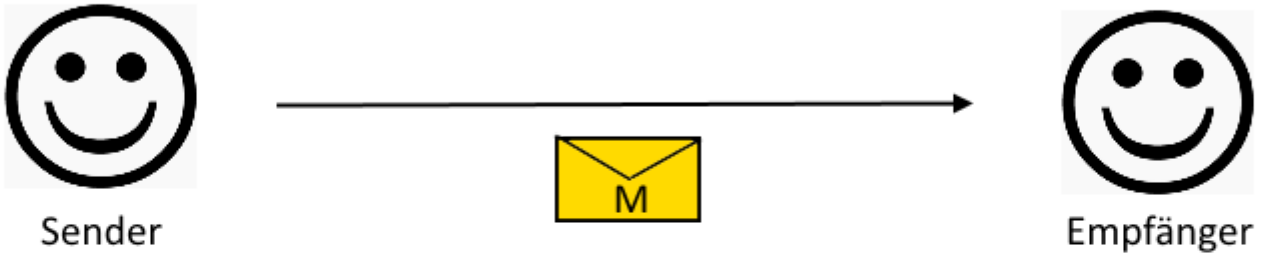
Hashbasierte Signaturen

Ein Vortrag von Oleg Geger

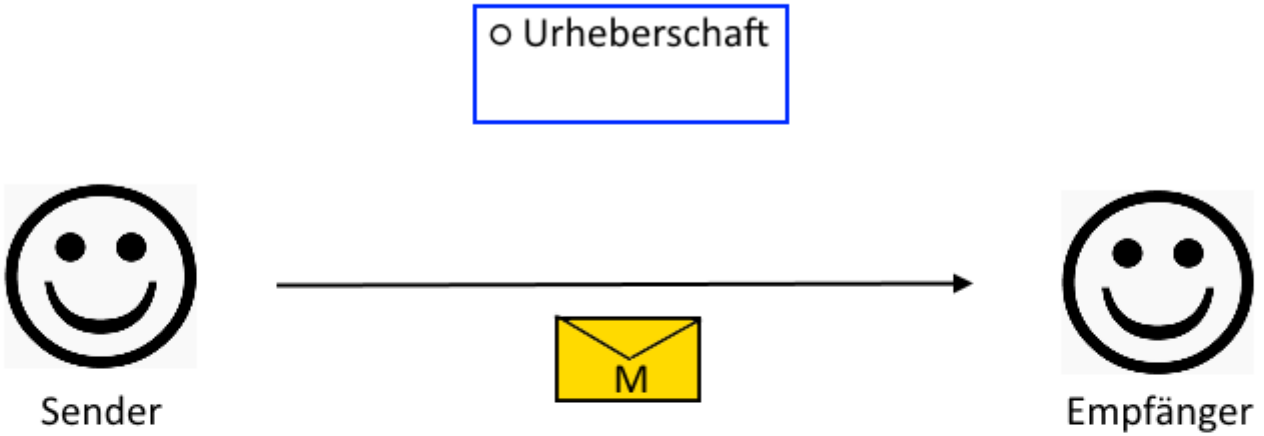
Gliederung

1. Digitale Signaturen - Allgemein
2. Motivation für hashbasierte Signaturen
3. Einmalsignaturen
   - 3.1 Lamport-Diffie-Einmalsignatur
   - 3.2 Winternitz-Einmalsignatur
4. Merkle-Signatur
   - 4.1 Standard-Verfahren
   - 4.2 Erweiterung
5. Fazit (Praktikabilität)
6. Quellen

