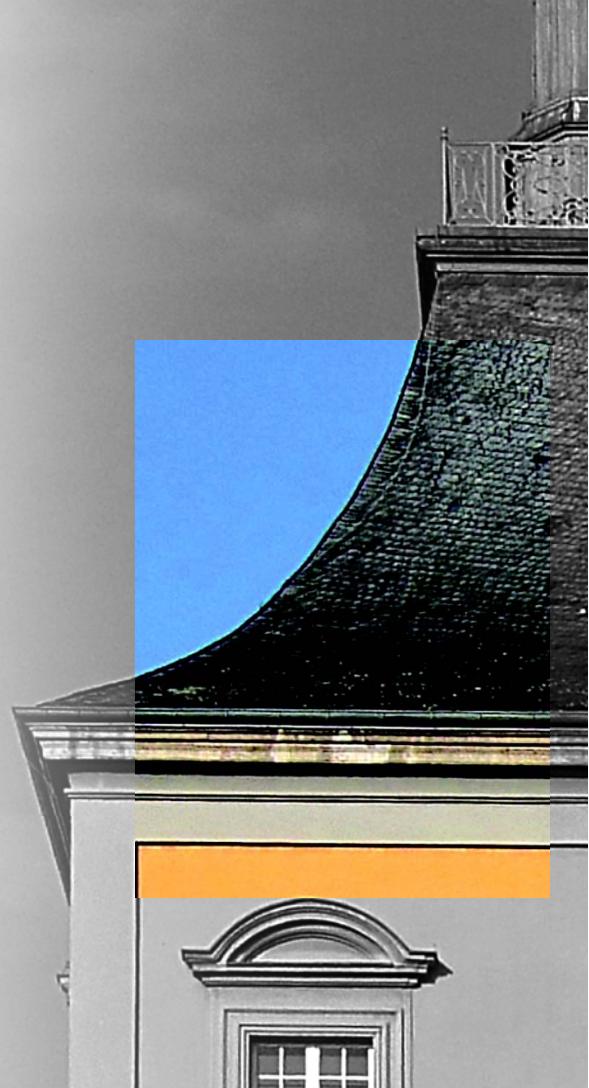


VORLESUNG NETZWERKSICHERHEIT

SOMMERSEMESTER 2021
MO. 14-16 UHR



KAPITEL 3

PUBLIC-KEY-CRYPTOGRAPHY

Motivation zur kryptographischen Absicherung der Kommunikation:

- Inhalte sind vertraulich und nur für Berechtigte entschlüsselbar
- Daten bei Übermittlung und Speicherung nicht unbemerkt veränderbar
- Sender und Empfänger verifizieren sich gegenseitig als Urheber oder Ziel
- Urheberschaft einer Nachricht nicht abstreitbar

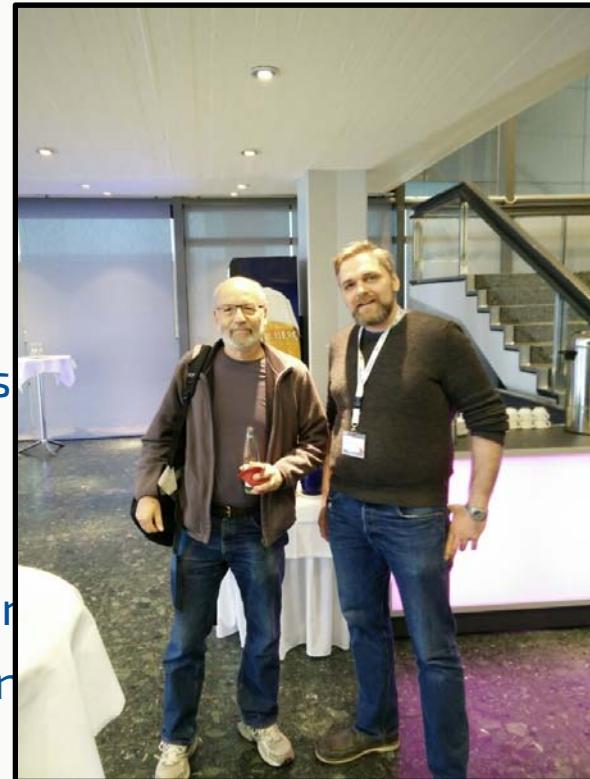


Verschlüsseln & Signieren

ACHTUNG: Nicht alle Ziele immer gleichzeitig erreichbar / gewünscht.

Asymmetrische Kryptographie

- Benötigt Schlüsselpaar
 - Öffentlicher Schlüssel
 - Privater Schlüssel
 - Öffentlicher Schlüssel von privatem Schlüssel
- Bekannte Algorithmen
 - DH (Diffie-Hellman; Schlüsseltausch)
 - ElGamal (ElGamal; Verschlüsseln & Signieren)
 - RSA (Rivest; **Shamir**; Adleman; Verschlüsseln)
- Quiz: Wer ist auf dem Foto?



Absicherung von Kommunikation

- TLS (SSL)
- GnuPG
- S/MIME

Absicherung von Softwareinstallation

- GnuPG

Hintergrund

- Public Key Cryptography Standards (PKCS)

TRANSPORT LAYER SECURITY (TLS)

TRANSPORT LAYER SECURITY (TLS)



Was meint Transport Layer Security?

- Absicherung der Transportschicht?
 - Absicherung durch darunterliegende Schichten
- Absicherung durch die Transportschicht?
 - Absicherung der darüber liegenden Schichten

Vorgänger: Secure Sockets Layer (SSL)

→ Eigene Schicht im Protokollstapel

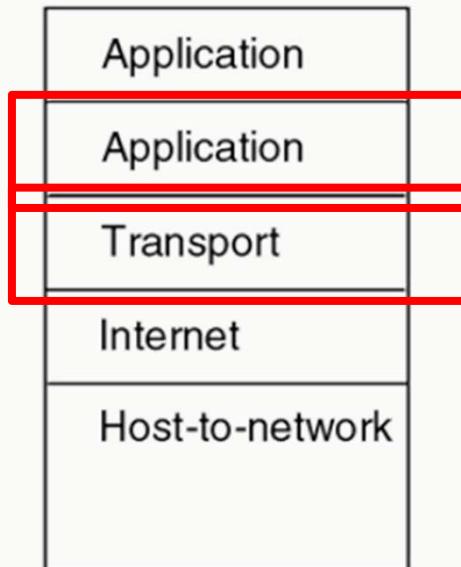
Ziel: Absicherung der Anwendungsschicht

- OSI Layer 5/6 (Sitzungs- und Darstellungsschicht)

TRANSPORT LAYER SECURITY (TLS)

TLS im TCP/IP – Protokollstapel

TCP/IP



- Betrachtung von TLS als Anwendung
- “Tunnel” von Anwendungsprotokollen durch TLS
- Bekannte Beispiele:
 - HTTP over TLS (HTTPS)
 - SMTP over TLS (SMTPS)
 - FTP over TLS (FTPS)

TRANSPORT LAYER SECURITY (TLS)

Historie

- 1994 SSLv1 (Netscape)
- 1995 SSLv2 (Netscape)
- 1996 SSLv3 (Netscape / Microsoft)
- 1999 TLSv1 (IETF Standard: RFC 2246)
- 2006 TLSv1.1 (RFC 4346)
- 2008 TLSv1.2 (RFC 5246)
- 2018 TLSv1.3 (RFC 8446)

TRANSPORT LAYER SECURITY (TLS)

Historie

- 1994 SSLv1 (Netscape)
- 1995 SSLv2 (Netscape)
- 1996 SSLv3 (Netscape / Microsoft)
- 1999 **TLSv1 (IETF Standard: RFC 2246)**
- 2006 TLSv1.1 (RFC 4346)
- 2008 TLSv1.2 (RFC 5246)
- 2018 TLSv1.3 (RFC 8446)

TLSv1 Updates:

- RFC 2712
- RFC 2817
- RFC 2818
- RFC 3268
- RFC 3546
 - Erweiterungen (z.B. SNI)
- RFC 5746
- RFC 6176
 - Prohibiting SSLv2
- RFC 7465
- RFC 7507
- RFC 7919

TRANSPORT LAYER SECURITY (TLS)

Historie

- 1994 SSLv1 (Netscape)
- 1995 SSLv2 (Netscape)
- 1996 SSLv3 (Netscape / Microsoft)
- 1999 TLSv1 (IETF Standard: RFC 2246)
- 2006 **TLSv1.1 (RFC 4346)**
- 2008 TLSv1.2 (RFC 5246)
- 2018 TLSv1.3 (RFC 8446)

TLSv1.1 Updates:

- RFC 4366
- RFC 4680
- RFC 4681
- RFC 5746
- RFC 6176
 - Prohibiting SSLv2
- RFC 7465
- RFC 7507
- RFC 7919

TRANSPORT LAYER SECURITY (TLS)

Historie

- 1994 SSLv1 (Netscape)
- 1995 SSLv2 (Netscape)
- 1996 SSLv3 (Netscape / Microsoft)
- 1999 TLSv1 (IETF Standard: RFC 2246)
- 2006 TLSv1.1 (RFC 4346)
- 2008 **TLSv1.2 (RFC 5246)**
- 2018 TLSv1.3 (RFC 8446)

TLSv1.2 Updates:

- RFC 5746
- RFC 5878
- RFC 6176
 - Prohibiting SSLv2
- RFC 7465
 - Prohibiting RC4
- RFC 7507
- RFC 7568
 - Deprecating SSLv3
- RFC 7627
- RFC 7685
- RFC 7905
- RFC 7919
- **RFC 8447**

TRANSPORT LAYER SECURITY (TLS)

Historie

- 1994 SSLv1 (Netscape)
- 1995 SSLv2 (Netscape)
- 1996 SSLv3 (Netscape / Microsoft)
- 1999 TLSv1 (IETF Standard: RFC 2246)
- 2006 TLSv1.1 (RFC 4346)
- 2008 TLSv1.2 (RFC 5246)
- 2018 **TLSv1.3 (RFC 8446)**

TLSv1.3 Updates:

- Bisher keine

Aufbau

- TLS definiert **zwei** eigene Schichten
 - Kontrollschicht
 - TLS Handshake Protocol
 - TLS Cipher Spec. Protocol
 - TLS Alert Protocol
 - TLS Application Data Protocol
 - Nutzdatenschicht
 - TLS Record Protocol

TLS HANDSHAKE PROTOCOL

- Ablauf
 - Cipher Auswahl / Abstimmung
 - **ACHTUNG:** Es gibt auch NULL-Encryption
 - Schlüsselaustausch für asymmetrische Verschlüsselung
 - Serverauthentifikation
 - Clientauthentifikation



Authentifikation mittels X509v3 Zertifikat

NULL-Encryption Ciphers:

- TLS_NULL_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_SHA
- **TLS_RSA_WITH_NULL_SHA256**

TLS HANDSHAKE PROTOCOL

public key client



private key client



Client

public key server



private key server

Server



RN_c

generate random number RN_c

client_hello (crypto information, RN_c)



RN_s RN_c

generate random number RN_s

server_hello (crypto information, RN_s)



RN_s RN_c

RN_c

RN_s



Phase 1

server certificate (incl. RN_s)

demand client certificate

check server certificate

Phase 2

X509V3 (ISO/IEC 9594-8)

- ITU-T-Standard für Public-Key-Infrastrukturen
 - **ITU** = Internationale Fernmeldeunion der Vereinten Nationen
 - **ITU-T** = Standardisierungs-Einheit der ITU
 - **X** = „Data networks and open system communications“
- Spezifizierte Datentypen
 - Public-Key-Zertifikat
 - Attributzertifikat
 - Certificate Revocation List (CRL)
 - Attribute Certificate Revocation List (ACRL)

TLS RECORD PROTOCOL

Anwendungsdaten



Fragmentierung



Kompression



MAC



Verschlüsseln



Header



TLS Record Header:

- Content-Type (8 Bit)
 - Handshake
 - Alarm
 - Data
- Version (16 Bit)
- Length (16 Bit)

CRYPTO-STANDARDS

Viele Standards, die heutiges Cryptographieumfeld prägen

- ITU-T (Vereinte Nationen)
 - X509-Zertifikate
- IEEE 802
 - 802.1X – Authentifikation am Ethernet-Port
- RSA Security Inc. Public-Key-Cryptography-Standard (PKCS)
 - 15 Standards und Definitionen für Public-Key-Crypto
- Request for Comments (RFC)
 - Organisationsübergreifende Veröffentlichung von Standards (bzw. Entwürfen und Updates)

PUBLIC-KEY-CRYPTOGRAPHY-STANDARDS

- PKCS#1 – RSA public key crypto
- ~~PKCS#2 – RSA encryption of message digests~~ Merged in PKCS#1
- PKCS#3 – Diffie-Hellman key agreement
- ~~PKCS#4 – RSA key syntax~~ Merged in PKCS#1
- PKCS#5 – Password based cryptography specification
- PKCS#6 – Extended certificate syntax
- PKCS#7 – Cryptographic message syntax
- PKCS#8 – Private key information syntax

- PKCS#9 – Selected attribute types
- PKCS#10 – Certification request standard
- PKCS#11 – Crypto token interface (cryptoki)
- PKCS#12 – Personal information exchange syntax
- PKCS#13 – Elliptic curve cryptography
- PKCS#14 – Pseudo random number generation
- PKCS#15 – Cryptographic token information format

PKCS#1 – RSA PUBLIC KEY CRYPTOGRAPHY

RFCs:

- RFC 2313 Version 1.5 März 1998
- RFC 2437 Version 2.0 Oktober 1998
- RFC 3447 Version 2.1 Februar 2003
- RFC 8017 Version 2.2 November 2016

Definitionen:

- RSA Schlüsseltypen für öffentliche und private Schlüssel
 - Öffentlicher Schlüssel:
 - n: modulus
 - e: öffentlicher exponent
 - Privater Schlüssel
 - n: modulus
 - d: privater exponent
- “Multi-prime” RSA (ab PKCS#1 v2.1):
 - Modulus ist das Produkt von mehr als zwei Primfaktoren

Definitionen:

- Umwandlung von Datentypen (Integer <-> Octet-String Primitive)
 - I2OSP
 - OS2IP
- Ver- und Entschlüsselung (Primitive und Operationen)
 - RSAEP $((n, e), m)$ mit m = Nachricht (Integer)
 - RSADP (K, c) mit K = privater Schlüssel & Parameter zur Erzeugung
- Signatur und Verifikation (Primitive und Operationen)
 - RSASP1 (K, m)
 - RSASV1 $((n, e), s)$

PKCS#1 – VERWENDET ASN.1

PKCS#1 sieht für die Repräsentation von Schlüsseln das ASN.1-Format vor:

