# Ring Signatures for Deniable AKEM: Gandalf's Fellowship

**Phillip Gajland**[1,2,3] & **Jonas Janneck**[2] & **Eike Kiltz**[2]

[1] Max Planck Institute for Security and Privacy

[2] Ruhr University Bochum

[3] IBM Research Europe – Zurich

CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, USA.





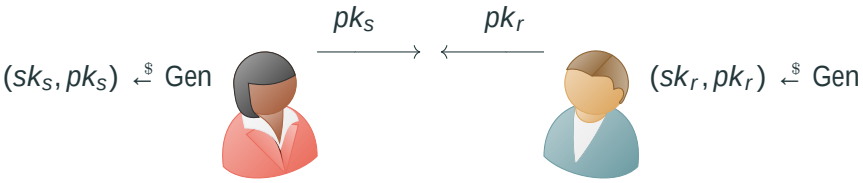
# authenticated key encapsulation mechanism (AKEM) [ABH+21]

# authenticated key encapsulation mechanism (AKEM) [ABH+21]

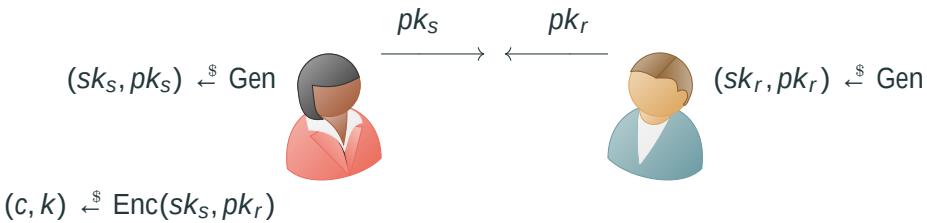
$(sk_S, pk_S) \xleftarrow{\$} \mathrm{Gen}$

$(sk_r, pk_r) \xleftarrow{\$} \mathrm{Gen}$

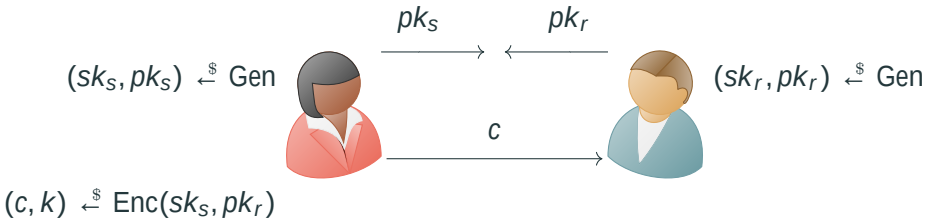
# authenticated key encapsulation mechanism (AKEM) [ABH+21]



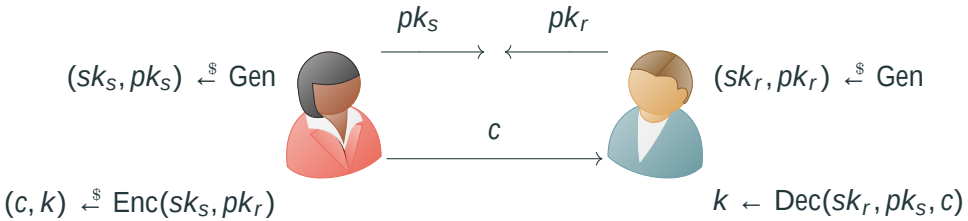
# authenticated key encapsulation mechanism (AKEM) [ABH+21]



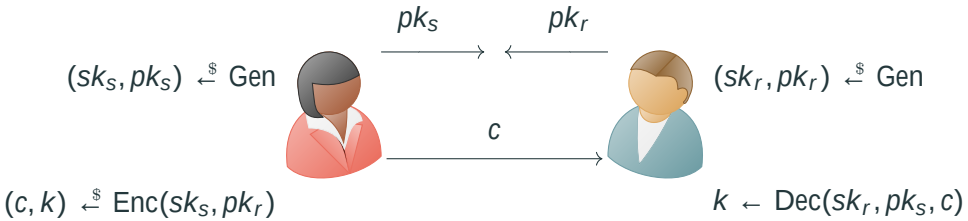
# authenticated key encapsulation mechanism (AKEM) [ABH+21]



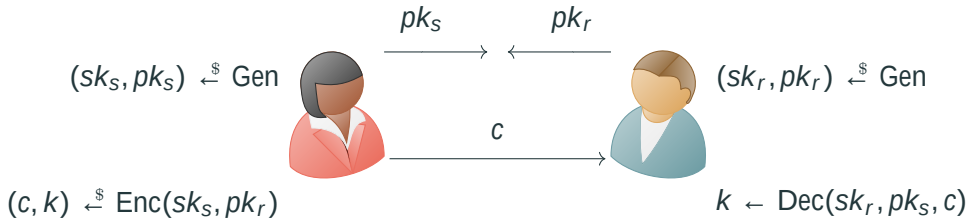
# authenticated key encapsulation mechanism (AKEM) [ABH+21]



# authenticated key encapsulation mechanism (AKEM) [ABH+21]

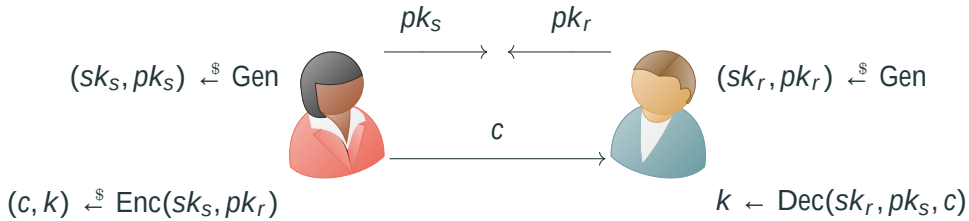


# authenticated key encapsulation mechanism (AKEM) [ABH+21]



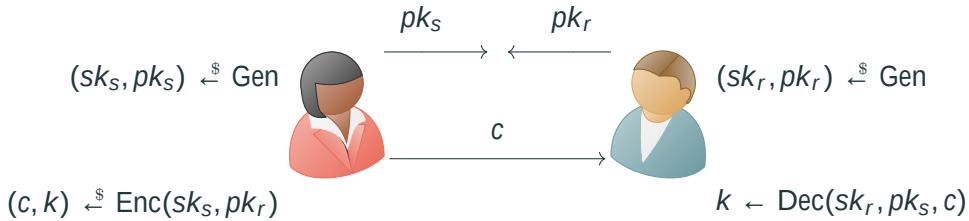

- The primitive behind HPKE [BBLW22] used in MLS [BBR+23]

# authenticated key encapsulation mechanism (AKEM) [ABH+21]



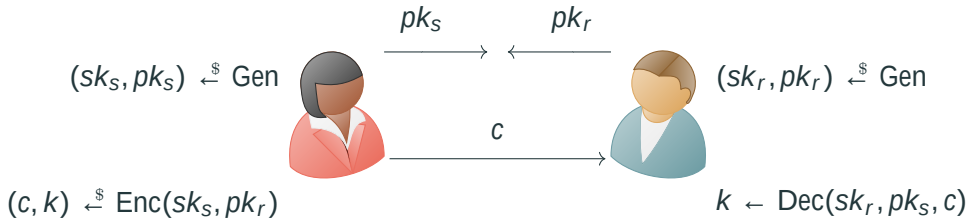

- The primitive behind HPKE [BBLW22] used in MLS [BBR+23]
- Confidentiality: Use CRYSTALS-Kyber [SAB+22]

# authenticated key encapsulation mechanism (AKEM) [ABH+21]



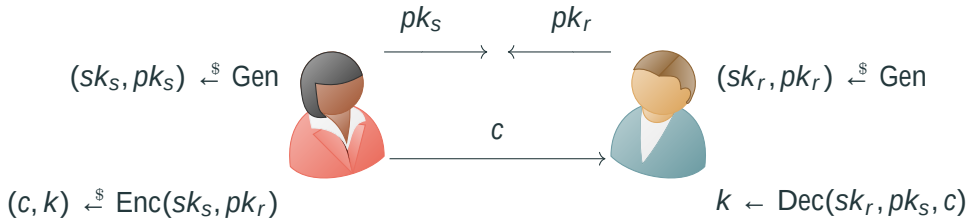

- The primitive behind HPKE [BBLW22] used in MLS [BBR+23]
- Confidentiality: Use CRYSTALS-Kyber [SAB+22]
- Authenticity: Use CRYSTALS-Dilithium [LDK+22]

# authenticated key encapsulation mechanism (AKEM) [ABH+21]




- The primitive behind HPKE [BBLW22] used in MLS [BBR+23]
- Confidentiality: Use CRYSTALS-Kyber [SAB+22]
- Authenticity: Use CRYSTALS-Dilithium [LDK+22]
- Deniability: ?

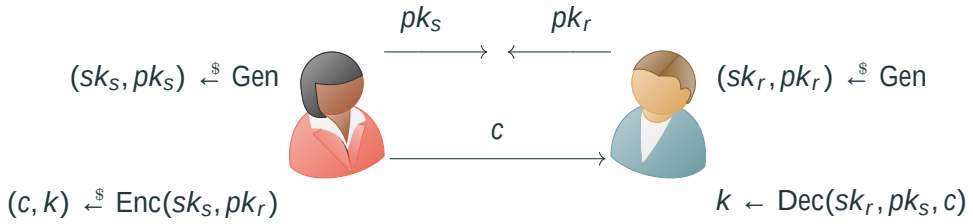
# authenticated key encapsulation mechanism (AKEM) [ABH+21]




- The primitive behind HPKE [BBLW22] used in MLS [BBR+23]
- Confidentiality: Use CRYSTALS-Kyber [SAB+22]
- Authenticity: Use CRYSTALS-Dilithium [LDK+22]
- Deniability: 7

} 3.5 KB

# authenticated key encapsulation mechanism (AKEM) [ABH+21]




- The primitive behind HPKE [BBLW22] used in MLS [BBR+23]
- Confidentiality: Use ~~CRYSTALS-Kyber [SAB+22]~~
- Authenticity: Use ~~CRYSTALS-Dilithium [LDK+22]~~
- Deniability: ~~7~~ 3

~~3.5 KB~~ 2 KB

This work

# outline

# outline

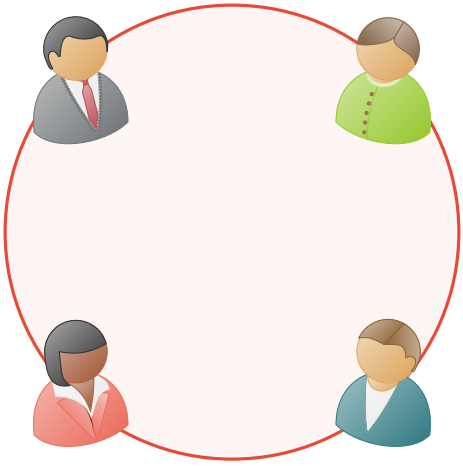
- Ring Signatures
  - Applications and Trade-offs
  - Gandalf Construction and Proof
  - Comparison

# outline

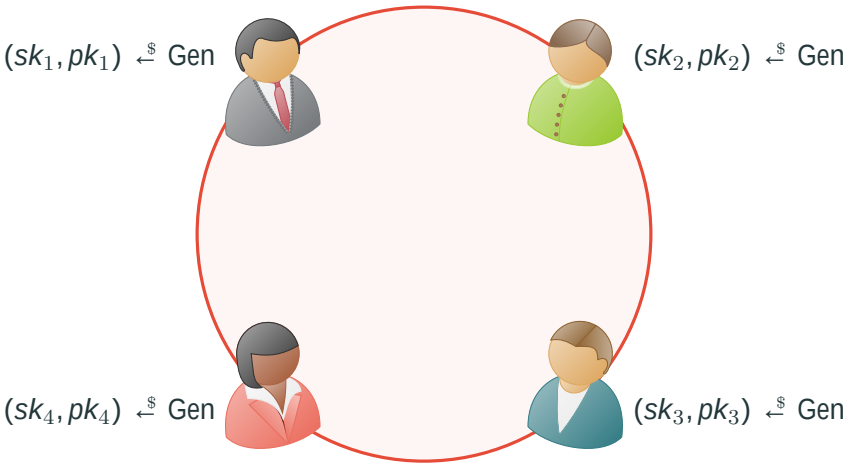
- Ring Signatures
  - Applications and Trade-offs
  - Gandalf Construction and Proof
  - Comparison
- Authenticated Key Encapsulation Mechanisms
  - Deniability
  - Black-box Construction and Security
  - Comparison

# ring signatures

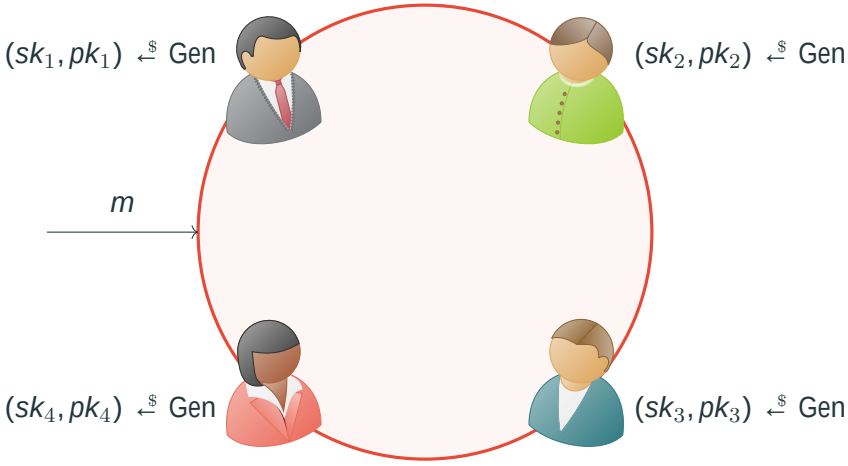
# ring signature scheme (RSig) [RST01]