Digitale Signaturen - Allgemein

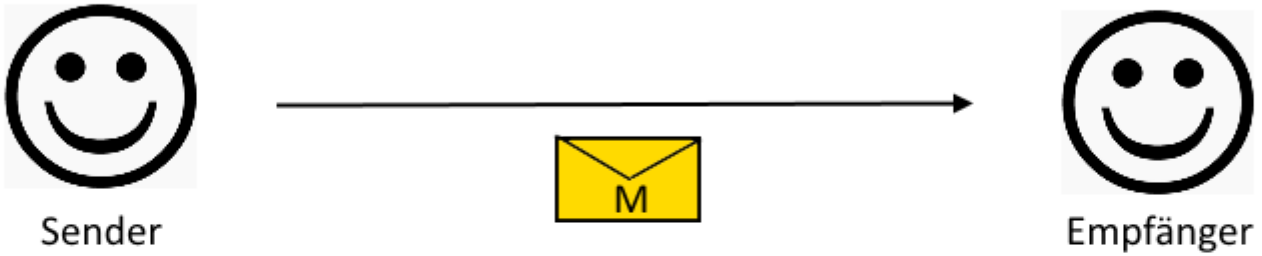


Digitale Signaturen - Allgemein



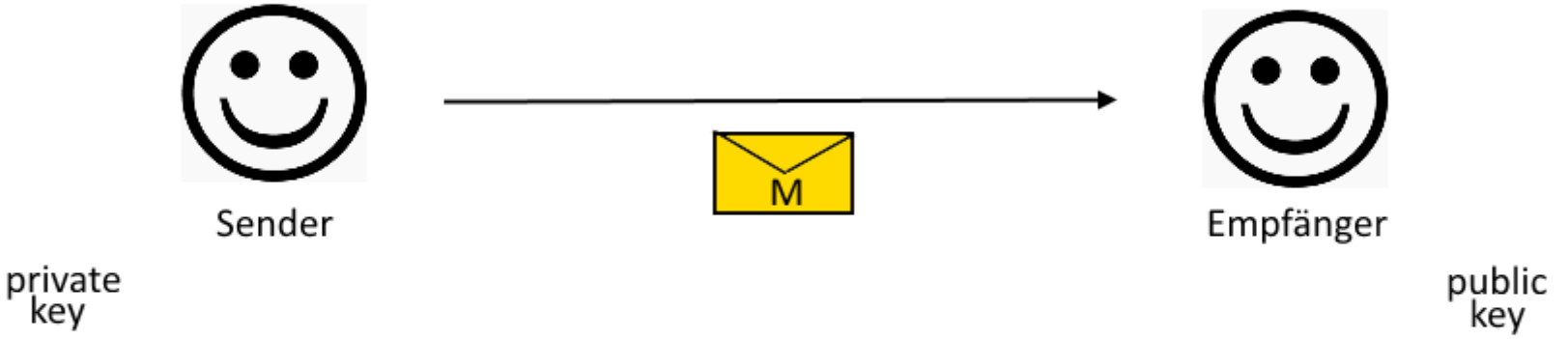
Digitale Signaturen - Allgemein

- Urheberschaft
- Integrität

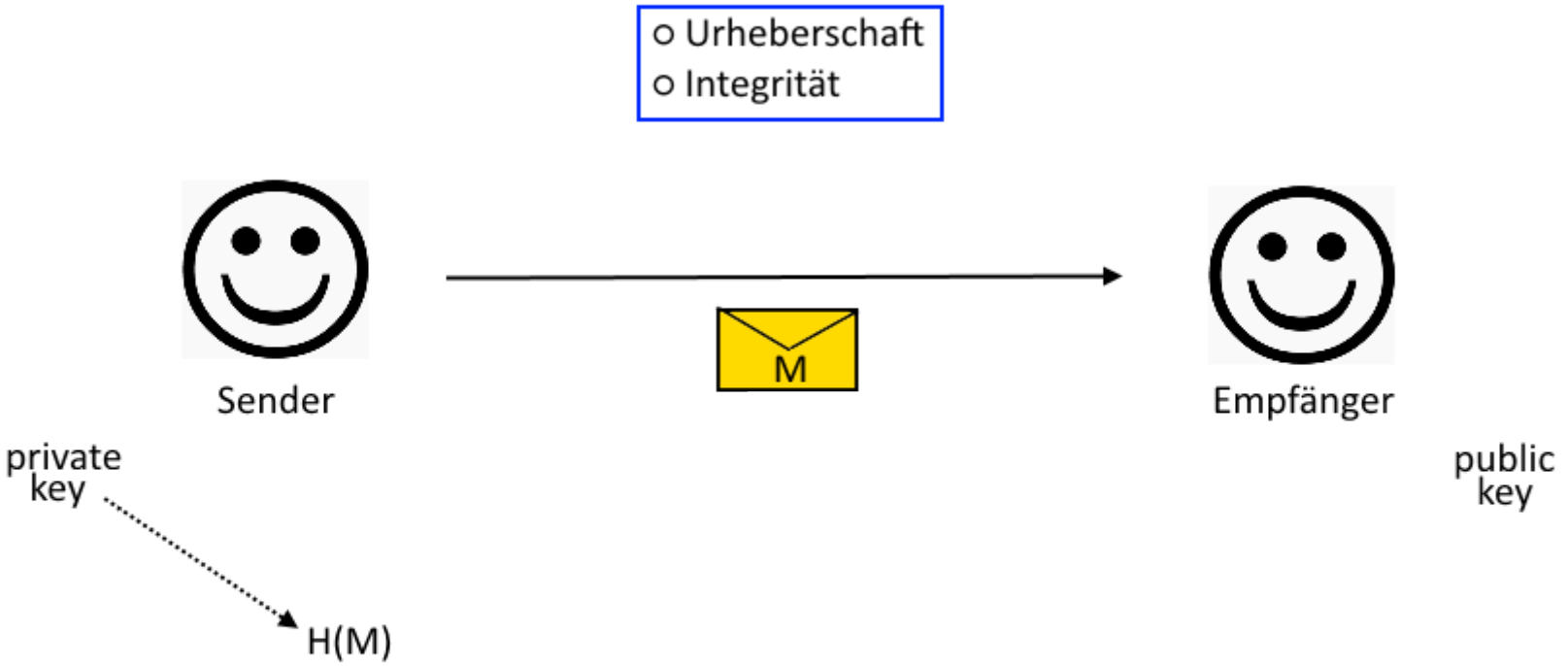


Digitale Signaturen - Allgemein

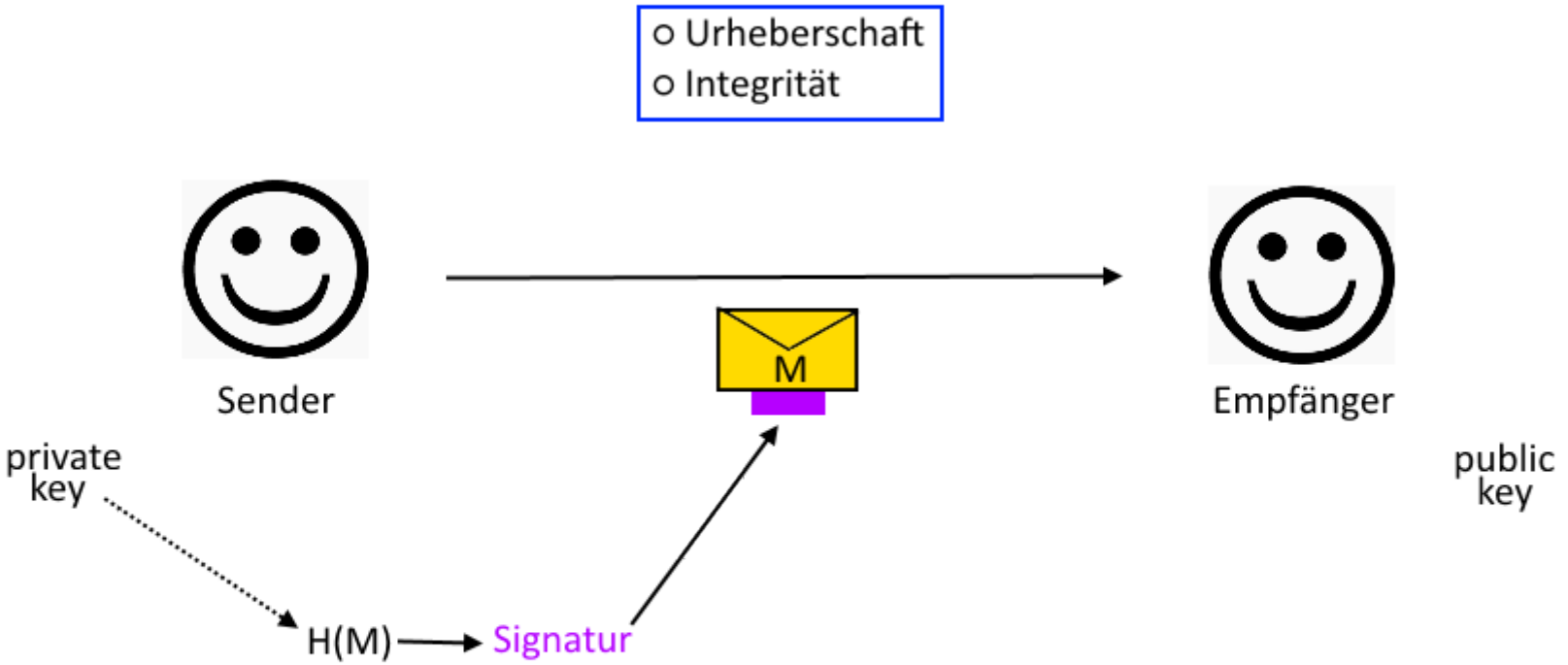
- Urheberschaft
- Integrität



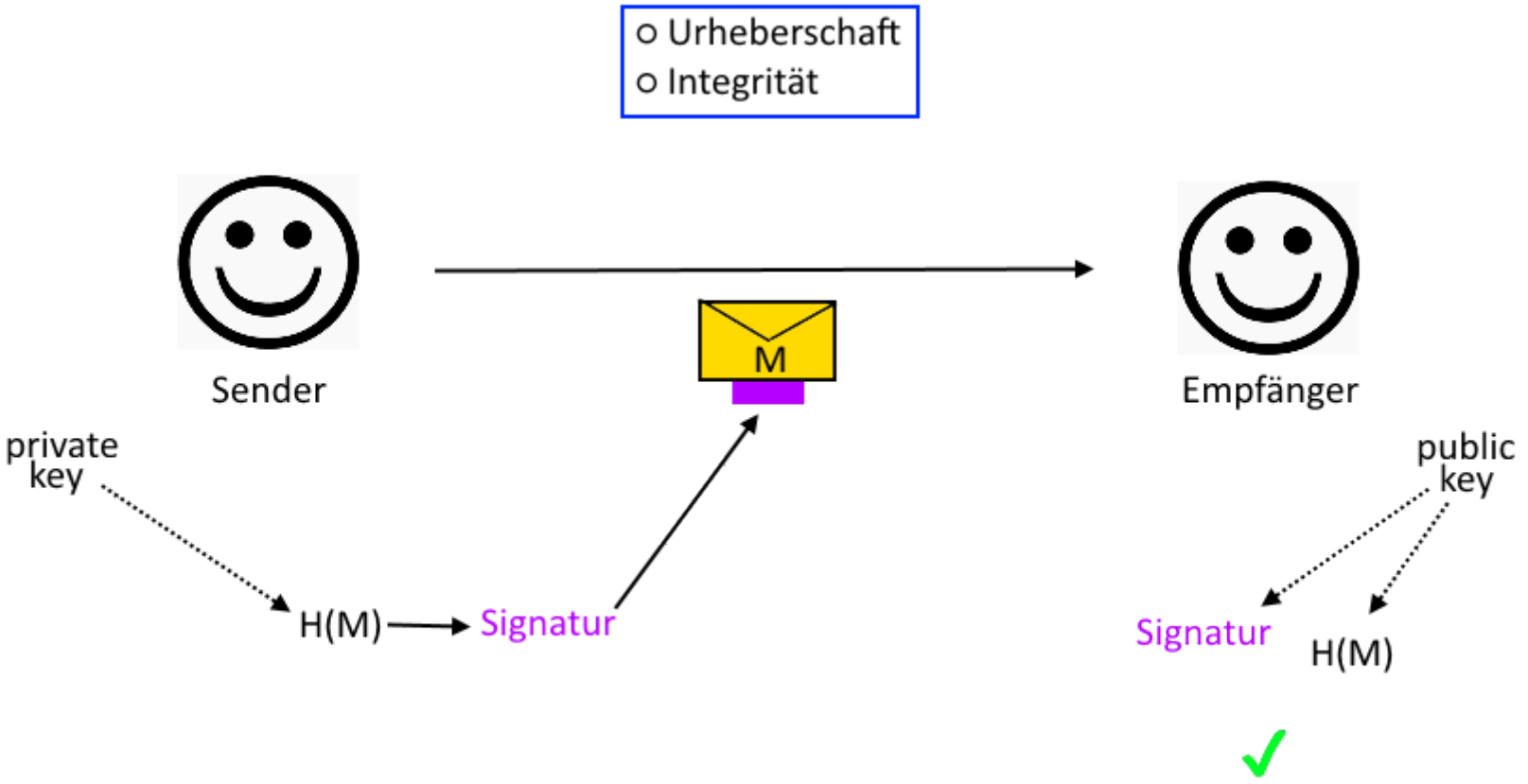
Digitale Signaturen - Allgemein



Digitale Signaturen - Allgemein

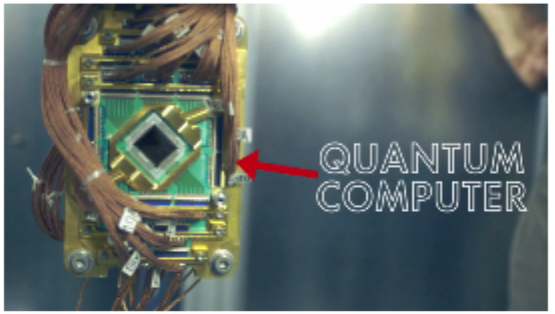


Digitale Signaturen - Allgemein

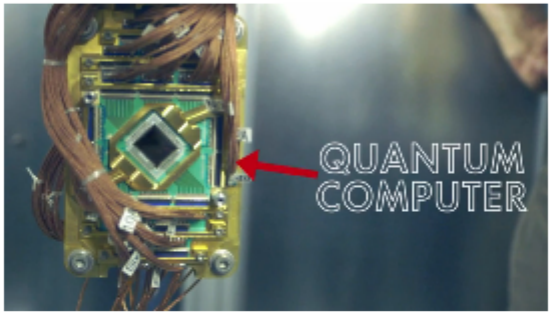


Motivation für hashbasierte Signaturen

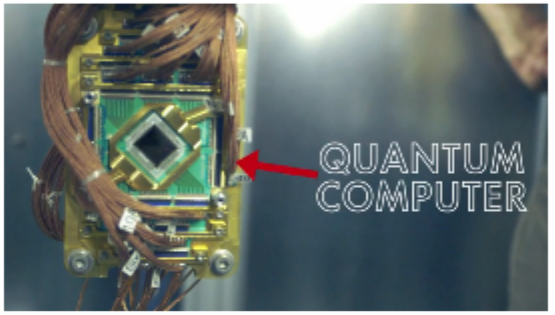
Motivation für hashbasierte Signaturen



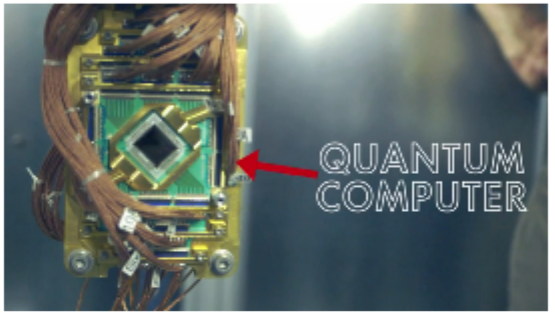
Motivation für hashbasierte Signaturen



Motivation für hashbasierte Signaturen

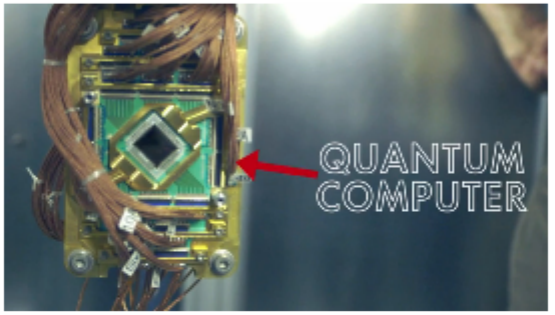


Motivation für hashbasierte Signaturen



RSA

Motivation für hashbasierte Signaturen



kollisionsresistente,
Einweg-Hashfunktionen

(hashbasierte) Einmalsignaturen

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1? c1? , ,)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1 , ,)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1 , a2? c2? ,)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) =	0		1		0	
S = (	a1	,	c2	,		)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) =	0		1		0	
S = (	a1	,	c2	,	a3?	c3?)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1 , c2 , a3)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1 , c2 , a3)

