New Constructions of Statistical NIZKs: Dual-Mode DV-NIZKs and More

Benoît Libert, Alain Passelègue, Hoeteck Wee, and David J. Wu

May 2020

Non-Interactive Zero-Knowledge (NIZK)

[BFM88]

$x \in \{0,1\}^*$

NP language $\mathcal{L} \subseteq \{0,1\}^*$

accept if $x \in \mathcal{L}$

prover

$$\xrightarrow{\pi}$$

verifier

Completeness: $\forall x \in \mathcal{L} : \Pr[\langle P, V\rangle(x) = \text{accept}] = 1$

"Honest prover convinces honest verifier of true statements"

Soundness: $\forall x \notin \mathcal{L},\ \forall P^* : \Pr[\langle P^*, V\rangle(x) = \text{accept}] \leq \varepsilon$

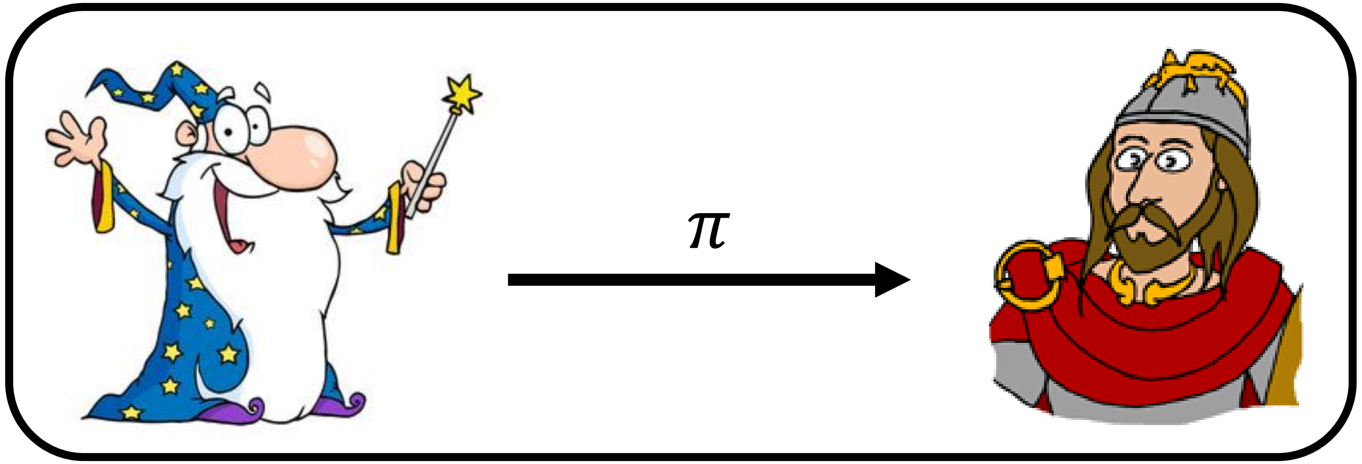
"No prover can convince honest verifier of false statement"

can consider both <u>computational</u> and <u>statistical</u> variants

Non-Interactive Zero-Knowledge (NIZK)

[BFM88]

NP language $\mathcal{L}$



$\approx_c$

real distribution

ideal distribution

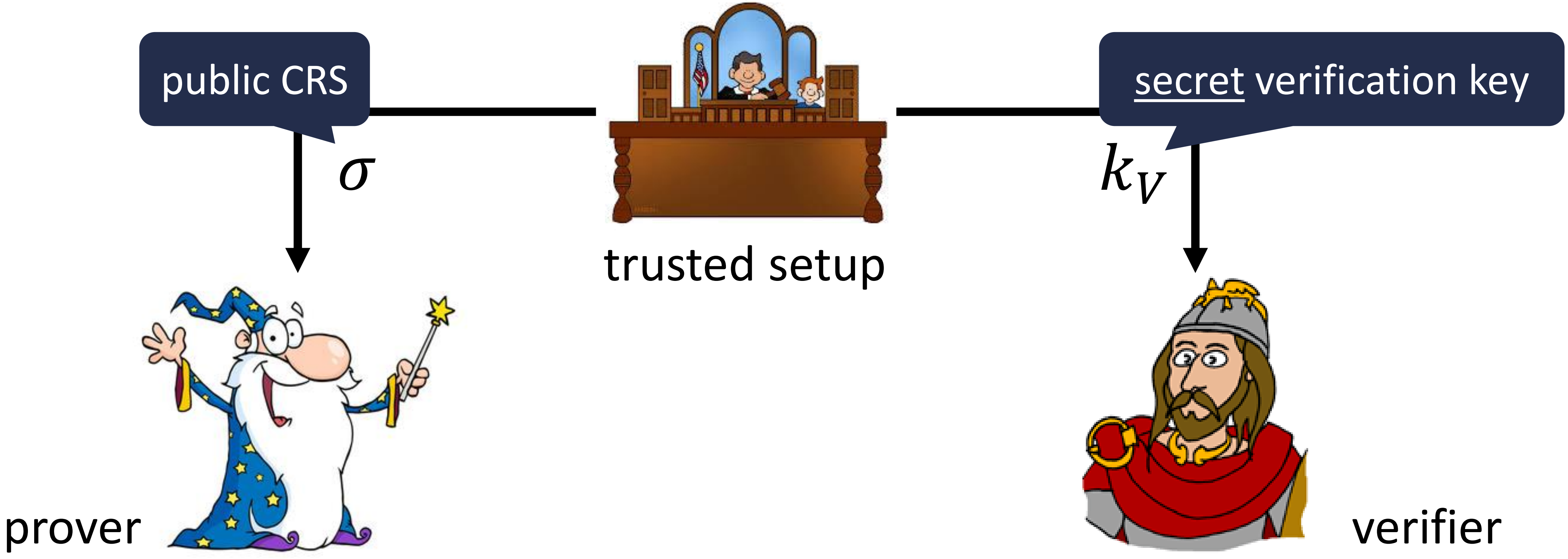
Zero-Knowledge: for all efficient verifiers V^*, there exists an efficient simulator $\mathcal{S}$ where

$$\forall x \in \mathcal{L} : \langle P, V^* \rangle(x) \approx \mathcal{S}(x)$$

can consider both <u>computational</u> and <u>statistical</u> variants

Designated-Verifier NIZKs

This work: focus primarily on the designated-verifier model



Designated-Verifier NIZKs

This work: focus primarily on the designated-verifier model

σ

Requirement: soundness should hold even if the prover has access to the verification oracle

