

# Klassische Kryptographie

# Kryptoanalyse

Torsten Lachmann

`tlachman@cs.uni-potsdam.de`

# Kryptoanalyse

- Kryptoanalyse  $\implies$  Verfahren zur Codeentschlüsselung
- Verschlüsselungsverfahren ist dem Angreifer bekannt (nach Kerckhoff)
- Angreifer versucht den verwendeten Schlüssel aufzudecken

Sender  $\rightarrow$  Verschlüsselung  $\rightarrow_{\text{Kanal}}$  Entschlüsselung  $\rightarrow$  Empfänger

$\downarrow$   
Gegner

# Angriffsszenarien

- Brute Force Attack (Rohe-Gewalt-Angriff)  
testet alle möglichen Schlüssel
- ciphertext only (Schlüsseltextangriff)  
Teilstücke des Geheimtextes sind bekannt
- known plaintext (Bekannte-Quelle-Angriff)  
Klartextstücke & zugehörige Geheimtextstücke bekannt
- chosen plaintext (Ausgewählte-Quelle-Angriff)  
temporärer Zugang zum Verschlüsselungssystem
- chosen ciphertext (Ausgewählte-Schlüsseltext-Angriff)  
temporärer Zugang zum Entschlüsselungssystem

Ziel ist Erlangen des verwendeten Schlüssels

# Brute Force Attack

- ziemlich plumpe, aber sehr gefährliche Attacke
- Verschlüsselungsverfahren wird mit nackter Gewalt angegriffen
- alle möglichen Schlüssel werden ausprobiert
- benötigt sehr hohe Rechenleistung, was heutzutage kein Problem mehr ist
- moderne Verfahren sind durch Schlüssellänge  $\geq 128$  Bit abgesichert

# Known Ciphertext Attack

- Gegner steht Stück Geheimtext zur Verfügung, was er attackiert
- Dieser Geheimtext wird auf Muster untersucht
- Vigenère-Chiffrierung lässt sich so knacken
- gegen moderne/sichere Verfahren ist diese Attacke wirkungslos

# Known Plaintext Attack

- Gegner ist zusammengehöriges Paar Klartext/Geheimtext bekannt
- es wird versucht den verwendeten Schlüssel aus dem Paar abzuleiten
- je mehr zusammengehörige Textstücke, desto einfacher

# Chosen Plaintext Attack

- Gegner besitzt ein Verfahren mit integriertem (aber unbekanntem) Schlüssel
- er kann selbst Texte verschlüsseln und dabei Rückschlüsse auf den verwendeten Schlüssel ziehen
- Verfahren, die diesem Angriff widerstehen, gelten als sehr sicher

# Chosen Ciphertext Attack

- ähnlich wie die chosen plaintext attack
- es stehen nur Geheimtexte zur Verfügung
- bei public-key-Systemen sinnvoll

# Fazit

- es gibt noch andere Attacken, z.B. Playback Attack
- Verfahren gilt als sicher, wenn keine der vorgestellten Attacken funktioniert
- gibt es eine Attacke, die leichter ist als Brute Force, so gilt das Verfahren (zumindest theoretisch) als gebrochen oder geknackt
- ein Verfahren kann erst nach vielen Jahren erfolgloser Kryptoanalyse als sicher angesehen werden
- es besteht immer ein Restrisiko, da es neue, schnellere Analyse-Verfahren geben kann
- ständige Vervielfachung der Rechenleistung durch bessere Chips und Cluster Computing
- einige Angriffe sind in der Praxis nicht umgesetzt worden

# Verfahren zur Entschlüsselung

Cryptoanalysis of the ..

