

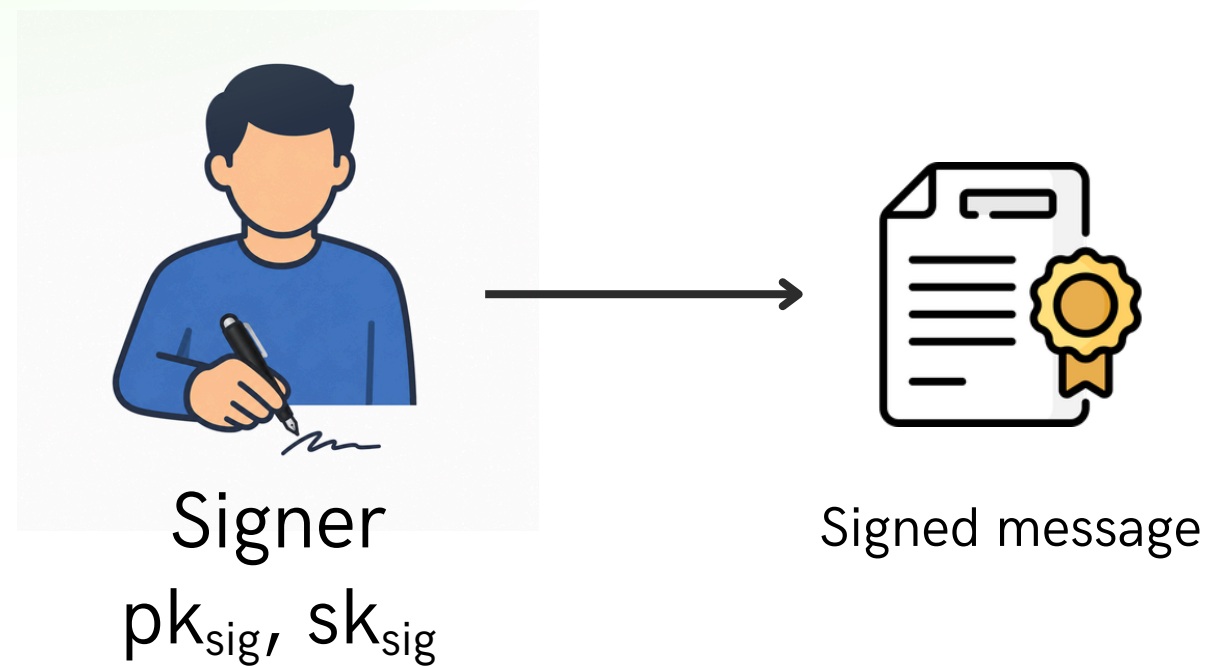
Dual-Forward-Secure Sanitizable Signatures



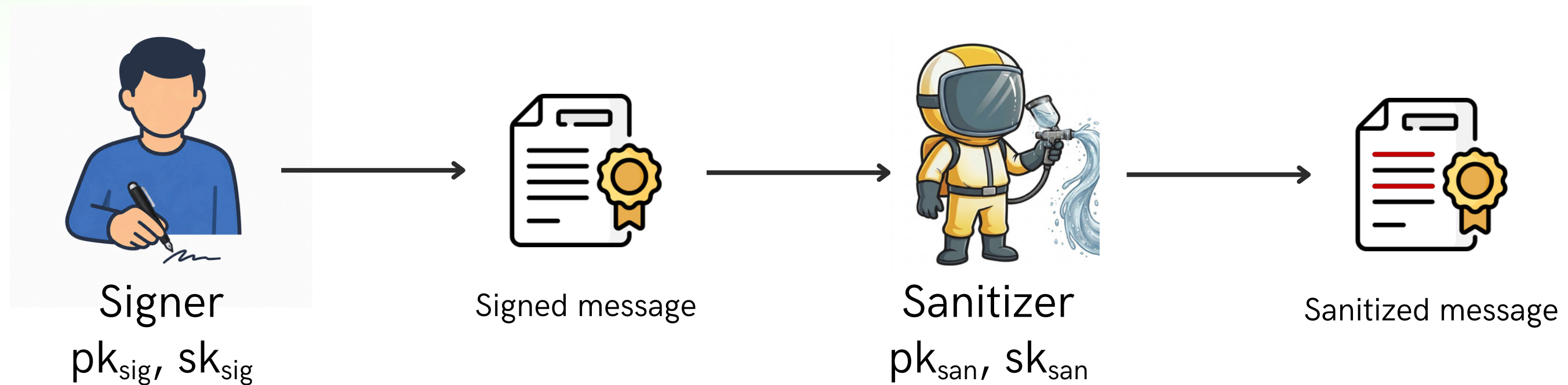
Roberta Cimorelli Belfiore, and Anna Lisa Ferrara

University of Molise, Italy

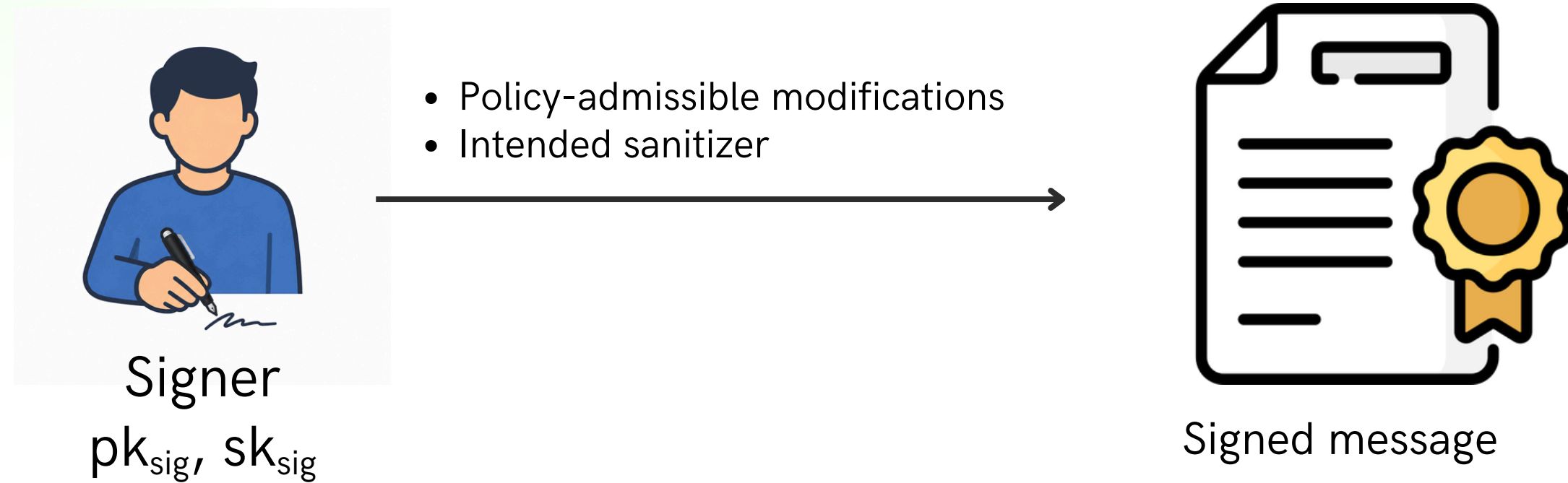
Sanitizable Signatures



Sanitizable Signatures



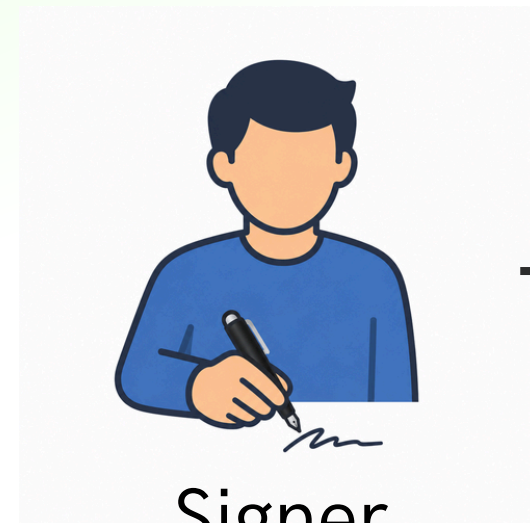
Sanitizable Signatures



Sanitizable Signatures

EXAMPLE POLICY:

selective mutable
block policy



Signer

pk_{sig}, sk_{sig}

- Policy-admissible modifications
- Intended sanitizer



Signed message

Sanitizable Signatures



Sanitizer
 pk_{san}, sk_{san}

- Modifying instruction
- Sanitizer secret key



Sanitized message

Only **intended** parts can be modified.

