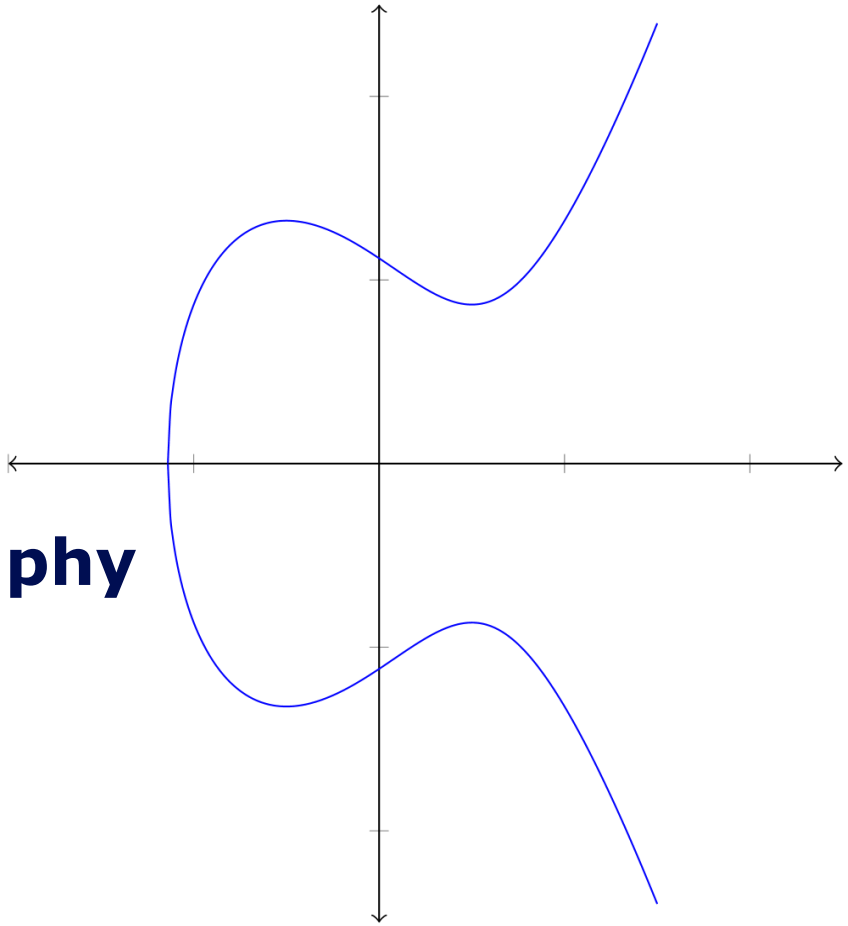
gematik

ECC

Elliptic Curve Cryptography



Christian Grümme
Gematik | Entwickler Communications
christian.gruemme@gematik.de
gruemme

Kryptographisches Verfahren basierend auf elliptischen Kurven

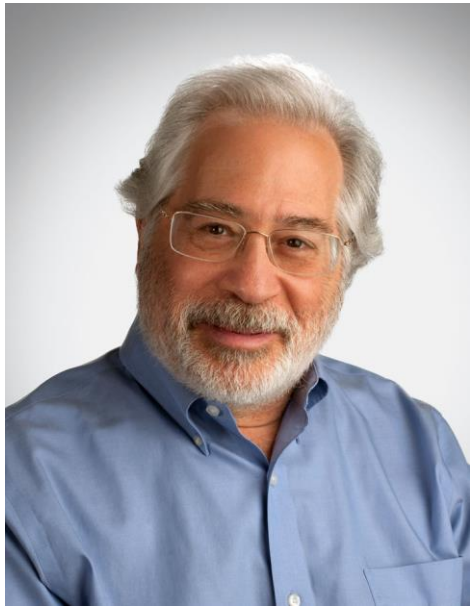
Verfahren als Alternative zu

- Diffie-Hellman-Schlüsselaustausch
- Digitale Signaturen
- Verschlüsselung
- ...

