Laconic Zero Knowledge to Public Key Cryptography

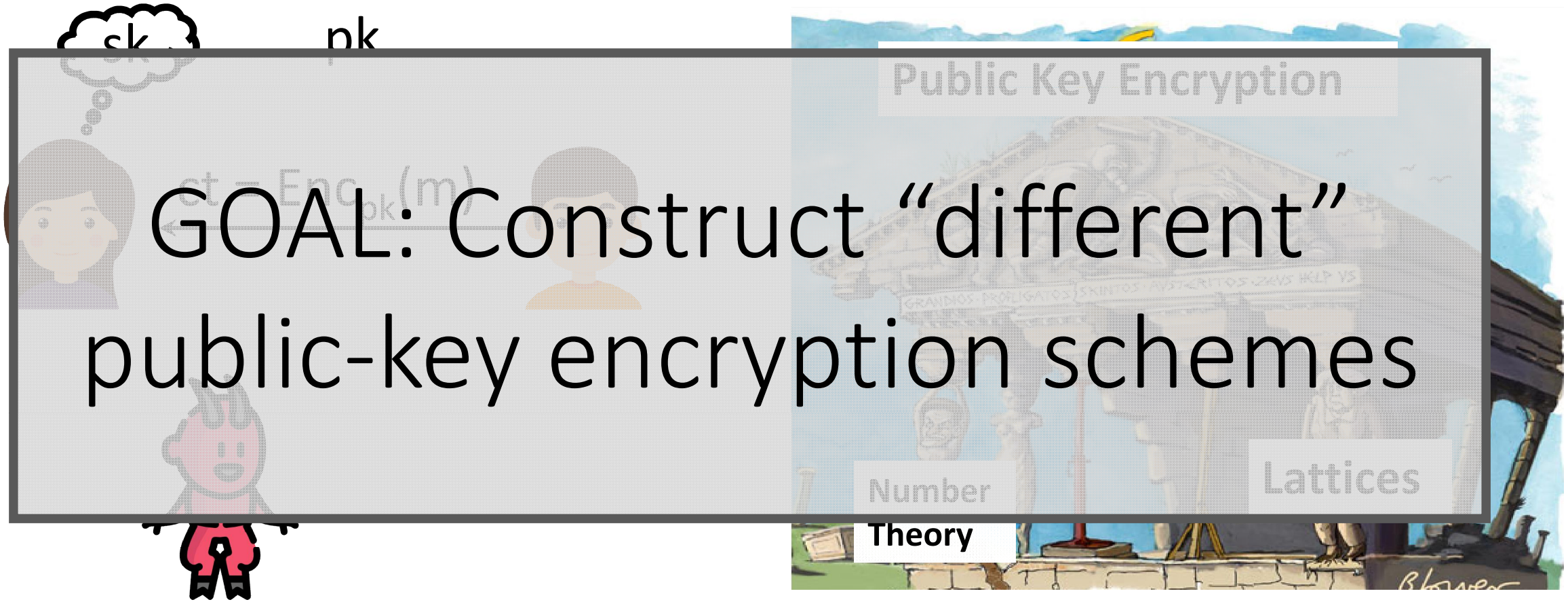
Akshay Degwekar
(MIT)

Joint work with Itay Berman (MIT), Ron Rothblum (MIT → Technion)

and Prashant Nalini Vasudevan (MIT → Berkeley).

Public Key Encryption (PKE)

[Diffie-Hellman76, Rivest-Shamir-Adelman78, Goldwasser-Micali82]



GOAL: Construct "different" public-key encryption schemes

What structure+hardness implies public-key crypto?