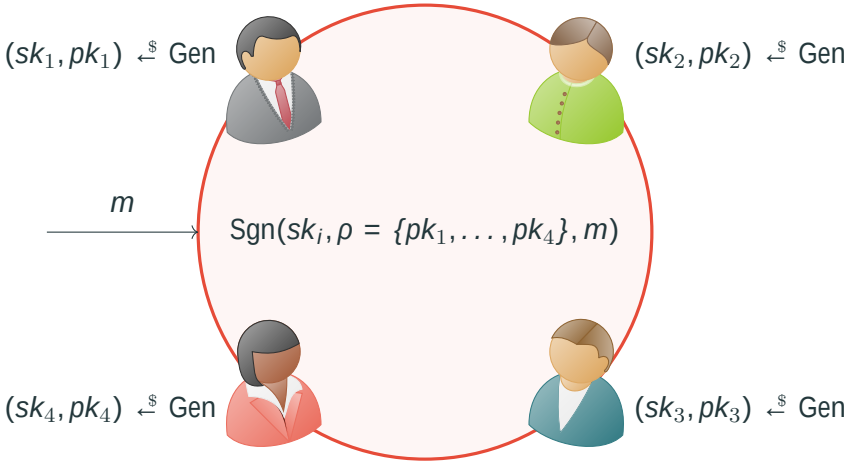
# ring signature scheme (RSig) [RST01]



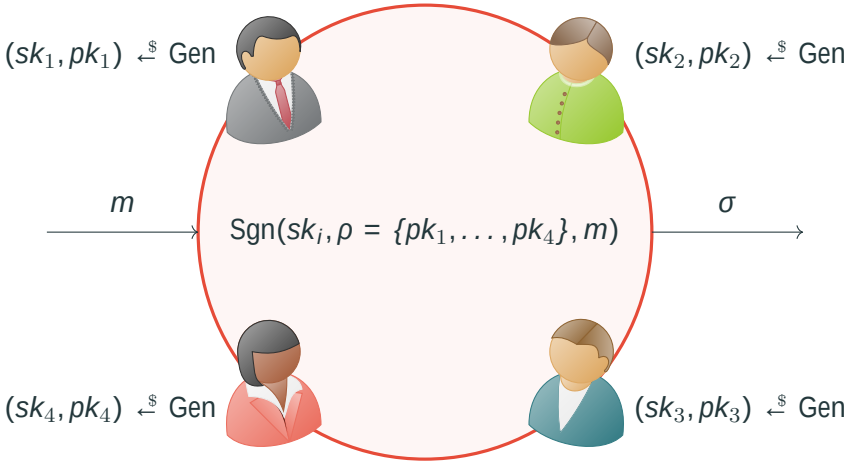
# ring signature scheme (RSig) [RST01]



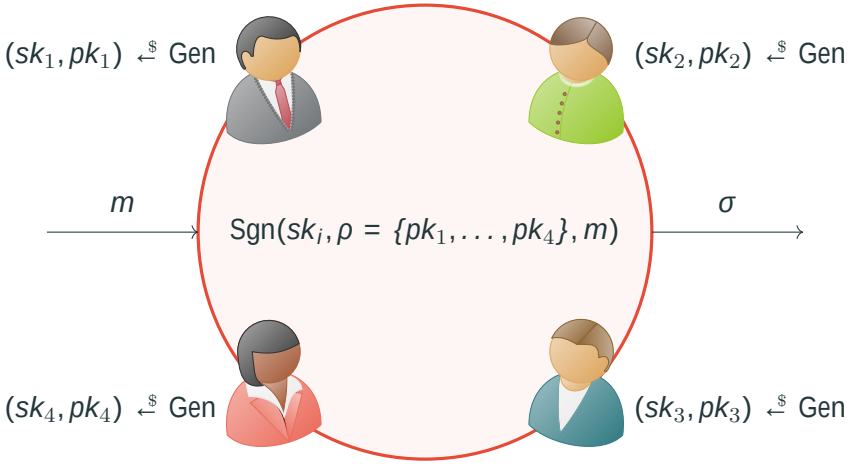
# ring signature scheme (RSig) [RST01]



# ring signature scheme (RSig) [RST01]

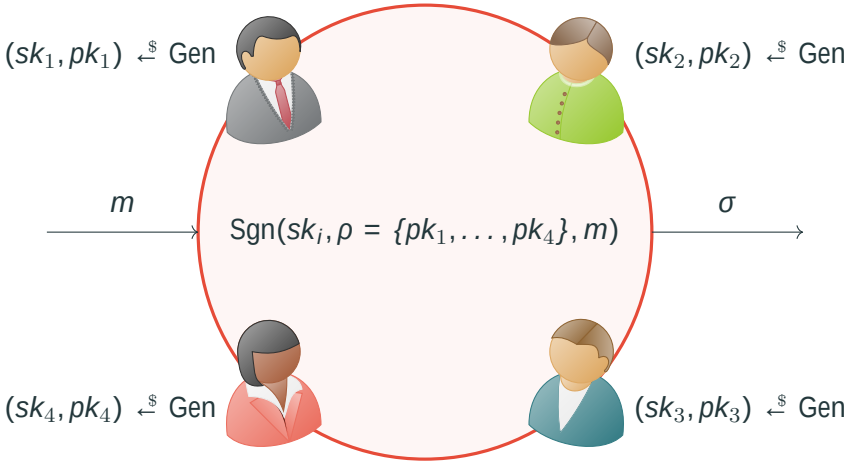


# ring signature scheme (RSig) [RST01]




- **Unforgeability**: One $sk_i$ is needed to generate $\sigma$ .

# ring signature scheme (RSig) [RST01]

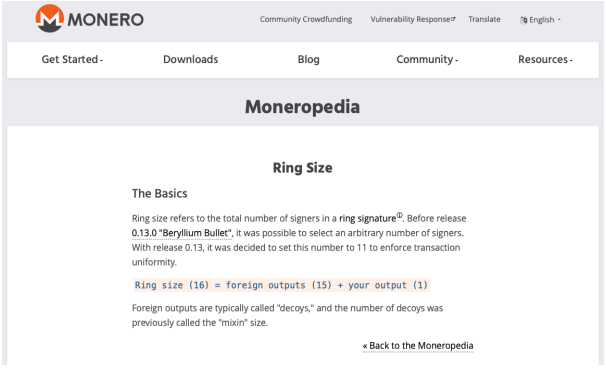



- **Unforgeability**: One $sk_i$ is needed to generate $\sigma$ .
- **Anonymity**: Given $\sigma$ and $\rho$, it is not possible to identify who signed.

# RSig schemes are deployed

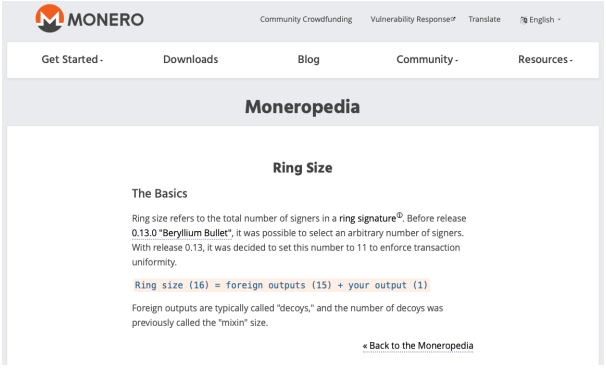
- Originally introduced as a mechanism to protect whistle-blowers.
- Currently used in voting systems and cryptocurrencies such as Dash & Monero.
- Deniable Authentication

# RSig schemes are deployed



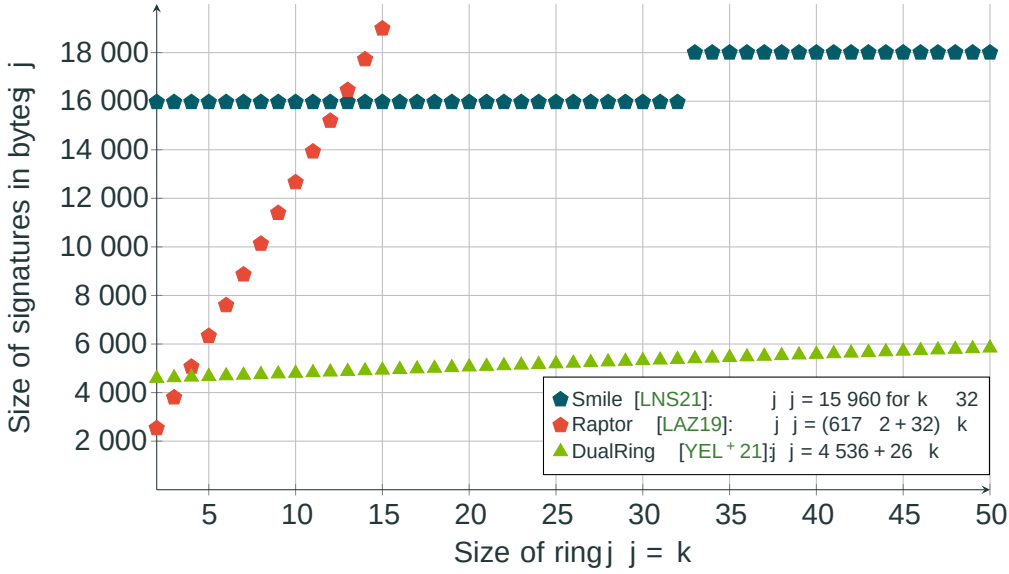

- Originally introduced as a mechanism to protect whistle-blowers.
- Currently used in voting systems and cryptocurrencies such as Dash & Monero.
- Deniable Authentication

# RSig schemes are deployed

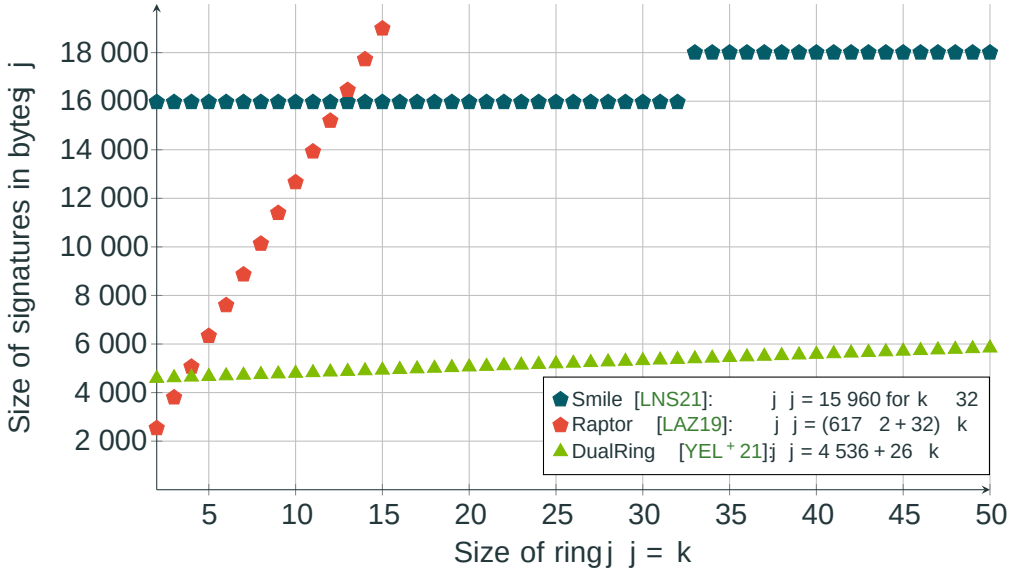



- Originally introduced as a mechanism to protect whistle-blowers.
- Currently used in voting systems and cryptocurrencies such as Dash & Monero.
- Deniable Authentication