Sanitizable Signatures Applications



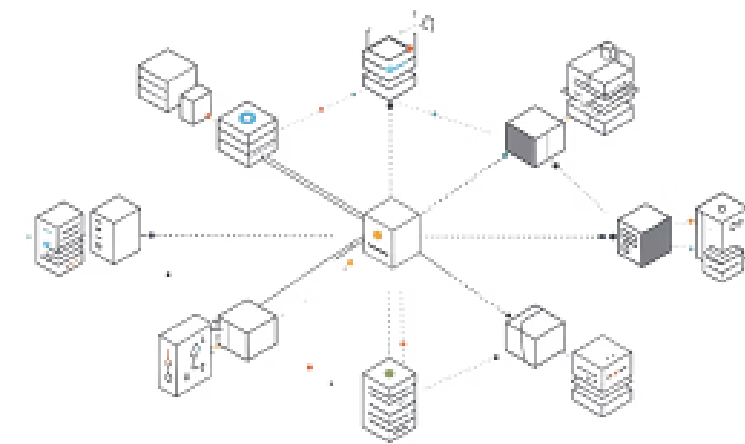
Medical Records

Patient names and private health data embedded in signed documents.



Court Filing

Confidential sources named in legally binding submissions.

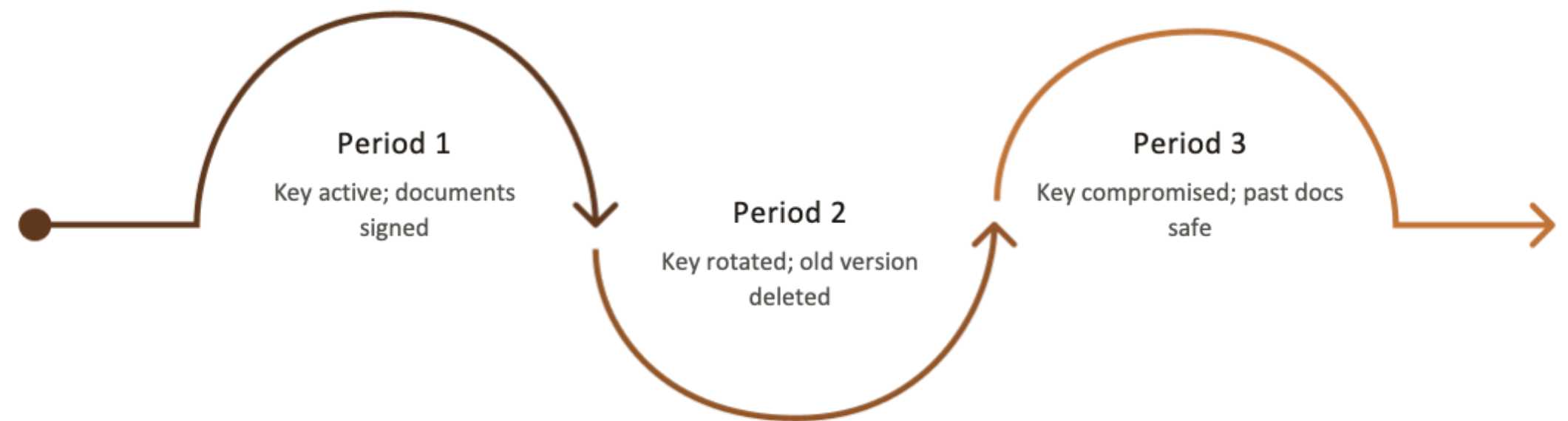


Blockchain Transactions

Personal data written immutably into distributed ledgers.

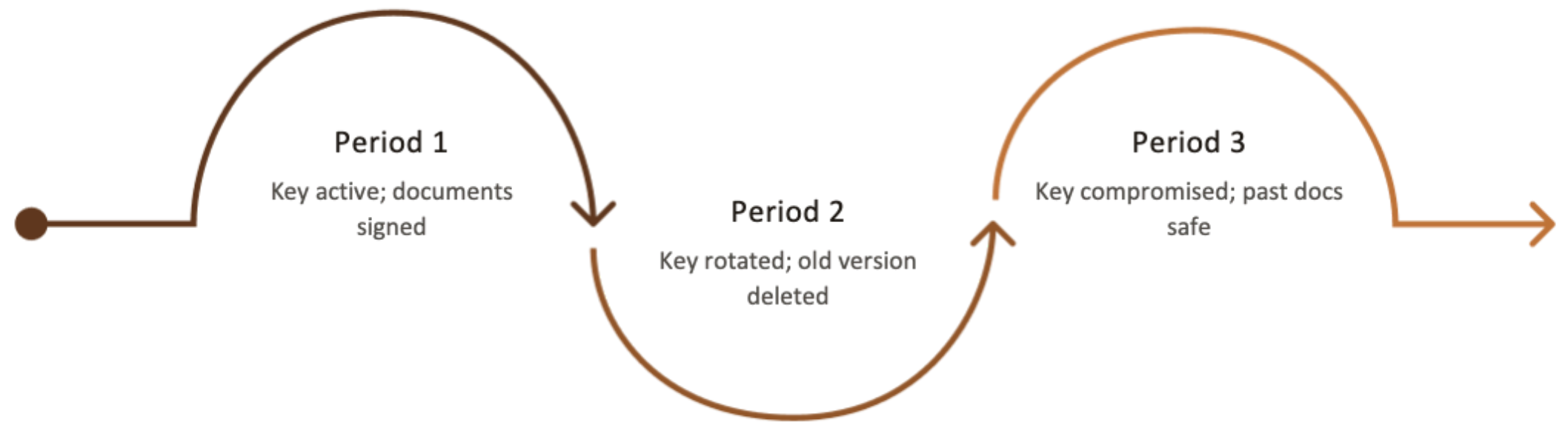
Forward Security

- Key evolves period by period.
- Old key versions are **deleted** after each update.
- Compromise only affects future documents, not past ones.



Forward Security

- Key evolves period by period.
- Old key versions are **deleted** after each update.
- Compromise only affects future documents, not past ones.



Applied to signers only
never to sanitizers.

An Unrealistic Assumption

The sanitizer's secret key stays secure forever.

In practice, this is not realistic.



Sanitizer
 sk_{san}



One stolen key lets an attacker **forge edits** on documents from years ago.



The core tension

Documents signed now may be sanitized **weeks or months later**.
This tension is unique to sanitizers, and **previously unaddressed**.



The core tension

Documents signed now may be sanitized **weeks or months later**.
This tension is unique to sanitizers, and **previously unaddressed**.

Delete the Old Key?

Can't edit older documents, even when allowed



The core tension

Documents signed now may be sanitized **weeks or months later**.
This tension is unique to sanitizers, and **previously unaddressed**.

Delete the Old Key?

Can't edit older documents, even when allowed.

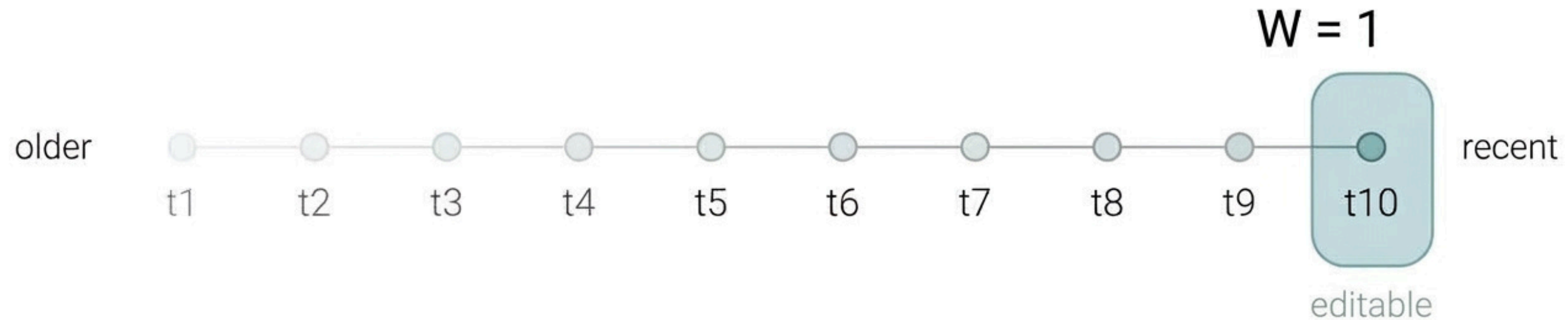
Keep the Old Key?

Future theft still compromises historical edits.

Our Solution: **The Window W**

The idea is that a sanitizer can modify a document only if it was signed within the last W periods.

Every application has its **requirements**: we let the editing capability adapt to the needs of the application.



