$\mathsf{Kg}(1^k)$	$\mathsf{Sign}(sk, m)$	$\mathsf{Vfy}(pk, m, \sigma)$
$sk \leftarrow \mathbb{Z}_p$	$r \xleftarrow{\$} \mathbb{Z}_p;\ I \leftarrow g^r$	$R \leftarrow g^s \cdot pk^{-e}$
$pk \leftarrow g^{sk}$	$e \leftarrow \mathsf{H}(pk \| I \| m)$	**if** $\mathsf{H}(pk \| R \| m) = e$
return (pk, sk)	$s \leftarrow r + sk \cdot e \mod p$	**return** 1
	return $\sigma = (s, e)$	**else return** 0

▷ **Answer 2:** Yes, **key-prefixed** short Schnorr signatures are secure!

Summary of Our Results

Research Questions

07

Are **short** Schnorr signatures secure against **preprocessing attacks**?

$4k$ bits

Signature Length:

Schnorr

Short Schnorr

$3k + \log S$ bits (with preprocessing)

$3k$ bits (without preprocessing)

size of hint

e.g., if $S = 2^{k/2}$ then we have a $3.5k$-bit signature

Summary of Our Results

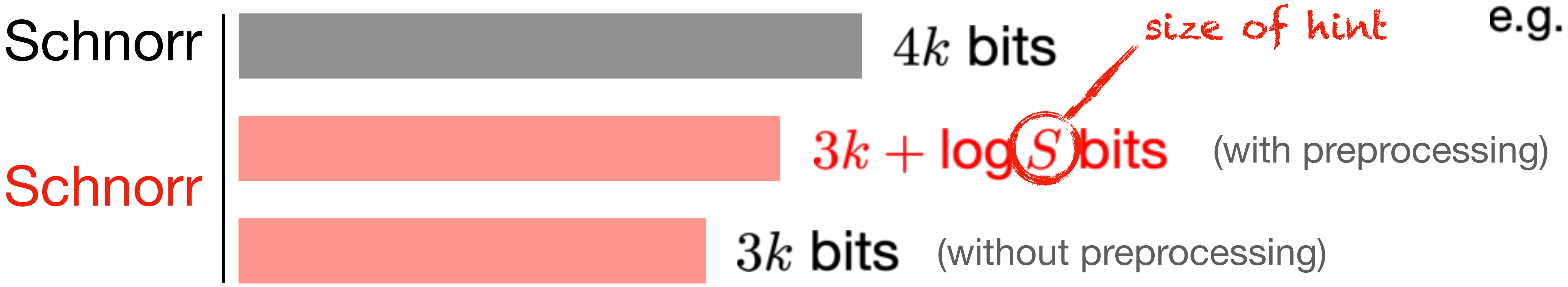
Research Questions

Are **short** Schnorr signatures secure against **preprocessing attacks**?

▷ **Answer 2:** Yes, **key-prefixed** short Schnorr signatures are secure!

$\mathsf{Kg}(1^k)$	$\mathsf{Sign}(sk, m)$	$\mathsf{Vfy}(pk, m, \sigma)$
1 : $sk \leftarrow \mathbb{Z}_p$	1 : $r \xleftarrow{\$} \mathbb{Z}_p;\ I \leftarrow g^r$	1 : $R \leftarrow g^s \cdot pk^{-e}$
2 : $pk \leftarrow g^{sk}$	2 : $e \leftarrow \mathsf{H}(pk \| I \| m)$	2 : **if** $\mathsf{H}(pk \| R \| m) = e$
3 : **return** (pk, sk)	3 : $s \leftarrow r + sk \cdot e \mod p$	3 : **return** 1
	4 : **return** $\sigma = (s, e)$	4 : **else return** 0

Signature Length:



e.g., if $S = 2^{k/2}$ then we have a $3.5k$-bit signature

Summary of Our Results

Research Questions

Are **short** Schnorr signatures secure against **preprocessing attacks**?

▷ **Answer 2:** Yes, **key-prefixed** short Schnorr signatures are secure!

$\mathsf{Kg}(1^k)$	$\mathsf{Sign}(sk, m)$	$\mathsf{Vfy}(pk, m, \sigma)$
1 : $sk \leftarrow \mathbb{Z}_p$	1 : $r \xleftarrow{\$} \mathbb{Z}_p$; $I \leftarrow g^r$	1 : $R \leftarrow g^s \cdot pk^{-e}$
2 : $pk \leftarrow g^{sk}$	2 : $e \leftarrow \mathsf{H}(pk \| I \| m)$	2 : **if** $\mathsf{H}(pk \| R \| m) = e$
3 : **return** (pk, sk)	3 : $s \leftarrow r + sk \cdot e \mod p$	3 : **return** 1
	4 : **return** $\sigma = (s, e)$	4 : **else return** 0

Caveats:

- Not a **standardized** implementation
- Preprocessing attacker is **time-bounded** (large enough for practical attacks)
- Complex proof technique: **compression argument**