- Elliptic Curve Diffie-Hellman (ECDH)
- Elliptic Curve Integrated Encryption Scheme (ECIES), auch Integrated Encryption Scheme (IES) genannt
- Elliptic Curve Digital Signature Algorithm (ECDSA)

Neal Koblitz

(University of Washington)

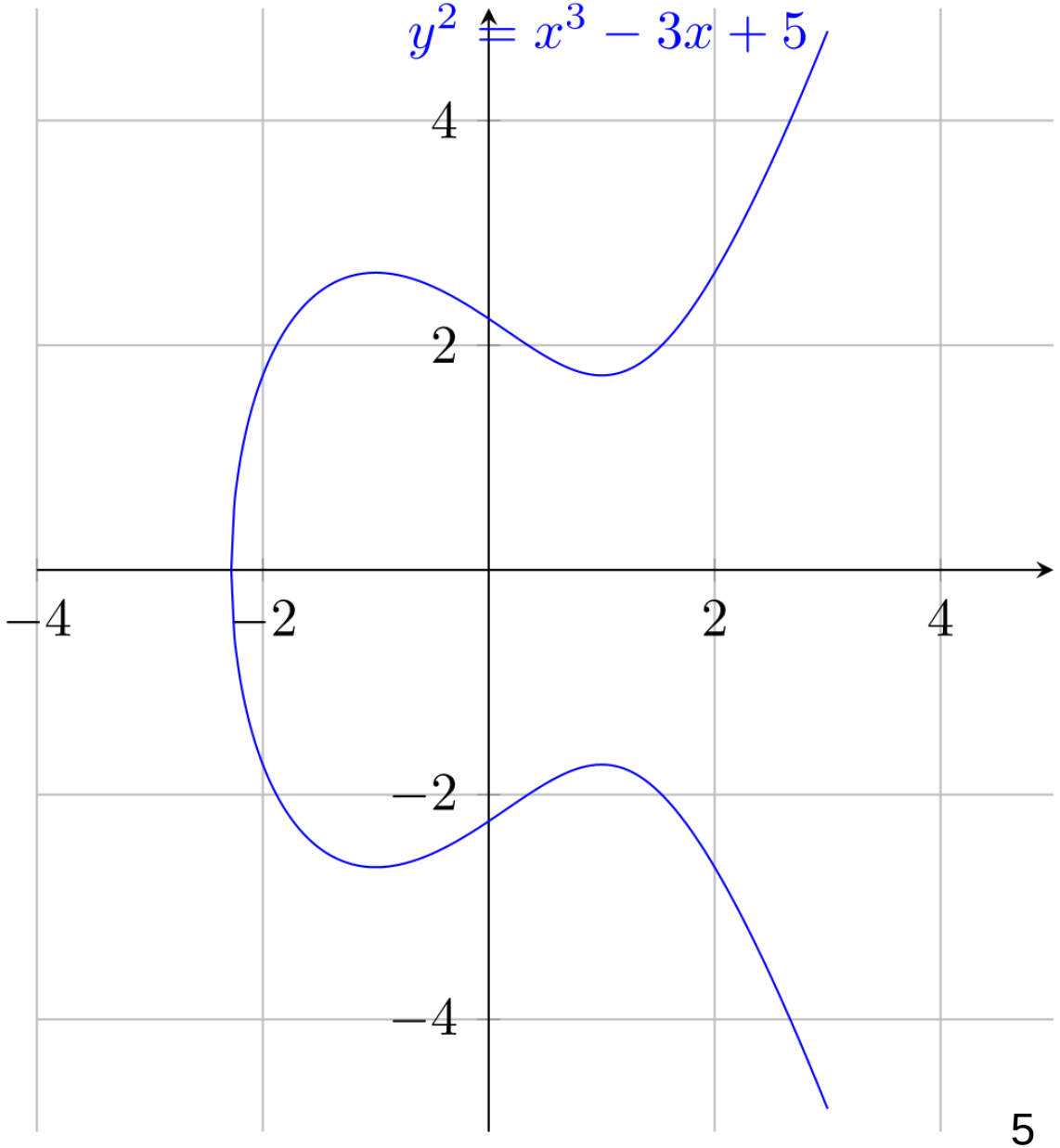


Victor S. Miller

(Thomas J. Watson Research Center, NY)