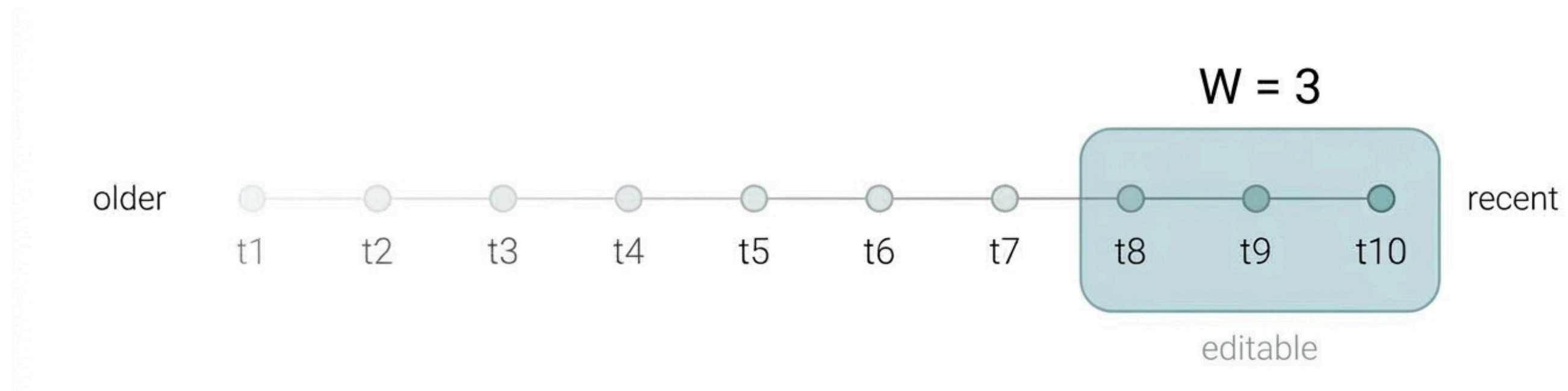
Messaging

$W = 1$



Immediate edits only
strongest protection

Our Solution: **The Window W**



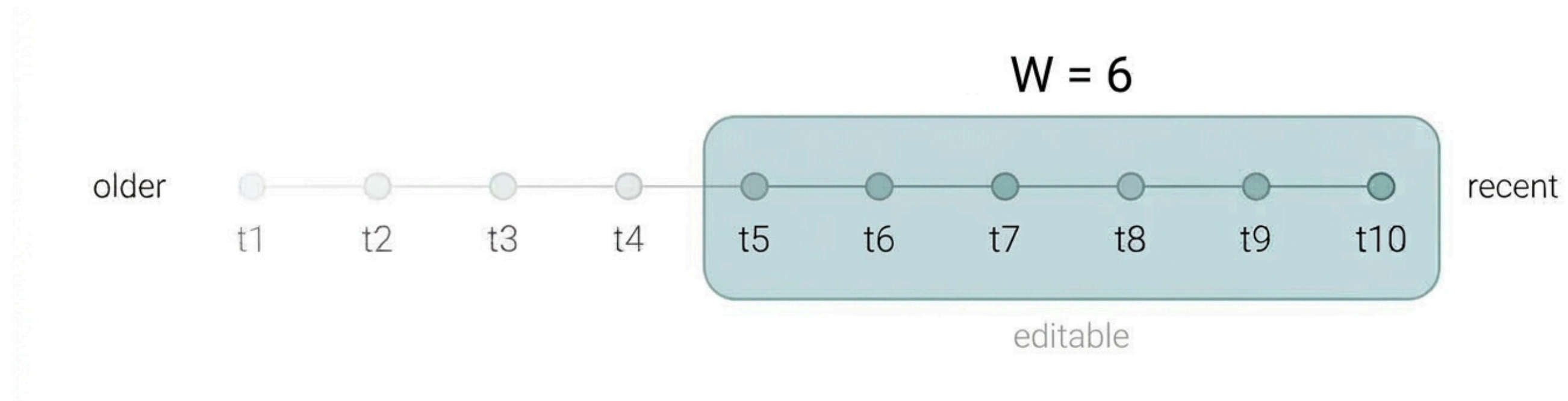
**News/Content
moderation**

$W = 3$



Edits allowed for a few periods
balanced trade-off

Our Solution: **The Window W**



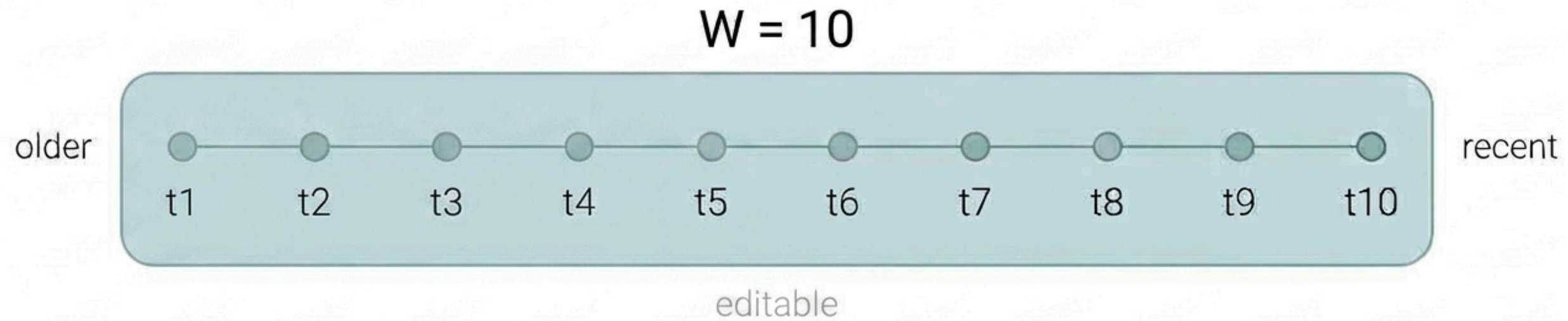
**Collaborative
platform**

$W = 6$



**Longer editing window
more flexibility**

Our Solution: **The Window W**

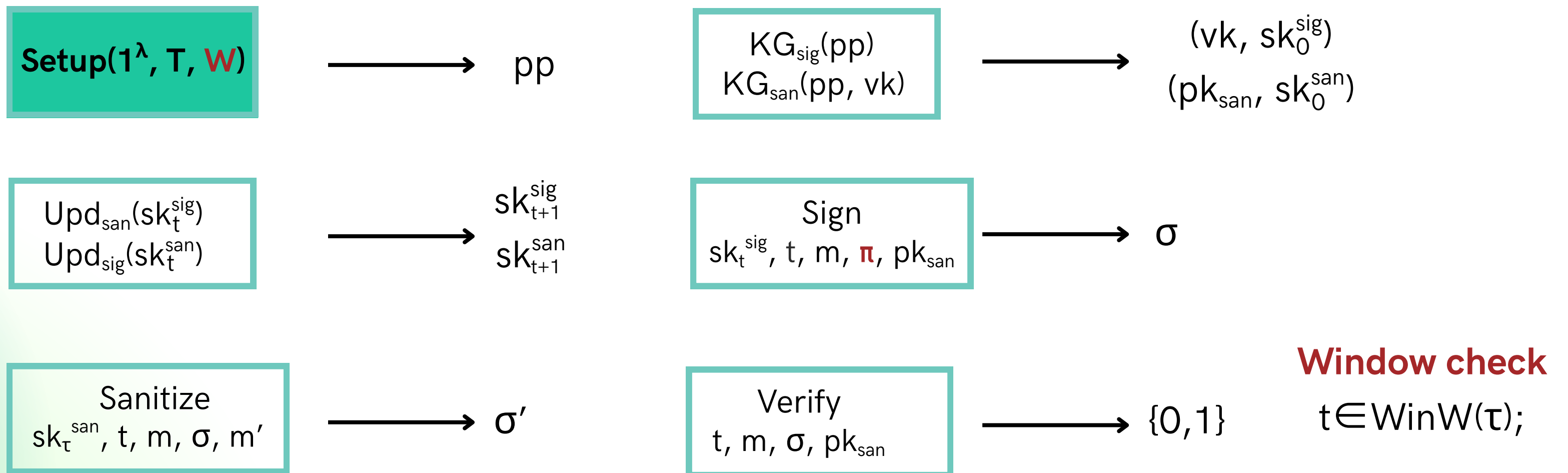


$W = 10 \longrightarrow$ No Forward Security

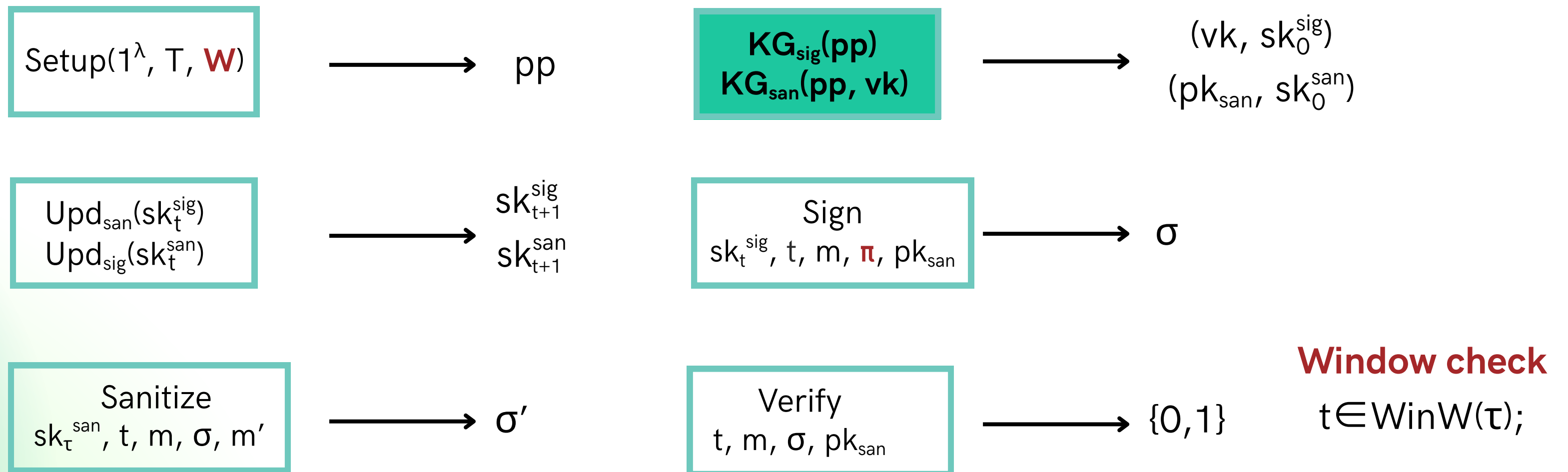
Constructing a Sanitizable
Signature Scheme that is
Dual Forward Secure (DFS-SS)
with also emphasis on **post**
quantum security.

