



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Authentizität und Verbindlichkeit

July 28, 2023

Marc Stöttinger



Authenticity is the bedrock of cryptography, for without it, even the strongest encryption is worthless.

Bruce Schneier

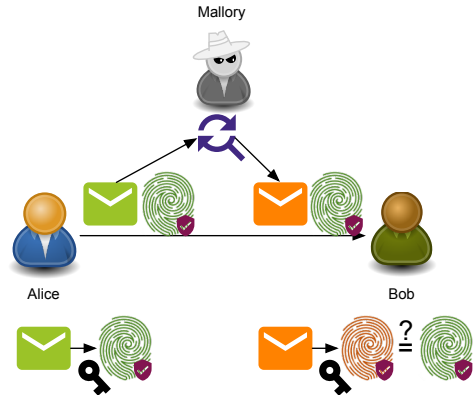
INTEGRITÄT UND AUTHENTIZITÄT DURCH MESSAGE AUTHENTICATION CODES

→ **Bedrohung:**

- Integrität: Mallory ändert die Nachricht
- Authentizität: Mallory fälscht eine Nachricht und gibt sich als Alice aus

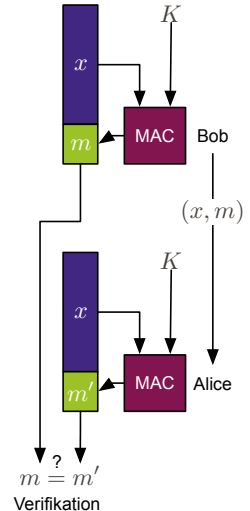
→ **Ziele:**

- Integrität: Bob kann prüfen, ob die Nachricht verändert wurde
- Authentizität: Bob kann prüfen, ob die Nachricht von Alice stammt



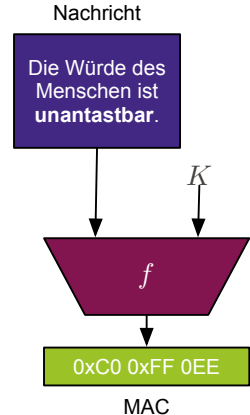
WIESO REICHT INTEGRITÄT NICHT AUS?

- Integrität garantiert nur, dass Veränderungen erkannt werden, z.B Fehler bei der Übertragung.
 - Mit einer Prüfsumme kann jeder den Ciphertext auf Fehler überprüfen.
- Es ist keine Aussage über eine Verfälschung möglich
 - Die Prüfsumme kann jeder selber berechnen
 - Bei einem verschlüsselten Ciphertext ist nicht bekannt, woher der Ciphertext stammt oder ob dieser authentisch ist.
- Es wird ein Mechanismus benötigt, dass die Prüfsumme nur mit Kenntnis eines Geheimnisse berechnet werden kann.



MESSAGE AUTHENTICATION CODES (MACS)

- **Message Authentication Code (MAC)** nutzen
Hashfunktionen und symmetrische Verschlüsselungsverfahren zusammen mit symmetrischen Schlüsseln, um Integrität und Authentizität zu gewährleisten
 - Schnell, da symm. Verschlüsselung und Hashfunktionen genutzt werden
 - Unverbindlich, da der Schlüssel beiden Parteien bekannt ist
- Benötigte Eigenschaften eines MAC Verfahrens:
 - **Einwegeigenschaft:** Der Schlüssel darf nicht aus MAC und Nachricht bestimmbar sein
 - **Schwache Kollisionsresistenz:** Keine andere Nachricht mit gleichem Schlüssel und gleicher MAC darf effizient berechenbar sein



