qquad \text{where } k \leftarrow\!\!- K$$

$$m \oplus k = C$$

Lemma: OTP has perfect secrecy.

Proof:

Ignore

$$\forall m, c: \quad \Pr_{k}[E(k,m)=c] = \frac{\#\text{keys } k \in \mathcal{K} \text{ s.t. } E(k,m)=c}{|\mathcal{K}|}$$

So: if $\forall m, c: \#\{k \in \mathcal{K} : E(k,m)=c\} = \text{const.} = 1$

$\Rightarrow$ cipher has perfect secrecy

Def: A cipher ***(E,D)*** over (K,M,C) has **perfect secrecy** if

$\forall m_0, m_1 \in M \quad (|m_0| = |m_1|) \quad \text{and} \quad \forall c \in C$

$$Pr[E(k,m_0)=c] = Pr[E(k,m_1)=c] \quad \text{where } k \xleftarrow{} K$$

Dan Boneh

Information Theoretic Security

Def: A cipher ***(E,D)*** over (K,M,C) has **perfect secrecy** if

$\forall m_0, m_1 \in M \quad (|m_0| = |m_1|) \quad \text{and} \quad \forall c \in C$

$$Pr[E(k,m_0)=c] = Pr[E(k,m_1)=c] \quad \text{where } k \xleftarrow{R} K$$

⇒ Given CT can't tell if msg is m_0 or m_1 (for all m_0, m_1)

⇒ most powerful adv. learns nothing about PT from CT

⇒ no CT only attack!! (but other attacks possible)

Dan Boneh

Let $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

How many OTP keys map m to c ?

None

1 ⟸

2

Depends on m

Dan Boneh

82

Lemma: OTP has perfect secrecy.

Proof:

For OTP: $\forall m, c$: if $E(k,m)=c$

$\Rightarrow k \oplus m = c \Rightarrow k = m \oplus c$

$\Rightarrow \boxed{\#\{k \in \mathcal{K} : E(k,m)=c\} = 1}$

$\Rightarrow$ OTP has perfect secrecy ∎

Dan Boneh

83

The bad news …

Thm: perfect secrecy $\Rightarrow$ $|\mathcal{K}| \geq |\mathcal{M}|$

i.e. perfect secrecy $\Rightarrow$ key-len $\geq$ msg-len

$\Longrightarrow$ hard to use in practice !!

Dan Boneh

How Good is One-Time Pad?

- Intuitively, it is secure …
  - The key is random, so the ciphertext is completely random
- How to formalize the confidentiality requirement?
  - Want to say "certain thing" is not learnable by the adversary (who sees the ciphertext). But what is the "certain thing"?
- Which (if any) of the following is the correct answer?
  - The key.
  - The plaintext.
  - Any bit of the plaintext.
  - **Any information about the plaintext.**
    - E.g., the first bit is 1, the parity is 0, or that the plaintext is not "aaaa", and so on

85
Dan Boneh

Perfect Secrecy: Shannon (Information-Theoretic) Security

- Basic Idea: Ciphertext should provide no "information" about Plaintext
- Have several equivalent formulations:
  - The two random variables **M** and **C** are independent
  - Observing what values **C** takes does not change what one believes the distribution **M** is
  - **Knowing what is value of M does not change the distribution of C**
  - **Encrypting two different messages m_0 and m_1 results in exactly the same distribution.**

86
Dan Boneh

Perfect Secrecy Definition 0

Definition. (**Gen,Enc,Dec)** over a message space $\mathcal{M}$ is perfectly secure if

$\forall$ probability distribution over $\mathcal{M}$

The random variables **M** and **C** are independent.

That is,

$\forall$ message $m \in \mathcal{M}$

$\forall$ ciphertext $c \in \mathcal{C}$

Pr [**M**=m $\wedge$C=c] = Pr [**M** = m] Pr [**C** = c]

87

Dan Boneh

Perfect Secrecy Definition 1

Definition

over a message space $\mathcal{M}$ is perfectly secure if

$\forall$ probability distribution over $\mathcal{M}$

$\forall$ message $m \in \mathcal{M}$

$\forall$ ciphertext $c \in \mathcal{C}$ for which $Pr[C=c] > 0$

We have

$$Pr[\mathbf{M}=m \mid C=c] = Pr[\mathbf{M} = m].$$

88

Dan Boneh

Definition 0 equiv. Definition 1

- Definition 0 implies Definition 1
  - Idea: Given Pr [**M**=m∧C=c] = Pr [**M** = m] Pr [**C** = c], for any c such that Pr [**C** = c] > 0, divide both sides of the above with Pr [**C** = c], we have
    Pr [**M**=m | C=c] = Pr [**M** = m].
- Definition 1 implies Definition 0
  - Idea: ∀ c∈$\mathcal{C}$ s.t. Pr[C=c] > 0
    Pr [**M**=m | C=c] = Pr [**M** = m], multiple both side by Pr[C=c], obtain Pr [**M**=m∧C=c] = **Pr [M = m] Pr [C = c]**
    ∀ c∈$\mathcal{C}$ s.t. Pr[C=c] = 0 we have
    Pr [M=m∧C=c] = 0 = Pr [M=m] Pr[C=c]

89
Dan Boneh

Perfect Secrecy. Definition 2.

Definition in Lemma. **(Gen, Enc, Dec)** over a message space $\mathcal{M}$ is perfectly secure if

$\forall$ probability distribution over $\mathcal{M}$

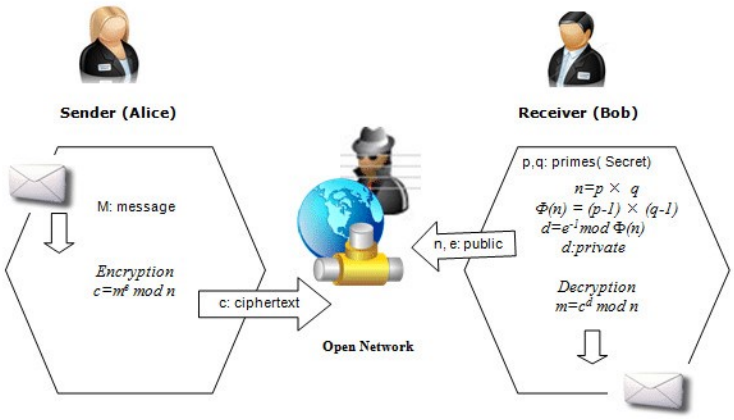
$\forall$ message $m \in \mathcal{M}$ (assuming Pr[**M**=m]>0)

$\forall$ ciphertext $c \in \mathcal{C}$

We have

Pr [C=c | M=m] = Pr [C = c].

Asymmetric Public- Key Cryptosystem
RSA Public-Key cryptosystem for Encryption / Decryption



Dan Boneh

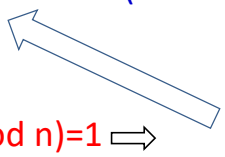
91

An asymmetric ciphers (RSA) Example

Choose **p = 3** and **q = 11** (secret)
Compute n = p * q = 3 * 11 = 33
Compute ϕ(n) = (p - 1) * (q - 1) = 2 * 10 = 20
Choose e such that 1 < e < ϕ(n) and e and n are coprime.
Let e = 7
Compute a value for d such that (d * e) mod ϕ(n) = 1.
One solution is **d = 3** [(3 * 7) mod 20 = 1]
Public key is (e, n) => (7, 33)
Private key is (d, n) => (3, 33)
The encryption of $m = 2$ is $c = 2^7 \mod 33 = 29$
The decryption of $c = 29$ is $m = 29^3 \mod 33 = 2$

Proof for the RSA Algorithm

- $C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+k\phi(n)} \equiv M^1 M^{k\phi(n)} \pmod n$
- $\equiv M^1 \mathbf{M^{\phi(n)\,k}} \pmod n$



- By Euler's theorem $M^{\phi(n)} \pmod n = 1$ ⇨
- $ed \equiv 1 \pmod{\phi(n)}$ $ed = 1 + k\phi(n)$

- example
- p=885320963, q=238855417,
- n=p · q=211463707796206571
- Let e=9007, ∴ d=116402471153538991
- M="cat"=30120, C=113535859035722866

Exercise
Perfect secrecy achieved with RSA?

End of Segment

95

Symmetric Ciphers: definition

Def: a **cipher** defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

is a pair of "efficient" algs (***E***, ***D***) where

$E: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, $D: \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

s.t. $\forall m \in \mathcal{M}, k \in \mathcal{K}: \; D(k, E(k, m)) = m$

- ***E*** is often randomized. ***D*** is always deterministic.

Dan Boneh

96

Stream Ciphers: making OTP practical

idea: replace "random" key by "pseudorandom" key

PRG is a function $G: \underbrace{\{0,1\}^s}_{\text{seed space}} \rightarrow \{0,1\}^n$ $\quad n \gg s$

(eff. computable by a *deterministic* algorithm)

Dan Boneh

Online Cryptography Course — Dan Boneh

Stream ciphers

Pseudorandom Generators

KEY = K
RANDOM NUMBER =R

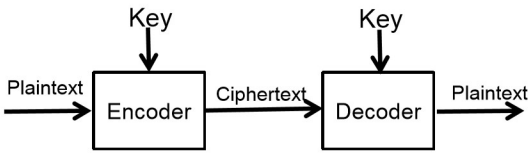


Figure 2: Encoding and Decoding

PRG is a function $G: \{0,1\}^s \rightarrow \{0,1\}^n$ $\quad n \gg s$

seed space

(eff. computable by a deterministic algorithm)

99
Dan Boneh

Threat Model for Encryption

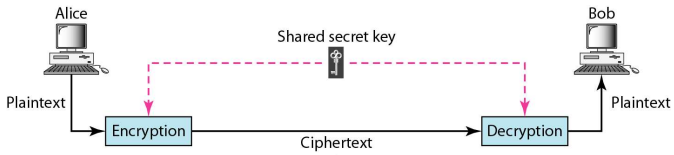
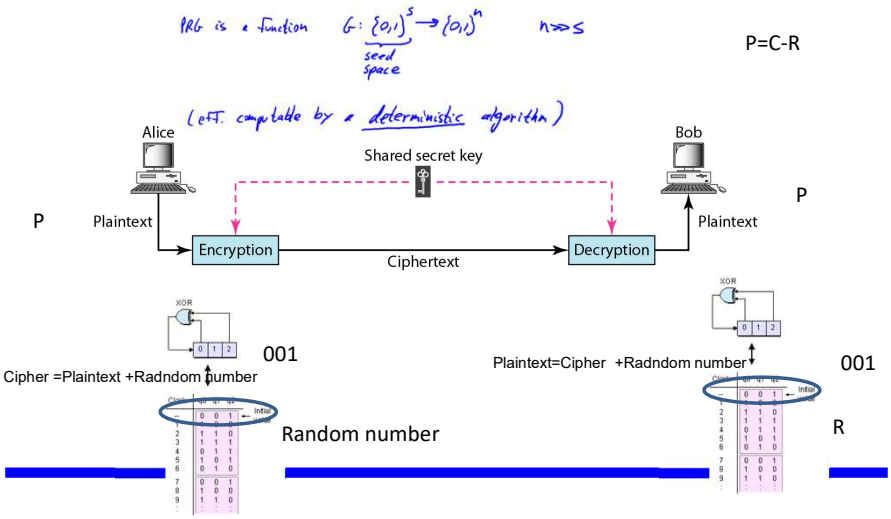