k_V

$$\pi = \text{Prove}(\sigma, x, w)$$

prover

verifier

The Landscape of (DV)-NIZKs

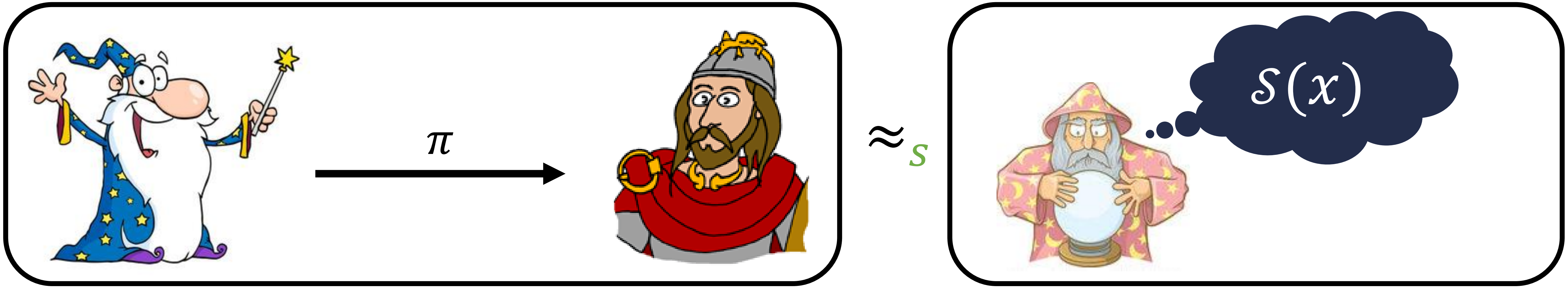
Construction	Soundness	Zero-Knowledge	Assumption
[FLS90]	statistical	computational	factoring
[CHK03]	statistical	computational	CDH (pairing group)
[GOS06]	stat. | comp.	comp. | stat.	k-Lin (pairing group)
[PS19]	stat. | comp.	comp. | stat.	LWE
[SW14]	computational	statistical	iO + OWFs
			publicly-verifiable
[QRW19, CH19, KNYY19]	statistical	computational	CDH
[LQRWW19]	computational	computational	CDH/LWE/LPN
[CDIKLOV19]	stat. | comp.	comp. | stat.	DCR
			malicious designated-verifier

The Landscape of (DV)-NIZKs

Statistical zero-knowledge seems more difficult to achieve

Construction	Soundness	Zero-Knowledge	Assumption
[FLS90]			ing
[CHK03]			g group)
[GOS06]	stat. | comp.	comp. | stat.	k-Lin (pairing group)
[PS19]	stat. | comp.	comp. | stat.	LWE
[SW14]	computational	statistical	iO + OWFs
			publicly-verifiable
[QRW19, CH19, KNYY19]	statistical	computational	CDH
[LQRWW19]	computational	computational	CDH/LWE/LPN
[CDIKLOV19]	stat. | comp.	comp. | stat.	DCR
			malicious designated-verifier

This Work: Statistical NIZKs

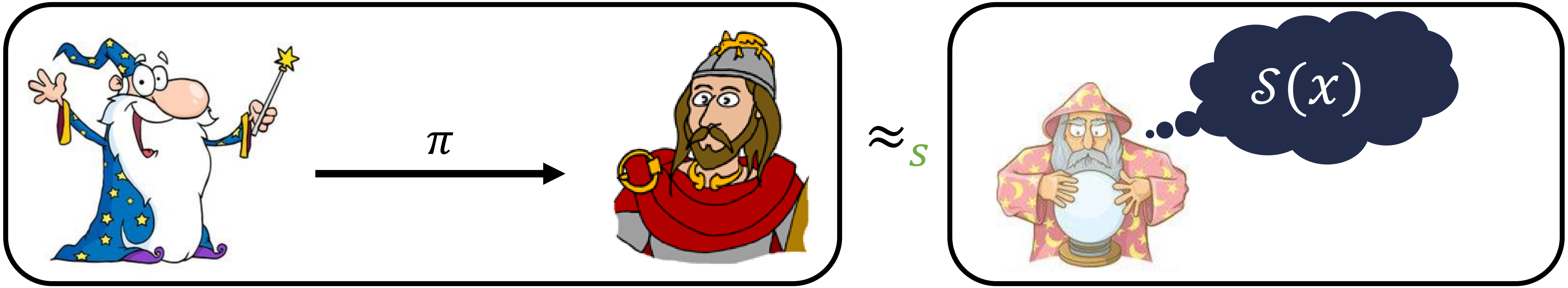


Statistical ZK provides <u>everlasting</u> privacy

This work: Compiling NIZKs in the hidden-bits model to statistical (DV)-NIZKs

- Statistical DV-NIZKs from DDH in pairing-free groups / QR / DCR

This Work: Statistical NIZKs



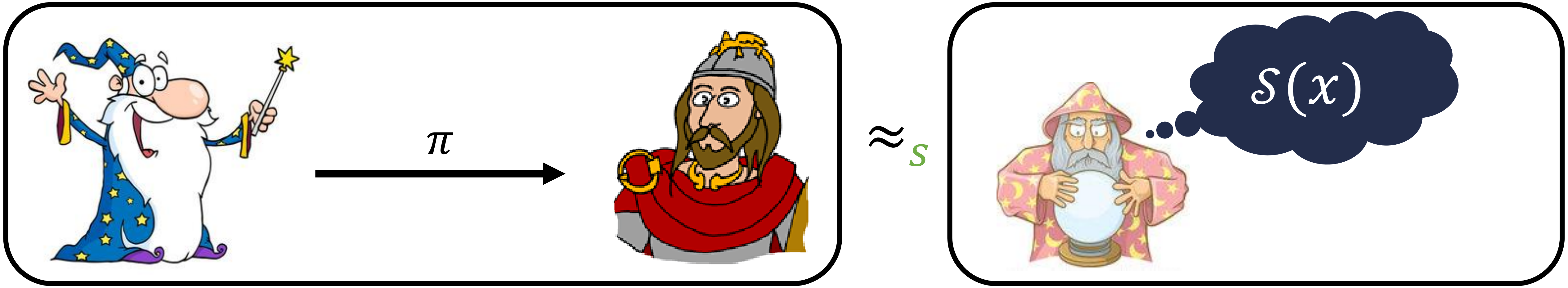
des everlasting privacy

More precisely: DV-NIZKs are "dual-mode" and maliciously secure

This work: Co ling NIZKs in the hidden-bits model to statistical (DV)-NIZKs

- Statistical DV-NIZKs from DDH in pairing-free groups / QR / DCR

This Work: Statistical NIZKs



Weaker assumption compared to [GOS06] which required k-Lin in both groups (k-KerLin is a search assumption implied by k-Lin)

ting privacy

to statistical (DV)-NIZKs

- Statistical DV-NIZKs from … in pairing-free groups / QR / DCR
- Statistical NIZKs from k-Lin ($\mathbb{G}_1$) + k-KerLin ($\mathbb{G}_2$) in a pairing group

The Landscape of (DV)-NIZKs

Construction	Soundness	Zero-Knowledge	Assumption
[FLS90]	statistical	computational	factoring
[CHK03]	statistical	computational	CDH (pairing group)
[GOS06]	stat. | comp.	comp. | stat.	k-Lin $(\mathbb{G}_1, \mathbb{G}_2)$
This work	**computational**	**statistical**	**k-Lin $(\mathbb{G}_1)$, k-KerLin $(\mathbb{G}_2)$**
[PS19]	stat. | comp.	comp. | stat.	LWE
[SW14]	computational	statistical	iO + OWFs
			publicly-verifiable
[QRW19, CH19, KNYY19]	statistical	computational	CDH
[LQRWW19]	computational	computational	CDH/LWE/LPN
[CDIKLOV19]	stat. | comp.	comp. | stat.	DCR
This work	**stat. | comp.**	**comp. | stat.**	**DDH/QR/DCR**
			malicious designated-verifier

