Elektronische Bezahlsysteme

Exkurs – Crashkurs Verschlüsselung

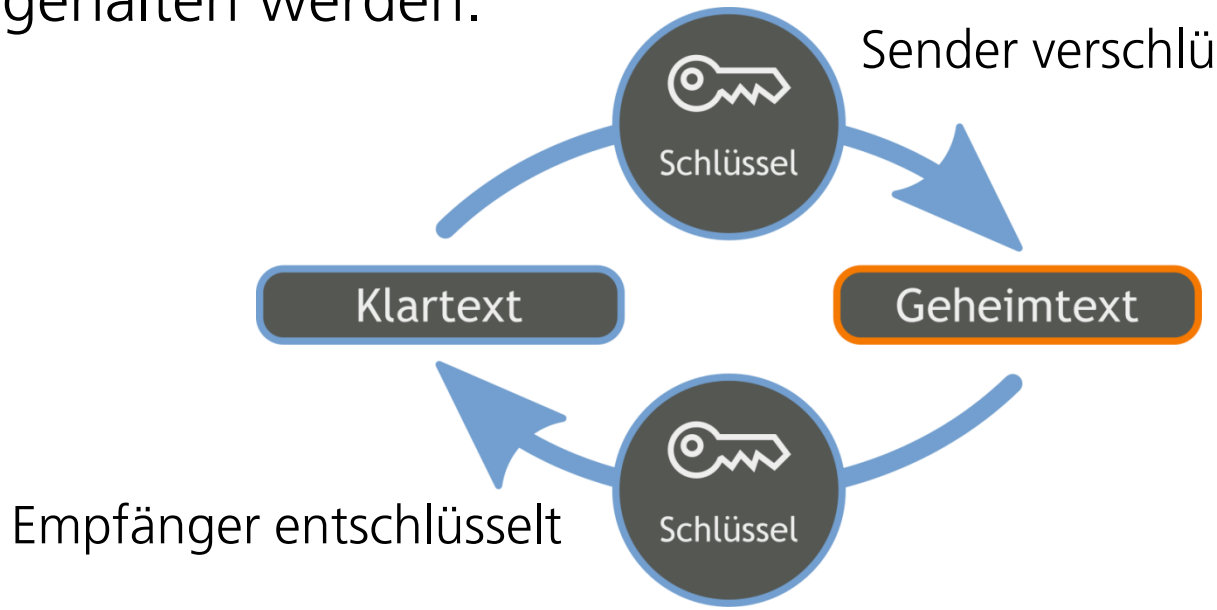
Hinweis: Der folgende Überblick ist stark vereinfacht und darf keinesfalls als geeignete Einführung in die verwendeten Verfahren verstanden werden. Er dient ausschließlich dazu, das Verständnis der in dieser Einheit diskutierten Bezahlverfahren zu erleichtern und stellt daher Einfachheit vor Genauigkeit.

Kryptosysteme

- Kryptosysteme **verschleiern** Informationen, sodass diese ohne Kenntnis einer geheimen Information (Schlüssel) nicht effizient wieder entschleiert werden können.
- Zur Verschleierung selbst wird ebenfalls ein Schlüssel verwendet, dieser kann entweder **geheim** oder **öffentlich** sein.
- Die Algorithmen sind **öffentlich** bekannt, denn die Geheimhaltung eines Verfahrens kann leicht durchbrochen werden und ist daher nicht geeignet, um Sicherheit zu gewährleisten.
- Neben der Verschleierung ist auch die **Signierung** von Informationen ein Anwendungsgebiet der Kryptographie.

