Digitale Signaturen

Sven Tabbert

Inhalt:

Digitale Signaturen

1. Einleitung
2. Erzeugung Digitaler Signaturen
3. Signaturen und
   Einweg Hashfunktionen
4. Digital Signature Algorithmus
5. Zusammenfassung

1 Einleitung

- Verlegung des Informationsflusses auf die elektronischen Medien
- Vergleichbares entwickeln zur traditionellen Unterschrift ➔ Digitale Signatur

1 Einleitung

Zwei Komponenten: Signierungsalgorithmus & Verifikationsalgorithmus

Formale Definition einer Digitalen Signatur:

1. P ist eine endliche Menge von möglichen Nachrichten
2. A ist eine endliche Menge von möglichen Signaturen
3. K (der Schlüsselraum) ist eine endliche Menge möglicher Schlüssel
4. Für jedes K $\in K$, gibt es einen Signierungsalgorithmus $sig_K \in S$ und einen entsprechenden Verifikationsalgorithmus $ver_K \in V$. Jedes $sig_K : P \rightarrow A$ und $ver_K : P \times A \rightarrow \{true, false\}$ sind Funktionen, so dass folgende Gleichstellung erfüllt ist für jede Nachricht $x \in P$ und für jede Signatur $y \in A$:

$$\text{ver}(x, y) = \begin{cases} true & if \quad y = sig(x) \\ false & if \quad y \neq sig(x) \end{cases}$$

Ein Paar (x, y) mit $x \in P$ und $y \in A$ wird Signatur Nachricht genannt

2 Erzeugung Digitaler Signaturen

Folgende Merkmale muss eine Digitale Signatur erfüllen:

1. Unterschrift kann nicht gefälscht werden
2. Die Unterschrift wurde willentlich unter das Dokument gesetzt
3. Sie kann nicht auf ein anderes Dokument übertragen werden
4. Nachträgliche Änderungen im Dokument sind nicht möglich
5. Die Unterschrift kann nachträglich nicht geleugnet werden

2 Erzeugung Digitaler Signaturen

Mehrere Verfahren diese Merkmale zu erfüllen:

1. Symmetrische Verfahren: sind ausgeschlossen, das Dokument kann im Nachhinein wieder verändert werden ➔ Merkmale nicht erfüllt

2. Asymmetrische Verfahren:
   - Klartext mit öffentlichen Schlüssel e_K verschlüsseln
   - mit privatem Schlüssel entschlüsseln d_K

Idee: Vertauschen von e_K und d_K ➔ $d_K = sig_K$ und $e_K = ver_K$

Beispiel mit RSA:

- $sig_K(x) = x^a \bmod n$
- $ver_K(x,y) = true \Leftrightarrow x \equiv y^b \pmod n$