NIZKs in the Hidden Bits Model

[FLS90]

$$\overbrace{\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{1}}^{n \text{ bits long}}$$

prover has access to uniformly random bit string of length n

prover

NIZKs in the Hidden Bits Model

[FLS90]

n bits long

0	1	1	1	0	0	1	0	1	1

prover has access to uniformly random bit string of length n

prover

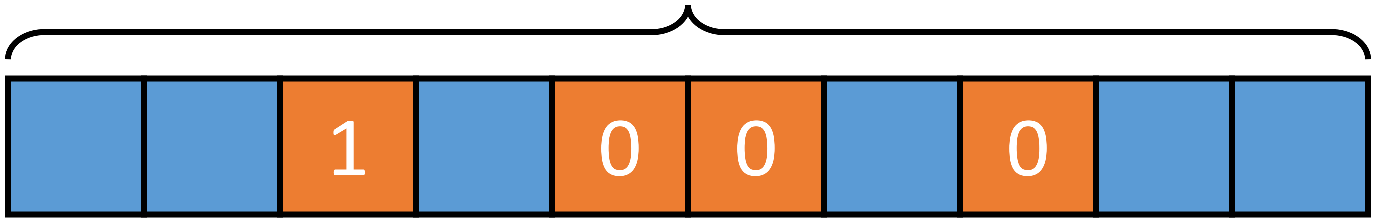
$I \subseteq [n], \pi$

prover outputs a subset $I \subseteq [n]$ and a proof π

NIZKs in the Hidden Bits Model

[FLS90]

n bits long



verifier only sees the subset of the bits in I and proof π

prover

$I \subseteq [n], \pi$

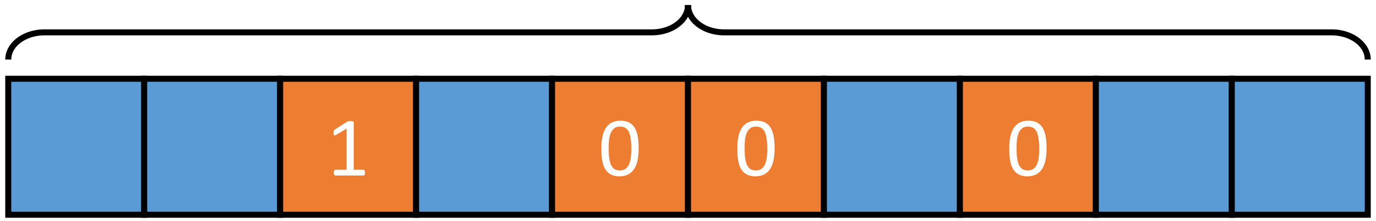
verifier

prover outputs a subset $I \subseteq [n]$ and a proof π

NIZKs in the Hidden Bits Model

[FLS90]

n bits long



verifier only sees the subset of the bits in I and proof π

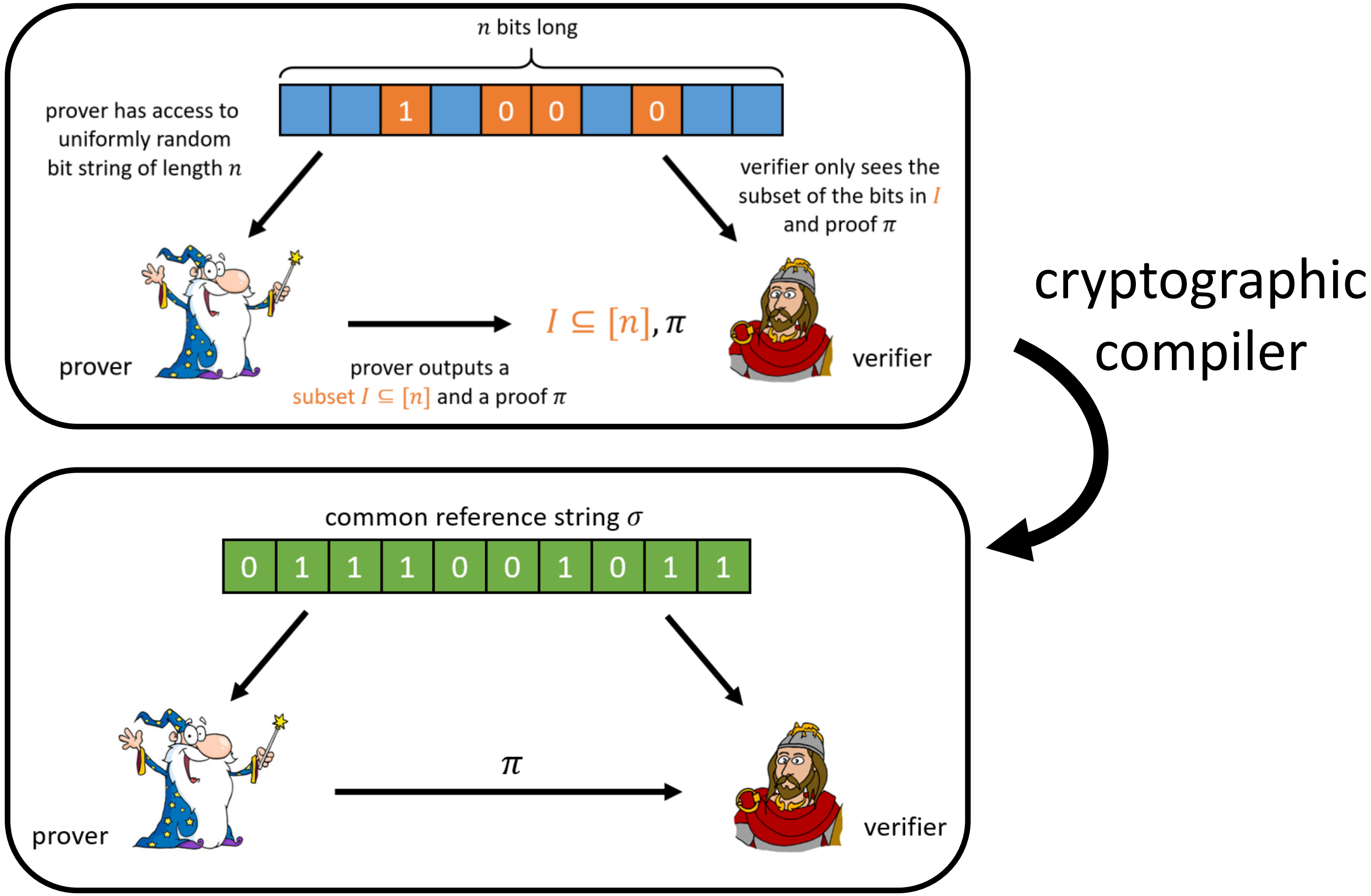
verifier

[FLS90]: There exists a perfect NIZK proof for any NP language in the hidden-bits model

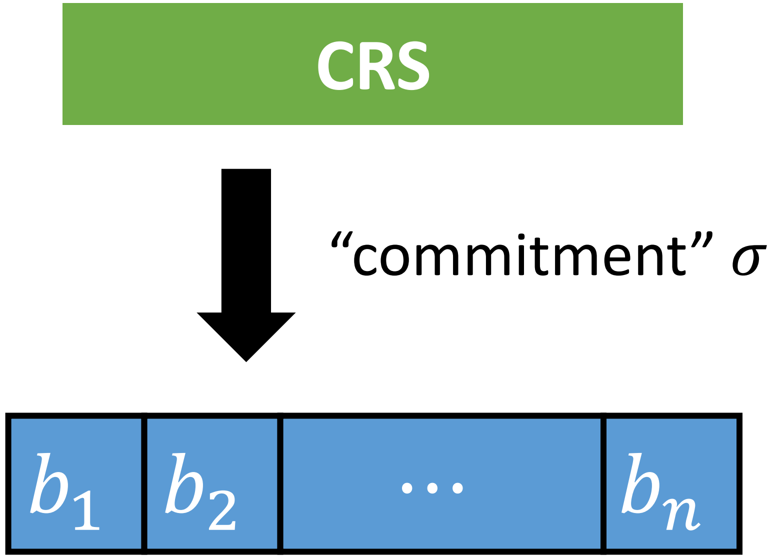
The FLS Compiler

[FLS90]

