

COUNT ME IN!

EXTENDABILITY FOR THRESHOLD

RING SIGNATURES

PKC '22



D. F. ARANHA, M. HALL-ANDERSEN, A. NITUDESCU, E. PAGNIN, AND S. YAKOUBOV

Digital Signatures

SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	 signer's pk	(msg, sgn)
-------	--------	------	--------	--	------------

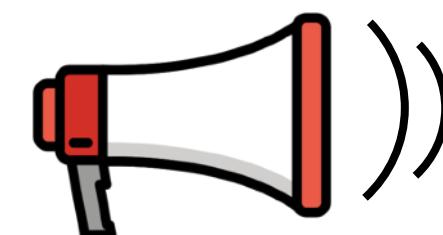
Ring Signatures (RS)

SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	signer's pk	●	(msg, sgn)
-------	--------	------	--------	---------------	---	------------

RING SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	ring of n potential signers	● ● ● ●	(msg, sgn)
-------	--------	------	--------	-------------------------------	---------	------------



Threshold Ring Signatures (TRS)

SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	signer's pk	(msg, sgn)
-------	--------	------	--------	---------------	------------

RING SIGNATURE SCHEME

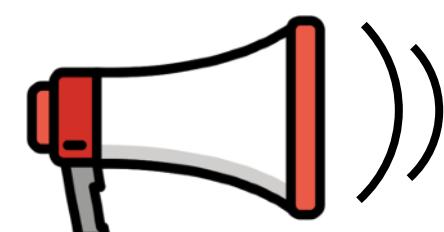
Setup	KeyGen	Sign	Verify	ring of n potential signers	(msg, sgn)
-------	--------	------	--------	-------------------------------	------------

THRESHOLD RING SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	t out of n signers	(msg, sgn)
-------	--------	------	--------	------------------------	------------



interaction among signers
agreement on the ring of public keys



Extendable Threshold Ring Signatures (ExTRS)

SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	signer's pk	(msg, sgn)
-------	--------	------	--------	---------------	------------

RING SIGNATURE SCHEME

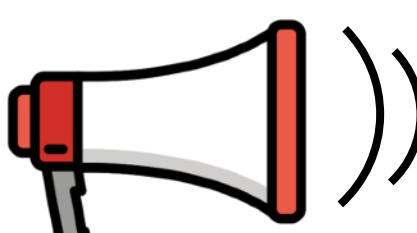
Setup	KeyGen	Sign	Verify	ring of n potential signers	(msg, sgn)
-------	--------	------	--------	-------------------------------	------------

THRESHOLD RING SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	t out of n signers	(msg, sgn)
-------	--------	------	--------	------------------------	------------

EXTENDABLE THRESHOLD RING SIGNATURE SCHEME

Setup	KeyGen	Sign	Verify	t out of n signers	(msg, sgn1)
-------	--------	------	--------	------------------------	-------------



count me in!



me too!

I endorse this!

Extend



t out of n' signers
(msg, sgn')

Join



$t+1$ out of n' signers
(msg, sgn+)



dynamic ring growth
no interaction among signers needed

Our Contribution



Extendability (enlarge the set of potential signers of a given signature)

Ring Signatures (RS)

Same-Message Linkable RS

Threshold Ring Signatures



Formal Syntax & Security Models



Constructions & Implementations

ERS from Signatures of Knowledge (SoK)

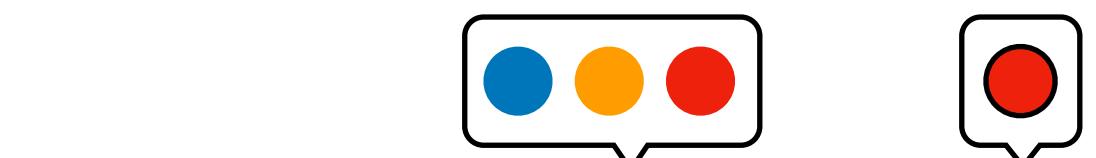
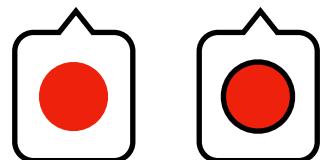
SMLERS black-box from ERS and SoK

ETRS black-box from SMLERS and from dLog

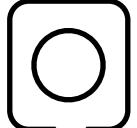
Threshold Ring Signatures (Syntax)

$\text{Setup}(\text{sec.par}) \rightarrow \text{pp}$

$\text{KeyGen}(\text{pp}) \rightarrow (\text{pk}, \text{sk})$



$\text{Sign}(m, \{\text{pk}_i\}_{i \in R}, \text{sk}) \rightarrow s$

 $\text{Verify}(t, m, s, \{\text{pk}_i\}_{i \in R}) \rightarrow 0/1$

at least t secret keys
were used to generate
the signature s for m

Bresson, Stern, and Szydlo: "Threshold ring signatures and applications to ad-hoc groups", CRYPTO, 2002.