- Abstract Syntax Notation One (ASN.1) – ITU-T-Standard (gemeinsam mit ISO)
- Definiert Repräsentation von
 - Schlüsseln (öffentlich/privat)
 - Zertifikatanfragen (CSR)
 - Zertifikaten
- Darstellungs-/Übertragungsformate:
 - DER
 - CER
 - PEM (nicht Teil von ASN.1) – oft Base64 encoded DER

Privacy Enhanced Mail
(definiert durch IETF)

- RFC 7468
- Encoding von kryptografischem Material

Ein Guter Einstieg (wer es wirklich wissen will):

- Olivier Dubuisson and Philippe Fouquart. **ASN.1: communication between heterogeneous systems**. San Francisco. 2001

Als OpenBook: <http://www.oss.com/asn1/resources/books-whitepapers-pubs/asn1-books.html#dubuisson>

Merkmale von ASN1:

- Beschreibt Datentypen (Syntax ähnlich einer BNF) und Encoding
- Zum Informationstausch zwischen unterschiedlichen Systemen
- Lange Versionshistorie (X.208 von Nov. 1988), aktuell: ASN1:2015

ABSTRACT SYNTAX NOTATION ONE (ASN1)

Datentypen

- Primitive Datentypen
 - BIT STRING (b7 bis b1)
 - BOOLEAN
 - IA5String (IA5String)
 - INTEGER
- Kombinierte Datentypen
 - SEQUENCE (seq)
 - SET (ungeordnet)
 - SEQUENCE / SET OF

	b7	O	O	O	O	1	1	1	1
	b6	O	O	1	1	O	O	1	1
	b5	O	1	O	1	O	1	O	1
	b4	O	1	2	3	4	5	6	7
b3	O	O	NUL	DLE	SP	O	@	P	'
b2	O	O	SOH	DC1	!	1	A	Q	a
b1	O	O	STX	DC2	"	2	B	R	b
	O	O	ETX	DC3	#	3	C	S	s
	O	I	O	O	EOT	DC4	¤	D	t
	O	I	O	1	ENQ	NAK	%	E	U
	O	I	1	O	ACK	SYN	&	F	e
	O	I	1	1	BEL	ETB	'	V	v
	I	O	O	O	BS	CAN	(W	g
	I	O	O	1	HT	EM)	g	w
	I	O	I	O	LF	SUB	*	H	x
	I	O	I	1	VT	ESC	+	X	h
	I	I	O	O	FF	IS4	:	i	y
	I	I	O	1	CR	IS3	;	Y	i
	I	I	I	O	SO	IS2	=	Z	y
	I	I	I	1	SI	IS1	.	J	j
	I	I	I	1			?	L	z
	I	I	I	1			O	\	{
	I	I	I	1			-	m	k
	I	I	I	1			=	^	l
	I	I	I	1			.	n	~
	I	I	I	1			/	o	DEL
	I	I	I	1			?	-	

ABSTRACT SYNTAX NOTATION ONE (ASN1)

```
RSAPrivateKey ::= SEQUENCE {
    version             Version,
    modulus              INTEGER, -- n
    publicExponent      INTEGER, -- e
    privateExponent     INTEGER, -- d
    prime1               INTEGER, -- p
    prime2               INTEGER, -- q
    exponent1            INTEGER, -- d mod (p-1)
    exponent2            INTEGER, -- d mod (q-1)
    coefficient          INTEGER, -- (inverse of q) mod p
    otherPrimeInfos      OtherPrimeInfos OPTIONAL
}
```

Schlüsselerzeugu

```
$> openssl genr
```

RSA PRIVATE KEY

```
[matze@tschitta] /tmp $ openssl genrsa  
Generating RSA private key, 2048 bit long modulus (2 primes)  
.....+++++
```

```
+----+
e is 65537 (0x010001)
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCQAQEA008YK4TzuJDJMvjZXrn1rBF4d74tx+23SV31tOMEmi3cDVTF
nNKBINR7Gfw0T6E0587rBaPSIRJd8pa346icjBonCqpmRWeskaij28cNYiLueILo
CncBlaKGqPPGOZF8wNokvWh77kIUAJCBHQvtjhLQuCgZH17CzBKM2HkUTcaH0ErE
XqOalApy7tPymI2ue5+C1fJrUp+91Kaj/aCUgANADUmAy8K4icUGI/vsQCXWl1k
D+shXYyy1FKow0iJfVn5A89UXwTzdp3RSRhrPB6Q3y21ae18AdERzzNIlophy2Ep1
GfRtbaF3N6WVYDYQIKO+qWFMCJ7talmSn4gSIWIDAQABoIBAH6fnvQ1L3ciC+8n
f9prxFPFzDkdFYIAyRyF2X8TquzzJMw4V+c5nXd23G2t1NoiWTPXoq42hOycfP+p
oXgi0R1jbpHNivfh1Dqbctv4amSsWgqNF7rdpW1tfxqvQFGvBPPrn2IhD9xvHLQVJ
kpU9WJUSYVB5dtr9jG2NkCkIJUqU3aGIDsjm4F4xqvDzrqj1e011zaVtWuTSvU8I
J3pnz5wl5MGA9FWWW/5EOqe2h2Cuqech0+baOUeGcGNpgE9GmeP8A6m+uyTcodlq
9D/Yebgw17IKTr30c0QvpWMFDWMiHlWJFbQrzkOpM+pJqdBYZefyAdsXM594ZXaz
E0Y9ccIEcgyEA+5xcC1dkDbbkzKse4Qt8cenyBbIoiF8t/GB9HyubwPmtjOfK2ViN
r3nm12oxccKPFxopIpMhsLn7rvax+aUq4j1LQJS0ig2tW4upm04DZs2sPox5d5n
YE6hwalUja/uE1zCZYqH0/gUBfqNQfTPEI2Dcnq2dcVjEtRiH8yCmwcCgYEAlv7B
agNTpiZ4qs9nAf3vVvRj1BiChr+zdiMXTj37qweB3Y3xHq7M3swt8Ln+y0kcbCF
xNfNBseNGr9DuuX1VaK4x+/YX20Xub7bTuXdDf1fL7YMrDMPx3NIMkBoJ/Em8sOo
QelserHXib2bMYJwFWxGkDQi1yhNKFjAyLfmlQUCgYEAlfZXZhNcojjThTx+2+Lv
borHoSp6K4nye2iTdghDiVFjdmuj1A92/Q549NoZv8D+jO8d1Y78bc4VH1jpxp
IjgmMhgX0mbVOOogqWxuSS+jsnDnsWw8pMsY2Naft2NPmKrWQAXsDujxw92Kwcw
DrR5sej1aTg9NdLspuD9NzsCgYBvWA6pLchjnQTG8iLB1YGF0wrN8BjIn9t9H85
BGj69a6zpJjrK7jJx7IKaWmlHtLyAj61HCi1saE1Sfr0z2WIjtZvbZq8hCKu8tY3
ftPho2+vKQ3tHKC+ZcjtjYgLr4vBTG1WFJwOBM5qzfys3n/XyHSe+DFXBruiH5Dw
IrXLVQKBgQDEXVEVznj/oKAc/Y/gwH6qnry7rcEy7stfcY1VQxOGOKHVbaQ62J
5HIV6474BuqmH38uzpjpojZhESQzOtB1tQzitZYdUwjSL/iNcahi2wpXAxr0AYM
w9RgdqSB4pm7AzHz4oia4cNP74udVYp4poolVZxrDD8Pai0xdG1/Pg==
-----END RSA PRIVATE KEY-----
```

Schlüsselerzeugung

\$> openssl gen

```
[matze@tschita] /tmp $ openssl genrsa 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
e is 65537 (0x010001)
-----BEGIN RSA PRIVATE KEY-----
MIIJKQIIBAAKCAgEap11SsYFqJDwWmbTP2CgEqCSKxRE2eNtWDy6f1Z0cMuZvRvcY
v9H/mknDmHGIMSCCaH64NnaInlqgP5UgksRpIgrmmQJh6MUkL-Xhs3rFVDzpgKdI
kRDsnkd7iIffYRqsG1012xeFku3cGot1vguaz70G37CKjfZ+vlnN4RNODns1ZkN
V0BTW/psMtq8FDxkm2ESL+nQoylii1wktvpo/etICHL2FN40DRuz13ab3njKliajE
nxBSsSWs1XrWmHDJc8Q+mbWxH26yNvBtHnHupt2tRgcnVNP9KRM1263d2zXFU04
5bPQwdwHT1ZK2ZSSjS81E3teWNR-T+d4TO7Ns06jgfSuttiN1id1x25YBaqt8uAl
ue8dKuPw96C7EtvxpS7J5M20AHJVD5BNvLMP8ctB8wTeM8SbjSnbwgeo/74elTF2
Tr9qkWOB9G91v3rnM//fa0ccj1r8rp7s371W1laRocQS5MKt1Fcud+ogQuhT6
h18/7gekV8t0Dy0JMKoYNgRQjMkoYNgRQjMkoYNgRQjMkoYNgRQjMkoYNgRQjMkoY
c92WLurIMv1326JhgMRB7vVOE1/3VuBmeFDT657ytw9v8YLZYpuL50e99mGi
401asoxnmhCmSgcc02r1u2Fh1wawtaNm0kP0wjeR/uZ/S92NND0CAWEA
AOKCqgkBAEg8r2erTRThAege9te5g4os2r10jwkw9dxUtrDHAloddu/Hc/NKvYzbH
v6QEWYrreZl0rVd1wqiPnDiwuOfbzJ513wzjokAJP9wz7o011Vw/mxicuSML1V
ctabw1Y1P-SnKTK10YOdhqnx5vhujhns02M6v72rXWkv3vfH1fa8CrD031p+JMTQn
pUWWwpALXkgnguByPwQcZqS1Z1U1jgte0Ob6exrwxv5nuXJ11D5j4uScJw0QPQ0S
dxy8be81IEQC63shmc61EHAEAKA00BeQfr02aqnahGvRIGAuauZ30RJgo6L1GTOP
A+wp*5CLtFxNMr9DdsWaNhzTwe7xtSdf0uKn0a1Ja72B0XPYpjocqfg/RJCY
08351JmmKEjyVELCrasNOutAUhdW84BJGJRDkBzqCT0M4jtRHypwLNZ3AfcbPfkC
CR05gn2YJTsVcnNet82Ktx7c81mCq7vNL9hpSwSbo+Gn4f7i7D7YPIwkt4R
wnHf0uAybP81xwD/8Csf02B3ycnUx8MqP+3gK+0vaM9dt1w63MSM1l0sQzQy
x4JJH6JZG1N2r78Sat1xrigPpljeIDQgLog5c+EtYR7jxkBhU7jtCBNEath
jwU0gQIxw9QXX1VmLMvvc/himucv8tFkVb4pejh1v63/lqtc4QKCAQEA1Vw11N0O
P81FM7BuYEd1AYSHoAf1mbzKrg6BM12DvhdSc42rTvk8k3+zr7dVEXY4Sw7P7jHqUca4FGeYLygtBm2
21PkdgQb5NY0Usxpfp2fMwgtn9YsDmjw9QpclyTzkyXjLyovquvNra41202am
/6U//Sh11+vOk1V0gdBEV-hsPN1bYOpsn9js2ZbJSbJqghWDyJMaWop0FvkPvt
/yMz2kausUpv12xjoWtY9C1Pke4Wd+va5E/rwLcjj3tpvXTTmvyhdd4ros8o
Uxs2oFic42VFrwKCAQEAy193kbzqzU14EKVv1lrrAHPHOuKm1w9wp1634bLS
vSSN8gcs2vJMaSh7K1fR0MQKt0zaOs3k05#7Kw02UssJwvTm9X1WCExxTAh
D2fTaQco1fybYltwDQtztyV8sptzeK01vcqgzk8RYWynbBjt5XAt7SzxeKt7sAkrmw
qc7TJwA01GyctTm1QoUw0RAvhmdE1IwppaxvTcJ+PMF2bNQ/Dnoqimv+4j8sg
e8t984zfdhflIghak9unZTytRvtTak/Abgp5w1Coj0ZpdFds5r1QUDH9XGDaiaRa0
OdZwfabN5Nm1LxsdixQ/U6ND607lyLrcwo+wW9HC9yyKCAQEA19Mhdug/My5E2r14
hbq2wLnzy5dhdeMv3wnfjyxEKSzi1snGvQhRPx3RGjoxURekJM0gRNCLE5bsti
SmqQXFyssPf73c66gjyeExD600H3fKE09AtpxcadG1jv3yIe-i2yGdyYuS4V
wo/vHM40czs5hnwQzdapbbjshu52vUrpb8Ms/vnc/dblnsPDpSwH0XQp8ngBKT
EE51coBez9u3s81cpphdYXbQwJ7bqv5eG9vMjQpW4uRkt1M66+c5vKfaLI14Cw
OHueyEl8pF0Ef0f1PAUv8vKO0Ke0qd1b9rI3hncrSuqzW423Ctpyq5BDRNkca
s0AtnwKCAQEAj5T02m0OIn/w2eusTCzPhM18x3O+3ryVS1HOrasii1X15UVyrc
bqFQOoxVLP3/RmnMmr7D121wNus12rPrGon6RQ6fyycyDHCMyB5c2aue2
9QDH16+R3DlxR2CwK457MULbj5u6Fvk21lu63cmHo2491jB2nZaq5+robdIPfH6P
Dho9Sv1gtmRkHpKjPUKL1L1ALj1PR1l6gTav1s0me4whJPA3dg5z3MBQpdHDS/G
Nvdawhnr4+PyW4k0D1HVjooHSvrxTJMrg14fx78L2G2snB3cf/PMGz5Qcx1lwpyHj
y3u3tZVY0HspIduh15Z/xdM1LU0F0r0AawKCAQBNT7r3nVdFwq1/10wcaxr5z7q
81Pyc0YK75mfQcmklgsSR73GOTOWL/Y+c4WU+cmz4093xjZM2atg0Qm7NmC/gY4
zxza5zgwX8By7Skaf/JUebpk4tjYvU0CwADMPEZ/BqnGnEp/FAAUf1x9QzWxx
yxvLzfcm/FD+ivIoupKqoIgwkt6xCo/Hv46bwJn2zAxYjFe7v5s76OC1E+rO+Bjnn
MDXb5F1jUsSHqo10HnHOVvN7JzB7G7ipelHdejtqewMqRt+ryiMcMsChMiS9yu0t
qxM5ciJxsZswaeGHu0gkX54t/Hmbv7OMvL8ULzgj14wdgi2PD0j0TeWgomZk
-----END RSA PRIVATE KEY-----
```