NIZKs in the hidden-bits model



NIZKs in the CRS model



hidden-bits string

Prover can selectively open σ to (i, b_i) for indices i of its choosing

The FLS Compiler

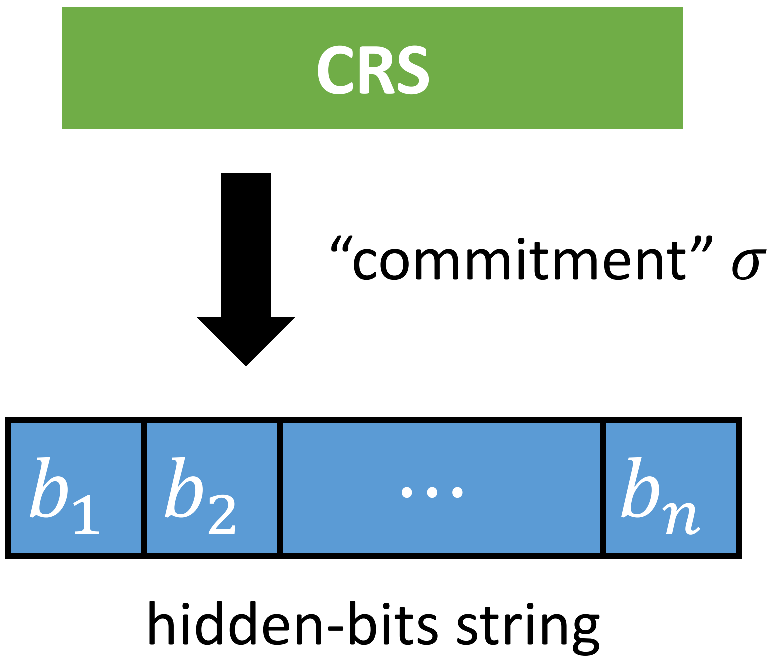
[FLS90]

Main properties:

- **Binding:** Can only open σ to a single bit for each position
- **Hiding:** Unopened bits should be hidden
- **Succinctness:** $|\sigma| \ll n$

Soundness: If $|\sigma| \ll n$ and there are not too many “bad” hidden-bits strings $\Rightarrow$ prover cannot find a “bad” σ that fools verifier

Zero-Knowledge: Unopened bits hidden to verifier

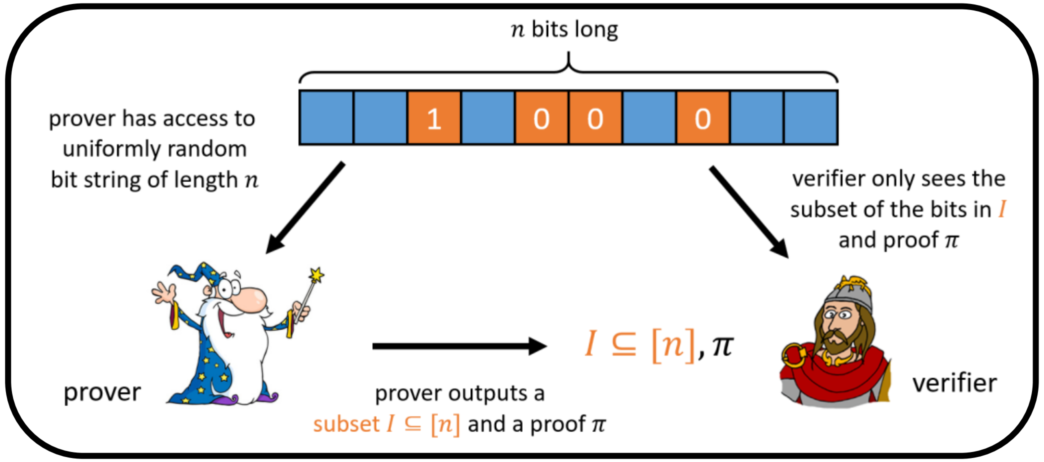


Prover can selectively open σ to (i, b_i) for indices i of its choosing

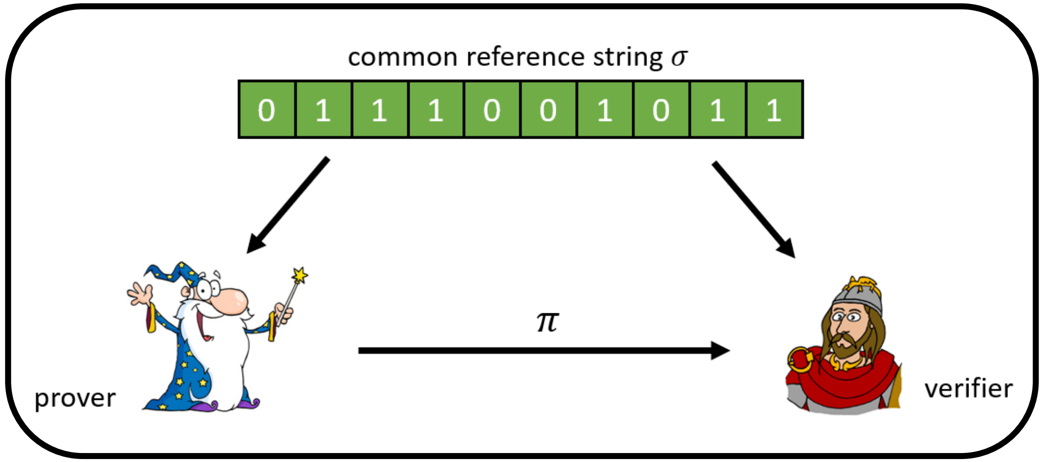
The FLS Compiler

[FLS90]

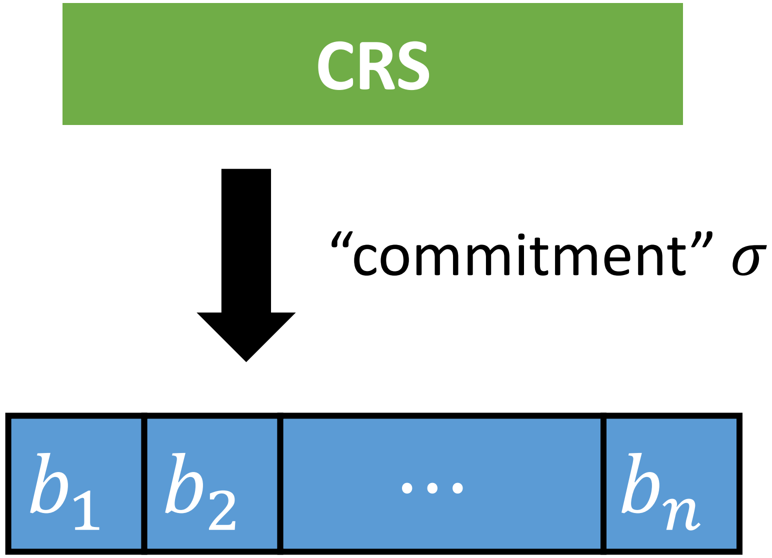
NIZKs in the hidden-bits model



cryptographic compiler



NIZKs in the CRS model



hidden-bits string

Instantiations:

[FLS90]: trapdoor permutations (computational NIZK proofs)

