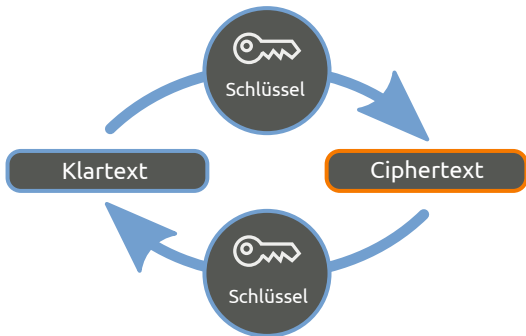


# Eine Mikroeinführung in die asymmetrische Kryptographie

Jakob Breier

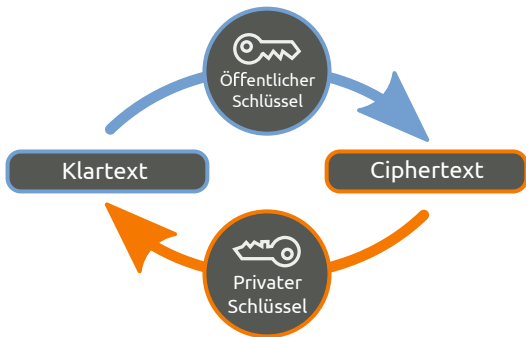
Cryptoparty 2014

# Symmetrische Kryptographie



- Nur ein Schlüssel zum Ver- und Entschlüsseln
- „Symmetric key“, „shared secret key“

# Asymmetrische Kryptographie



- Ein öffentlicher Schlüssel zum Verschlüsseln und ein privater Schlüssel zum Entschlüsseln
- Deshalb auch: „public key cryptography“

# Die wichtigsten asymmetrischen Primitiven

- Verschlüsseln / Entschlüsseln von Daten
- Signieren / Verifizieren von Daten
- Schlüsselaustausch von symmetrischen Schlüsseln

# Verschlüsseln / Entschlüsseln von Daten

- Alice erstellt Schlüsselpaar bestehend aus öffentlichen und privaten Schlüssel.
- Alice verbreitet öffentlichen Schlüssel.
- Bob besorgt öffentlichen Schlüssel, überzeugt sich, dass er wirklich von Alice stammt.
- Bob verschlüsselt Nachricht mit öffentlichem Schlüssel.
- Nur Alice kann Nachricht mittels privaten Schlüssel entschlüsseln.
- Systeme: RSA, ElGamal, Rabin

# Signieren / Verifizieren von Daten

- *Alice erstellt Schlüsselpaar bestehend aus öffentlichem und privatem Schlüssel.*
- *Alice verbreitet öffentlichen Schlüssel.*
- Alice signiert Nachricht mit privatem Schlüssel.
- *Bob besorgt öffentlichen Schlüssel, überzeugt sich, dass er wirklich von Alice stammt.*
- Bob verifiziert Nachricht mit öffentlichem Schlüssel.
- Nur Alice kann Nachrichten mittels privaten Schlüssel signieren.
- Systeme: RSA, DSA, ECDSA

# Schlüsselaustausch: Diffie-Hellman

- Protokoll definiert Gruppe  $G$  und Erzeuger  $g$ .

Alice		Bob
wählt zufällig $a$		wählt zufällig $b$
berechnet $g^a$		berechnet $g^b$
	$\xrightarrow{g^a}$	
	$\xleftarrow{g^b}$	
berechnet $g^{ab} = (g^b)^a$		berechnet $g^{ab} = (g^a)^b$

- Alice und Bob nutzen  $g^{ab}$
- Diffie-Hellman Hypothese: Nur Alice und Bob können  $g^{ab}$  berechnen.
- Gruppen:  $\mathbb{Z}_p$  (  $g^a \bmod p$  ), elliptische Kurven

# Man-in-the-Middle Angriff

Alice	Eve	Bob
wählt zufällig $a$	wählt zufällig $e$	wählt zufällig $b$
berechnet $g^a$	berechnet $g^e$	berechnet $g^b$
$\xrightarrow{g^a}$	$\xleftarrow{g^e} \quad \xrightarrow{g^e}$	$\xleftarrow{g^b}$
berechnet $g^{ae}$	berechnet $g^{ae}, g^{be}$	berechnet $g^{eb}$

- Eve entschlüsselt und verschlüsselt transparent Nachrichten zwischen Alice und Bob.