Kryptosysteme

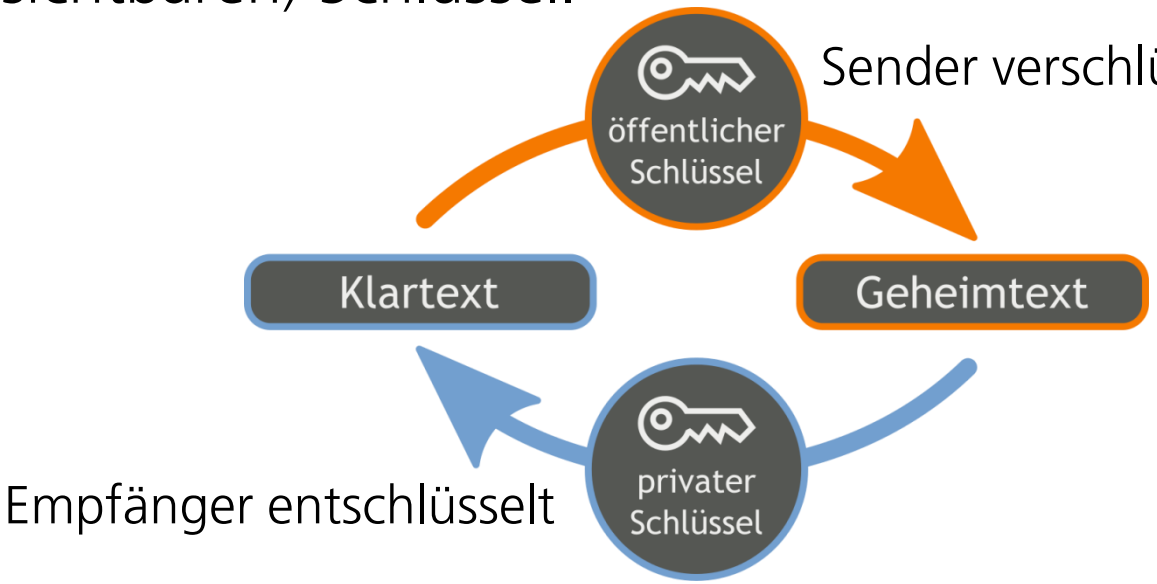
Bei **symmetrischen** Kryptosystemen benutzen Sender und Empfänger denselben Schlüssel. Dieser muss vorab auf geeignetem Wege übertragen und auf beiden Seiten geheimgehalten werden.



Diese Verfahren sind im Vergleich zu asymmetrischen Verfahren schneller und kommen mit kürzeren Schlüssellängen aus.

Kryptosysteme

Bei **asymmetrischen** Kryptosystemen besitzt der Empfänger ein Schlüsselpaar aus privatem (geheimen) und öffentlichem (für alle sichtbaren) Schlüssel.



Mit Hilfe des für alle sichtbaren öffentlichen Schlüssels des Empfängers(!) verschleiert der Sender die Information und sendet sie. Der Empfänger (und nur er) kann sie mit seinem privaten Schlüssel wieder entschleiern.

Asymmetrische Verschlüsselung – Beispiel RSA

Das wohl bekannteste Verfahren asymmetrischer Verschlüsselung ist nach seinen Erfindern benannt: ***R**ivest*, ***S**hamir* und ***A**dleman*.

Bei diesem Verfahren erzeugt jeder Empfänger einmalig ein individuelles Schlüsselpaar aus zwei Tupeln:

- Das Tupel (N, e) dient als **privater Schlüssel**
- und (N, d) als **öffentlicher Schlüssel**.

N, d und e sind so gewählt, dass für eine Nachricht $m < N$ gilt:

$$(m^d \bmod N)^e \bmod N = (m^e \bmod N)^d \bmod N = m.$$

Also: Eine Nachricht wird mit $c = m^d \bmod N$ verschlüsselt und kann mit $m = c^e \bmod N$ wieder entschlüsselt werden.

RSA – Beispiel

Privater Schlüssel: $(\boldsymbol{N = 77, e = 13})$

Öffentlicher Schlüssel: $(\boldsymbol{N = 77, d = 37})$

Nachricht „AFFE“, numerisch 1 6 6 5 (Position im Alphabet)

Zeichen	m	$c = m^{\mathbf{37}} \bmod \mathbf{77}$	$m = c^{\mathbf{13}} \bmod \mathbf{77}$
A	1	1	1
F	6	41	6
F	6	41	6
E	5	47	5

Asymmetrische Verschlüsselung – Beispiel RSA

Hinweis: N, d und e können natürlich nicht beliebig gewählt werden. Auf die Erzeugung des Schlüsselpaares gehen wir hier jedoch nicht ein, da dazu ein tieferes Verständnis für das Rechnen mit Kongruenzen notwendig ist.

Bei Interesse: R.L. Rivest, A. Shamir, and L. Adleman: *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*.