- **Affine Cipher**  
benutzt statistische Eigenschaften der Sprache
- **Substitution Cipher**  
benutzt häufig vorkommende Buchstabengruppen
- **Vigenere Cipher**  
untersucht Abstände zwischen Wiederholungen
- **Hill Cipher**  
effektiv bei bekannten Klar- &Geheimtextpaaren
- **LFSR Stream Cipher**  
berechnet Schlüssel aus bekannten Klar- &Geheimtextpaaren  
bei bekannter Textlänge

# Buchstabenhäufigkeit

$\Sigma$	%	$\Sigma$	%	$\Sigma$	%
E	14.7004	L	2.9312	V	0.7350
N	8.8351	C	2.6733	Ü	0.5799
R	6.8577	G	2.6672	P	0.4992
I	6.3770	M	2.1336	Ä	0.4907
S	5.3881	O	1.7717	Ö	0.2547
T	4.7310	B	1.5972	J	0.1645
D	4.3854	Z	1.4225	Y	0.0173
H	4.3554	W	1.4201	Q	0.0142
A	4.3309	F	1.3598	X	0.0129
U	3.1877	K	0.9558		

[Buchstabenhäufigkeit: Fischer Lexikon, Technik, Bd.4]

# Cryptoanalysis of the Affine Cipher

- Angreifer besitzt Teilstück vom verschlüsselten Text  
⇒ Known Ciphertext Attack
- Vorkommen der einzelnen Buchstaben zählen
- Vorkommen mit der normalen Buchstabenhäufigkeit vergleichen

Aus der Kryptofunktion  $y = (ax + b)_{mod26}$   
erhält man durch Umstellung  $x = a^{-1}(y - b)_{mod26}$

# Beispiel zum Knacken der Affine Cipher

- Ciphertext:

UXSGSDTFQCFSQFJDQHYESXYFJHQDS

- kleines selbstgeschriebenes Java-Programm

RelAbsH.java

- Ausgabe der Buchstabenhäufigkeit des Ciphertextes

```

import java.io.*;
class RelAbsH {
    public static void main( String args []) throws Exception {
        int [] charAnz = new int [26];
        File input = new File(args [0]);
        FileReader in = new FileReader (input);
        int c, sum=0;
        while((c = in.read()) != -1) {
            if((((char) c) >= 'A' && ((char) c) <= 'Z') ||
                (((char) c) >= 'a' && ((char) c) <= 'z')) {
                if((((char) c) >= 'A' && ((char) c) <= 'Z'){
                    charAnz[c-65] += 1;
                }
                sum += 1;
            }
            if((((char) c) >= 'a' && ((char) c) <= 'z'){
                charAnz[c-97] += 1;
                sum += 1;
            }
        }
    }
    System.out.println();
    System.out.println("Anzahl der Buchstaben im Text von " + args[0] + " : " + sum);
    System.out.println();
    for(c=0;c<=25;c++){
        System.out.println("Das " + ((char) (c+65)) +
            " kommt " + charAnz[c] + " " + (char) 9 + "mal vor, das sind " +
            Math.round((100*charAnz[c]/(double) sum)) + " % " + (char) 9 + "( " +
            (float)((100*charAnz[c])/(double) sum) + " %" + (char) 9 + " )");
    }
}
}

```

```
shell>java RelAbsH ciphertext.txt
```

```
Anzahl der Buchstaben im Text von ciphertext.txt : 29
```

```
Das A kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das B kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das C kommt 1 mal vor, das sind 3 % ( 3.4482758 % )
Das D kommt 3 mal vor, das sind 10 % ( 10.344828 % )
Das E kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das F kommt 4 mal vor, das sind 14 % ( 13.793103 % )
Das G kommt 2 mal vor, das sind 7 % ( 6.8965516 % )
Das H kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das I kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das J kommt 2 mal vor, das sind 7 % ( 6.8965516 % )
Das K kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das L kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das M kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das N kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das O kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das P kommt 2 mal vor, das sind 7 % ( 6.8965516 % )
Das Q kommt 4 mal vor, das sind 14 % ( 13.793103 % )
Das R kommt 1 mal vor, das sind 3 % ( 3.4482758 % )
Das S kommt 5 mal vor, das sind 17 % ( 17.241379 % )
Das T kommt 1 mal vor, das sind 3 % ( 3.4482758 % )
Das U kommt 1 mal vor, das sind 3 % ( 3.4482758 % )
Das V kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das W kommt 0 mal vor, das sind 0 % ( 0.0 % )
Das X kommt 2 mal vor, das sind 7 % ( 6.8965516 % )
Das Y kommt 1 mal vor, das sind 3 % ( 3.4482758 % )
Das Z kommt 0 mal vor, das sind 0 % ( 0.0 % )
```