Possible answers:

NP-hardness

- ✗ No Crypto Known
- ✗ Some impossibility results [Brassard79, Feigenbaum-Fortnow93, Bogdanov-Trevisan03, Goldreich-Goldwasser98, AkaviaGoldreichGoldwasserMoshkovitz06]

One-Way Functions

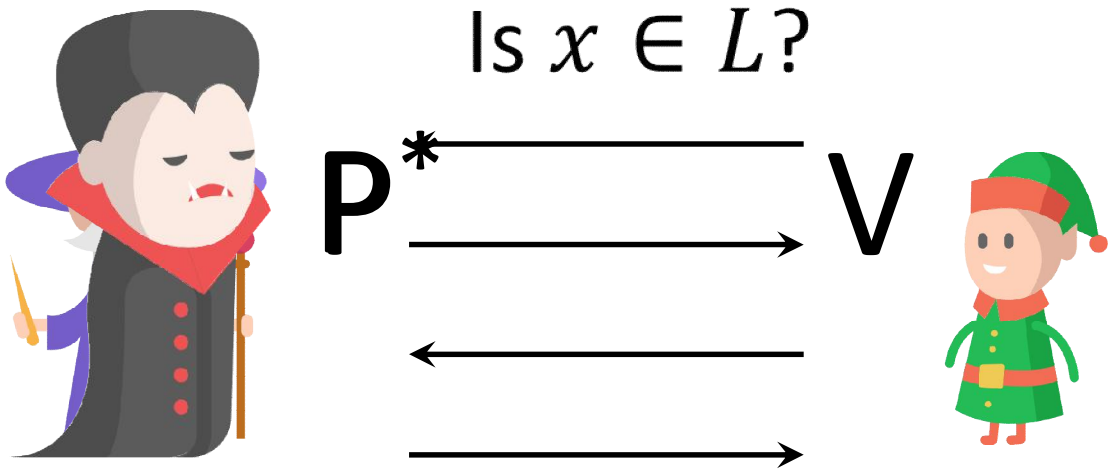
- ✗ Some barriers [Impagliazzo-Rudich89, Brakerski-Katz-Segev-Yerukhimovich11, Dachman-Soled16, Garg-Hajiabadi-Mahmoody-Mohammed18]

SZK-hardness (SZK = Statistical Zero Knowledge)

- ✓ Implies OWFs [Ostrovsky91]
- ✓ Many problems in SZK imply PKE

Statistical Zero Knowledge (SZK)

[Goldwasser-Micali-Rackoff85]



Completeness:

$$x \in L \implies \text{V accepts}$$

Soundness:

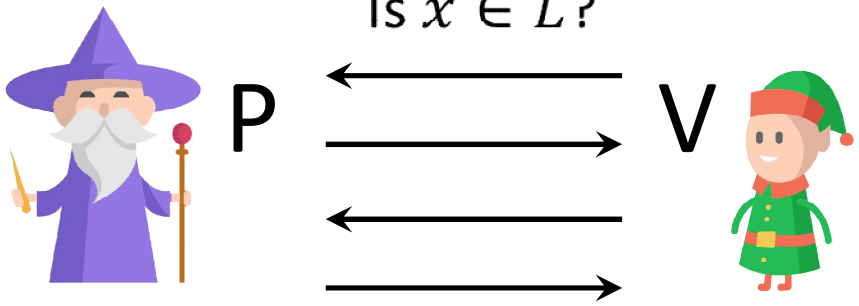
$$x \notin L \implies \text{V rejects w.h.p}$$

Proof : All powerful P^*

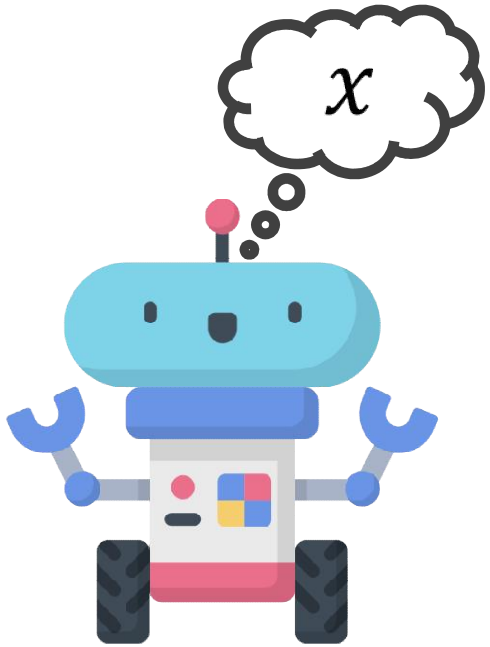
Argument : Efficient P^*

Honest-Verifier Statistical Zero Knowledge:

[Goldwasser-Micali-Rackoff85]



$\approx_s$

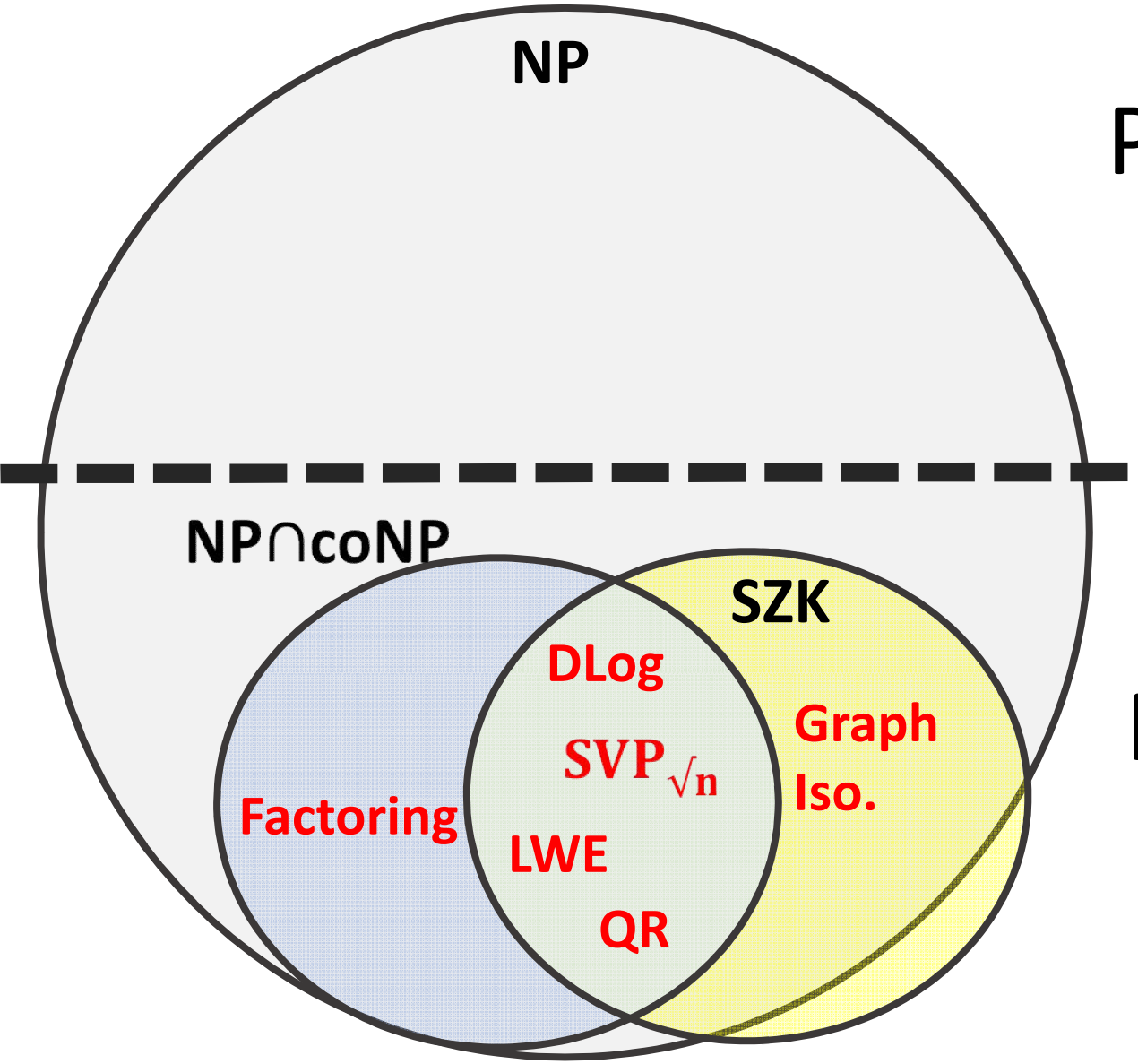


Simulator:

Exists S, for $x \in L$

$$\mathrm{P} \leftrightharpoons \mathrm{V}(x) \approx_s \mathrm{S}(x)$$

Statistical Zero Knowledge



PKE from SZK-Hardness?