(http://people.csail.mit.edu/rivest/Rsapaper.pdf).

Hashfunktionen

- Hashfunktionen bilden Werte aus einer großen Quellmenge auf eine wesentlich kleinere Zielmenge ab. Sie bilden also eine Art „**Zusammenfassung**“ (engl.: *Digest*) des jeweiligen Wertes aus der Quellmenge.

- Durch die unterschiedlichen Größen der Mengen werden mehrere Werte der Quellmenge auf dieselben Zielwerte abgebildet. Daher sind Hashfunktionen nicht umkehrbar und werden auch als **Einweg-** oder **Falltürfunktionen** bezeichnet.

- **Benachbarte** Werte in der Quellmenge bilden normalerweise **nicht** auf **benachbarte** Werte der Zielmenge ab (*hash*: zerhacken). Veränderungen des Quellwertes haben also merkliche Auswirkungen auf den Hashwert.

Hashfunktionen

- Die Falltüreigenschaft bedeutet, dass bei Kenntnis eines Hashwertes nicht auf dessen Ursprung (Wert aus der Quellmenge) geschlossen werden kann.
- Sie ist jedoch per se keine kryptographische Eigenschaft, denn es fehlt jedwede Methode zur „Entschlüsselung“.
- Hashfunktionen können auch kryptographische Elemente enthalten (bspw. bei MD5), sind aber zunächst ohne Bezug zur Kryptographie!

Hashfunktionen – Beispiele

- **Divisionsrestmethode:** $h(m) = m \bmod N$
  - Beispiel mit $N = 13$: $h(15) = 15 \bmod 13 = 2$
  - Die Größe des Hashraumes wird durch N bestimmt.
  - Nachbarschaftserhaltend, daher nicht so gut geeignet.
- **Multiplikative Methode:** $h(m) = \lfloor N \cdot (mA - \lfloor mA \rfloor) \rfloor$
  - Anschaulich: Ein Roh-Hashwert zwischen 0 und 1 wird gebildet, indem die Nachricht mit einer Konstante A multipliziert und der ganzzahlige Anteil abgeschnitten wird. Dieser Rohwert wird mit der gewünschten Größe des Hashraumes N multipliziert und davon der Vorkommateil verwendet.

Hashfunktionen – Beispiele

- In der Praxis wird für A oft der Wert $\frac{\sqrt{5}-1}{2}$ verwendet, da er zu besonders guten Verteilungen führt.
- Beispiel $h(15)$ - Rohwert: $15 \cdot \frac{\sqrt{5}-1}{2} - \left\lfloor 15 \cdot \frac{\sqrt{5}-1}{2} \right\rfloor$

$$= 9{,}270509831248422723 06 - 9$$

$$= 0{,}270509831248422723 06$$

- Soll der Hashraum 100 Werte ($N = 100$) umfassen, so wir A mit 100 multipliziert. Der entstehende Vorkommateil entspricht dann dem endgültigen Hashwert:

$$h(15) = \lfloor 100 \cdot 0{,}270509831248422723 06 \rfloor = 27$$

Hashfunktionen – Beispiele

- **Mittquadratmethode:** Quadriere m und wähle als Hashwert die Ziffern in der Mitte.

- Die Größe des Hashraumes wird dadurch bestimmt, wie viele Stellen aus der Mitte genommen werden. Für einen Hashraum mit N Werten entnimmt man $\lceil \log_{10} N \rceil$ Stellen.
  - Beispiel: Es soll $h(15)$ bei einem Hashraum von 10 ermittelt werden. Dazu bildet man zunächst $15^2 = 225$.
  - Da der Hashraum nur 10 Werte umfassen soll ($\lceil \log_{10} 10 \rceil = 1$), genügt es, genau die Stelle in der Mitte zu entnehmen, also ist $h(15) = 2$.
  - Soll der Hashraum 100 Werte umfassen, müssen zwei Stellen entnommen werden ($\lceil \log_{10} 100 \rceil = 2$). Welche das sind, muss vorab vereinbart werden.