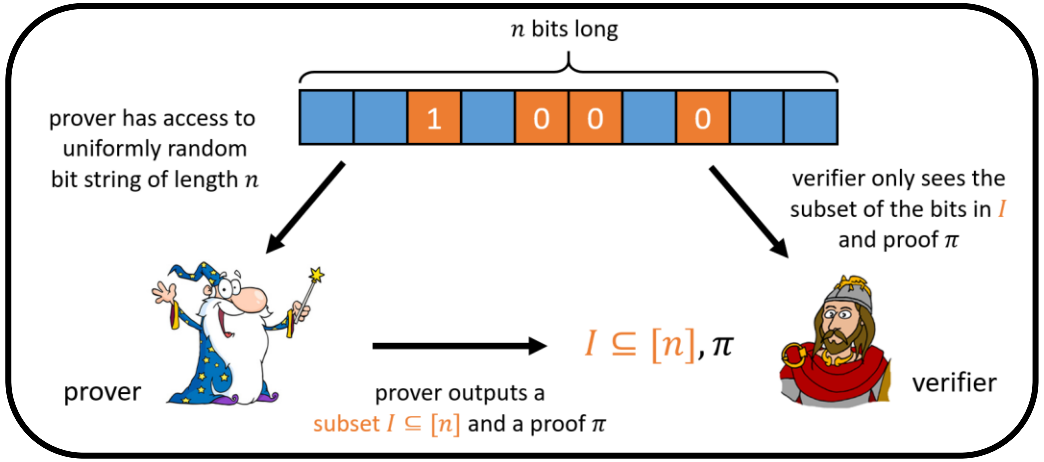
[CHK03]: CDH over a pairing group (computational NIZK proofs)

[QRW19, CH19, KNYY19]: hidden-bits generators from CDH (computational DV-NIZK proofs)

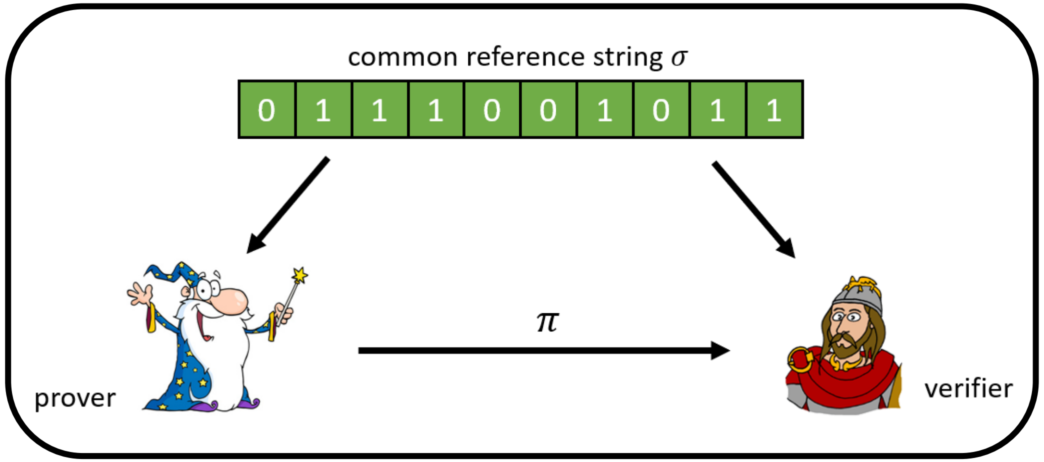
The FLS Compiler

[FLS90]

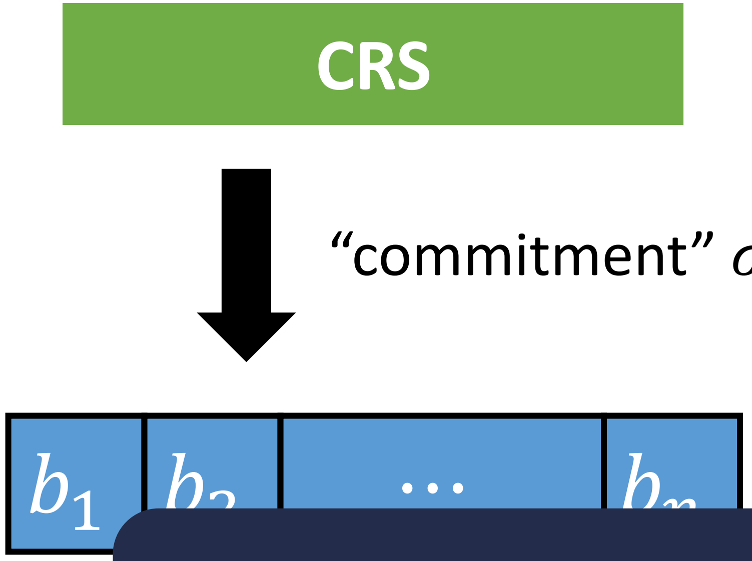
NIZKs in the hidden-bits model



cryptographic compiler



NIZKs in the CRS model



Possible to instantiate FLS to obtain statistical ZK?

Instantiations:

[FLS90]: trapdoor permutations (computational NIZK proofs)

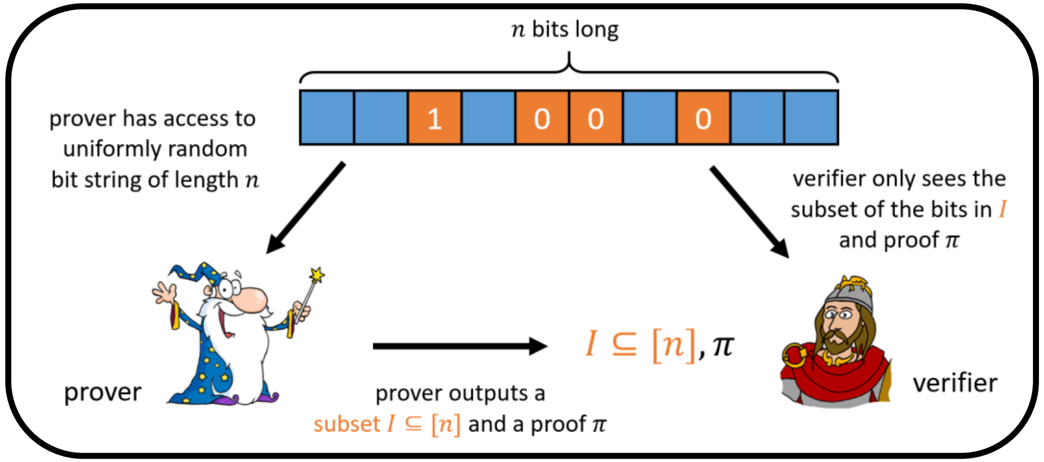
[CHK03]:CDH over a pairing group (computational NIZK proofs)

[QRW19, CH19, KNYY19]:hidden-bits generators from CDH (computational DV-NIZK proofs)