Schlüsselerzeugung

\$> openssl gen

\$> openssl gen

RSA PRIVATE KEY

```
[matze@tschitta] /tmp $ o[matze@tschitta] /tmp $ openssl genrsa  
.....++Generating RSA private key, 2048 bit long modulus (2 primes)  
.....+++++  
-----BEGIN PRIVATE KEY-----  
MIIEVABEADANBgkqhkiG9w0BAQE is 65537 (0x010001)  
-----BEGIN RSA PRIVATE KEY-----  
/YP+bkC3YCBaVxRP1vK9lu/8-----PKCS#8-----  
W+374zMvEQQsCqyarJ1As8KyMIIEPAIBAAKCQjEA008tK4tzu0DJMvjZXrn1rBF4d74tx+23SV31tOMEmi3cDVTf  
KJOOsLdkSnpFBJ18c1Wo6/c3inNKBINR7Gfwot6Eo587rBaPSIRJd8t4a46iojBonCqpMRwEskAij28cNYiLueILO  
DAOPZcRpM/7suu5SjWeQSRgEcCncBlaKGqPPGOZF3wNokvWH77j1c1CBHQvtjhLQuCgZH17CzBKM2HkUTcaH0ErE  
1qCR17p15eh5UuNudy9WVm1xXqOalApy7tPymI2ue5+c1E-----PKCS#8-----  
Yw8KnKbLAGMBAEECggEAQdkzD+shXYyy1FKow0iJfVn5A89UxWTZdp3RSHrPB6Q3y21ae18AdERzzNI1Ophy2Ep1  
P21YBTOnav3tPu6Blwh4h61LMGfRtbaF3N6WVYDYQIKo+qWfMCJ7talmSn4gSIwIDAQABAoIBAH6fnvQ1L3ciC+8n  
9G5fI+tAV7yAqGLrOOnVBbuEtf9prxFPFzDkdFYIAyRF2X8TquzzJMw4V+c5nXd23G2t1NoiwTPXoq42h0ycfP+p  
uXSU3CPDVfqxck0BPpM8c4CQixXgi0R1jbPHNivfh1DqbctV4amSSWgqNF7rdpW1tfxQvQFGvBPrn2iHd9xvHLQVJ  
km7cPAG9f/fKw1sWaicmGC;s1kpU9WJUSV5dtR9jG2NkCIJUq3aGIDsjm4F4xqvDzrqJ1e011zaVtWuTSvU8I  
SZxpZbnQivnpfcZ2RdbH6sCrI9D/Yebgw17IKTr3OcOQvpwMFDWMIh1WJFBqrzkOpM+pJqdR1yAdsxM594ZxaZ  
LoYkhDYhIjcHMrw2U7ArIBv1E0Y9ccECgYEAt5xcCIDkDbBkzSE4Qt8cenyBbIoIF24cChyubwPmTjOFK2VIn  
k1URhvndpOaOxd/fuGqk2TQ3+r3nm12oXccKFpxopIpMhSLn7rvax+aUq4j1LQJS01g2twupmOU4DZs2sPox5d5n  
KE1qW2jwhscoQrGxVe9Y3ZwZ>YE6hwauUja/uE1zCZYqH0/gUBfqNQfTPEI2Dcnq2dcVjEtRiH8yCMwcCgYEAlv7B  
FFVO2SQFJv/k3JBEmOzSXfcAagNTpiz4qs9nAfh3vVvRjBiCicHr+zd1MXTj37qweB3Y3xHq7M3swt8Ln+y0kcbCF  
zVwIop8qXn+Ma5119ZAJprVN4xNfNBseNGr9DuuX1VaK4x+/YX2OXub7bTuXdDf1fL7YMrDMPx3NIMkBoJ/Ems8oo  
62aElnCuFQKBgElCEPoaafqj4QelserHXIb2bMYJwFWxGKDQi1yhnhKFjAyLfmlQUCgYEAlfZXZhNCojjThTx+2+Lv  
4eheOV2JXs+86P/rYesndoyWborHoSxp6K4nye2iTdghcD1VFjdmUj1A92/Q549NoZv8D+JO8d1Y78bC4VH1jpxp  
j6JehmpWrOwIDGTfE16k28n1fIjgmMhgX0mbVOOogqWxuSs+jsnDNsWw8pMsY2NafPT2NPmKrWQAXsDujxw92Kwcw  
MDZ1ISNjtQ+uA+UumpvLEbDcIDrR5sejlaTg9NdLspuD9NsCsGyBvWA6pLcHjnQTG8iLB1YGF0wirN8Bjnt9t9H85  
MJzCxiM047X1Jh/icvS/b3SjB69a6bzPjkrK7jx7IKaWm1HtLyAj61HCi1saE1sfr0z2WIjtZvbZq8hCKu8ty3  
LSY4vXKVqWyVE9TTisFH+Gjp1ftpHo2+vKtHCKC+ZcjtJYgLR4vBTG1WFwJOCBM5qzfys3n/XyHSe+DFXBriuH5Dw  
VwhqPU9etcxTWgpGZxjIP7Yp1IrXLVQKBgQDEXVEVznj/okAec/Y/gwH6qnry7rcEy7stfcYlVQxOGOKHvbaQ62J  
zel06ADP4ogDZd8wtJ+DIjXYI5HIv6474BuqmH38lyzpjpojZhESQzotB1tQzitZYdUwjSL/iNcahi2wpAxr0AYM  
aQaqDIVxsppSRW5kvML0g== w9RgDqSB4pm7AzHz4oiia4cNP74udVYp4poolVZxRDD8Pai0XdG1/Pg==  
-----END PRIVATE KEY-----END RSA PRIVATE KEY-----
```

RSA PRIVATE KEY – ASN1

```
[matze@tschitta] /tmp $ cat vlnws1.rsa.pkcs1 | openssl asn1p
0:d=0 hl=4 l=1190 cons: SEQUENCE
4:d=1 hl=2 l= 1 prim: INTEGER :00
7:d=1 hl=4 l= 257 prim: INTEGER :B5DE4DE2DC9
217866CC2E424F4204A57B6A928A2E2C47FF2165176FB546BED494AE519
6B09718B3B2A102E3CD36D1680047F1410EA129B26B0F65C14008557700
062CE28EAE2B2042DDAB99D3D97F68B53F87553FF
268:d=1 hl=2 l= 3 prim: INTEGER :010001
273:d=1 hl=4 l= 257 prim: INTEGER :9E687A189E9
4681F3AFE69EF52000A20EA45DED239888414DDA1A96BF2356B61CA6D50
42672CF11BC3C299658CA7DBFAD70DC5ABAE2386B62447151DE232E4A45
0796929FEC22A8C863C8B2574A9DA6907AFC80AD1
534:d=1 hl=3 l= 129 prim: INTEGER :E41E23FFE22
80CE9DD12913111C5CE3F125E864F90DF95AA2F0E7FC852DBE3DAF4E74D
666:d=1 hl=3 l= 129 prim: INTEGER :CC190288048
E32A1D7B93DFA32A23FDF4156263D68AAA9D7A367D24A6FD4968D093B92
798:d=1 hl=3 l= 129 prim: INTEGER :D1AD3EABC1E
7F05EB9BC8A464F2EC12FCA23A652D6381D2A4B8C8939C9A7A25DAD043A
930:d=1 hl=3 l= 129 prim: INTEGER :C563BECDFBD
FC24A424FB6C03EAE2AA5EA13BEA20F9356F5E5E4A83E338FE0170101EB
1062:d=1 hl=3 l= 129 prim: INTEGER :E23E5E06E33
2F10769EA2A3B26E9173784792474D83ADBB7C9CEE3D84F889B50099A76
```

```

[unreliable]schitali@unreliable:~$ ./a6p S cat vlnmws rsa pkcs1 grep -v '^--' | base64 -d | hexdump -c
00000000 30 82 04 06 02 01 00 02 00 b5 d5 4d e4 0d [REDACTED]
00000010 dc 94 da b5 95 a5 76 ae 0d 7d 4e 78 73 70 87 75 [REDACTED]
00000020 c5 d5 8d 35 0d 7d 4f c5 68 ec 0d 7d 4e 78 73 70 [REDACTED]
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000040 46 72 a2 04 b2 cc 7b 15 b3 d2 17 86 ee c4 0d [REDACTED]
00000050 24 f4 20 a4 57 a6 28 a2 e2 cd 7f f2 16 51 76 [REDACTED]
00000060 1b 54 b5 ed 49 a4 e5 40 7c 7d 52 b2 5a 0b [REDACTED]
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000080 1c cb 00 01 3c 7d 77 79 25 06 ae 84 84 8c 08 03 [REDACTED]
00000090 d2 b5 09 01 c3 53 75 5b 55 aa ef 4d 6b 88 0a 54 [REDACTED]
000000a0 bd 69 09 71 8b 2b 3a 20 23 0c d3 16 80 04 73 [REDACTED]
000000b0 18 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
000000c0 94 57 44 19 25 73 25 42 85 38 0c 7d 71 65 [REDACTED]
000000d0 9e 3a 26 b5 43 20 08 08 a7 c6 77 99 a5 33 9d 6c [REDACTED]
000000e0 67 d2 b2 97 95 04 22 09 1b 1b 80 14 cf cd 85 [REDACTED]
000000f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000100 8e 63 06 65 dc 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000110 01 02 82 01 00 01 9e 6b 78 18 9e 9e 3f 8c 19 e0 [REDACTED]
00000120 30 44 e6 c6 15 22 20 68 ac ec 7d 76 00 94 [REDACTED]
00000130 08 26 15 8d 40 7c 46 ac af df 95 36 de 06 0a 08 [REDACTED]
00000140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000150 fe fa 02 17 44 68 14 1a 09 ef 42 00 00 20 20 [REDACTED]
00000160 45 de 02 39 88 84 14 0d a1 a9 6b 2f 35 65 61 ca [REDACTED]
00000170 60 52 29 59 98 55 0b 0b a8 a8 37 77 88 fd 64 [REDACTED]
00000180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
000001a0 29 20 2c f7 24 da 40 be 87 4d 67 23 f1 1b [REDACTED]
000001b0 c3 c9 65 85 87 af db 7d 07 c5 ab ae 23 86 bd [REDACTED]
000001c0 24 47 15 1d e2 32 d4 85 88 dd ca c1 4a af af [REDACTED]
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
000001e0 51 bf ae 49 04 30 2c 09 ef 2d 83 a4 44 83 ec 40 [REDACTED]
000001f0 61 d4 3a fc 28 2f 48 58 bf 5d 34 22 b1 33 cd [REDACTED]
00000200 09 50 79 69 29 cf 2a 86 86 3e 8b 25 74 9d a9 [REDACTED]
00000210 69 07 01 f8 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000230 34 e2 64 36 69 20 cf ad 5c 14 01 78 bf 29 25 [REDACTED]
00000240 3e b3 dc ab 03 13 c1 85 56 22 0f 8d 67 31 e0 [REDACTED]
01 3d 11 3d 95 98 54 0b 0b 05 0d 0d 92 93 11 00 [REDACTED]
00000250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000270 bd 93 fa 4f 97 4d 50 54 31 65 23 2d 05 db bb 9d [REDACTED]
00000280 30 5e 68 66 67 82 38 5b 68 16 66 67 as 18 13 [REDACTED]
00000290 9e 5d 8d 07 05 25 04 0a 69 02 81 81 00 cc 19 [REDACTED]
000002a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
000002b0 bd af 77 29 ff 47 64 39 16 13 cf 66 38 0d 84 [REDACTED]
000002c0 af ac 1f 0d 9b 9b 9f 2d 13 43 f3 hc b6 cd ea [REDACTED]
000002d0 2b 21 aa 86 65 65 51 ar 04 5c 00 1e 31 al d1 [REDACTED]
000002e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
000002f0 67 d4 64 34 64 64 64 09 sf 92 00 5f 43 93 c7 37 [REDACTED]
00000300 b2 26 90 26 da 4b 8b 95 wf 4d ia 1a 37 02 55 [REDACTED]
00000310 0a 51 7d 93 77 25 b2 2b 5f 04 db 68 27 02 81 [REDACTED]
01 81 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000340 d1 f8 93 cc 22 77 00 0f 07 0f 06 06 04 26 64 84 [REDACTED]
00000350 04 d2 a2 7e 27 1c 36 8a 61 42 65 57 4a 88 0e ad [REDACTED]
00000360 97 f9 5b b9 46 46 46 ce 21 cf 2d 53 23 as 52 d2 [REDACTED]
00000370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000380 54 31 67 91 01 51 62 0d cf fd 07 5c 67 14 [REDACTED]
00000390 2e ad 30 3d 42 05 01 05 36 eb 11 0b 03 aa d7 [REDACTED]
00000400 54 19 02 81 81 00 05 63 bd cd fh 01 ad fd 7d b3 [REDACTED]
00000410 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000420 c9 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000440 57 68 8e 08 13 3b 7b 57 01 94 91 68 25 7b 2d [REDACTED]
00000450 13 be 02 09 56 95 45 wf 48 3e 33 82 00 17 01 [REDACTED]
00000460 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000470 77 03 b2 b2 57 c2 48 42 4f b6 c0 3e ae 2a 55 ee [REDACTED]
00000480 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000490 48 8e a4 54 94 6b 02 81 00 02 e2 se 06 03 9d [REDACTED]
000004a0 f6 c9 7b 7b 51 02 51 59 f1 9b 68 13 8f 71 f0 [REDACTED]
000004b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
000004c0 45 57 61 68 63 0b 0d 09 00 00 00 00 00 00 00 00 [REDACTED]
000004d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
000004e0 3b b2 d2 55 93 47 45 fd 42 07 69 ee 2b 2b 26 [REDACTED]
000004f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000500 49 17 37 84 79 24 74 da 3a bd b7 c9 ce 08 48 [REDACTED]
00000510 88 9b 00 09 96 7e 26 7b 9b 77 47 76 7d ad bz [REDACTED]
00000520 79 8b 16 ce 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]
00000540 92 15 00 7a 7d 98 09 17 07 54 [REDACTED]
00000550 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [REDACTED]

```

- Angriff gegen PKCS#1 (Version 1.5) – Bleichenbacher Attack
 - Chosen-Ciphertext-Angriff
 - Idee: PKCS#1 definiert **0x00 0x02** als Nachrichten-Prefix bei Padding
 - Einsatzbeispiel:
Eavesdropping des Austauschs eines Session-Keys in TLS
(Session-Key < RSA-Schlüssel)
 - Methode:
 - Ciphertext anpassen, bis der Server beim Entschlüsseln einen Erfolg meldet
 - Session-Key aus mitgelesenen Datenpaket entschlüsseln
 - Komplexität: 30.000 – 130.000 Versuche bis zum Erfolg

PKCS#3 – DIFFIE-HELLMAN KEY EXCHANGE

- Standardisierung des DH-Schlüsseltauschs (vgl. TLS-Handshake Phase 1)
 - Generierung der DH-Parameter durch eine dritte Instanz
 - Wahl einer Primzahl p
 - Wahl der Basis g mit $1 < g < p$
 - Optional: Wahl des Längen-Parameters l des Secrets
(Wahl der Primzahl oder Länge des Secrets als Performance-Setting)
 - DH – Phase I
 - Erzeugung des Secrets pro Partner und Octet-to-String-Konvertierung
 - DH – Phase II
 - Exponentiation des Partner-Secrets => Ableitung des PSK

Historie

- Version 2 im September 2000
- NIST-Empfehlung im Jahr 2010
- Update zu RFC 8018 im Januar 2017
- Password based Cryptography Specification
 - Ableitung eines kryptografischen Schlüssels aus einem Passwort
 - Key-Derivation-Function (KDF) basierend auf kryptografischen Hashes
 - Verwendung eines Salts
 - Variation in der Anzahl der Iterationen

Verwendung eines Salts

- $\text{DK} = \text{KDF}(\text{P}, \text{S})$; DK = Derived Key, P = Password, S = Salt
- Vorteile:
 - Salts erschweren die Vorberechnung sogenannter Rainbow-Tables für Hash-Verfahren
 - Salts verringern die Wahrscheinlichkeit von Kollisionen des Hash-Verfahrens
- Empfehlungen:
 - Zufällige Wahl des Salts mit einer Länge von mindestens 64 Bit (8 Zeichen)
 - Optional: Hinzufügen einer festen Bytefolge (z.B. des Zwecks des DK)
 - Keine versehentliche Nutzung desselben Salts für unterschiedliche Zwecke

Anzahl der Iterationen

- $DK = KDF(KDF(KDF(\dots KDF(P,S))) \dots)$
- Vorteile:
 - Erhöht die Komplexität der Berechnung eines Schlüssels
(Stichwort: Bruteforce)
 - c Iterationen erhöhen die Sicherheit des Schlüssels um $\log_2(c)$
- Empfehlungen
 - NIST: [...] A minimum iteration count of 1,000 is recommended. For especially critical keys, or for very powerful systems or systems where user-perceived performance is not critical, an iteration count of 10,000,000 may be appropriate.