Verifikation

wir wissen:
(H(a1), H(c2), H(a3)) = (b1, d2, b3)

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1 , c2 , a3)

Verifikation

wir wissen:
(H(a1), H(c2), H(a3)) = (b1, d2, b3)

S = (s1, s2, s3)

H(M) = 010

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys:	A	C
public Keys:	B	D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1 , c2 , a3)

Verifikation

wir wissen:
(H(a1), H(c2), H(a3)) = (b1, d2, b3)

S = (s1, s2, s3)
H(M) = 010

prüfe:
(H(s1), H(s2), H(s3)) == (b1, d2, b3) ???

Lamport-Diffie-Einmalsignatur

M = 10101101
H(M) = 010

private Keys: A C
public Keys: B D

Schlüsselerzeugung

erzeuge private Keys zufällig:
z.B. A = (a1, a2, a3) = (010, 001, 110)
C = (c1, c2, c3) = (100, 101, 010)

erzeuge public Keys:
B = (b1, b2, b3) = (H(a1), H(a2), H(a3))
D = (d1, d2, d3) = (H(c1), H(c2), H(c3))

Signieren

H(M) = 0 1 0

S = (a1 , c2 , a3)

Verifikation

wir wissen:
(H(a1), H(c2), H(a3)) = (b1, d2, b3)

S = (s1, s2, s3)
H(M) = 010

prüfe:
(H(s1), H(s2), H(s3)) == (b1, d2, b3) ???

Signaturgröße

Bei Verwendung von SHA-256:

256 * 256 = 65536 (Bit)

ziemlich gross !

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: $\{0, 1\}^* \rightarrow \{0, 1\}^{256}$

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
yi = H^{255}(xi)

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
yi = H^{255}(xi)

Signieren

H(M) = (h64, h63, ... , h1)

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
yi = H^{255}(xi)

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
yi = H^{255}(xi)

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

(h66, h65, ... , h1)
dabei hi aus {0 ... 255}

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
yi = H^{255}(xi)

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

(h66, h65, ... , h1)
dabei hi aus {0 ... 255}

S = (s66, s65, ... , s1) mit
si = H^{hi} (xi)

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
yi = H^{255}(xi)

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

(h66, h65, ... , h1)
dabei hi aus {0 ... 255}

S = (s66, s65, ... , s1) mit
si = H^{hi} (xi)

Verifikation

berechne:
(h66, h65, ... , h1)

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
yi = H^{255}(xi)

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

(h66, h65, ... , h1)
dabei hi aus {0 ... 255}

S = (s66, s65, ... , s1) mit
si = H^{hi} (xi)

Verifikation

berechne:
(h66, h65, ... , h1)

Z = (z66, z65, ... , z1) mit
zi = $H^{255 - hi}$ (si)

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
$y_i = H^{255}(x_i)$

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

(h66, h65, ... , h1)
dabei hi aus {0 ... 255}

S = (s66, s65, ... , s1) mit
$s_i = H^{h_i}(x_i)$

Verifikation

berechne:
(h66, h65, ... , h1)

Z = (z66, z65, ... , z1) mit
$z_i = H^{255 - h_i}(s_i)$

Bsp: Sei h1 = 240
Dann: $z1 = H^{15}(s1) = H^{15}(H^{240}(x1)) = H^{255}(x1) = y1$

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
H: {0, 1}* --> $\{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
$y_i = H^{255}(x_i)$

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

(h66, h65, ... , h1)
dabei hi aus {0 ... 255}

S = (s66, s65, ... , s1) mit
$s_i = H^{h_i}(x_i)$

Verifikation

berechne:
(h66, h65, ... , h1)

Z = (z66, z65, ... , z1) mit
$z_i = H^{255 - h_i}(s_i)$

Bsp: Sei h1 = 240
Dann: $z1 = H^{15}(s1) = H^{15}(H^{240}(x1)) = H^{255}(x1) = y1$

prüfe: Z == Y ?

Winternitz-Einmalsignatur

M bzw. H(M) zu signieren
w = 8
$H: \{0, 1\}^* \rightarrow \{0, 1\}^{256}$

=> Schlüssel aus 66 256-Bit-Strings

Schlüsselgenerierung

private Key:
X = (x66, x65, ... , x1) mit
xi = zufällige 256-Bit-Folge

public Key:
Y = (y66, y65, ... , y1) mit
$y_i = H^{255}(x_i)$

Signieren

H(M) = (h64, h63, ... , h1)
berechne h65 und h66 als
Prüfsumme aus h1 bis h64

(h66, h65, ... , h1)
dabei hi aus {0 ... 255}

S = (s66, s65, ... , s1) mit
$s_i = H^{h_i}(x_i)$

Verifikation

berechne:
(h66, h65, ... , h1)

Z = (z66, z65, ... , z1) mit
$z_i = H^{255 - h_i}(s_i)$

Bsp: Sei h1 = 240
Dann: $z1 = H^{15}(s1) = H^{15}(H^{240}(x1)) = H^{255}(x1) = y1$

prüfe: Z == Y ?

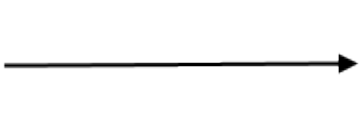
Signaturgröße

66 * 256 = 16896 (Bit) < 65536 (Bit) (aus LD-Sign.)

Probleme mit Einmalsignaturen?

Merkle-Signatur

Merkle-Signatur



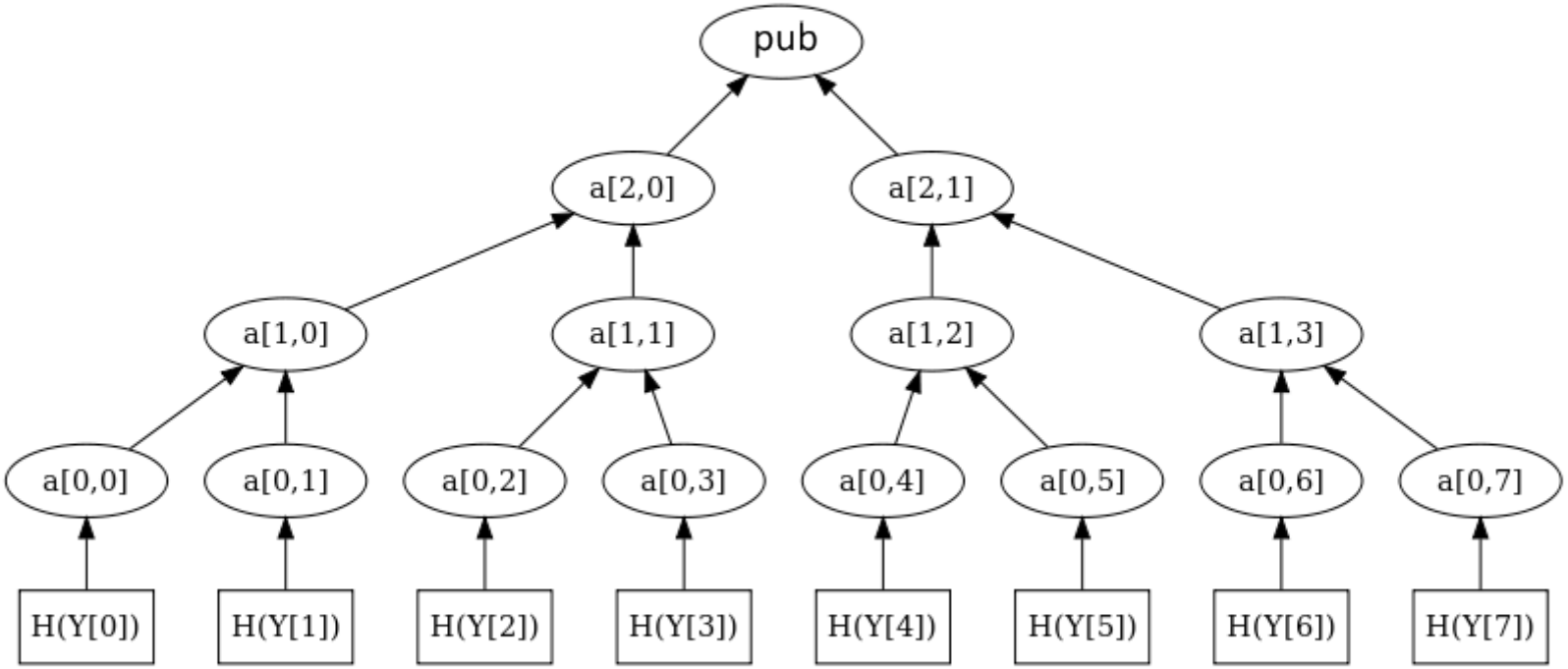
pub

Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys

Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys



Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys

Signieren

Bsp: wir benutzen X2 und Y2

Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys

Signieren

Bsp: wir benutzen X2 und Y2

signiere wie gewohnt mit der
Einmalsignatur (z.B. Winternitz)

==> sig'

Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys

Signieren

Bsp: wir benutzen X2 und Y2

signiere wie gewohnt mit der
Einmalsignatur (z.B. Winternitz)

==> sig'

Sende an Empfänger:
Message M
Signatur sig'

Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys

Signieren

Bsp: wir benutzen X2 und Y2

signiere wie gewohnt mit der
Einmalsignatur (z.B. Winternitz)

==> sig'

Sende an Empfänger:
Message M
Signatur sig'
public key Y2

Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys

Signieren

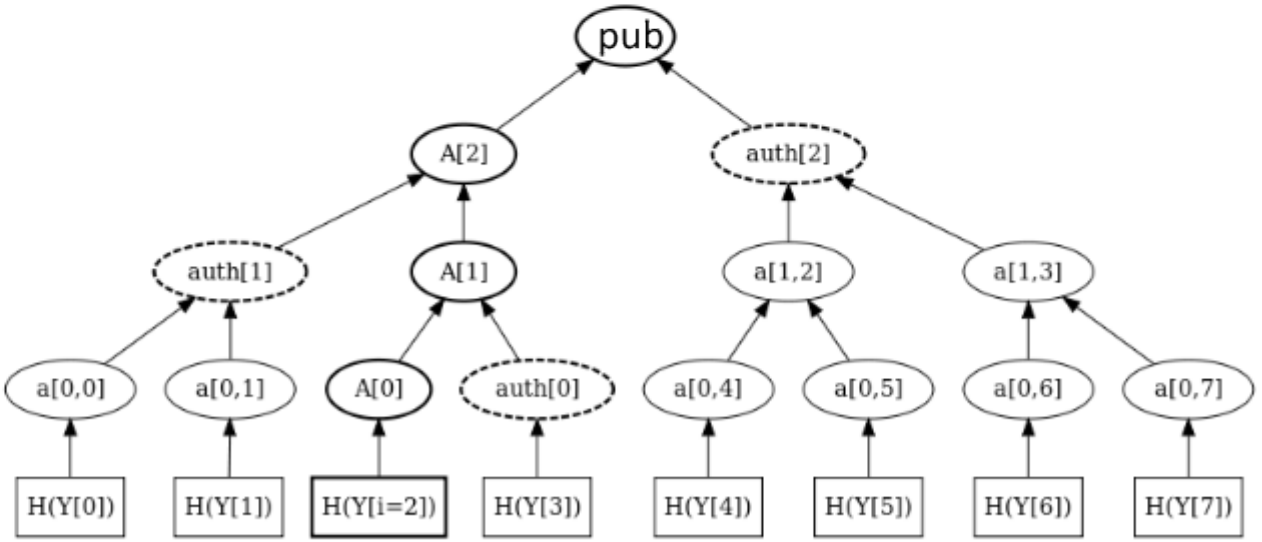
Bsp: wir benutzen X2 und Y2

signiere wie gewohnt mit der
Einmalsignatur (z.B. Winternitz)

==> sig'

Sende an Empfänger:
Message M
Signatur sig'
public key Y2

auth[0]
auth[1]
auth[2]



Merkle-Signatur (Standardverfahren)

Seien X0 bis X7 meine private Keys und
Y0 bis Y7 meine public keys

Signieren

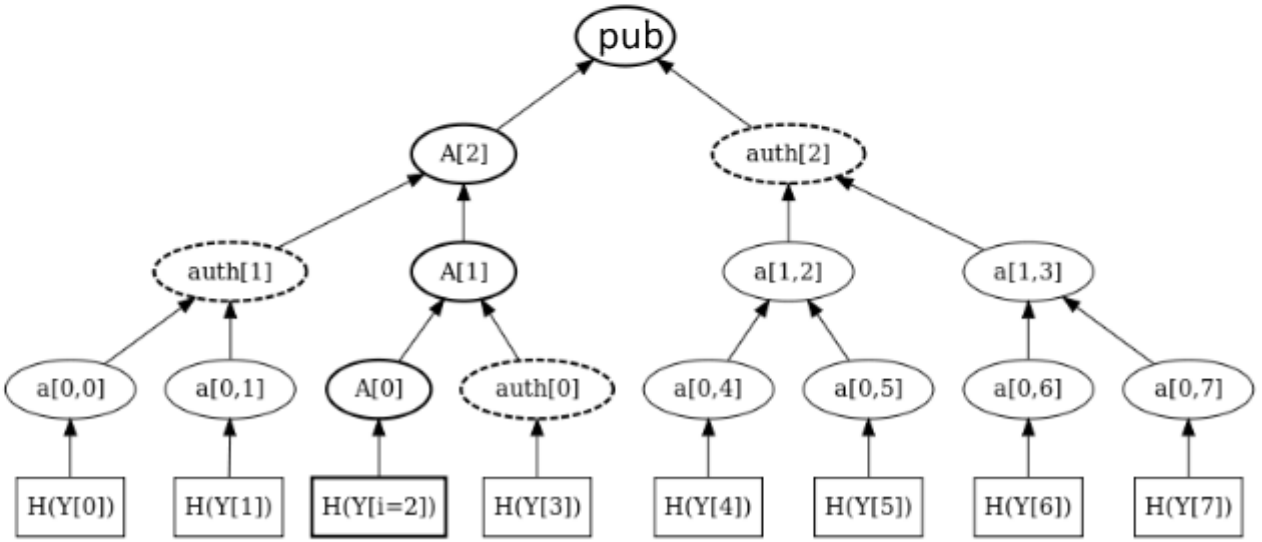
Bsp: wir benutzen X2 und Y2

signiere wie gewohnt mit der Einmalsignatur (z.B. Winternitz)

==> sig'

Sende an Empfänger:
Message M
Signatur sig'
public key Y2
auth[0]
auth[1]
auth[2]

Signatur sig



Merkle-Signatur

Probleme mit dem Standardverfahren?

Merkle-Signatur

8 711 228 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 Einträge

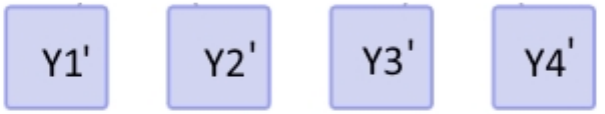
Merkle-Signatur (Erweiterung)

Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

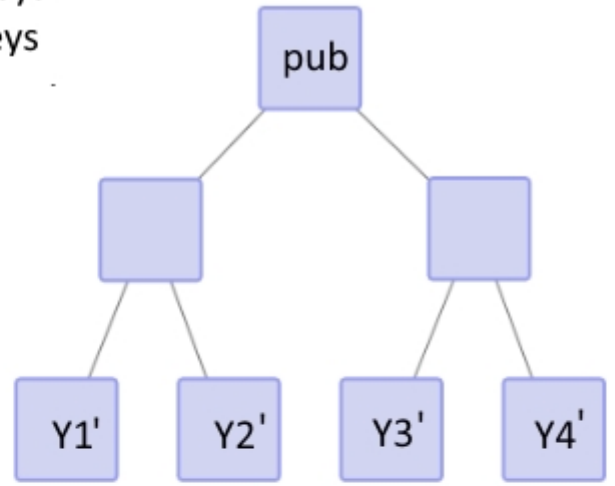
Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys



Merkle-Signatur (Erweiterung)

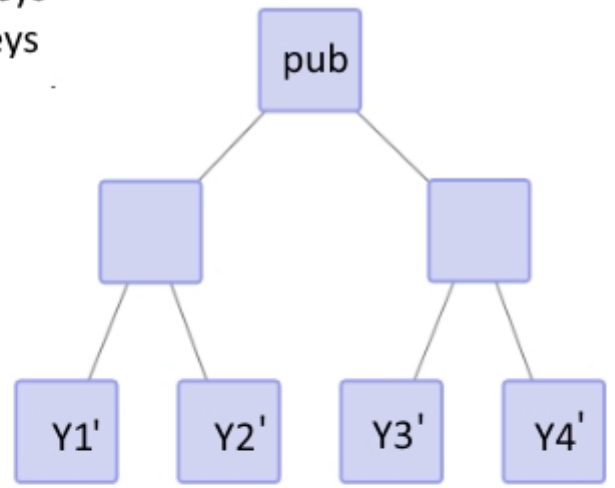
Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3



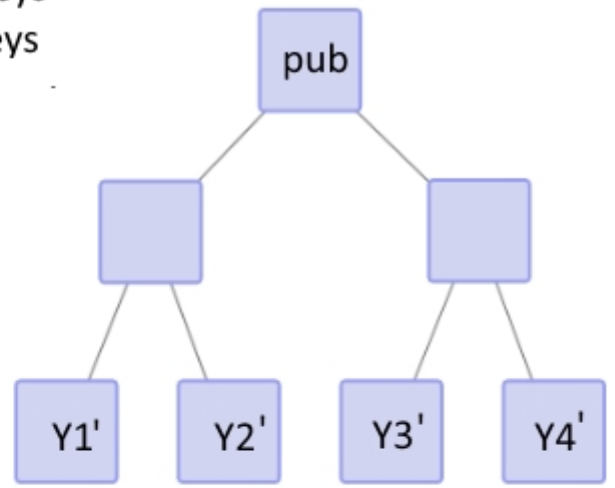
Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)



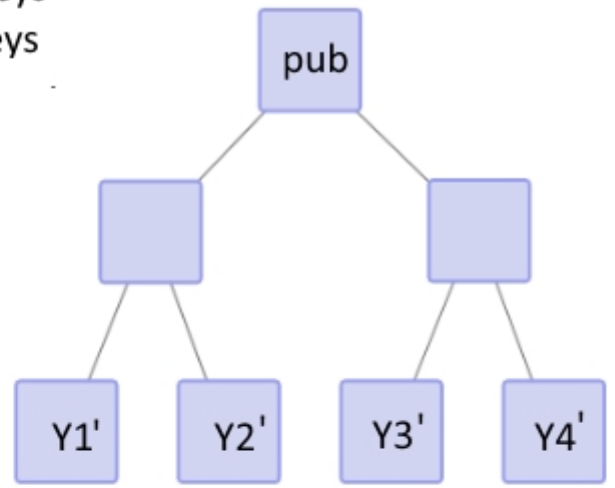
Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)