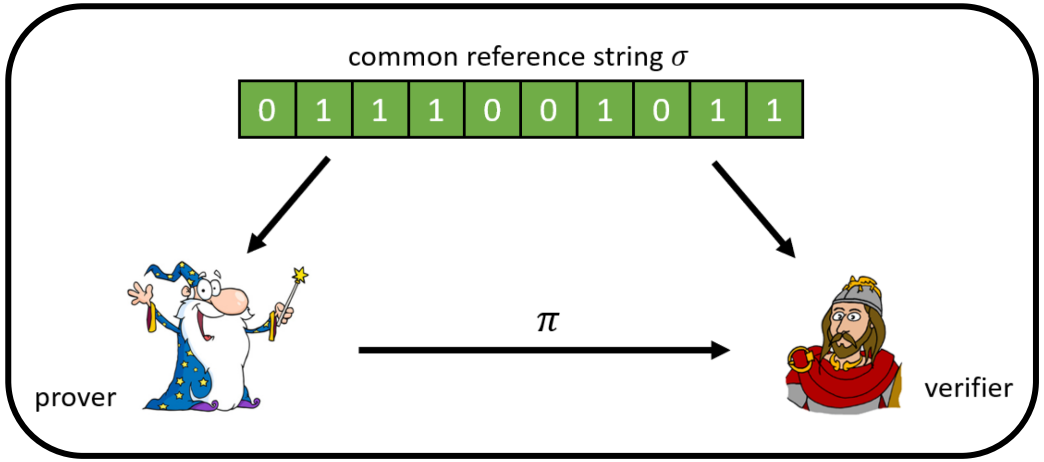
The FLS Compiler

[FLS90]

NIZKs in the hidden-bits model



NIZKs in the CRS model



This work: dual-mode hidden bits generator

- "Binding mode:" computational DV-NIZK proofs
- "Hiding mode:" statistical DV-NIZK arguments

Warm-Up: The FLS Compiler from CDH

[CHK03, QRW19, CH19, KNYY19]

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $g, h_1 = g^{w_1}, \ldots, h_n = g^{w_n} \in \mathbb{G}$ $\quad w_1, \ldots, w_n \leftarrow \mathbb{Z}_p$

Each exponent $y \in \mathbb{Z}_p$ defines a hidden bits string

b_1	b_2	$\cdots$	b_n

$b_i \coloneqq \mathrm{hc}\left(h_i^y\right)$

hard-core bit

Committing to a hidden-bits string:

Prover samples $y \leftarrow \mathbb{Z}_p$ and commits to hidden bits string with $\sigma = g^y \in \mathbb{G}$

Opening $\boldsymbol{\sigma}$ to a bit $\boldsymbol{b_i}$: reveal h_i^y and prove that $\left(g, g^y, h_i, h_i^y\right)$ is a DDH tuple

[CHK03]: Use a pairing: $e(g^y, h_i) = e\left(g, h_i^y\right)$ — **publicly-verifiable**

[QRW19, CH19, KNYY19]: Use Cramer-Shoup hash-proof system [CS98, CS02, CKS08] — **designated-verifier**

Warm-Up: The FLS Compiler from CDH

[CHK03, QRW19, CH19, KNYY19]

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $g, h_1 = g^{w_1}, \dots, h_n = g^{w_n} \in \mathbb{G}$ $\quad w_1, \dots, w_n \leftarrow \mathbb{Z}_p$

Each exponent $y \in \mathbb{Z}_p$ defines a hidden bits string

$$\boxed{b_1}\boxed{b_2}\boxed{\quad\cdots\quad}\boxed{b_n}$$

hard-core bit

$b_i \coloneqq \mathrm{hc}\left(h_i^y\right)$

Committing to a hidden-bits string:

Prover samples $y \leftarrow \mathbb{Z}_p$ and commits to hidden bits string with $\sigma = g^y \in \mathbb{G}$

Statistical binding: choice of σ (with $h_1, \dots, h_n$) <u>completely</u> defines $b_1, \dots, b_n$

Resulting NIZK satisfies statistical soundness

Warm-Up: The FLS Compiler from CDH

[CHK03, QRW19, CH19, KNYY19]

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $g, h_1 = g^{w_1}, \dots, h_n = g^{w_n} \in \mathbb{G}$

$w_1, \dots, w_n \leftarrow \mathbb{Z}_p$

Each exponent $y \in \mathbb{Z}_p$ defines a hidden bits string

$b_1 \quad b_2 \quad \cdots \quad b_n$

hard-core bit

$b_i \coloneqq \mathrm{hc}\left(h_i^y\right)$

Committing to a hidden-bits string:

Prover samples $y \leftarrow \mathbb{Z}_p$ and commits to hidden bits str

Need to compute $g^{w_i y}$ from g^{w_i} and g^y which is precisely CDH

Computational hiding: unopened bits computationally hidden since hc is hard-core

Resulting NIZK satisfies computational zero-knowledge

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

$[\boldsymbol{v}]$ plays the role of the family g

$[\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ play the role of $g^{w_1}, \dots, g^{w_n}$

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

Key idea: replace scalars in the CRS with vectors

Notation: for a vector $\boldsymbol{v} \in \mathbb{Z}_p^n$, we write $[\boldsymbol{v}] \coloneqq (g^{v_1}, \dots, g^{v_n})$

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

$[\boldsymbol{v}]$ plays the role of the generator g

$[\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ play the role of $g^{w_1}, \dots, g^{w_n}$

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

Observation: under DDH, these two distributions for $\boldsymbol{w}_i$ are computationally indistinguishable

similar principle as used to construct lossy PKE from DDH [HJR16]

Notation: for a vector $\boldsymbol{v} \in \mathbb{Z}_p^n$, we write $[\boldsymbol{v}] \coloneqq (g^{v_1}, \dots, g^{v_n})$

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

Each vector $\boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ defines a hidden bits string

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

b_1	b_2	$\cdots$	b_n

$b_i \coloneqq H([\boldsymbol{y}^T \boldsymbol{w}_i])$

$H : \mathbb{G} \to \{0,1\}$ is universal hash

Prover's commitment: $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}] \in \mathbb{G}$

Statistically binding in binding mode: choice of σ (and CRS) completely defines $b_1, \dots, b_n$

$$\boldsymbol{y}^T \boldsymbol{w}_i = s_i \boldsymbol{y}^T \boldsymbol{v} = s_i \sigma$$

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

Each vector $\boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ defines a hidden bits string

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

b_1	b_2	$\cdots$	b_n

$b_i \coloneqq H([\boldsymbol{y}^T \boldsymbol{w}_i])$

$H: \mathbb{G} \to \{0,1\}$ is universal hash

Prover's commitment: $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}] \in \mathbb{G}$

Statistically hiding in hiding mode: choice of σ (and CRS) <u>completely</u> hides $b_1, \dots, b_n$

if $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$ are linearly independent and $\boldsymbol{y} \leftarrow \mathbb{Z}_p^{n+1}$, $\boldsymbol{y}^T \boldsymbol{w}_i$ is <u>uniform</u> given $\boldsymbol{y}^T \boldsymbol{v}$, $\boldsymbol{y}^T \boldsymbol{w}_j$ for $j \neq i$

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

Each vector $\boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ defines a hidden bits string

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

b_1	b_2	$\cdots$	b_n

$b_i \coloneqq H([\boldsymbol{y}^T \boldsymbol{w}_i])$

$H: \mathbb{G} \to \{0,1\}$ is universal hash

Prover's commitment: $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}] \in \mathbb{G}$

Binding mode $\Rightarrow$ statistically-binding hidden bits $\Rightarrow$ statistical soundness

Hiding mode $\Rightarrow$ statistically-hiding hidden bits $\Rightarrow$ statistical zero-knowledge

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \ldots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

Each vector $\boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ defines a hidden bits string

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

b_1	b_2	$\cdots$	b_n

$b_i \coloneqq H([\boldsymbol{y}^T \boldsymbol{w}_i])$

$H: \mathbb{G} \to \{0,1\}$ is universal hash

Prover's commitment: $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}] \in \mathbb{G}$