# Beispiel zum Knacken der Affine Cipher

- Text enthält 57 Buchstaben
- häufigste Vorkommen S (5 mal), F,Q (4 mal) und D (3 mal)
- Zuordnung zwei verschiedener Zeichen  $a \rightarrow b$  und  $c \rightarrow d$  liefert i.a. den Schlüssel  $K = (s,t)$

- Vermutung S entspricht e & F entspricht n

$$\text{Test: } ggT(18 - 5, 26) = 13 \Rightarrow \text{nicht möglich}$$

- Vermutung F entspricht e & Q entspricht n

$$\text{Test: } ggT(16 - 5, 26) = 1 \Rightarrow \text{möglich}$$

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13
O	P	Q	R	S	T	U	V	W	X	Y	Z		
14	15	16	17	18	19	20	21	22	23	24	25		

# Beispiel zum Knacken der Affine Cipher

- $s = (a - c)^{-1} * (b - d)_{mod26}$
- $t = b - (a * s)_{mod26} = d - (c * s)_{mod26}$
- $s = (13 - 4)^{-1} * (16 - 5)_{mod26} = 9^{-1} * 11_{mod26} = 3 * 11_{mod26} = 33_{mod26} = 7$
- $t = 16 - (13 * 7)_{mod26} = 16 - (91)_{mod26} = 16 - 13 = 3$
- Schlüssel ist  $K=(7,3)$
- eingesetzt in  $x = a^{-1}(y - b)_{mod26}$  erhält man
$$x = 7^{-1}(y - 3)_{mod26}$$
also  $x = 15(y - 3)_{mod26}$