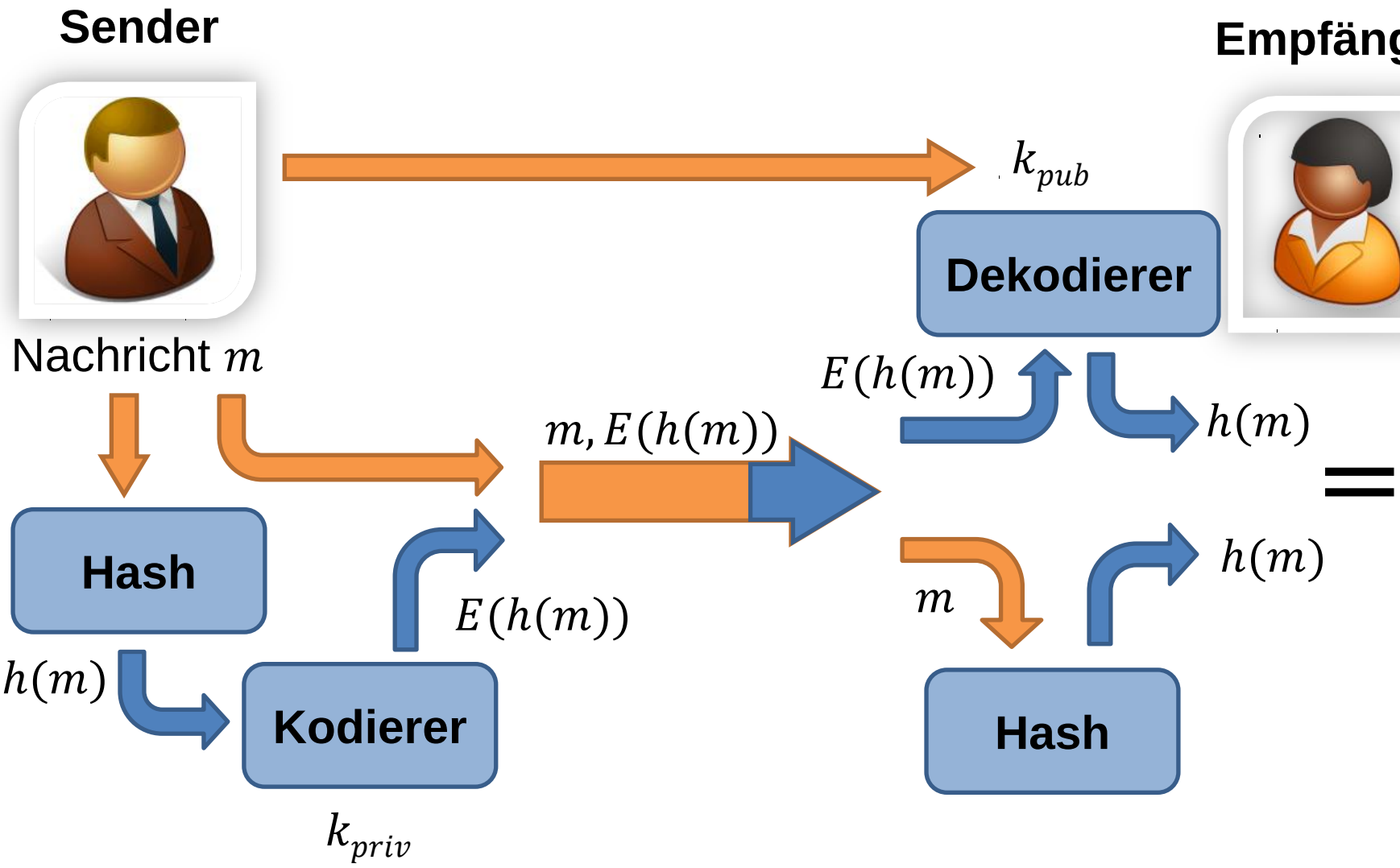
Signaturen

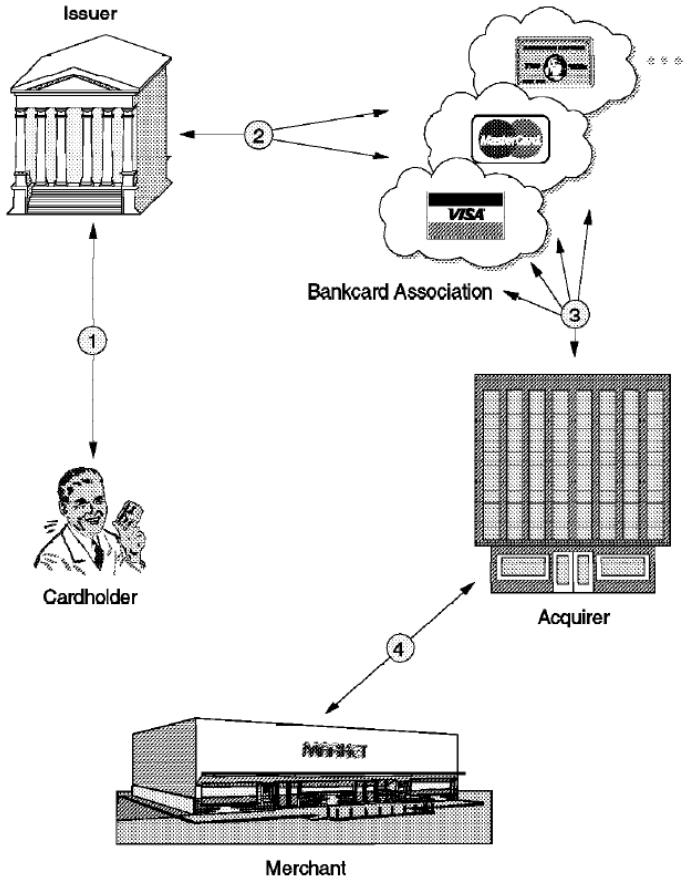
Asymmetrische Verfahren erlauben typischerweise eine öffentlich überprüfbare **Signatur**, die garantiert, dass die Nachricht tatsächlich durch den vorgegebenen Sender erstellt und auf dem Weg nicht verändert wurde.

Dazu bildet der Sender zunächst einen Hashwert der Nachricht. Diesen **verschlüsselt** er mit seinem privaten(!) Schlüssel und hängt ihn an die Nachricht an.

Der Empfänger kann nun den **öffentlichen** Schlüssel des Absenders nutzen, um den verschlüsselten Hashwert wieder zu entschlüsseln. Außerdem bildet er selbst manuell den Hashwert der empfangene Nachricht. Stimmen beide Werte überein, stammt sie tatsächlich vom erwarteten Sender und wurde auf dem Weg **nicht verändert**.

Signaturen


Sender
Empfänger
k_{pub}
Dekodierer
Nachricht m
$E(h(m))$
$h(m)$
$m, E(h(m))$
=
Hash
$E(h(m))$
$h(m)$
m
$h(m)$
Kodierer
Hash
k_{priv}



Secure Electronic Transaction (SET)

Secure Electronic Transaction (SET)

- SET ist ein Protokoll zur **Absicherung** elektronischer Bezahlungen über **unsichere Netze**, im Speziellen das Internet. Es wurde offiziell 1997 von Visa und MasterCard eingeführt.
- Dabei handelt es sich nicht um ein eigenes Bezahlsystem, sondern um einen **Aufsatz** auf die bestehende Kreditkarten-Bezahlinfrastruktur.
- SET bietet
  - Vertraulichkeit und Integrität
  - Authentifizierung von Käufer und Verkäufer
  - Nicht-Abstreitbarkeit
- Die Transaktionsabsicherung basiert auf **dualen Signaturen**.