Y5' Y6' Y7' Y8'

Y9' Y10' Y11' Y12'

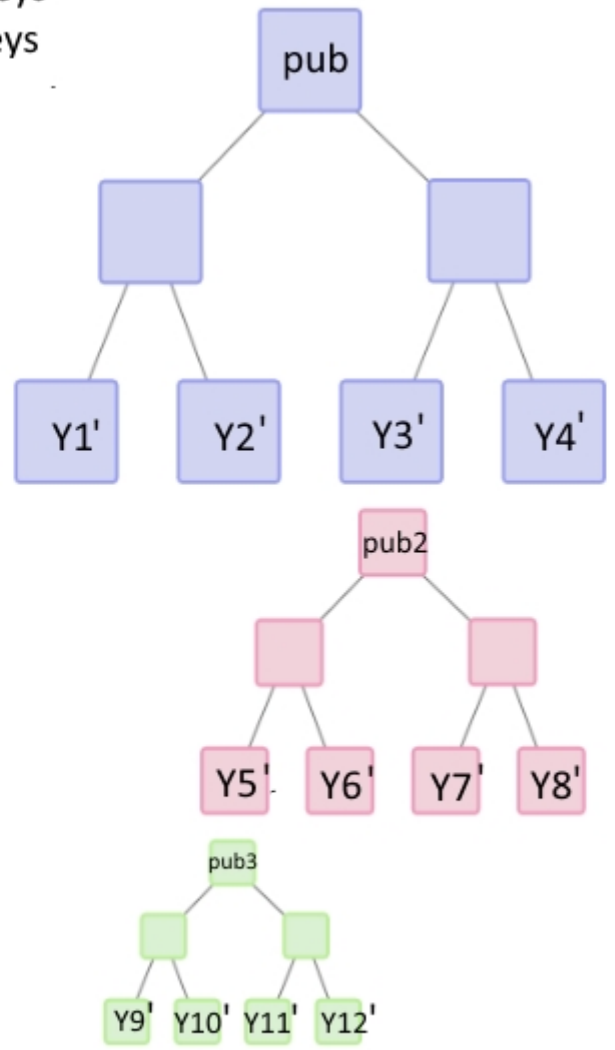
Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)



Merkle-Signatur (Erweiterung)

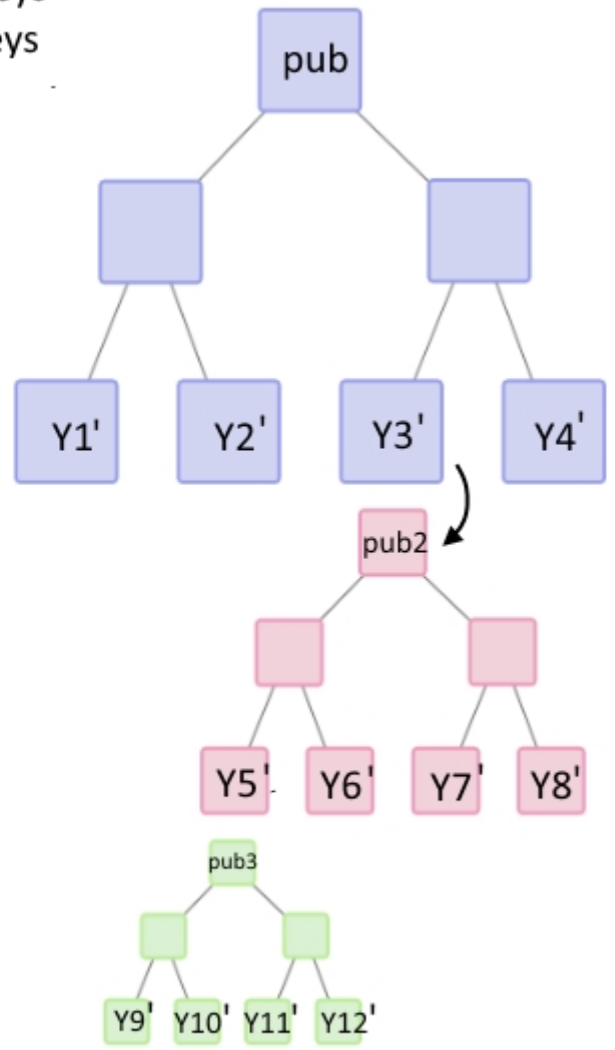
Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

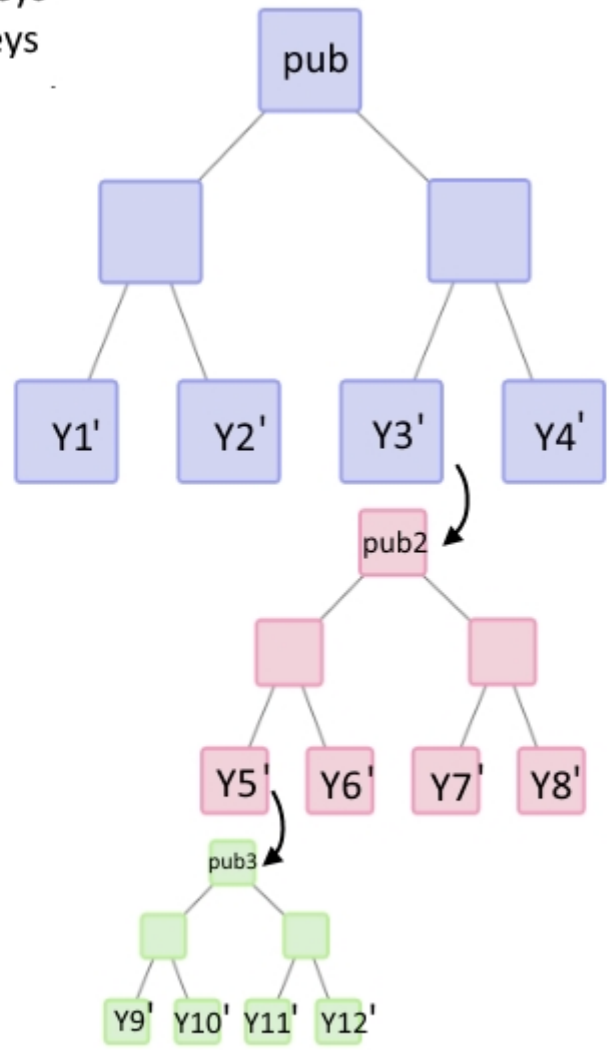
Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

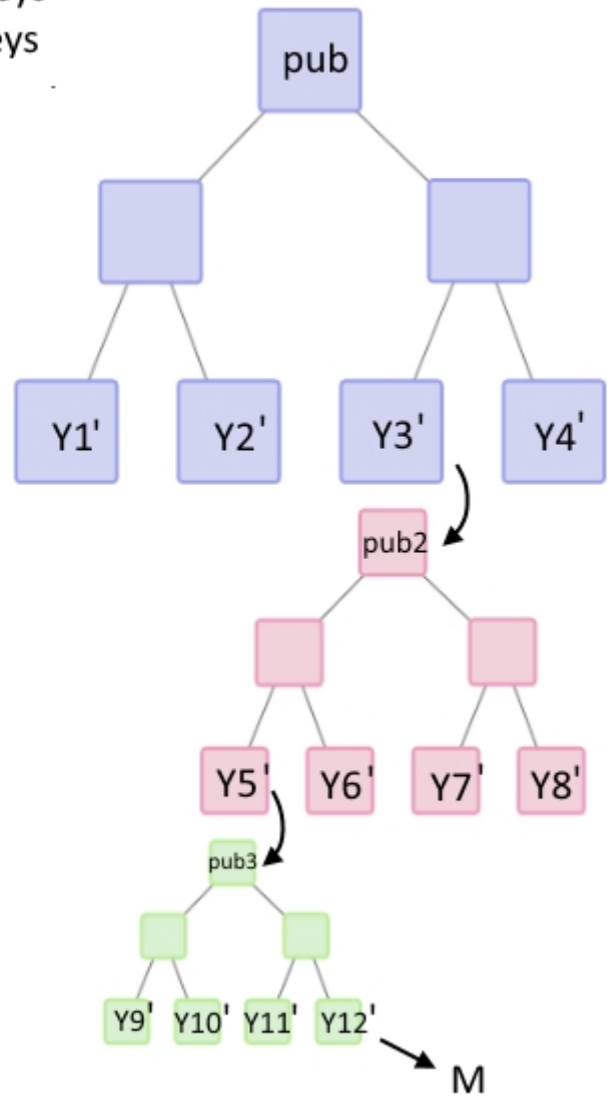
Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