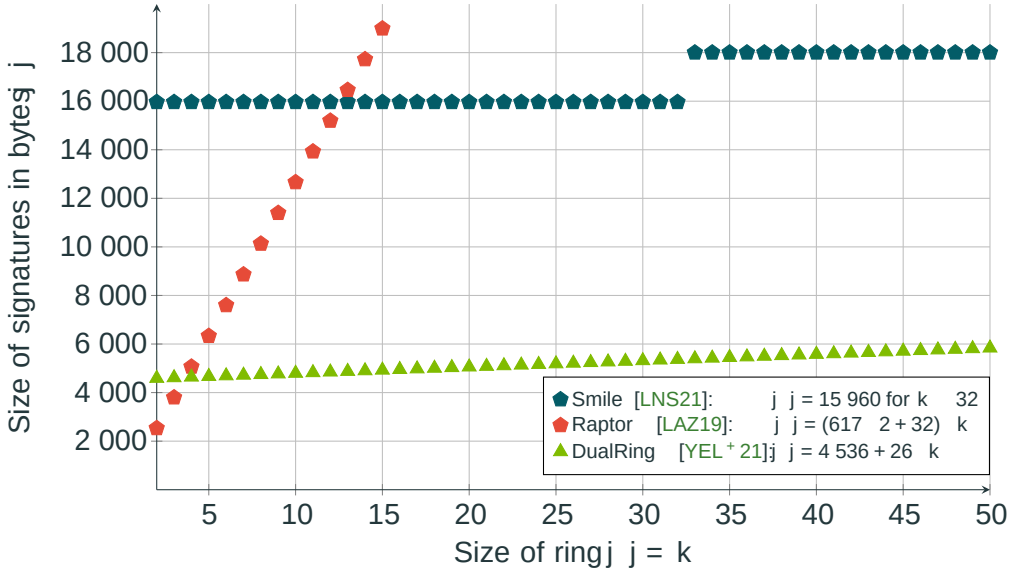
# RSig schemes: linear vs sub-linear



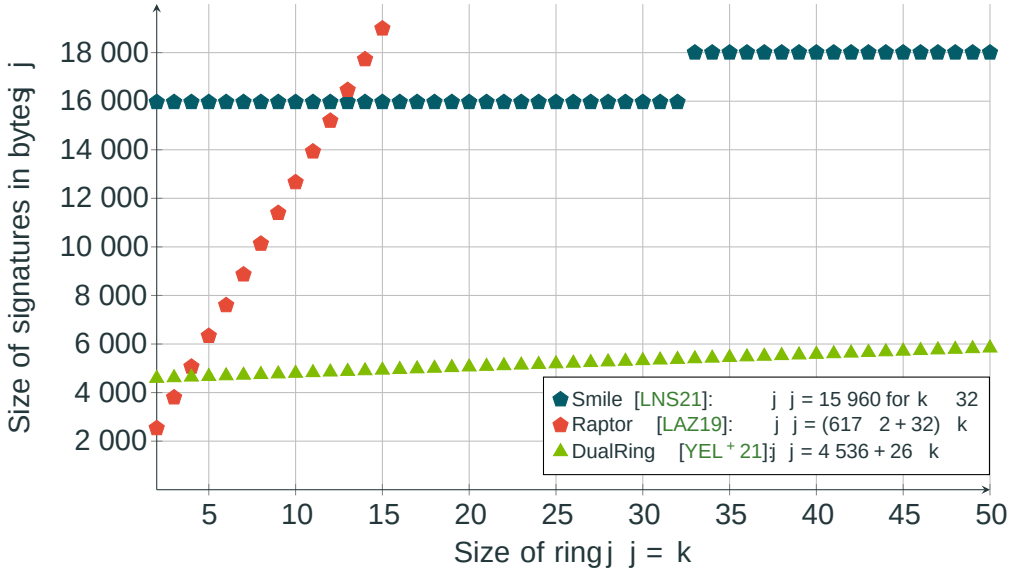
# RSig schemes: linear vs sub-linear



# RSig schemes: linear vs sub-linear



# RSig schemes: linear vs sub-linear



# warmup: a linear RSig construction

$$f_h : R_q \; R_{q} \; ! \; R_{q}$$
$$(u, v) \; 7! \; h \quad u + v$$

# warmup: a linear RSig construction

$$f_h : R_q \; R_q \to R_q$$
$$(u, v) \mapsto h \; u + v$$

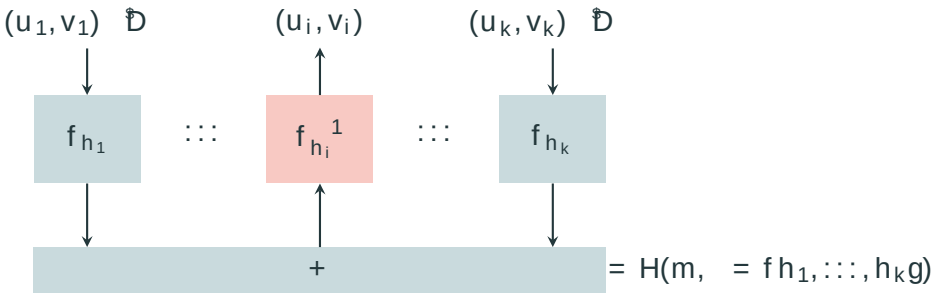
$$f_h^{-1} : c \mapsto (u, v) \; D_{jh}^{2} \; u + v = c$$

# warmup: a linear RSig construction

$f_h : R_q \; R_q \to R_q$

$(u, v) \mapsto h \; u + v$

$f_h^{-1} : c \mapsto \{(u, v) \in D^2 \mid jh \; u + v = c\}$

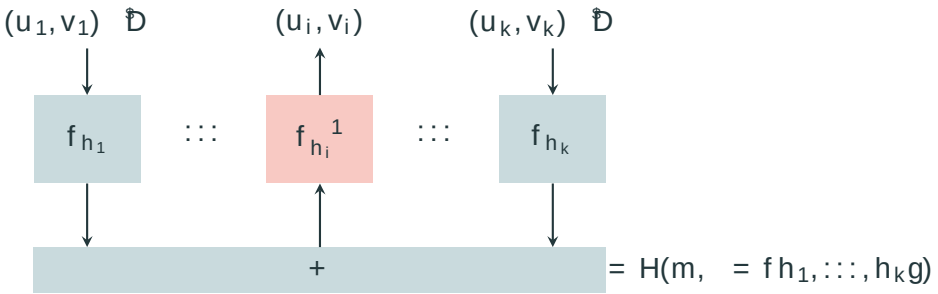


# warmup: a linear RSig construction

$f_h : R_q \; R_q \to R_q$

$(u, v) \mapsto h \; u + v$

$f_h^{-1} : c \mapsto \{(u, v) \in D^2 \mid jh \; u + v = c\}$

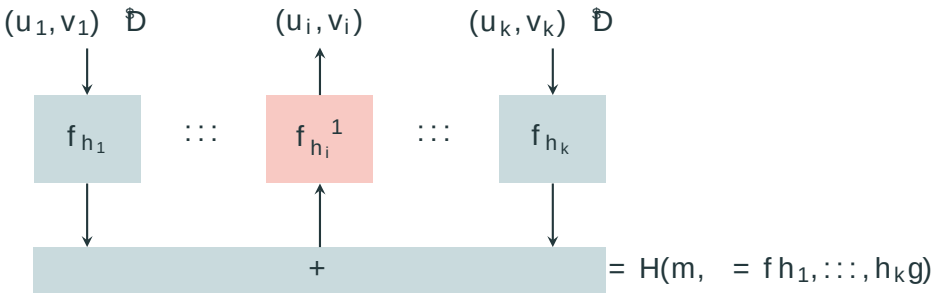


# warmup: a linear RSig construction

$f_h : R_q \; R_q \to R_q$

$(u, v) \mapsto h \; u + v$

$f_h^{-1} : c \mapsto \{(u, v) \in D^2 \mid jh \; u + v = c\}$

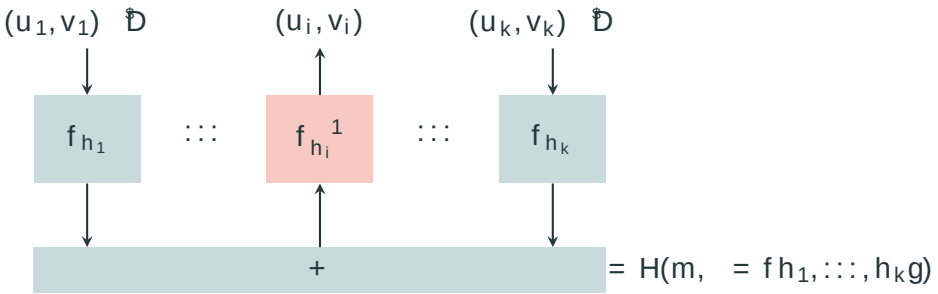


# warmup: a linear RSig construction

$f_h : R_q \; R_q \to R_q$

$(u, v) \mapsto h \; u + v$

$f_h^{-1} : c \mapsto \{(u, v) \in D^2 \mid jh \; u + v = c\}$

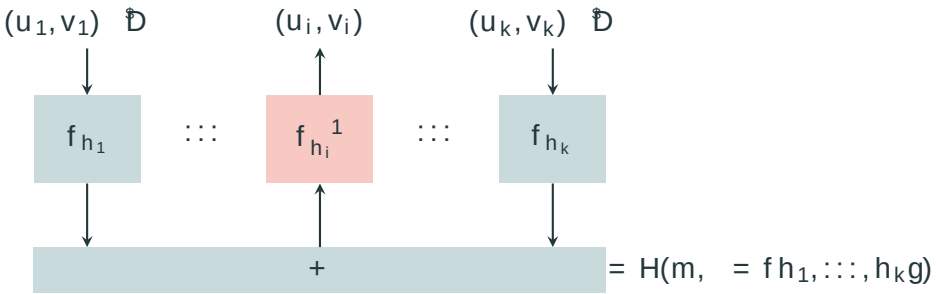


# warmup: a linear RSig construction

$f_h : R_q \; R_q \to R_q$

$(u, v) \mapsto h \; u + v$

$f_h^{-1} : c \mapsto \{(u, v) \in D^2 \mid jh \; u + v = c\}$

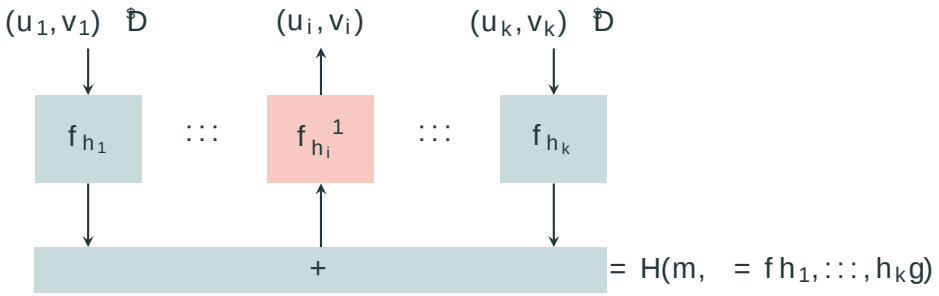


# warmup: a linear RSig construction

$f_h : R_q \; R_q \to R_q$

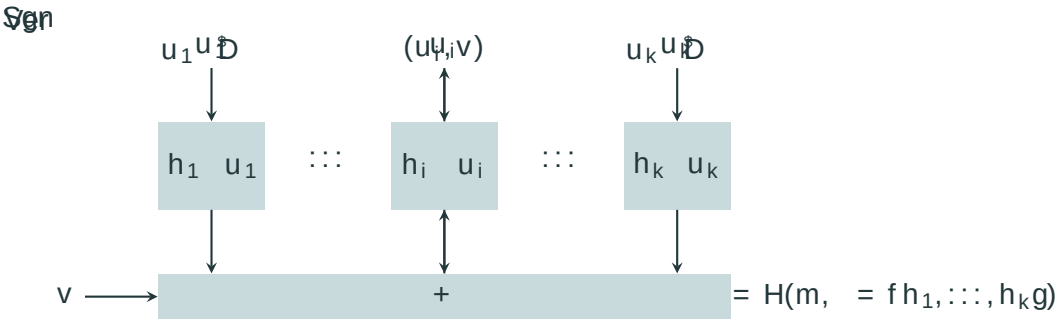
$(u, v) \mapsto h \; u + v$

$f_h^{-1} : c \mapsto (u, v) \; D \; \sum_{jh \; u+v=c}^{2}$




Output = $(u_1, v_1 ::: , u_k, v_k)$ such that $8 \; i \; 2 \; [k] : k(u_i, v_i)k_2$

# RSig construction: Gandalf



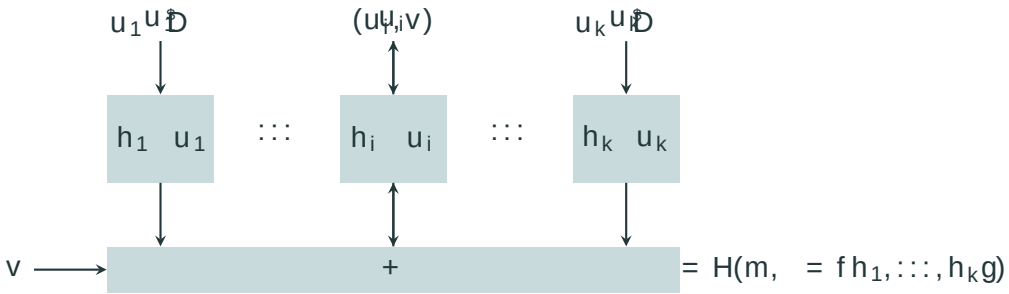
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign



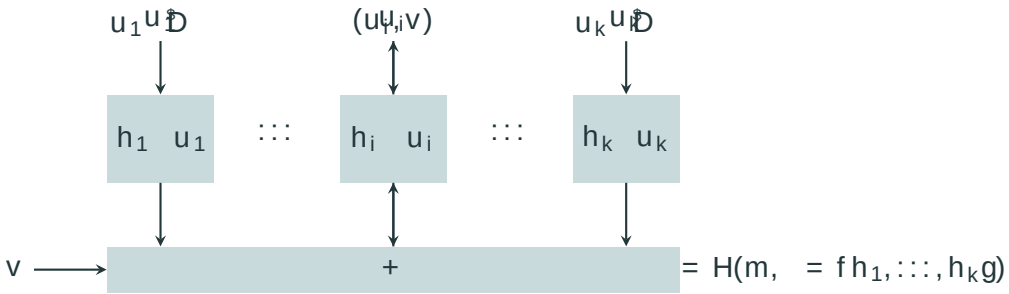
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign



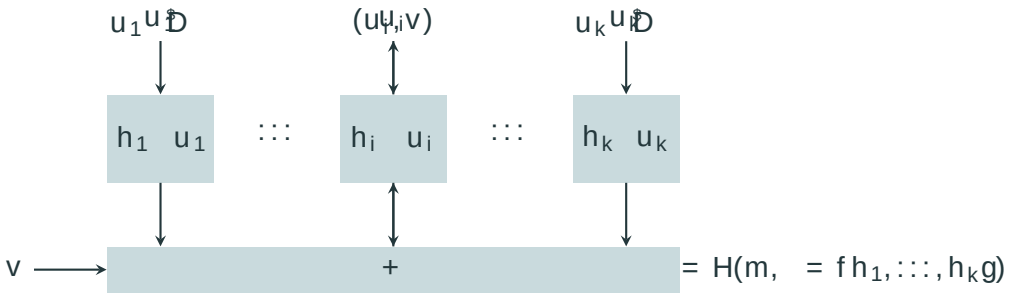
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign



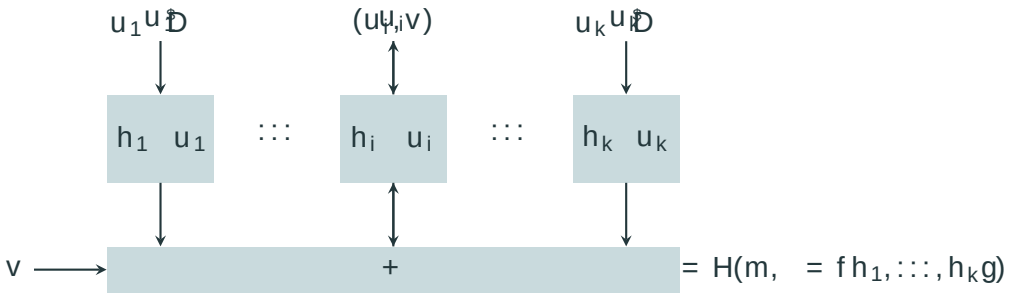
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign



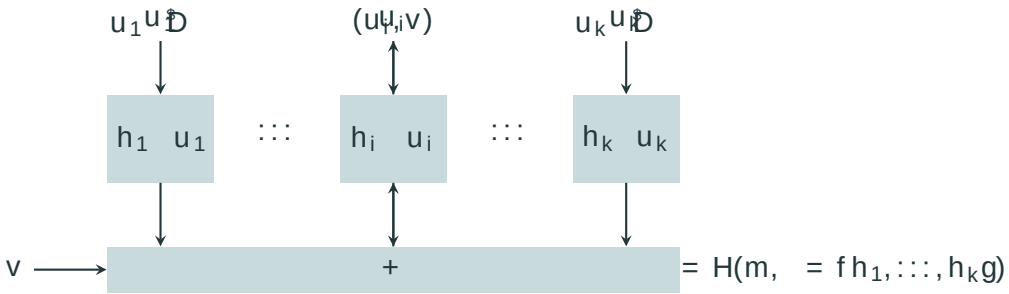
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign



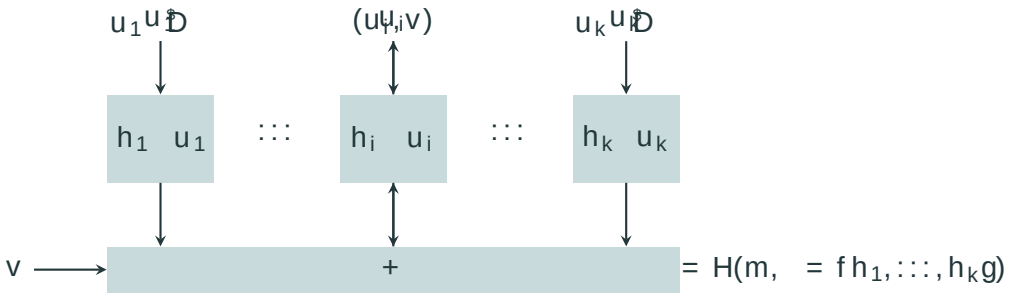
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign

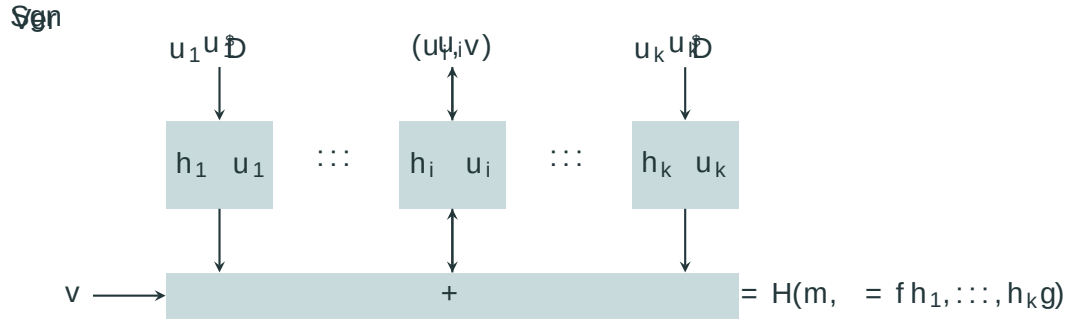


# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) \; 7! \; h \; u + v$

$f_h^{\,1} : c \; 7! \; (u, v) \; D \; {}^{2}_{jh \; u+v=c}$




Output $= (u_1, :::, u_k) \; 2 \; R_q^k$

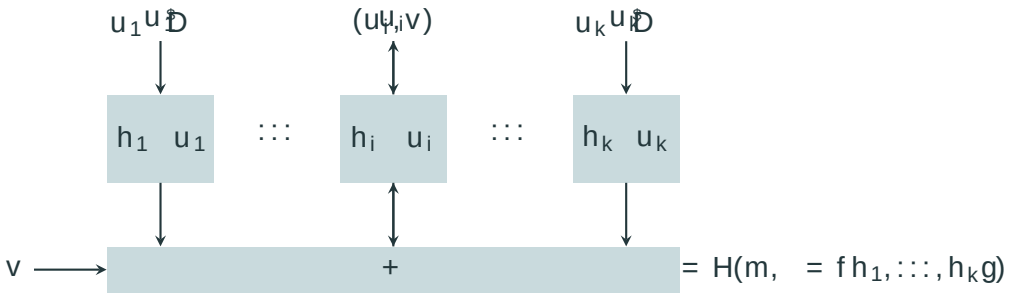
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign



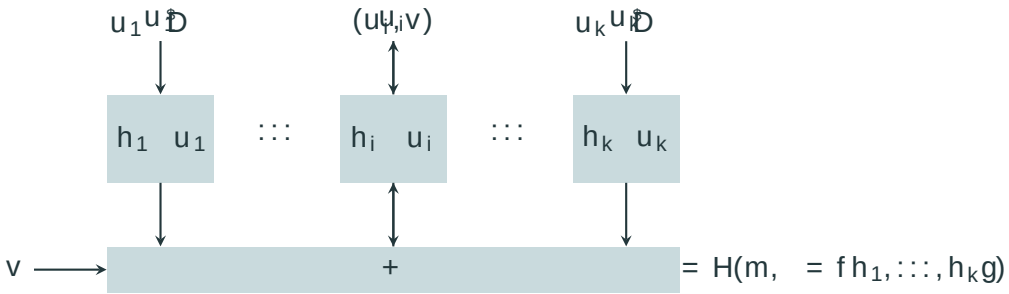
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign



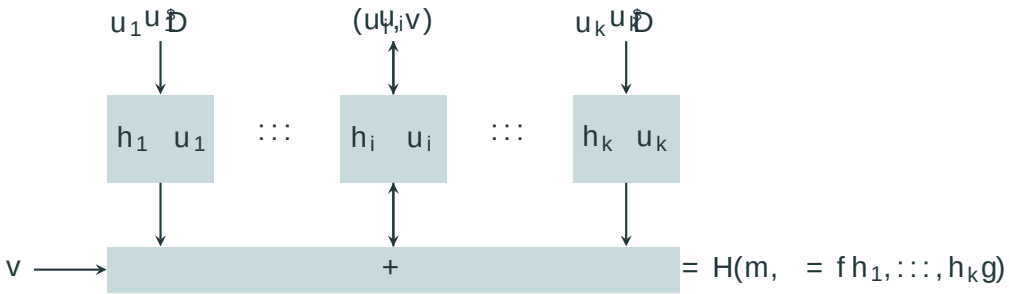
# RSig construction: Gandalf

$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

$f_h^{\,1} : c 7! (u, v) \quad D_{jh \; u+v=c}^{2}$

Sign

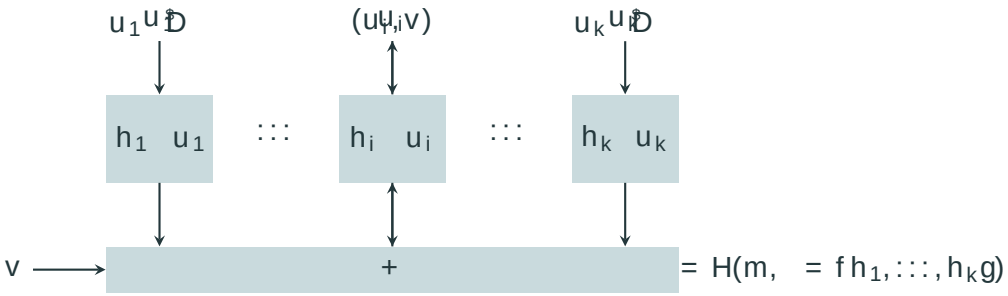


$f_h : R_q \; R_q ! R_q$

$(u, v) 7! h \quad u + v$

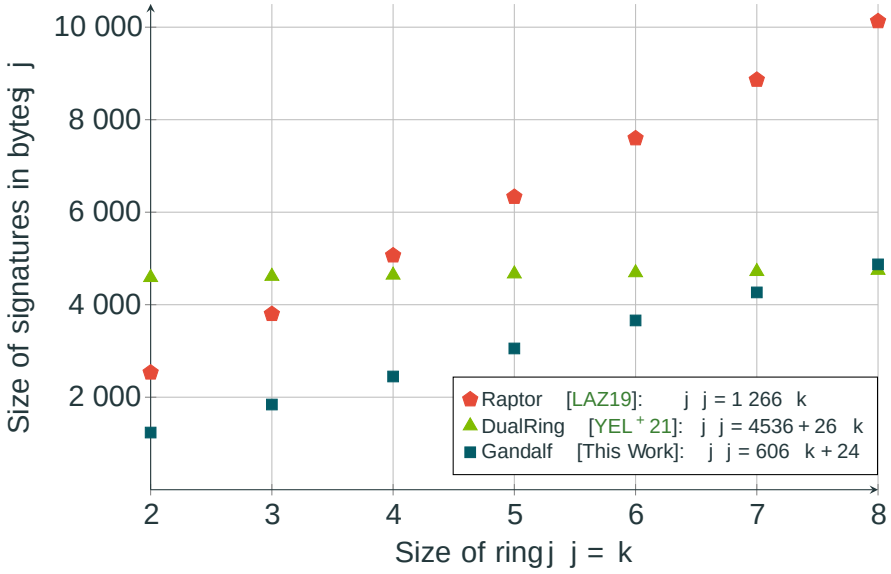
$f_h^{\,1} : c 7! (u, v) \quad D^{\,2}_{jh \; u+v=c}$

Sign

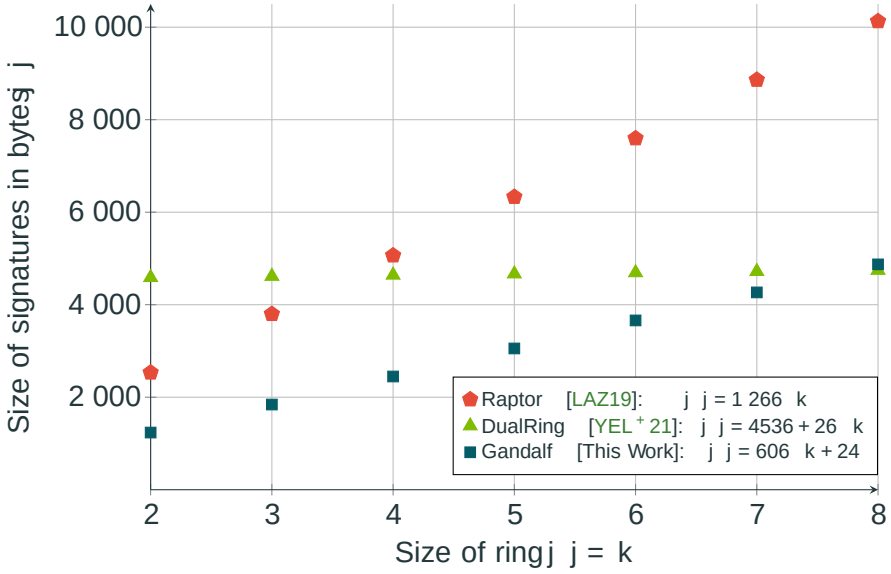



Check that: $k(u_1, :::, u_k, v)k_2$

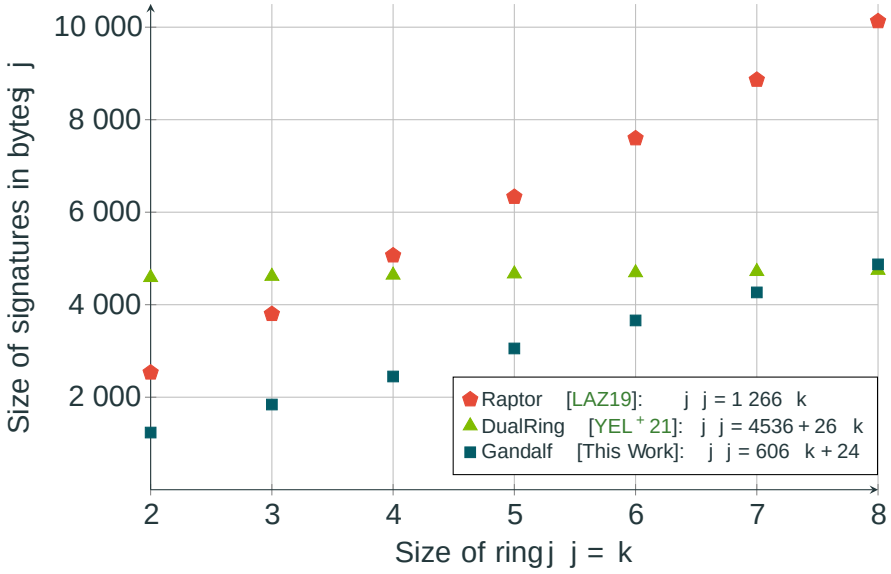
# Gandalf: comparison to other linear RSig schemes



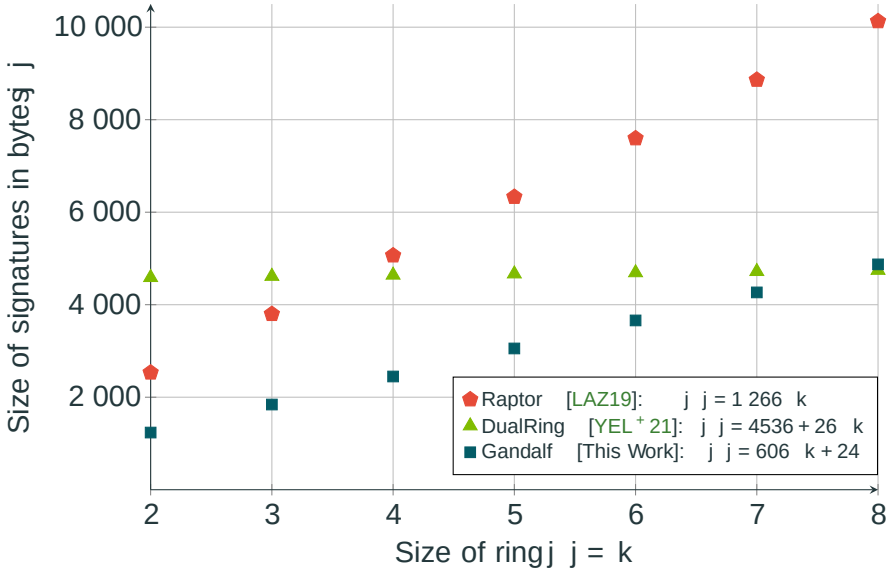
# Gandalf: comparison to other linear RSig schemes