BEKANNTE MAC-ALGORITHMEN

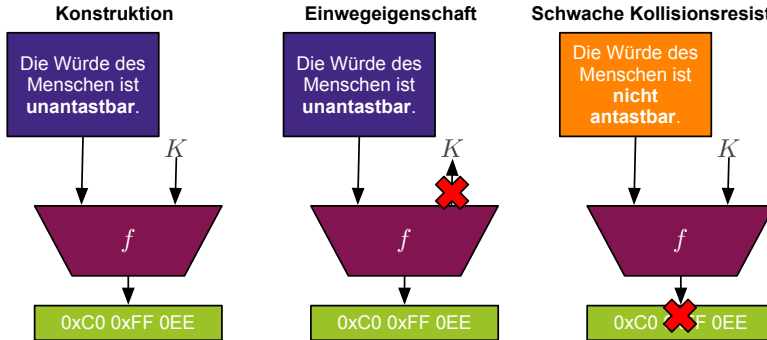
→ Es existieren spezielle MAC-Konstruktionen, um ein MAC-Verfahren aus einer Hashfunktion oder einem symmetrischen Verschlüsselungsverfahren zu konstruieren

Verfahren		Schlüssellänge	Kommentar
Hash-based (HMAC)	MAC	Blockgröße der Hashfunktion (z.B. 256-bit bei SHA2-256)	Gute Sicherheit, da starke Kollisionsresistenz, allerdings vergleichsweise langsam
Blockchiffre GMAC und CMAC	Modi	Schlüssellänge der Blockchiffre (z.B. 128 bei AES-128)	In bestimmten Fällen geringe Kollisionsresistenz
Blockchiffre GCM	Modus	Schlüssellänge der Blockchiffre (z.B. 128 bei AES-128)	In bestimmten Fällen geringe Kollisionsresistenz, bietet aber zusätzlich Verschlüsselung an

MESSAGE AUTHENTICATION CODES (MACS) SICHERHEIT HASHFUNKTIONEN

→ MACs via Hashfunktionen sind sicher, da:

- Einweg: Schlüssel kann nicht aus MAC und Nachricht berechnet werden
- Schwache Koll.: Andere Nachricht mit gleichem MAC und Schlüssel schwer findbar
- Je nach verwendeter Hashfunktion ist eine Maskierung der Digest notwendig



ANGRIFF GEGEN HMACS MIT VORANGESTELTLEM GEHEIMNIS

Alice

Mallory

Bob

$$x = (x_1, \dots, x_n)$$

$$\xleftarrow{(x, m)} m = h(k || x_1, \dots, x_n)$$

abgefangen

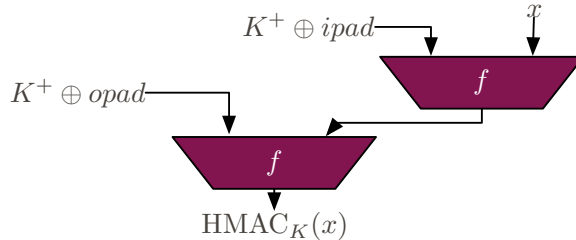
$$x_0 = (x_1, \dots, x_n, x_{n+1})$$

$$m' = h(k || x_1, \dots, x_n, x_{n+1}) \xleftarrow{(x_0, m_0)} m_0 = h(m || x_{n+1})$$

Valide MAC, da $m' = m_0$

HASH-BASED MAC

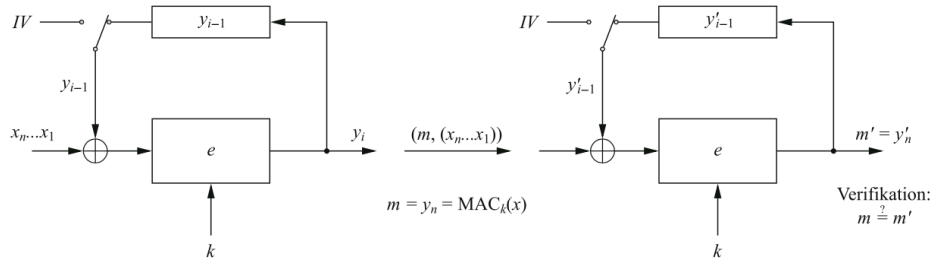
- Die [HMAC] Konstruktion sollte immer eine Padding-Mask nutzen, um die Hashfunktion $f(x)$ in einen HMAC zu wandeln: $HMAC_K(x) = f((K^+ \oplus opad) | f((K^+ \oplus ipad) | x))$
- Konstanten $ipad$ und $opad$ werden als Padding-Masken genutzt.
- K wird auf die Blockgröße des Hash aufgefüllt und K^+ bezeichnet