Key Derivation Functions

- PBKDF1 und PBKDF2 (Password Based Key Derivation Function)
DEPRECATED
- Ablauf:
 1. Wahl von Salt s und Anzahl der Iterationen c
 2. Wahl der Länge des resultierenden Schlüssels (DKLen)
 3. Führe KDF^c mit (P, S) durch und
 4. Gebe den resultierenden Schlüssel (Derived Key) aus

PBKDF2

- KDF basiert auf Pseudozufallsfunktionen (PRF; z.B. Hash oder HMAC)
- DK wird aus allen Zwischenergebnissen der PRF erstellt (concat)
 - Dabei ist hlen die Länge des resultierenden Hashes
 - $DK = T_1 \parallel T_2 \parallel \dots \parallel T_l$ mit $l = DKLen / hlen$
 - $T_x = F(P, S, c, i)$, $i = x$ und $F = U_1 \oplus U_2 \oplus \dots \oplus U_c$ mit \oplus = XOR
 - $U_1 = PRF(P, S \parallel INT(i))$
 - $U_2 = PRF(P, U_1)$
 - \dots
 - $U_c = PRF(P, U_{c-1})$

REKURSION VERHINDERT PARALLELISIERUNG!

PKCS#5 – PASSWORD BASED CRYPTO. SPEC.

PBES1/PBES2 (Password Based Encryption Scheme)

- Kombiniert die Verschlüsselung einer Nachricht m mit dem Schlüssel aus der entsprechenden PBKDF1/2
- Verschlüsselung mit beliebigem Verfahren, z.B. Blockchiffre (DES oder RC2)
- Parameter (Beispiel für DKLen=32):
 - $K = DK_{0..15}$
 - $IV = DK_{16..31}$
 - Padding (aufgefüllt mit 01,0202,..., 08⁸, ... nach Anzahl benötigter Zeichen)

PBMAC (Password Based Message Authentication Code)

- MAC-Berechnung analog zu PBES basierend auf unterliegender MAC-Funktion

PKCS#6 – EXTENDED CERTIFICATE SYNTAX

Definiert Erweiterungen zu X.509-Zertifikaten

- Aktuelle Version 1.5 (November 1993)
- Erlaubt das (beliebige) Hinzufügen von Informationen zu X.509-Strukturen
 - Genutzt etwa für E-Mail-Adressen
- Definiert die Zertifizierung (Signatur) hinzugefügter Informationen
- **DEPRECATED** seit X.509V3

PKCS#7 – CRYPTOGRAPHIC MESSAGE SYNTAX

Als RFC 2315 veröffentlicht im März 1998

- Syntax für Daten, auf denen Kryptografie angewandt wurde
- Grundlage für die Verschlüsselung und Digitale Signatur von Nachrichten
- Basiert auf Zertifikaten aus Public-Key-Infrastrukturen (PKI)
- Anwendung: S/MIME, OpenSSL-Verschlüsselung, PKCS#12
- Definiert sechs Inhaltstypen:
 - Data
 - Signed Data
 - Enveloped Data
 - Signed-and-enveloped data
 - Digested data
 - Encrypted data

- Basistyp (Data)
 - Enthält beliebige Daten (Octet-Strings)
- Erweiterte Typen (die fünf anderen)
 - Definiert ASN1-Syntax für die entsprechenden Anwendungsfälle
 - Für Signaturen z.B. Informationen über den „Signer“
- Verschlüsselung ist Kompatibel mit der von PEM, falls
 1. Content-Info ist vom Typ Data
 2. Kryptografische Parameter sind kompatibel zu PKCS#1 RSA-Encryption
- Padding: Analog zu PKCS#5 (allerdings für Blockgrößen bis 255 Byte)

PKCS#8 – PRIVATE KEY INFORMATION SYNTAX

Definiert die Speicherung (allgemeiner) privater Schlüssel

- Motivation: PKCS#1 speichert nur RSA-Schlüssel
- RFC 5208 von Mai 2008
- Datentypen:
 - Privater Schlüssel
 - Verschlüsselter privater Schlüssel (mit PKCS#5 (PBES) verschlüsselt)
- OpenSSL ersetzt **genrsa** mit **genpkey**
\$> openssl genpkey -algorithm rsa

PKCS#8 – PRIVATE KEY INFORMATION SYNTAX

Definiert die Spei

- Motivation: F
- RFC 5208 vor
- Datentypen:
 - Privater Schlu
 - Verschlüsselung
- OpenSSL erlaubt

\$> openssl genpkey -algorithm rsa

```
[matze@tschita] /tmp $ openssl genpkey -algorithm rsa
.....+++++
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQcGKL51+1n0QOMN
Cc4Tq7H14pYokets9V2mJw+MI3mGIoQesEGPh0vRYV1xN8vwC3OzXhs2TkVIBDd
q2H0kQ+aUkU8AC07yY52USfSMNKjNT+sT2H9Gvtiy7HF8fGEeK+AuLJyb3saVzvK
JwcT2DOdx+mZsP04iPhYHI4K1tE2pgAyXNbG/6kFrbbBu9ZoyqjywHHcMNLVyxR
fiCnGkeySIP+EdK2rbn/MBIZnvh8jcCXhojEZwf6FqjU0to3B6XWaDNS4N5n13xN
R22rF1PnPnCqj3FaIqsupzR4P6vuY164JPIiOCmITfbK2fV58nqDcHxJs40MkiJ6EX
uc01YBztAqMBAECggEBAlhdB7J6wJHBG1mzHZntmjsaI096Ij8TJYBYI88p5JvS
2PZZTI+oN1R6YaoC4D0iDhkgQSzqLyDSrwkFjSVqh3FRR1wDe+Uvm1DGjulaBa
H6sEIxwsCLiVfK0v3T+tq/VEzyfEF5nZV+hDagVoqTJOK0aSfjRdyl1a2gHUEIgs
AYCD1tI3seM/95mos4yW2IaEjb/6Gj3/bkrkJz8DQd7W6a13WUezFrnb/DqBuzU4
TJ3FdIqRJ0UNy8CxRZKgAxF1I5rbvnka42Df84Y1x0cmUA4x6nGKA6r82kVcIQSQ
JiNBD0KR55UkxIRkcxGNsVFUBML90Zh1D9VE40cNQgECgYEAA0Vuv8qGrnad7jY7m
YS0uDhidMBkxM7hv/8Ffq5zdiaJGFwDbcBhcHx5HLy2+//suaPvmqWWVhn5ktELS
y3XUYifxQpbToZtbOcHn0GhZIdEvBsTeJhcsnsISu3Xld17OmQDsUzmt0BceNWX
JKoAvYxbmJ4I4jYgyc4spbPHGg0CgYEAAw9cXw9byGGFT8L3f1oWnfKs/vICDJKiP
ouOAlqB1KpSLM17UHXG5/1PmLbcHCq6QBz4eAHnTmOyIxZQTkSxt6evr1fgA3B3p
6uMS5irvo4rbWS4kM5ce+WRjRn8sJaF6tDhnfxZM/ek1AO1lfSPzCX2K7y90YjqX
X5b1QLsTteECgYEAgDYdR4LYvaDe2m9ECQkJJlrOkLEs35apHAqJk2hqh6pYMCg3
OqvAZti9F9h1GwDxiBuUQ/NRID9vvrEztL5BraNjYdtL+bFRqcplM81joIEhwox3
223cwrxx3WlQ2JedqsGrPjh9ZP6prtzhfCnyOJaa75XgKyrYJWg0qDIRpj1UCgYBw
ksQB130Ucm0zRZBx2CURTau3C8vMuG6LpNT2v38m+HcCshEuU35buf1svyDMU9s
01I4Z84ualc7ah/NJEKRYKAQPxsPaspXzunADxGvoeLu4czBmq4FH6TIkrSndKf5
CxvPb75VfESRIeNiCNXbsKO5VAeLpT4aHbvdsxufQQKBgA4im1KQBGZ+QPv/D32D
PEe+T7EVwh0+zORL1zsacRJYbzhl1EQsliMMLiH9awtOS5jPMMuQxaj+eq6ZCsM0
3dKHzzJ9W3zCW6PvGkCWavwymTGIInc64PcRc4XuM2iBqWJJDz7WsPxUDrMkAVvfET
oyY9tgFF1zoE8JW1JqOeWBj
-----END PRIVATE KEY-----
```

hlüsselt)

PKCS#8 – PRIVATE KEY INFORMATION SYNTAX

Definiert die Speicherung (allgemeiner) privater Schlüssel

```
[matze@tschitta] /tmp $ openssl genpkey -algorithm rsa | openssl asn1parse  
-----+----+-----+----+  
 0:d=0  hl=4 l=1214 cons: SEQUENCE  
 4:d=1  hl=2 l=  1 prim: INTEGER          :00  
 7:d=1  hl=2 l= 13 cons: SEQUENCE  
 9:d=2  hl=2 l=  9 prim: OBJECT          :rsaEncryption  
20:d=2  hl=2 l=  0 prim: NULL  
22:d=1  hl=4 l=1192 prim: OCTET STRING  [HEX DUMP]:308204A40201000282010100B4E891757D491E97F1B275ACAE17EDDF071E04EB400C  
5F10C5C2E8A916FB87DF98FA9623586C88D4E59FB5C40E99A76CA7A026D877365333123A40002DCEA837B99FA1D9F05D2505E8550B1ED763B4BF1BF817C7EBA1  
635AF561B4F86AC9B9990671A7F56B1CC0259A10796FFC22390ED03BC9529E23F602CEFFF3C35142EF8EB19BC9120DE40CB0713018C7B698D3CB7B1F5840  
463DE5A9932394F36D468BC2FA00123A18B1AEA2D627604EDE0FA7571AAFC2A643CF803745F02030100010282010100A5B0BC48D74EB8EE8137E9FF16146490  
94B2BC914B9A99A284A78918F0CB5A3646F4053725BF24C0071AF2D533BEC0F23BD033990D8DEB0A4FDF4CBD95E1B0086C9442380F05CDF23E1FEC208A54D74  
FA7438699C57EC6B66257FE3C658B99981566662E3E8FDA162782B4D5E60C3DCA1D93557BB7D877BE62CB40343B4F59FC3823A29C024B0C358A6460B0D60F25  
31DBED70315F92C912A79275931FDA529A7A6F045DFFE3860749ABC79C9642117E9CED1BCF7C8731A69D48102818100DD490F7386B4CC844C2A869A646EFO  
8D79AFD9CE80E4D270C2DBB6AB98BB079360B64C6C2B04134185B85B0D82CEB8CABE75159E9EF5787AA7589298AD0EAA5575C6E857E4D68431BE45807BFAEB5  
6B308702818100D149F2785BD77F524BDA934A46F3F2A56D90ACA0F0435C332DC9F5BBCF05ED2C5B49AD7E007474571F20A54A68449EE133DCB074B42D7B17D  
3D89E197E9AB14F071ADB537B1FAAA106E7238CE64DAD0111FA3639B36649EDC7272F0A7E4A9ACD3CCE5A5BD78AC34B6902818100A7E3B60C50AF003B7607E8  
AC5506B7ABCFB1553805CC4BF71343634DAEE6F00696DD8021593716D16AA58DC245D197094CA79D4E45B204CBF6A56462B3629C94C3BA6A5438268CB355E72  
BFEFE839096BAE30281804E92507157298427454AFDD8F8E244CA4E63EE2B4D883C690A5BB3E19A4B434B4FCA4D53EC9FCBB9760C17EF2533F0A023CE2B42  
C4B653B6953065A9DE318AD5C4036B47E20F90FC906CCC4CFA04654E767D3D1458E6905A201FCCF4B4D5810FBEF8B55A2A423F61028180252C6C551310AD892  
405EB250E27AE29CC636FE2A1AE836F453C224D4ED07161E2898577CACE8F03E2E701870FB0845C8F18F44A9DB365AD8F3D493DEE82ACC41EAE1FAAF85E143D  
88549884E1C27E3D81E9
```

- Also auch Elliptische Kurven als Schlüsselmaterial möglich!