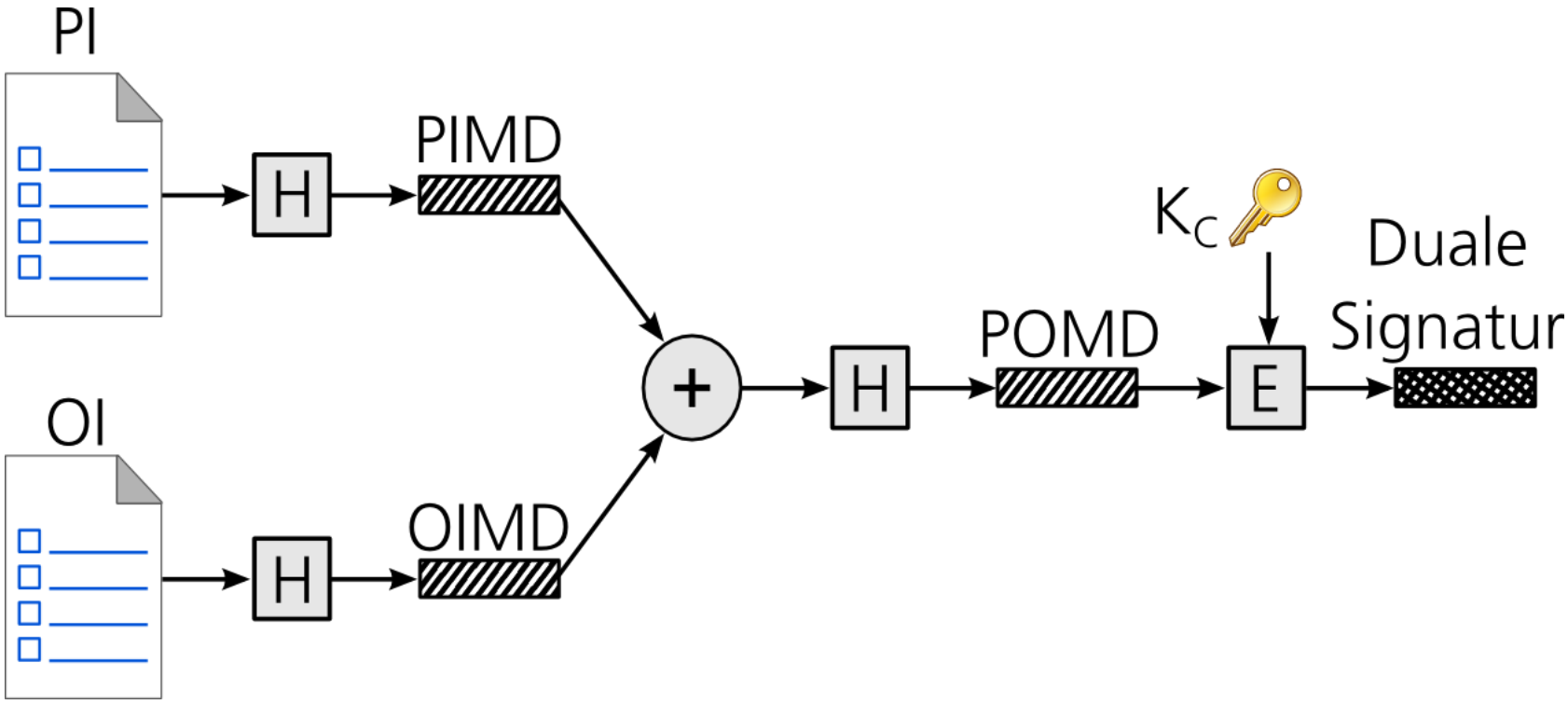
Duale Signaturen

Ein Kunde will Bestellinformationen an den Verkäufer und Bezahlinformationen an die Bank schicken.

- Der Verkäufer benötigt integre Bestellinformationen (*Order Information*, OI).
- und fordert Bestätigung, dass der Bezahlvorgang bei der Bank angwiesen wurde, soll aber keine Kontoinformationen erhalten.
- Die Bank muss den Bezahlauftrag erhalten (*Payment Instruction*, PI) und wissen, auf welche Bestellung sich der Bezahlvorgang bezieht, soll aber keine Details der Bestellung kennen.

Ziel: Die Zuordnung von PI zu OI soll jederzeit eindeutig möglich sein, allerdings soll jede Partei nur die Informationen einsehen können, die für sie bestimmt sind.