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

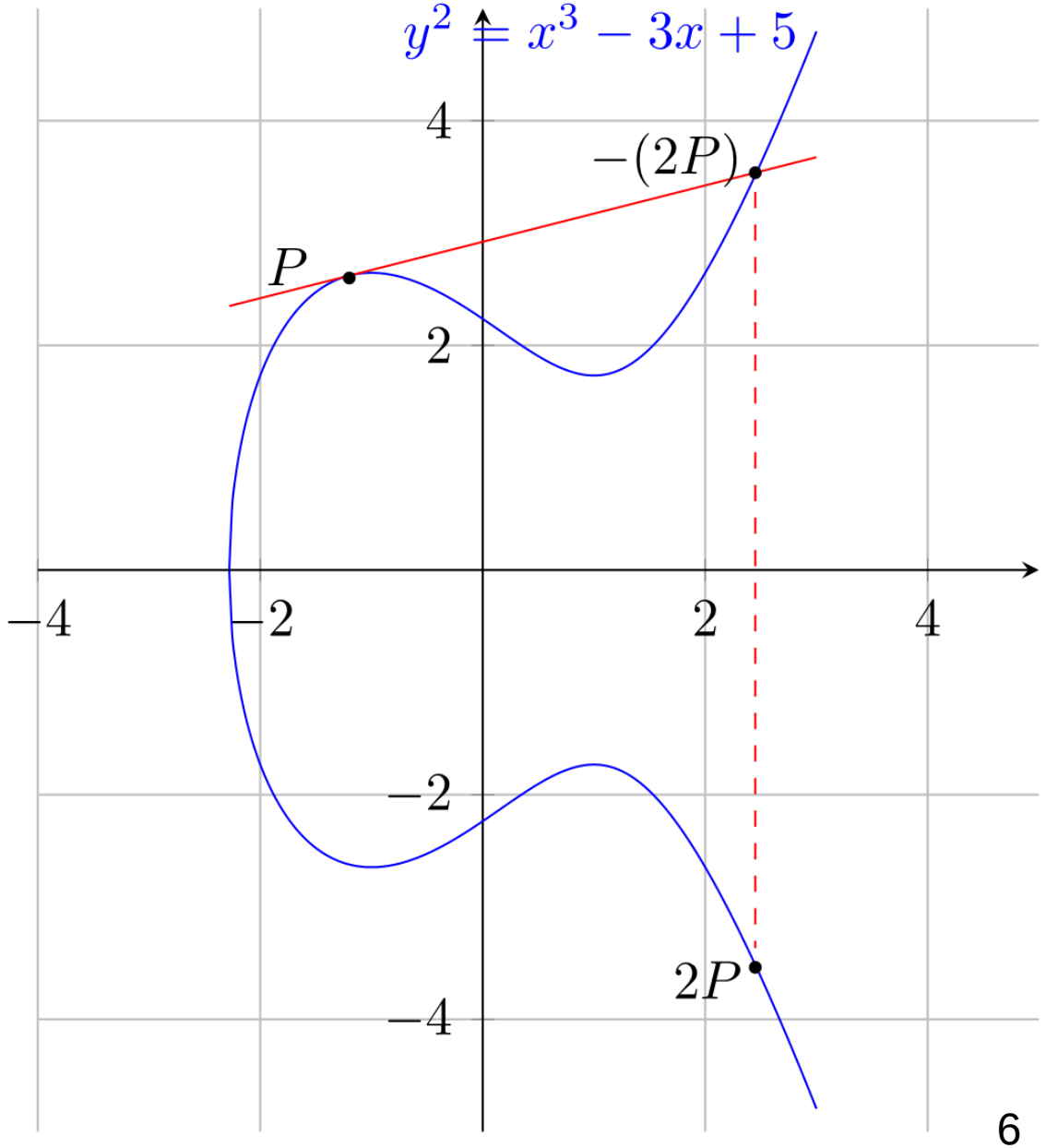
a und *b* müssen geeignet gewählt werden