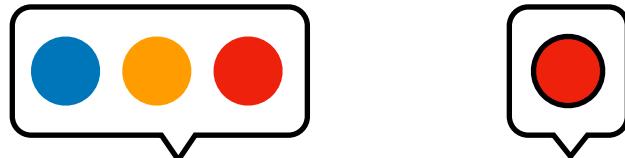
Tsang, Wei, Chan, Au, Liu, Wong: "Separable linkable threshold ring signatures", Indocrypt, 2004.

Melchor, Cayrel, Gaborit, Laguillaumie: "A new efficient threshold ring signature scheme based on coding theory", IEEE-ToIT, 2011.

Munch-Hansen, Orlandi, Yakoubov: "Stronger Notions and a More Efficient Construction of Threshold Ring Signatures." 2020/678. 7

Extendable Threshold Ring Signatures (Syntax)

Setup(sec.par) \rightarrow pp

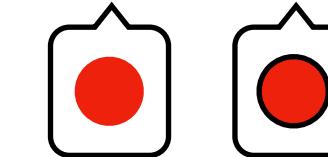


Sign($m, \{pk_i\}_{i \in R}, sk$) \rightarrow s



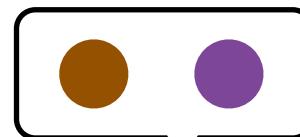
Verify($t, m, s, \{pk_i\}_{i \in R}$) \rightarrow 0/1

KeyGen(pp) \rightarrow (pk, sk)



at least t secret keys
were used to generate
the signature s for m

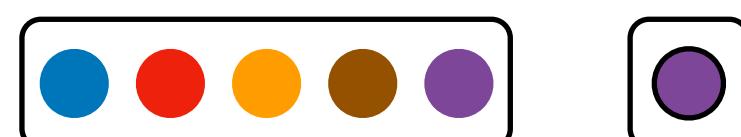
Extend($m, s, \{pk_i\}_{i \in R_1}, \{pk_i\}_{i \in R_2}$) \rightarrow s'



the new signature s' verifies for
the same threshold t as s did and
for the larger ring

