Kryptographie – eine „doppelte Herausforderung“ für die Lehre

Univ.-Prof. Priv.Doz. Dipl.Ing. Dipl.Ing. Dr. Stefan Rass

Kryptographie ist alltäglich: ID Austria, https://...

Quelle: www.oesterreich.gv.at

BUNDESGESETZBLATT
FÜR DIE REPUBLIK ÖSTERREICH

Jahrgang 2008 **Ausgegeben am 7. Jänner 2008** **Teil II**

3. Verordnung: **Signaturverordnung 2008 – SigV 2008**

3. Verordnung des Bundeskanzlers über elektronische Signaturen (Signaturverordnung 2008 – SigV 2008)

Auf Grund des § 25 des Signaturgesetzes, BGBl. I Nr. 190/1999, zuletzt geändert durch das Bundesgesetz BGBl. I Nr. 8/2008, wird im Einvernehmen mit der Bundesministerin für Justiz verordnet:

Gebühren für Aufsichtstätigkeiten

§ 1. (1) Von den Zertifizierungsdiensteanbietern (ZDA) sind für Leistungen im Rahmen der Aufsichtstätigkeit folgende Gebühren zu entrichten:

1. Prüfung des Sicherheits- und Zertifizierungskonzepts anlässlich der Aufnahme

Quellen: https://www.ris.bka.gv.at/

§ 16. Diese Verordnung samt Anhang tritt mit Kundmachung in Kraft. Gleichzeitig tritt die Verordnung des Bundeskanzlers über elektronische Signaturen, BGBl. II Nr. 30/2000, in der Fassung der Verordnung BGBl. II Nr. 527/2004, samt Anhang außer Kraft.

Gusenbauer

ANHANG

Algorithmen und Parameter für qualifizierte elektronische Signaturen

1. Definitionen

1. Signatursuite: Eine Signatursuite besteht aus folgenden Komponenten:
- einem Signaturalgorithmus mit Parametern,
- einem Algorithmus zur Schlüsselerzeugung,
- einem Padding-Verfahren und
- einer kryptographischen Hashfunktion.

2. Bitlänge: Die Bitlänge einer natürlichen Zahl p ist r, wenn $2^{r-1} \leq p < 2^r$ gilt.

3. Kryptographische Hashfunktion: Der Algorithmus „Hash-Funktion" ist eine nicht umkehrbare Funktion, die eine umfangreiche Datenmenge (in der Regel einen Text) auf eine im Allgemeinen wesentlich kleinere Zielmenge fester Länge (Hash-Wert) abbildet.

2. Abkürzungen

A9C	„Article 9 Committee" (Ausschuss für elektronische Signaturen gemäß Art. 9 der Richtlinie 1999/93/EG)
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ECGDSA	Elliptic Curve German Digital Signature Algorithm
RSA	Verfahren von Rivest, Shamir und Adleman

3. Zulässige Signatursuiten

Algorithmen und Parameter für qualifizierte elektronische Signaturen dürfen nur in vordefinierten Kombinationen verwendet werden, die als Signatursuiten bezeichnet werden.

Falls eine Komponente der Suite ungültig ist, ist auch die gesamte Suite ungültig. Falls eine Komponente der Suite aktualisiert worden ist, ist auch die gesamte Suite zu aktualisieren.

Tabelle 1a – Liste der zulässigen Signatursuiten:

Kennzahl des Signatursuite-Eintrags	Signatur-Algorithmus	Parameter des Signaturalgorithmus	Algorithmus zur Schlüsselerzeugung	Padding-Verfahren	Kryptographische Hashfunktion
001	rsa	MinModLen = 1024	rsagen1	siehe Tabelle 3	siehe Tabelle 2
002	dsa	pMinLen = 1024 qMinlen = 160	dsagen1	-	siehe Tabelle 2
		qMinLen = 160			