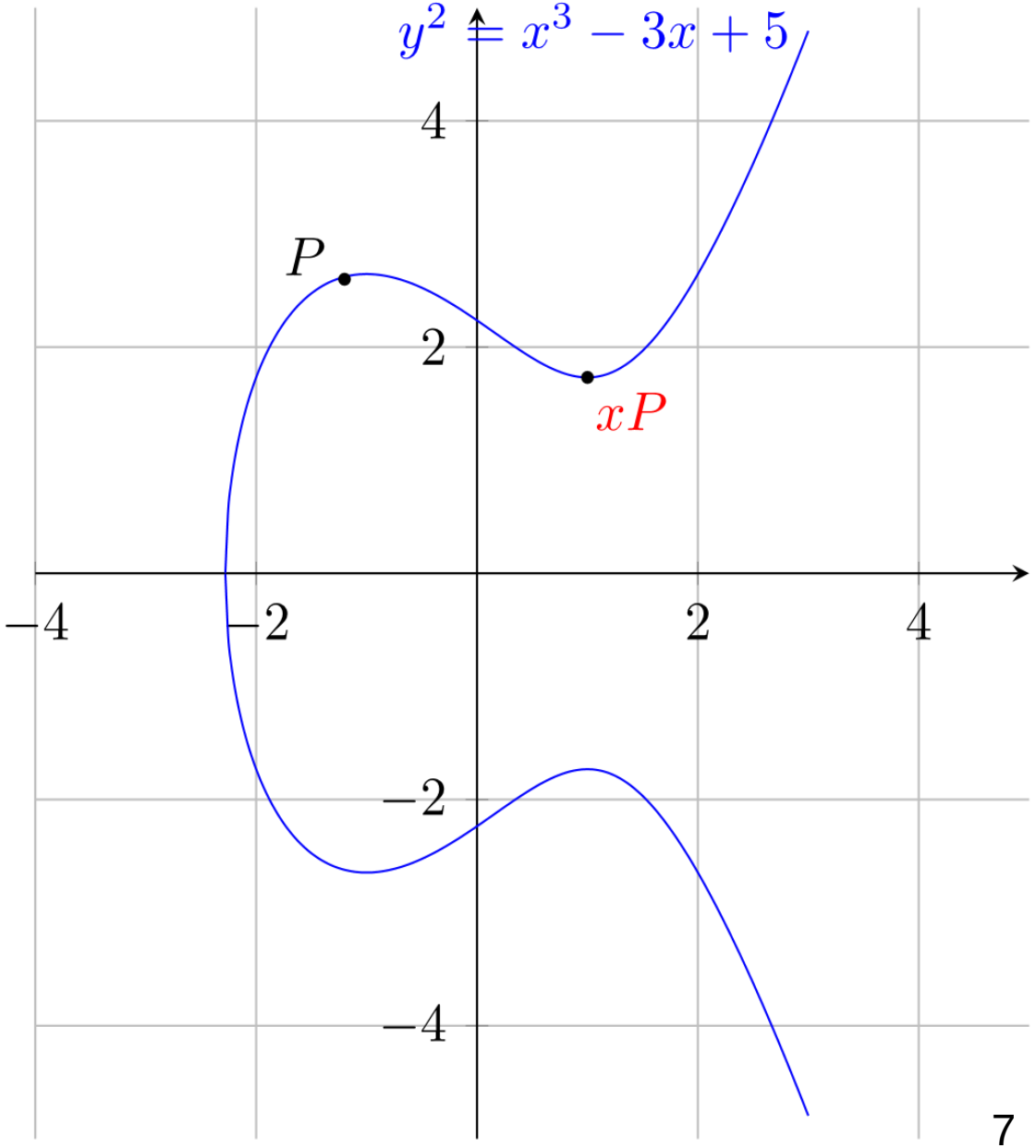
Punkt *P* mit *2* multiplizieren,

also *2P* = *P* + *P*:

1. Tangente an den Punkt *P* legen
2. Schnittpunkten mit der Kurve finden
3. Schnittpunkt an der *x*-Achse spiegeln



- Was ist *x*?