# Gandalf: comparison to other linear RSig schemes



# Gandalf: comparison to other linear RSig schemes



# proof

# proof

# Gandalf security bounds

# Gandalf security bounds

- Unforgeability: One-per-message unforgeability under chosen ring attacks

# Gandalf security bounds

I Unforgeability: One-per-message unforgeability under chosen ring attacks

$$\mathrm{Adv}^{(n,\ ,Q_{Sgn})\text{-}\mathrm{UF\text{-}CRA1}}_{\mathrm{Gandalf}\ [\mathrm{TpdGen},\mathrm{PreSmp}]} \quad Q_H \quad \mathrm{Adv}^{R\text{-}\mathrm{LWE}}_{m=1,\ q,\ ,s}$$

# Gandalf security bounds

I Unforgeability: One-per-message unforgeability under chosen ring attacks

$$\mathrm{Adv}^{(n,\ ,Q_{Sgn})\text{-UF-CRA1}}_{\text{Gandalf}\ [\text{TpdGen},\text{PreSmp}]} \quad Q_H\ \ \mathrm{Adv}^{R\text{-LWE}}_{m=1,\ q,\ ,s} + c\ \ Q_H\ \ \mathrm{Adv}^{R\text{-ISIS}}_{m=n,q,\ ,}$$

# Gandalf security bounds

I Unforgeability: One-per-message unforgeability under chosen ring attacks

$$\mathrm{Adv}^{(n,\ ,Q_{Sgn})\text{-UF-CRA1}}_{\text{Gandalf}\ [\text{TpdGen},\text{PreSmp}]} \quad Q_H \quad \mathrm{Adv}^{R\text{-LWE}}_{m=1,\ q,\ ,s} + c \quad Q_H \quad \mathrm{Adv}^{R\text{-ISIS}}_{m=n,q,\ ,}$$

$$\text{for} \quad = \quad s \sqrt{\frac{p}{(\quad +1)\,N}}$$

# Gandalf security bounds

I Unforgeability: One-per-message unforgeability under chosen ring attacks

$$\mathrm{Adv}^{(n,\ ,Q_{Sgn})\text{-UF-CRA1}}_{\text{Gandalf}\ [\mathrm{TpdGen},\mathrm{PreSmp}]} \quad Q_H \quad \mathrm{Adv}^{R\text{-LWE}}_{m=1,\ q,\ ,s} + c \quad Q_H \quad \mathrm{Adv}^{R\text{-ISIS}}_{m=n,q,\ ,}$$

$$\text{for} \quad = \quad s \sqrt{(\quad + 1)\, N} \quad \text{and} \quad c = \sqrt{2} \quad R_2 \quad (\mathrm{PreSmp}jj\ D)^{Q_{Sgn}}.$$

# Gandalf security bounds

I Unforgeability: One-per-message unforgeability under chosen ring attacks

$$\mathrm{Adv}^{(n,\ ,Q_{Sgn})\text{-UF-CRA1}}_{\text{Gandalf}\ [\mathrm{TpdGen},\mathrm{PreSmp}]} \quad Q_H \quad \mathrm{Adv}^{R\text{-LWE}}_{m=1,\ q,\ ,s} + c \quad Q_H \quad \mathrm{Adv}^{R\text{-ISIS}}_{m=n,q,\ ,} + \frac{c}{\lvert R_q \rvert},$$

$$\text{for} \quad = \quad s \sqrt{(\ +1)\,N} \ \text{and}\ c = \sqrt{2}\ R_2\ (\mathrm{PreSmp}\lVert\ D)^{Q_{Sgn}}.$$

# Gandalf security bounds

- Unforgeability: One-per-message unforgeability under chosen ring attacks

$$\mathrm{Adv}^{(n,\ ,Q_{Sgn})\text{-UF-CRA1}}_{\text{Gandalf}\ [\mathrm{TpdGen},\mathrm{PreSmp}]} \quad Q_H \ \mathrm{Adv}^{R\text{-LWE}}_{m=1,\ q,\ ,s} + c \ Q_H \ \mathrm{Adv}^{R\text{-ISIS}}_{m=n,q,\ ,} + \frac{c}{jR_q j},$$

$$\text{for} \quad = s\sqrt{(\ +1)N} \text{ and } c = \sqrt{2}\ R_2\ (\mathrm{PreSmp}jj\ D)^{Q_{Sgn}}.$$

- Anonymity: Under full key exposure and multiple challenges

# Gandalf security bounds

- Unforgeability: One-per-message unforgeability under chosen ring attacks

$$\mathrm{Adv}^{(n,\ ,Q_{Sgn})\text{-UF-CRA1}}_{\text{Gandalf}\ [\mathrm{TpdGen},\mathrm{PreSmp}]} \quad Q_H\ \ \mathrm{Adv}^{R\text{-LWE}}_{m=1,\ q,\ ,s} + c\ \ Q_H\ \ \mathrm{Adv}^{R\text{-ISIS}}_{m=n,q,\ ,} + \frac{c}{jR_q j},$$

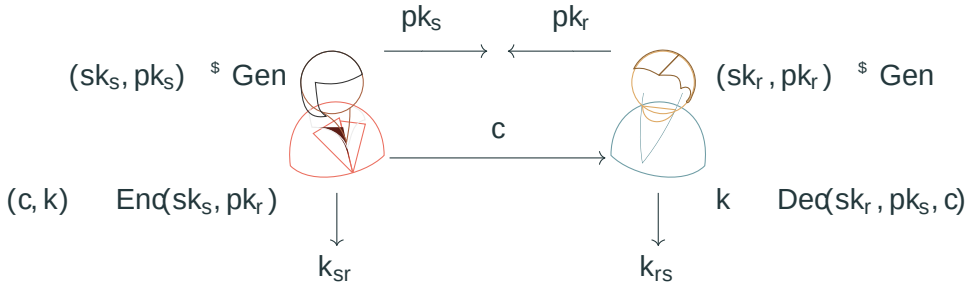
$$\text{for} \quad = s\sqrt{(\ +1)N} \text{ and } c = \sqrt{2}\ \ R_2\ \ (\mathrm{PreSmp}jj\ D)^{Q_{Sgn}}.$$

- Anonymity: Under full key exposure and multiple challenges

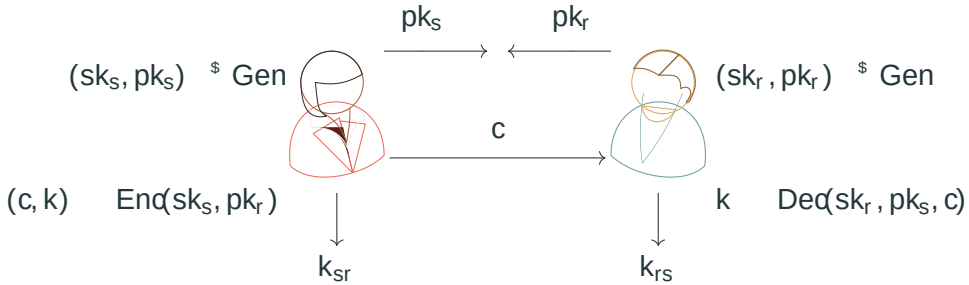
$$\mathrm{Adv}^{(n,Q_{Chl})\text{-MC-Ano}}_{\text{Gandalf}\ [\mathrm{TpdGen},\mathrm{PreSmp}]} \quad Q_{Chl}\ \ KL\ (\mathrm{PreSmp}jj\ D).$$

# akem

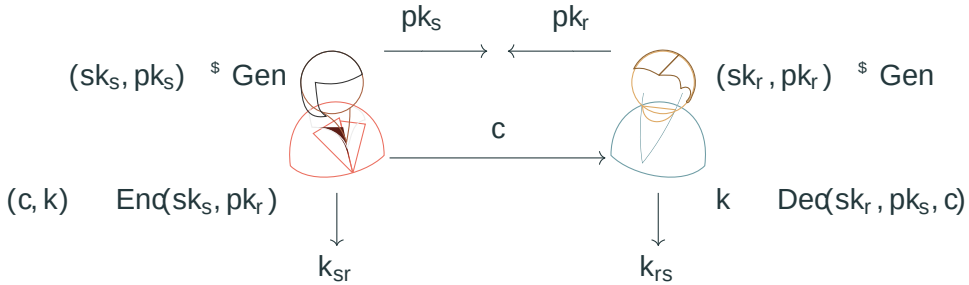
# authenticated key encapsulation mechanism ( AKEM) [ABH+21]



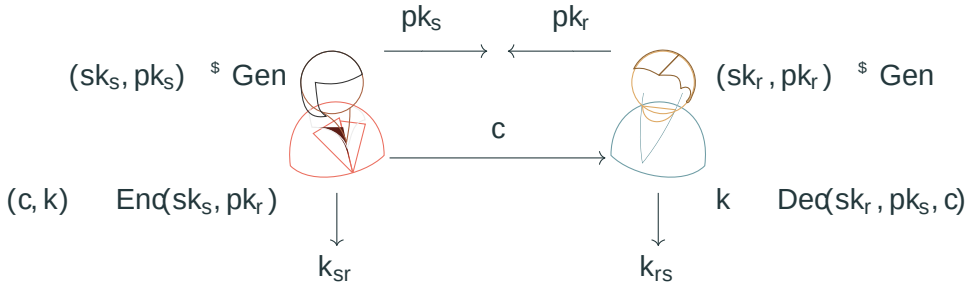
# authenticated key encapsulation mechanism ( AKEM) [ABH+21]



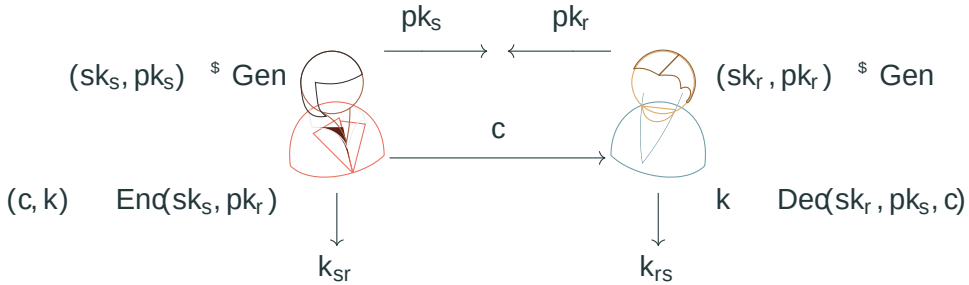
# authenticated key encapsulation mechanism ( AKEM) [ABH+21]



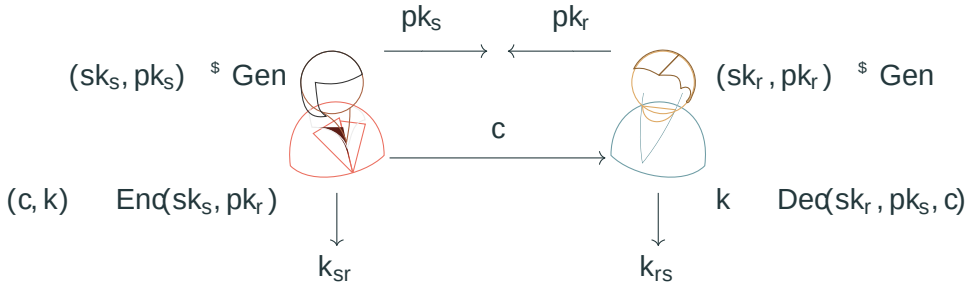
# authenticated key encapsulation mechanism ( AKEM) [ABH+21]



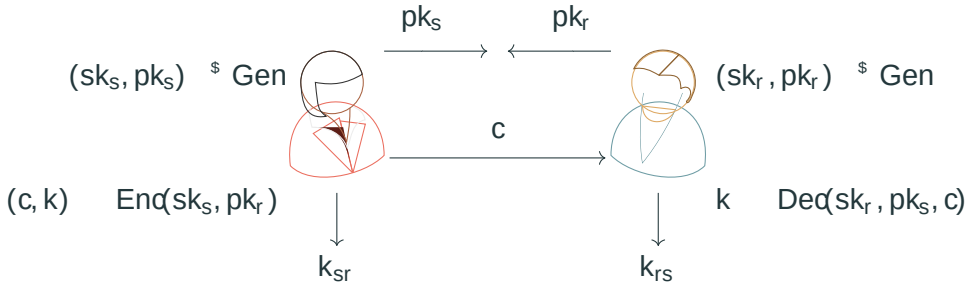
# authenticated key encapsulation mechanism ( AKEM) [ABH+21]



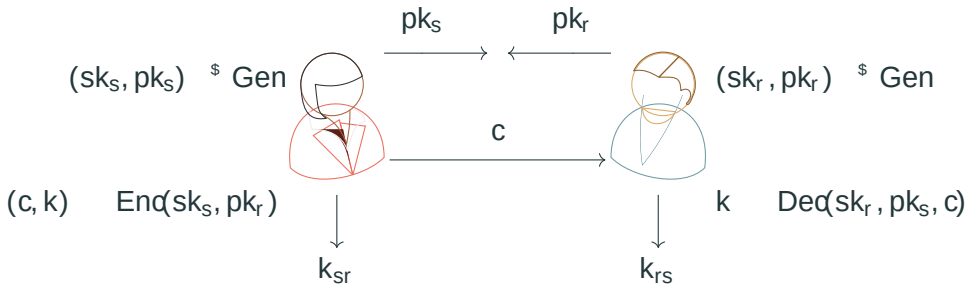
# authenticated key encapsulation mechanism ( AKEM) [ABH+21]