DSA … digital signature algorithm → Programmier-Aufgabe zum besseren Verständnis

Datenschutz, Kryptographie und aktuelle Bedrohungen

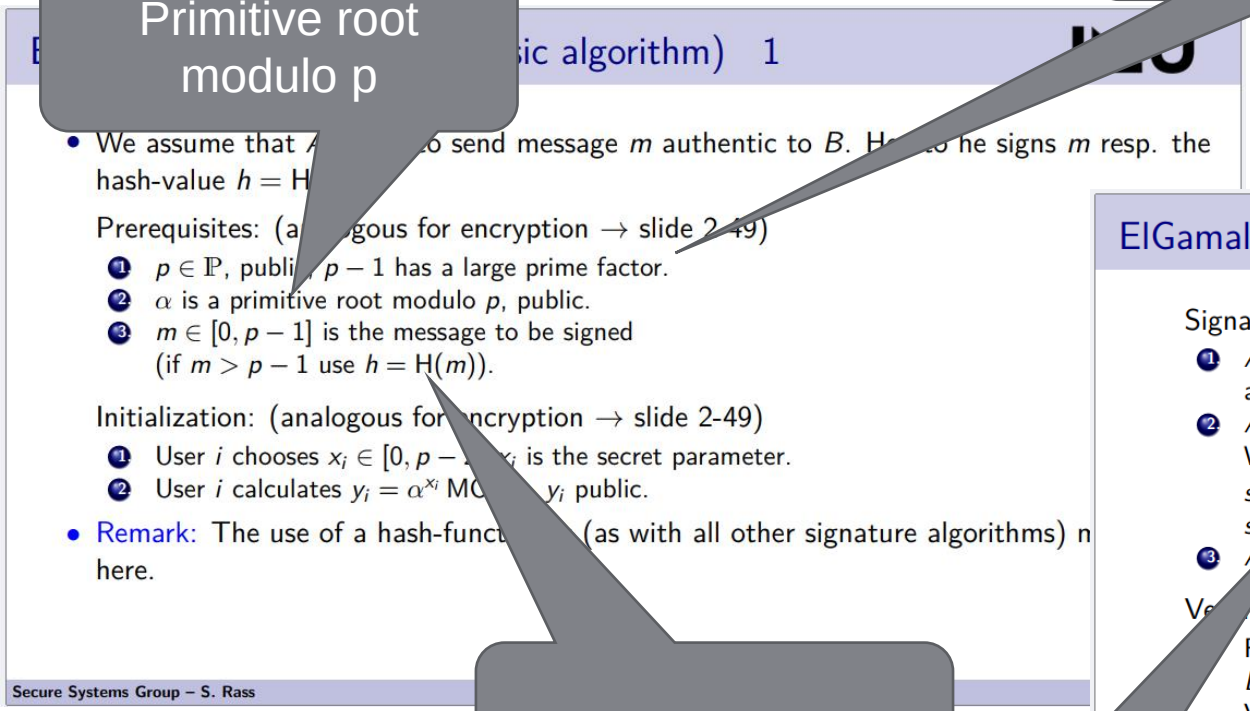
- Kryptographie:
  - Wichtigkeit nimmt zu wegen Angriffen durch Seitenkanäle, Quantencomputer, …
  - Interesse bisweilen groß
- Schwierig zu unterrichten, denn es erfordert:
  - Gute Programmierkenntnisse
  - Und tiefe mathematische Grundlagen
    $\rightarrow$ spielerischer Zugang (try-and-error) nur sehr schwer möglich