Figure 3 *Symmetric-key cryptography*



Dan Boneh

101

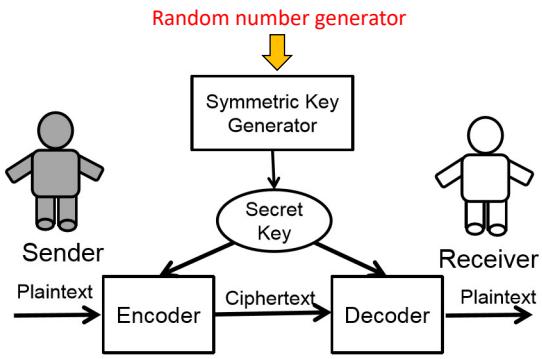
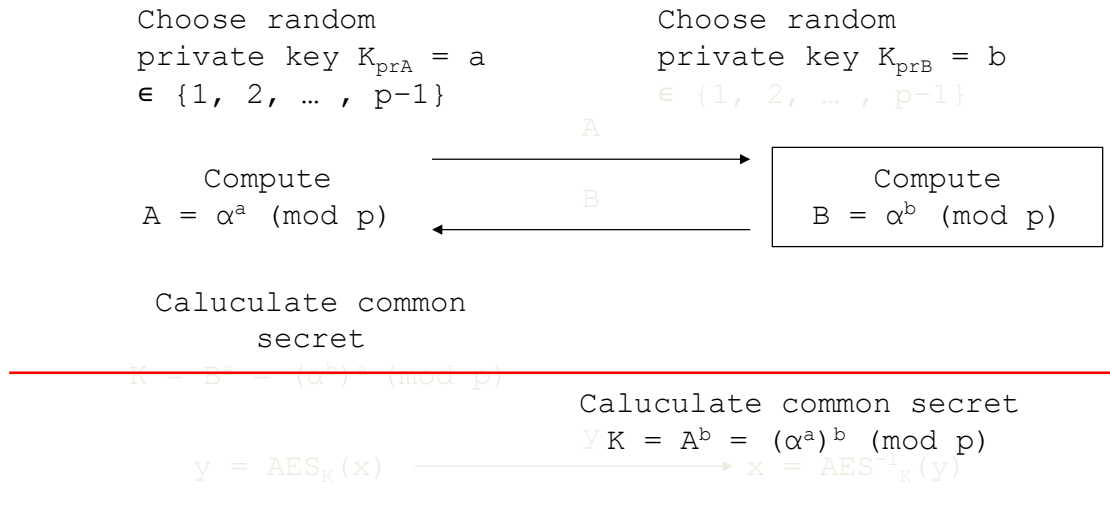


Figure 3: Symmetric Key Algorithm

Diffie-Hellman Key Exchange

- Set-up
  - Choose a large prime p
  - Choose an integer $\alpha \in \{2, 3, \ldots, p-2\}$
  - Publish p and α

Diffie-Hellman Key Exchange



104
Dan Boneh

Applications of LFSRs

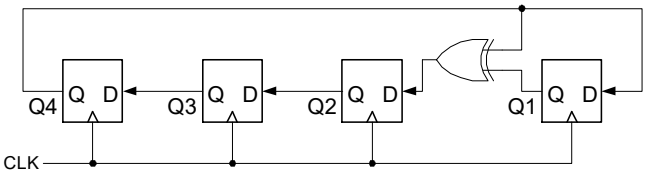
- Performance:
  - In general, xors are only ever 2-input and never connect in series.
  - Therefore the minimum clock period for these circuits is:

    $T > T_{2\text{-input-xor}}$ + clock overhead
  - Very little latency, and independent of n!
- This can be used as a fast counter, if the particular sequence of count values is not important.
  - Example: micro-code micro-pc
- Can be used as a random number generator.
  - Sequence is a pseudo-random sequence:
    - numbers appear in a random sequence
    - repeats every 2^n-1 patterns
  - Random numbers useful in:
    - computer graphics
    - cryptography
    - automatic testing
- Used for error detection and correction
  - CRC (cyclic redundancy codes)
  - ethernet uses them

Dan Boneh

Linear Feedback Shift Registers (LFSRs)

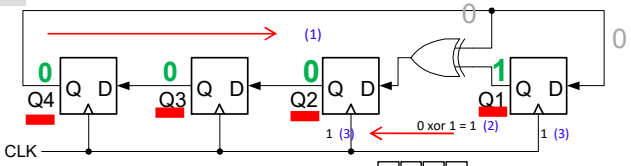
- These are n-bit counters exhibiting *pseudo-random* behavior.
- Built from simple shift-registers with a small number of xor gates.
- Used for:
  - random number generation
  - counters
  - error checking and correction
- Advantages:
  - very little hardware
  - high speed operation
- Example 4-bit LFSR:

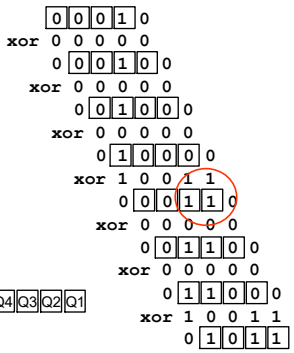


106
Dan Boneh

$\alpha^4 = x^4 \bmod x^4 + x + 1$
$= x^4 \text{ xor } x^4 + x + 1$
$= x + 1$

4-bit LFSR





0001
0010
0100
1000
0011
0110
1100
1011
0101
1010
0111
1110
1111
1101
1001
0001

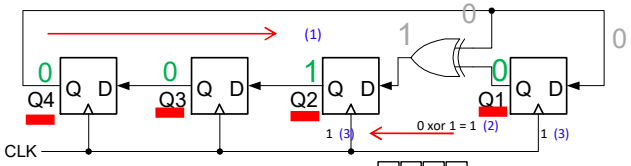
- Circuit counts through 2^4-1 different non-zero bit patterns.
- Left most bit determines shift or more complex operation
- Can build a similar circuit with any number of FFs, may need more xor gates.
- In general, with n flip-flops, 2^n-1 different non-zero bit patterns.
- (Intuitively, this is a counter that *wraps around* many times and in a strange way.)

107

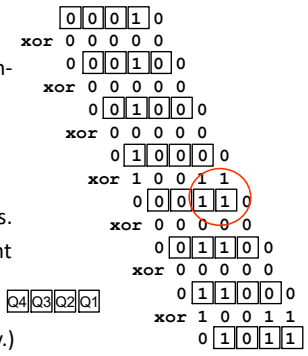
Dan Boneh

$\alpha^4 = x^4 \bmod x^4 + x + 1$
$= x^4 \text{ xor } x^4 + x + 1$
$= x + 1$

4-bit LFSR



- Circuit counts through 2^4-1 different non-zero bit patterns.
- Left most bit determines shift or more complex operation
- Can build a similar circuit with any number of FFs, may need more xor gates.
- In general, with n flip-flops, 2^n-1 different non-zero bit patterns.
- (Intuitively, this is a counter that *wraps around* many times and in a strange way.)



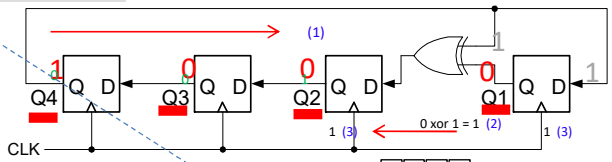
0001
0010
0100
1000
0011
0110
1100
1011
0101
1010
0111
1110
1111
1101
1001
0001

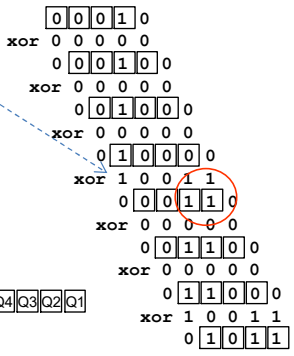
108

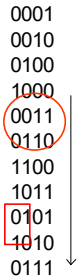
Dan Boneh

$\alpha^4 = x^4 \bmod x^4 + x + 1$
$= x^4 \text{ xor } x^4 + x + 1$
$= x + 1$

4-bit LFSR





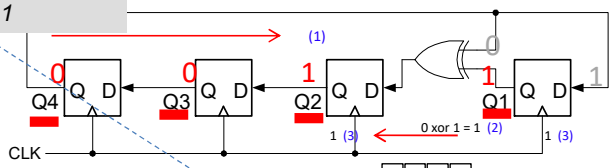


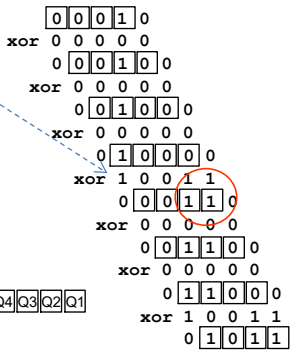
- Circuit counts through 2^4-1 different non-zero bit patterns.
- Left most bit determines shiftl or more complex operation
- Can build a similar circuit with any number of FFs, may need more xor gates.
- In general, with n flip-flops, 2^n-1 different non-zero bit patterns.
- (Intuitively, this is a counter that *wraps around* many times and in a strange way.)

Dan Boneh

$\alpha^4 = x^4 \bmod x^4 + x + 1$
$= x^4 \text{ xor } x^4 + x + 1$
$= x + 1$

4-bit LFSR





- Circuit counts through 2^4-1 different non-zero bit patterns.
- Left most bit determines shiftl or more complex operation
- Can build a similar circuit with any number of FFs, may need more xor gates.
- In general, with n flip-flops, 2^n-1 different non-zero bit patterns.
- (Intuitively, this is a counter that *wraps around* many times and in a strange way.)

