Algorithm $\mathsf{RandWrap}_{sk,t_1,t_2}(x) \to x'$

A randomness wrapper based on RFC8937 [CGS$^+$20] that ties the security of a CSPRNG to a signing key, parametrized by $N = L = L' = \kappa$, and requiring:

- •A signature scheme $\mathsf{Sign}(sk, m) \to \sigma$ (e.g., Section **??**) and a private key $sk \in \mathbb{G}$.
- •A hash function H (e.g., SHA3 [Dwo15])
- •A key derivation function $\mathsf{KDF}(salt, m) \to k \in \mathbb{Z}_2^L$ (e.g., HKDF-Extract[KE10])
- •A pseudo-rand. function $\mathsf{PRF}(k, info) \to x' \in \mathbb{Z}_2^N$ (e.g., HKDF-Expand[KE10])
- •t_1, a context-dependent bit-string (e.g., device MAC, OS version…)
- •$t_2 \in \mathbb{Z}_2^{L'}$, a unique nonce per PrngWrapper call (e.g., a counter)

Inputs: $x \in \mathbb{Z}_2^\kappa$, a seed (default: use the OS randomness API)
Outputs: $x' \in \mathbb{Z}_2^\kappa$, a seed to be consumed by a CSPRNG.

```
$h_\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{PrngWrapper.t_1})$          (Can be precomputed and stored)
$k \leftarrow \mathsf{KDF}(h_\sigma, x)$
$x' \leftarrow \mathsf{PRF}(k, \mathsf{PrngWrapper}.t_2)$
Increase $\mathsf{PrngWrapper}.t_2$ by one for next calls
return $x'$
```

References

[CGS$^+$20] C Cremers, L Garratt, S Smyshlyaev, N Sullivan, and C Wood. Rfc 8937: Randomness improvements for security protocols, 2020.

[Dwo15] Morris Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions, 2015-08-04 2015.

[KE10] Dr. Hugo Krawczyk and Pasi Eronen. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869, May 2010.