# authenticated key encapsulation mechanism ( AKEM) [ABH+21]

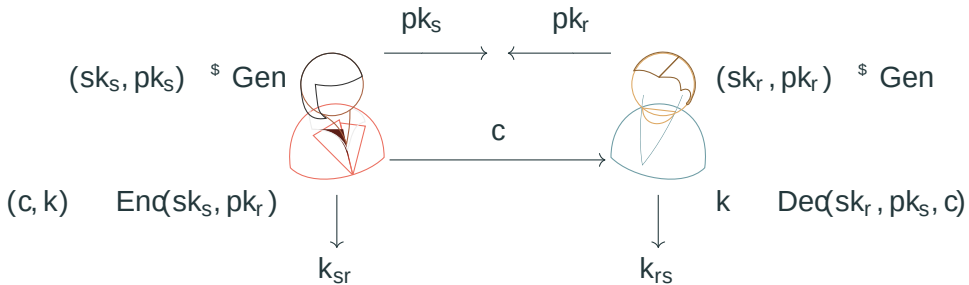


# authenticated key encapsulation mechanism ( AKEM) [ABH+21]



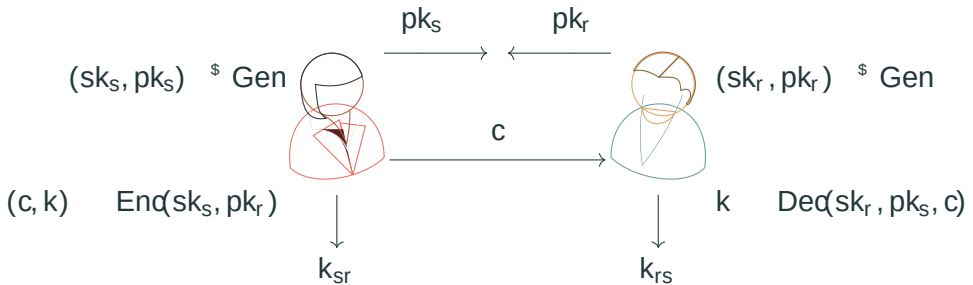

I Con dentiality: $k_{sr}$ and $k_{rs}$ should look random. 3

# authenticated key encapsulation mechanism ( AKEM) [ABH+21]

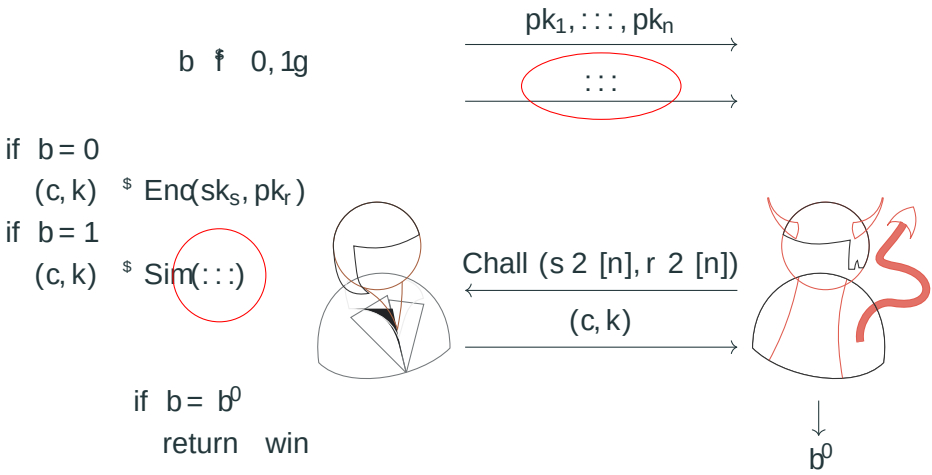


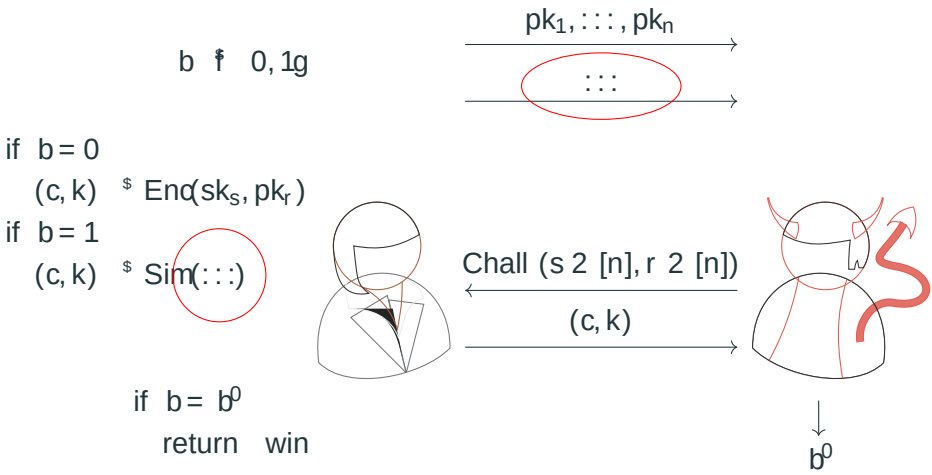

- Con dentiality: $k_{sr}$ and $k_{rs}$ should look random. 3
- Authenticity: r knows s sent the ciphertext c. 3

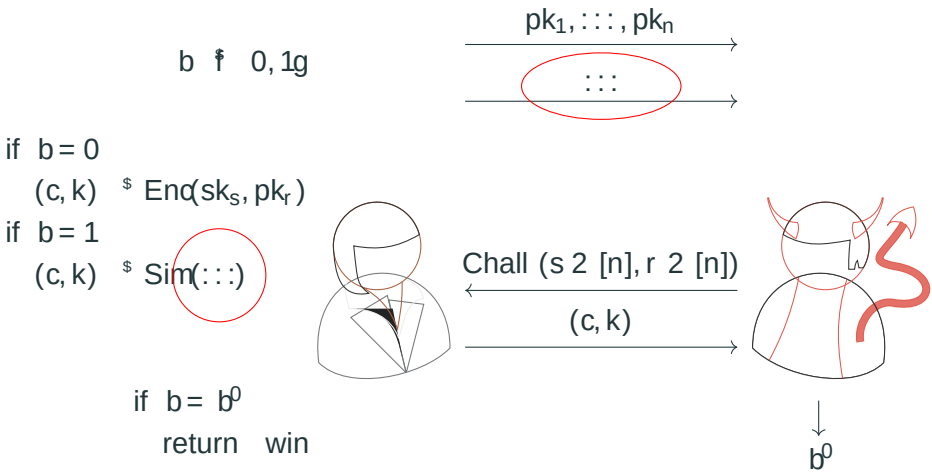
# authenticated key encapsulation mechanism ( AKEM) [ABH+21]

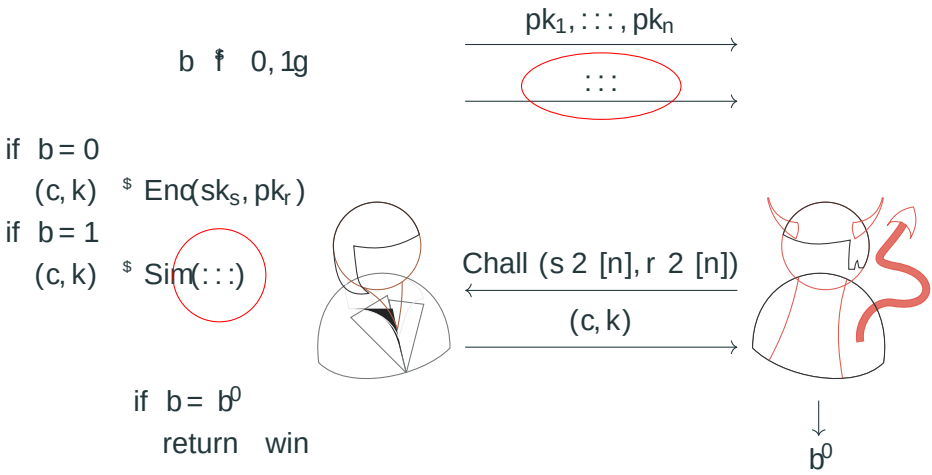


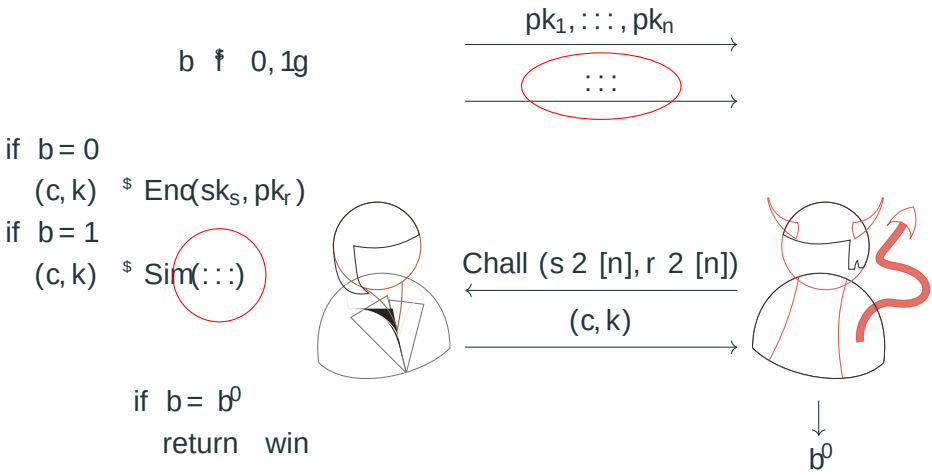

- Con dentiality: $k_{sr}$ and $k_{rs}$ should look random. 3
- Authenticity: r knows s sent the ciphertext c. 3
- Deniability: s can deny having sentc to r .