\$> openssl genpkey -algorithm EC -pkeyopt ec_paramgen_curve:ED25519

PKCS#8 – PRIVATE KEY INFORMATION SYNTAX

Definiert die Speicherung (allgemeiner) privater Schlüssel

- Motivation: PKCS#1

```
[matze@tschitta] /tmp $ openssl genpkey -algorithm ED25519
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VwBCIEIFWe0ltk9okQtbkVr77860SDi4vk9ePJt0Q4STaLZeFF
-----END PRIVATE KEY-----
```

- RFC 5208 von Mai 2008

```
[matze@tschitta] /tmp $ cat ec.priv | openssl asn1parse
0:d=0  hl=2 l= 46 cons: SEQUENCE
 2:d=1  hl=2 l=  1 prim: INTEGER          :00
 5:d=1  hl=2 l=  5 cons: SEQUENCE
 7:d=2  hl=2 l=  3 prim: OBJECT           :ED25519
12:d=1  hl=2 l= 34 prim: OCTET STRING   [HEX DUMP]:042079E877D76EA4344CA63E6B716EDEF87449898E39E6D524EE7E7C70CF2B0AC0F4
```

- Verschlüsselter privater Schlüssel (mit PKCS#5 (PBES) verschlüsselt)
- OpenSSL ersetzt genrsa mit genpkey
 - \$> openssl genpkey -algorithm rsa
 - Also auch Elliptische Kurven als Schlüsselmaterial möglich!
 - \$> openssl genpkey -algorithm EC -pkeyopt ec_paramgen_curve:ED25519

BEVOR ES WEITER GEHT...

- Pizza-Challenge
 - Keine Fehler gefunden ?!
- RSA-Dokumente:
<https://ftp.gnome.org/mirror/archive/ftp.sunet.se/pub/security/docs/PCA/PKCS/ftp.rsa.com/>

PKCS#9 - SELECTED ATTRIBUTE TYPES

RFC 2985 (November 2000)

Definiert zwei übergeordnete Objekte zur Verwendung in anderen Standards

- PKCSEntity
 - Allgemeines Objekt zur Speicherung von PKCS-Standardattributen
 - pKCS7PDU
 - userPKCS12
 - pKCS15Token
 - encryptedPrivateKeyInfo

PKCS#9 - SELECTED ATTRIBUTE TYPES

RFC 2985 (November 2000)

Definiert zwei übergeordnete Objekte zur Verwendung in anderen Standards

- NaturalPerson
 - emailAddress
 - unstructuredName
 - unstructuredAddress
 - dateOfBirth
 - Gender
- countryOfCitizenship
- countryOfResidence
- pseudonym
- serialNumber
- ...

Verwendet etwa in PKCS#7, PKCS#10, PKCS#12, PKCS#15
– auch für kryptografische Informationen in Directories (LDAP)

PKCS#10 – CERTIFICATION REQUEST STANDARD

```
[matze@tachita] ~ $ cat /tmp/csr | openssl asn1parse
0 d=0 hl=4 l= 825 cons: SEQUENCE
4 d=1 hl=4 l= 545 cons: SEQUENCE
8 d=2 hl=2 l= 1 prim: INTEGER :00
11 d=2 hl=3 l= 176 cons: SEQUENCE
14 d=3 hl=2 l= 11 cons: SET
16 d=4 hl=2 l= 9 cons: SEQUENCE
18 d=5 hl=2 l= 3 prim: OBJECT :countryName
23 d=5 hl=2 l= 2 prim: PRINTABLESTRING :DE
27 d=3 hl=2 l= 12 cons: SET
29 d=4 hl=2 l= 10 cons: SEQUENCE
31 d=5 hl=2 l= 3 prim: OBJECT :stateOrProvinceName
36 d=5 hl=2 l= 3 prim: UTF8STRING :NRW
41 d=3 hl=2 l= 13 cons: SET
43 d=4 hl=2 l= 11 cons: SEQUENCE
45 d=5 hl=2 l= 3 prim: OBJECT :localityName
50 d=5 hl=2 l= 4 prim: UTF8STRING :Bonn
56 d=3 hl=2 l= 27 cons: SET
58 d=4 hl=2 l= 25 cons: SEQUENCE
60 d=5 hl=2 l= 3 prim: OBJECT :organizationName
65 d=5 hl=2 l= 18 prim: UTF8STRING :University of Bonn
85 d=3 hl=2 l= 20 cons: SET
87 d=2 hl=2 l= 18 cons: SEQUENCE
89 d=5 hl=2 l= 3 prim: OBJECT :organizationalUnitName
94 d=5 hl=2 l= 11 prim: UTF8STRING :IT-Security
107 d=3 hl=2 l= 30 cons: SET
109 d=4 hl=2 l= 28 cons: SEQUENCE
111 d=5 hl=2 l= 3 prim: OBJECT :commonName
116 d=5 hl=2 l= 21 prim: UTF8STRING :Matthias Wübbeling
139 d=2 hl=2 l= 49 cons: SET
141 d=4 hl=2 l= 47 cons: SEQUENCE
143 d=5 hl=2 l= 9 prim: OBJECT :emailAddress
154 d=5 hl=2 l= 34 prim: IA5STRING :matthias.wuebbeling@cs.uni-bonn.de
190 d=2 hl=4 l= 290 cons: SEQUENCE
194 d=1 l= 12 cons: SEQUENCE
196 d=4 hl=2 l= 9 prim: OBJECT :rsaEncryption
207 d=2 hl=2 l= 0 prim: NULL
209 d=3 hl=4 l= 271 prim: BIT STRING
484 d=2 hl=2 l= 67 cons: cont [ 0 ]
486 d=3 hl=2 l= 31 cons: SEQUENCE
488 d=2 hl=2 l= 9 prim: OBJECT :unstructuredName
499 d=4 hl=2 l= 18 cons: SET
501 d=5 hl=2 l= 16 prim: UTF8STRING :Uni Bonn Company
519 d=3 hl=2 l= 32 cons: SEQUENCE
521 d=4 hl=2 l= 9 prim: OBJECT :challengePassword
532 d=4 hl=2 l= 19 cons: SET
534 d=5 hl=2 l= 17 prim: UTF8STRING :ChallengePassword
553 d=1 hl=2 l= 13 cons: SEQUENCE
555 d=2 hl=2 l= 9 prim: OBJECT :sha256WithRSAEncryption
566 d=2 hl=2 l= 0 prim: NULL
568 d=1 hl=4 l= 257 prim: BIT STRING
0000 - 00 12 ae af 4d e5 87 85-1b ae bc f0 9e f1 2b 58 ... M.....+X
0010 - 7a 16 bf ab 00 71 69 09-5c f4 77 38 84 fb 9e 85 z...gi...w8
0020 - 45 39 22 3d 19 6c 07 f2-44 04 0d 0e cd 33 bd 8a E9'=J..D..3.
0030 - a4 df cb 5b 23 08 c5 04-86 d0 43 c4 3f 61 ff 38 ...[#...C?a.8
0040 - 2f 69 f2 92 d0 0f 6c-58 f0 4a ef ff c2 91 30 .../..).IX.J...0
0050 - 56 da db cd 03 7d 0e 98-81 80 e5 01 dc cc 45 03 V.....E.
0060 - 3a d7 71 91 03 87 a5 4a-12 c6 04 a2 17 1f 9d 64 ...q...J....d
0070 - 56 44 c9 d4 aa e5 65-48 4e 27 5f 42 8b 75 a1 VD...ehN..B.u
0080 - bb 21 3f 84 2a 1a ca-af 1b 4a fc f5 7f da ???.D...
0090 - 2d dd d1 01 91 6a 24 74-b3 af ff dd a1 64 e4 a2 ...nSt...d.
00a0 - f0 da a7 2c 28 96 dd 99-c0 49 c1 96 b6 a8 2d 31 ...(.I...-1
00b0 - b7 ab f3 0b a1 4f ae 63-0f 20 41 75 76 b3 2e 54 ...O.c.AuV..T
00c0 - 68 cd 21 5d 7e 96 34 ba-b6 d4 3b 4e 56 2a 4a 0d h.!~.4..NV.J
00d0 - ff 55 a2 d7 87 14 4a 19-09 a5 3f bb c0 70 fb 15 U...J...?..p.
00e0 - f6 a0 ef 1f 91 08 e6 b0-01 c8 0e 99 0f 86 f1 59 ...L...3...b.
00f0 - 82 bb 4c f9 ee b2 a3 33-a9 af eb 15 f3 62 f1 bb ...L...3...b.
0100 - 4c
```

■ RFC 2816 – Certification Request (CSR)

Atributen: Version, Subject (Public Key), Hash Algorithm, Password zum späteren Entziffern mit dem Private Key, Certificate Requests, Certificate Requests

PKCS#11 – CRYPTO TOKEN INTERFACE (CRYPTOKI)

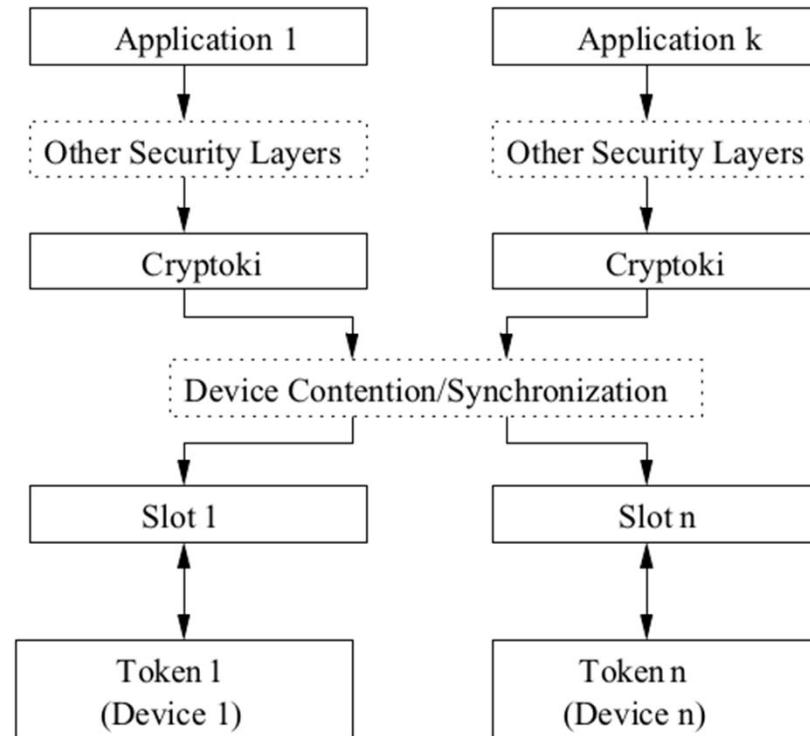
Kein RFC (Version 2.2)

Pflege des Standards
Structured Information

- OASIS PKCS#11 Te

API zur Verwendung von
(Smartcards, Cryptotoki)

- ANSI C (1990) Standard
- Abstrahiert das z



Advancement of

s (HSM), ...)

Anwendung(en)

PKCS#12 – PERSONAL INFORMATION EXCHANGE SYNTAX

Historie

- Version 1 (Juni 1999)
- Version 1.1 (Juli 2014)

Definiert ein portables Format zum Speichern und Transportieren von privaten Schlüsseln, Zertifikaten, Geheimnissen, etc.

Basiert auf Microsofts PFX-Dateiformat (und ist kompatibel dazu)

Verwendet in

Java (Java Key-Store) z.B. Tomcat;

Microsoft: z.B. IIS, Exchange, etc.

Unterstützung aber auch in gängigen Tools (Firefox, Chrome, Thunderbird, ...)

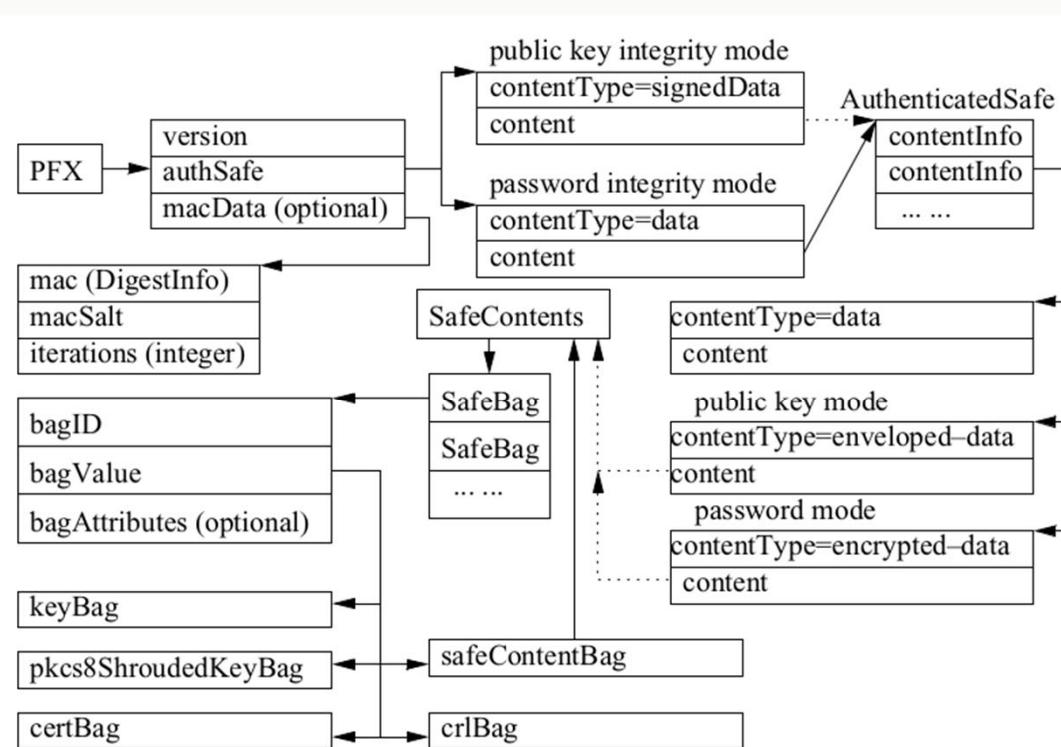
PKCS#12 – PERSONAL INFORMATION EXCHANGE SYNTAX

Lässt sich als “Erweiterung” von PKCS#8 betrachten, gespeicherte private Schlüssel können um weitere Informationen, etwa Zertifikate angereichert werden.