2 Erzeugung Digitaler Signaturen

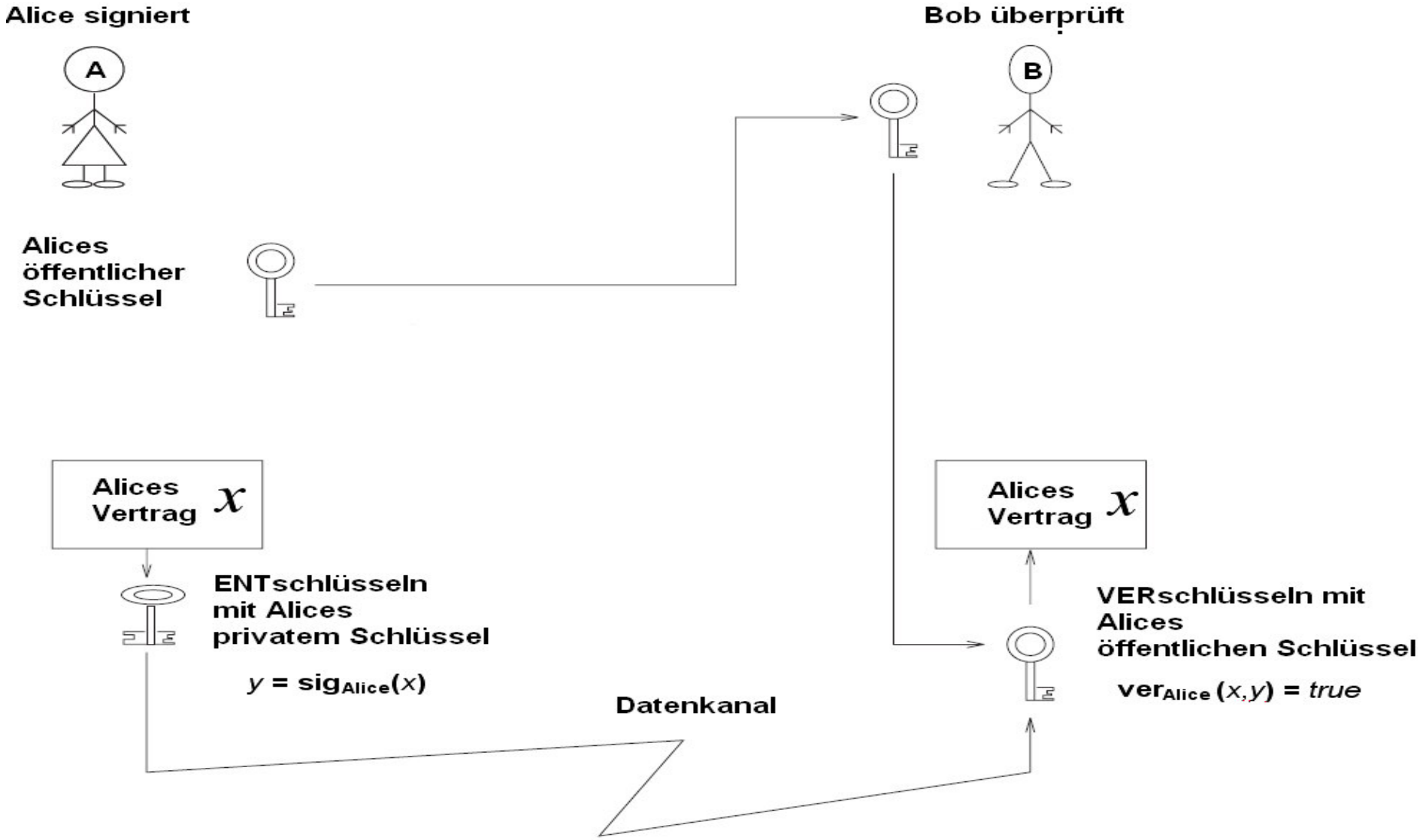

Abb. 1: Ver- und Entschlüsselung unter Verwendung des RSA-Algorithmus

2 Erzeugung Digitaler Signaturen

Aus Beispiel mit RSA folgt:

- Ver- und Entschlüsselung müssen Vertauschbar sein
- Unsere 5 Merkmale sind erfüllt
- Sicherheit beruht darauf, das Privater Schlüssel nicht bekannt wird

Nachteile: Bei der Erzeugung von Digitalen Signaturen unter Verwendung des RSA-Algorithmus

1. Asymmetrische Verfahren sind langsam
2. Signatur mindestens, so groß wie das Dokument selbst
3. Existentielle Fälschungen und Selective Fälschungen sind möglich, auf Grund der Multiplikativität von RSA

3 Signaturen und Einweg Hashfunktionen

- Nachteile beim RSA lassen sich umgehen, indem man Hashfunktionen benutzt: $h : \{0, 1\}^* \rightarrow Z$

- Erst Dokument mit Hashfunktion verschlüsseln $z = h(x)$, dann RSA drauf anwenden $y = \text{sig}_K(z)$

<u>Merkmale von Hashfunktionen:</u>

1. Erzeugen eines kompakten repräsentativen Abbilds einer Nachricht
2. Es soll nicht möglich sein, zu einem existierenden Hashwert, eine Nachricht zu finden➔ Einweg Hashfunktion (Urbild Problem)
3. Es soll nicht möglich sein, zu einer gegebenen Nachricht mit zugehörigem Hashwert, eine zweite Nachricht zu finden, die den gleichen Hashwert hat (2. Urbild Problem)
4. Kollision soll nicht auftreten, d.h. es soll sich kein Hashwertpaar finden lassen mit gleichen Hashwerten unter vernünftigen Aufwand (Geburtstagsangriff)