MESSAGE AUTHENTICATION CODES (MACS) SICHERHEIT SYM. VERSCHLÜSSELUNG

→ MACs via symmetrischer Verschlüsselungsverfahren sind sicher, da:

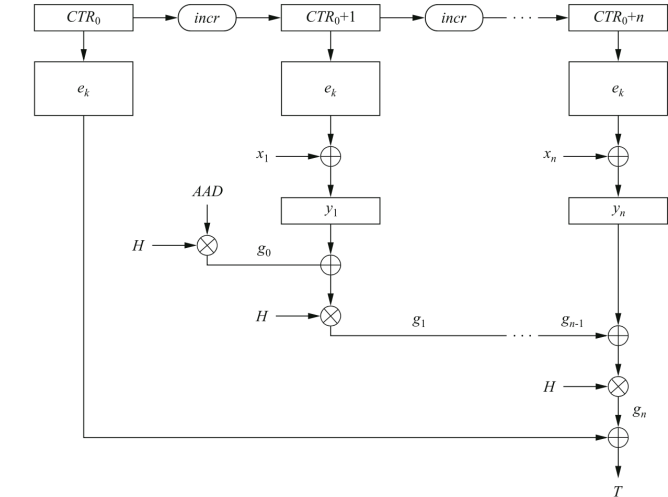
- **Einweg**: Schlüssel kann nicht aus Plaintext $P = x_1, \dots, x_n$ und Ciphertext $C = m$ berechnet werden
- **Schwache Koll.**: Jeder Plaintext P wird in einen zufälligen und eindeutigen Ciphertext C unter fixem Schlüssel k verschlüsselt.



Quelle: Christoph Paar, Jan Pelz: Kryptografie verständlich, 2016, Springer

BLOCKCHIFFRE BETRIEBSMODI GALOIS COUNTER MODE (GCM)

- Konstruktion um Blockchiffre in MAC-Verfahren umzuwandeln
- Multiplikationskonstante $H = ENC_k(0)$
- ADD zusätzliche Authentisierungsdaten
- T ist der Authentisierungstoken, der als MAC genutzt werden kann.



DISKUSSION IN KLEINEN GRUPPEN

Sind MACS uneingeschränkt vertrauenswürdig?

Eine Gruppe von Freunden hat einen symmetrischen Schlüssel miteinander getauscht, um sich gegenseitig via AES-GCM sichere und vertrauenswürdige Nachrichten zu schicken. Was für ein Problem könnte in diesem Setting entstehen?

Verbindlichkeit

Überlegen Sie sich einen Fall, in dem das oben identifizierte Problem zum Tragen kommen kann.

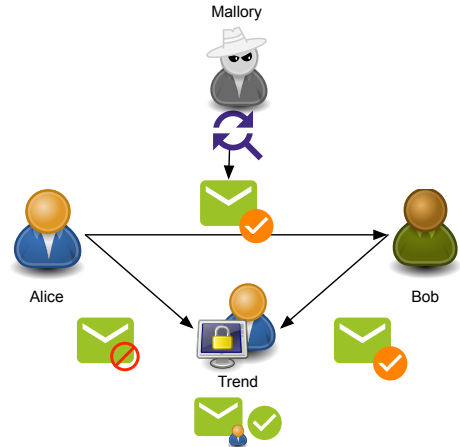
INTEGRITÄT, AUTHENTIZITÄT UND VERBINDLICHKEIT

→ **Bedrohung:**

- Integrität: Mallory ändert die Nachricht
- Authentizität: Mallory fälscht eine Nachricht und gibt sich als Alice aus
- **Nicht-Abstreitbarkeit:** Alice bestreitet eine Nachricht an Bob gesendet zu haben

→ **Ziele:**

- Integrität: Bob kann prüfen, ob die Nachricht verändert wurde
- Authentizität: Bob kann prüfen, ob die Nachricht von Alice stammt
- **Nicht-Abstreitbarkeit:** Bob kann gegenüber einer vertrauenswürdigen, dritten Instanz nachweisen, dass eine Nachricht von Alice stammt