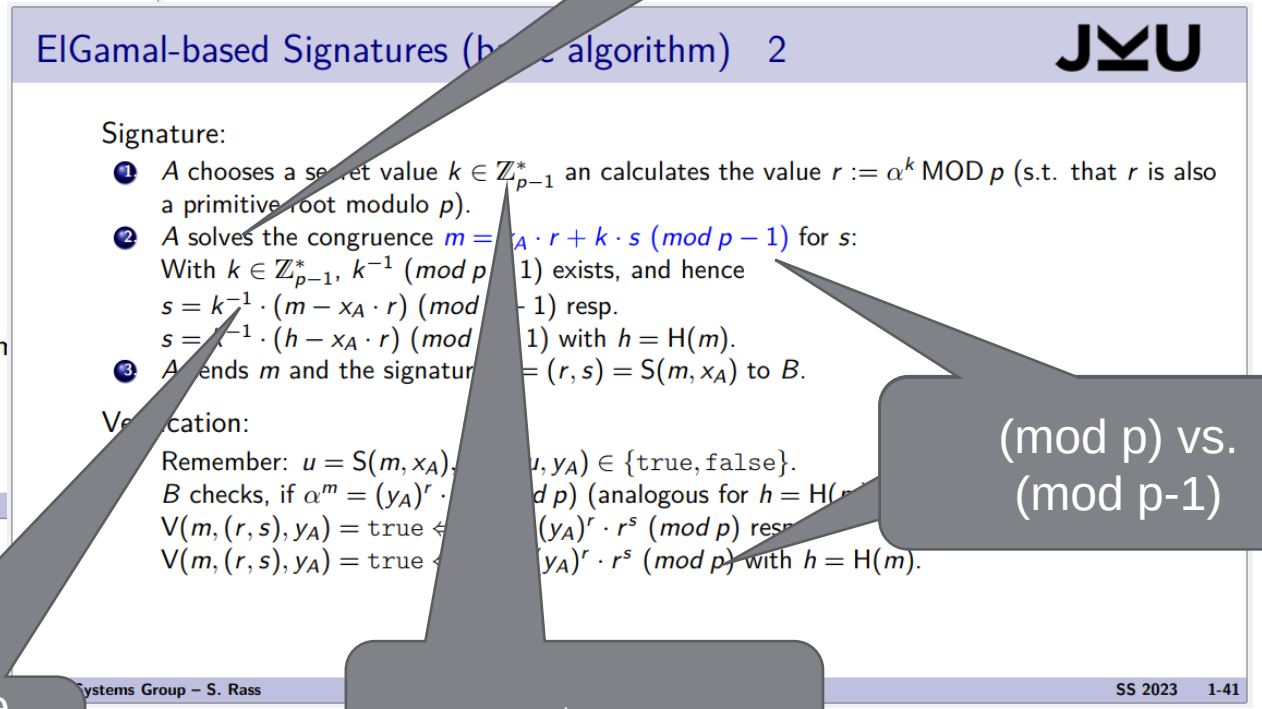
Digitales Signaturverfahren (im Detail)

„Large" prime factor

„solve the congruence“

Primitive root modulo p





Hash function

(mod p) vs. (mod p-1)

k^{-1}, aber keine „klassische“ Division!

$\mathbb{Z}^*_{p-1}$

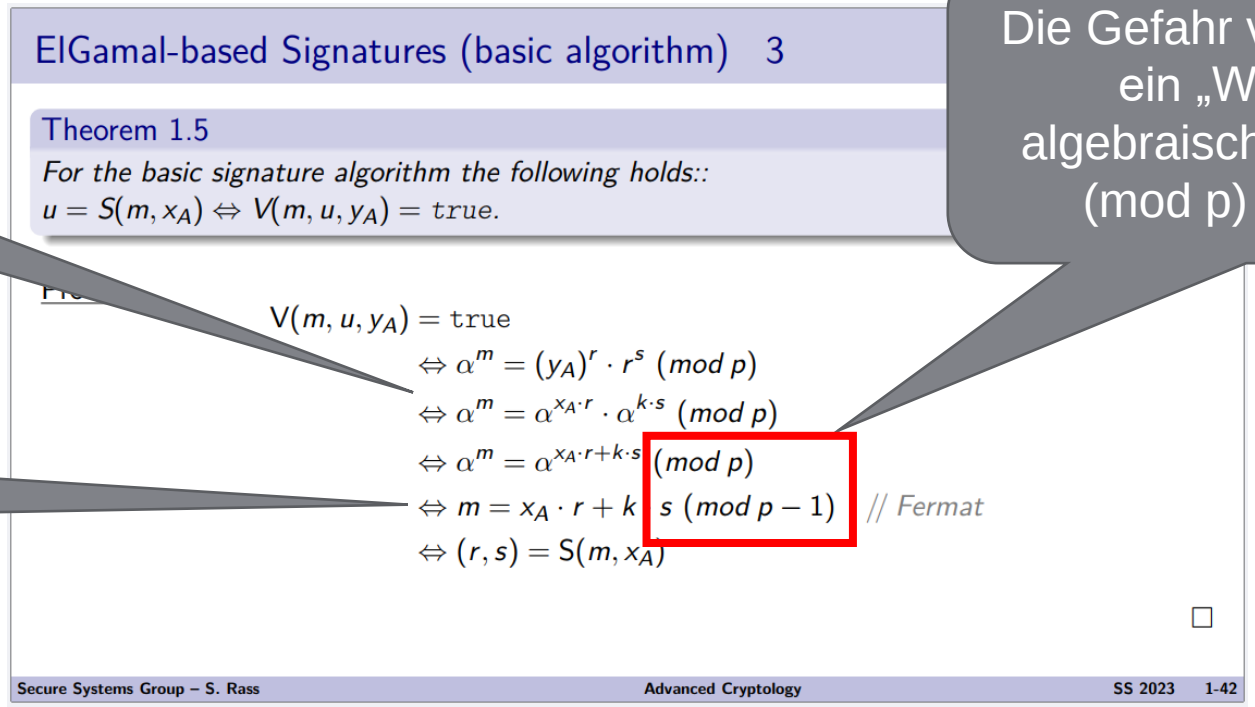
Die Mathematik scheint einfach, die Implementierung ebenfalls … beider Schein trügt!