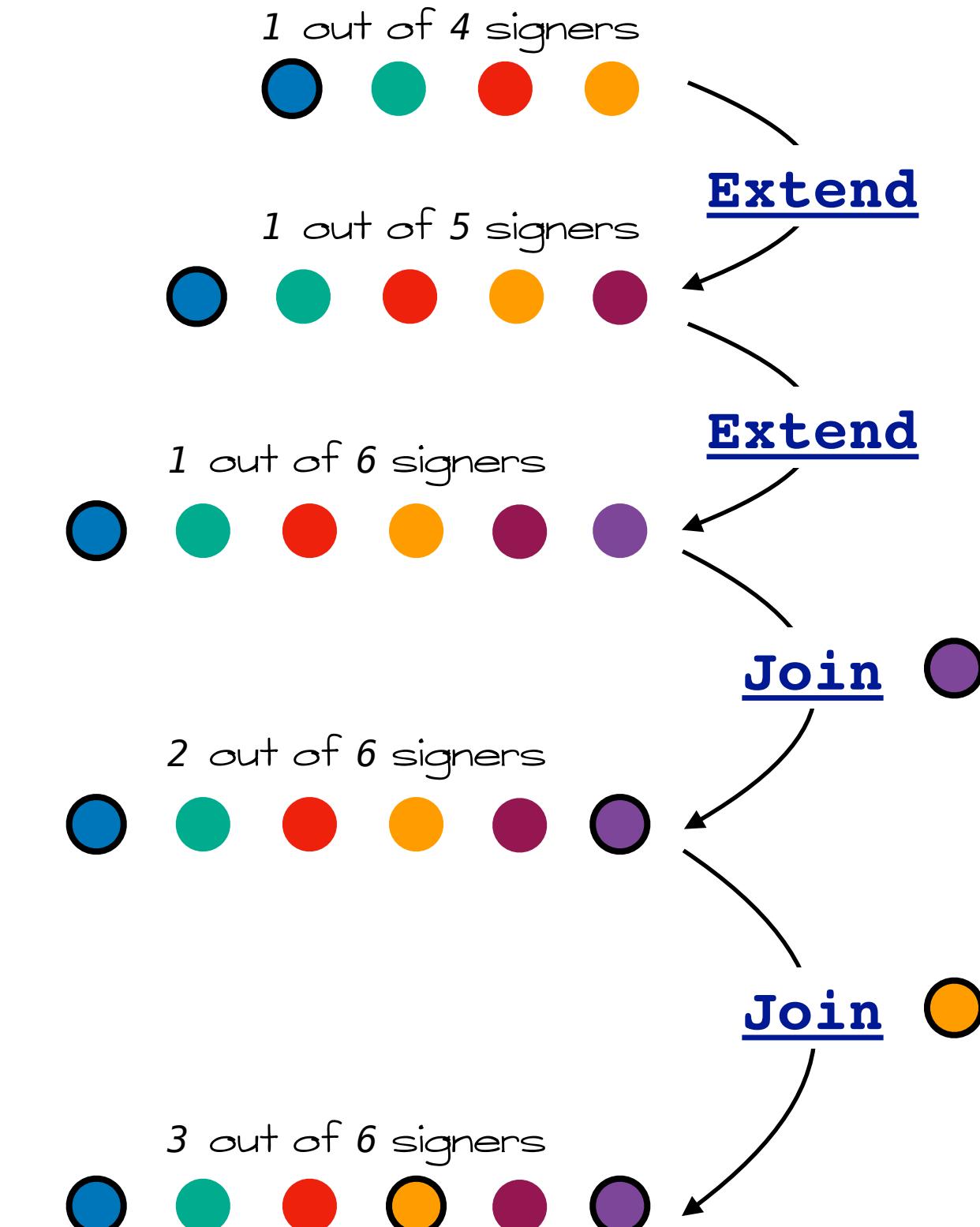