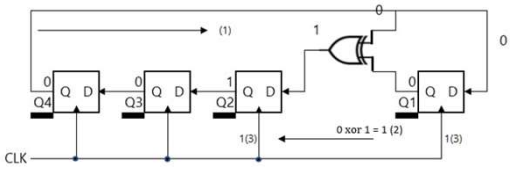
Q4 Q3 Q2 Q1

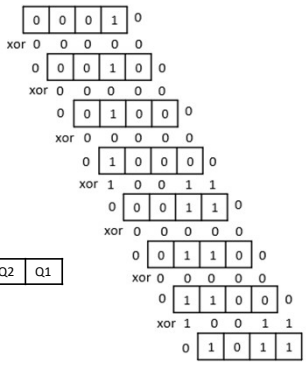
Dan Boneh

4-bit LFSR

$$a^4 = x^4 \bmod x^4 + x + 1$$
$$= x^4 \text{ xor } x^4 + x + 1$$
$$= x + 1$$

其實是$a_{n+4} = a_{n+1} + a_n$
的數列
再由數列生成行多項式





0001
0010
0100
1000
0011
0110
1100
1011
0101
1010
0111
1110
1111
1101
1001
0001

- Circuit counts through $2^4 - 1$ different non-zero bit patterns.
- Left most bit determines shift or more complex operation.
- Can build a similar circuit with any number of FFs, may need more xor gates.
- In general, with n flip-flops, $2^4 - 1$ different non-zero bit patterns.
- (Intuitively, this is a counter that wraps around many times and in a strange way.)

Q4	Q3	Q2	Q1

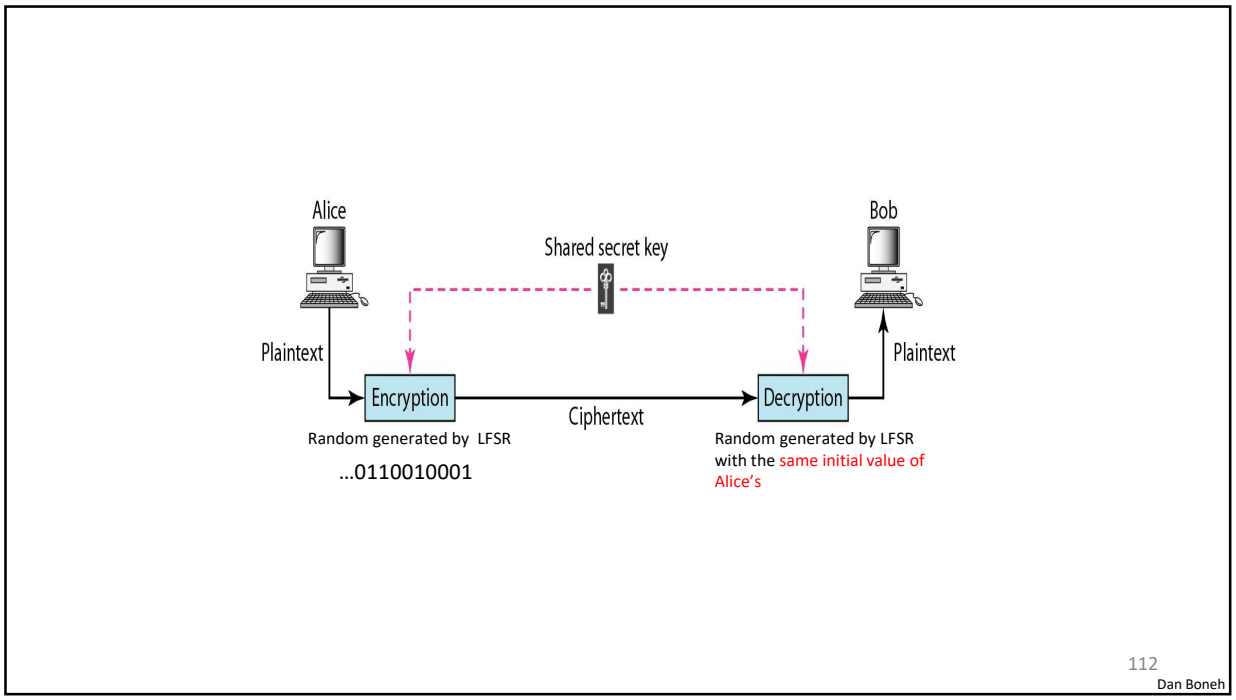
$a_n = r^n, r$待定　　$r^{n+4} = r^{n+1} + r^n,\ r^4 = r + 1$

其實是$a_{n+4} = a_{n+1} + a_n$的數列
再由數列生成行多項式

$a_1 a_2 a_3 a_4 a_5 a_6\ a_7 a_8 a_9 a_{10} a_{11} a_{12}$.....
0 0 0 1 0 0 1 1 0 1 0 1 .....

Dan Boneh

111


Alice
Bob
Shared secret key
Plaintext
Encryption
Ciphertext
Decryption
Plaintext
Random generated by LFSR
...0110010001
Random generated by LFSR
with the same initial value of
Alice's
112
Dan Boneh

112

Galois Fields – Primitives

$\alpha^0 = 1$
$\alpha^1 = x$
$\alpha^2 = x^2$
$\alpha^3 = x^3$
$\alpha^4 = x + 1$
$\alpha^5 = x^2 + x$
$\alpha^6 = x^3 + x^2$
$\alpha^7 = x^3 + x + 1$
$\alpha^8 = x^2 + 1$
$\alpha^9 = x^3 + x$
$\alpha^{10} = x^2 + x + 1$
$\alpha^{11} = x^3 + x^2 + x$
$\alpha^{12} = x^3 + x^2 + x + 1$
$\alpha^{13} = x^3 + x^2 + 1$
$\alpha^{14} = x^3 + 1$
$\alpha^{15} = 1$

- Note this pattern of coefficients matches the bits from our 4-bit LFSR example.

$\alpha^4 = x^4 \bmod x^4 + x + 1$
$= x^4 \text{ xor } x^4 + x + 1$
$= x + 1$

- In general finding primitive polynomials is difficult. Most people just look them up in a table, such as:

Dan Boneh

Galois Fields - Primitives

	Q_4	Q_3	Q_2	Q_1
$a^0 =$				
$a^1 =$				x
$a^2 =$			x^2	
$a^3 =$		x^3		
$a^4 =$				$x+1$
$a^5 =$			x^2	$+x$
$a^6 =$		x^3	$+x^2$	
$a^7 =$		x^3		$+x+1$
$a^8 =$			x^2	$+1$
$a^9 =$		x^3		$+x$
$a^{10} =$			x^2	$+x+1$
$a^{11} =$		x^3	$+x^2$	$+x$
$a^{12} =$		x^3	$+x^2$	$+x+1$
$a^{13} =$		x^3	$+x^2$	$+1$
$a^{14} =$		x^3		$+1$
$a^{15} =$		x^3	$+x^2$	$+x+1$

以上都有解

- Note this pattern of coefficients matches the bits from our 4-bit LFSR example

$x^4 \; mod \; x^4 + x + 1$
$x^4 = (x^4 + x + 1)$商$+(x+1)$餘

$$a^4 = x^4 \; mod \; x^4 + x + 1$$
$$= x^4 \; xor \; x^4 + x + 1$$
$$= x + 1$$

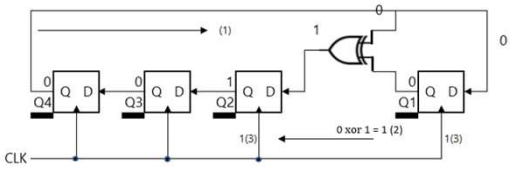
- In general finding primitive polynomials is difficult. Most people just look them up in a table, such as:

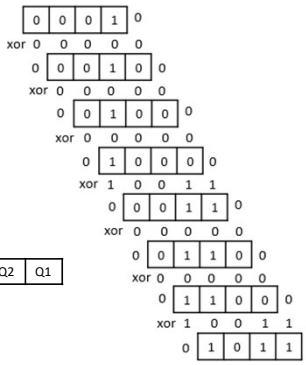
Dan Boneh

4-bit LFSR

$$a^4 = x^4 \; mod \; x^4 + x + 1$$
$$= x^4 \; xor \; x^4 + x + 1$$
$$= x + 1$$

其實是$a_{n+4} = a_{n+1} + a_n$
的數列
再由數列生成行多項式





0001
0010
0100
1000
0011
0110
1100
1011
0101
1010
0111
1110
1111
1101
1001
0001

- Circuit counts through $2^4 - 1$ different non-zero bit patterns.
- Left most bit determines shift or more complex operation.
- Can build a similar circuit with any number of FFs, may need more xor gates.
- In general, with n flip-flops, $2^4 - 1$ different non-zero bit patterns.
- (Intuitively, this is a counter that wraps around many times and in a strange way.)

Q4	Q3	Q2	Q1

$a_n = r^n, r$待定　　$r^{n+4} = r^{n+1} + r, \; r^4 = r + 1$

Dan Boneh

115

Characteristic Polynomial

A shift register generator is simply an arrangement of n stages in a row with the last stage, plus any of the other stages, modulo-two added together and returned to the first stage. This arrangement of stages and tap points can be symbolized by an algebraic expression called the characteristic polynomial.

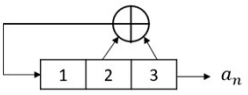


Figure 21

Let stages 1, 2, and 3 of Figure be represented by exponential terms x^1, x^2, x^3, respectively. Since the feedback loop acts as if it were a stage before stage 1, we will represent it by the exponential term x^0. This will result in the equation $x^0 = x^2 + x^3$. By adding x^0 to both sides $x^0 + x^0 = x^0 + x^2 + x^3$. Remembering $x^0 + x^0 = 0$ in mod-two addition, our final equation is $0 = x^0 + x^2 + x^3$. The right-hand side of the equation is the characteristic polynomial for the shift-register generator in Figure. The characteristic polynomial is also known as the generating polynomial.

116

Primitive Polynomials

$x^2 + x + 1$
$x^3 + x + 1$
$x^4 + x + 1$
$x^5 + x^2 + 1$
$x^6 + x + 1$
$x^7 + x^3 + 1$
$x^8 + x^4 + x^3 + x^2 + 1$
$x^9 + x^4 + 1$
$x^{10} + x^3 + 1$
$x^{11} + x^2 + 1$
$x^{12} + x^6 + x^4 + x + 1$
$x^{13} + x^4 + x^3 + x + 1$
$x^{14} + x^{10} + x^6 + x + 1$
$x^{15} + x + 1$
$x^{16} + x^{12} + x^3 + x + 1$
$x^{17} + x^3 + 1$
$x^{18} + x^7 + 1$
$x^{19} + x^5 + x^2 + x + 1$
$x^{20} + x^3 + 1$
$x^{21} + x^2 + 1$
$x^{22} + x + 1$
$x^{23} + x^5 + 1$
$x^{24} + x^7 + x^2 + x + 1$
$x^{25} + x^3 + 1$
$x^{26} + x^6 + x^2 + x + 1$
$x^{27} + x^5 + x^2 + x + 1$
$x^{28} + x^3 + 1$
$x^{29} + x + 1$
$x^{30} + x^6 + x^4 + x + 1$
$x^{31} + x^3 + 1$
$x^{32} + x^7 + x^6 + x^2 + 1$

Galois Field — Hardware

Multiplication by x ⇔ shift left

Taking the result mod $p(x)$ ⇔ XOR-ing with the coefficients of $p(x)$ when the most significant coefficient is 1.