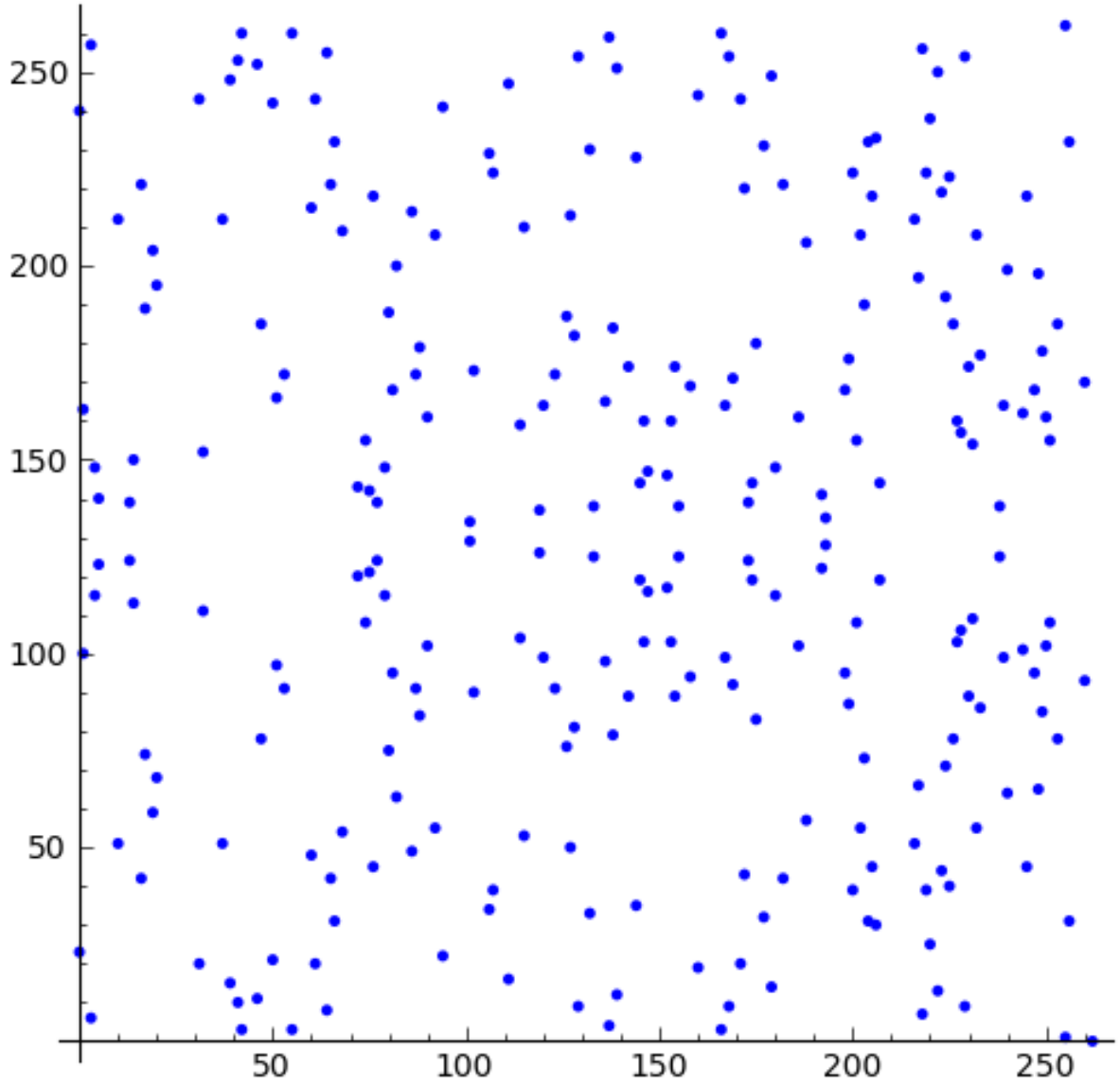


Diskretisierung der Kurve durch Modulo (*G,p,h*)