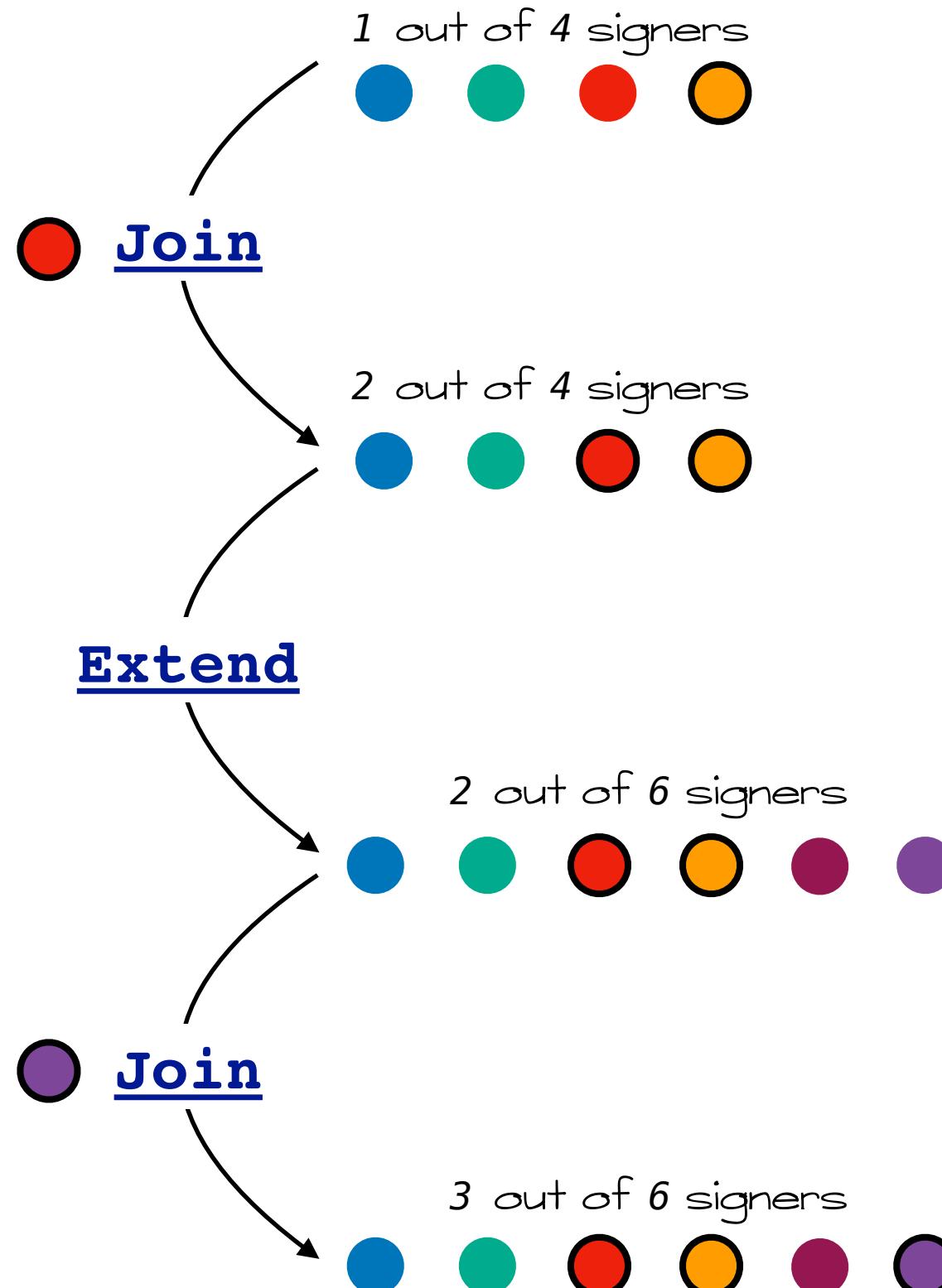