Remaining ingredient: need a way for prover to open commitments to hidden bits

To open the commitment $[\sigma]$ to value b_i, prover sends $[t_i] = [\boldsymbol{y}^T \boldsymbol{w}_i]$ together with a proof that $\exists \boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ such that $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}]$ and $[t_i] = [\boldsymbol{y}^T \boldsymbol{w}_i]$

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

Each vector $\boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ defines a hidden bits string

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

b_1	b_2	$\cdots$	b_n

$b_i \coloneqq H([\boldsymbol{y}^T \boldsymbol{w}_i])$

$H: \mathbb{G} \to \{0,1\}$ is universal hash

Prover's commitment: $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}] \in \mathbb{G}$

Remaining ingredient: Can use Cramer-Shoup techniques nts to hidden bits

To open the commitme

together with a proof that $\exists \boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ such that $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}]$ and $[t_i] = [\boldsymbol{y}^T \boldsymbol{w}_i]$

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

Each vector $\boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ defines a hidden bits string

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

$b_1 \mid b_2 \mid \cdots \mid b_n$

$b_i \coloneqq H([\boldsymbol{y}^T \boldsymbol{w}_i])$

Prover's commitment: $[\sigma] = [\boldsymbol{y}^T \boldsymbol{v}] \in \mathbb{G}$

Prover's opening: $[t_i] = [\boldsymbol{y}^T \boldsymbol{w}_i]$

proof that $\exists \boldsymbol{y} \in \mathbb{Z}_p^{n+1} : [\sigma] = [\boldsymbol{y}^T \boldsymbol{v}]$ and $[t_i] = [\boldsymbol{y}^T \boldsymbol{w}_i]$

Implication: dual-mode DV-NIZK from DDH

- **Binding mode:** computational NIZK proofs
- **Hiding mode:** statistical NIZK arguments

Dual-Mode Instantiation from DDH

Ingredient: let $\mathbb{G}$ be a prime-group of order p with generator g

CRS: $[\boldsymbol{v}], [\boldsymbol{w}_1], \dots, [\boldsymbol{w}_n]$ where $\boldsymbol{v}, \boldsymbol{w}_1, \dots, \boldsymbol{w}_n \in \mathbb{Z}_p^{n+1}$

$\boldsymbol{v} \leftarrow \mathbb{Z}_p^{n+1}$

Each vector $\boldsymbol{y} \in \mathbb{Z}_p^{n+1}$ defines a hidden bits string

Two distributions for $\boldsymbol{w}_i$:

- **Binding mode**: $\boldsymbol{w}_i \leftarrow s_i \boldsymbol{v}$ where $s_i \leftarrow \mathbb{Z}_p$
- **Hiding mode:** $\boldsymbol{w}_i \leftarrow \mathbb{Z}_p^{n+1}$

b_1	b_2	$\cdots$	b_n

$b_i \coloneqq H([\boldsymbol{y}^T \boldsymbol{w}_i])$

Extensions:

- Replace DDH with k-Lin family of assumptions (for any $k \geq 1$)
- Replace DDH with subgroup indistinguishability assumptions (e.g., QR/DCR)
- Use a pairing to <u>publicly</u> implement verification
  - Yields statistical NIZK argument (*not* dual-mode) from k-Lin ($\mathbb{G}_1$) and k-KerLin ($\mathbb{G}_2$)

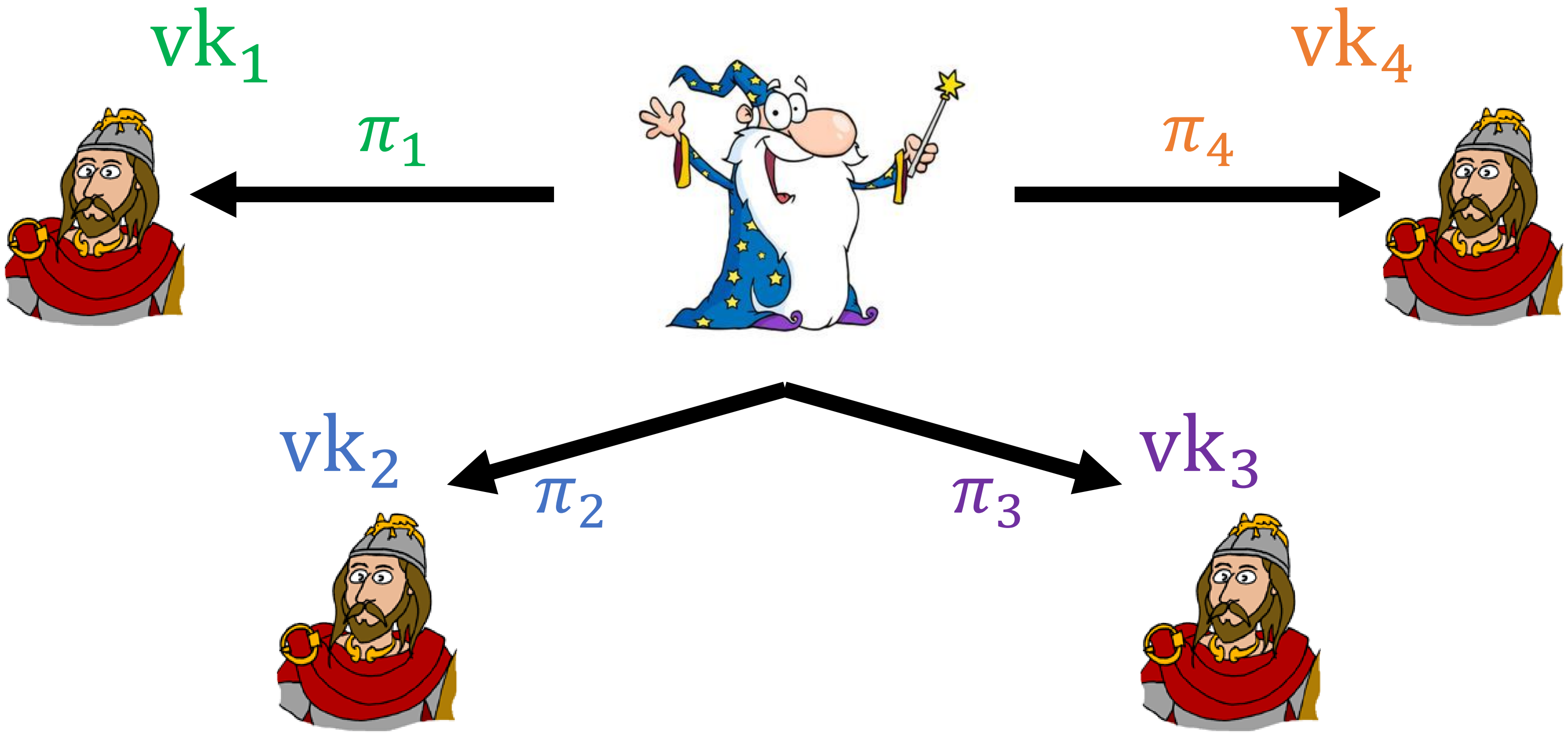
Malicious Designated-Verifier Security

[QRW19]

common <u>random</u> string

111010011011111100110110000001

<u>only</u>
trusted setup



verifiers can choose their own verification key;
zero-knowledge should hold even if vk_i chosen maliciously