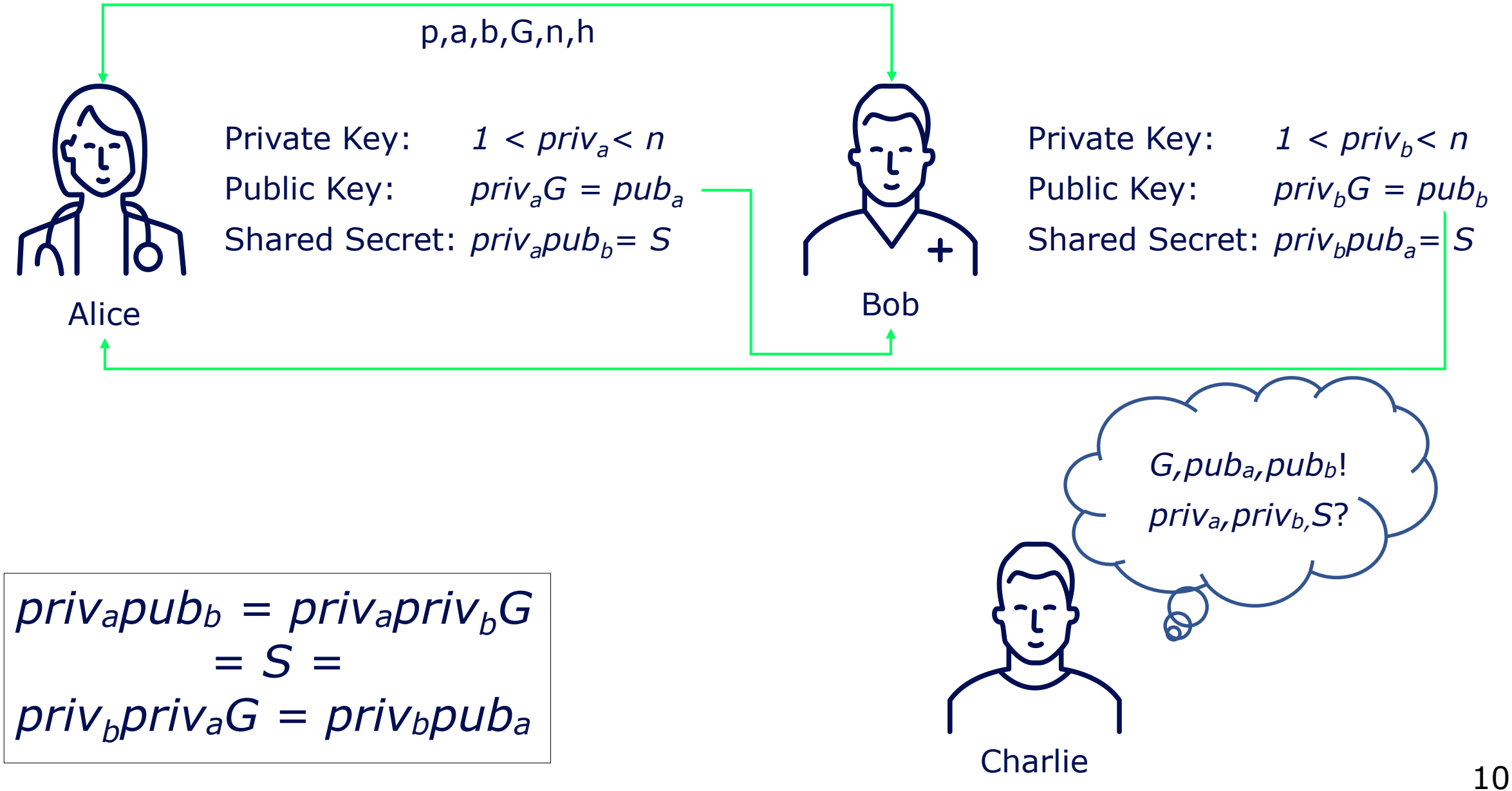
Was ist *x*?