- Verbesserter Schutz der Informationen im Vergleich zu PKCS#8
 - Privacy-Mode / Verschlüsselung mit
 - Passwort
 - Public-Key **EMPFOHLEN**
 - Integrity-Mode / Signatur mit
 - Passwort (Passwort MAC)
 - Digitale Signatur (mit privatem Schlüssel) **EMPFOHLEN**

Der PKCS#12-Sta Verschachtelu

PKCS#12 – PERSONAL INFORMATION EXCHANGE SYNTAX



PKCS#12 – PERSONAL INFORMATION EXCHANGE SYNTAX

Der PKCS#12-Standard ist sehr komplex und erlaubt (fast) beliebige Verschachtelungen von Schlüsselmaterial

In normalen Anwendungsszenarien aber fast ausschließlich ein privater Schlüssel mit entsprechendem Zertifikat (inkl. Zertifikatskette)

Erstellen einer PKCS#12-Datei (Privater Schlüssel + Zertifikatskette)

```
$> openssl pkcs12 -export -in vlnwsi.crt -inkey vlnwsi.key -out vlnwsi.p12
```

PKCS#13 – ELLIPTIC CURVE CRYPTOGRAPHY

Existiert nur als Entwurf (wurde nie veröffentlicht)

Wird nicht weiter gepflegt

Andere Elliptic-Curve-Standa

- ANSI X9.62
 - NIST
 - IEEE
 - BSI
 - ISO
 - SafeCurves
- 

PKCS#14 – PSEUDO RANDOM NUMBER GENERATION

Keine existierenden Dokumente (reserviert)

Zufallszahlen kommen aus

- Pseudo Random Number Generators (PRNG)
 - Deterministische Zahlenfolgen mit guten zufälligen Eigenschaften
 - Guter Zufall, viele Werte
- True Random Number Generator (TRNG)
 - Nutzt (zufällige) physikalische Prozesse / Werte als Zufallswert
 - Echter Zufall, wenige Werte
- Optimaler Weise PRNG mit regelmäßigem TRNG Seed

PKCS#15 – CRYPTOGRAPHIC TOKEN INFORMATION FORMAT

Version 1.1 (Juni 2000; RSA Laboratories)

Kein RFC

Definiert die Datenstruktur auf Cryptotoken

Ziel: Interoperabilität über Software- und Hardware-Grenzen hinweg

Vier unterschiedliche Objekttypen

- Schlüssel
- Authentifikationsobjekte
- Zertifikate
- Datenobjekte

Objekte können privat oder öffentlich sein

Zugriff geschützt durch biometrische oder wissensbasierte Verfahren (z.B. PIN)

PKCS#15 – CRYPTOGRAPHIC TOKEN INFORMATION FORMAT

MF = Master File (root)

DF = Dedicated File (Verzeichnisse)

EF = Elementary File (Dateien)

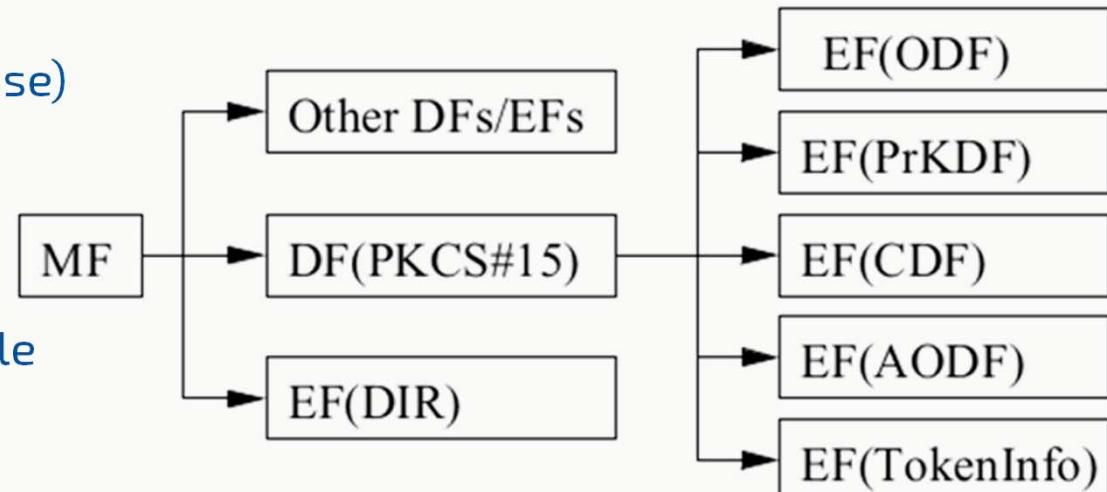
ODF = Object Directory File

PrKDF = Private Key Directory File

CDF = Certificate Directory File

AODF = Authentication Object Directory File

TokenInfo = Informationen über das Token (z.B. Seriennummer, unterstützte Dateitypen, implementierte Algorithmen)



PKCS#15 – CRYPTOGRAPHIC TOKEN INFORMATION FORMAT

\$ > pkcs15-tool --dump

PIN

Private Key

Zertifikat

```
AODF:  
    Com. Flags  : private, modifiable  
    Auth ID    : 01  
    Flags      : [0x32], local, initialized, needs-padding  
    Length     : min_len:4, max_len:8, stored_len:8  
    Pad char   : 0x00  
    Reference  : 1  
    Encoding   : ASCII-numeric  
    Path       : 3F005015  
  
PrKDF:  
    Com. Flags  : private, modifiable  
    Com. Auth ID: 01  
    Usage       : [0x32E], decrypt, sign, signRecover, unwrap, derive, nonRep  
    Access Flags: [0x1D], sensitive, alwaysSensitive, neverExtract, local  
    ModLength   : 1024  
    Key ref    : 0  
    Native      : yes  
    Path        : 3F00501530450012  
    ID          : 45  
  
X.509 Certificate [/C=BE/ST=...]  
    Com. Flags  : modifiable  
    Authority  : no  
    Path       : 3f0050154545  
    ID          : 45
```

Digitale Signaturen:

- Ein Unterzeichner signiert eine Nachricht so, dass jeder überprüfen kann, dass die Nachricht nur vom Unterzeichner und sonst niemandem verändert werden konnte
- Message-Digest- und Public-Key-Algorithmen zum Hashen und Signieren des Hashes.
- Spezifikationen
 - Message-Digest-Algorithmen (PKCS#1)
 - Public-Key-Algorithmen (PKCS#1, PKCS#3, PKCS#13 [Entwurf])
 - Algorithmenunabhängige Syntax für digital signierten Nachrichten (PKCS#7)

Digitale Signaturen:

- Ein Unterzeichner signiert eine Nachricht so, dass jeder überprüfen kann, dass die Nachricht nur vom Unterzeichner und sonst niemandem verändert werden konnte
- Message-Digest- und Public-Key-Algorithmen zum Hashen und Signieren des Hashes.

Spezifikationen

- Syntax für private Schlüssel (PKCS#1, PKCS#8)
- Syntax für verschlüsselte private Schlüssel (PKCS#8)
- Methoden zum Ableiten geheimer Schlüssel aus Passwörtern (PKCS#5)

Digitale Umschläge

- Ein Absender versiegelt eine Nachricht, dass nur der Empfänger diese öffnen kann. Die Nachricht ist verschlüsselt mit einem geheimen Schlüssel und dieser Schlüssel ist verschlüsselt mit dem öffentlichen Schlüssel des Empfängers

Spezifikationen

- Algorithmenunabhängige Syntax für digitale Umschläge (PKCS#7)
- Syntax für private Schlüssel (PKCS#1, PKCS#8)
- Syntax für verschlüsselte private Schlüssel (PKCS#8)
- Methoden zum Ableiten geheimer Schlüssel aus Passwörtern (PKCS#5)

Digitale Zertifikate

- Eine Zertifizierungsstelle (CA) signiert eine spezielle Nachricht, die mindestens den Namen und den öffentlichen Schlüssel einer Person enthält so, dass jeder verifizieren kann, dass diese spezielle Nachricht nur von der CA verändert wurde und der öffentliche Schlüssel somit vertrauenswürdig ist. Die spezielle Nachricht wird Zertifikatanfrage (Certificate Signing Request; CSR) genannt und wird mit einem digitalen Signaturalgorithmus signiert.

Spezifikationen

- Algorithmenunabhängige Syntax für Zertifikatanfragen (PKCS#10)
- Syntax für öffentliche Schlüssel (PKCS#1)
- Spezifische Signaturalgorithmen (PKCS#1)

Schlüsseltausch

- Zwei Kommunikationspartner einigen sich auf einen gemeinsamen geheimen Schlüssel, ohne vorherige Absprachen. Typischerweise gibt es dafür Algorithmen mit zwei Phasen: Ein Kommunikationspartner initiiert den Schlüsseltausch in der ersten Phase. Anschließend tauschen beide Partner das Ergebnis der ersten Phase aus und berechnen in der zweiten Phase den gemeinsamen geheimen Schlüssel

Spezifikationen

- Algorithmenunabhängige Syntax für Nachrichten zum Schlüsseltausch (PKCS#3)
- Spezifische Algorithmen zum Schlüsseltausch (PKCS#3)

Verschlüsselung und Signatur von E-Mails

- Ursprünglich von RSA Data Security (PKCS#7) entworfen
- Mittlerweile Standard der Internet Engineering Task Force (IETF)
- PKCS#7 wird zu Cryptographic Message Syntax
- Mehrere RFCs mit diversen Weiterentwicklungen
 - RFC 2311 - Version 2 (März 1998)
 - RFC 2633 – Version 3 (Juni 1999)
 - ...
 - RFC 8551 – Version 4 (April 2019)

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

Zertifikatshandling und Sperrlisten (S/MIME Certificate Handling)

Historie

- RFC 2632 – Version 3 (Juni 1999)
- ...
- RFC 8550 – Version 4 (April 2019)

HIER NICHT WEITER RELEVANT

Schutzziele

- Zurechenbarkeit
- Integrität
- Nicht-Abstreitbarkeit
- Privacy / Datenschutz
- Vertraulichkeit

Zwei unterschiedliche „Güteklassen“ bei der Zertifikatausstellung

- Klasse 1: Die Absender-E-Mail-Adresse wird überprüft
- Klasse 2: Die Person hinter der Absender-E-Mail-Adresse wird überprüft

Best practices

- Verwendung unterschiedlicher Schlüssel für Verschlüsselung und Signatur
- Obwohl eine Nachricht nur für den Empfänger verschlüsselt wird, wird zumeist ein eigenes Schlüsselpaar gefordert.
 - Dieses wird benötigt, wenn der Absender die E-Mail später noch einmal lesen möchte (und sie verschlüsselt im Ordner Gesendet liegt)
- In der Regel werden die Zertifikatinformationen von der CA öffentlich verfügbar gemacht (z.B. über Verzeichnisdienste)

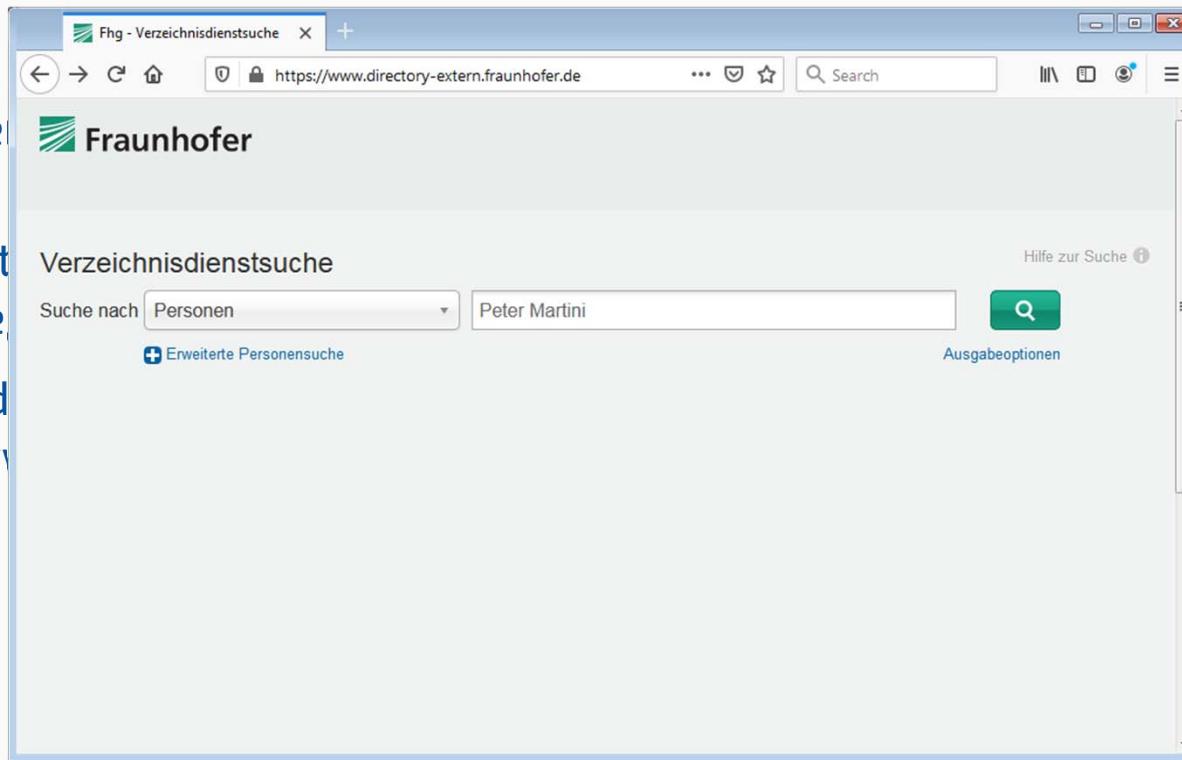
Nachteile von S/MIME

- Keine Überprüfung des E-Mail-Inhalts durch SPAM-Filter oder Virenscanner möglich
 - Alternativ: Der private Schlüssel der Benutzer (aller?!) muss für die Prüfung hinterlegt werden
- E-Mail-Adressen sind öffentlich verfügbar (Beispiel: Fraunhofer Directory)
<https://www.directory-extern.fraunhofer.de/>

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

Nachteile von

- Keine Übertragung möglich
 - Alternativer hinterlegter E-Mail-Adresse
- E-Mail-Adresse über S/MIME <https://www.directory-extern.fraunhofer.de>

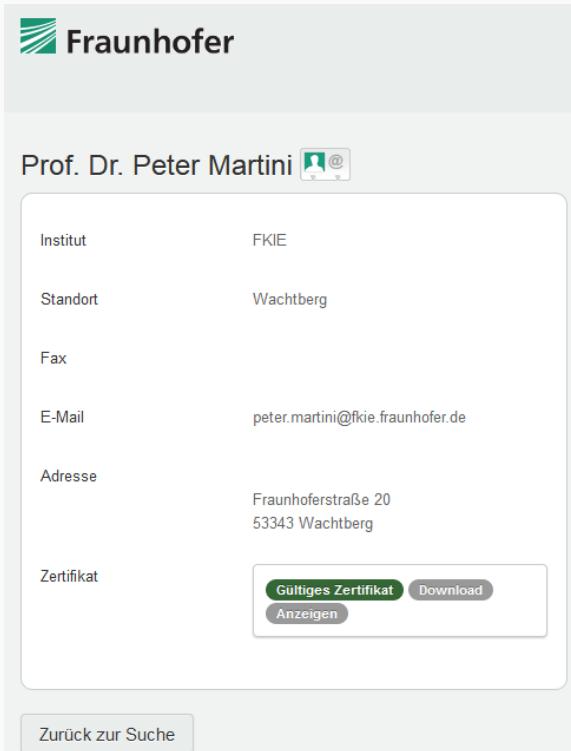


enscanner

die Prüfung

directory)