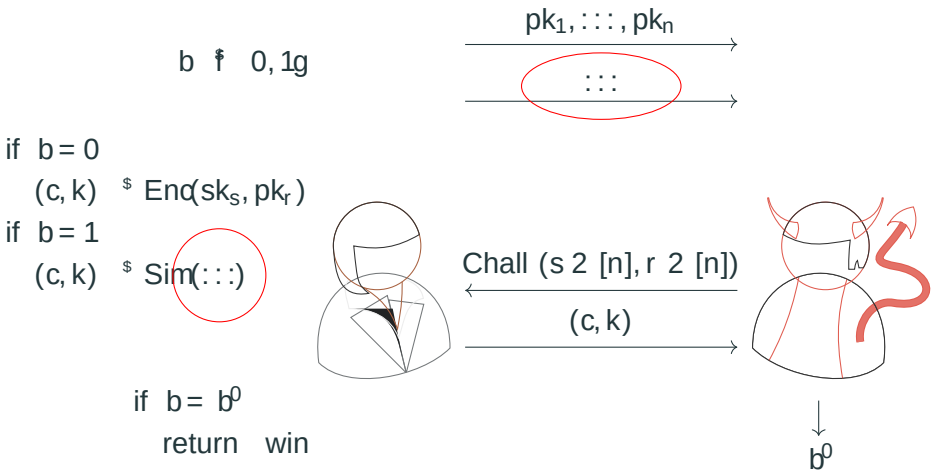

b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0

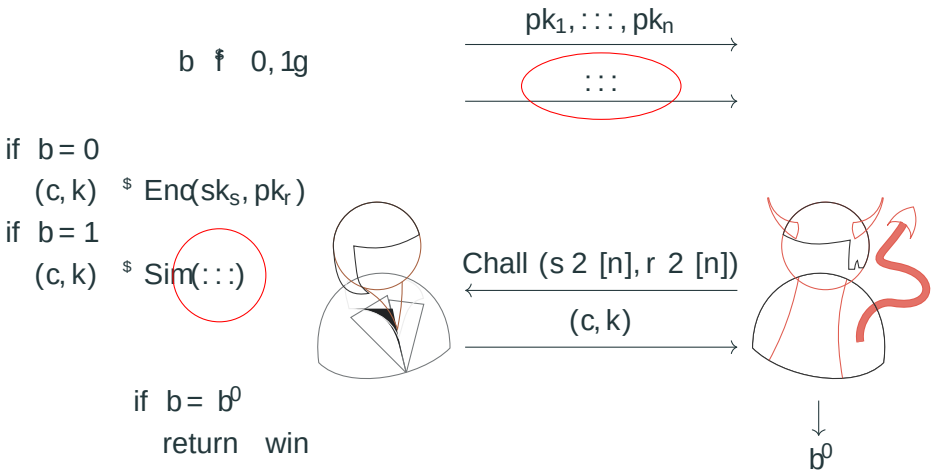

b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0

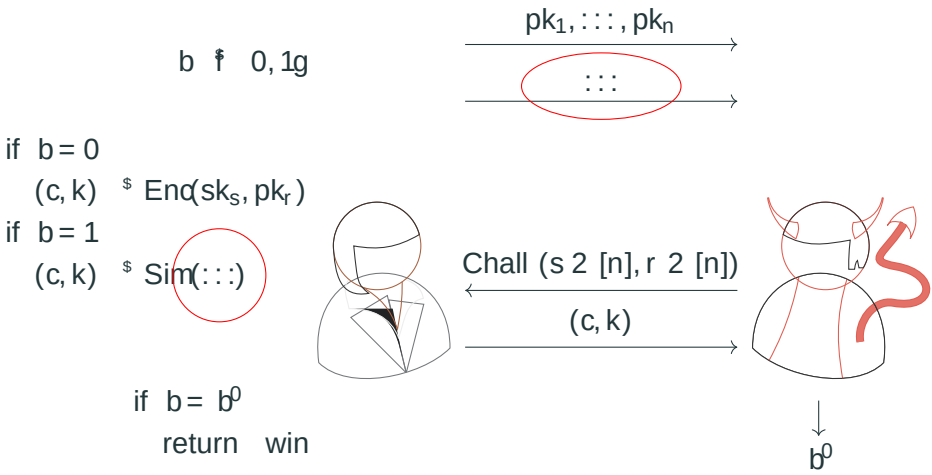

b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0

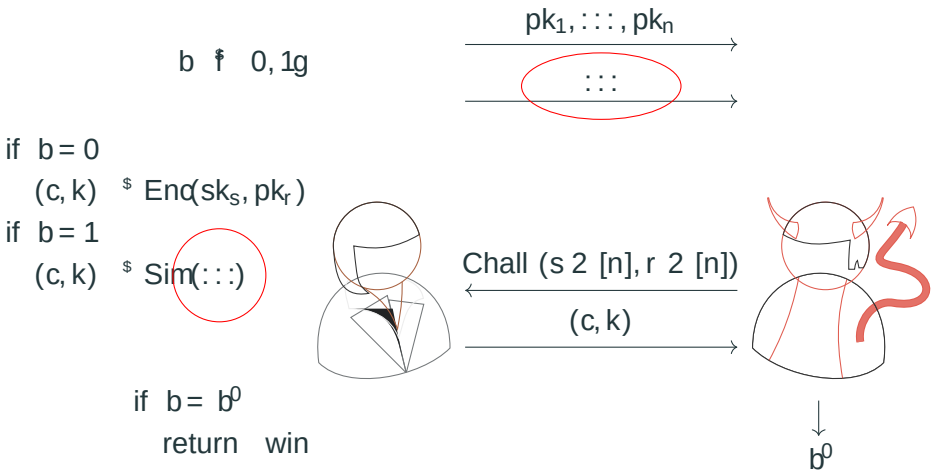

b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0


b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0

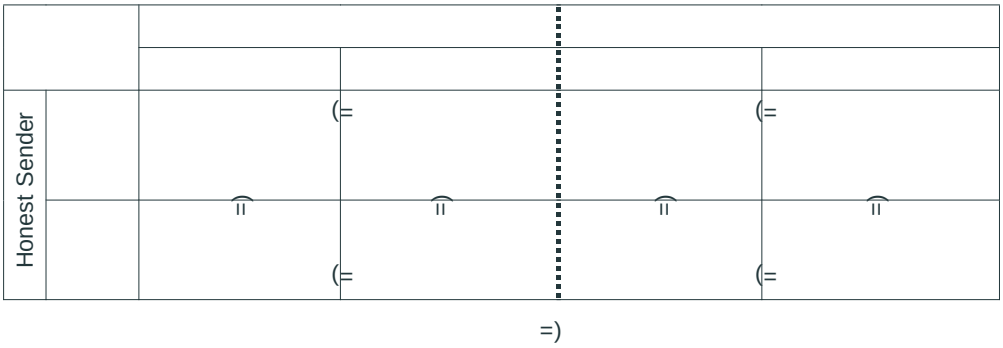

b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0


b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0

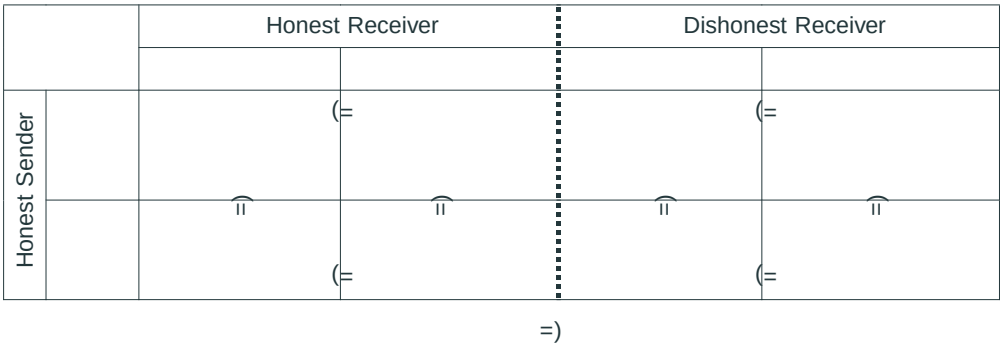

b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0


b f 0, 1g
pk1, : : : , pkn
: : :
if b = 0
(c, k) $ Enc(sks, pkr )
if b = 1
(c, k) $ Sim(: : :)
Chall (s 2 [n], r 2 [n])
(c, k)
if b = b0
return win
b0

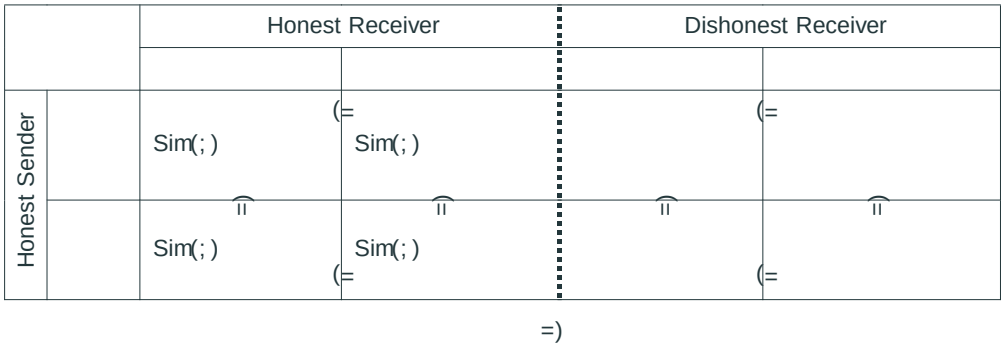
# deniability for AKEM: different notions



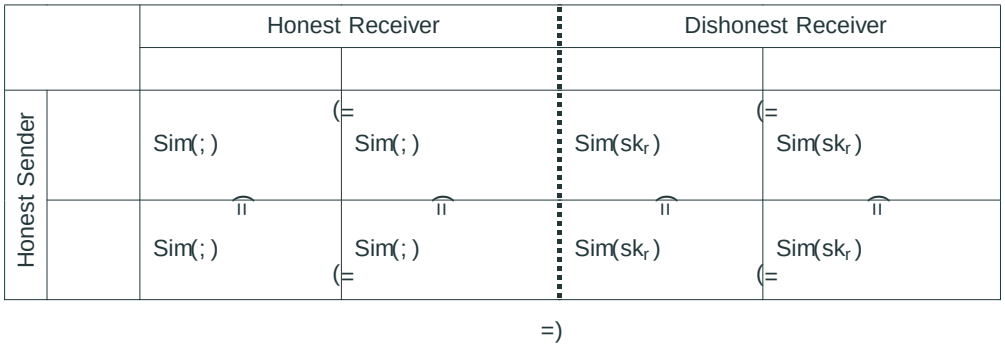
# deniability for AKEM: different notions



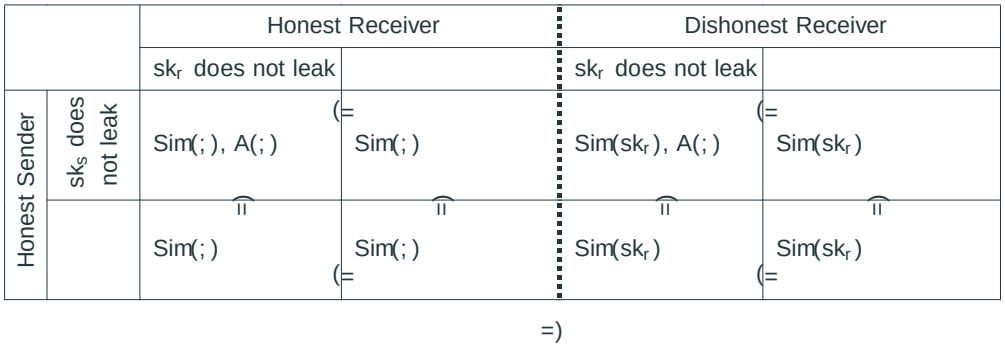
# deniability for AKEM: different notions



# deniability for AKEM: different notions



# deniability for AKEM: different notions



# deniability for AKEM: different notions

| | | Honest Receiver | | Dishonest Receiver | |
|---|---|---|---|---|---|
| | | $sk_r$ does not leak | | $sk_r$ does not leak | |
| Honest Sender | $sk_s$ does not leak | $\approx$ $Sim(;), A(;)$ | $Sim(;)$ | $\approx$ $Sim(sk_r), A(;)$ | $Sim(sk_r)$ |
| | $sk_s$ leaks | $\approx$ $Sim(;), A(sk_s)$ | $\approx$ $Sim(;)$ | $\approx$ $Sim(sk_r), A(sk_s)$ | $\approx$ $Sim(sk_r)$ |

$\approx$

# deniability for AKEM: different notions

| | | Honest Receiver | | Dishonest Receiver | |
|---|---|---|---|---|---|
| | | $sk_r$ does not leak | $sk_r$ leaks | $sk_r$ does not leak | $sk_r$ leaks |
| Honest Sender | $sk_s$ does not leak | $Sim(;), A(;)$ | $Sim(;), A(sk_r)$ | $Sim(sk_r), A(;)$ | $Sim(sk_r), A(sk_r)$ |
| | $sk_s$ leaks | $Sim(;), A(sk_s)$ | $Sim(;), A(sk_s, sk_r)$ | $Sim(sk_r), A(sk_s)$ | $Sim(sk_r), A(sk_s, sk_r)$ |

# deniability for AKEM: different notions

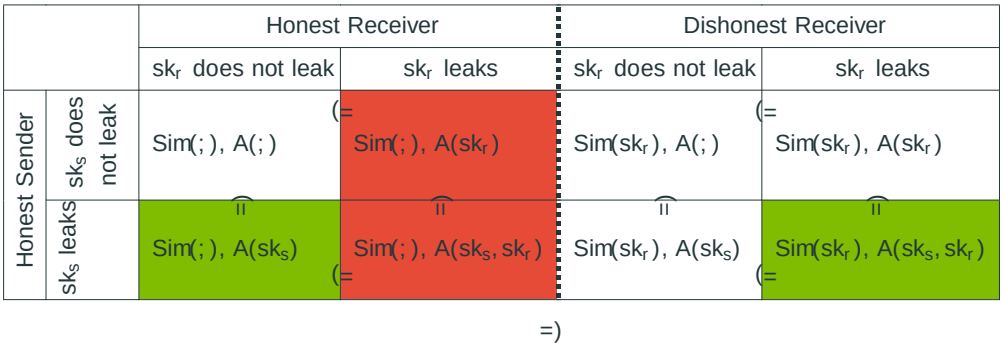
| | | Honest Receiver | | Dishonest Receiver | |
|---|---|---|---|---|---|
| | | $sk_r$ does not leak | $sk_r$ leaks | $sk_r$ does not leak | $sk_r$ leaks |
| Honest Sender | $sk_s$ does not leak | $Sim(;), A(;)$ | $Sim(;), A(sk_r)$ | $Sim(sk_r), A(;)$ | $Sim(sk_r), A(sk_r)$ |
| | $sk_s$ leaks | $Sim(;), A(sk_s)$ | $Sim(;), A(sk_s, sk_r)$ | $Sim(sk_r), A(sk_s)$ | $Sim(sk_r), A(sk_s, sk_r)$ |

# deniability for AKEM: different notions

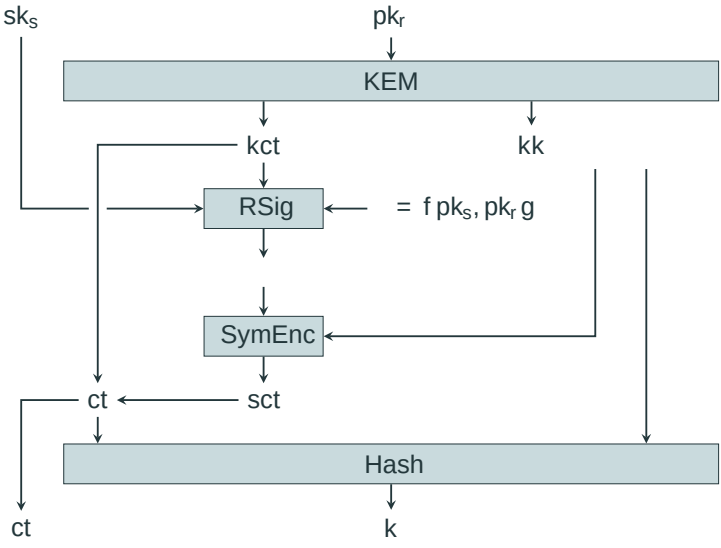
| | | Honest Receiver | | Dishonest Receiver | |
|---|---|---|---|---|---|
| | | $sk_r$ does not leak | $sk_r$ leaks | $sk_r$ does not leak | $sk_r$ leaks |
| Honest Sender | $sk_s$ does not leak | $Sim(;), A(;)$ | $Sim(;), A(sk_r)$ | $Sim(sk_r), A(;)$ | $Sim(sk_r), A(sk_r)$ |
| | $sk_s$ leaks | $Sim(;), A(sk_s)$ | $Sim(;), A(sk_s, sk_r)$ | $Sim(sk_r), A(sk_s)$ | $Sim(sk_r), A(sk_s, sk_r)$ |

(= (= (= (= (= =)

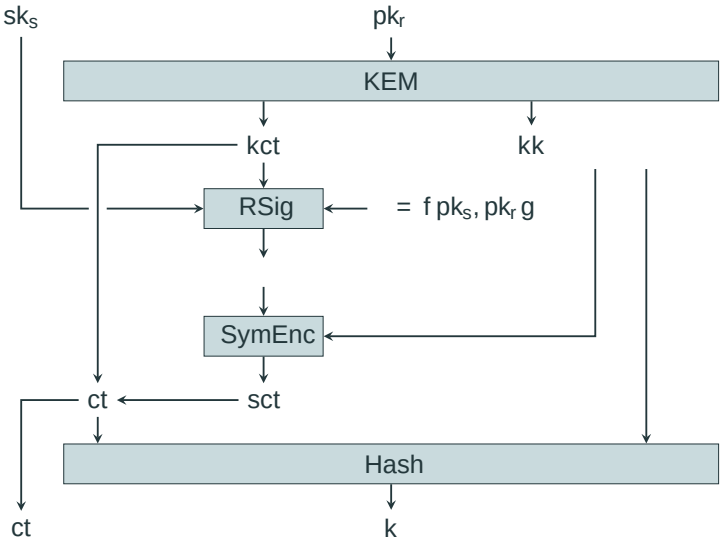
# deniability for AKEM: different notions