## Beispiel zum Knacken der Affine Cipher

$$y = 20(U) \Rightarrow x = 15(20 - 3)_{26} = 15 * 17_{26} = 255_{26} = 21(v)$$

$$y = 23(X) \Rightarrow x = 15(23 - 3)_{26} = 15 * 20_{26} = 300_{26} = 14(o)$$

$$y = 18(S) \Rightarrow x = 15(18 - 3)_{26} = 15 * 15_{26} = 225_{26} = 17(r)$$

$$y = 6(G) \Rightarrow x = 15(6 - 3)_{26} = 15 * 3_{26} = 45_{26} = 19(t)$$

$$y = 18(S) \Rightarrow x = 15(18 - 3)_{26} = 15 * 15_{26} = 225_{26} = 17(r)$$

$$y = 3(D) \Rightarrow x = 15(3 - 3)_{26} = 15 * 0_{26} = 0_{26} = 0(a)$$

$$y = 19(T) \Rightarrow x = 15(19 - 3)_{26} = 15 * 16_{26} = 240_{26} = 6(g)$$

$$y = 5(F) \Rightarrow x = 15(5 - 3)_{26} = 15 * 2_{26} = 30_{26} = 4(e)$$

$$y = 16(Q) \Rightarrow x = 15(16 - 3)_{26} = 15 * 13_{26} = 195_{26} = 13(n)$$

$$y = 2(C) \Rightarrow x = 15(2 - 3)_{26} = 15 * -1_{26} = -15_{26} = 11(l)$$

$$y = 5(F) \Rightarrow x = 15(5 - 3)_{26} = 15 * 2_{26} = 30_{26} = 4(e)$$

$$y = 18(S) \Rightarrow x = 15(18 - 3)_{26} = 15 * 15_{26} = 225_{26} = 17(r)$$

$$y = 16(Q) \Rightarrow x = 15(16 - 3)_{26} = 15 * 13_{26} = 195_{26} = 13(n)$$

$$y = 6(G) \Rightarrow x = 15(6 - 3)_{26} = 15 * 3_{26} = 45_{26} = 19(t)$$

$$y = 9(J) \Rightarrow x = 15(9 - 3)_{26} = 15 * 6_{26} = 90_{26} = 12(m)$$

$$y = 3(D) \Rightarrow x = 15(3 - 3)_{26} = 15 * 0_{26} = 0_{26} = 0(a)$$

$$y = 16(Q) \Rightarrow x = 15(16 - 3)_{26} = 15 * 13_{26} = 195_{26} = 13(n)$$

$$y = 7(H) \Rightarrow x = 15(7 - 3)_{26} = 15 * 4_{26} = 60_{26} = 8(i)$$

$$y = 9(J) \Rightarrow x = 15(9 - 3)_{26} = 15 * 6_{26} = 90_{26} = 12(m)$$

$$y = 4(E) \Rightarrow x = 15(4 - 3)_{26} = 15 * 1_{26} = 15_{26} = 15(p)$$

$$y = 18(S) \Rightarrow x = 15(18 - 3)_{26} = 15 * 15_{26} = 225_{26} = 17(r)$$

$$y = 23(X) \Rightarrow x = 15(23 - 3)_{26} = 15 * 20_{26} = 300_{26} = 14(o)$$

$$y = 25(Y) \Rightarrow x = 15(25 - 3)_{26} = 15 * 22_{26} = 330_{26} = 18(s)$$

$$y = 5(F) \Rightarrow x = 15(5 - 3)_{26} = 15 * 2_{26} = 30_{26} = 4(e)$$

$$y = 9(J) \Rightarrow x = 15(9 - 3)_{26} = 15 * 6_{26} = 90_{26} = 12(m)$$

$$y = 7(H) \Rightarrow x = 15(7 - 3)_{26} = 15 * 4_{26} = 60_{26} = 8(i)$$

$$y = 16(Q) \Rightarrow x = 15(16 - 3)_{26} = 15 * 13_{26} = 195_{26} = 13(n)$$

$$y = 3(D) \Rightarrow x = 15(3 - 3)_{26} = 15 * 0_{26} = 0_{26} = 0(a)$$

$$y = 18(S) \Rightarrow x = 15(18 - 3)_{26} = 15 * 15_{26} = 225_{26} = 17(r)$$

# Beispiel zum Knacken der Affine Cipher

Ergebnis:

vortragenlerntmanimproseminar

# Cryptoanalysis of the Substitution Cipher

- Angreifer besitzt Teilstück vom verschlüsselten Text  
⇒ Known Ciphertext Attack
- Vorkommen der einzelnen Buchstaben zählen
- Vorkommen mit der normalen Buchstabenhäufigkeit vergleichen
- Vorkommen von Bigrammen zählen
- Vorkommen von Trigrammen zählen

# Cryptoanalysis of the Vigenere Cipher

- Angreifer besitzt Teilstück vom verschlüsselten Text  
⇒ Known Ciphertext Attack
- in Abhängigkeit eines Schlüsselwortes werden mehrere Geheimtextalphabete verwendet
- Schlüsselwortlänge? ( $m$ )
  - Kasiski Test (Friedrich Kasiski 1863)
  - Verfahren bereits 1854 von Charles Babbage entdeckt
- 2 identische Schlüsseltextteilworte haben gleichen Klartext, falls ihr Abstand  $\beta = 0_{mod m}$  ist

# Cryptoanalysis of the Vigenere Cipher

Beispiel aus [1]

Schlüssel	?
Klartext	<b>DIELILIEDIEROSEUNDDIETULPE</b>
Geheimtext	<b>JMPMOPTFJMPSUWPVTHOJKXFMVI</b>

- "DIE" wird 2x mit JMP und 1x mit OJK verschlüsselt  
→ Schwäche
- Abstand zwischen "JMP"s = 8 → Schlüssellänge 8,4 oder 2
- Vermutung 4

# Cryptoanalysis of the Vigenere Cipher

Beispiel aus [1]