3 Signaturen und Einweg Hashfunktionen

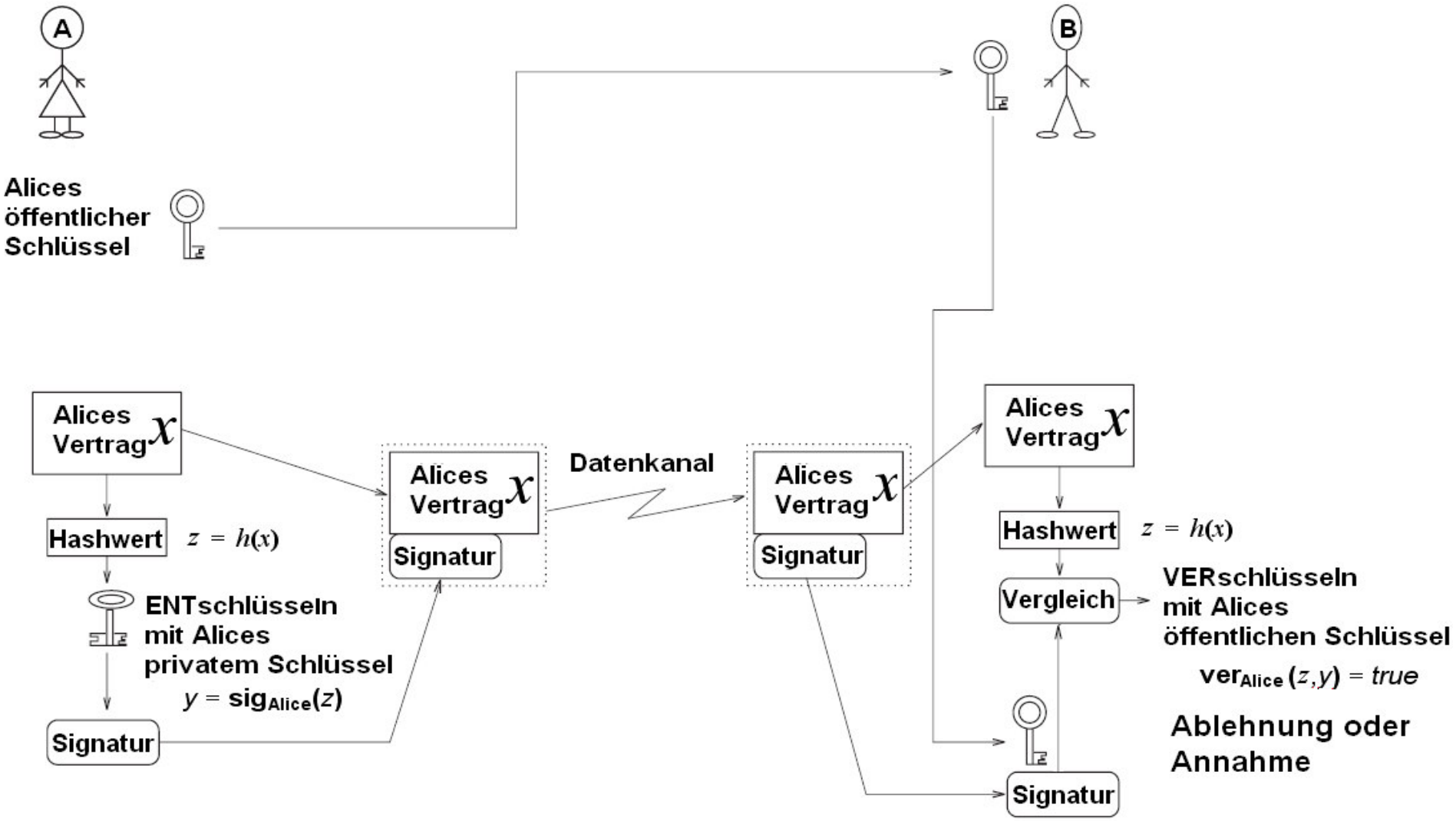

Abb. 2: Unterzeichnen mit Einweg-Hashfunktion und RSA (Digitale Signatur)