→ Schwer zu lösen

→ *Elliptic Curve Discrete Logarithm Problem*



Parameter bei ECDH	
p	Primzahl (Für Modulo)
a	Kurvenparameter a
b	Kurvenparameter b
G	Basispunkt
n	Ordnung von G
h	cofaktor (Ideal 1)



$$priv_apub_b = priv_apriv_bG = S = priv_bpriv_aG = priv_bpub_a$$

	DHKE	ECDH
Berechnung Shared-Secret	$g^{mn} \bmod p$	$mnG \bmod p$
Aufwand Shared-Secret	$O(\sqrt{nm})$	$O(\log(mn))$
Min. Schlüsselgröße (BSI)	2000 bit	250 bit

- Viele Kurven sind patentiert
- Funktioniert nur mit guter Zufallsfunktion
  - Attacken gegen schlechte Zufallsfunktion existieren
- Sicherheit noch nicht komplett mathematisch bewiesen
- Hintertüren einbaubar, bei spezieller Wahl der Parameter der diskreten Kurve

- Lösen des Diskreter Logarithmus auf Elliptischen Kurven schwieriger als Diskreter Logarithmus
- Für Diskreten Logarithmus sind schon Angriffe bekannt, für ECDH noch nicht
- Deutlich kleinere Bitlängen für die Schlüssel nötig

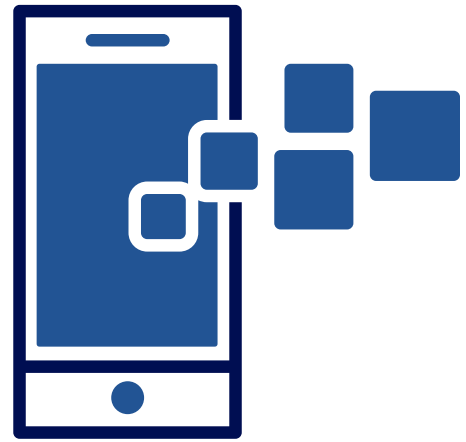
- Koblitz, N. (1987). "Elliptic curve cryptosystems". Mathematics of Computation. 48 (177): 203–209
- Miller, V. (1985). "Use of elliptic curves in cryptography". Advances in Cryptology — CRYPTO '85 Proceedings. CRYPTO. Lecture Notes in Computer Science. 85. pp. 417–426
- Certicom Research, Standards for efficient cryptography, SEC 1: Elliptic Curve Cryptography, May 21, 2009.
- D. Fu, J. Solinas (2010). "ECP Groups for IKE and IKEv2". RFC 5903, May 18, 2010
- http://safecurves.cr.yp.to/rho.html
- https://www.keylength.com/
- BSI - "Technical Guideline TR-02102-2 Cryptographic Mechanisms:". Recommendations and Key Lengths, 2021

- Neal Koblitz (Quelle: University of Washington, https://math.washington.edu/people/neal-i-koblitz)
- Victor S. Miller (Quelle: Eduard-Rhein-Stiftung, https://www.eduard-rhein-stiftung.de/preistrager/)
- Diskrete Elliptische Kurve - Sage Plot einer diskreten Kurve mit a = 2 und b = 3 über Finites Feld von 263 Elementen (Quelle: Johannes Bauer, https://www.johannes-bauer.com/compsci/ecc/#anchor37)