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

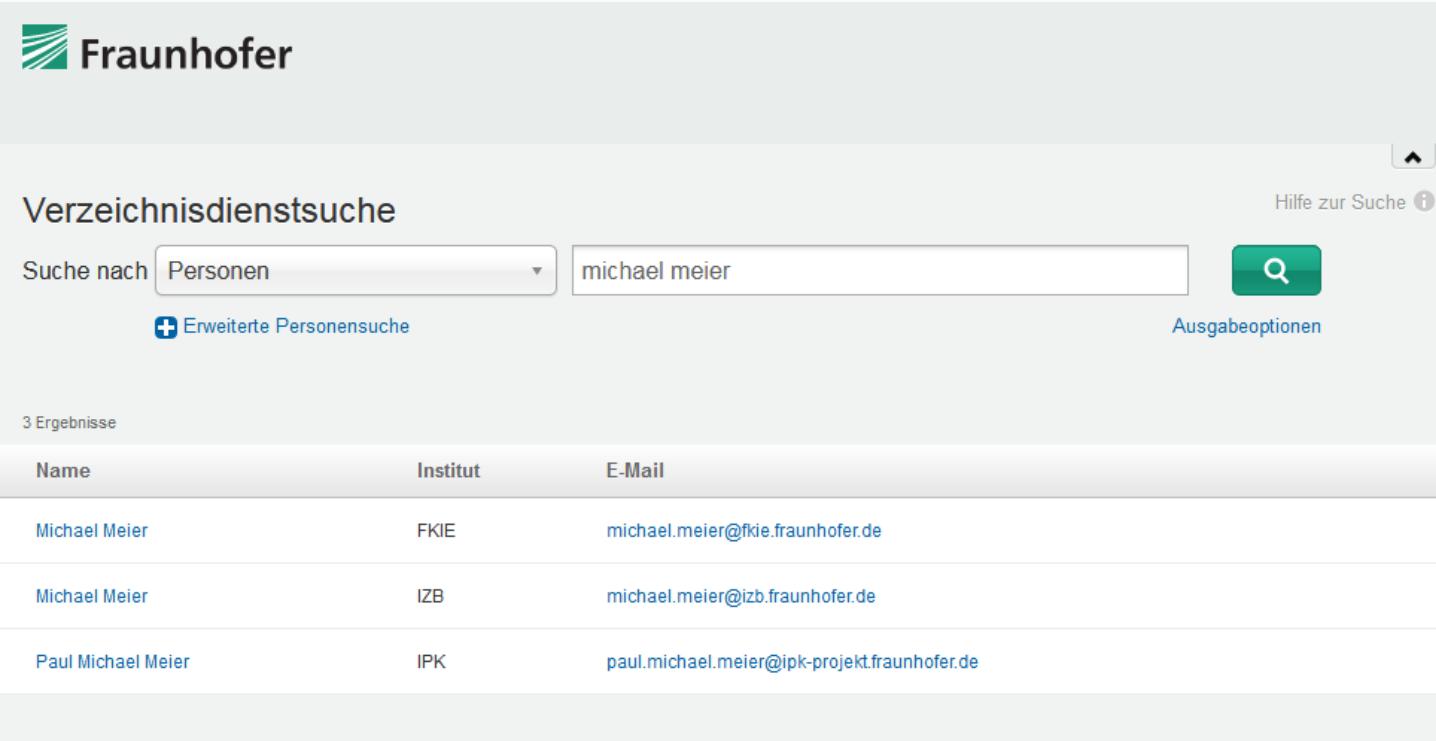


The screenshot shows a contact form for Prof. Dr. Peter Martini. At the top, there is a logo for Fraunhofer and the name "Prof. Dr. Peter Martini" followed by an email icon. Below this, there are several input fields with labels and corresponding values:

Institut	FKIE
Standort	Wachtberg
Fax	
E-Mail	peter.martini@fkie.fraunhofer.de
Adresse	Fraunhoferstraße 20 53343 Wachtberg
Zertifikat	Gültiges Zertifikat Download Anzeigen

At the bottom of the form is a button labeled "Zurück zur Suche".

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)



Fraunhofer

Verzeichnisdienstsuche Hilfe zur Suche ⓘ

Suche nach Personen 🔍

+ Erweiterte Personensuche Ausgabeoptionen

3 Ergebnisse

Name	Institut	E-Mail
Michael Meier	FKIE	michael.meier@fkie.fraunhofer.de
Michael Meier	IZB	michael.meier@izb.fraunhofer.de
Paul Michael Meier	IPK	paul.michael.meier@ipk-projekt.fraunhofer.de

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

- Anwendungsbeispiele
 - Bei der Universität Bonn:
 - HRZ betreibt eine CA für die Universität Bonn über die DFN-PKI
<https://pki.pca.dfn.de/dfn-ca-global-g2/pub/>
 - Nutzbar für Client/User-Zertifikate
 - Nutzbar für Server-Zertifikate (Uni-Dienste)

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

■ Anwendung

Zertifikatdaten

Apply here for a new certificate.



■ Bei der Ur

- HRZ bei
<https://>

- Nutzbar
- Nutzbar

Certificate profile

The chosen "Certificate profile" determines the possible usages of the certificate. ([Description of certificate profiles \[German\]](#))

Neuer Antrag

Create certificate request

The following data is used to create a new certificate request.

▼ Email addresses with domain names from this list can be used without further confirmation. Email addresses with all other domain names must be confirmed separately.

Name (CN)

Enter your first and last name(s) here. Do not use umlauts and diacritics. For group certificates, use prefix 'GRP:' or 'GRP - '. For pseudonym cer

Email

Email address

Organisational unit (OU, optional)

If you specify an organisational unit here, it will be included as OU-attribut in the certificate name.

Namespace (The chosen namespace will be used to complete the final certificate name.)

O=Rheinische Friedrich-Wilhelms-Universitaet Bonn,L=Bonn,ST=Nordrhein-Westfalen,C=DE

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

Ihre Daten

Enter further data below. What you enter here will not be found in the certificate.

Revoke-PIN

Revoke-PIN - at least 8 character

Revoke-PIN - Confirmation

Please enter the revoke-PIN again to confirm.

This PIN is required if you want to revoke your certificate. Please make a note of this PIN.

Personal note (optional)

You may enter an optional note for this certificate application here. The comment will solemnly be saved in your local certificate application data file.

Personal note (optional)

- I am committed to comply with the regulations contained in [Informationen für Zertifikatinhaber](#).
- Optional: I agree to the publication of the certificate with the contained names and e-mail addresses.
You can withdraw this agreement by sending an e-mail to pki@dfn.de.

Next

■ Anwen

Ihre Daten

Enter further data below. What you enter here will not be found in the certificate.

■ Bei d

Revoke-PIN

Revoke-PIN - at least 8 character

Revoke-PIN - Confirmation

Please enter the revoke-PIN again to confirm.

■ Nu

Personal note (optional)

You may enter an optional note for this certificate application here. The comment will solemnly be saved in your local certificate application data file.

Personal note (optional)

I am committed to comply with the regulations contained in [Informationen für Zertifikatinhaber](#).

Optional: I agree to the publication of the certificate with the contained names and e-mail addresses.

You can withdraw this agreement by sending an e-mail to pki@dfn.de.

Next

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

- Anwendungsbeispiele
 - Bei der Universität Bonn:
 - HRZ betreibt eine CA für die Universität Bonn über die DFN-PKI
<https://pki.pca.dfn.de/dfn-ca-global-g2/pub/>
 - Nutzbar für S/MIME-Zertifikate
 - Nutzbar für x509-Zertifikate für Webserver / Uni-Dienste
 - Volksverschlüsselung
 - Betrieb durch Fraunhofer SIT
<https://volksverschluesselung.de>
 - Klasse-2-Zertifikate (Überprüfung durch Person)

LÄUFT NUR
UNTER WINDOWS

■ Anwendungen

■ Bei der Polizei

- HRZ Polizei Berlin
<https://www.hrz.polizei.de>

■ Nutzen

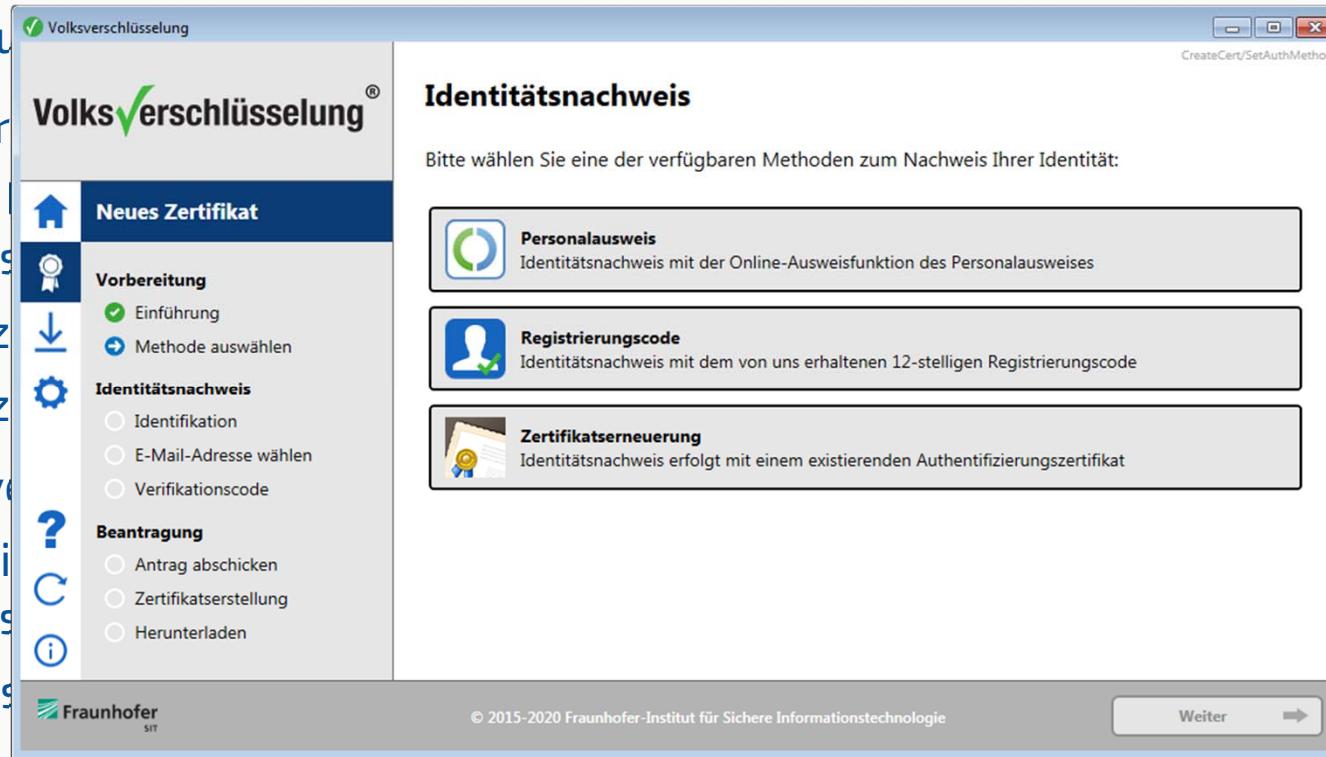
■ Nutzen

■ Volksverschlüsselung

■ Betrieb

<https://www.volksverschlusselung.de>

■ Klassische



SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

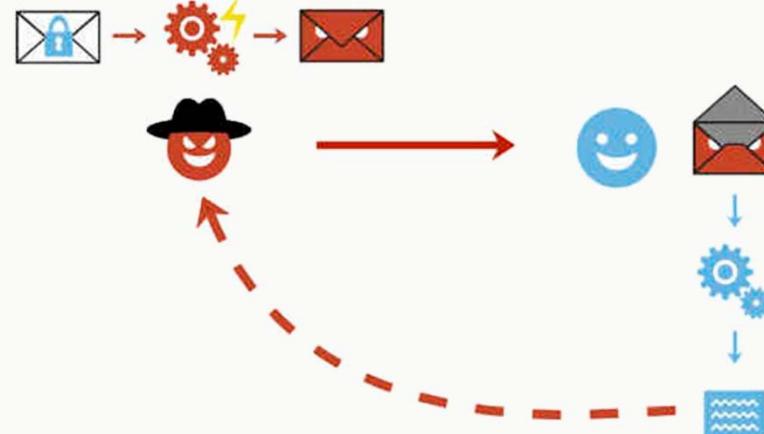
- Angriffe gegen S/MIME

- E-Fail

- CVE-2017-17688 , CVE-2017-17689

- Änderung des Ciphertexts, so dass die Nachricht nach dem Entschlüsseln an den Angreifer gesendet wird