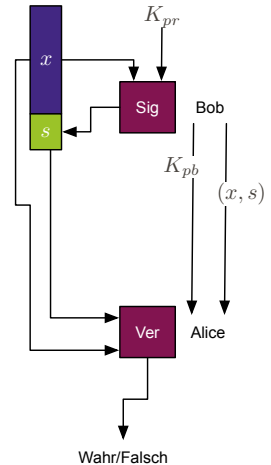
DIGITALE SIGNATUREN

- Digitales Pendant zur handgeschriebenen Unterschrift
 - Zu einer öffentlichen Nachricht m soll es eine digitale Signatur sig geben
- Anforderungen an ein digitales Signaturverfahren:
 1. Jeder muss die Signatur von sig zu m verifizieren können
 2. Nur Alice darf eine gültige Signatur sig zur Nachricht m erzeugen
- Vergleich mit Anforderungen bei Verschlüsselung mit asymmetrischer Kryptographie:
 1. Jeder darf eine Nachricht an Alice verschlüsseln können
 2. Nur Alice darf den Ciphertext entschlüsseln können



GENERELLER ABLAUF

- Bob erstellt als erstes ein Private-Public-Schlüsselpaar (K_{pr} und K_{pb})
 - K_{pr} wird benötigt, um die Signatur zu erzeugen
 - K_{pb} wird an Alice geschickt, damit sie die Signatur verifizieren kann
- Signieren der Nachricht m :
 - Über einen Einweg- oder Falltür-Funktion ($F_{trap}(\cdot)$) wird s erzeugt:
 $F_{trap}(K_{Pr}, m) \rightarrow S$
 - **Die Falltür Funktion ist im allg. keine Verschlüsselung!**
 $F_{trap}(K_{Pr}, m) \rightarrow s \neq ENC_{K_{pr}, m} \rightarrow s$
- Verifikation der Signatur s :
 - Mit einer Verifikationsfunktion wird mit Hilfe von K_{pb} und m geprüft, ob s valide ist



ALICE SENDET VIA RSA SIGNIERTE NACHRICHT AN BOB

Alice

Bob

$K_{pb} = (N, e)$ und $K_{pr} = d$ $\xrightarrow{\text{Kanal1: Sende } K_{pb} = (N, e)}$

Signiere $s = m^d \mod N$ $\xrightarrow{\text{Kanal2: } (m, s)}$

Berechne $m' = s^e \mod N$
Verifizier ob $m = m'$

Signaturprüfung funktioniert, da $s^e \mod N = m^{ed} \mod N = m$

- Jede Partei darf $K_{pb} = (N, e)$ kennen und Signaturen prüfen
- Nur Alice kennt $K_{pr} = (s)$ und kann somit Nachrichten signieren

RECAP: BOB SENDET VIA RSA VERSCHLÜSSELTE NACHRICHT AN ALICE

Alice

Bob

$$K_{pb} = (N, e) \text{ und } K_{pr} = d$$

Kanal1: Sende $K_{pb} = (N, e)$

$$\longrightarrow$$

$$\text{Entschlüssele } P = C^d \mod N \longleftarrow \text{Kanal2: } (C)$$

Verschlüssel $C = P^e \mod N$
Verifizier ob $m = m'$

Entschlüsselung funktioniert genauso wie bei der Signatur nur

- Sonderfall für Schulbuch RSA Entschlüsselung entspricht der Signierfunktion!
- Es ist sehr gefährlich, einfach Schulbuch RSA zu benutzen!

EXISTENZIELLE FÄLSCHUNG

Alice

Mallory

Bob

 $\xleftarrow{(N, e)}$
 $\xleftarrow{(N, e)} K_{pr} = (d), K_{pb} = N, e$
1. Wähle Signatur: $s \in \mathbb{Z}_N$

2. Berechne die Nachricht:

 $\xleftarrow{(x, s)} m \equiv s^e \pmod{N}$

Verifikation:

 $m' \equiv s^e \pmod{N} = m$
Signatur ist valide!

→ Probabilistische Signaturverfahren verhindern diesen Angriff.