Obtaining all 2^n-1 non-zero elements by evaluating x^k for $k = 1, \ldots, 2^n-1$ ⇔ Shifting and XOR-ing 2^n-1 times.

Dan Boneh

117

Review

Cipher over (K,M,C): a pair of "efficient" algs (***E***, ***D***) s.t.

$\forall\, m \in M,\ k \in K$: $\boldsymbol{D}(k, \boldsymbol{E}(k, m)) = m$

Weak ciphers: subs. cipher, Vigener, …

A good cipher: **OTP** $M=C=K=\{0,1\}^n$

$E(k, m) = k \oplus m$, $D(k, c) = k \oplus c$

Lemma: OTP has perfect secrecy (i.e. no CT only attacks)

Bad news: perfect-secrecy ⇒ key-len ≥ msg-len

Dan Boneh

118

Stream Ciphers: making OTP practical

idea: replace "random" key by "pseudorandom" key

PRG is a function $G: \underbrace{\{0,1\}^s}_{\text{seed space}} \rightarrow \{0,1\}^n$ $\quad n \gg s$

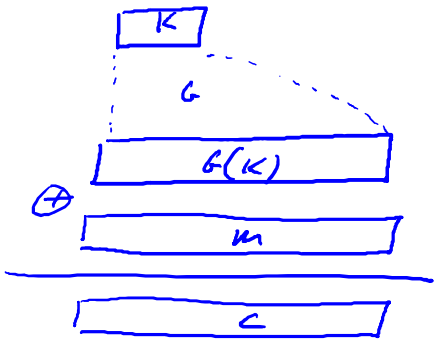
(eff. computable by a *deterministic* algorithm)

Dan Boneh

Stream Ciphers: making OTP practical

$C := E(k,m) = m \oplus G(k)$

$D(k,c) = c \oplus G(k)$



Dan Boneh

Can a stream cipher have perfect secrecy?

- ○ Yes, if the PRG is really “secure”
- ○ No, there are no ciphers with perfect secrecy
- ○ Yes, every cipher has perfect secrecy
- ○ No, since the key is shorter than the message ←

121

Stream Ciphers: making OTP practical

Stream ciphers cannot have perfect secrecy !!

- Need a different definition of security
- Security will depend on specific PRG

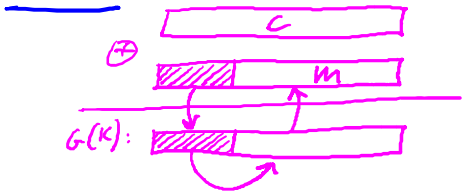
Dan Boneh

PRG must be unpredictable

Suppose PRG is predictable:

$\exists i$: $G(k)\big|_{1,\ldots,i} \xrightarrow{alg} G(k)\big|_{i+1,\ldots,n}$

Then:



even $G(k)\big|_{1,\ldots,i} \rightarrow G(k)\big|_{i+1}$ is a problem!

Dan Boneh

123

PRG must be unpredictable

We say that $G: K \longrightarrow \{0,1\}^n$ is **predictable** if:

$\exists$ "eff" alg. A and $\exists\, 0 \le i \le n-1$ s.t.

$$\Pr_{k \xleftarrow{R} K}\left[A(G(k))\big|_{1,\ldots,i} = G(k)\big|_{i+1} \right] > \tfrac{1}{2} + \varepsilon$$

For non-negligible ε (e.g. $\varepsilon = 1/2^{30}$)

Def: PRG is **unpredictable** if it is not predictable

$\Rightarrow$ $\forall i$: no "eff" adv. can predict bit (i+1) for "non-neg" ε

Dan Boneh

Suppose $G:K \longrightarrow \{0,1\}^n$ is such that for all k: **XOR(G(k)) = 1**

Is G predictable ??

Yes, given the first bit I can predict the second

No, G is unpredictable

Yes, given the first (n-1) bits I can predict the n'th bit ⟸

It depends

Dan Boneh

Weak PRGs (do not use for crypto)

Lin. Cong. generator with parameters a,b,p:

$r[i] \leftarrow a \cdot r[i-1] + b \mod p$

output bits of $r[i]$

i++

seed ≡ $r[0]$

glibc random():

$r[i] \leftarrow (r[i-3] + r[i-31]) \ \% \ 2^{32}$

output $r[i] >> 1$

never use random() for crypto !!
(e.g. Kerberos V4)

Dan Boneh

線性同餘亂數產生器

1. 基本觀念

線性同餘法基本上只有一個公式，**X(n+1) = (a * X(n) + b) mod c**，
但並非所有 a, b, c 都適用。以 a=5, b=0, c=10 為例，假定 X(0)=3 代入
X(1) = (5 * 3 + 0) mod 10 = (15 + 0) mod 10 = 5
X(2) = (5 * 3 + 0) mod 10 = (15 + 0) mod 10 = 5
這完成沒辦法產生亂數，0~9 裡面，只有 5 會不停重覆產生。

2. 常數選擇

重點便是在 a, b, c 應選何值、怎麼選為佳。有些規則可供參考，下列三項規則取自 TAOCP
X(n+1) = (a * X(n) + b) mod c
(1) 乘數 a 與模數 c 互質。
(2) 對於整除 c 之每個質數 p，a-1 抑可整除 p。
(3) 若 c 為 4 之倍數， b 也為 4 之倍數。
在一些文獻記載中，有提到幾個 a, b, c 之選取，可達到全週期之產生器
a = 16807 , b=0, c=2147483647
a = 630360016, b=0, c=2147483647

Dan Boneh

127

End of Segment

128

Online Cryptography Course — Dan Boneh

Stream ciphers

Negligible vs. non-negligible

129

Negligible and non-negligible

- In practice: ε is a scalar and
  - ε non-neg: $\varepsilon \geq 1/2^{30}$ (likely to happen over 1GB of data)
  - ε negligible: $\varepsilon \leq 1/2^{80}$ (won't happen over life of key)

- In theory: ε is a function $\varepsilon: Z^{\geq 0} \longrightarrow R^{\geq 0}$ and
  - ε non-neg: $\exists d: \varepsilon(\lambda) \geq 1/\lambda^d$ inf. often ($\varepsilon \geq$ 1/poly, for many λ)
  - ε negligible: $\forall d, \lambda \geq \lambda_d: \varepsilon(\lambda) \leq 1/\lambda^d$ ($\varepsilon \leq$ 1/poly, for large λ)

Dan Boneh

130

- Global silicon production amounted to an estimated total of eight million metric tons in 2020.
- In 2019, the production of silicon worldwide reached a record high, at 8.41 million metric tons.

Dan Boneh

131

Few Examples

$\varepsilon(\lambda) = 1/2^{\lambda}$: negligible

$\varepsilon(\lambda) = 1/\lambda^{1000}$: non-negligible

$$\varepsilon(\lambda) = \begin{cases} 1/2^{\lambda} & \text{for odd } \lambda \\ 1/\lambda^{1000} & \text{for even } \lambda \end{cases}$$

Negligible

Non-negligible 

Dan Boneh

PRGs: the rigorous theory view

PRGs are "parameterized" by a security parameter λ

- **PRG** becomes "more secure" as λ increases

Seed lengths and output lengths grow with λ

For every $\lambda=1,2,3,\ldots$ there is a different PRG G_λ:

$$G_\lambda : K_\lambda \longrightarrow \{0,1\}^{n(\lambda)}$$

(in the lectures we will always ignore λ)

Dan Boneh

133

An example asymptotic definition

We say that $G_\lambda : K_\lambda \longrightarrow \{0,1\}^{n(\lambda)}$ is **predictable** at position i if:

there exists a polynomial time (in λ) algorithm A s.t.

$$Pr_{k \leftarrow K_\lambda}\left[A\left(\lambda,\ G_\lambda(k)\big|_{1,\ldots,i}\right) = G_\lambda(k)\big|_{i+1} \right] > 1/2 + \varepsilon(\lambda)$$

for some non-negligible function $\varepsilon(\lambda)$

Dan Boneh

End of Segment

Online Cryptography Course — Dan Boneh

Stream ciphers

Attacks on OTP and stream ciphers

136

Review

OTP: $E(k,m) = m \oplus k$, $D(k,c) = c \oplus k$

Making OTP practical using a PRG: $G: K \longrightarrow \{0,1\}^n$

Stream cipher: $E(k,m) = m \oplus G(k)$, $D(k,c) = c \oplus G(k)$

Security: PRG must be unpredictable (better def in two segments)

Dan Boneh

Attack 1: **two time** pad is insecure !!

Never use stream cipher key more than once !!