Goal
←

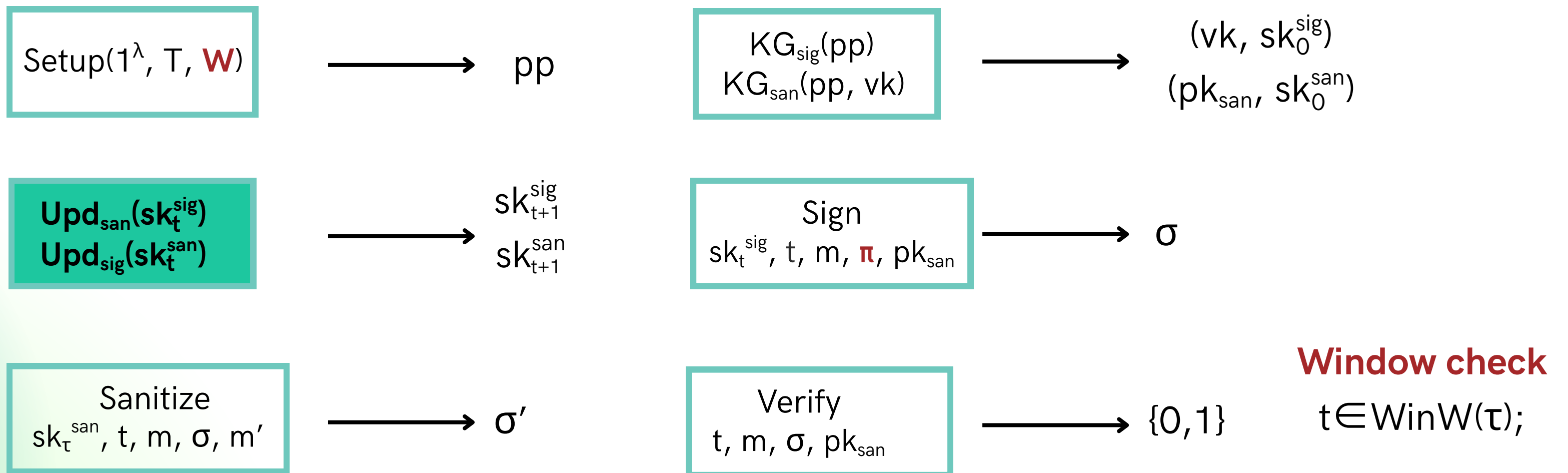
The Model



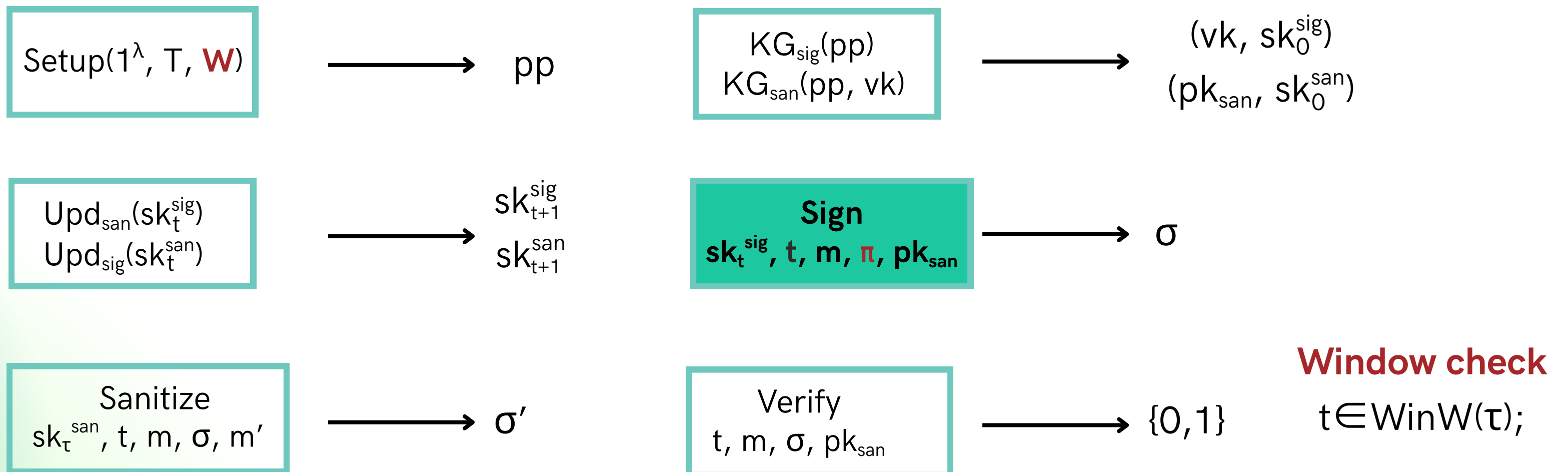
The Model



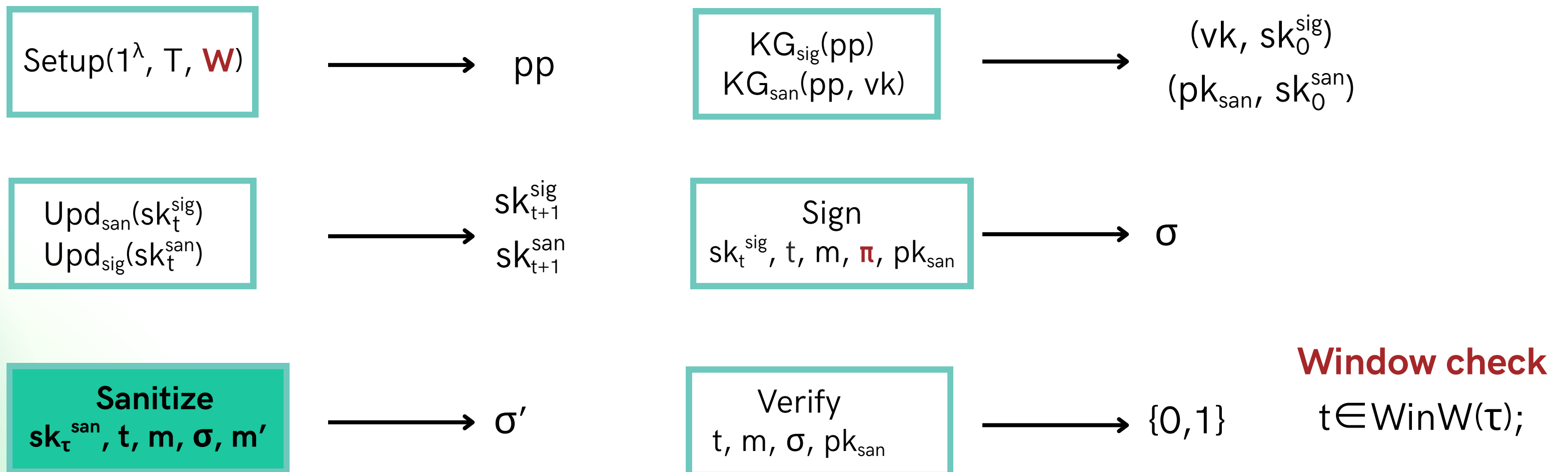
The Model



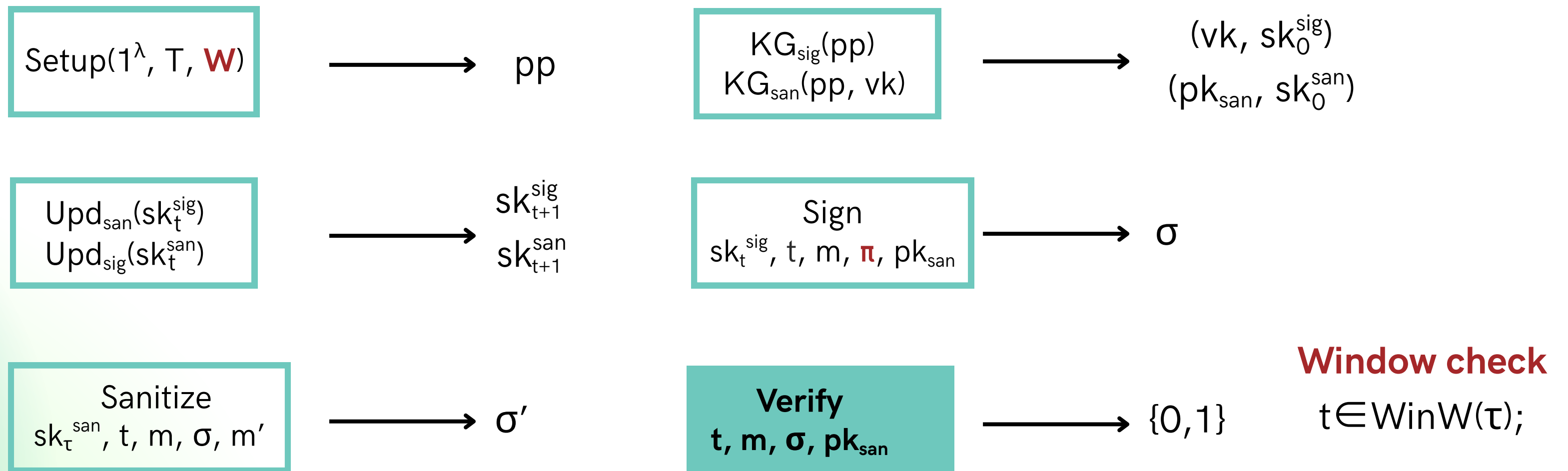
The Model



The Model



The Model



Security Model

Security Model

**WHAT CAN THE
ADVERSARY CORRUPT?**



Security Model

WHAT CAN THE
ADVERSARY CORRUPT?



signer key

sanitizer
key

neither of
them

both keys

Security Model

WHAT CAN THE
ADVERSARY CORRUPT?



WHAT ARE THEY
TRYING TO FORGE?

A **NEW** signature

A **MODIFICATION** of a
previously signed message

Security Model



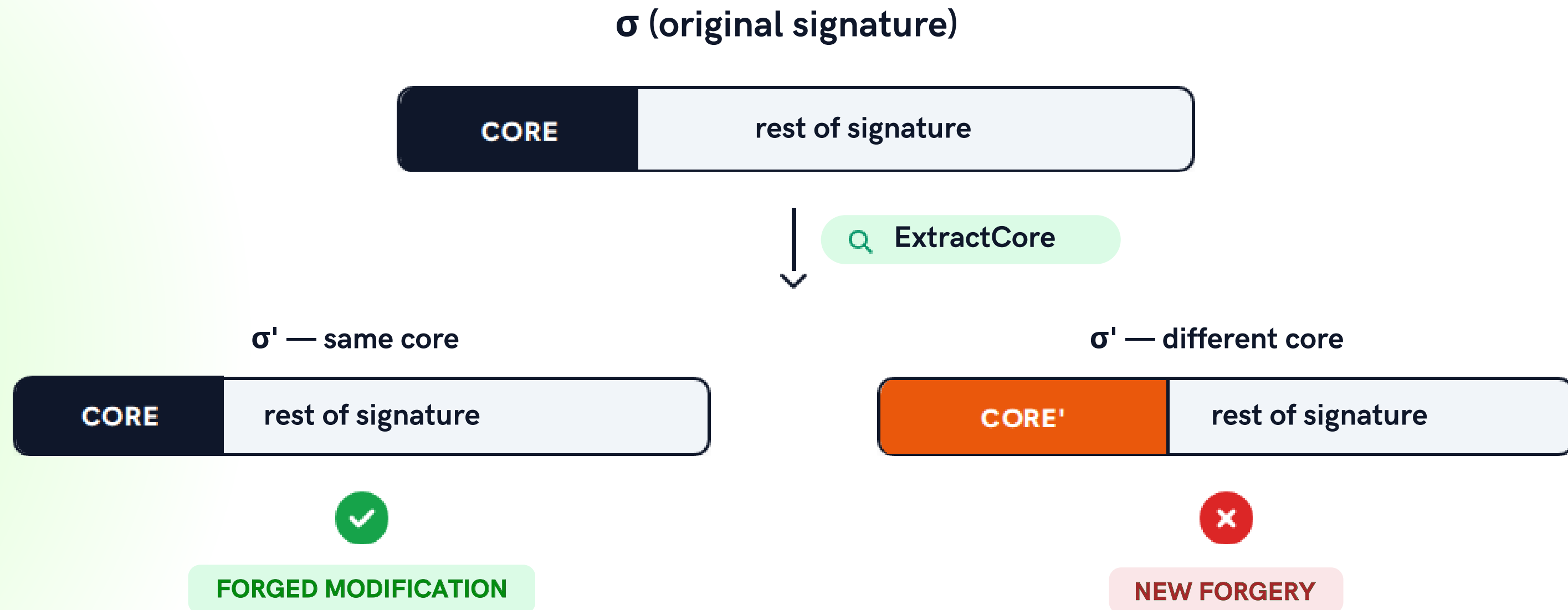
Security Model



- Flag honest sanitizations as forgeries (**false positives**).
- Miss real attacks on the sanitizer entirely (**false negative**).

Security Model

Telling the Cases Apart



Security Properties

EUF-CMA & FS-EUF

EUF-CMA

is the standard unforgeability, no signing key ever exposed.

FS-EUF