→ Das **RSA-EMSA-PSS** Schema für Signaturen sollte im Fall von RSA genutzt werden

DIGITALE SIGNATURVERFAHREN IN DER PRAXIS

- Weitere Verfahren zur Signaturberechnung existieren:
 - Digital Signature Algorithm (DSA)
 - Elliptic Curve DSA (ECDSA)
 - Elgamal Signatur
 - Merkle Signatur
- Für Verbindlichkeit müssen weitere Informationen an den öffentlichen Schlüssel gebunden werden (⇒ **Zertifikate** und **PKI** im Kapitel Protokolle)

DIGITAL SIGNATURE ALGORITHM (DSA) ENTSTEHUNG UND VERWENDUNG

- Digital Signature Algorithm (DSA) wurde 1994 standardisiert [DSA]
 - Von der Benutzung von DSA wird mittlerweile abgeraten!
- Die Sicherheit von DSA beruht auf dem diskreten Logarithmen Problem

DSA Algorithmus	Input	Output	Durchgeführt von
Parametergenerierung	-	Parameter (p, q, g)	Vertrauenswürdige Partei
Schlüsselgenerierung	(p, q, g)	Schlüssel $K_{pb} = y, K_{pr} = x$	Alice (Sender*in)
Signieren	$(p, q, g), x, m$	Signatur (r, s) zu M	Alice (Sender*in)
Signieren	$(p, q, g), x, m, (r, s)$	Wurde (r, s) für M von Besitzer*in von K_{pb} erzeugt?	Bob (Empfänger*in)

DIGITAL SIGNATURE ALGORITHM (DSA) SCHLÜSSELGENERIERUNG

Alice

Bob

Parametergenerierung (öffentlich)

Wähle eine Primzahl p

Wähle eine Primzahl q die $p - 1$ teilt

Berechne $g \equiv h^{(p-1)/q} \mod p$

Schlüsselgenerierung

Wähle x mit $1 \leq x \leq q$

Berechne $y \equiv g^x \mod p$

Setze $K_{pb} = y$ und $K_{pr} = x$

$\xleftarrow{(p, q, g)}$ für ein zufälliges h $\xrightarrow{(p, q, g)}$

$\xrightarrow{(K_{pb}) = y}$

DIGITAL SIGNATURE ALGORITHM (DSA) SIGNIEREN UND VERIFIZIEREN

Alice

Bob

Wähle k mit $1 \leq k \leq q$

Berechne: $r \equiv (g^k \bmod p) \bmod q \neq 0$

Berechne: $s \equiv (k^{-1} \cdot (m + r \cdot x)) \bmod q \neq 0$

$\xrightarrow{m, (r, s)}$

Berechne $w \equiv s^{-1} \bmod q$

Berechne $u_1 \equiv m \cdot w \bmod q$

Berechne $u_2 \equiv r \cdot w \bmod q$

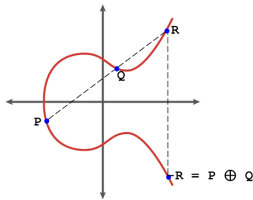
Berechne $v \equiv (g^{u_1} \cdot y^{u_2} \bmod p) \bmod q$

Falls $v = r \rightarrow$ valide

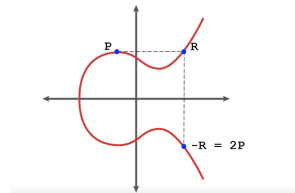
ASYMMETRISCHE VERSCHLÜSSELUNG ELLIPTISCHE KURVEN (1/2)

- Eine Alternative zu primen Restklassenringen sind **elliptische Kurven (ECC)**
 - **Elliptische Kurve**: Menge an Punkten die eine Gleichung erfüllen, z.B.: $y^2 = x^3 + ax + b$
- Seien A, P zwei Punkte auf einer Kurve mit $a \cdot P = A$
 - **Einfach**: Aus P und a den Punkt A zu berechnen ($A = a \cdot P$)
 - **Schwer**: Aus P und A den Wert a zu berechnen ($a = P/A$)

Punktaddition Kurve $y^2 = x^3 + ax + b$



Punktmultiplikation Kurve $y^2 = x^3 + ax + b$



Quelle: <https://blog.intothesynergy.com/2019/07/on-isogenies-verifiable-delay-functions.html>