Seems Challenging:

Discrete Log, Graph Iso have SZK proofs but no PKE known.

Need more Structure?

Example: Quadratic Non-Residuosity

(Or: From GMR to GM)

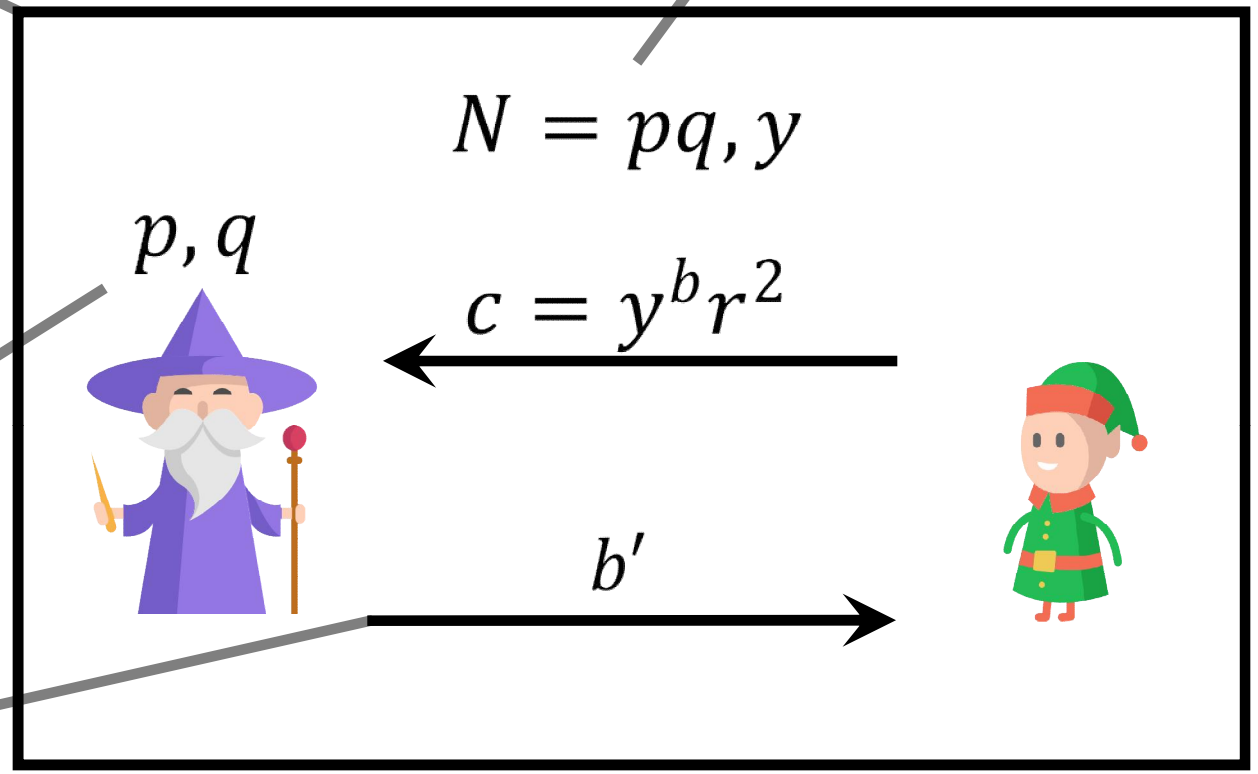
[Goldwasser-Micali82, Goldwasser-Micali-Rackoff85]

(Honest-Verifier)
Statistical
Zero-Knowledge
Proof

Efficient
Prover

Prover talks
very little

Can sample
hard instances
w/ witnesses



Our Results: These Properties are Sufficient!