Join($m, s, \{pk_i\}_{i \in R}, sk$) \rightarrow s'



the new signature s' verifies
for threshold $t_s + 1$

Visualizing Extendibility (Ladders)



ETRS: Security Model

UNFORGEABILITY



O.KeyGen
O.Corrupt
O.Sign

$(t, m, s, \{pk_i\}_{i \in R})$

\mathcal{A} wins the EUF game if:

- 1- $\text{Verify}(t, m, s, R) = 1$
- 2- # corrupted signers in $R < t$
- 3- # O.Sign queries for $m < t$

ANONYMITY & ANONYMOUS EXTENDABILITY



O.KeyGen
O.Corrupt
O.Sign

$(m, lad0, lad1)$

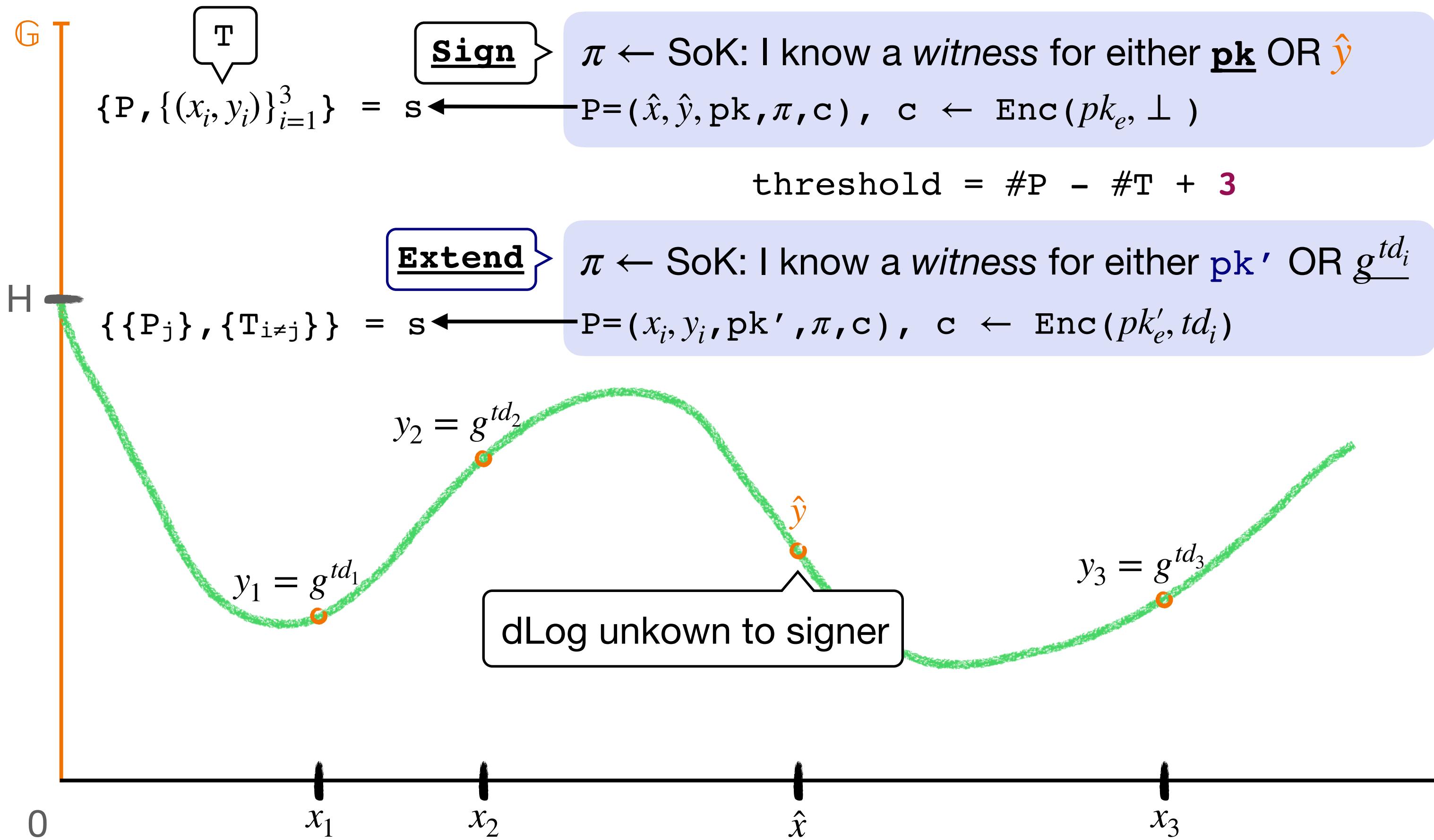
s



\mathcal{A} wins if it correctly guesses what ladder has been chosen to generate s

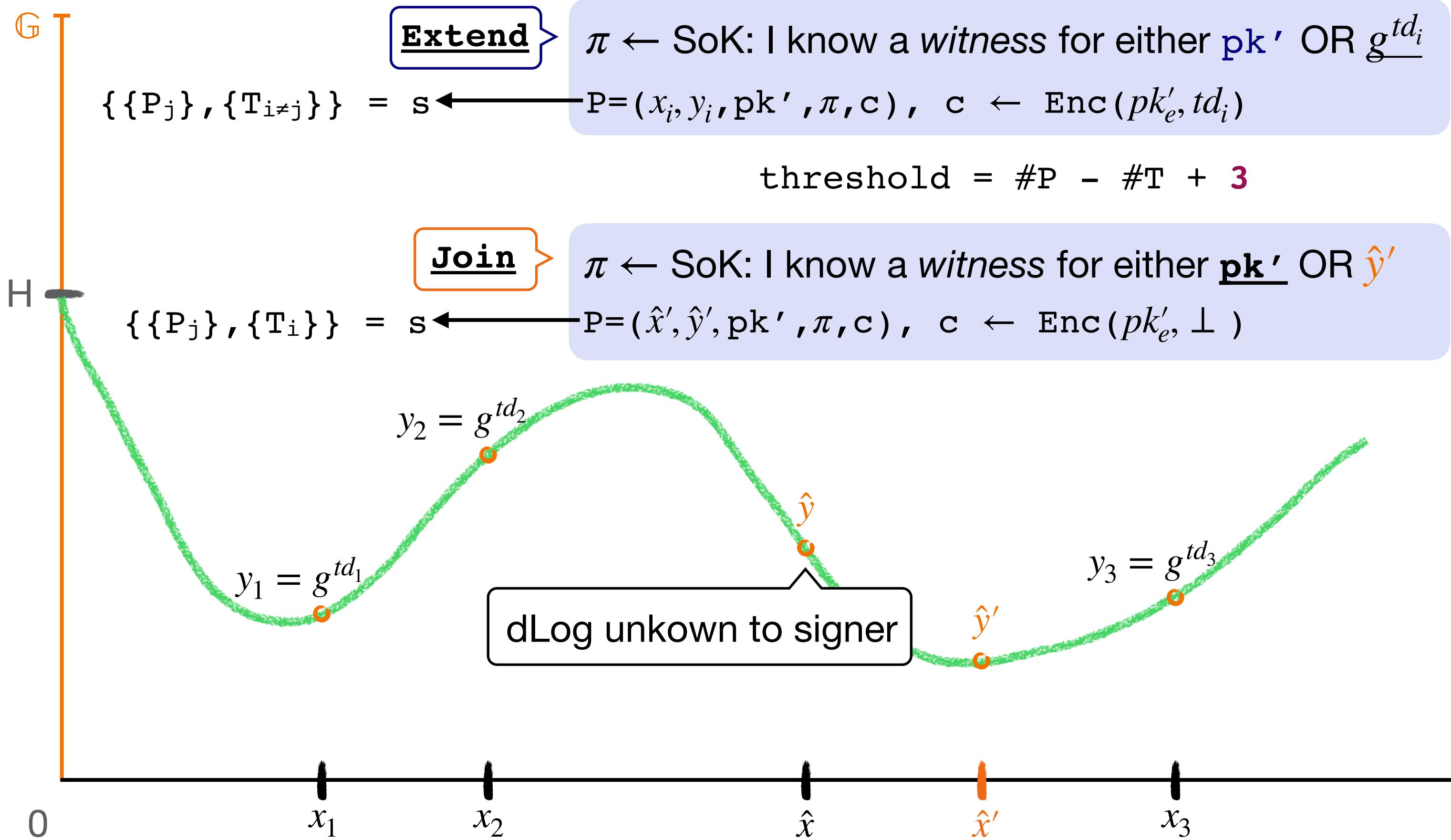
Our Construction: ETRS From dLog (and PKE)