captures forward security for the signing functionality. The signing key has been exposed at some period t_{exp} .

ATTENTION!

The adversary may still hold the sanitizer's key, so **policy-reachable messages** don't count as a win.



Security Properties

Immutability & FS-Immutability

Immutability

captures that without the sanitizer key, **no modification should be possible.**

FS-Immutability

same as Immutability, extended to hold even after the signer's key is exposed, for past periods.



Security Properties

FS-SAN(W)

FS-SAN(W)

captures forward security for the sanitization capability. The sanitizer's key has been exposed at some period t_{exp} .

ATTENTION!

The **signer-side core** must be preserved.



Security Properties

Sanitization Soundness

Sanitization Soundness

Even with the sanitizer's full state exposed, a valid modification must still **respect** the policy.

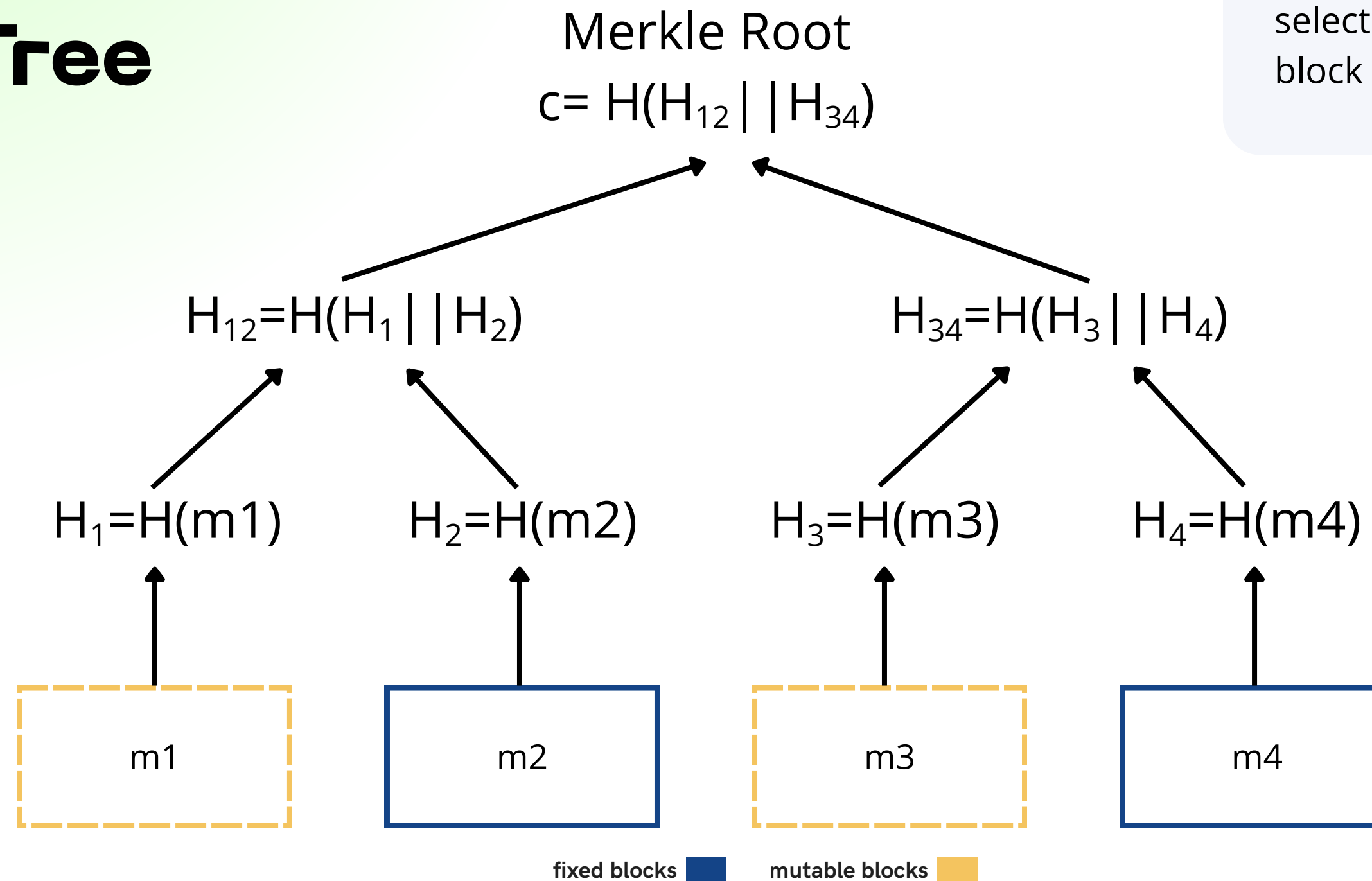


ATTENTION!