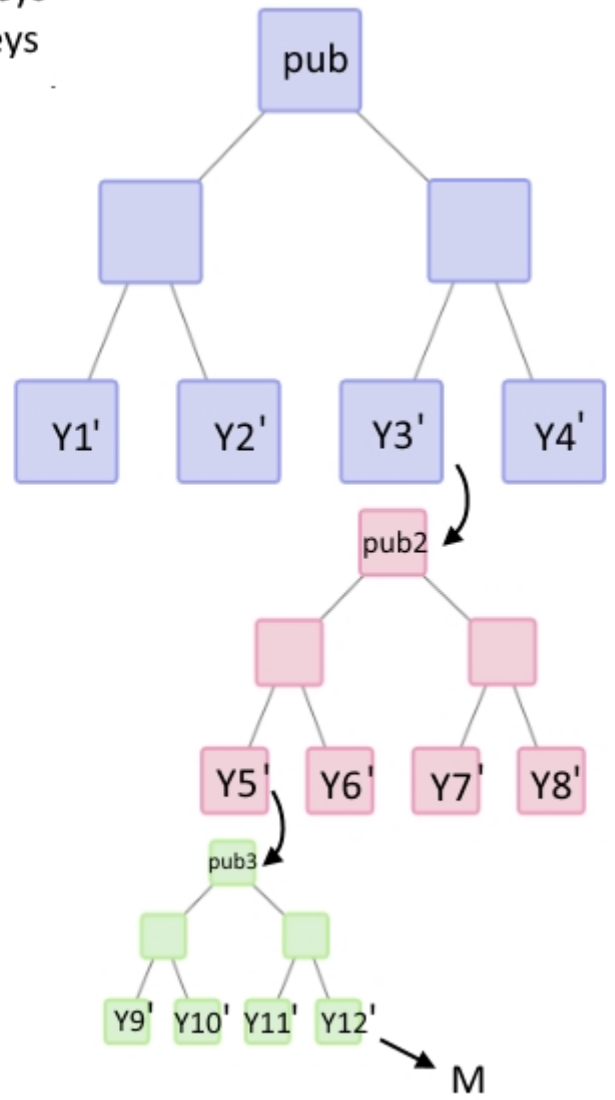
signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M

s1, s2, s3
Y3, Y5, Y12



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

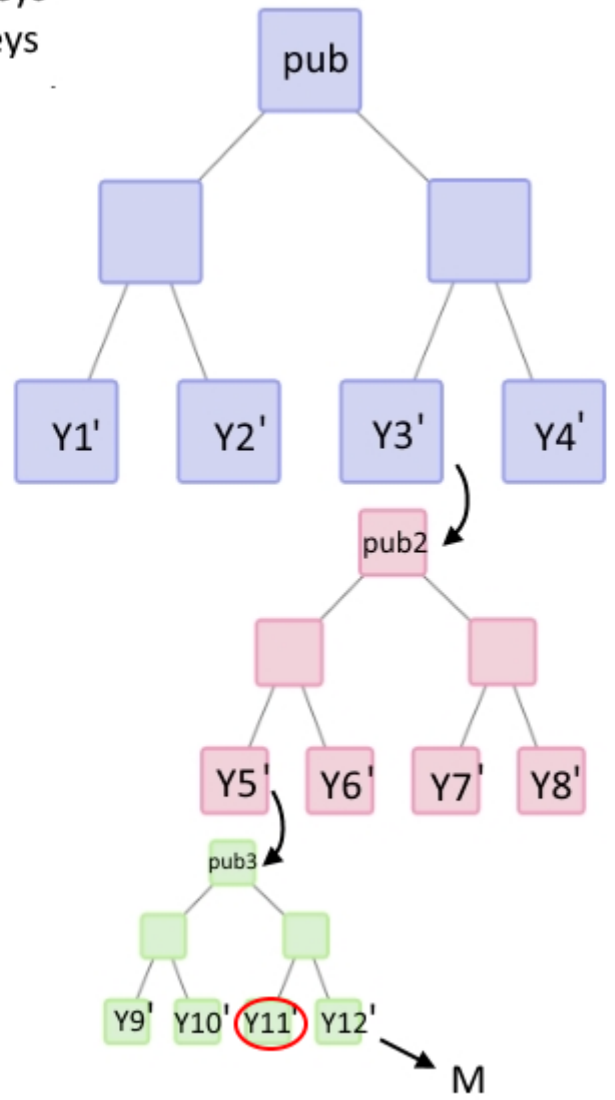
signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M

s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn ◯



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

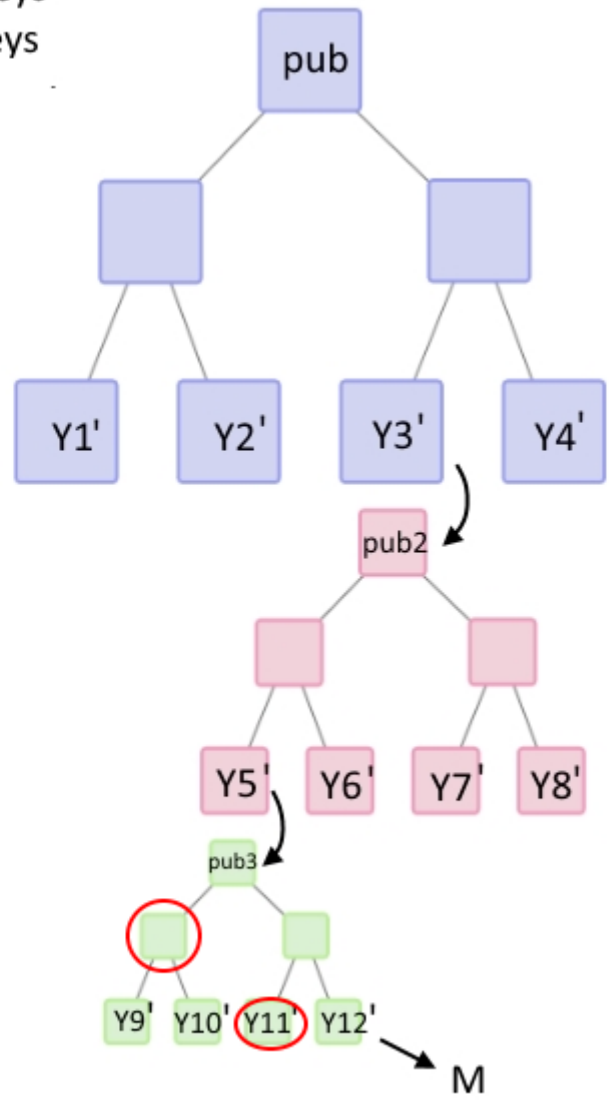
signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M

s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn ○



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

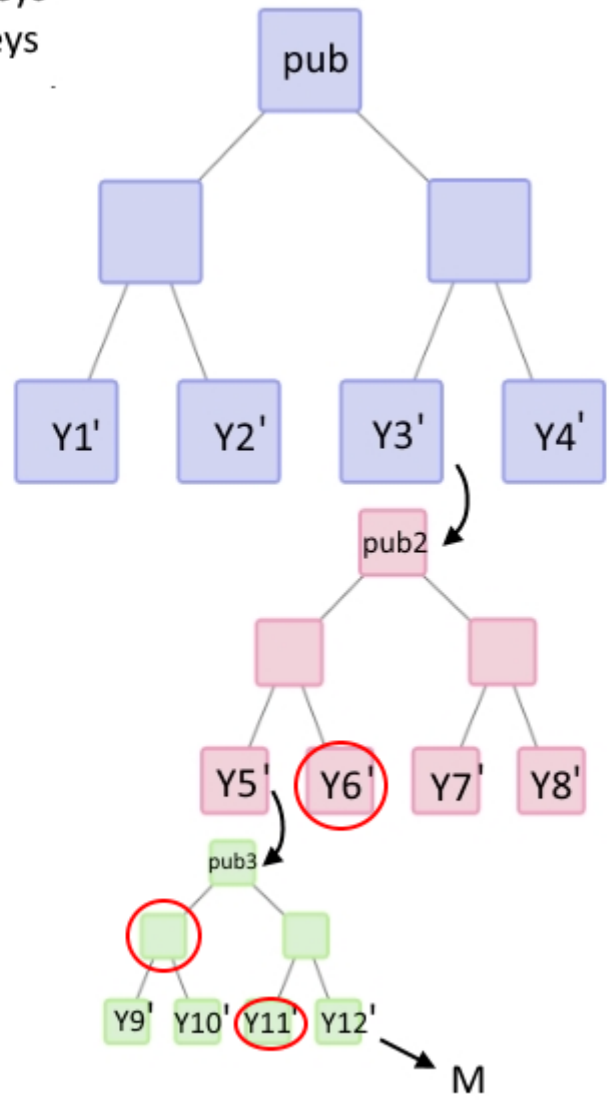
signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M

s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn ○



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

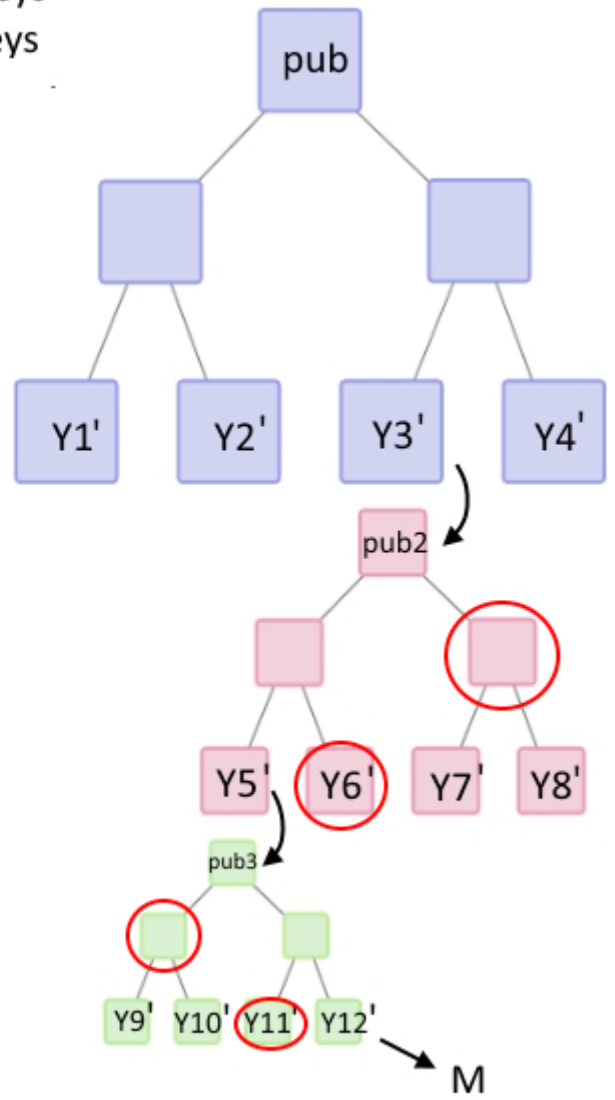
signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M