3 Signaturen und Einweg Hashfunktionen

Aus Beispiel mit Einweg Hashfunktionen und RSA folgt:

- 5 Merkmale einer Digitalen Signatur sind erfüllt
- Nachteile vom RSA entfallen da:

1. Hashwerte sehr kurz sind, meistens 20 Byte lang
2. Das Anwenden des RSA auf diesen Hashwert wenig Zeit kostet, da er nur 20 Byte lang ist
3. Die Hashwerte nicht vorrausagbar sind bei Einweg-Hashfunktionen ➔ keine Existentiellen und Selectiven Fälschungen möglich

4 Digital Signature Algorithmus

- Entwickelt vom NIST, speziell für Signaturen
- Modifizierte Form des ELGamal Kryptosystems ➔ basiert auf dem Problem der Diskreten Logarithmen

Geschichtliches:

- 1991 vorgeschlagen vom NIST, aber Kritik daran wurde laut
- 1994 wurde der DSA zum Standard im Digital Signatur Standard (DSS)

4 Digital Signature Algorithmus

Beschreibung des Digital Signatur Algorithmus:

Schlüsselerzeugung:

1. Erzeuge eine Primzahl p der Länge 512-1024 Bit, so dass das Diskrete Logarithmen Problem in Z_p schwer ist
2. Erzeuge eine Primzahl q mit der Länge 160 Bit und $q \mid (p\text{-}1)$
3. Wähle ein $\alpha \in Z_p$ das eine q te Wurzel von 1 modulo p ist, d.h. es gelte $\alpha^q \equiv 1 \pmod{p}$.
4. Schlüsselmenge K des DSA ist:

$$K = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}, 0 \leq a \leq q\text{-}1\}$$

Wobei die Werte p, q, α und β den öffentlichen Schlüssel bilden und a der private Schlüssel ist

4 Digital Signature Algorithmus

Signieren:

Für $x \in \{0,1\}^*$, $K = (p, q, \alpha, a, \beta)$ und eine (geheime) Zufallszahl k, $1 \leq k \leq q\text{-}1$ definieren wir die Signatur:

$$\text{sig}_K(x, k) = (\gamma, \delta)$$

mit $\gamma = (\alpha^k \bmod p) \bmod q$

und $\delta = (\text{SHA-1}(x) + a\gamma)k^{-1} \bmod q$

Wenn $\gamma = 0$ oder $\delta = 0$, wähle ein neues k

4 Digital Signature Algorithmus

Verifizieren:

Für $x \in \{0,1\}^*$ und $(\gamma, \delta) \in Z_p \times Z_p$, muss die Verifikation folgende Berechnungen durchführen:

$$e_1 = \text{SHA-1}(x)\ \delta^{-1} \bmod q$$
$$e_2 = \gamma\delta^{-1} \bmod q$$

$$\text{ver}_K(x, (\gamma,\delta)) = \text{true} \iff (\alpha^{e1}\beta^{e2} \bmod p) \bmod q = \gamma$$