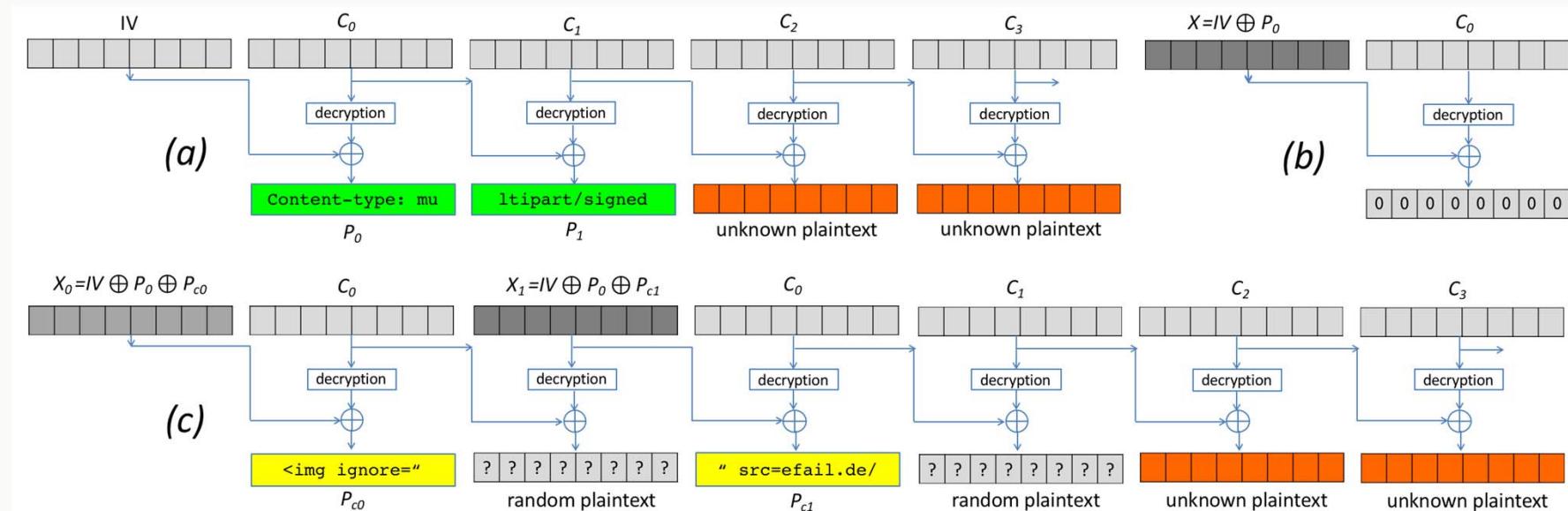
- CBC/CFB Gadget Attack



SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

- Angriffe gegen S/MIME

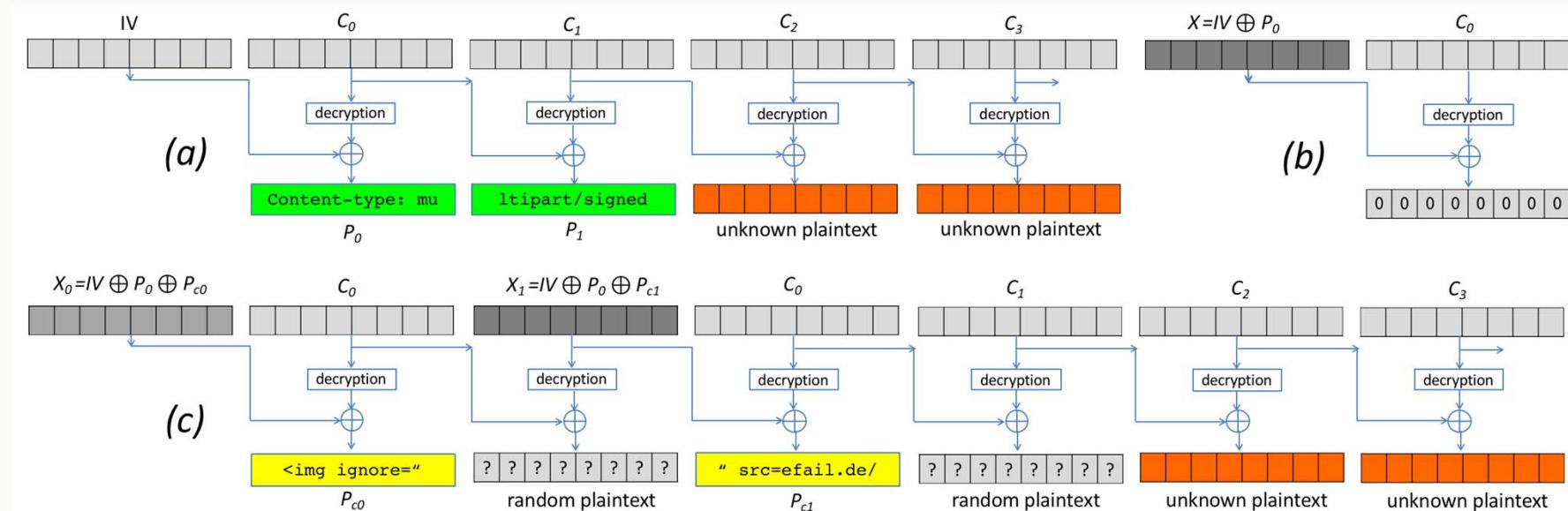
- E-Fail



SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

- Angriffe gegen S/MIME

- E-Fail



SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

```
From: attacker@efail.de
To: victim@company.com
Content-Type: multipart/mixed;boundary="BOUNDARY"

--BOUNDARY
Content-Type: text/html


--BOUNDARY--
```

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

```
From: attacker@efail.de  
To: victim@company.com  
Content-Type: multipart/mixed;boundary="BOUNDARY"
```

```
--BOUNDARY
```

```
Content-Type: text/html
```

```

```

```
--BOUNDARY
```

```
Content-Type: text/html
```

```
">
```

```
--BOUNDARY--
```

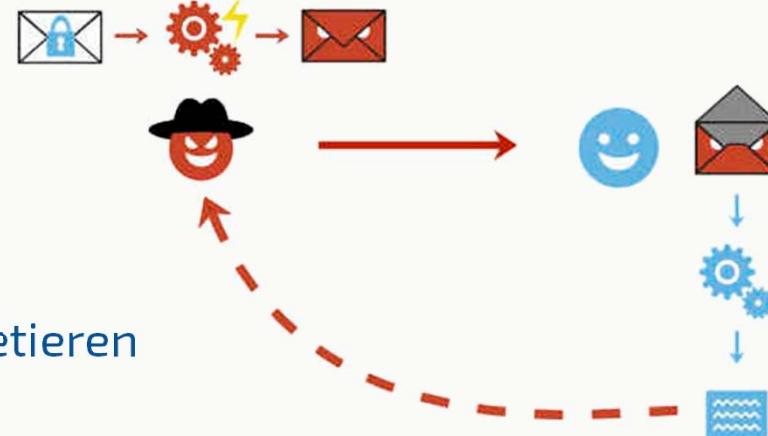
SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

- Angriffe gegen S/MIME

- E-Fail

- CVE-2017-17688 , CVE-2017-17689

- Änderung des Ciphertexts, so dass die Nachricht nach dem Entschlüsseln an den Angreifer gesendet wird



- CBC/CFB Gadget Attack

- Direct Exfiltration

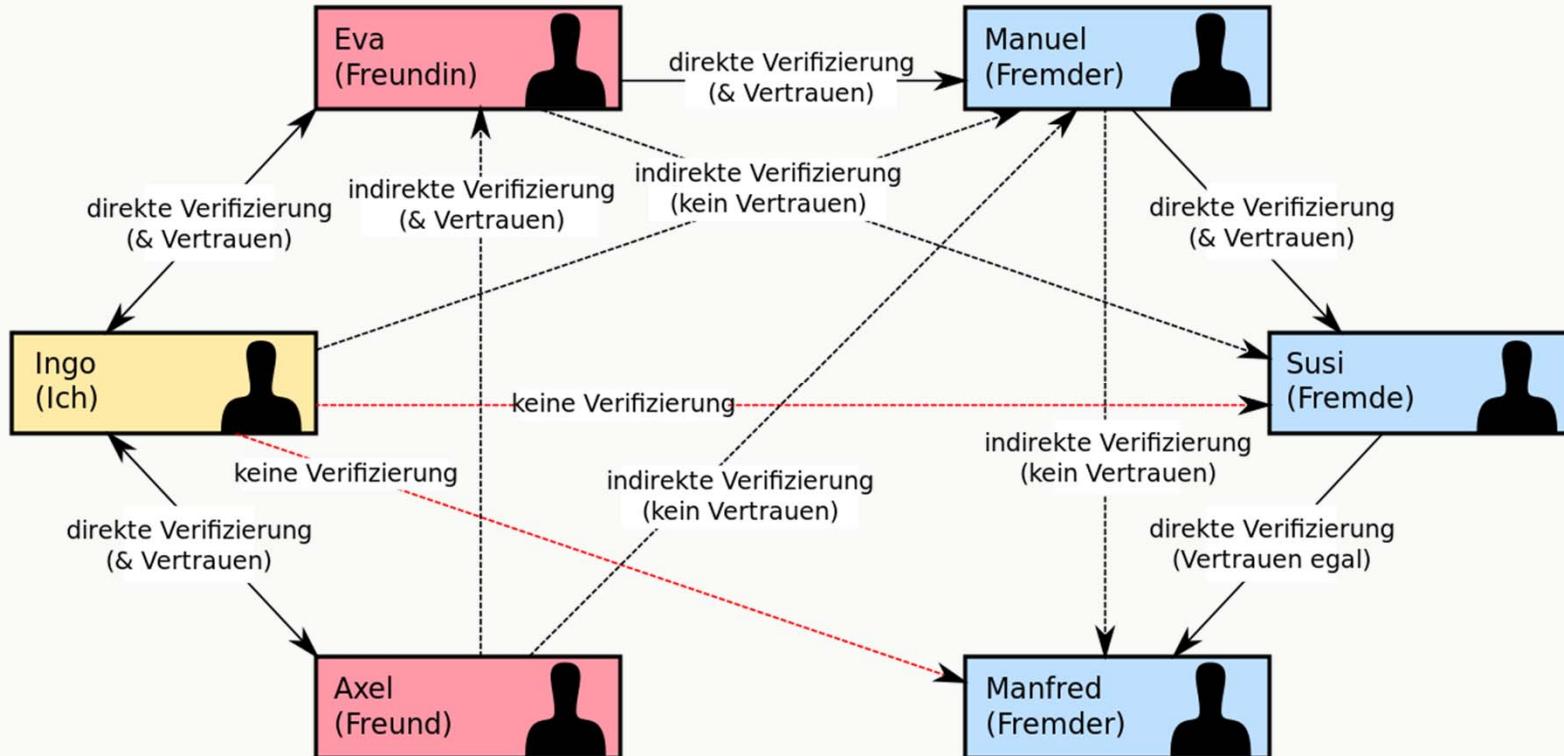
- Problem: Aktive Inhalte

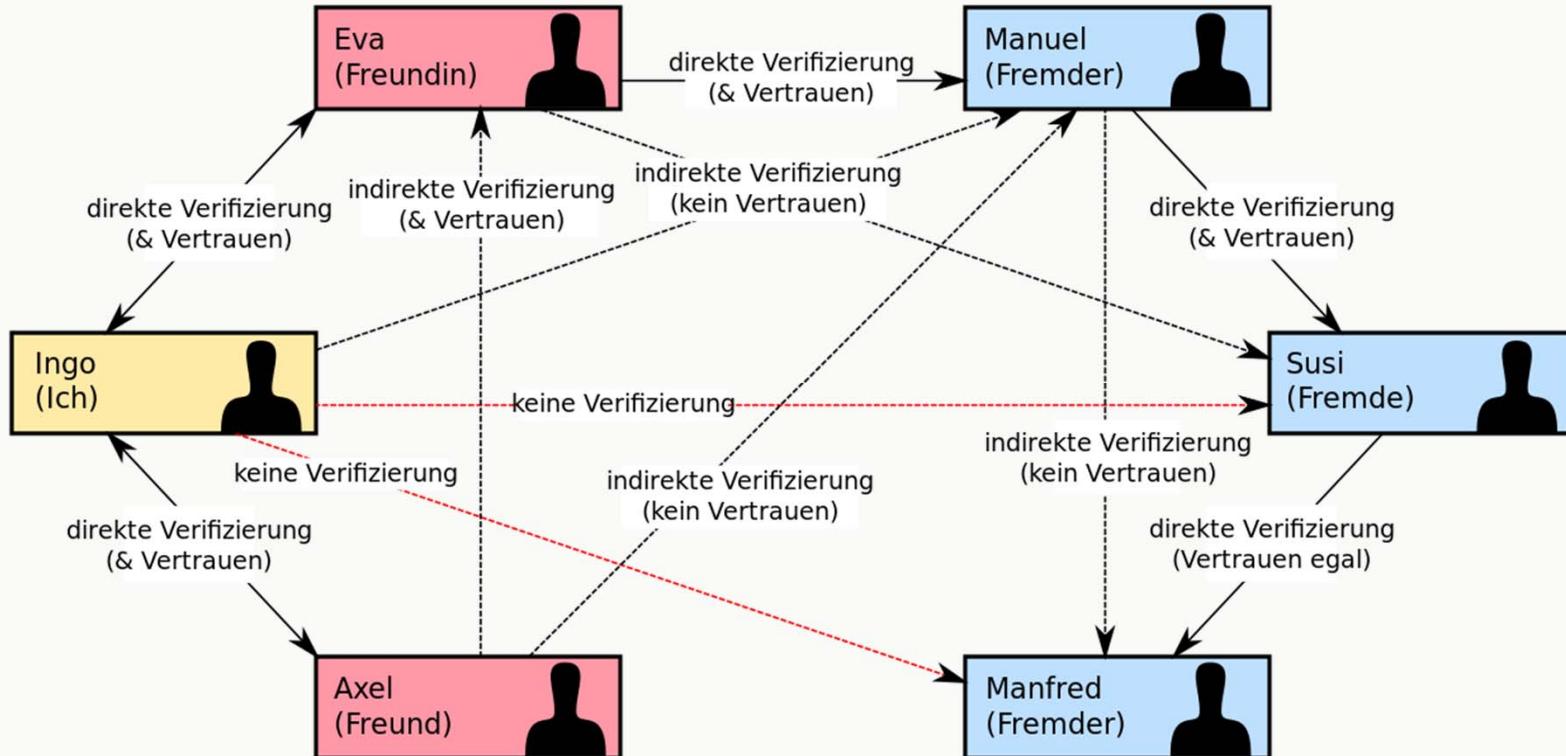
- Skripte ausführen / HTML interpretieren

- Bilder nachladen

- RFC 4880 (OpenPGP)
 - Basiert auf dem kommerziellen Produkt „Pretty Good Privacy (PGP)“
 - Ähnlich wie S/MIME, hybride Verschlüsselung

	S/MIME	OpenPGP	PGP 5.x
Asymmetrische Verschlüsselung	RSA	RSA, ElGamal, Elliptic Curves, Diffie-Hellman	RSA, ElGamal
Asymmetrische Signatur	RSA	RSA, DSA, ElGamal, ECDSA	RSA, DSA
Symmetrische Algorithmen	TripleDES, DES, RC2	TripleDES, CAST5, IDEA, Blowfish, SaferSK128, Twofish	TripleDES, IDEA, CAST5
Hash-Algorithmen	MD5, SHA-1	MD5, SHA-1, RIPE/MD-160, MD2, Double-width SHA	MD5, SHA-1





- RFC 4880 (OpenPGP)
 - Basiert auf dem kommerziellen Produkt „Pretty Good Privacy (PGP)“
 - Ähnlich wie S/MIME, hybride Verschlüsselung, gleiche Schutzziele
 - Unterstützt mehr Algorithmen als S/MIME und PGP
 - Web-of-Trust statt hierarchischer PKI
 - Programmier-API seit Version 2.0 (libgcrypt)
- Ebenfalls, wie S/MIME, verwundbar gegen Efail!
 - Update: Modification Detection Codes (MDC) verhindern Änderungen des Ciphertexts

Vielen Dank für die Aufmerksamkeit!

Fragen?

Nächste Vorlesung:

- Montag, 31. Mai 2021

Nächste Übung:

- Dienstag, 18. Mai 2021 – 16 Uhr
- Abgabe des Übungszettels bis morgen – 16 Uhr