$C_1 \leftarrow m_1 \oplus PRG(k)$

$C_2 \leftarrow m_2 \oplus PRG(k)$

Eavesdropper does:

$C_1 \oplus C_2 \rightarrow$ 

Enough redundancy in English and ASCII encoding that:

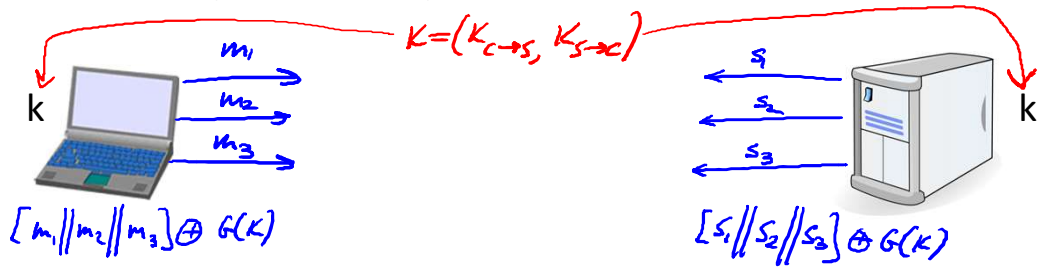
$m_1 \oplus m_2 \rightarrow m_1, m_2$

Dan Boneh

138

Real world examples

- Project Venona
- MS-PPTP (windows NT):

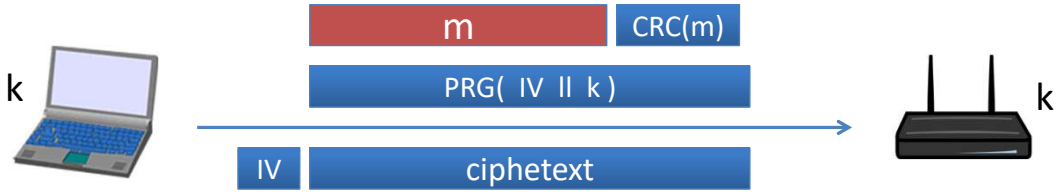


Need different keys for C⟶S and S⟶C

Dan Boneh

Real world examples

802.11b WEP:



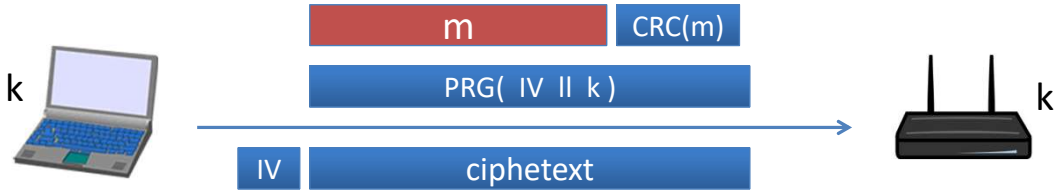
Length of IV: 24 bits

- Repeated IV after $2^{24} \approx 16M$ frames
- On some 802.11 cards: IV resets to 0 after power cycle

Dan Boneh

Avoid related keys

802.11b WEP:



key for frame #1: (1 ll k)

key for frame #2: (2 ll k)

⋮

24 bits

1024 bits

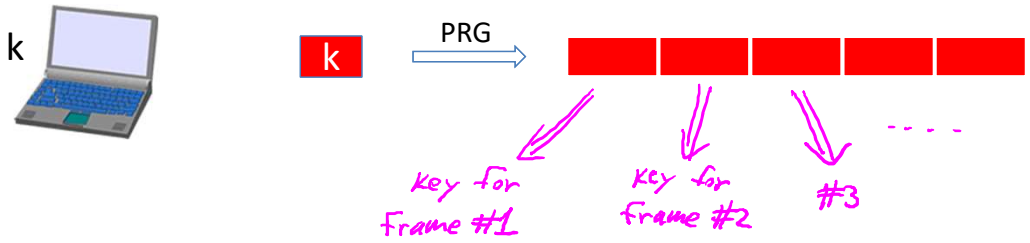
For the RC4 PRG:

FMS2001 ⇒ can recover k after 10^6 frames

Recent attacks ≈ 40,000 frames

Dan Boneh

A better construction

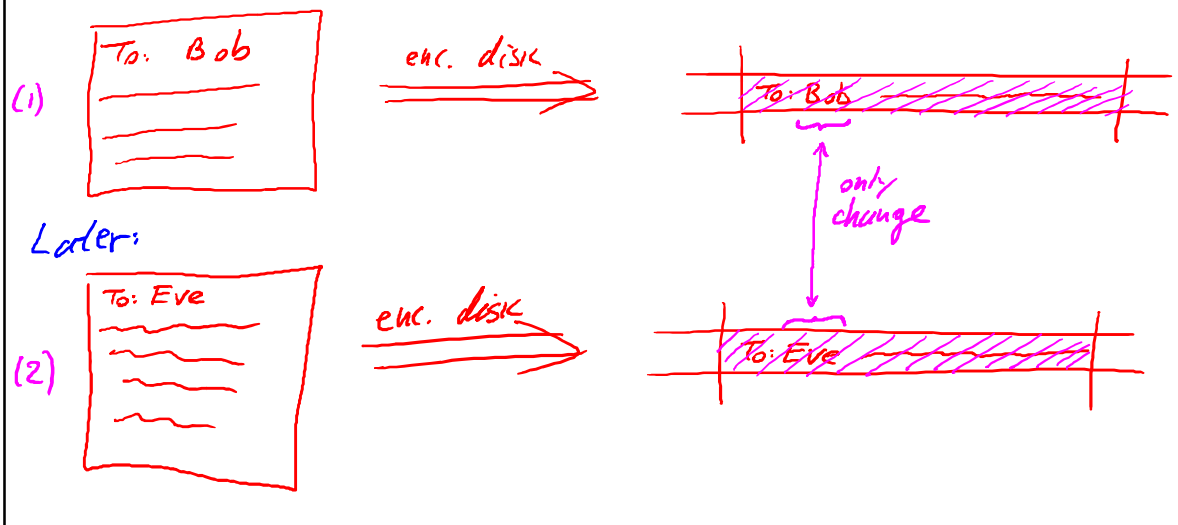


⇒ now each frame has a pseudorandom key

better solution: use stronger encryption method (as in WPA2)

Dan Boneh

Yet another example: disk encryption



Dan Boneh

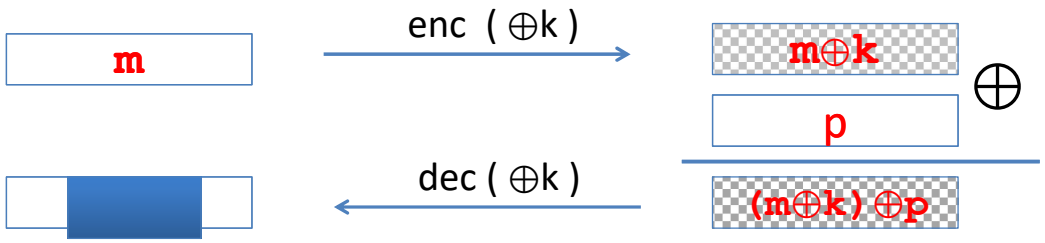
Two time pad: summary

Never use stream cipher key more than once !!

- Network traffic: **negotiate new key for every session** (e.g. TLS)
- Disk encryption: typically do not use a stream cipher

Dan Boneh

Attack 2: no integrity (OTP is malleable)

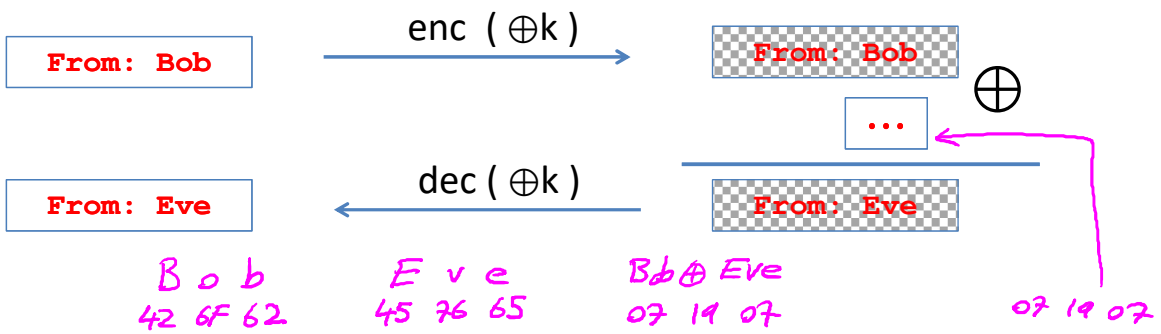


Modifications to ciphertext are undetected and have **predictable** impact on plaintext

Dan Boneh

145

Attack 2: no integrity (OTP is malleable)



Modifications to ciphertext are undetected and have predictable impact on plaintext

Dan Boneh

End of Segment

147

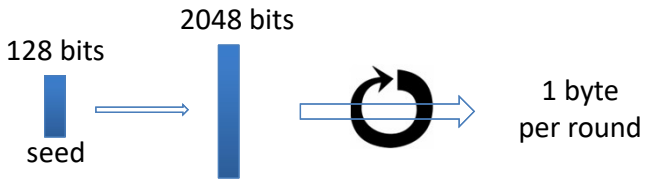
Online Cryptography Course — Dan Boneh

Stream ciphers

Real-world Stream Ciphers

148

Old example (software): RC4 (1987)



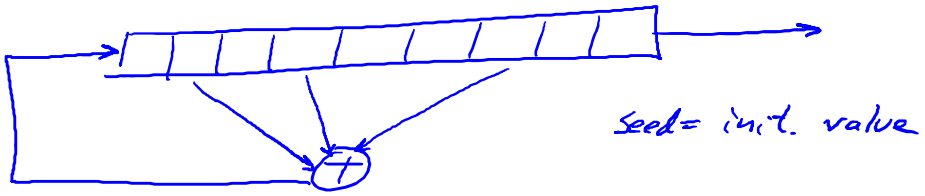
- Used in HTTPS and WEP
- Weaknesses:
  1. Bias in initial output: $\Pr[\ 2^{nd} \text{ byte} = 0\ ] = 2/256$
  2. Prob. of (0,0) is $1/256^2 + 1/256^3$
  3. Related key attacks

Dan Boneh

149

Old example (hardware): CSS (badly broken)

Linear feedback shift register (LFSR):



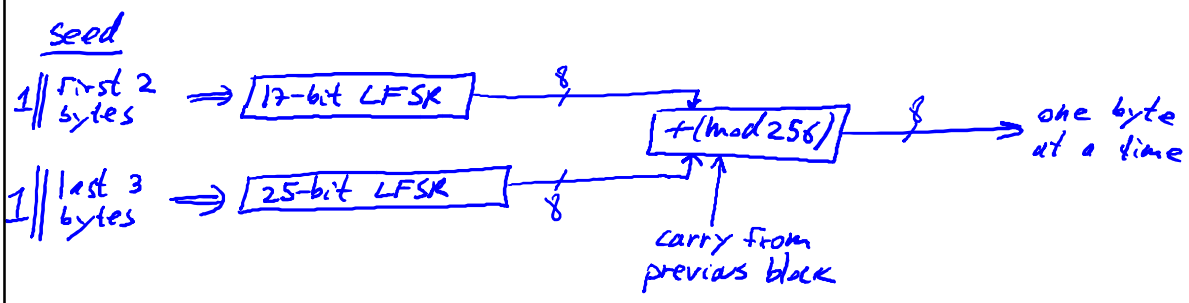
DVD encryption (CSS): 2 LFSRs
GSM encryption (A5/1,2): 3 LFSRs
Bluetooth (E0): 4 LFSRs

} all broken

Dan Boneh

Old example (hardware): CSS (badly broken)

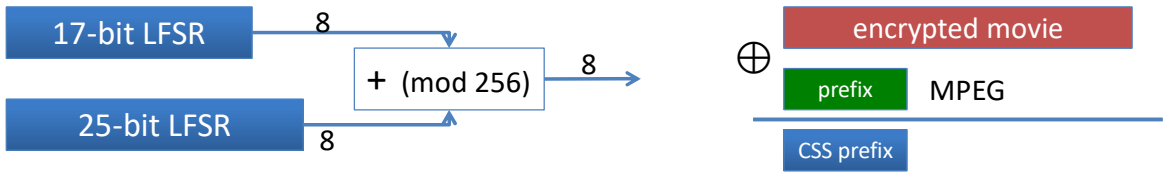
CSS: seed = 5 bytes = 40 bits



Easy to break in time $\simeq 2^{17}$

Dan Boneh

Cryptanalysis of CSS (2^{17} time attack)



For all possible initial settings of 17-bit LFSR do:

- Run 17-bit LFSR to get **20 bytes of output (160 bits)**
- **Subtract from CSS prefix ⇒ candidate 20 bytes output of 25-bit LFSR**
- If consistent with 25-bit LFSR, found correct initial settings of both !!