interpolation in the exponent: polynomial of degree 3 on $\{H, g^{td_1}, g^{td_2}, g^{td_3}\}$



Our Construction: ETRS From dLog (and PKE)

interpolation in the exponent: polynomial of degree 3 on $\{H, g^{td_1}, g^{td_2}, g^{td_3}\}$

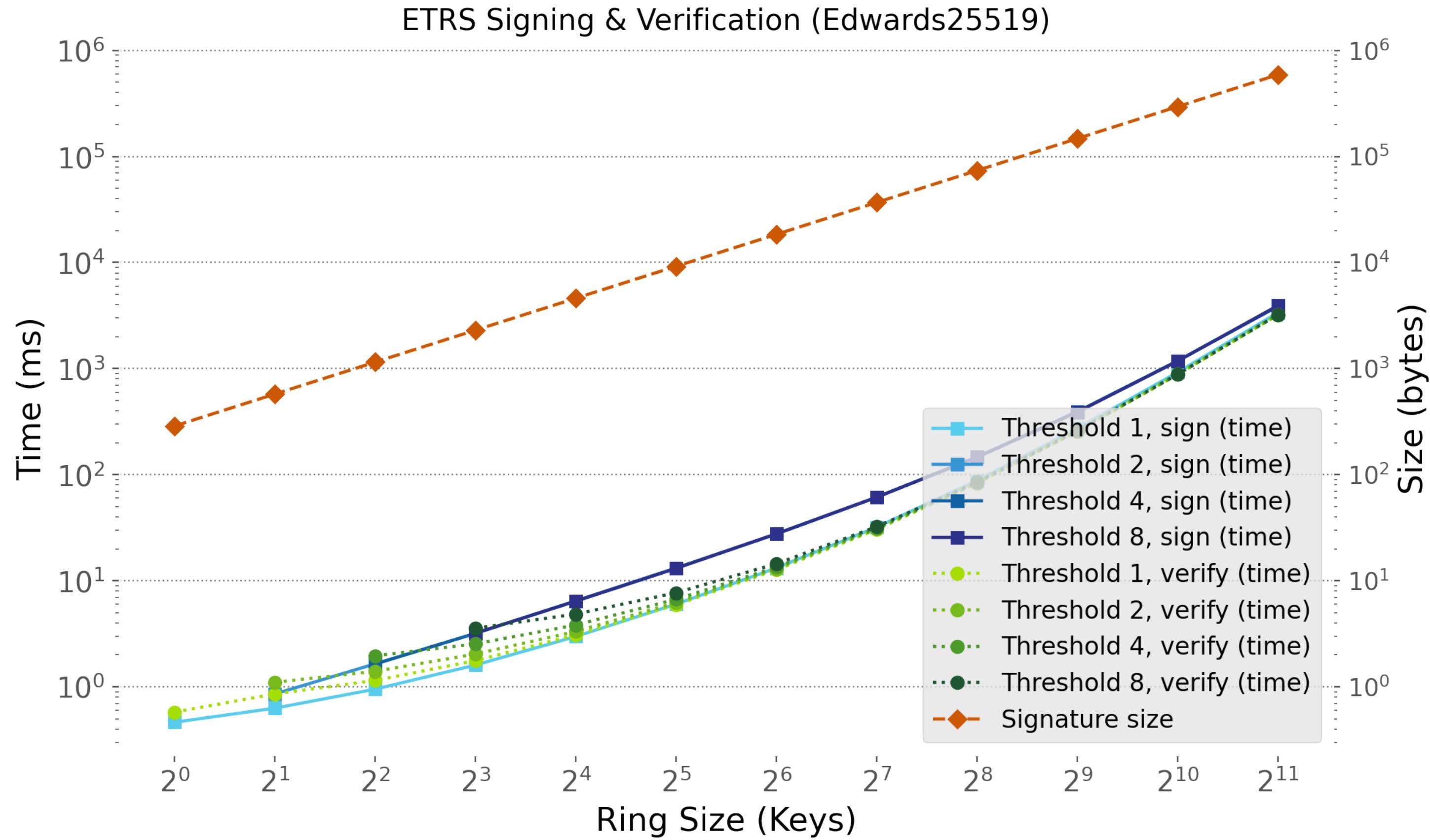


Our Construction

$$\mathcal{R}_{\mathbb{G}} = \{(\phi, w) = (h, \mathbf{pk}, x) \in \mathbb{G} \times \mathbb{G} \times \mathbb{Z}_p : g^x = h \vee g^x = \mathbf{pk}\}$$