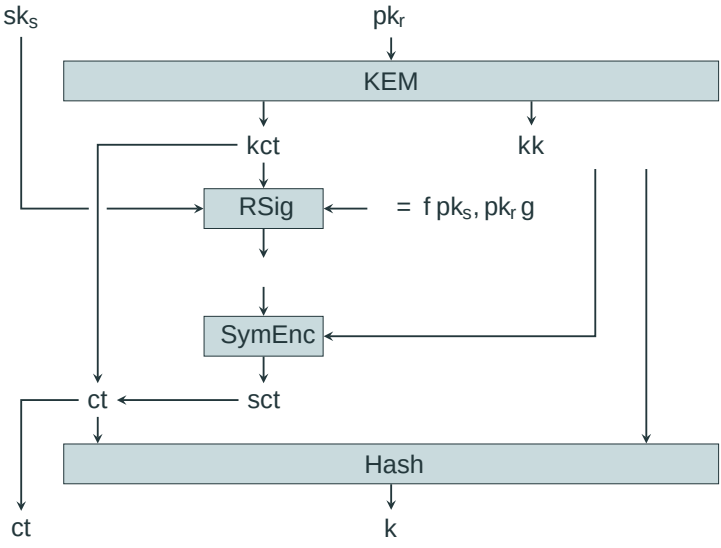
# deniable AKEM: black-box construction



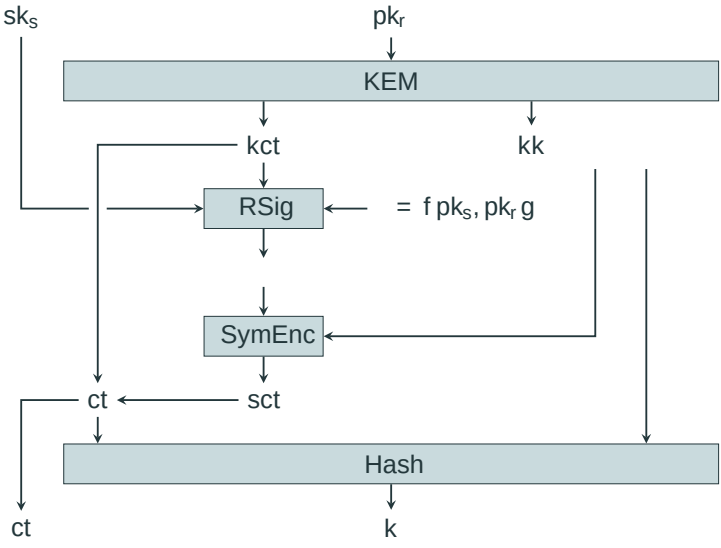
# deniable AKEM: black-box construction



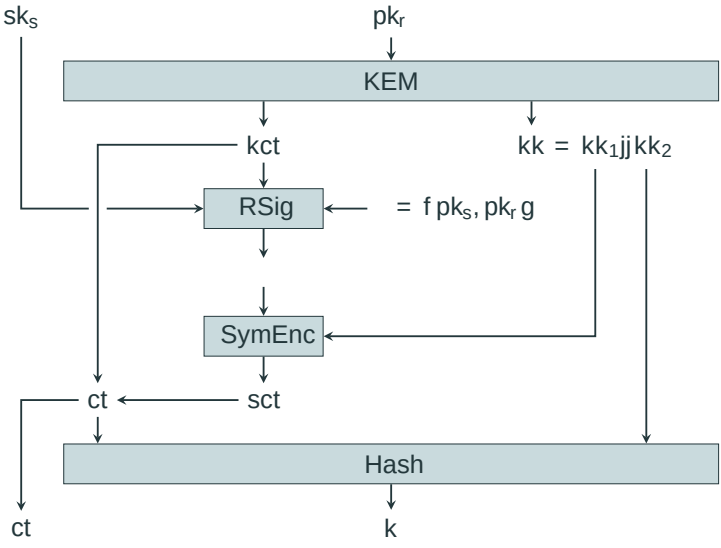
# deniable AKEM: black-box construction



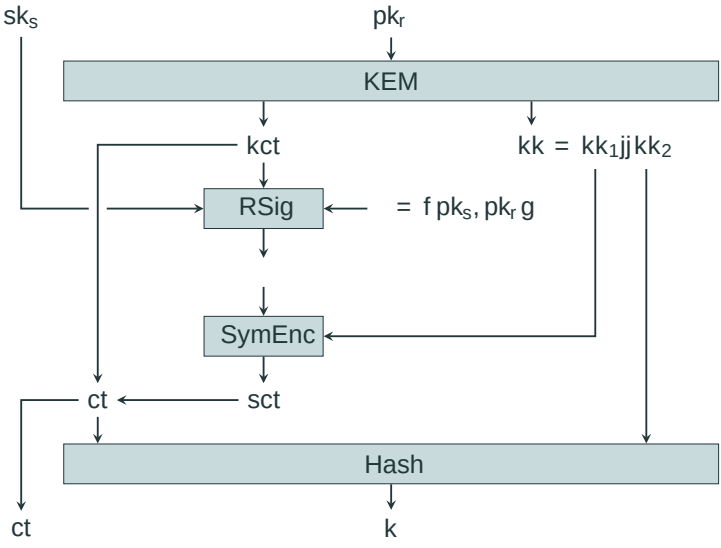
# deniable AKEM: black-box construction



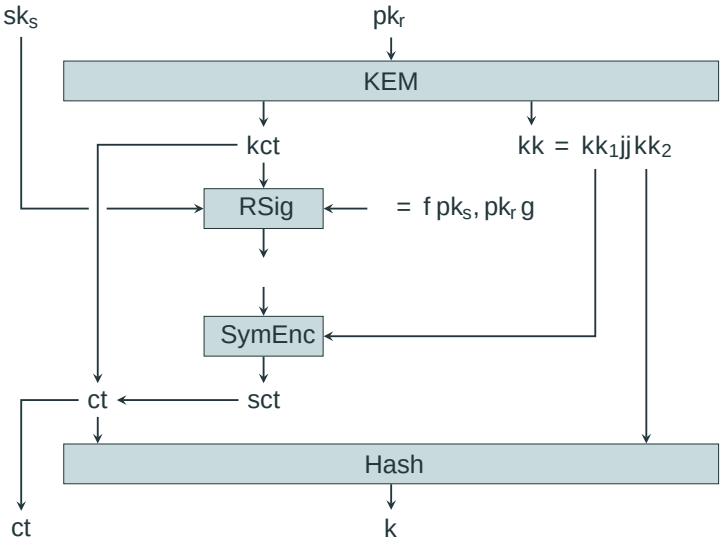
# deniable AKEM: black-box construction



# deniable AKEM: black-box construction



# deniable AKEM: black-box construction



# AKEM comparison

| Scheme (variant) | Con dentiality | Authenticity | Deniability | PQ | Size (in bytes) | |
|---|---|---|---|---|---|---|
| | | | | | c | pk |
| DH-AKEM ( Curve25519) [ABH+21] | Ins-CCA | Out-Aut | DR -Den | 7 | 32 | 32 |
| EtStH-AKEM ( NTRU-A + Antrag ) [AJKL23] | Ins-CCA | Out-Aut | | 3 | 1 414 | 1 664 |
| NIKE-AKEM ( Swoosh[1]) [AJKL23] | Ins-CCA | Out-Aut | DR -Den | 3 | > 221 184 | > 221 184 |
| FrodoKEX+ [CHDN+24] | IND-1BatchCCA | UNF-1KCA | DR -Den | 3 | 72 | 21 300 |
| This Work (NTRU-A + Gandalf ) | Ins-CCA | Out-Aut | HR -Den & DR -Den | 3 | 2 004 | 1 664 |

# summary

# summary

Contributions:

- Gandalf — an NTRU-based ring signature scheme:
  - 50% reduction in signature size compared to Raptor [LAZ19].
  - For rings of size two, $|j| = 1236$ bytes, a quarter the size of DualRing [YEL+21].
- Formalised deniability for AKEM, the primitive behind HPKE used in MLS.
- Black-box construction of deniable AKEM:
  - Ciphertext size of 2004 bytes when instantiated with Gandalf.

ia.cr/2024/890


# phillip.gajland@{mpi-sp.org,rub.de}
Š p4i11ip

# summary

Contributions:

- Gandalf — an NTRU-based ring signature scheme:
  - 50% reduction in signature size compared to Raptor [LAZ19].
  - For rings of size two, $|\sigma| = 1236$ bytes, a quarter the size of DualRing [YEL+21].
- Formalised deniability for AKEM, the primitive behind HPKE used in MLS.
- Black-box construction of deniable AKEM:
  - Ciphertext size of 2004 bytes when instantiated with Gandalf.

ia.cr/2024/890

# phillip.gajland@{mpi-sp.org,rub.de}
Š p4i11ip

# summary

Contributions:

- Gandalf — an NTRU-based ring signature scheme:
  - 50% reduction in signature size compared to Raptor [LAZ19].
  - For rings of size two, | | = 1236 bytes, a quarter the size of DualRing [YEL+21].
- Formalised deniability for AKEM, the primitive behind HPKE used in MLS.
- Black-box construction of deniable AKEM:
  - Ciphertext size of 2004 bytes when instantiated with Gandalf.

ia.cr/2024/890

# phillip.gajland@{mpi-sp.org,rub.de}
Š p4i11ip

# summary

Contributions:

- Gandalf — an NTRU-based ring signature scheme:
  - 50% reduction in signature size compared to Raptor [LAZ19].
  - For rings of size two, | | = 1236 bytes, a quarter the size of DualRing [YEL+21].
- Formalised deniability for AKEM, the primitive behind HPKE used in MLS.
- Black-box construction of deniable AKEM:
  - Ciphertext size of 2004 bytes when instantiated with Gandalf.

ia.cr/2024/890


# phillip.gajland@{mpi-sp.org,rub.de}
Š p4i11ip

# summary

Contributions:

- Gandalf an NTRU-based ring signature scheme:
  - 50% reduction in signature size compared to Raptor [LAZ19].
  - For rings of size two, | | = 1236 bytes, a quarter the size of DualRing [YEL+21].
- Formalised deniability for AKEM, the primitive behind HPKE used in MLS.
- Black-box construction of deniable AKEM:
  - Ciphertext size of 2004 bytes when instantiated with Gandalf .

ia.cr/2024/890

# phillip.gajland@{mpi-sp.org,rub.de}
Š p4i11ip

# summary

Contributions:

- Gandalf an NTRU-based ring signature scheme:
  - 50% reduction in signature size compared to Raptor [LAZ19].
  - For rings of size two, $|\sigma| = 1236$ bytes, a quarter the size of DualRing [YEL+21].
- Formalised deniability for AKEM, the primitive behind HPKE used in MLS.
- Black-box construction of deniable AKEM:
  - Ciphertext size of 2004 bytes when instantiated with Gandalf.


- ia.cr/2024/890

- phillip.gajland@{mpi-sp.org,rub.de}
- p4i11ip

# summary

Contributions:

- Gandalf an NTRU-based ring signature scheme:
  - 50% reduction in signature size compared to Raptor [LAZ19].
  - For rings of size two, $|\sigma| = 1236$ bytes, a quarter the size of DualRing [YEL+21].
- Formalised deniability for AKEM, the primitive behind HPKE used in MLS.
- Black-box construction of deniable AKEM:
  - Ciphertext size of 2004 bytes when instantiated with Gandalf.


- ia.cr/2024/890

- phillip.gajland@{mpi-sp.org,rub.de}
- p4i11ip

# references I


[ABH+21] Joël Alwen, Bruno Blanchet, Eduard Hauck, Eike Kiltz, Benjamin Lipp, and Doreen Riepel. Analysing the HPKE standard. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 87–116, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland.

[AJKL23] Joël Alwen, Jonas Janneck, Eike Kiltz, and Benjamin Lipp. The pre-shared key modes of HPKE. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VI*, volume 14443 of *Lecture Notes in Computer Science*, pages 329–360, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.

[BBLW22] Richard Barnes, Karthikeyan Bhargavan, Benjamin Lipp, and Christopher A. Wood. Hybrid Public Key Encryption. RFC 9180, February 2022.

[BBR+23] Richard Barnes, Benjamin Beurdouche, Raphael Robert, Jon Millican, Emad Omara, and Katriel Cohn-Gordon. The Messaging Layer Security (MLS) Protocol. RFC 9420, July 2023.

[CHDN+24] Daniel Collins, Loïs Huguenin-Dumittan, Ngoc Khanh Nguyen, Nicolas Rolin, and Serge Vaudenay. K-waay: Fast and deniable post-quantum x3dh without ring signatures. Cryptology ePrint Archive, Paper 2024/120, 2024. https://eprint.iacr.org/2024/120.

[LAZ19] Xingye Lu, Man Ho Au, and Zhenfei Zhang. Raptor: A practical lattice-based (linkable) ring signature. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19: 17th International Conference on Applied Cryptography and Network Security*, volume 11464 of *Lecture Notes in Computer Science*, pages 110–130, Bogota, Colombia, June 5–7, 2019. Springer, Cham, Switzerland.

[LDK+22] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

# references II


[LNS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. SMILE: Set membership from ideal lattices with applications to ring signatures and confidential transactions. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 611–640, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, Gold Coast, Australia, December 9–13, 2001. Springer, Berlin, Heidelberg, Germany.

[SAB+22] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[YEL+21] Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. DualRing: Generic construction of ring signatures with efficient instantiations. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 251–281, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

# Gandalf parameters

| Parameter | Description | Value |
|---|---|---|
| $N$ | dimension of $R := \mathbb{Z}[X]/(X^N + 1)$ | 512 |
| | Smoothing parameter order | $\sqrt{\frac{1}{Q_{\mathsf{Sgn}} \cdot \lambda}}$ |
| $\delta_{KL}$ | maximum KL-divergence of PreSmp | 2 |
| $a$ | Rényi order | $2\lambda$ |
| $R_a$ | maximum Rényi divergence of PreSmp | $1 + 2a^{-2}$ |
| $\alpha$ | quality of NTRU trapdoor | 1.15 |
| $q$ | prime modulus | 12289 |
| $s$ | standard deviation of Gaussian sampler | $\frac{1}{\pi} \cdot \sqrt{\frac{\ln(4N(1+1/\epsilon))}{2}} \cdot \alpha \cdot \sqrt{q}$ |
| $\tau$ | tailcut rate of signatures | $[1.08, 1.22]$ |
| $\kappa$ | maximum size of signing ring | $\geq 2$ |
| $\lvert\rho\rvert = k$ | size of signing ring | $[2, \kappa]$ |
| $\beta$ | maximum norm of signatures | $\tau \cdot s \cdot \sqrt{(\kappa + 1)N}$ |
| $\lvert pk\rvert$ | verification key size (bytes) | 896 |
| $\lvert\sigma\rvert$ | signature size (bytes) | $606 \cdot k + 24$ |