Using key, generate entire CSS output

Dan Boneh

Weakness #1: LFSR Cipher

- Brainless:
  - 2^{40} isn't really very big – just brainlessly brute-force the keys
- With 6 Output Bytes:
  - Guess the initial state of LFSR-17.
  - Clock out 4 bytes.
  - Use those 4 bytes to determine the corresponding 4 bytes of output from LFSR-25.
  - Use the LFSR-25 output to determine LFSR-25's state.
  - Clock out 2 bytes on both LFSRs.
  - Verify these two bytes. Celebrate or guess again.
  - This is a 2^{16} attack.

Dan Boneh

153

Weakness #1: LFSR Cipher (cont)

- With 5 Output Bytes:
- Guess the initial state of LFSR-17
- Clock out 3 bytes
- Determine the corresponding output bytes from LFSR-25
- This reveals all but the highest-order bit of LFSR-25
- Try both possibilities:
  - Clock back 3 bytes
  - Select the setting where bit 4 is 1 (remember this is the initial case).
  - It is possible that both satisfy this – try both.
- Verify as before
- This is a 2^{25} attack

Dan Boneh

154

Weakness #2: Mangled Output

(You might want to refer to the key decryption slide)

- With Known ciphertext and plainttext
  - Guess L_{k4}
  - Work backward and verify input byte
  - This is a 2^8 attack.
  - Repeat for all 5 bytes – this gives you the 5 bytes of known output for prior weakness.

Dan Boneh

Modern stream ciphers: eStream

PRG: $\{0,1\}^s \times R \longrightarrow \{0,1\}^n$

seed, nonce

Nonce: a non-repeating value for a given key.

$E(k, m ; r) = m \oplus PRG(k ; r)$

The pair (k,r) is never used more than once.

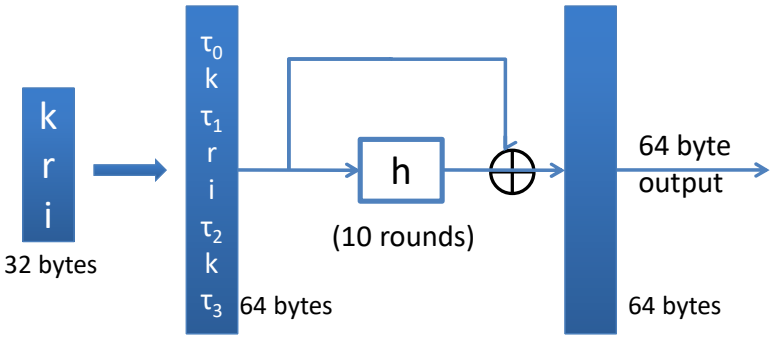
Dan Boneh

eStream: Salsa 20 (SW+HW)

Salsa20: $\{0,1\}^{128 \text{ or } 256} \times \{0,1\}^{64} \longrightarrow \{0,1\}^{n}$ (max $n = 2^{73}$ bits)

nonce

Salsa20(k ; r) := H(k , (r, 0)) ll H(k , (r, 1)) ll …



h: invertible function. designed to be fast on x86 (SSE2)

Dan Boneh

157

Is Salsa20 secure (unpredictable) ?

- Unknown: no known **provably** secure PRGs
- In reality: no known attacks better than exhaustive search

Dan Boneh

158

Performance:

Crypto++ 5.6.0 [Wei Dai]

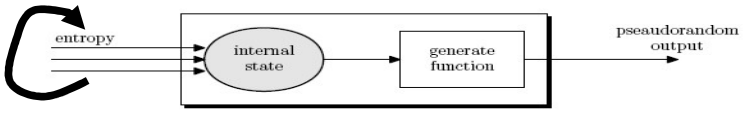
AMD Opteron, 2.2 GHz (Linux)

	PRG	Speed (MB/sec)
	RC4	126
eStream	Salsa20/12	643
eStream	Sosemanuk	727

Dan Boneh

159

Generating Randomness (e.g. keys, IV)



Pseudo random generators in practice: (e.g. /dev/random)

- Continuously add entropy to internal state
- Entropy sources:
  - Hardware RNG: Intel **RdRand** inst. (Ivy Bridge). 3Gb/sec.
  - Timing: hardware interrupts (keyboard, mouse)

NIST SP 800-90: NIST approved generators

Dan Boneh

End of Segment

Online Cryptography Course — Dan Boneh

Stream ciphers

PRG Security Defs

162

- **NIST Special Publication 800-22:**
- A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications

https://www.itread01.com/content/1548685474.html

https://www.sohu.com/a/256079825_468736

Dan Boneh

- **In theory:**
  - Randomness: Entropy
    - Shannon → $H(x) = -\sum p(x) \log_2(p(x))$
    - Min-Entropy → $H(x) = -\log_2 \max(p_i(x))$

Unpredictability: Conditionnal Entropy

x is a bit vector

- Stochastic process
  - H(state_{i} | state_{i-1})?

- **In practice**

- Assesment Methods
  - Statistical tests
  - Build a stochastic model

Dan Boneh

Statistical tests

- **Off-line tests**
  - FIPS140-2
  - NIST SP800-22
  - DIEHARD
  - AIS31
- **On-line tests**
  - To detect in real time
    - failures abnormal behaviour
    - Health tests
  - partial statistical tests

Dan Boneh

FIPS140-2 tests

- **4 tests applied on 20000-bit sequences**
- **Can be embedded in the circuit:**
  - Monobit (pr(bit=0))
  - Poker : uniform distribution of 4-bit groups
  - Runs : check the number of run sequences of '0' and '1' of length between 1 and 6 0111 1110 7E 1111111100000000
  - Long runs : no run of '0' or '1' with a length equal or greater than 26 should occur

NIST FIPS (Federal Information Processing Standards). Security Requirements for Cryptographic Modules publication 140-2, May 25 2001. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

Dan Boneh

166

NIST test suite SP800-22

- **NIST test suite (SP800-22)**
  - Among them:
    - Similar to what was first included in FIPS140-2 +
    - Random Binary Matrix Rank : check for linear dependency among fixed substrings
    - Check for periodicity (DFT, template matching tests (2 variants))
    - Compression test (Maurer's "universal statistical" test, similar to a Lempel-Ziv compression, LFSR)
    - Random walk (Cumulative sum, random excursion tests (2 variants))

Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications, april 2010.

Dan Boneh

167

- Poker Test[2,3]
  - Checks the frequency of m-bit non-overlapping sequences
    - m is a parameter; for FIPS 140-1 m = 4
- Runs Test[2,3]
  - Checks the frequency of runs of ones or zeroes of various lengths
    - The length of the runs is a parameter; for FIPS 140-1, $1 \leq i \leq 6$, with runs greater than 6 counted same as 6; zeroes and ones counted separately
- Long Run Test[2]
  - Reports the longest run of ones or zeroes in the sample block
- Autocorrelation Test[3]
  - Compares bits stream to shifted version of itself
    - Number of shifts is a parameter
  - Detects bias and most correlations, if several shifts are considered
    - Can also detect sine-wave interference

Dan Boneh

168

- Block Test
  - NIST FIPS 140-3 (paragraph 4.9.2) requires continuously comparing adjacent blocks of bits[1] and, if identical, rejecting the 2nd block
  - FIPS 140-3 also requires an "Entropy Source Test"

- Frequency Test (a.k.a.: Monobit Test)[2,3]
  - Count "ones" as a percentage of total bits in a block
  - Variations:
    - Sliding window vs. non-overlapping blocks (FIR filter)
    - Exponentially decaying response window (IIR filter)
  - Measures DC bias directly

- Serial Test (a.k.a. Two-bit Test)[3]
  - Checks the frequency of two-bit overlapping sequences
  - Tests for bias and some correlations

Dan Boneh

169

AIS-31

Bundesamt für Sicherheit in der Informationstechnik

- **Strongly common criteria oriented**
- **New approach: testing TRNG output as well as the randomness source**
  - 9 statistical tests
  - Recommends using on-line tests,
    - Entropy source failure tests
    - Statistical defect tests on the raw random sequence
    - Statistical defect tests on the post-processed sequence
  - Defines TRNG classes (PTG.x)

Dan Boneh

170

Let $G: K \longrightarrow \{0,1\}^n$ be a PRG

<u>Goal</u>: define what it means that

$[k \xleftarrow{R} K,$ output $G(k)]$

is "indistinguishable" from

$[r \xleftarrow{R} \{0,1\}^n,$ output $r]$

Dan Boneh

171

Statistical Tests

Statistical test on $\{0,1\}^n$:

not random — random

an alg. A s.t. A(x) outputs "0" or "1"

Examples:

(1) $A(x)=1$ iff $\left| \#0(x) - \#1(x) \right| \leq 10 \cdot \sqrt{n}$

(2) $A(x)=1$ iff $\left| \#00(x) - \frac{n}{4} \right| \leq 10 \cdot \sqrt{n}$

Dan Boneh

Statistical Tests

More examples:

(3) $A(x)=1$ iff max-run-of-0$(x) < 10 \cdot \log_2(n)$

Advantage

Let $G: K \longrightarrow \{0,1\}^n$ be a PRG and A a stat. test on $\{0,1\}^n$

Define:

$$Adv_{PRG}[A,G] = \left| \Pr_{K \xleftarrow{R} K}[A(G(K))=1] - \Pr_{r \xleftarrow{R} \{0,1\}^n}[A(r)=1] \right| \in [0,1]$$

Adv close to 1 $\Rightarrow$ A can dist. G from random

Adv close to 0 $\Rightarrow$ A cannot

A silly example: A(x) = 0 $\Rightarrow$ $Adv_{PRG}[A,G]$ = 

Dan Boneh

Suppose $G:K \longrightarrow \{0,1\}^n$ satisfies **msb(G(k)) = 1** for 2/3 of keys in K

Define stat. test $A(x)$ as:

if [msb(x)=1] output "1" else output "0"

Then

$$Adv_{PRG}[A,G] = \left| \overbrace{Pr[A(G(k))=1]}^{2/3} - \overbrace{Pr[A(r)=1]}^{1/2} \right| =$$

Dan Boneh

175

Secure PRGs: crypto definition

Def: We say that $G:K \longrightarrow \{0,1\}^n$ is a **secure PRG** if

$\forall$ "eff" stat. tests A:

$Adv_{PRG}[A,G]$ is "negligible"

Are there provably secure PRGs?

but we have heuristic candidates.