This assumes the signer's key stays **honest**. If both keys were compromised, the adversary could just re-sign anything from scratch.

Our Construction

Step 1: Merkle Tree

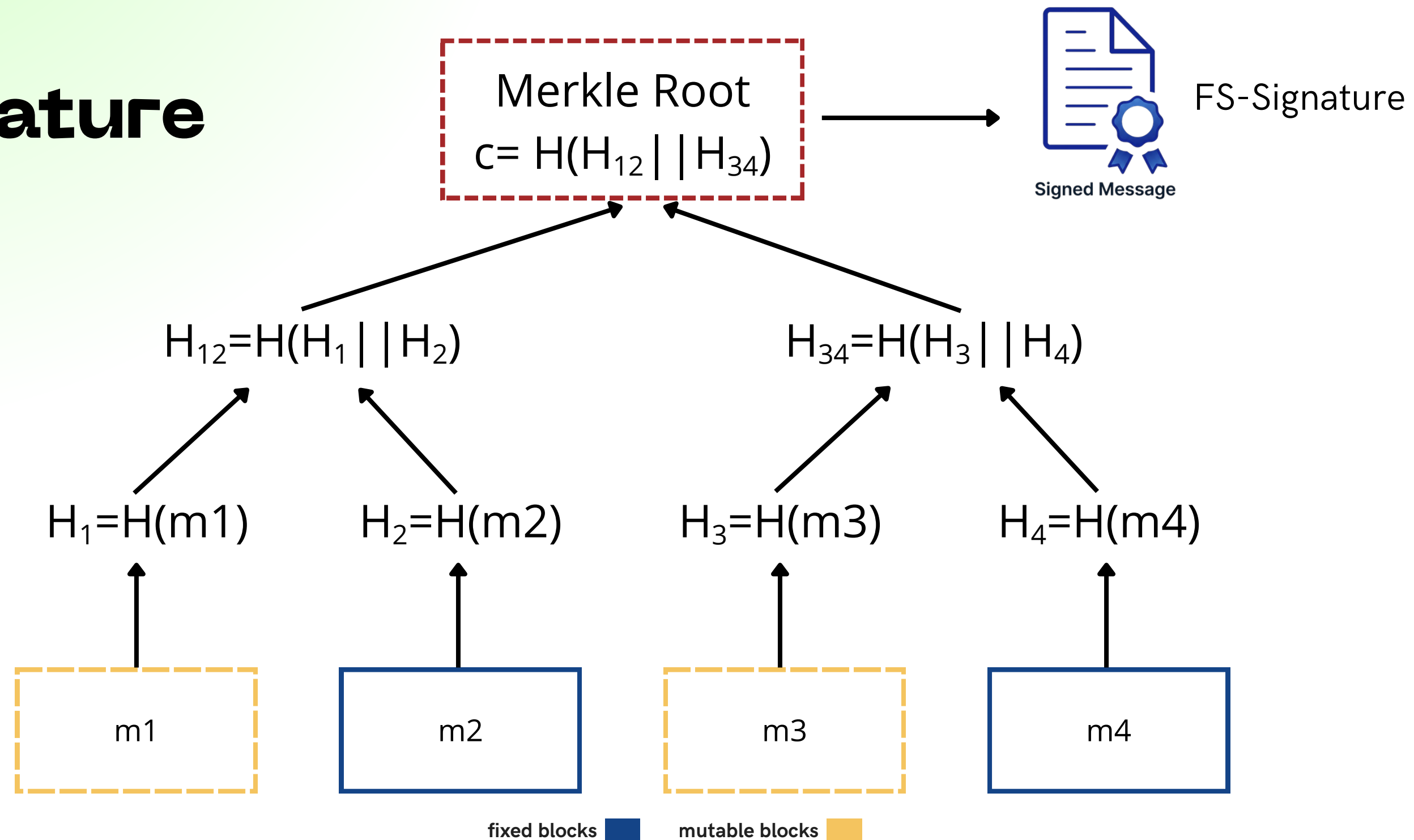


POLICY:

selective mutable
block policy

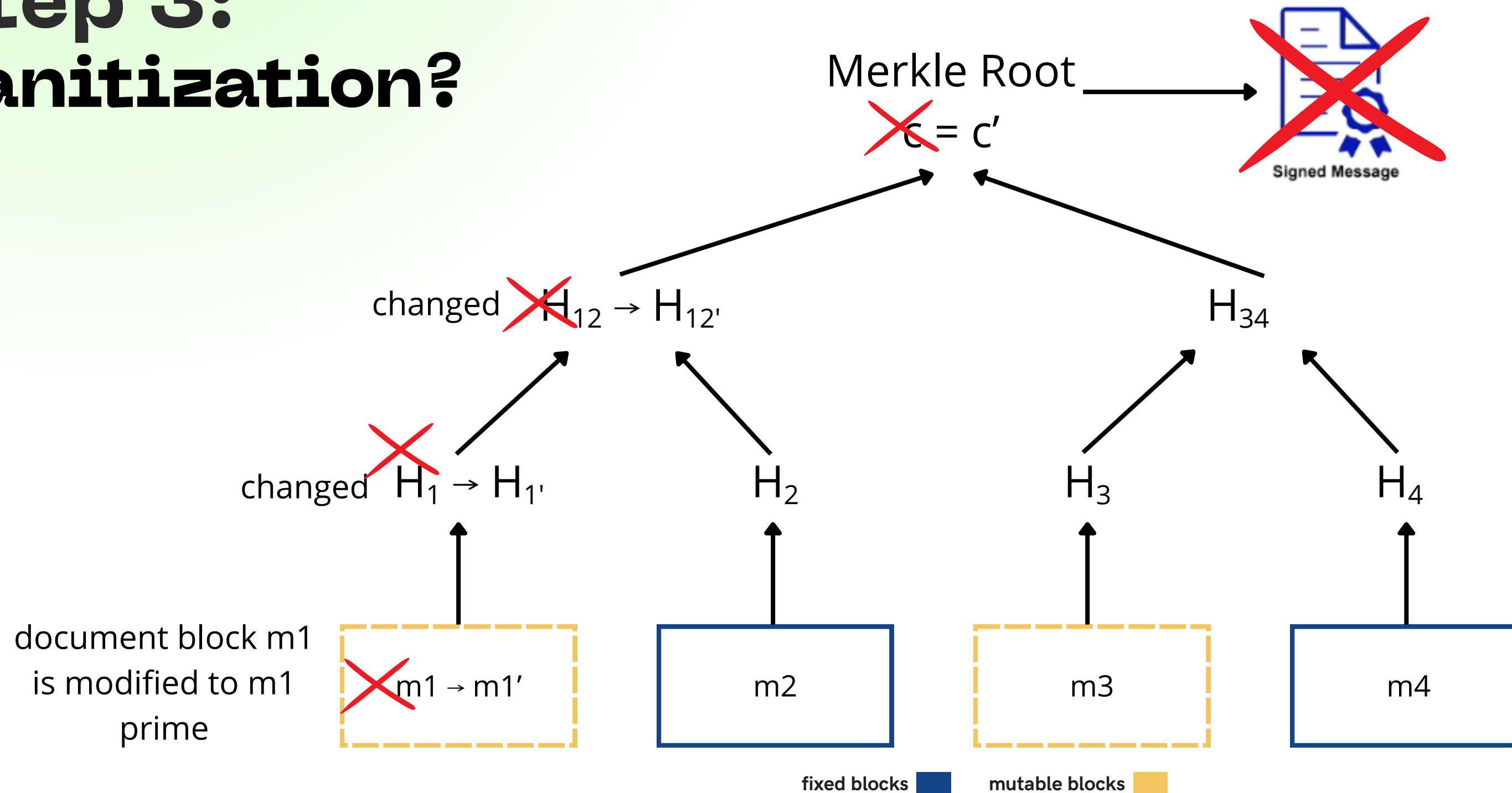
Our Construction

Step 2: FS-Signature



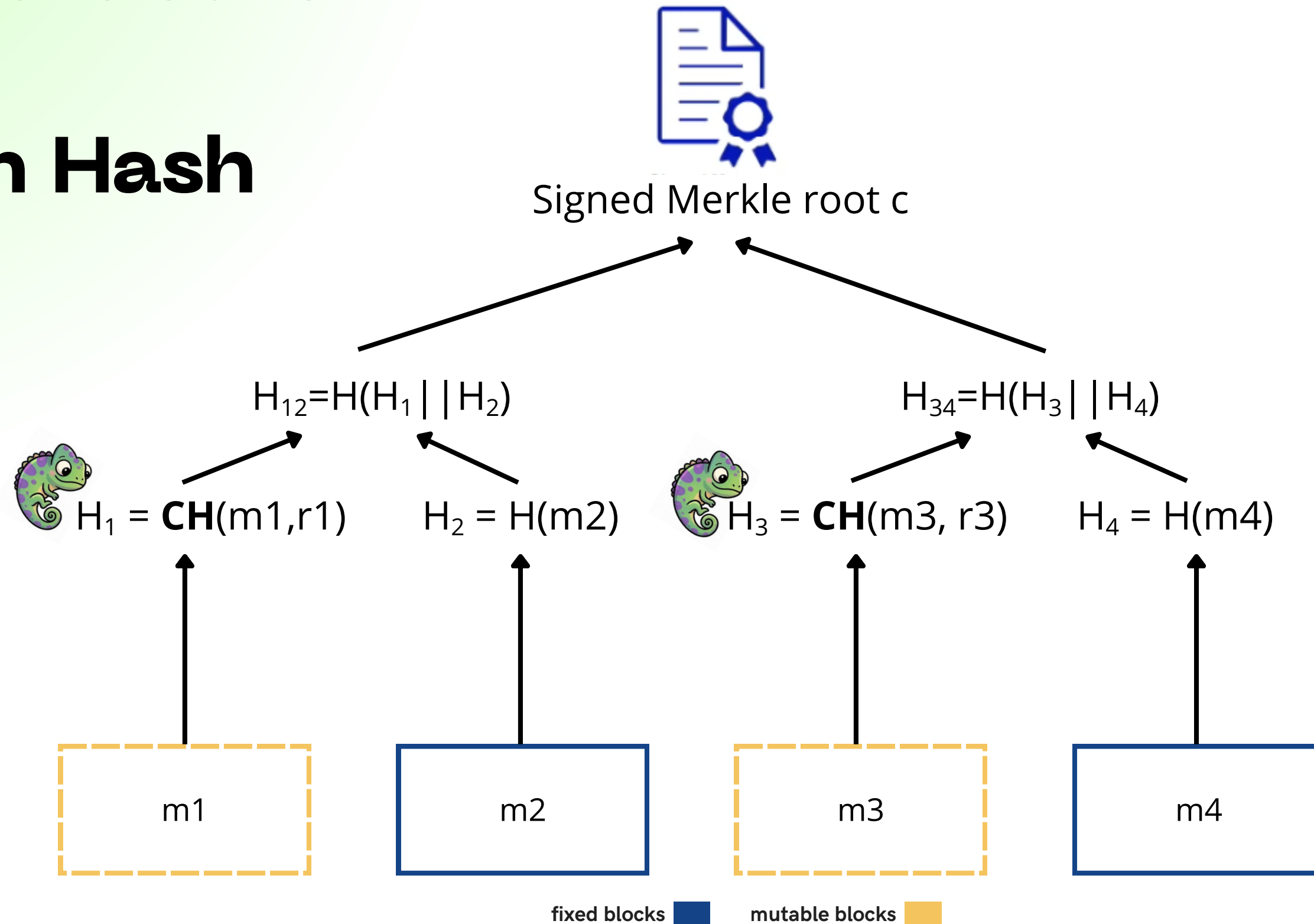
Our Construction

Step 3: Sanitization?

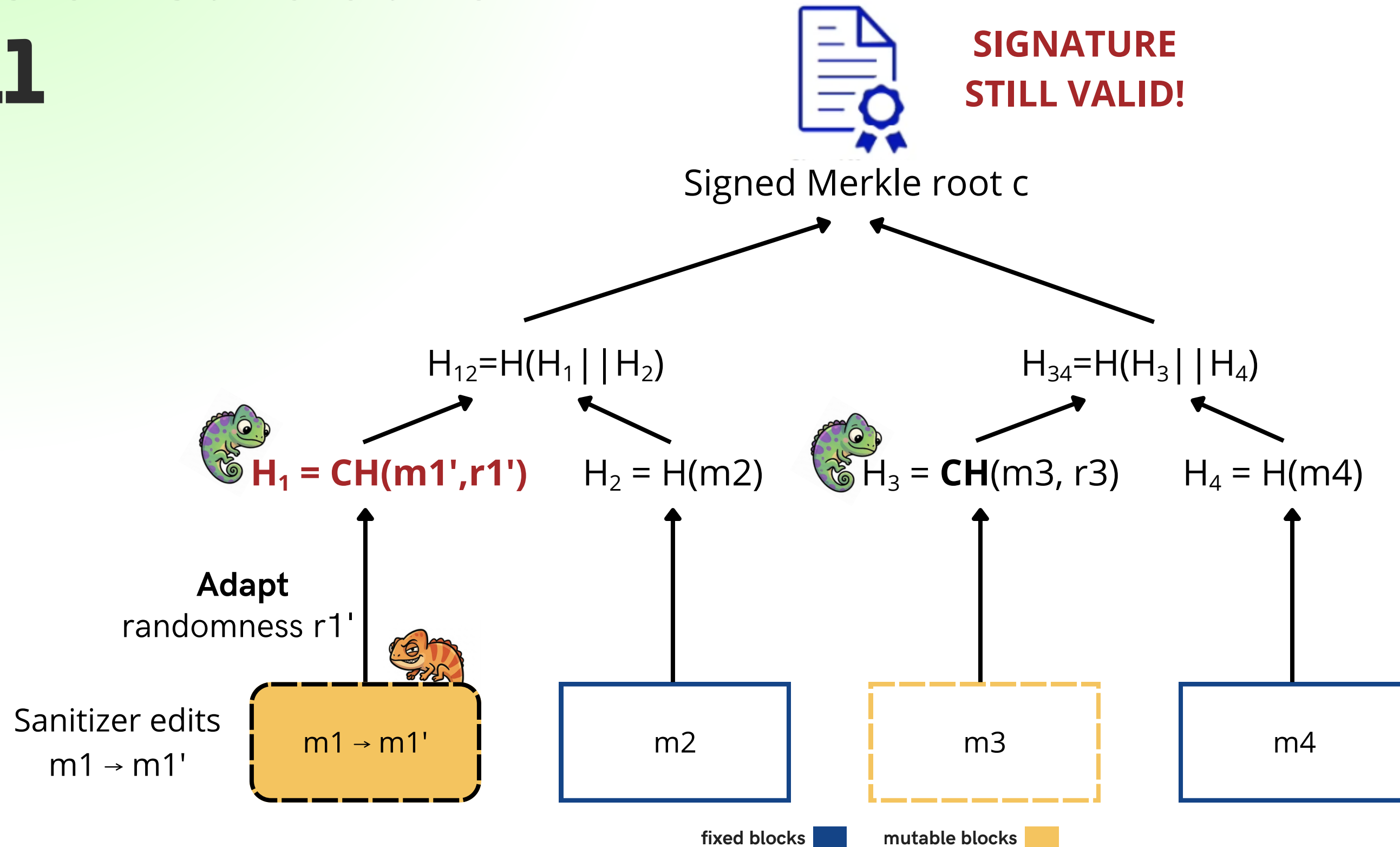


Our Construction

Step 4: Chameleon Hash



Our Construction Final



Our Construction Final



**SIGNATURE
STILL VALID!**

Signed Merkle root c

$$H_{12} = H(H_1 || H_2)$$

$$H_{34} = H(H_3 || H_4)$$



$$H_1 = \mathbf{CH}(m1', r1')$$

$$H_2 = H(m2)$$



$$H_3 = \mathbf{CH}(m3, r3)$$

$$H_4 = H(m4)$$

Adapt
randomness $r1'$

Sanitizer edits
 $m1 \rightarrow m1'$

$m1 \rightarrow m1'$

m2

m3

m4

fixed blocks



mutable blocks



NOTE:
 $CH(m1', r1')$
=
 $CH(m1, r1)$

Security of Our Construction

EUF-CMA & FS-EUF

Mainly relies on unforgeability of the underlying signature scheme.

Immutability & FS-Immutability

Collision Resistance of the Merkle Tree H .