# Anwendungsbeispiel TLS (früher SSL)

## Üblicher Ablauf:

- Client (z.B. Browser) verbindet sich zum Server
- Nur Server soll authentifiziert werden („Spreche ich wirklich mit google.com“)
- Beim Handshake:  
Austausch eines symmetrischen Schlüssels mittels asymmetrischer Kryptographie
- Nach Handshake:  
Verschlüsselung und Authentifizierung durch symmetrische Kryptographie

# TLS Schlüsselaustausch: RSA

Client		Server
wählt zufällig $r$		
$c = RSAEnc(r)$		
	$\xrightarrow{c}$	
		$r = RSADec(c)$

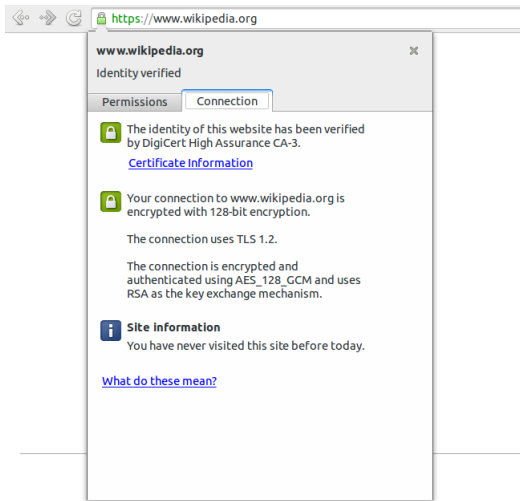
# TLS Schlüsselaustausch: RSA

The screenshot shows a web browser window with the address bar displaying <https://www.amazon.de>. A security overlay is visible, showing the following information:

- Identity verified**
- Permissions** (tab selected)
- Connection** (tab selected)
- The identity of this website has been verified by VeriSign Class 3 Secure Server CA - G3.**  
[Certificate Information](#)
- Your connection to www.amazon.de is encrypted with 128-bit encryption.**
- The connection uses TLS 1.0.**
- The connection is encrypted using RC4\_128, with SHA1 for message authentication and RSA as the key exchange mechanism.**
- Site Information**  
You have never visited this site before today.  
[What do these mean?](#)

The background shows the Amazon.de homepage with various product categories and advertisements, including a Kindle Paperwhite advertisement.

# TLS Schlüsselaustausch: RSA

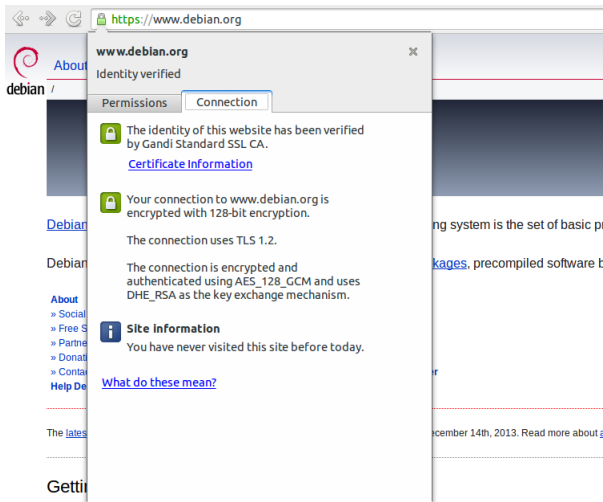


# TLS Schlüsselaustausch: RSA-DHE

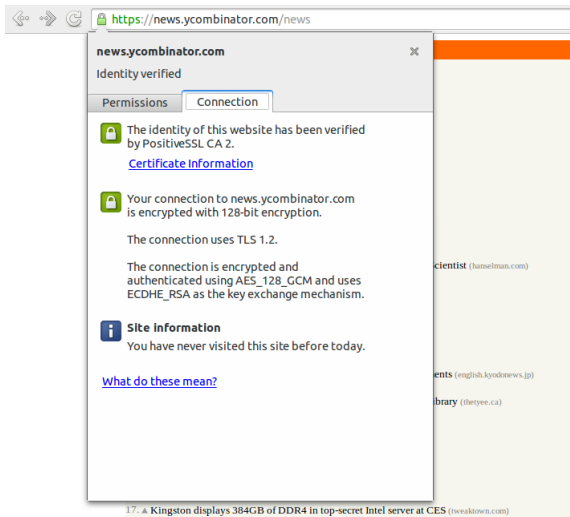
Client		Server
wählt zufällig $a$		wählt zufällig $b$
berechne $g^a$		berechne $g^b$
		$s = \text{RSASign}(g^b)$
	$\xrightarrow{g^a}$	
	$\xleftarrow{g^b, s}$	
$\text{RSAVerify}(g^b, s)$		
berechne $g^{ab}$		berechne $g^{ab}$

⇒ Perfect Forward Secrecy

# TLS Schlüsselaustausch: RSA-DHE



# TLS Schlüsselaustausch: RSA-DHE



# TLS Schlüsselaustausch: RSA-ECDHE