Malicious Designated-Verifier Security

[QRW19]

common random string

111010011011111001101100000001

only

trusted setup

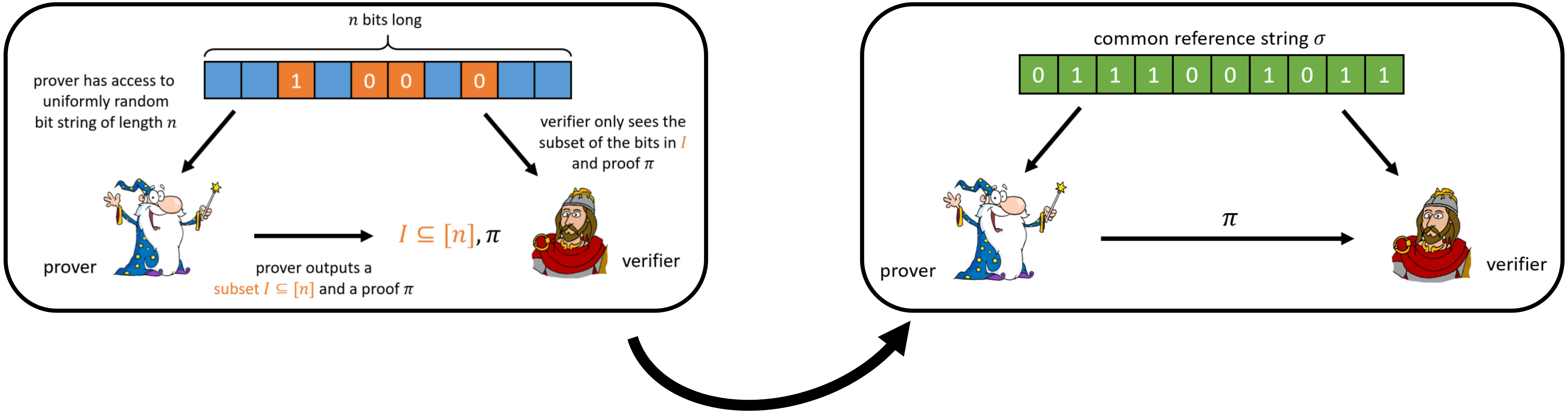
All of our DV-NIZK constructions easily adapted to satisfy malicious security (MDV-NIZKs)

- Technique similar to [QRW19], but relies on a linear independence argument rather than a rewinding argument
- **[QRW19]:** computational MDV-NIZK proofs from "one-more CDH"
- **This work:** dual-mode MDV-NIZKs from DDH (or k-Lin) / QR / DCR

[see paper for details]

Summary

NIZKs in the hidden-bits model | NIZKs in the CRS model



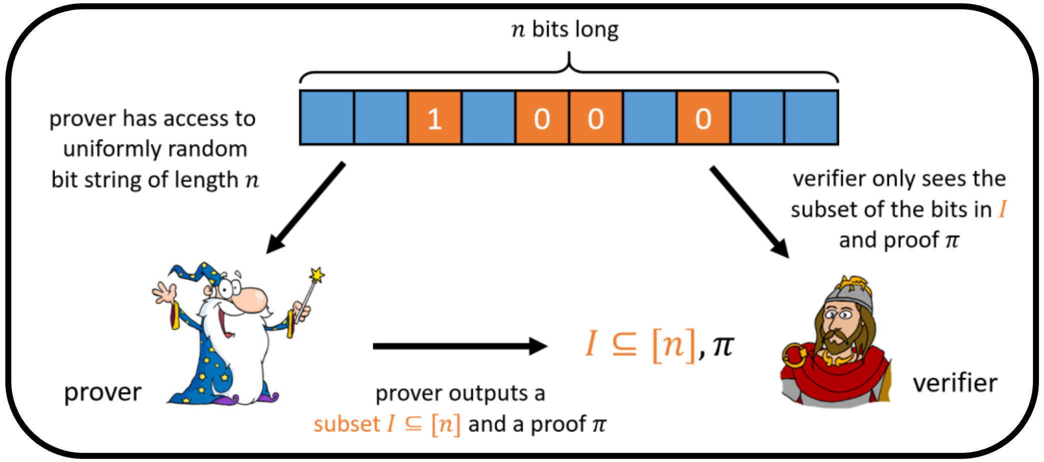
cryptographic compiler

This work: Leverage the FLS compiler to achieve <u>statistical</u> zero-knowledge

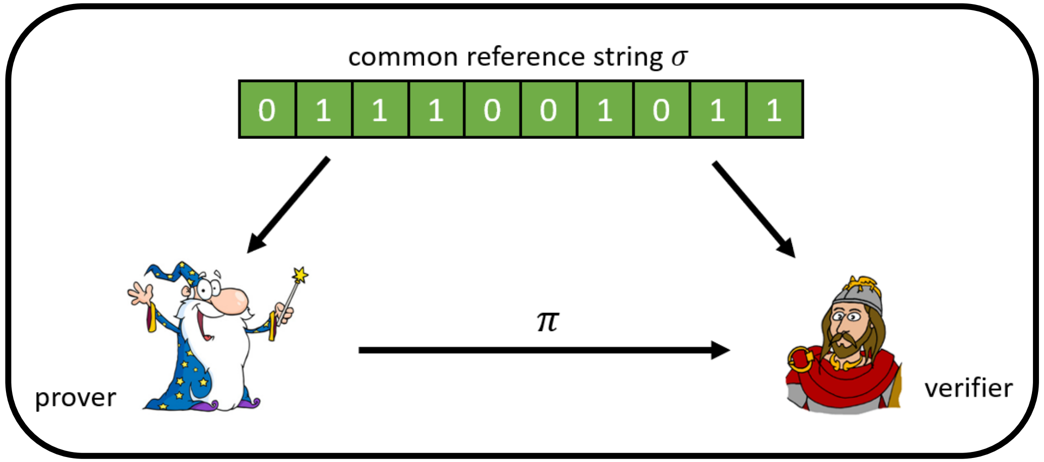
- Dual-mode malicious DV-NIZKs from k-Lin in pairing-free groups / QR / DCR
- Statistical NIZKs from k-Lin ($\mathbb{G}_1$) + k-KerLin ($\mathbb{G}_2$) in a pairing group

Open Questions

NIZKs in the hidden-bits model



NIZKs in the CRS model



Statistical NIZK arguments from factoring?

- **[FLS90]:** computational NIZK proofs from factoring
- **This work:** dual-mode malicious DV-NIZKs from QR / DCR

Other assumptions: Statistical (DV)-NIZKs from LPN? from CDH?

The Landscape of (DV)-NIZKs

Construction	Soundness	Zero-Knowledge	Assumption
[FLS90]	statistical	computational	factoring
[CHK03]	statistical	computational	CDH (pairing group)
[GOS06]	stat. | comp.	comp. | stat.	k-Lin $(\mathbb{G}_1, \mathbb{G}_2)$
This work	**computational**	**statistical**	**k-Lin $(\mathbb{G}_1)$, k-KerLin $(\mathbb{G}_2)$**
[PS19]	stat. | comp.	comp. | stat.	LWE
[SW14]	computational	statistical	iO + OWFs
			publicly-verifiable
[QRW19, CH19, KNYY19]	statistical	computational	CDH
[LQRWW19]	computational	computational	CDH/LWE/LPN
[CDIKLOV19]	stat. | comp.	comp. | stat.	DCR
This work	**stat. | comp.**	**comp. | stat.**	**DDH/QR/DCR**
			malicious designated-verifier

Thank you!

https://eprint.iacr.org/2020/265