4 Digital Signature Algorithmus

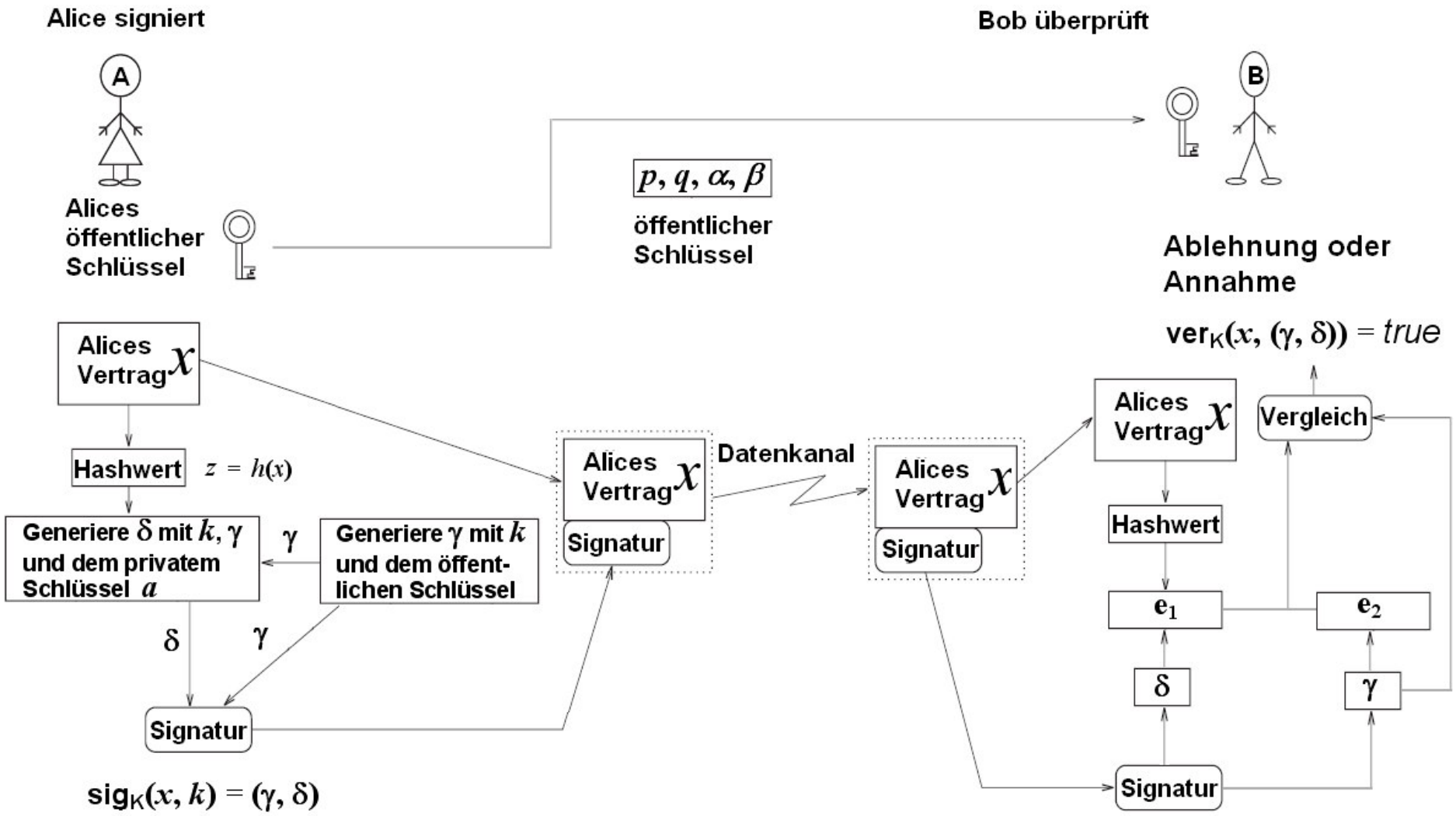

Abb. 3: Ver- und Entschlüsselung unter Verwendung des DSA-Algorithmus

4 Digital Signature Algorithmus

- DSA eignet sich hauptsächlich für Digitale Signaturen
- Parameterwahl ist bei DSA vorgeschrieben q = 160 Bit und p = 512 –1024 Bit
- Hashfunktion wird explizit vorgegeben, SHA-1
- k soll pseudozufällig sein, nicht doppelt verwenden
- Gilt (noch) als Sicher, solange das DLP nicht gelöst ist

- Spezieller Algorithmus des DSA wird für Smartcards eingesetzt, der Elliptische Kurven DSA, da er weniger Speicher benötigt, weil die Berechnung auf den Punkten der Elliptischen Kurve basiert

5 Zusammenfassung

- Digitale Signatur soll konventionelle Unterschrift ersetzen
- Realisierung mittels Asymmetrischer Verfahren und Einweg Hashfunktionen
- Speziell Entwickelter Algorithmus für Digitale Signaturen ist der DSA, entwickelt von der NIST
- DSA gilt zurzeit (noch) als sicher

Literaturverzeichnis

- Douglas R. Stinson: Cryptography: Theory and Practice. 2nd Edition, Chapman & Hall/CRC 2002
- Reinhard Wobst: Abenteuer Kryptologie 3. Überarbeitete Auflage Addison Wesly 2001

Fragen?