KeyGen() $\mapsto (\mathbf{pk}, \mathbf{sk})$	Extend ($\mu, \{\mathbf{pk}_j\}_{j \in \mathcal{R}}, \sigma, \mathbf{pk}$) $\mapsto \sigma'$
1 : $(\mathbf{pk}_s, \mathbf{sk}_s) \leftarrow \mathbf{ERS.KeyGen}()$	1 : if $\mathbf{pk} \in \{\mathbf{pk}_j\}_{j \in \mathcal{R}}$: return \perp
2 : $(\mathbf{pk}_e, \mathbf{sk}_e) \leftarrow \mathbf{PKE.KeyGen}()$	2 : $(\hat{x}, \hat{\mathbf{td}}) \leftarrow_R T$ // Pick eval-point and trapdoor
3 : return $(\mathbf{pk} = (\mathbf{pk}_s, \mathbf{pk}_e), \mathbf{sk} = (\mathbf{sk}_s, \mathbf{sk}_e))$	3 : $c' \leftarrow \mathbf{Enc}(\mathbf{pk}_e, \hat{\mathbf{td}})$ // enable future endorsing
Sign (μ, \mathbf{sk}) $\mapsto \sigma$	// interpolate a unique representation of the polynomial
1 : $X \leftarrow_R \binom{\mathbb{Z}_p^*}{n'}$ // pick n' distinct evaluation points	4 : $(y', \pi') \leftarrow \mathbf{PolySign}(P, T, \hat{x}, w := \hat{x}, \mathbf{pk}, \mu)$
2 : $T := \emptyset; P := \emptyset$	5 : $T \leftarrow T \setminus \{(\hat{x}, \hat{\mathbf{td}})\}$ // erase used trapdoor
3 : for $x \in X$	// Add simulated signature to the set of proofs
4 : $\mathbf{td} \leftarrow_R \mathbb{Z}_p$ // generate trapdoors for poly. values	6 : $P \leftarrow P \cup \{(\hat{x}, y', \mathbf{pk}_s, \pi', c')\}$
5 : $T \leftarrow T \cup \{(x, \mathbf{td})\}$ // populate trapdoor set	7 : Randomly permute P
6 : $c \leftarrow \mathbf{Enc}(\mathbf{pk}_e, \perp)$ // no info to pass on	8 : return $\sigma' := (T, P)$
7 : $\hat{x} \leftarrow_R \mathbb{Z}_p^* \setminus X$ // pick a new evaluation point	
8 : $(y, \pi) \leftarrow \mathbf{PolySign}(P, T, \hat{x}, w := \mathbf{sk}, \mathbf{pk}, \mu)$	
9 : $P := \{(\hat{x}, y, \mathbf{pk}_s, \pi, c)\}$	
10 : return $\sigma := (T, P)$	
Join ($\mu, \{\mathbf{pk}_j\}_{j \in \mathcal{R}}, \mathbf{sk}, \sigma$) $\mapsto \sigma'$	Verify ($t, \mu, \{\mathbf{pk}_j\}_{j \in \mathcal{R}}, \sigma$) $\mapsto \text{accept/reject}$
// check if current signer's \mathbf{pk}_s is in P	1 : if $\{\mathbf{pk}_j\}_{j \in \mathcal{R}} \neq \{\mathbf{pk}_i\}_{(\cdot, \cdot, \mathbf{pk}_i, \cdot, \cdot) \in P}$:
1 : if $\exists (x, y, \mathbf{pk}, \pi, c) \in P$ s.t. $\mathbf{pk} = \mathbf{pk}_s$	2 : return reject
// remove simulated proof for the signer who wants to join	// check y 's are consistent with a degree n' polynomial
2 : $P \leftarrow P \setminus \{(x, y, \mathbf{pk}_s, \pi, c)\}$	3 : $\mathcal{Z} := \{(0, H)\} \cup \{(x, g^{\mathbf{td}})\}_{(x, \mathbf{td}) \in T}$
// retrieve trapdoor value	4 : $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{(x, y)\}_{(x, y, \mathbf{pk}, c, \pi) \in P}$
3 : $\mathbf{td} \leftarrow \mathbf{Dec}(\mathbf{sk}_e, c)$	5 : Pick $\hat{\mathcal{Z}} \subseteq \mathcal{Z}$ s.t. $ \hat{\mathcal{Z}} = n' + 1$
// add eval. point and \mathbf{td} to the set of available trapdoors	6 : $\mathcal{X} := \{x\}_{(x, y) \in \mathcal{Z}}; \hat{\mathcal{X}} := \{x\}_{(x, y) \in \hat{\mathcal{Z}}}$
4 : $T \leftarrow T \cup \{(x, \mathbf{td})\}$	7 : for $(x, y) \in \mathcal{Z}$:
5 : $c' \leftarrow \mathbf{Enc}(\mathbf{pk}_e, \perp)$ // no info to pass on	8 : if $y \neq \prod_{(\hat{x}, \hat{y}) \in \hat{\mathcal{Z}}} \hat{y}^{L_{(\hat{\mathcal{X}}, \hat{x})}(x)}$: return reject
6 : $\hat{x} \leftarrow_R \mathbb{Z}_p^* \setminus X$ // pick a new evaluation point	// Interpolation over the standard set $\{1, \dots, n'\}$
// interpolate a unique representation of the polynomial	9 : for $i \in [n']$: $V_i \leftarrow \prod_{(x, y) \in \mathcal{Z}} y^{L_{(\mathcal{X}, x)}(i)}$
7 : $(y', \pi') \leftarrow \mathbf{PolySign}(P, T, \hat{x}, w := \mathbf{sk}, \mathbf{pk}, \mu)$	10 : $\hat{\mu} := (\mu, \{V_i\}_{i \in [n']})$
8 : $P \leftarrow P \cup \{(\hat{x}, y', \mathbf{pk}_s, \pi', c')\}$	11 : for $(x, y, \mathbf{pk}_s, \pi, c) \in P$ // check proofs individually
9 : Randomly permute P	12 : $\phi := (y, \mathbf{pk}_s)$
10 : return $\sigma := (T, P)$	13 : if $\mathbf{SoK.Verify}(\hat{\mu}, \mathcal{R}_{\mathbb{G}}, \phi, \pi) = \text{reject}$
	14 : return reject
	15 : if $ T + P \geq t + n'$ return accept
	16 : else return reject

ETRS From dLog



**THANK YOU FOR YOUR
ATTENTION**