ZK PROOF SYSTEM

Honest Verifier SZK Argument

- Efficient-Prover
- Laconic $O(\log^{1/3} n)$

$\Longrightarrow$

Public-Key Encryption

+

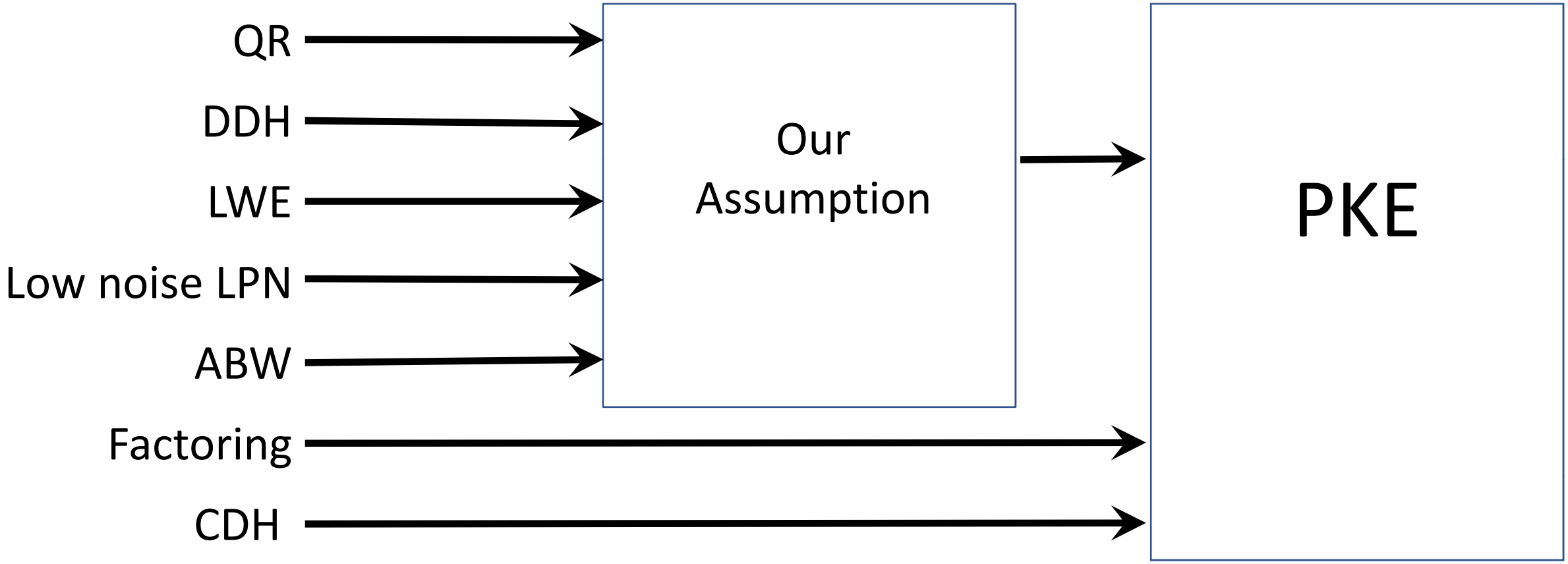
CRYPTO HARD LANGUAGE

YES dist $\approx$ NO dist

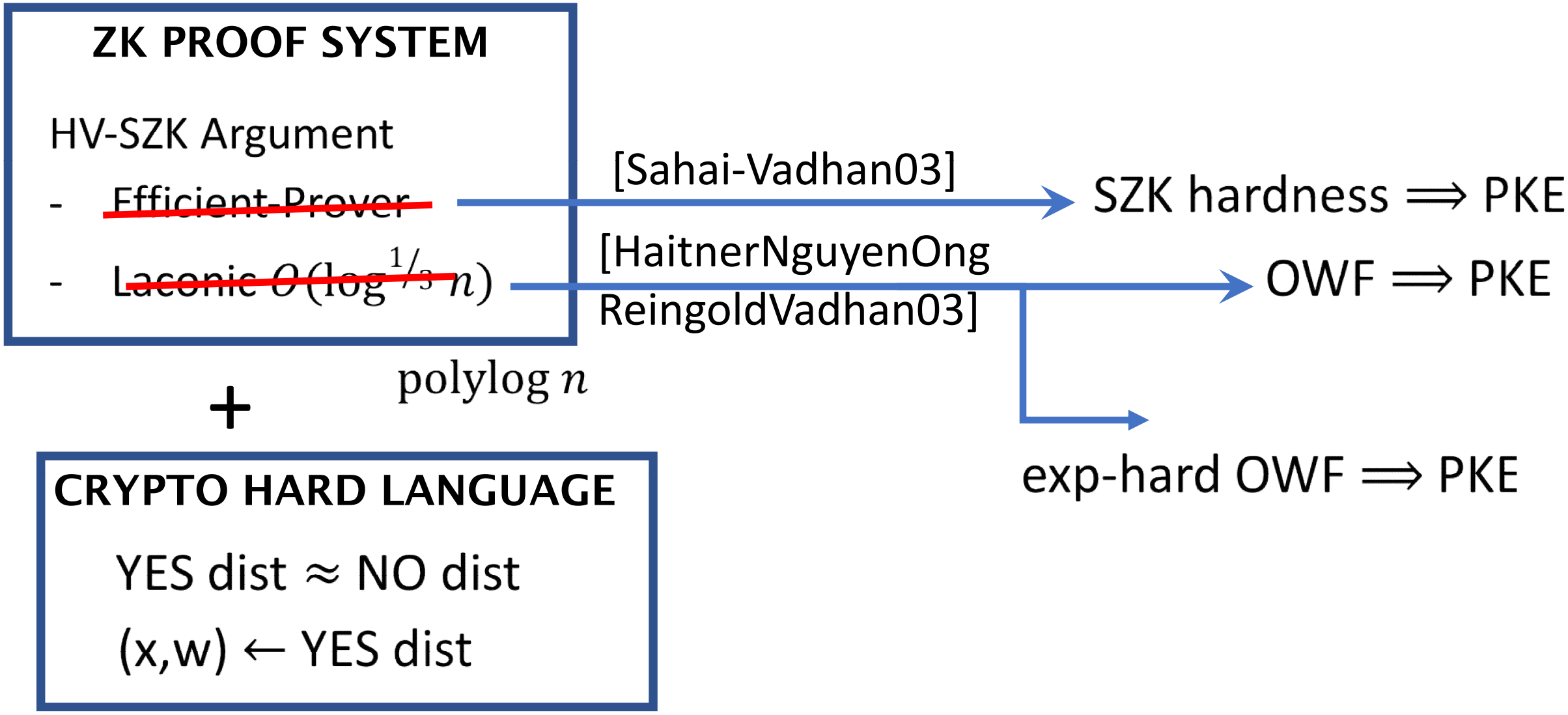
(x,w) $\leftarrow$ YES dist

Implies One-Way Functions

Instantiations



Perspective: Relaxing the Assumption



Characterization

WEAK

ZK PROOF SYSTEM

HV-SZK Argument

- Efficient-Prover
- Laconic $O(\log^{1/3} n)$

WEAK: soundness, completeness hold on average

$\Rightarrow$

Public-Key Encryption

\+ **DISTRIBUTIONS**

CRYPTO HARD ~~LANGUAGE~~

YES dist $\approx$ NO dist