- Die Korrektheit des Signaturverfahrens lässt sich sehr leicht nachrechnen…aber nicht so leicht nach-programmieren!

Nur Einsetzen + Potenzrechenregeln

+ 1x ein „Logarithmus“

Die Gefahr verbirgt sich hier: ein „Wechsel“ der algebraischen Struktur von (mod p) → (mod p-1)



ElGamal-based Signatures (basic algorithm) 3

Theorem 1.5

For the basic signature algorithm the following holds::
$u = S(m, x_A) \Leftrightarrow V(m, u, y_A) = \texttt{true}$.

$$
\begin{aligned}
V(m, u, y_A) = \texttt{true} & \\
\Leftrightarrow \alpha^m &= (y_A)^r \cdot r^s \pmod p \\
\Leftrightarrow \alpha^m &= \alpha^{x_A \cdot r} \cdot \alpha^{k \cdot s} \pmod p \\
\Leftrightarrow \alpha^m &= \alpha^{x_A \cdot r + k \cdot s} \pmod p \\
\Leftrightarrow m &= x_A \cdot r + k \cdot s \pmod{p-1} \quad \text{// Fermat} \\
\Leftrightarrow (r, s) &= S(m, x_A)
\end{aligned}
$$

□

Secure Systems Group – S. Rass | Advanced Cryptology | SS 2023 | 1-42

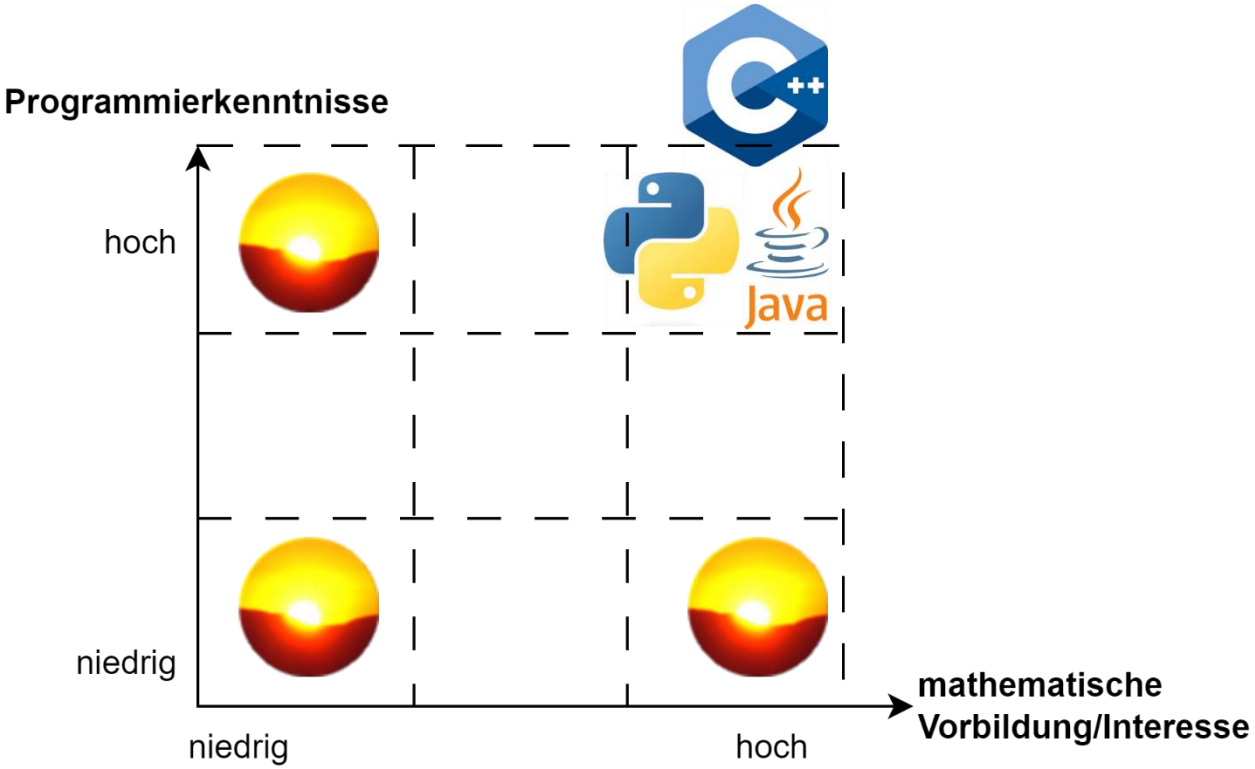
Unser Ansatz um Kryptographie zu lehren

- Das System „Sunset/FFapl“

  - FFapl = die Programmiersprache
  - Sunset = die Entwicklungsumgebung

Was du mir sagst, das vergesse ich.
Was du mir zeigst, daran erinnere ich mich.
Was du mich tun lässt; das verstehe ich.

Konfuzius (551 - 479 v. Chr.)



https://github.com/stefan-rass/sunset-ffapl

Universelles Fälschen – „Learning by Doing“

- Gegeben: eine (beliebige) digitale Signatur zu einem vorgegebenen Dokument m_1
- Gesucht: eine ebenso gültige Signatur zu einem beliebig gefälschten Dokument m_2
- Sofern die Systemparameter nicht authentifiziert werden, ist das möglich! → um zu verstehen …hnt es sich den Angriff selber durchzuspielen

Der Angriff („nur 3 Formeln“)

JKU