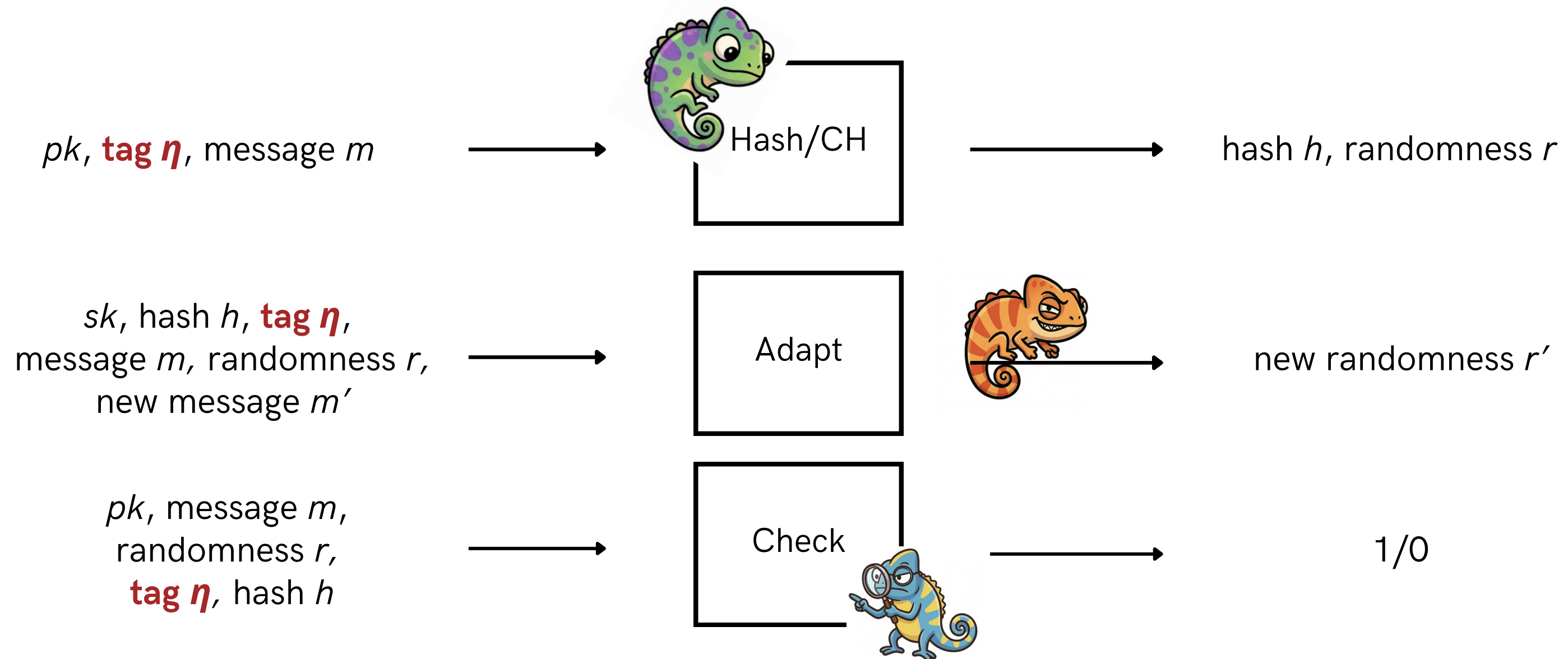
Sanitization Soundness

Mainly relies on security of the underlying signature scheme.

FS-SAN(W)

?

Tagged Chameleon Hash (TCH)



The tag acts as a **contextual binding** parameter that ties the chameleon hash to a specific signing period, policy, and block index, ensuring that modifications are valid only within the intended scope.

Forward Secure TCH

- An FS-TCH provides an **evolving** trapdoor to compute hash collisions within the current window.
- Past collisions become **infeasible** after key evolution and erasure.
- New security notion FS-COLL(W).



Hash/CH



Adapt

Check



Update



Security of Our Construction

EUF-CMA & FS-EUF

Mainly relies on unforgeability of the underlying signature scheme.

Immutability & FS-Immutability

Collision Resistance of the Merkle Tree H .

Sanitization Soundness

Mainly relies on security of the underlying signature scheme.

FS-SAN(W)

Follows from FS-COLL(W) of the FS-TCH primitive.

Instantiations

FORWARD SECURE
SIGNATURE SCHEME

XMSS-SHA2

HASH

SHAKE-256

FS-TCH

Tag structure from
Li-Liu (PKC 2024)



Our Lattice
based FS-TCH

Instantiations

FORWARD SECURE
SIGNATURE SCHEME

XMSS-SHA2

3.28MB

Signature Size

HASH

SHAKE-256

128-bit

Post-Quantum Security

FS-TCH

Tag structure from
Li-Liu (PKC 2024)



Our Lattice
based FS-TCH

10

Mutable Blocks

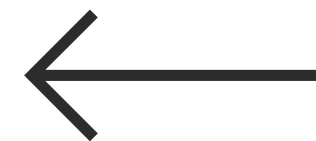


Reduce signature size and computational overhead.

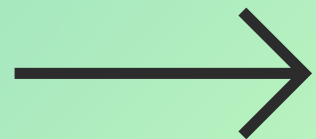


Extend dual forward security to other properties such as unlinkability, transparency and accountability.

Future works



Conclusions



Generic construction
of a DFS-SS scheme with:

- a **tunable** window W
- **ad-hoc** security model
- **post quantum** security

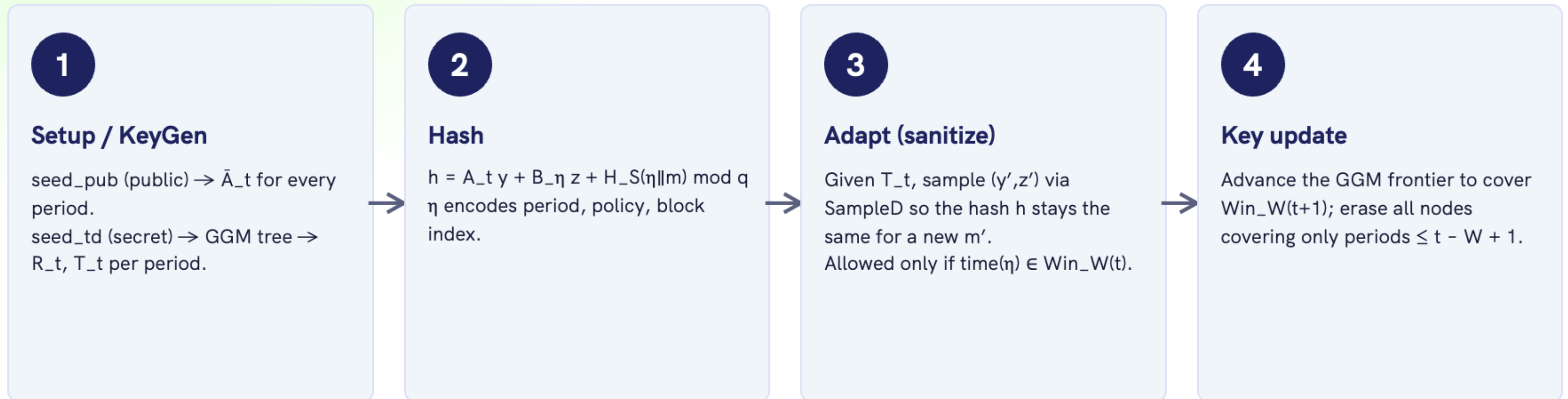
Thanks for your attention



Roberta Cimorelli Belfiore, and Anna Lisa Ferrara

University of Molise, Italy

Lattice-based FS-TCH



Why it stays valid: Adapt only changes the randomness (y, z) , never the hash value h , so the Merkle root and the signer's XMSS signature σ_Σ never change. No re-signing needed.

Lattice-based FS-TCH

Key and state sizes

Component	Expression	Size
vk	XMSS vk	64 B
pk_san	seed_pub + root_A2 + vk	144 B
σ_Σ	XMSS signature	2820 B
h_i	$k \cdot d \cdot \ell_q$ bits	2400 B
(y_i, z_i) per block	$(m_R + k \ell_q) \cdot d \cdot 20$ bits	≈ 328 KB
$A_t^{(2)}$	$k^2 \ell_q^2 d$ bits	≈ 176 KB
auth_t	$\lceil \log_2 T \rceil \times 48$ B	960 B ($T=2^{20}$)

Total signature size

$ \pi = 10$ blocks	Size
Including $A_t^{(2)}$	≈ 3.46 MB
Excluding $A_t^{(2)}$ (verifier has D)	≈ 3.28 MB

Lattice-based FS-TCH

Comparison with related schemes ($|\pi|=10$, $W=10$)

Scheme	$ \sigma $	FS-Sig	FS-San	PQ
Ateniese et al. [6]	$O(\pi)$	—	—	—
Clermont et al. [13]	≈ 50 KB	—	—	✓
Li-Liu [23]	$O(\pi)$	—	—	✓
This work	≈ 3.28 MB	✓	✓	✓