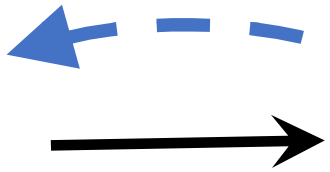
(x,w) $\leftarrow$ YES dist

Summary

Laconic, Efficient Prover,
HVSZK ARGUMENT

+

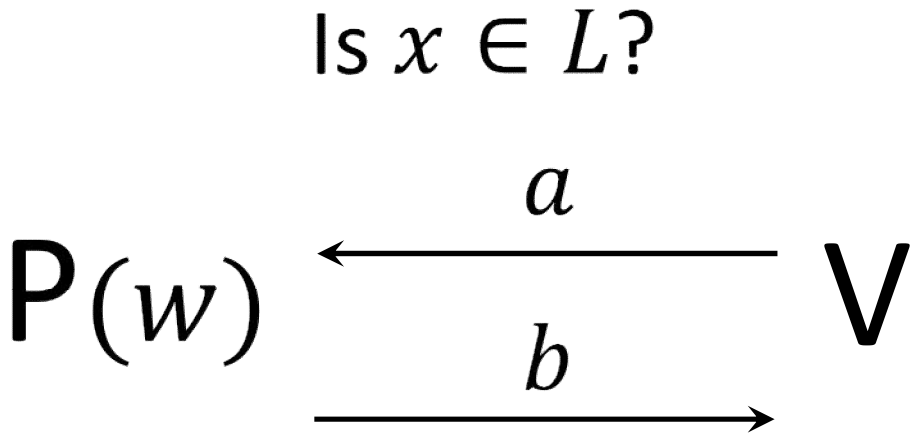
CRYPTO HARD LANGUAGE



Public Key
Encryption

Techniques

Warmup: 2-Msg, Deterministic Prover*

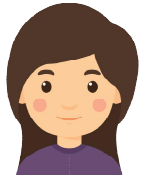


To Simplify:
Constant Soundness error s
Perfect Completeness,
Perfect Zero Knowledge.

$$(a', b') \leftarrow \mathrm{Sim}(x)$$

* a.k.a Hash Proof System [Cramer-Shoup02]

Weak Key Agreement



$(x, w) \leftarrow \text{YES}$ $\xrightarrow{\quad x \quad}$ $(a', b') \leftarrow \text{Sim}(x)$

$b = P(x, w, a')$ $\xleftarrow{\quad a' \quad}$ Outputs b'