- … by mani…g the used primitive root (changing α to β):
  (r_1, s_1) is a …lGamal-Signature for message m_1 (with primitive root α). Now a signature (with prim…tive root β) for m_2 can be constructed: Determine the following values and exchange α by β.
  $t := (m_2 - m_1) \cdot s_1^{-1} \text{ MOD}(p-1) \Rightarrow m_2 = (m_1 + t \cdot s_1) \text{ MOD}(p-1)$
  $v := \alpha^t \text{ MOD } p, \beta := \alpha^v \text{ MOD } p$
- Remarks:
  - Calculating v and t is always possible. The only thing to know is a signature (r_1, s_1) to some message m_1.
  - β is a primitive root mod p, iff $\gcd(v, p-1) = 1$. The probability for that is about $6/\pi^2$.

Secure Systems Group – S. Rass | Advanced Cryptology | SS 2023 | 1-48

Universal Forgery 2

JKU

Lemma 1.8

$(r_2, s_2) = (r_1 \cdot v, s_1 \cdot v)$ is an ElGamal-Signature for message m_2, whereby the primitive root β has to be used!

Proof: … Remember that $r = \alpha^k \pmod p, \ldots, m = x \cdot r + k \cdot s \pmod{p-1}$. It is sufficient to show that the verification (with β) does work.

$$\begin{aligned} y_A^{r_2} \cdot r_2^{s_2} &= \alpha^{x_A \cdot r_1 \cdot v} (r_1 \cdot v)^{s_1 \cdot v} = \alpha^{v \cdot (x_A \cdot r_1 + k_1 \cdot s_1)} \cdot v^{s_1 \cdot v} \\ &= \beta^{m_1} \cdot \alpha^{t \cdot s_1 \cdot v} = \beta^{m_1} \cdot \beta^{t \cdot s_1} = \beta^{m_1 + t \cdot s_1} \\ &= \beta^{m_2} \pmod p \end{aligned}$$

Die Begründung (3 Zeilen einfacher Umformungen)

Secure Systems Group – S. Rass | Advanced Cryptology | SS 2023 | 1-49

Kryptographie – eine „doppelte Herausforderung“ für die Lehre

https://github.com/stefan-rass/sunset-ffapl

Frei verfügbar unter GPL

Univ.-Prof. Priv.Doz. Dipl.Ing. Dipl.Ing. Dr. Stefan Rass