s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn ○



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

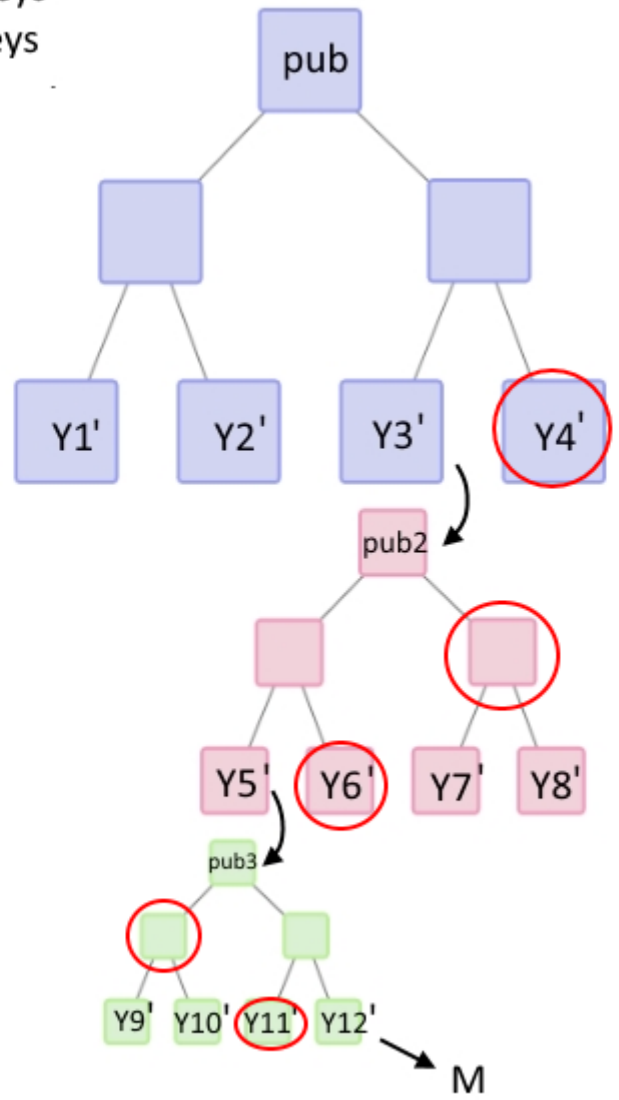
signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M

s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn ○



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

generiere Y5 bis Y12
(auch X5 bis Y12)

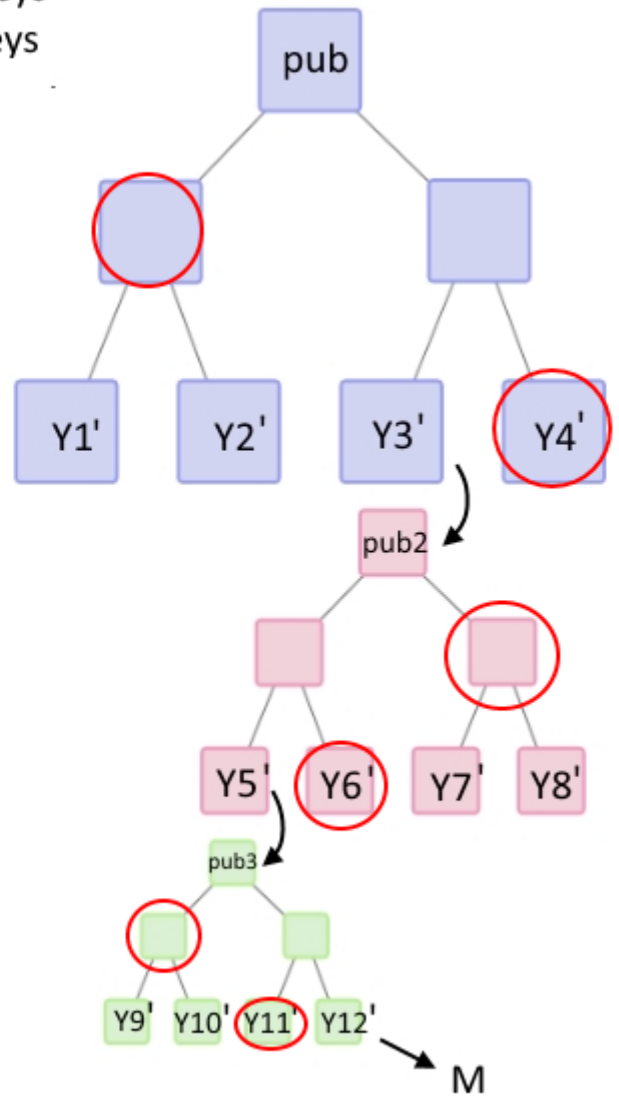
signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M

s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn ○



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

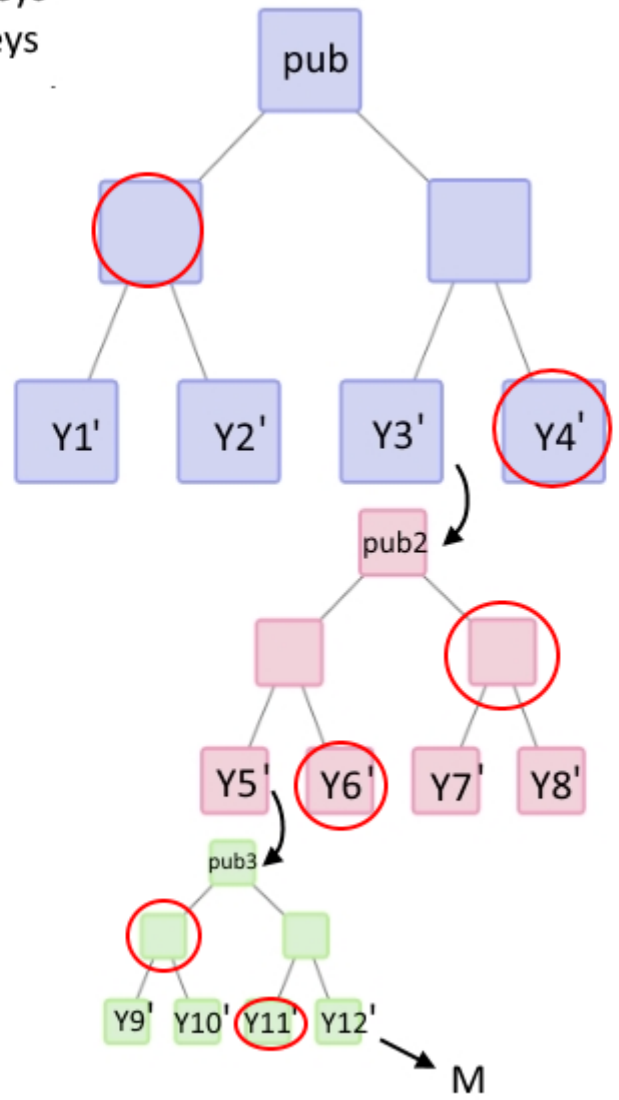
generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M
Signatur S
s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn



Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

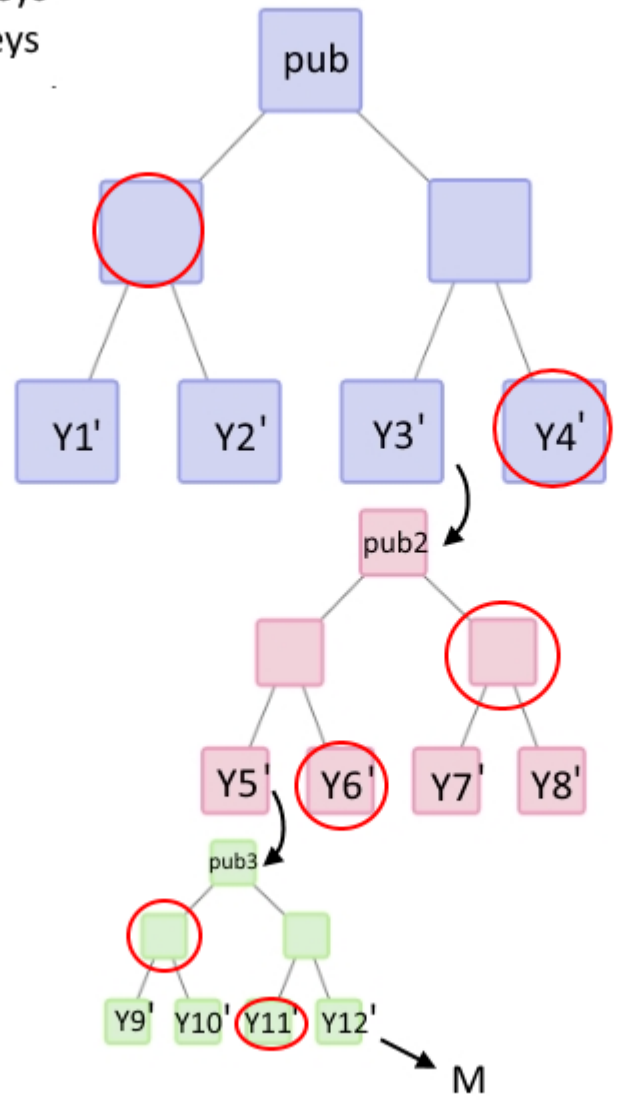
generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M
Signatur S
s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn



Verifikation

Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

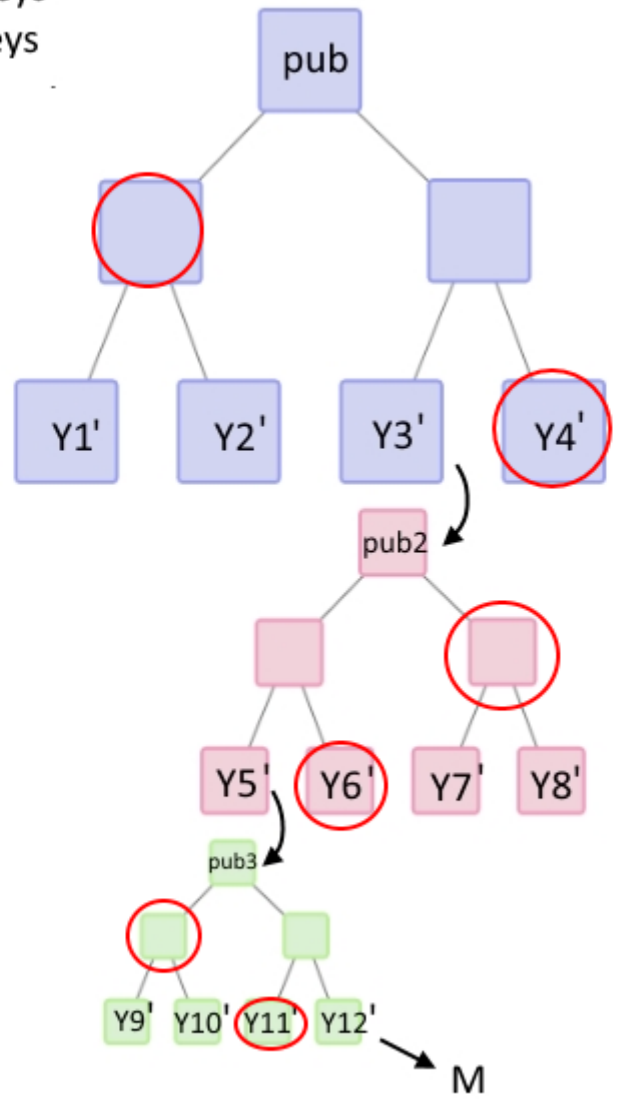
generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M
Signatur S
s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn



Verifikation

Signaturgröße und Performanz

Nehme an:
Baum mit 2^16 Blättern
pro Teilbaum und 10 Ebenen

Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

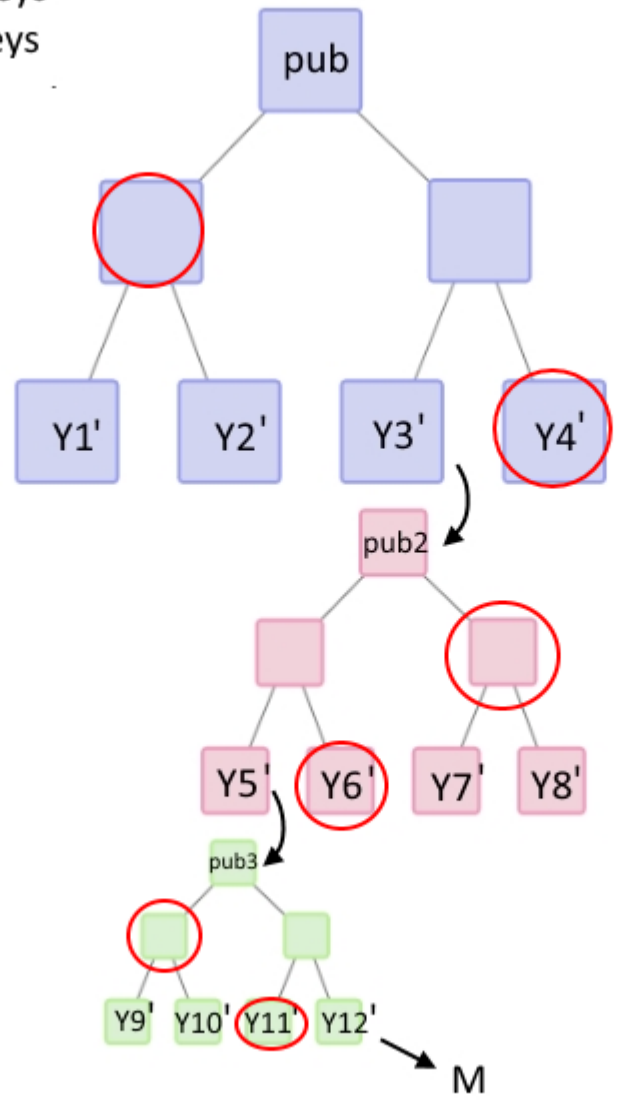
generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M
Signatur S
s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn



Verifikation

Signaturgröße und Performanz

Nehme an:
Baum mit 2^16 Blättern
pro Teilbaum und 10 Ebenen

==> 2^160 Blätter

Merkle-Signatur (Erweiterung)

Seien Y1 bis Y4 meine public keys
und X1 bis X4 meine private keys

Wähle Anzahl Ebenen: 3

Signieren

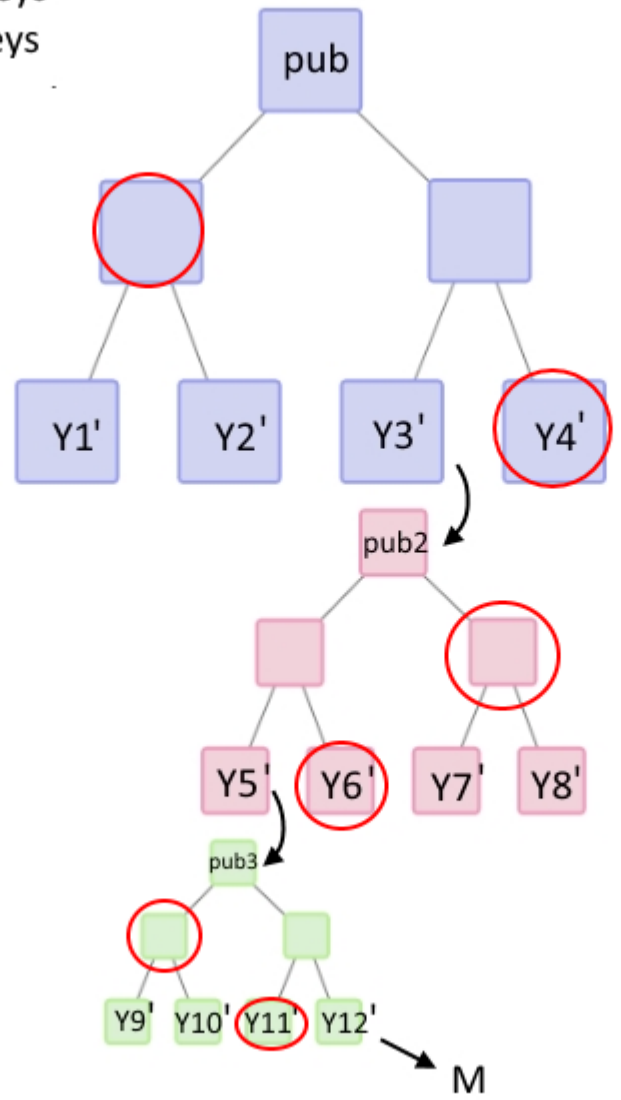
generiere Y5 bis Y12
(auch X5 bis Y12)

signiere pub2 mit X3
==> Signatur s1

signiere pub3 mit X5
==> Signatur s2

signiere Message M mit X12
==> Signatur s3

Sende an Empfänger:
Message M
Signatur S
s1, s2, s3
Y3, Y5, Y12
Pfadnachbarn



Verifikation

Signaturgröße und Performanz

Nehme an:
Baum mit 2^16 Blättern
pro Teilbaum und 10 Ebenen

==> 2^160 Blätter

Signaturgröße: 34 KB
Signierungsdauer: 0,8 s

Fazit

Nur bedingt einsetzbar!

z.B. nicht für PGP-Messages, aber für Software-Updates

Quellen (1 von 2)

- https://www.imperialviolet.org/2013/07/18/hashsig.html
- https://de.wikipedia.org/wiki/Post-Quanten-Kryptographie
- https://books.google.de/books?id=VB598lO47NAC&pg=PA38&lpg=PA38&dq=winternitz+ots&source=bl&ots=lwpag2Sq_0&sig=JlcOFgEpAhInTlcbDtv4WGW5Eg8&hl=de&sa=X&ei=FfSCVLG1OorNygP7yICQDw&ved=0CD8Q6AEwAw#v=onepage&q=winternitz%20ots&f=false
- http://imperia.rz.rub.de:9085/imperia/md/content/seminare/itsss08/becker.pdf
- https://de.wikipedia.org/wiki/Lamport-Diffie-Einmal-Signaturverfahren
- https://de.wikipedia.org/wiki/Digitale_Signatur

Quellen (2 von 2)

Bildquellen:

- http://smilys.net/riesige_smilies/smiley5102.gif
- http://www.basicthinking.de/blog/wp-content/uploads/2013/10/quantencomputer.jpg
- http://aktuell.ruhr-uni-bochum.de/mam/images/fittosize_440_0_85987c4ab02b85b473adeb1a0f4ba522_russland-flagge.jpeg
- http://www.unabhaengige-tester.de/wp-content/uploads/2014/07/china-flagge.jpg
- http://www.kunstkopie.de/kunst/juergen_priewe/israelische_flagge_hi.jpg
- http://www.extremetech.com/wp-content/uploads/2013/02/CameraDrone.jpg
- http://3.bp.blogspot.com/_WWUzaO9pYcA/SRVp_QxqpqI/AAAAAAAAJ3E/D8Guk3xvnk4/s400/03465.jpg
- http://bilder.augsburger-allgemeine.de/img/friedberg/crop24866211/2177215389-ctopTeaser/15441583.jpg
- http://www.inc.com/uploaded_files/image/970x450/keys_29082.jpg
- http://www.goldenkeyresources.com/Golden_Key_Resources/Golden_Key_Story_files/GoldenKey.jpg
- https://upload.wikimedia.org/wikipedia/commons/thumb/9/90/MerkleTree1.svg/800px-MerkleTree1.svg.png
- https://upload.wikimedia.org/wikipedia/commons/thumb/2/2e/MerkleTree2.svg/800px-MerkleTree2.svg.png
- https://www.imperialviolet.org/binary/hashsig-forest.svg

THE END