Output b

Correctness:

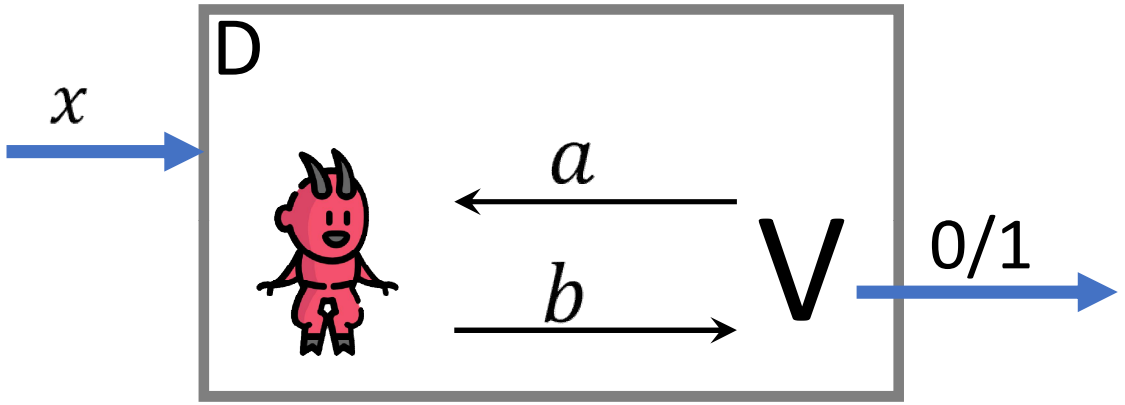
Deterministic prover $\Rightarrow$ Every verifier challenge has unique prover response

Perfect ZK $\Rightarrow$ Sim's output b' is same as Prover's output b

Claim: Weak Security:  cannot predict b' with prob $> s$

Break average-case hardness

Adv = Cheating Prover



$$x \in L \Longrightarrow \Pr(D(x) = 1) \geq \Pr\left(\text{Eve predicts } b'\right) > s$$

$$x \notin L \xrightarrow{\text{soundness}} \Pr(D(x) = 1) < s$$

Contradiction. D breaks average-case hardness.

Amplify from weak PKE to PKE using HolensteinRenner05

We saw: PKE from deterministic, 2-msg SZK Proof System.