Dan Boneh

Easy fact: a secure PRG is unpredictable

We show: PRG predictable ⇒ PRG is insecure

Suppose A is an efficient algorithm s.t.

$$\Pr_{k \xleftarrow{R} \mathcal{K}}\left[A(G(k)|_{1,\ldots,i}) = G(k)|_{i+1} \right] > \frac{1}{2} + \varepsilon$$

for non-negligible ε (e.g. ε = 1/1000)

Dan Boneh

Easy fact: a secure PRG is unpredictable

Define statistical test B as:

$B(x) =$ if $A(x|_{1,\ldots,i}) = x_{i+1}$ output 1; else output 0

$r \xleftarrow{R} \{0,1\}^n$: $Pr[B(r)=1] = \frac{1}{2}$

$r \xleftarrow{R} \mathcal{K}$: $Pr[B(G(k))=1] > \frac{1}{2} + \varepsilon$

$\Rightarrow Adv_{PRG}[B,G] = \left| Pr[B(r)=1] - Pr[B(G(k))=1] \right| > \varepsilon$

Dan Boneh

Thm (Yao'82): an unpredictable PRG is secure

Let $G:K \longrightarrow \{0,1\}^n$ be PRG

"Thm": if $\forall i \in \{0, \ldots, n-1\}$ PRG G is unpredictable at pos. i
then G is a secure PRG.

If next-bit predictors cannot distinguish G from random
then no statistical test can !!

Dan Boneh

Let $G:K \longrightarrow \{0,1\}^n$ be a PRG such that

from the last n/2 bits of G(k)

it is easy to compute the first n/2 bits.

Is G predictable for some $i \in \{0, \ldots, n-1\}$?

- Yes ⟸
- No

180

More Generally

Let P_1 and P_2 be two distributions over $\{0,1\}^n$

Def: We say that P_1 and P_2 are

computationally indistinguishable (denoted $P_1 \approx_p P_2$)

if $\forall$ "eff" stat. tests A

$$\left| \Pr_{x \leftarrow P_1}[A(x)=1] - \Pr_{x \leftarrow P_2}[A(x)=1] \right| < \text{negligible}$$

Example: a PRG is secure if $\{ k \xleftarrow{R} K : G(k) \} \approx_p \text{uniform}(\{0,1\}^n)$

Dan Boneh

End of Segment

182

Online Cryptography Course — Dan Boneh

Stream ciphers

Semantic security

Goal: secure PRG ⇒ "secure" stream cipher

What is a secure cipher?

Attacker's abilities: **obtains one ciphertext** (for now)

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k, m_1) = m$

attempt #2: **attacker cannot recover all of plaintext**

$E(k, m_0 \| m_1) = m_0 \| m_1 \oplus k$

Recall Shannon's idea:

CT should reveal no "info" about PT

Dan Boneh

Recall Shannon's perfect secrecy

Let (E,D) be a cipher over (K,M,C)

(E,D) has perfect secrecy if $\forall\, m_0, m_1 \in M \quad (\ |m_0| = |m_1|\ )$

$\{ E(k,m_0) \} \quad = \quad \{ E(k,m_1) \}$ where $k \leftarrow\!\!- K$

(E,D) has perfect secrecy if $\forall\, m_0, m_1 \in M \quad (\ |m_0| = |m_1|\ )$

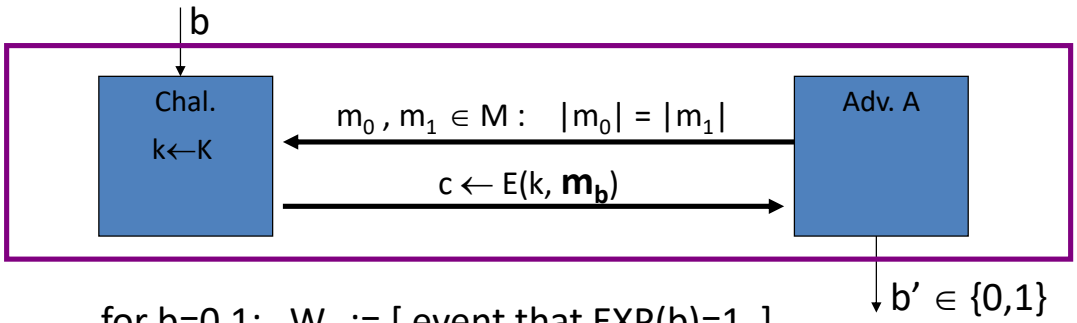
$\{ E(k,m_0) \} \approx_p \{ E(k,m_1) \}$ where $k \leftarrow\!\!- K$

… but also need adversary to exhibit $m_0, m_1 \in M$ explicitly

Dan Boneh

Semantic Security (one-time key)

For b=0,1 define experiments EXP(0) and EXP(1) as:



for b=0,1: W_b := [event that EXP(b)=1]

$Adv_{SS}[A,E] := \left| Pr[W_0] - Pr[W_1] \right| \in [0,1]$

Dan Boneh

186

Semantic Security (one-time key)

Def: $\mathbb{E}$ is **semantically secure** if for all efficient A

$Adv_{SS}[A,\mathbb{E}]$ is negligible.

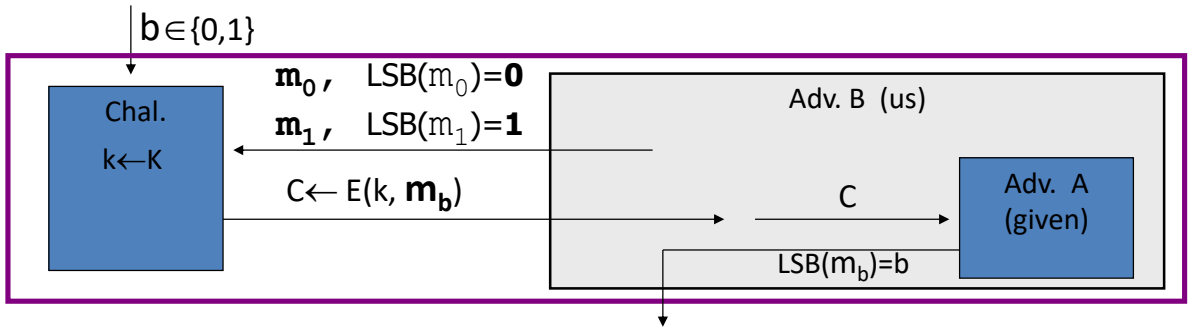
$\Rightarrow$ for all explicit $m_0, m_1 \in M$: $\{ E(k,m_0) \} \approx_p \{ E(k,m_1) \}$

Dan Boneh

Examples

Suppose efficient A can always deduce LSB of PT from CT.

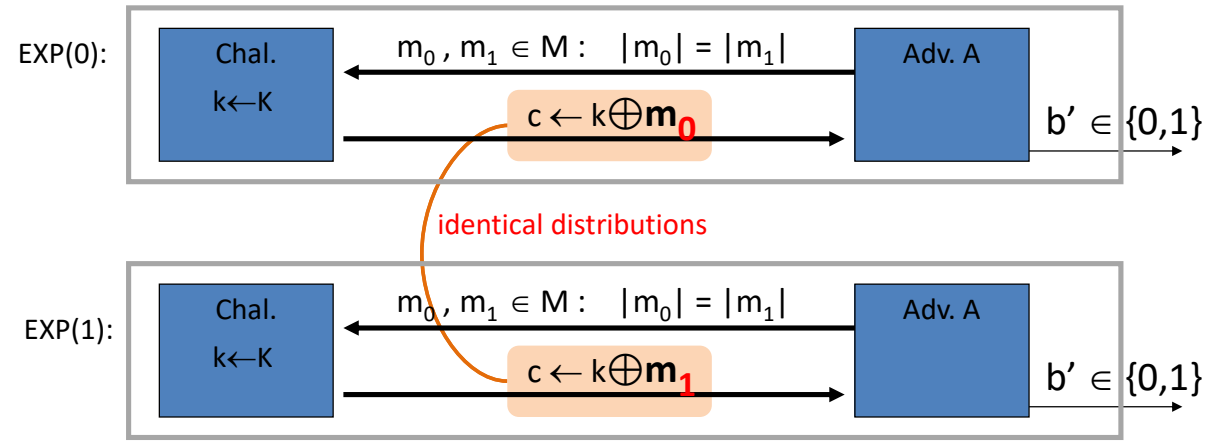
⇒ $\mathbb{E}$ = (E,D) is not semantically secure.