gematik

Mobile Plattformen

TLS 1.3 verwenden



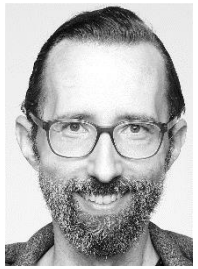
Martin Fiebig

Gematik | iOS Development E-Rezept

martin.fiebig@gematik.de

mrtnfbg

mfiebig



Joachim Gärtner

Gematik | Android Development E-Rezept

joachim.gaertner@gematik.de

fnordlicht

fnordlicht

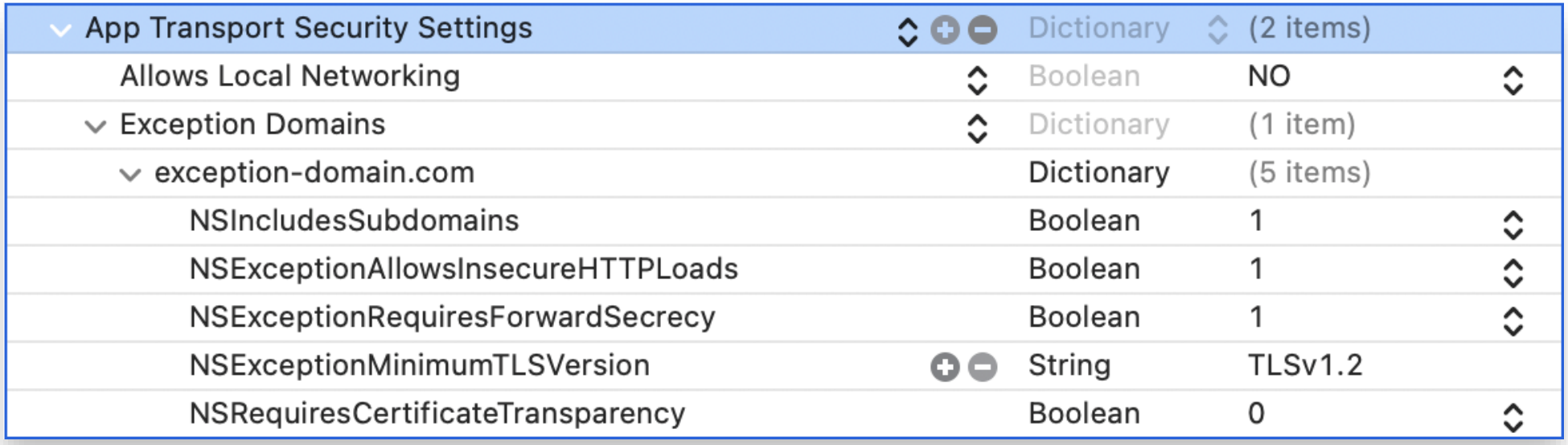
TL;DR

Wir sind fertig

- TLS Konfiguration ist Teil der **A**pp **T**ransport **S**ecurity (ATS) Konfiguration
  - Konfiguration via *Info.plist*
  - Konfiguration via Quellcode
- Referenz: https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html

Konfiguration via *Info.plist*

- Globale exceptions
- Domain specific exceptions



Info.plist

Konfiguration Quellcode

- Teil von URLSessionConfiguration
- Guter Startpunkt: URLSessionConfiguration.ephemeral

```
36      public init(
37          urlSessionConfiguration: URLSessionConfiguration,
38          interceptors: [Interceptor] = [],
39          delegateQueue: OperationQueue? = nil
40      ) {
41          let delegate = ProxyDelegate()
42
43          // [REQ:gemSpec_Krypt:GS-A_4385,A_18467,A_18464,GS-A_4387]
44          // [REQ:gemSpec_Krypt:GS-A_5322] TODO: Check if limiting SSL Sessions is poss
45          // swiftlint:disable:previous todo
46          // [REQ:gemSpec_IDP_Frontend:A_20606] Live URLs not present in NSAppTransport
47          // HTTP communication
48          // [REQ:gemSpec_eRp_FdV:A_20206]
49
50          urlSessionConfiguration.tlsMinimumSupportedProtocolVersion = .TLSv12
51          urlSession = .init(configuration: urlSessionConfiguration, delegate: delegate
52          self.interceptors = interceptors
53          self.delegate = delegate
```

Sources/HTTPClient/DefaultHTTPClient.swift#L36

Square OkHttp

- OkHttp.ConnectionSpec

```
286     private fun getConnectionSpec(): List<ConnectionSpec> = ConnectionSpec
287         .Builder(ConnectionSpec.RESTRICTED_TLS)
288         .tlsVersions(
289             TlsVersion.TLS_1_2,
290             TlsVersion.TLS_1_3
291         )
292         .cipherSuites(
293             // TLS 1.2
294             CipherSuite.TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
295             CipherSuite.TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
296             CipherSuite.TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
297             CipherSuite.TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
298             // TLS 1.3
299             CipherSuite.TLS_AES_128_GCM_SHA256,
300             CipherSuite.TLS_AES_256_GCM_SHA384,
301             CipherSuite.TLS_CHACHA20_POLY1305_SHA256
302         )
303         .build()
304         .let { listOf(it) }
```

NetworkingModule.kt#L286

Danke für Ihre Aufmerksamkeit