Challenges:

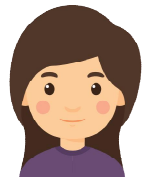
Randomized Prover

Multi-round Proof System

Stateful Prover

Lesser Challenges: Relaxing perfect ZK, perfect completeness

Coping with Randomized Provers



$(x, w) \leftarrow \text{YES}$ $\xrightarrow{x}$ $(a', b') \leftarrow \text{Sim}(x)$

Outputs $b = P(x, w, a')$ $\xleftarrow{a'}$ Outputs b'

Weak Security: ✓

Correctness: ✗

Our Assumption → **Trapdoor Pseudoentropy Generator** → **PKE**

Our Assumption → Trapdoor Pseudoentropy Generator → PKE

$$pk, sk$$

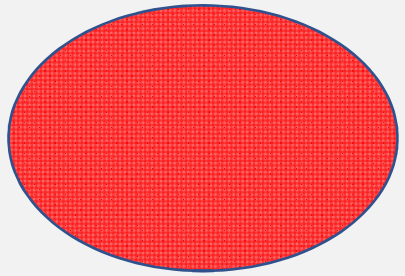
$$pm, sm \leftarrow \mathbf{Encode}(pk) \qquad ct, b \leftarrow \mathbf{Encrypt}(pk)$$

$$sm' \leftarrow \mathbf{Decode}(sk, pm) \qquad b = \mathbf{Decrypt}(sk, ct)$$

pm **does not** fix sm

Decode does not recover sm. Samples from dist.

Security: Adv can only sample from "bigger" dist.



Formalized using pseudoentropy [HILL99]

Our Assumption → Trapdoor Pseudoentropy Generator → PKE


$(x, w) \leftarrow \text{YES}$ $\quad$ $(a', b') \leftarrow \text{Sim(x)}$

x →

← a'

Outputs $b = P(x, w, a')$ $\quad$ Outputs b'

$(pk, sk) = (x, w)$

Encode: $(a', b') \leftarrow \text{Sim(x)}$

pm, sm

Decode: Output $P(x, w, a')$

Challenges: Many rounds

[Ostrovsky 91] Terminate at random round.

Stateful Prover

Laconic. Rejection Sampling

Amplification Theorem

Technically difficult half

Uses connections between
Pseudorandomness & Unpredictability

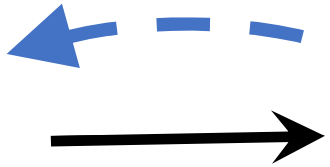
Ingredients from:
OWFs => PRG (HILL99, VadhanZheng12)

Conclusion and Open Problems

Laconic, Efficient Prover,
HVSZK ARGUMENT

+

CRYPTO HARD LANGUAGE



Public Key
Encryption

Big Open Q:

Design new PKE schemes

Thank You!

Trapdoor Pseudoentropy Generator

$pk, sk \leftarrow \mathbf{Gen}$

$pm, sm \leftarrow \mathbf{Encode}(pk)$

$sm' \leftarrow \mathbf{Decode}(sk, pm)$

Public Key Encryption

$pk, sk \leftarrow \mathbf{Gen}$

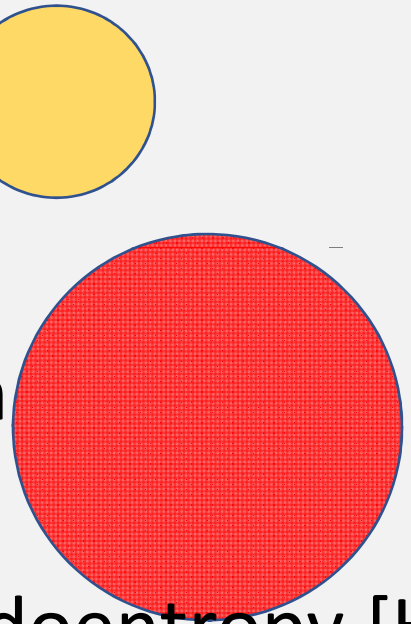
$ct, b \leftarrow \mathbf{Encrypt}(pk)$

$b = \mathbf{Decrypt}(sk, ct)$

pm **does not** fix sm

Decode samples

Security: Gap between Decode & adversary



ct fixes bit b

Decrypt outputs b

Security:
b is completely hidden

Formalized using pseudoentropy [HILL99]