Then $Adv_{SS}[B, \mathbb{E}]$ = | Pr[**EXP(0)**=1] − Pr[**EXP(1)**=1] | =

Boneh

OTP is semantically secure



For **all** A: $Adv_{SS}[A,OTP] = \left| Pr[\, \mathbf{A}(k \oplus \mathbf{m_0}) = 1 \,] - Pr[\, \mathbf{A}(k \oplus \mathbf{m_1}) = 1 \,] \right|$

Dan Boneh

189

End of Segment

190

Stream ciphers

Stream ciphers are semantically secure

Goal: secure PRG ⇒ semantically secure stream cipher

Stream ciphers are semantically secure

Thm: $G:K \longrightarrow \{0,1\}^n$ is a secure PRG $\Rightarrow$

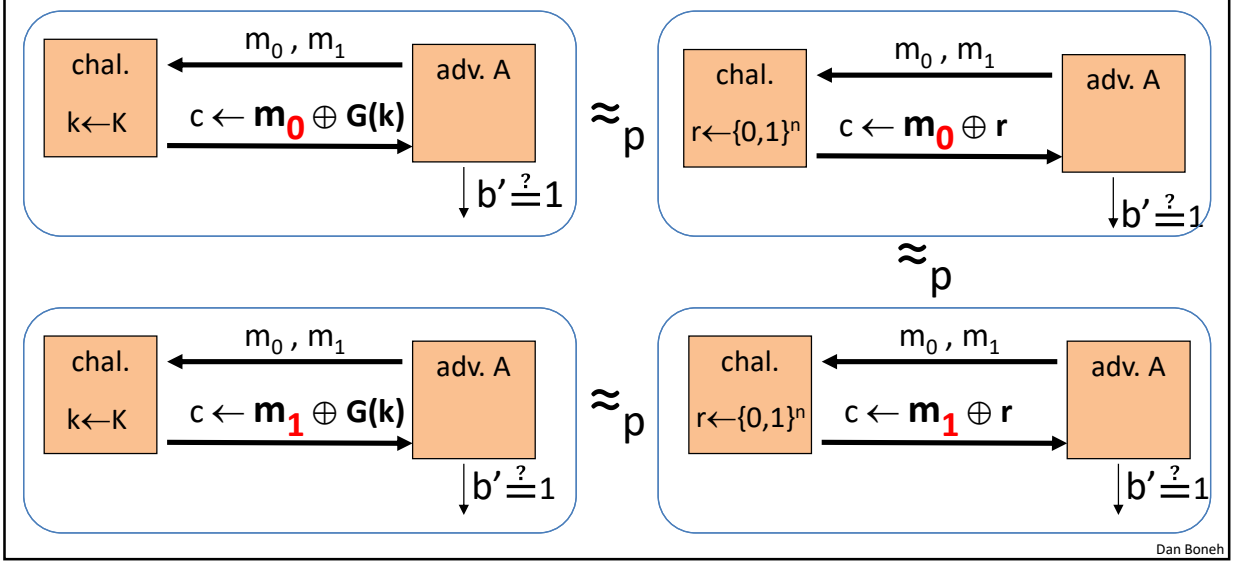
stream cipher E derived from G is sem. sec.

$\forall$ sem. sec. adversary A , $\exists$a PRG adversary B s.t.

$$Adv_{SS}[A,E] \leq 2 \cdot Adv_{PRG}[B,G]$$

Dan Boneh

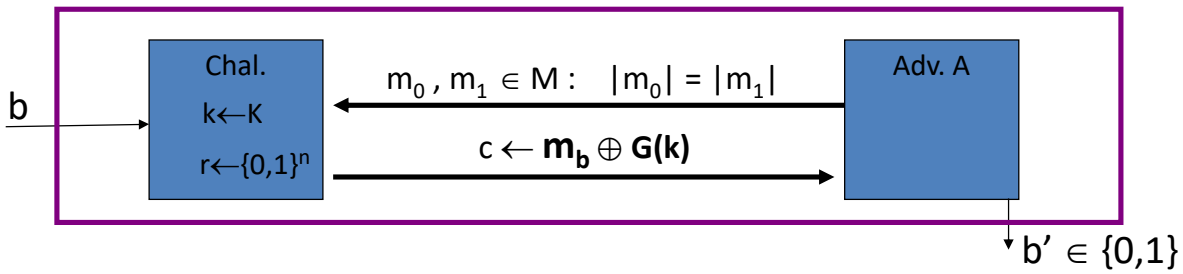
Proof: intuition



Dan Boneh

193

Proof: Let A be a sem. sec. adversary.

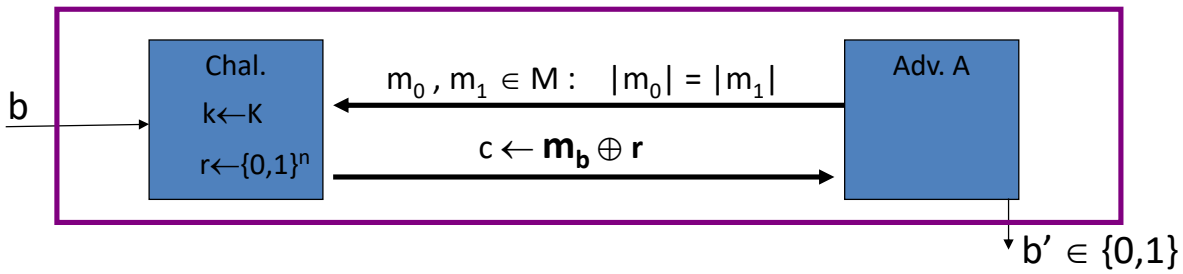


For b=0,1: W_b := [event that b'=1].

$$Adv_{SS}[A,E] = \left| Pr[W_0] - Pr[W_1] \right|$$

Dan Boneh

Proof: Let A be a sem. sec. adversary.



For b=0,1: W_b := [event that b'=1].

$$Adv_{SS}[A,E] = \left| Pr[W_0] - Pr[W_1] \right|$$

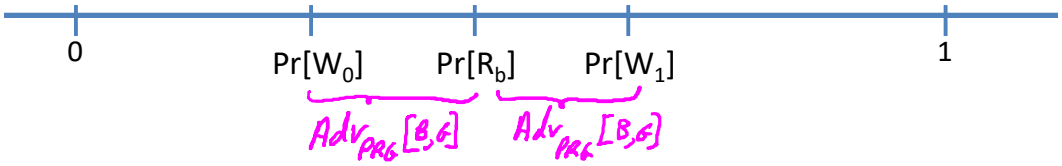
For b=0,1: R_b := [event that b'=1]

Dan Boneh

Proof: Let A be a sem. sec. adversary.

Claim 1: $\left|\Pr[R_0] - \Pr[R_1]\right| = Adv_{SS}[A, OTP] = 0$

Claim 2: $\exists B$: $\left|\Pr[W_b] - \Pr[R_b]\right| = Adv_{PRG}[B, G]$ For $b=0,1$



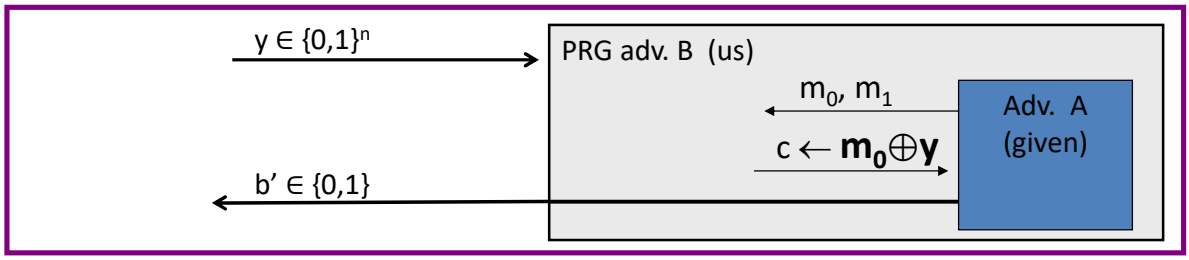
$\Rightarrow$ $Adv_{SS}[A,E] = \left|\Pr[W_0] - \Pr[W_1]\right| \leq 2 \cdot Adv_{PRG}[B,G]$

Dan Boneh

196

Proof of claim 2: $\exists B: \left|\Pr[W_0] - \Pr[R_0]\right| = \mathrm{Adv}_{PRG}[B,G]$

Algorithm B:



$$\mathrm{Adv}_{PRG}[B,G] = \left| \Pr_{r \xleftarrow{R} \{0,1\}^n}[B(r)=1] - \Pr_{K \xleftarrow{R} \mathcal{K}}[B(G(k))=1] \right| = \left|\Pr[R_0] - \Pr[W_0]\right|$$

Dan Boneh

197

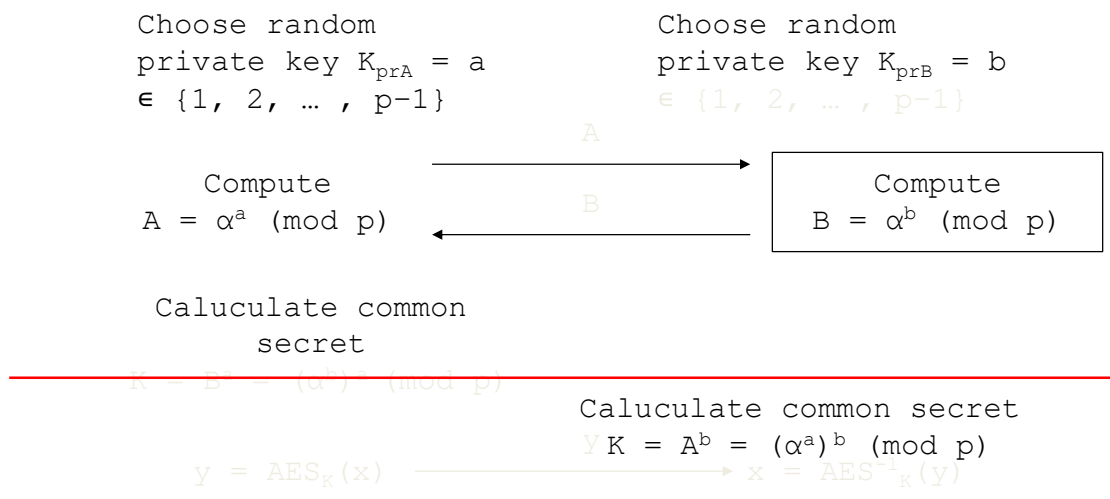
End of Segment

198

Diffie-Hellman Key Exchange

- Set-up
  - Choose a large prime p
  - Choose an integer $\alpha \in \{2, 3, \ldots, p-2\}$
  - Publish p and α

Diffie-Hellman Key Exchange



200

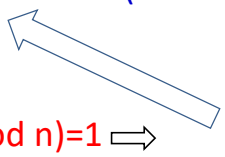
Dan Boneh

An asymmetric ciphers (RSA) Example

Choose **p = 3** and **q = 11** (secret)
Compute n = p * q = 3 * 11 = 33
Compute ϕ(n) = (p - 1) * (q - 1) = 2 * 10 = 20
Choose e such that 1 < e < ϕ(n) and e and n are coprime.
Let e = 7
Compute a value for d such that (d * e) mod ϕ(n) = 1.
One solution is **d = 3** [(3 * 7) mod 20 = 1]
Public key is (e, n) => (7, 33)
Private key is (d, n) => (3, 33)
The encryption of $m = 2$ is $c = 2^7 \mod 33 = 29$
The decryption of $c = 29$ is $m = 29^3 \mod 33 = 2$

Proof for the RSA Algorithm

- $C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+k\phi(n)} \equiv M^1 M^{k\phi(n)} \pmod n$
- $\equiv M^1 \mathbf{M^{\phi(n)\,k}} \pmod n$



- By Euler's theorem $M^{\phi(n)} \pmod n = 1$ ⇨
- $ed \equiv 1 \pmod{\phi(n)}$ $ed = 1 + k\phi(n)$

- example
- p=885320963, q=238855417,
- n=p · q=211463707796206571
- Let e=9007, ∴ d=116402471153538991
- M="cat"=30120, C=113535859035722866