Schlüssel	????
Klartext	<b>DIELILIEDIEROSEUNDDIETULPE</b>
Geheimtext	<b>JMPMOPTFJMPSUWPVTHOJKXFMVI</b>

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13
O	P	Q	R	S	T	U	V	W	X	Y	Z		
14	15	16	17	18	19	20	21	22	23	24	25		

- D wurde mit J verschlüsselt, also G-Alphabet
- I wurde mit M verschlüsselt, also E-Alphabet
- E wurde mit P verschlüsselt, also L-Alphabet
- L wurde mit M verschlüsselt, also B-Alphabet

# Cryptoanalysis of the Vigenere Cipher

Beispiel aus [1]

Schlüssel	GELBGELBGELBGELBGELBGELBGE
Klartext	<b>DI</b> ELILIED <b>IE</b> ROSEUN <b>DD</b> IETULPE
Geheimtext	<b>J</b> MPMOPT <b>F</b> J <b>M</b> PSUWP <b>V</b> TH <b>O</b> J <b>K</b> XFM <b>V</b> I

Schlüsselwort ist GELB

# Cryptoanalysis of the Hill Cipher

- sehr schwierig mit Known Ciphertext Attack, aber viel einfacher mit Known Plaintext Attack
- Schlüssellänge ist bekannt , sonst durchlaufen lassen( 2,3,4,...)
- Angreifer ist im Besitz von Klartext/Schlüsseltext-Paare

# Cryptoanalysis of the Hill Cipher

- Klartext:  $x_j = (x_{1j}, x_{2j}, \dots, x_{mj})$
- Schlüsseltext  $y_j = (y_{1j}, y_{2j}, \dots, y_{mj})$
- für  $1 \leq j \leq m$  gilt:  $y_j = e_K(x_j)$
- gegeben seien 2  $m \times m$  Matrizen

$$X = (x_{ij})$$

$$Y = (y_{ij})$$

mit  $Y = XK$ , wobei  $K$  der Schlüssel ist

- hat  $X$  ein Inverses, dann gilt  $K = X^{-1}Y$
- hat  $X$  kein Inverses, muss man ein anderes Klartext/Schlüsseltext-Paar suchen

# Cryptoanalysis of the Hill Cipher

- sei friday mit  $m = 2$  verschlüsselt

- der Schlüsseltext ist PQCFKU

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13
O	P	Q	R	S	T	U	V	W	X	Y	Z		
14	15	16	17	18	19	20	21	22	23	24	25		

- $e_K(5, 17) = (15, 16)$ ,  $e_K(8, 3) = (2, 5)$  und  $e_K(0, 24) = (10, 20)$

- ersten beiden Paare ergeben die Matrixgleichung:

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} K$$

# Cryptoanalysis of the Hill Cipher

- inverse Matrix:  $A^{-1} = (\frac{1}{D})(A^T)$

- $\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}$

- $K = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$

# Cryptoanalysis of the LFSR Cipher

- alternativer Ansatz Strom-Chiffrierung anstatt Block-Chiffrierung
- Plaintext und Ciphertext werden binär repräsentiert
- Schlüsselstromgenerierung wird über ein LFSR (linear feedback shift register) realisiert
- ein known plaintext Angriff ist möglich

# Cryptoanalysis of the LFSR Cipher

- Schlüsseltext ergibt sich aus  $+_{mod2}$  von Klartext und Schlüssel
- $y_i = (x_i + z_i)$
- Beispiel:

$$y = 101101011110010$$

$$x = 011001111111000$$

$$z = 110100100001010$$

# Quellen

- [1] Singh, "Geheime Botschaften"
- Cryptography (Chapman,Hall), Kapitel 1.2
- [http://de.wikipedia.org/wiki/Bild:Alphabet\\_bigramm.png](http://de.wikipedia.org/wiki/Bild:Alphabet_bigramm.png)
- [http://de.wikipedia.org/wiki/Bild:Alphabet\\_trigramm.png](http://de.wikipedia.org/wiki/Bild:Alphabet_trigramm.png)