ASYMMETRISCHE VERSCHLÜSSELUNG ELLIPTISCHE KURVEN (2/2)

- Elliptische Kurven können in Verfahren genutzt werden, die auf dem diskreten Logarithmusproblem basieren
- Vorteil von elliptischen Kurven ist, dass die Kurven kleinere Bit-Werte besitzen

Bitlänge Schlüssel	sym.	Bitlänge Primzahl	Bitlänge ECC	Ratio Bitlänge Primzahl / ECC
80		1024	160	6,4
128		3072	256	12
256		15360	512	30

ECDSA SCHLÜSSELGENERIERUNG

Alice

Bob

Parametergenerierung (öfftl.)

Wähle eine eine Kurve:

$$E(p, a, b, q, A)$$

Schlüsselgenerierung

Wähle einen Zufallswert d mit $0 < d < q$

Berechne $B = dA$

Setze $K_{pb} = (p, a, b, q, A, B)$

und $K_{pr} = d$

$$\xleftarrow{E(p, a, b, q, A)}$$

$$\xrightarrow{E(p, a, b, q, A)}$$

$$\xrightarrow{(K_{pb}) = (p, a, b, q, A, B)}$$

ECDSA SIGNIEREN UND VERIFIZIEREN

Alice

Bob

Wähle ephemeral k_E mit $0 \leq k_E \leq q$

Berechne: $R = k_E A$

Setze: $r = x_R$

Berechne: $s \equiv (h(x) + d \cdot r)k_E^{-1} \pmod{q} \xrightarrow{(x, r, s)}$

Berechne $w \equiv s^{-1} \pmod{q}$

Berechne $u_1 \equiv w \cdot h(x) \pmod{q}$

Berechne $u_2 = r \cdot w \pmod{q}$

Berechne $P = u_1 A + u_2 B$

Falls $x_P \equiv r \pmod{q} \rightarrow$ valide

ANGRIFFE AUF SIGNATURVERFAHREN

- Die Sicherheit von ECDSA hängt stark vom Zufallswert K ab:
 - Falls k bekannt wird, kann $K_{pr} = x$ berechnet werden
 - Falls k wiederverwendet wird, kann $K_{pr} = x$ berechnet werden [PS3]
- Bei ECDSA müssen bestimmte Signaturen abgefangen werden (z.B. $s = 0$ [Orac])
- RSA Verschlüsselung und Signaturen niemals mit dem gleichen Schlüsselpaar!
 - Verschlüsselte Nachricht könnte entschlüsselt werden via Anfrage zur Signatur
 - Unbeabsichtigte Signatur könnte erzeugt werden via Anfrage zur Entschlüsselung
- RSA benötigt Paddingverfahren (z.B. RSA-PSS) zur sicheren Signaturerzeugung

ASYMMETRISCHE SIGNATURVERFAHREN

- Direktes Signieren und Verifizieren von großen Nachrichten ist sehr ineffizient
 - Signieren: $s = k^{-1} \cdot (M + r \cdot r) \mod q$
 - Verifizieren: $u_1 = M \cdot w \mod q$
- Analog zur hybrider Verschlüsselung: Große Nachricht mit Hilfsfunktion in einen kleinen, eindeutigen Fingerabdruck umwandeln, der dann signiert wird
- Anforderung an Hilfsfunktion und Fingerabdruck
 - Jede Person sollte die Hilfsfunktion berechnen können
 - Es sollte nicht möglich sein, vom Fingerabdruck auf eine Nachricht zurückzurechnen

ZUSAMMENFASSUNG

- MACs basierend auf Hashfunktionen und symmetrischer Verschlüsselung
- Besprechung von verschiedenen MAC Verfahren
- Unterschiede zwischen MACs und digitalen Signaturen
- Korrekter Einsatz von MACs oder digitalen Signaturen beurteilen
- Sicherheitsgarantien der Digitalen Signaturen
- Beziehung zwischen dem RSA Verschlüsselungs- und Signaturverfahren
- Existierende Digitale Signaturverfahren
- Elliptischen Kurven gegenüber primen Restklassenringen