Konstruktion einer Dualen Signatur

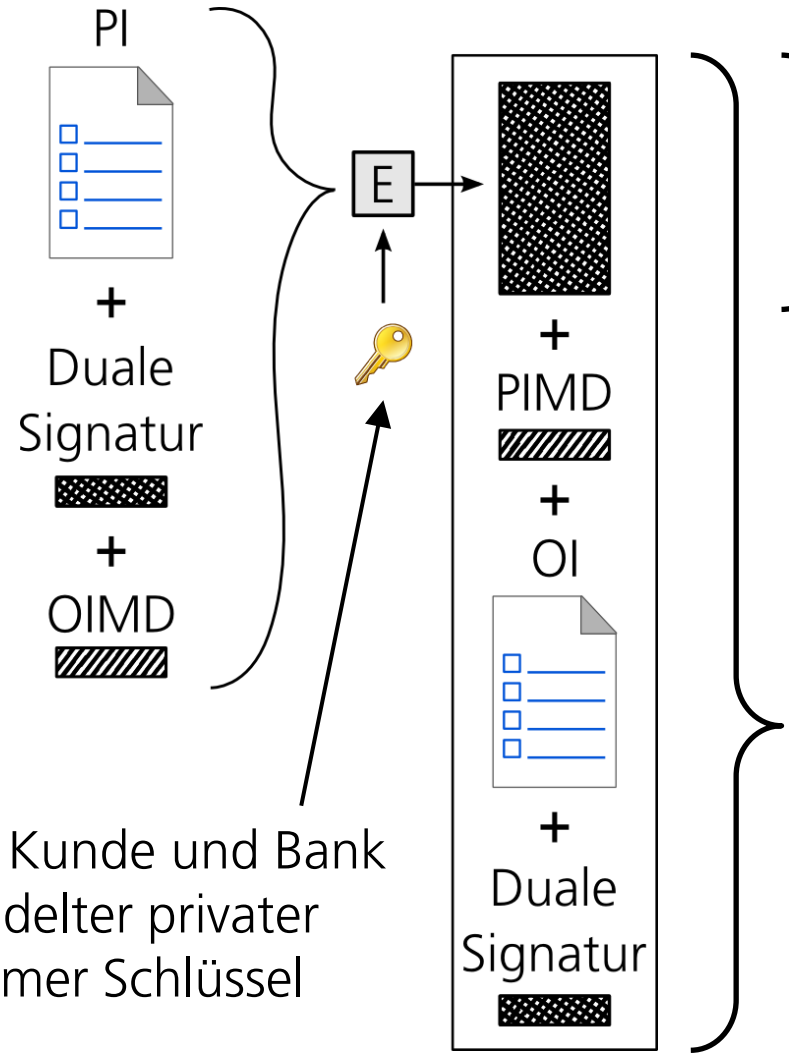


PI	Payment Information
OI	Order Information
H	Hashfunktion
+	Konkatenation

PIMD	Payment Information Message Digest
OIMD	Order Information Message Digest
POMD	Payment/Order Message Digest
K_C	Privater Schlüssel des Kunden
E	Verschlüsselung

SET Kunde: Bestellnachricht

Kunde erzeugt Nachricht:



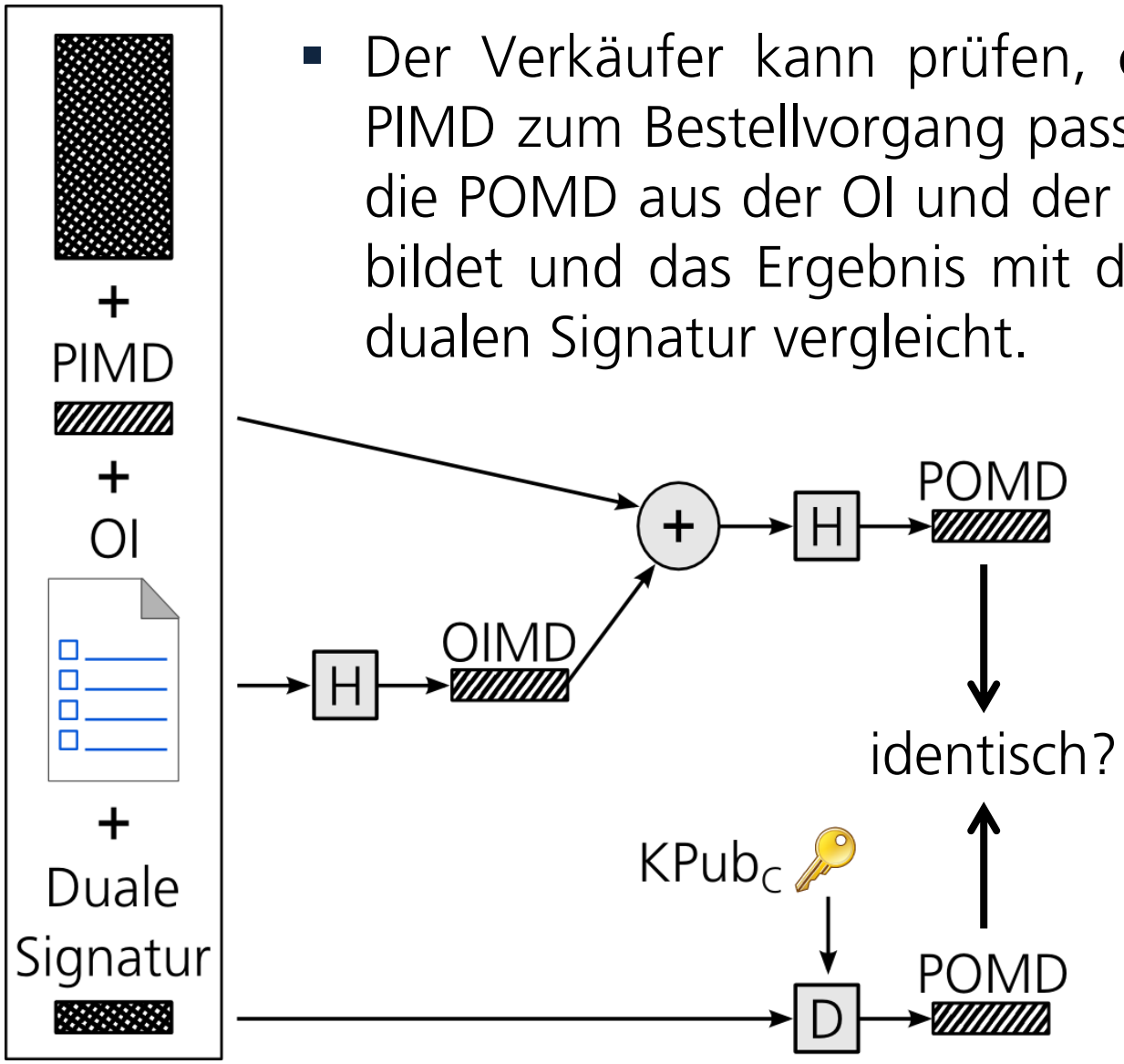
Die Bezahlinformationen können nur von der Bank entschlüsselt werden. Der Verkäufer leitet sie dahin weiter. Die Bank bestätigt die Zahlung an den Verkäufer.

Der Verkäufer erhält zunächst die gesamte Nachricht.

Zwischen Kunde und Bank ausgehandelter privater gemeinsamer Schlüssel

SET Verkäufer: Integritätstest

- Der Verkäufer kann prüfen, ob die gesendete PIMD zum Bestellvorgang passt, indem er selbst die POMD aus der OI und der PIMD vom Käufer bildet und das Ergebnis mit der POMD aus der dualen Signatur vergleicht.



Eigenschaften

- Dadurch, dass der Käufer zur Signatur seinen privaten Schlüssel benutzt, kann die Authentizität des Käufers mit Hilfe seines öffentlichen Schlüssels sowohl von der Bank als auch vom Verkäufer geprüft werden.
- Dasselbe gilt für die Integrität der dualen Signatur.
- Durch die POMD wird sichergestellt, dass Verkäufer und Bank mit Hilfe der ihnen zur Verfügung stehenden Informationen prüfen können, dass Bestellung und Bezahlung zueinander gehören, ohne dafür alle Informationen kennen zu müssen.
- Sendet der Käufer unterschiedliche Signaturen an Bank und Verkäufer, können Bank und Verkäufer dies untereinander aufdecken und den Vorgang abbrechen.

Kritik an SET

- Das verwendete Verschlüsselungsverfahren zwischen Kunde und Bank ist schwach (DES).
- Das Verfahren wird insgesamt als zu kompliziert wahrgenommen. Dies liegt unter anderem daran, dass auf Käuferseite ein geeigneter Client (e-Wallet) installiert und verwendet werden muss.
- SET ist Marktdurchdringung trotz massiver Werbung seitens VISA und MasterCard nicht gelungen.
- Onlineverfahren wie 3-D Secure („Verified by VISA“, „MasterCard SecureCode“) sind aktueller Stand der Technik. Wissenschaftlich sind diese Verfahren uninteressant, weil sie sehr gradlinig funktionieren.